Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

von der Fakultät für
Wirtschaftswissenschaften
des Karlsruhe Institut für Technologie (KIT)
genehmigte
DISSERTATION

Consumer Preferences and Bid-Price Control for Cloud Services

von

Dipl.-Inform.Wirt Arun Anandasivam

Tag der mündlichen Prüfung: 15.06.2010

Referent: Prof. Dr. Christof Weinhardt

Korreferent: Prof. Dr. Stefan Tai

2010 Karlsruhe

Abstract

Nowadays, instead of investing into large and expensive IT infrastructures, it is possible to buy computing environments or (complex) electronic services on-demand in order to perform certain business activities. This approach is typically known as Cloud Computing. For consumers, Cloud Computing embraces a service-oriented architecture (SOA) and implies potential for reduced total cost of ownership, great flexibility as well as reduced information technology overhead (Vouk, 2008). Consequently, one of the most relevant advantages is the increased technical and financial flexibility for Cloud Computing consumers.

Consumers have various requirements regarding Cloud services depending on their industry and business, and hence are different with regard to their preferences and valuations. Providers of Cloud services can be limited in their resource capacities or flexibility, and they are exposed to the challenge of how to sell their services efficiently. To address distinctive consumer preferences, service providers can offer numerous classes of services according to service level agreements (SLA). These classes of services are priced differently. Furthermore, providers allocating their capacity to various types of consumers have incentives to maximize output, which is the revenue yielded by the sale of capacity units. Revenue Management deals with complex decision problems concerning sales, demand, and pricing of services from a provider's perspective (Talluri and van Ryzin, 2004b). Recently, the consumer's perspective also becomes increasingly important to enable an integrated view of the complex interaction between the provider and the consumer.

In this thesis both views are examined. At first, the applicability of Revenue Management methods in Cloud Computing is discussed. The properties of Cloud services are analyzed and compared to the requirements of Revenue Management. Current literature does not consider Revenue Management in Clouds and how this may impact the design of services. Secondly, Revenue Management concepts like accept/reject policies, dynamic pricing or advance reservation are not always accepted by the consumers. Consumers in some domains are not used to dynamic prices and occasionally deny it. In 2000, Amazon for example failed to introduce dynamic pricing for the online store due to customer complaints about the frequent price changes (Weiss and Mehrotra, 2001). A survey as a part of this thesis was conducted to un-

derstand the consumers' perception of Revenue Management methods to avoid an unsuccessful adoption like Amazon, since no research work has covered this topic in Cloud Computing so far. The design of sophisticated services and computation of the appropriate price requires an interpretation of consumers' preferences and requirements (Chellappa and Gupta, 2002). Conjoint analysis was chosen as a research method to elicit the preferences and the results were analyzed statistically. The results of the survey foster the application of Revenue Management concepts in Clouds. Furthermore, the survey revealed on the Infrastructure-as-a-Service (IaaS) level that operating system and price are the most important factors of the IaaS offer. Consumers seem to prefer a well-known platform instead of comparing the availability rate of different providers or between the services of one provider and choosing the cheapest one. Providers can design their customized services based on these results.

After knowing the consumers' preferences and designing the services appropriately, the provider may face the problem of how to price the services and how this pricing may impact the resource utilization. One aspect of this problem is how Cloud service providers would decide to accept or reject requests for services when the resources for offering these services become scarce. A decision support policy called *Customized Bid-Price Policy (CBPP)* is proposed in this thesis to decide efficiently, when a large number of (complex) services can be offered over a finite time horizon. This heuristic outperforms well-known policies, if interior prices cannot be updated frequently during incoming requests and an automated update of bid prices is required to achieve more accurate decisions. Since CBPP approximates the revenue offline before the requests occur, it has a low runtime compared to other approaches during the online phase (when requests appear). The performance is examined via simulation and the pre-eminence of CBPP is statistically proven.

Acknowledgements

Completing a thesis is a large project, which is not possible without the support of and collaboration with many people. First and foremost, I am indebted to my advisor Professor Dr. Christof Weinhardt, who was responsible for starting this Ph.D. He gave me the great opportunity to gain a lot of experience through the European Project SORMA and encouraged me in my ideas. I would also like to thank Professor Dr. Dirk Neumann for continuously supporting me in my work. I have benefited from his valuable advices. I am also grateful to the members of my thesis committee: Professor Dr. Stefan Tai as the co-advisor asked me challenging technical questions and provided me with good insights and research suggestions, Professor Dr. Frank Schultmann and Professor Dr. Karl-Heinz Waldmann have been so kind to be part of the board of examiners.

My sincere thanks go out to my colleagues from the Information & Market Engineering group, who constantly gave me feedback on my work and improved my work enormously. In particular, I thank Dr. Jochen Stoesser, Carsten Block and Nikolay Borissov for improving my research and implementation skills. I would like to express my gratitude to Dr. Benjamin Blau, Dr. Simon Caton and Christian Haas for proof-reading parts of my thesis and giving me valuable comments, which improved this work at hand a lot. Moreover, not only in the office but also during social activities the positive atmosphere of the entire team always motivated me to continue my research and project work at this institute. During my research stay at the University of Melbourne, it was a pleasure for me to work with Professor Dr. Rajkumar Buyya and his team on interesting research topics. Special thanks to Saurabh Garg, Dr. Srikumar Venugopal and Dr. Chee Shin Yeo for their support. Thanks also go to Dr. Simon See, who invited me to Sun Microsystems in Singapore; the research stay was a great experience and you have an excellent team.

Finally, I would like to thank the people who have influenced me the most: my parents Kanagaratnam and Ahila for their unconditioned love and their hearts of gold, my brother Ramann for the fruitful and long discussions beside research as well as my wife Kathrin for her love, her patience and for optimizing my work-life balance. Without you this project would not have been possible. I dedicate this thesis to you.

Contents

I Foundations **1**

1 Introduction **3**
1.1 Background & Motivation . . . 3
1.2 Objectives & Contributions . . . 6
1.3 Thesis Structure . . . 8

2 Preliminaries **9**
2.1 IT Infrastructure and Service Paradigms . . . 9
2.1.1 Clouds, Grids, and their Predecessors . . . 10
2.1.2 Cloud Computing Definitions . . . 14
2.1.3 Cloud Services . . . 21
2.2 Traditional Revenue Management . . . 27
2.2.1 Capacity Control for Single Leg . . . 29
2.2.2 Network Model of Capacity Control . . . 31
2.2.3 Customer Choice Model . . . 32
2.3 Research Approach . . . 35
2.3.1 Towards Revenue Managed Clouds . . . 35
2.3.2 Applied Methodologies and Research Questions . . . 42

II Model Design, Implementation and Evaluation **45**

3 Customer Choice in Clouds **47**
3.1 Introduction . . . 47
3.2 Related Work . . . 48
3.2.1 Previous Surveys . . . 49
3.2.2 Conjoint Analysis . . . 50
3.2.3 Customer Choice Models . . . 54
3.3 Model Formulation . . . 58
3.3.1 Theoretical Background for the Survey Questions . . . 59
3.3.2 Hypotheses . . . 62
3.4 Customer Choice Survey . . . 64
3.4.1 Survey Design . . . 64
3.4.2 Preferences, Stimuli and Data Collection Method . . . 66

3.4.3 Choice Set for the Conjoint Analysis . . . 70
3.5 Results & Implications . . . 74
3.5.1 Descriptive Results . . . 74
3.5.2 Inductive Results . . . 79
3.5.3 Implication . . . 82

4 Capacity Management in Clouds **85**
4.1 Introduction . . . 85
4.2 Related Work . . . 87
4.3 Optimization Approach . . . 88
4.3.1 Bid Price Control . . . 92
4.3.2 Customized Bid Price Policy . . . 96
4.3.3 Non Optimal Outcome . . . 100
4.4 Simulation Environment . . . 106
4.4.1 Genetic Algorithm . . . 106
4.4.2 Simulation Process and Hypotheses . . . 111
4.5 Results & Implications . . . 115
4.5.1 Statistical Results . . . 115
4.5.2 Sensitivity Analysis . . . 120
4.5.3 Implication . . . 123

III Finale **127**

5 Conclusion and Outlook **129**
5.1 Summary of Contributions . . . 129
5.2 Integrated View of the Results . . . 132
5.3 Future Work . . . 133

IV Appendix **137**

A Customer Choice Survey **139**
A.1 Questionnaire . . . 139
A.2 Profile Cards . . . 145
A.3 Theoretical Models Overview . . . 149

B Genetic Algorithm **153**

Bibliography **155**

List of Figures

1.1 Structure of this thesis 7

2.1 Three service layer architecture in the Cloud. 25
2.2 Revenue Management process flow in the e-Business context (adapted from Talluri and van Ryzin (2004b) and Bichler et al. (2002)). 30
2.3 An example for the dependencies between services and resources on the IaaS layer. 31
2.4 Requirements for applying Revenue Management. 36
2.5 Outage examples from January 2008 to June 2009 41

3.1 Comparison of compositional and decompositional models (Hahn, 1997) 51
3.2 Characteristics of the participants in the survey 75
3.3 Reasons for changing current provider 76
3.4 Expected price clustered by each user group for hourly and monthly prices 77

4.1 Incoming requests for different services in different timeslots 89
4.2 Demand definition in a finite time 92
4.3 Runtime and revenue analysis with different number of population and evolution steps for a 4x5 setting. 110
4.4 The three simulation phase 113
4.5 Revenue for 50 different runs 116
4.6 Bid price development over time for each resource (4x4 setting). . . . 123

A.1 Websurvey Questionnaire Part 1 140
A.2 Websurvey Questionnaire Part 2 141
A.3 Websurvey Questionnaire Part 3 142
A.4 Websurvey Questionnaire Part 4 143
A.5 Websurvey Questionnaire Part 5 144
A.6 Websurvey Questionnaire Part 6 145
A.7 Websurvey Conjoint Analysis Part 1 146
A.8 Websurvey Conjoint Analysis Part 2 147
A.9 Websurvey Conjoint Analysis Part 3 148

List of Tables

2.1 Comparison of different computing concepts 20

3.1 Steps for operating a conjoint analysis (Green and Srinivasan, 1990) . 53
3.2 Impact of consumer behavior, provider offer and trading object factors on the Revenue Management requirements 62
3.3 Attributes and their levels 72
3.4 Utility estimates and standard error rates for each level 80
3.5 Hypotheses results 81
3.6 Crosstabulation based on participants' valuation for a one hour instance and their usage predictability 82

4.1 Dual variables example of the DLP problem 94
4.2 Resource usage by services in Clouds 100
4.3 Non-optimality example 1 - service-resource mapping and service prices 101
4.4 Non-optimality example 1 - available capacity for acceptance of one more request 101
4.5 Non-optimality example 1 - capacity left at $t = 1$ after acceptance of service i 101
4.6 Non-optimality example 1 - set of bid prices 102
4.7 Non-optimality example 2 - requests arriving 104
4.8 Non-optimality example 2 - possible strategies in example 2 105
4.9 Standard deviation for all settings 110
4.10 Results for Hypothesis 4.1: Revenue difference between CBPP and CEC-S (* denotes significance at the level of $p = 0.05$, ** at $p = 0.01$, and *** at $p = 0.001$.) 117
4.11 Results for Hypothesis 4.1: Revenue difference between CBPP and DLP-S (* denotes significance at the level of $p = 0.05$, ** at $p = 0.01$, and *** at $p = 0.001$.) 118
4.12 Results for Hypothesis 4.2: small price changes of a specific service affect revenue difference between CBPP and CEC-S 119
4.13 Results for Hypothesis 4.2: small price changes of a specific service affect revenue difference between CBPP and DLP-S 119
4.14 Results for Hypothesis 4.3: Upper bounds affect revenue 119
4.15 Service-resource mapping and price in the basic scenario 121
4.16 Price variations for randomly selected scenarios 122

4.17 Capacity occupation for specific scenarios in variation 21 and 25 at the end of the period 122

A.1 Overview of the related customer choice models from Revenue Management (part 1/2) 150

A.2 Overview of the related customer choice models from Revenue Management (part 2/2) 151

B.1 Analysis of upper bound settings on outcome of CBPP 153

List of Abbreviations

ANOVA	Analysis Of Variance	52
API	Application Programming Interface	19
CAPEX	Capital Expenditure	59
CBPP	Customized Bid-Price Policy	7
CDN	Content Delivery Network	75
CEC	Certainty Equivalent Control	95
CIO	Chief Information Officer	50
CPU	Central Processing Unit	27
CRM	Customer Relationship Management	3
DCR	Demand-to-Capacity ratio	106
DLP	Deterministic Linear Programming	88
EMSR	Expected Marginal Seat Revenue	30
IaaS	Infrastructure-as-a-Service	16
IT	Information Technology	3
NCC	Network Capacity Control	29
NHPP	Non Homogeneous Poisson Process	124
NIST	National Institute of Standards and Technology	19
OLS	Ordinary Least Squares	68
OPEX	Operating Expenditure	59
PaaS	Platform-as-a-Service	16
PC	Personal Computer	10
REST	Representational State Transfer	18
RLP	Randomized Linear Programming	92
SaaS	Software-as-a-Service	6
SABP	Self-Adjusting Bid-Price	96
SLA	Service Level Agreement	14
SME	Small and Medium Enterprise	4
SOA	Service-oriented Architecture	4
SOAP	Simple Object Access Protocol	18
VM	Virtual Machine	12
VO	Virtual Organization	12

List of Symbols

h Resource 90
i Service 90
m Total number of resources 89
n Total number of services 89
q_i Protection level for service i 29
c_h Total capacity of a resource 90
$\bar{c}_{ht}$ Amount of capacity reserved at time t 90
r_i Price for service i 90
D_{iT} Total expected demand of service i 91
$\hat{D}_{it}$ Demand arrived until time t 91
D_{it} Expected demand from t until end of the booking period 91
a_{hi} Amount of resource h used by one unit of service i 90

Part I

Foundations

Chapter 1

Introduction

The problems of Revenue Management are as old as business itself. [...] The true innovation of Revenue Management lies in the method of decision making.

[Talluri and van Ryzin, 2004]

1.1 Background & Motivation

Throughout every business sector, Information Technology (IT) is the backbone providing almost instantly the right information at the right time to the requesting user. Recently, services based on IT systems are offered over the Internet and start to replace desktop applications. Salesforce.com[1] as a prominent example allows users to manage their business contacts online with their Customer Relationship Management (CRM) software by accessing the application through a web browser. The upcoming paradigm termed *Cloud Computing* comprises of services offered over the Internet from basic computational services like virtual instances of operating systems (e.g. Amazon Web Services[2] or 3Tera[3]) to complex services such as Jamcracker Services Delivery Network[4] aggregating and enhancing basic services. According to Armbrust et al. (2009) the original idea behind Cloud Computing, i.e. computing as a utility, was already mentioned by Parkhill (1966) and has now become a reality through new technologies such as Web 2.0, virtualization and the interconnected business world via the Internet.

The impact of Cloud Computing on the worldwide IT industry is very significant. A survey conducted by IDC[5] predicts that the worldwide IT spending on

[1] Salesforce.com (`http://www.salesforce.com`)
[2] Amazon Web Services (`http://aws.amazon.com`)
[3] 3Tera (`http://www.3tera.com`)
[4] Jamcracker Services Delivery Network (`http://www.jamcracker.com/Platform`)
[5] IDC (`www.idc.com`)

services in Clouds will increase substantially and is expected to reach US $ 42 billion by 2012 (Gens, 2008). Moreover, 50% of the respondents indicate that the main reason for using Clouds is to save costs, and almost 20% intend to apply Cloud-based services to quickly implement new business processes. Instead of investing in large and expensive IT infrastructures, Small and Medium Enterprises (SMEs) especially benefit from buying computing environments or Cloud services in order to perform certain business activities. Studies have revealed that data centers were only using 10% to 35% of their actual processing power and 50% to 60% of storage is wasted due to lack of viable, large-scale utility model (Carr, 2005). This situation enables possibilities for new markets. Cloud Computing providers offer their available resources and try to make a profit by using economies of scale (Boss et al., 2007). Consequently, one of the most relevant advantages is the increased flexibility for Cloud consumers. They are able to buy services on-demand and no longer have to pay for the development and maintenance of their own IT infrastructure anymore. Cloud Computing embraces a Service-oriented Architecture (SOA) and implies the potential for a reduced total cost of ownership, increased flexibility as well as reduced IT overhead for the end-user (Vouk, 2008). The usage of Cloud services will improve the business operations of consumers, thus allowing them to invest in core business activities and not into supporting IT systems. The risk of supporting IT infrastructure operations thereby shifts towards the providers of IT data processing centers. Cloud providers who face these risks have to find ways to increase their revenues. For example, the pay-as-you-go model benefits consumers but leads to unmanageable utilization patterns.

Consumers have varying requirements depending on the nature of their industry and business, and hence are different with regard to their preferences and valuations. Providers as well as resellers of Cloud services can be limited in their resources or basic service capacities, and are thus exposed to the challenge of how to sell their capacities efficiently. To address distinctive consumer preferences, service providers can offer numerous classes of services according to service-level agreements. These classes of services are priced differently. Furthermore, providers allocating their capacity to various types of consumers have incentives to maximize output, which is the revenue yielded by the sale of capacity units. Sophisticated metering and pricing systems as well as flexible services tailored to the needs of heterogeneous consumers are the key factors in successfully designing and offering utility models like Cloud services (Albaugh and Madduri, 2004).

Revenue Management addresses the problem of allocating a company's capacity to its customers in order to find a revenue maximizing allocation (Williamson, 1992). Much literature on Revenue Management is concerned with the airline industry. A typical situation there is the sale of different types of flight tickets at different times and at different fares. By offering different fare classes (e.g. economy class, business

class, and first class) the airline company can segment its customers. The capacity of an airline is the amount of seats available on a certain flight on a certain date. Capacity can be allocated flexibly according to the different fare classes.

A company is exposed to numerous complex decisions, for example, how to sell its capacity, how to set the prices for a certain period of time, or how to segment its customers. Revenue Management deals with complex decision making processes involving sales, demand, and pricing (Talluri and van Ryzin, 2004b). These decision making problems often have to be solved in an uncertain environment due to incomplete information and dynamic changes in the customers' demand. Several factors influence a company's revenue. Managing demand plays a key role from a company's point of view. The term Revenue Management reveals that its goal is to maximize a company's revenue by making suitable demand management and pricing decisions. Revenue Management comprises methods for finding the right decisions concerning demand, prices, and sales. Kimes (1989) emphasizes four characteristics of Yield or Revenue Management: "Yield Management is the application of information systems and pricing strategies to allocate the *right* capacity to the *right* customer at the *right* place at the *right* time." Talluri and van Ryzin (2004b) give a classification for typical decision making problems in Revenue Management:

- **Structural decision making problems** deal with issues concerning how to sell services and which channels to use (e.g. auction or posted price) or how to choose market segmentation. Decisions concerning the bundling of services or committing to certain prices in advance due to advertisement have a major impact on the overall business and the revenue of a company.
- **Decisions on quantity** are concerned with allocating capacity to certain customer segments. Furthermore, the decisionof when to sell a service to a certain consumer or when to cancel the sale is a key component. In a scarce service market the request of a consumer can either be accepted or rejected in favor of a better paying consumer or a higher valued service request.
- **Pricing decisions** are natural methods in the wide area of Revenue Management. Pricing is a key factor to control demand as well as revenue. It determines how much to charge for different groups of customers, how to adjust prices over a given sales period, or how to grant discounts to customers.

These decision making problems are rather complex. It strongly depends on the industry, the field of operation, and the application context, which category of decision is the most appropriate for a certain enterprise. In practice, a company's Revenue Management concept may incorporate all three categories of decisions. Due to possible but sometimes complex adaptations Revenue Management is not limited

to the airline industry. Many research papers have proven that these concepts can be applied to other domains such as hotels (Vinod, 2004), restaurants (Sheryl, 2003), car rental (Carroll and Grimes, 1995) or Internet service providers (Nair and Bapna, 2001). In this thesis, the first two decision making problems are examined, whereby the pricing decision is disregarded.

1.2 Objectives & Contributions

The Revenue Management framework provides promising tools and methods to increase revenue of Cloud service providers. Research papers from Dube et al. (2005), Urgaonkar et al. (2002) or Nair and Bapna (2001) have analyzed the application of certain Revenue Management methods to the IT domain in general. Currently, there is no literature examining the applicability of Revenue Management methods to Cloud services. The Revenue Management literature has identified several requirements, when its methods are applicable to a certain domain. Cloud services have to fulfill some properties like perishability or consumer heterogeneity so that Revenue Management methods and tools can be applied in Clouds (see Section 2.3.1). However, the previously mentioned papers have not analyzed the requirements for applying Revenue Management methods to the IT domain. Hence, one objective of this thesis is to analyze the requirements of Revenue Management, classify these requirements according to their applicability to consumer and provider and to scrutinize the coherency with the Cloud service properties. This approach will help to understand how the Revenue Management methods will influence the design of Cloud services.

The emergence of Cloud Computing was mainly driven by new technologies like virtualization. It enables the provision of computing power or Software-as-a-Service (SaaS) over the Internet without the burden of a costly infrastructure for the consumer. However, the consumer demand for these kinds of service has not been thoroughly analyzed before. Around year 2000 the "application service provider" failed to create a viable business model (Desai and Currie, 2003; Currie, 2004). The authors identified that standard applications did not provide any competitive advantage in the market and tailor-made applications were to costly for the consumers. Consumer needs were disregarded to a certain extent. Designing services to match consumer needs is a well-known method to increase market share and profitability (Anderson and Sullivan, 1993; Heskett, 1990). The SaaS paradigm, put into practice by companies like Salesforce.com, was only a first step towards Cloud Computing. The question arises what consumers prefer and how the infrastructure, platform or software services can enhance their utility. This thesis focuses on the infrastructure layer and analyzes consumers' preferences. A better understanding of preferences

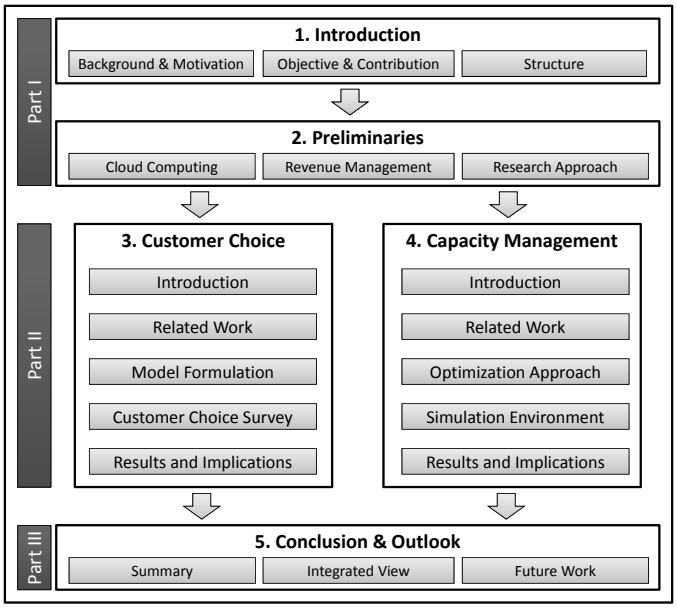


Figure 1.1: Structure of this thesis

allows Cloud service providers to precisely design services at the infrastructure level and appropriately set the prices for these services. A survey was conducted to empirically examine the consumer behavior (see Chapter 3) using conjoint analysis as a research method.

From a provider's perspective, Revenue Management methods help to efficiently manage the resource utilization, price the services more accurately and to increase the revenue. The optimization problem for allocating the right services to the right customer at the right time and place is an NP-hard problem, since a Bellman equation has to be solved. Existing heuristics approach this problem by deciding to allocate the resources to the service in the online phase, when requests occur. Klein (2007) proposed a novel idea of having an automated update of the bid prices. The bid prices of services are a threshold value, whether a service for a certain price should be sold or not. First, the parameters for the optimal allocation are calculated in an offline phase. Then, a linear function automatically updates the bid prices in the online phase. This thesis scrutinizes and extends this idea known from the airline business and shows how the automated bid price update, which is called Customized Bid-Price Policy (CBPP), can be applied to Clouds. Furthermore, different demand scenarios can be evaluated via CBPP to determine good price settings for the services.

1.3 Thesis Structure

This thesis comprises five chapters (Figure 1.1). After the introduction in Chapter 1, the preliminaries in Chapter 2 provide the basis to understand the relation between Cloud Computing and Revenue Management. Section 2.1 describes the difference between Cloud Computing and other paradigms such as Grid Computing. It further discusses the properties of Cloud services. In Section 2.2, the basics of Revenue Management relevant for this thesis are briefly explained. Section 2.3 discusses the applicability of Revenue Management methods and tools for Cloud services. Further, the research questions that constitute this thesis are derived.

Part II is divided into two self-contained chapters and commences with the discussion of the research questions. Chapter 3 analyzes the preferences of Cloud service consumers. The theoretical foundation is derived from the Revenue Management literature. The conjoint analysis method and the survey design are described in detail and the choice sets for the survey are defined. The results from the survey are analyzed descriptively and statistically.
Chapter 4 focuses on the capacity management of computing resources for Cloud service providers. After explaining the Cloud scenario, the application of Revenue Management heuristics for capacity management is discussed in detail. The CBPP is proposed to process the rapidly incoming requests in Clouds and to provide a combination of an offline algorithm for the complex calculation and an online algorithm for an automated update of the bid prices. The applied research method is a simulation-based optimization. In both chapters the research questions defined in Section 2.3 are evaluated and answered.

Part III concludes this thesis and examines the integrated view of the results from Chapter 3 and 4. It explains why these results are relevant for Cloud service providers and raises further interesting research questions.

Parts of the results presented in this thesis were published in Anandasivam and Neumann (2009), Anandasivam and Premm (2009), Anandasivam et al. (2010) and Anandasivam and Weinhardt (2010).

Chapter 2

Preliminaries

It is the new and different that is always most vulnerable to market research

[Malcolm Gladwell, 2005]

This chapter explains the dependency between Cloud Computing and Revenue Management. At first, the Cloud Computing concept is set in contrast to other distributed computing paradigms. Subsequently, services in the Cloud are discussed and the specific characteristics are outlined. In Section 2.2 the general Revenue Management framework is presented and the areas of interest are briefly described. Finally, both streams are consolidated and the appropriate research methods as well as the research questions are discussed in Section 2.3.

2.1 IT Infrastructure and Service Paradigms

Historically – before the emergence of personal computers in the 1970's – time-sharing services were widely spread, allowing access to computing machinery to those without their own mainframe. The idea of these systems was to rent input-output equipment instead of a computer. Bemer (1957) introduced the concept of time-sharing systems already in 1957 derived from his favorite hamburger restaurant where orders were put into a revolving drum with elastic bands and the cooks were often following this lottery principle. McCarthy assumed in 1961 that "computation may someday be organized as a public utility just as the telephone is a public utility" (Ivanov, 2008; Foster et al., 2008). The era of time-sharing systems was followed by the invention of personal computers, which provided more freedom to users. Individuals use their own software and customize the system according to their needs. Today's concept of Cloud Computing is not a return to the architecture of time-sharing systems, but in fact a reversal of the long continuing trend of individualization; users are willing to give up possession of data centers and let service

providers control the operation, support and development. This section aims at providing a common background for understanding the ideas behind Cloud Computing and how the term is derived from existing terminology like Cluster Computing, Voluntary Computing or Grid Computing[1] (Stoesser, 2009).

2.1.1 Clouds, Grids, and their Predecessors

Cluster Computing:
After mainframes became essential for the daily business of enterprises and academia, Cluster Computing was a new paradigm in the 1980's and emerged more prominently with the success of Personal Computers (PCs). Clusters are a pool of dedicated single computing systems working together as a single, integrated computing resource and are accessible via PCs to execute computing jobs (Bell and Gray, 2002; Baker et al., 1999). Traditional applications especially built for mainframes were ported to PCs in order to facilitate programming and management. Furthermore, the computer user was allowed great latitude for creating applications and having access to computing resources not only on a time-shared basis. PCs provided resources for calculation on a small scale, while clusters were exploited for jobs with large resource demand.

Scalability, availability and cost effectiveness were the three major challenges for a successful collaborative computing environment (Fox et al., 1997). Cluster Computing helped to master these three obstacles providing scalability for parallel workloads due to the mostly homogeneous[2], large-scale system, and allowing ad hoc scaling by adding commodity hardware. High availability is secured through independent nodes and the capability to perform upgrades by only disabling the nodes that need to be replaced. Cost effectiveness is achieved by using commodity hardware that is robust and stable. The performance of the clusters was comparable to the existing mainframes, but the cost for a cluster was much lower (Baker et al., 1999).

Summing up, Clusters are managed centrally within an organization and are physically placed at a single location. Each workstation has to share its resources with other workstations. The systems are homogeneous and lend themselves to easily adapt and implement changes. The organization prefers to offer a higher quality of service by keeping the failure rate and response time low.

[1] Terms like utility computing or on-demand computing are not considered separately, since the idea is very similar to Cloud on the infrastructure level and these terms have not been defined and discussed widely in literature.

[2] One counterexample would be the Beowulf cluster (`http://www.beowulf.org/`) with a heterogeneous environment. In general, the underlying hardware can be heterogeneous due to virtualization. However, the thesis refers to the idea of clusters in its beginning

Volunteer Computing:
In academia, researchers take a great interest in working together on resource-demanding computer simulations. Academic institutes face two problems. They are limited in their budget for buying hardware resources and payment for each submitted job as an incentive for contributing resources is not an option. Thus, they prefer to cooperate and share resources on a voluntary basis. The management of the Cluster system is centralized. Hence, computing resources can be declared as dedicated to one specific job. In Volunteer Computing the resources are shared among different organizations or individuals. A prominent example is the SETI@Home[3] project with the goal to detect extraterrestrial life by analyzing signals from a radio telescope. Volunteers have to install a small application on their PC to provide their idle resources to the project (Sarmenta, 2001). In 2001, this project had 2.4 million clients to process 1.1 billion candidate signals (Werthimer et al., 2001).

Although the management of the jobs and the application to run the jobs is accomplished on a central server in Volunteer Computing, the computing jobs are executed on geographically distributed machines with different operating systems and hardware specification (Bonorden et al., 2006). A certain quality level is not guaranteed and jobs are only executed if idle resources are available. Hence, resource availability in Volunteer Computing is unpredictable in contrast to Cluster Computing, where resources are dedicated to the computer users. The distributed resources are not under a centralized control. The central entity managing the application can only access the distributed resources if the contributor authorizes it. Another distinction to Cluster Computing is that the type of applications are limited to batch jobs. Interactive applications cannot be realized, since neither a guaranteed execution time nor a deadline is available. In a Volunteer Computing scenario, there is one consumer, namely the central entity, and various providers, i.e. the participants, who like to contribute resources to the project.

Grid Computing:
Parallel to the evolution of Volunteer Computing, the idea of Grid Computing emerged in 1998 proposed by Foster and Kesselman (1998) to utilize geographically distributed computing resources. In the beginning the development of the infrastructure was driven by scientific applications. The major areas of application are, similar to previously described concepts, computationally-intensive scientific, mathematical, and academic problems on the one hand. However, Grid Computing is also relevant to companies performing large-scale simulations in research & development departments of commercial enterprises and data processing (Austin et al., 2004b).

[3]SETI@Home (`http://seticlassic.ssl.berkeley.edu/`)

In contrast to Volunteer Computing the focus in Grid Computing was to build a generic middleware as a common platform for communication and as a basic infrastructure for further development of applications[4]. Foster (2002) emphasized the necessity of standard, open protocols and interfaces to provide functionalities like authentication, authorization, resource discovery, and resource access. Institutions such as OGF[5] and OASIS[6] expedited the standardization process of communication protocols and architectures of the Grid system (Foster et al., 2008). Furthermore, the quality of services should be non-trivial, i.e response, throughput and other user requirements have to be delivered reliably. The independence of the computing resource location and the focus on collaborative usage across different administrative domains implies a decentralized management of the available computational power and storage. Compared to Cluster Computing, Grids tend to be more loosely coupled, heterogeneous, and geographically dispersed (Neumann, 2007). Grid Computing has the ability to pool together resources, both hard- and software which are distributed among users (Alkadi and Alkadi, 2006; Knight, 2006). Due to the decentralized management, Foster and Kesselman (1998) defined the necessity of Virtual Organizations (VOs). VOs comprise multiple institutions working together and sharing computational resources. The goal of VOs is to enable federated management of resources in distributed environments (Foster et al., 2008).

With further development of Grid networks new challenges arise, for e.g. stemming from the heterogeneity of the available resources that do not meet the specific needs of applications or services. Dynamic changes in software requirements and resource availability can lead to under-utilization of resources, if users are not able to quickly adapt to the new situation (Foster et al., 2006). Concurrently, computation jobs running on the same machine can execute insecure code to elicit private information. Integrity cannot be guaranteed at the operating system level (Figueiredo et al., 2003). Therefore, the authors suggest an approach that is "based on a combination of 'classic' operating system level Virtual Machine (VM) and Grid middleware mechanisms to manage VMs in a distributed environment" to solve these issues. The concept of virtualization provides advantages in security mechanisms through isolation, by customization of virtual machines according to users' needs, in resource control at instantiation time, and site-independence[7] meaning that a VM can be stopped, migrated, and restarted again.

Adabala et al. (2005) describe the architecture of virtual Grids, which are "dynamic pools of virtual resources that can be aggregated on-demand for application-specific user-specific Grid Computing". This architecture is similar to the classical

[4]Currently, three major middleware are dominant in academia: Globus, gLite and Unicore
[5]Open Grid Forum
[6]Organization for the Advancement of Structured Information Standards
[7]assuming that the same VM tools are used

Grid architecture described by Foster (2002) but expands it as the virtual components lead to easier customization and a higher level of security. Only the virtual environments have to be managed across physical resources. The customized services built on top of the Grid can be allocated to the requesting consumers. Researchers have developed various concepts to manage the resource allocation problem ranging from technical heuristics (e.g. fair-share or user priorities) over cost functions to market models (Neumann, 2007).

However, Grids still lack a public utility, as access is limited to those who own the resources or those who are willing to share their resources. A sustainable business model is missing and no commercial Grid Computing provider emerged from these efforts (Foster et al., 2008). Most Grid Computing users are still from academia (e.g. the Large Hadron Collider project at CERN[8]).

As claimed by many researchers (e.g. Youseff et al. (2008)) the elements of Cloud Computing are not a technical innovation itself. Sharing computer resources or on demand services by SaaS has been there before the term Cloud Computing appeared. Weiss (2007) concludes that the real revolution of Clouds is the combination of those different IT aspects into a new business model. Virtualization of datacenter infrastructure helped to increase their utilization by offering storage and computer performance to third parties. On demand software offers possibilities to combine different software solutions into one environment(see e.g. SingTel's myBusiness platform). But combining these software services or development platforms with virtualized infrastructure really offers opportunities for new services and an increase in efficiency by economies of scale. Indeed, the idea of recombining different components of a technology to new products or services is not new. Already, Schumpeter (1935) explained the importance of combination in order to explain why innovation often appear in waves. Varian et al. (2004) set this idea in the context of IT and gives another important explanation of innovation in waves. He emphasized that the development of complements is crucial for the success of an innovation. An example is the innovation of the automobile, which would not have been possible without paved roads and the availability of gasoline. Paved roads were initially created for bicycles and gasoline for stationary engines. But without these complements, the car would have never been able to be so predominant. This idea well suits the Cloud Computing. As mentioned before, IT concepts such as Cluster and Grid Computing and Web 2.0 enabled applications built up the basis. Complements like virtual machines and a customer-centric view finally pushed the innovation of Cloud Computing.

[8]CERN (`http://lhc.web.cern.ch`)

2.1.2 Cloud Computing Definitions

Cloud providers offer their services similar to Grid Computing resources. However, contrary to the mainly scientific driven Grid scenarios, Cloud providers have to define Service Level Agreement (SLA) and apply business models (Weiss, 2007; Weinhardt et al., 2009). The virtualization technologies in Clouds enable the definition of the exact resource usage for one or more services of a Cloud provider. Moreover, Grid participants are contemporaneously consumer and provider, whereas Cloud providers and consumers can be clearly distinguished. If participants in the Grid do not use their resources, they can provide them to others. In case their own computation power does not suffice, they can tap into the Grid and get additional resources. The concept of Cloud Computing has a commercial background and the idea is to have dedicated providers offering different kinds of services over the Internet. In Cloud Computing resellers like Jollat[9] or RightScale[10] come into play by enhancing standard services from Amazon[11] with new services (Buyya et al., 2008; Armbrust et al., 2009).

The concept of Cloud Computing tries to pursue the goals that have been around since the early beginning of computing: giving users on-demand access to a public computing utility with high availability and scalability providing a cost advantage compared to traditional systems. Hitherto, McCarthy's vision of computing as a public utility like the telephone in 1961 has not become true, although the concept of Cloud Computing promises to reach that state in medium term (Ivanov, 2008; Foster et al., 2008). At present, there is no commonly accepted definition for the term "Cloud Computing". One reason is the fact that part of the concept of Cloud Computing is still in its developmental phase and at the same time proven computing methods, like virtualization, are part of the concept. For McKnight and Bailey (1997) "the term 'Cloud' implies that a user does not need to think much about what happens inside this system of networks", whereas they referred it to the Internet in general.

Boss et al. (2007) are among the first who tried to share a definition of Cloud Computing, one that includes both the platform and the type of application. The platform is characterized as "dynamically provisions, configures, reconfigures, and deprovisions servers as needed. Servers in the Cloud can be physical machines or virtual machines. Advanced Clouds typically include other computing resources such as storage area networks, network equipment, firewall and other security devices". Cloud Computing applications, however, "are extended to be accessible through the Internet. These Cloud applications use large data centers and pow-

[9] Graphical client for Amazon Web Services (`http://www.jollat.com/`)

[10] Cloud Management Platform based on Amazon Web Services (`http://www.rightscale.com/`)

[11] Amazon Web Services (`http://aws.amazon.com/`)

erful servers that host Web applications and Web services. Anyone with a suitable Internet connection and a standard browser can access a Cloud application".

Buyya et al. (2008) provide a definition with higher granularity defining the Cloud as "a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on SLA established through negotiation between the service provider and consumers". Subsequently, they outline that Clouds are not a combination of clusters and Grids but "next-generation data centers with nodes 'virtualized' through hypervisor technologies such as VMs, dynamically 'provisioned' on-demand as a personalized resource collection to meet a specific SLA, which is established through a 'negotiation' and accessible as a composable service via 'Web 2.0' technologies". However, currently most providers make a posted price offer. Consequently not every provider allows negotiation of prices.

Wang et al. (2008) outline enabling technologies that drive the development of Cloud Computing. These are virtualization concepts that have been proven in areas like Grid Computing and include virtual machines as well as virtual network advances. Other techniques are the automation of serviceflow and workflow orchestration using service-oriented architectures and Web services following industry standards. In addition, Web 2.0 and mashup technologies can be considered as drivers of communities and collaboration among users. Thereby Weiss (2007) concludes that the real innovation that comes with Cloud Computing is the integration of the existing technologies, explicitly "the combination of utility computing and datacenters". Lin et al. (2009) also emphasize that Cloud Computing is nothing new, but a confluence of technology and business development within the Internet leveraging new technologies.

Armbrust et al. (2009) agree on the view of Weiss (2007) and Lin et al. (2009) and add three additional aspects that emerged with the advent of Cloud Computing. These are "the on-demand availability of resources which lead to the illusion of infinite computing resources", "abolition for an up-front commitment to resources by users", and "the possibility to pay for what you use only, regardless of the time horizon". In their definition Cloud Computing "refers to both, the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services". Staten (2008) summarizes these attributes by defining Cloud Computing as "a pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end customer applications and billed by consumption".

Vaquero et al. (2009) offers a good overview of more than 20 definitions. As Cloud Computing involves a broad range of computing aspects, the definitions vary

as well. There are definitions focusing on the infrastructure layer and others trying to reduce the characteristics to a common denominator over all layers and include business model aspects of Cloud Computing. However, the Vaquero et al. (2009) limit the definition of Cloud Computing to the pay-as-you-go pricing model. Currently, there is no dominating pricing model in the Cloud Computing market, since usage-based pricing as well as flat fee are very common (**?**).

Foster et al. (2008) compare Grid Computing with Cloud Computing and conclude that both have the same vision such as reduce cost or increase flexibility. However, the commercial interest to create a large-scale system to analyze massive data makes Clouds more attractive to the industry. Moreover, different levels of services, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS), enable the delivery of customized services to consumers. These services can be dynamically configured and combined to more complex services[12] via virtualization. Foster et al. (2008) stress the role of virtualization as an 'indispensable ingredient' for Clouds, whereas Grids do not necessarily require virtualization, although efforts are being made to integrate it. The resource management concept of Grid and Cloud Computing differs according to the type of applications. In Grid Computing jobs are batch applications, since Grids do not natively support interactive applications. Clouds offer SLAs and thus must support a certain quality level for the consumers. Economies of scale and on-demand delivery are further advantages of Cloud Computing. However, security in Clouds is based on encrypted electronic communication and passwords are sent via email for a quick initiation process. In Grids the acceptance of a new participant incorporates an interaction with a trusted party via mail for identification in order to receive electronic credentials. The initiation process is more secure, but also time-consuming.

All of these definitions assume that the term Cloud Computing is more of a collective term for Internet-based services both from the technical as well as the economic perspective. Physical resources required for the offered services will be owned by one provider, although providers can buy services from a third-party and resell or enhance them into (more) complex services. Most providers will prefer to have homogeneous physical resources to facilitate management similar to Cluster Computing, which is mostly organized within an organization. However, to offer different kind of services, e.g. different operating systems, virtualization is an essential part of Clouds, whereas in Grids virtualization, although important, is still in its infancy (Foster et al., 2008). Virtualization enables the provider to offer heterogeneous services.

[12] Blau (2009) defines complex services as services which "typically involve the assembly and invocation of many pre-existing services possibly found in diverse enterprises to complete a multi-step business interaction"

From the organizational point of view, the function of a participant entering the Cloud can be distinguished into three roles. *Providers* offer services based on the above mentioned definition of Cloud. *Consumers* access the services offered in the Cloud over a Web 2.0 interface and use this information or infrastructure. *Integrators* buy these services from providers, aggregate or enhance them and sell the new service to the consumer. A typical example for the latter case is CastIron[13]. In Cluster and Grid Computing the participants act as a provider and consumer. All of them can send jobs into the network or the resources are used in idle cases. Volunteer Computing, however, only allows providers in a network and the central entity is the only consumer sending high amount batch jobs to all participants. While in Grids the number of both parties is almost equal[14], in Clouds the number of providers is rather small compared to the number of consumers. Hence, virtualization can help to offer scalable services for the large consumer group.

The applications running in a VM give the resource manager more flexibility and control to define appropriate SLAs. Although SLAs are not important in an intra-organizational context like in Cluster Computing, they play a major role in Clouds. Some providers such as Flexiscale even offer a 100% reliability of their infrastructure service. These SLAs enable consumers to work with interactive applications, since workflow systems can be configured reliably and time-sensitively. Usually, Grid and Volunteer Computing do not take SLAs into account due to the decentralized nature of control. However, there are approaches in the Grid domain analyzing the introduction and enforcement of SLAs (e.g. SLA4D-Grid[15] project or Balakrishnan et al. (2008) and Leff et al. (2003)), but none of them have provided a sustainable model yet or are successfully applied in larger Grid environments.

By offering reliable services, this characteristic entails a viable business model. Thus, Cloud services are often priced on a pay-as-you-go basis or include a flat fee model. On the contrary, Cluster and Grid Computing have a shared resource agreement, since the academia background of these concepts condemns the introduction of pricing models[16]. Volunteer Computing, as the name implies, assumes that the participants in the network are voluntarily sharing their idle resources among them-

[13] CastIron(`http://www.castiron.com/`)

[14] In Grids some participants only consume services, because they do not have any resources. Hence, not every participant acts as both a provider and a consumer in the Grid. In research projects like the Large Hadron Collider (LHC) institutes without their own resources can have access to the data of the LHC to be able to contribute to the field of research.

[15] Project SLA4D-Grid (`http://www.sla4d-grid.de/`)

[16] Researchers deny the usage of money for sharing resources for two main reasons. Firstly, researchers are not allowed to spend money without presenting valid reasons for their expenses to the sponsor. They always take care of the investment they make. An investment can lead to revenue. In some countries it is prohibited by law for institutes to make profit (e.g. Germany). Secondly, it is regarded unethical that those institutes with a deeper pocket are in a position to do better research than others. This is one reason why LHC gives institutes without their own resources access to data produced by the collider.

selves. Hence, no pricing model should be adopted. This also has an impact on the mode of allocating resources. In Volunteer Computing resources cannot be declared as dedicated for the application from the central entity, since the resources are not under centralized control. The allocation is based on voluntary sharing. The participants have full control over their resources and can offer them to or remove them from the network. In Cluster and Clouds consumers get dedicated or best-effort resource access. The central manager or Cloud provider offers low-priced best-effort services or the more expensive services with a guaranteed SLA. A similar idea is realized by the Amazon Spot Instances[17], where consumers can buy cheap resources without knowing when their instances are executed. Grids can offer different services as well, but they do not have centralized control. Hence, providers and consumers in Grids have to agree on the appropriate service (i.e. dedicated or shared).

Advance reservation is another aspect to distinguish between service offers. While some Cloud service providers offer advance reservation to get dedicated services (e.g. RenderRocket[18]), Volunteer Computing does not support this kind of interaction. In Grids it is possible and planned for the Globus Toolkit to implement such a feature, but is currently not available (Stoesser, 2009). Clusters allow dedicated allocation of resources and thus enable advance reservation for certain computing jobs. Still, some of the big players in the Cloud such as Amazon or Google do not offer advance reservation yet.

In Clusters and Grids the development of the applications is done locally. Clouds offer services running under centralized control and thus the applications and services are mostly developed on the server. The application in Volunteer Computing is fully developed on the server and the participants get a software, which is installed and executed locally. Services in the Cloud are accessed via standardized Web protocols like Representational State Transfer (REST) or Simple Object Access Protocol (SOAP). Grid Computing has a specific Grid middleware such as Globus Toolkit[19] or gLite[20]. Cluster and Volunteer Computing are built on open source or proprietary software such as Condor[21], Moab Cluster Suite[22] or BOINC[23].

However, the standard Web protocols in Clouds only give access to the services. The services themselves cannot be easily migrated to other providers, which can lead to high switching cost. It's easier for any Grid user to switch from one Grid provider's resources to another. Currently, Cloud providers have no big interest in applying standards, which ultimately makes it harder for potential customers

[17] Amazon Spot Instances (`http://aws.amazon.com/ec2/spot-instances/`)
[18] RenderRocket (`http://www.renderrocket.com/`)
[19] Globus Toolkit (`http://www.globus.org/`)
[20] gLite (`http://www.glite.org/`)
[21] Condor (`http://www.cs.wisc.edu/condor/`)
[22] Moab Cluster Suite (`http://www.clusterresources.com/`)
[23] BOINC (`http://boinc.berkeley.edu/`)

to switch among providers. For example, an Amazon EC2 instance cannot be executed at the Flexiscale platform. There are some attempts such as the OpenCloudManifesto[24] to introduce standards. And Eucalyptus[25] enables multiple provider Application Programming Interfaces (APIs). Cluster and Grid Computing have a standardized environment allowing them to send their jobs to almost any provider in the network. Participants in Volunteer Computing do not care about switching costs, since they are the resource provider for one central entity. Instead of providing another definition of Cloud Computing, this thesis outlines criteria to distinguish the different computing concepts in their current state (Table 2.1).

Depending on the accessibility to the services in the Cloud there is a distinction between *public and private* Clouds (Armbrust et al., 2009). Public Cloud providers allow the general public to access their services. Private Clouds are operated within businesses or other organizations and utilize internal datacenter hardware and software. While public Clouds need to have standardized interfaces to communicate with a large audience, private Clouds might even have proprietary protocols on purpose e.g. to avoid typical cloud connected problems such as security, privacy and lock-in effects. Some also predict the rise of *hybrid Clouds*, a composition of public and private infrastructure (Won, 2009; Mell and Grance, 2009). In this scenario critical data is processed in a private environment and public Clouds are used for non-critical data processing, which are provided by third parties. The National Institute of Standards and Technology (NIST)[26] augment this definition by introducing the term *community Cloud*. Several organizations share the community Cloud infrastructure, since they need these infrastructure for collaboration to achieve a (common) objective (Mell and Grance, 2009). While the definition of public, private and hybrid Clouds are reasonable and fit into the context of the various computing concepts, community Clouds is blurry. If resources are shared among communities, it is hard to distinguish community Cloud from other concepts like Grid or Cluster Computing. It is not clear who owns the resources or if switching costs play a role.

[24] OpenCloudManifesto (`http://www.opencloudmanifesto.org/`)
[25] Eucalyptus (`http://www.eucalyptus.com/`)
[26] An Agency of the U.S. Department of Commerce

Table 2.1: Comparison of different computing concepts

Criteria	Cluster Computing	Volunteer Computing	Grid Computing	Cloud Computing
Type of resources	homogeneous	heterogeneous	heterogeneous	homogeneous / heterogeneous
Virtualization	widespread	no	in its beginning	essential
Type of application	batch & interactive	batch	batch	batch & interactive
Development	locally or on a central server	central server	locally on a central server	in the Cloud
Access	via proprietary / open source software	via proprietary / open source software	via Grid middleware	via standard Web protocols
Role of participant	mostly provider and consumer	provider	mostly provider and consumer	either provider, consumer or integrator
Participation model	sharing	voluntary	sharing	pricing
SLAs / Liability	enforceable	not enforceable	not (yet) enforceable	essential
Control	centralized	decentralized	decentralized	centralized
Switching cost	low	none	low	high
Advance reservation	yes	no	possible	yes
Mode of allocation	dedicated / shared	shared	dedicated / shared	dedicated / shared

2.1.3 Cloud Services

In the so-called developed nations such as the United States of America, Japan and Germany, the percentage of labor in the service sector is above 50% (Maglio et al., 2006). Stahel (1997) states that a sustainable society is incompatible with the goals of the industrial economy, which will lead to the service economy with new and innovative services. According to Verma et al. (2002) the rapid development in IT has a great impact on the shift towards a service economy. Although services seem to be essential for the future growth of the world economy, the term *service* is often used ambiguously and not defined in a consistent way across different domains. Rathmell (1966) already outlined in 1966 that goods are perceived as tangible economic products, which can be seen and touched. Services, however,"seem to be every else; and an understanding of them is not clear" (Rathmell, 1966). In the 60s, marketing researchers were interested in the service sector and identifying its characteristics. A first distinction between goods and services can be made by the following classification for services (Judd, 1964):

- Rented-goods services: the right to possess and use a product,
- Owned-goods services: a value creation on an owned product like repair service,
- Non-goods services: no product is mainly involved (perhaps only supportive), but the experience or emotional change of a person is the result (e.g. entertainment).

Rathmell (1966) generalizes the difference by determining a good as a *thing* and a service as an *act or performance*. Although there might exist pure services or goods, most transactions between a buyer and a seller comprise of services as well as products, which underlines the mixed nature of the transaction object (Vargo and Lusch, 2004). Hill (1999) emphasizes that a service automatically involves a relationship between the service producer and the service consumer, since a service has to be provided to another economic unit. This process implies a change of condition of one or more persons, their property or good. A service cannot exist independently of its producer or consumer. Since a service is not an entity like a good, ownership rights are not applicable over a service. Contrary to the definition of Judd (1964), Hill (1999) gives a more abstract definition, which does not depend on the underlying good. Consequently, a service cannot be transferred from one economic unit to another. For example, a service cannot be stored in a warehouse and sent to other economic units in other countries like a good. Blau (2009) summarizes the key characteristics of services and, in particular, e-services.

Uno-Actu: The provision and consumption of a service cannot be separated (Hill, 1999). A barber can only cut the hair of a customer if both are at the same loca-

tion. According to Blau (2009), this property of a service is the main distinctive characteristic from a conventional good. It further implicates the following characteristics as well. The producer and consumer are continuously involved during the provision and consumption process.

Not storable: Services in general are perishable by nature. Volatile demands have a higher impact on services than on goods, since services are created and executed immediately. Services can neither be produced in advance nor be stored for future consumption.

Co-Creation: The interaction between consumer and provider enables the consumer to influence the outcome of a service. The service consumer can have different roles in the service execution process (Bitner et al., 1997). First, consumers can be passive, but their presence might be required. The execution should be isolated as much as possible from the consumers (e.g. the time of delivery of standard mails usually cannot be influenced by the consumer). Second, the participation of the consumer is necessary to fulfill the requested service, e.g. health care or entertainment. The third category describes the consumer as a competitor to the service provider. The consumer can either choose to create part of the services independently or to buy the service externally. A car owner can either choose to repair the car on his own, to hand it over to a car repair shop or to share the work (simple tasks himself and complex task by the repair shop).

Intangible value creation: Services are created without the transfer of ownership in this type of service. If the change of condition of an economic unit is accomplished, the consumer gains value from the change, which is often based on his expectations (Lovelock and Wirtz, 2001). This rather subjective evaluation of the delivered service can be challenging regarding the assessment of service quality by objective attributes.

Fuzzy inputs and outputs: The coincident production and consumption cannot be always created in a standard way. For example, a consultancy service may vary depending on the constitution of the service provider. Furthermore, a co-created service with a consumer will be influenced by the consumers behavior as well. Thus, the same service will not necessarily lead to the same outcome or the expected results (Gallouj and Weinstein, 1997).

The success of IT has entailed new business models and services in the last decade. The Internet enabled new opportunities to design, provide and consume services as well as to involve the consumer much better in the service creation process. As soon as a service is provisioned over an electronic network, it is known as an

e-service (Rust and Kannan, 2003). This definition includes the exchange of information over the Internet or other electronic networks in general such as IT services, Web Services or infrastructure services. For example, the operation of an ATM machine is an e-service as well. The authors emphasize that e-services is a customer-centric concept, which is in line with the co-creation characteristic of a service. Furthermore, a service is not storable by nature, since the service is created and executed while it is used. Electronic data can be stored, but the access to the data is a service, which is provisioned and consumed when the consumer downloads the data, is not storable. Hoffman (2003) arguments that an e-service is available 24/7 and if this service is not consumed one day it is available the next. The major difference, though, is that the value of the service is generated, when it is provisioned and consumed. The availability of the service does not directly increase the utility of the consumer. In particular in a market with dynamic prices, the utility often depends on the current price. If the price is too high, the consumer will not buy the service and hence no value is created. However, in some cases only the availability of services can indirectly increase the utility for a consumer. For example, a high number of providers will foster the competition on the service market and subsequently the prices often drop compared to monopoly.
The value created by downloading the data is only assessable by the consumers themselves. If the data service does not contain any valuable information, it is useless. However, as with all information services the value can only be evaluated by consumption. This can further result in fuzzy inputs and outputs. A vaguely defined electronic request by the consumer will perhaps not return the expected result. Though, most electronic services are defined more precisely than a service in general (Blau, 2009). Especially, *Web services* have a predefined description language and standard interface for communication (Papazoglou, 2008). An electronic service for gathering information about the weather can be defined by clear input parameter such as country, location name and zip code.
E-services also follow the uno-actu principle. For example, the definition and implementation of a Web service interface will be done once by the provider (production). In certain cases one or multiple requests can be served at the same time (provision and consumption). An example for the former case is an ATM machine, which can only serve one person at a time. A Web Service can be accessed by more than one person simultaneously. IT allows an automate process via algorithms in a way that no physical interaction of the provider is necessary. This allows the scalability of certain e-services.

Since the term *Web service* has been mentioned several times in the context of e-services, there exist ambiguous definitions of this term. Alonso et al. (2004) discuss these definitions and conclude that the World Wide Web consortium is quite accu-

rate[27]: "A Web service is a software system identified by a [Uniform Resource Identifier], whose public interfaces and bindings are defined and described using XML. As the name implies, a Web service leverages the Internet to communicate with other software systems. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols" (Austin et al., 2004a). These definitions apparently stick Web Services close to XML. The authors emphasize that the standardization aspect (XML-definition, -description and -discovery) of a Web service is as important as the loosely coupled development of these services underlining the service-oriented paradigm. Papazoglou (2008) augments this definition by a business-oriented characteristic. He says that a Web service can be a simple passive information service or a complex business application combining information from multiple sources.

This thesis focuses on *Cloud services.* Cloud services can be subsumed under e-services, since e-service is a much broader term. However, some characteristics might only apply to Cloud services compared to other e-services like ATM machines. As already mentioned above, e-services provided over the Internet, such as Cloud services, have the advantage to serve multiple requests simultaneously. Furthermore, the condition of a physical economic unit is changed or can be changed remotely (writing data on a server). In fact, the service gives access to remote data or computing resources for consumption. A transportation of a physical good or the provision of the consumers themselves (like in a barber shop) is not applicable. But physical resources are still necessary to create, provide and consume Cloud services[28]. The consumers gain access to Cloud services with a browser or a well-defined protocol to access the service via a desktop application. A browser interface is mandatory to give consumers access from various devices such as mobile phones, since the vision is to access Cloud services from anytime and anywhere.
According to Papazoglou (2008), Web services operate at the code level and are invoked by other software systems. Software applications incorporate Web services to enable human interaction. However, a Cloud service uses Web service technology for communication and comprises abstracted services from the underlying computational resources. In particular, a Cloud service provides a software system for a consumer over a Web browser to offer 'anytime, anywhere' access. While Web services are not targeted for human interaction, the main purpose of Cloud services is to deliver the user an enhanced communication interface (via browser and optionally software application) to hardware resources (e.g. storage) or software environments such as the Windows Azure Platform supporting .Net developers or

[27]Note that the definition from 2004 has slightly changed to the definition Alonso et al. (2004) has reviewed from 2002

[28]This is an important aspect. In this thesis it is assumed that every service can be mapped to the consumed physical resources.

Salesforce's CRM Cloud service. Thus, the web access via browser is a necessary part of a Cloud service. The provider can additionally offer APIs or proprietary software to access these services as well. Furthermore, Cloud services can be combined to complex services. Complex services provide added value for the consumer by aggregating or combining functionalities of other (basic or complex) services (Papazoglou, 2008). Mell and Grance (2009) identify multi-tenancy as one of the main characteristics in the software architecture of a Cloud service to distinguish it form other services. However, in general, it is hard to distinguish Cloud services from other electronic services and there is no explicit definition of this term. Since there is no agreement on the unique attributes of Cloud services and the term is often used in a broader sense, a definition here is disregarded.

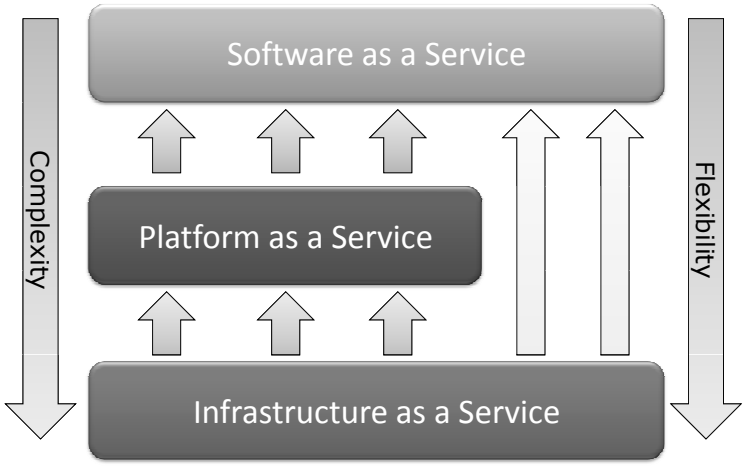


Figure 2.1: Three service layer architecture in the Cloud.

Youseff et al. (2008), Motahari-Nezhad et al. (2009) and other researchers basically differentiate between three levels of services, on which Cloud services can be provided (see Figure 2.1): Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). Others like Armbrust et al. (2009) do not believe in such a "XaaS" separation as to their understanding the different layers are not clearly defined: "but we were unable to agree even among ourselves what the precise differences among them might be". They comment that SaaS has been there already before the rise of Cloud Computing and only on an infrastructure layer a Cloud revolution is happening, so that they only call the infrastructure providers Cloud providers. Hence, this thesis will mainly focus on the IaaS layer, although the differences between the three layers are outlined below to enable a concrete definition for the remainder of this thesis.

Infrastructure as a Service:
IaaS is the virtualized provision of (external) datacenter services. The IaaS layer consists of storage and computational services (Vaquero et al., 2009). Consumers

can access providers' datacenters over the Internet and can dynamically adjust their requested computing resources. In order to separate different consumer requests within the same datacenter, virtual machines play a crucial role. They allow the setting up of different computing environments for each consumer, so that the virtualized instance appears to the consumer like a personal datacenter. Still, the infrastructure layer remains the most basic level of Cloud Computing, requesting consumers to have profound skills of datacenter technology or of the virtual environment. Armbrust et al. (2009) point out that especially the infrastructure layer builds the new aspects of Cloud Computing with the illusion of infinite computing resources available on-demand, the elimination of an up-front commitment by Cloud users and the ability to pay for use of computing resources on a short-term basis and release them as needed. Especially, the infrastructure as a service paradigm offers the possibility to increase datacenters' utilities. Currently, datacenters suffer from very low utilization rates (10-35%) as they are built to confront peak load times (Carr, 2005). With the possibility to share and dynamically adjust datacenter resources, the infrastructure layer offers great possibilities to cut down datacenter running costs through outsourcing.

Platform as a Service:
PaaS is situated at a higher abstraction level than IaaS, as Briscoe and Marinos (2009) and Youseff et al. (2008) agree consistently. The offered platform services provide an application environment instead of plain computing resources. This environment can help developers to accelerate the deployment of their applications in a Cloud environment and makes concerns about scalability inconsiderable. Developers can gain several advantages from platform services. They can profit from included "automatic scaling and load balancing, as well as integration with other services (e.g. authentication services, email services, user interface)" (Youseff et al., 2008). Briscoe and Marinos (2009) mention similar advantages but name as well that the developers have to deal with constraints as they are limited to the platform providers programming language. The most famous ambassador of the platform providers is Google's App Engine, which offers a python as well as a Java programming language environment. Though, other vendors like SingTel, a big Southeast Asian telecommunications company, are now entering the market. They have recently launched their SingTel Innovation Exchange platform[29] in July 2009. Within this platform service, developers cannot only benefit from the previously listed advantages, but also have direct access to SingTel's 250 million customer base which offers additional value to developers to deploy their applications.

[29]SingTel Innovation Exchange (`http://business.singtel.com/innovation/index.asp`)

Software as a Service:
SaaS is the most visible layer to end-users, because the SaaS paradigms comprehends on-demand online applications for a wide variety of users. SaaS providers can bundle software applications and therefore offer additional service within one packet and deliver it to their end customer across the Internet (Ma and Seidmann, 2008). Vaquero et al. (2009) describe those services as an alternative to locally run applications. As Youseff et al. (2008) mention, SaaS offers the advantage that end-users are no longer burdened by "software maintenance and the ongoing operation and support costs". Furthermore, they can profit from less restrictions on their hardware equipment and run even Central Processing Unit (CPU)- and memory-intensive applications without owning the expensive hardware. Users can "access the service 'anytime, anywhere', share data and collaborate more easily" than in previous solutions (Armbrust et al., 2009). The SaaS providers on the other hand can benefit from simplified software development. The central control of the application makes versioning and upgrading of applications much easier than individual requests to each end-user. The standard of a Cloud environment offers in addition opportunities to combine several different services in order to create a new value for end-users. SaaS was available prior to the hype of Cloud Computing, however, the advantages of SaaS remain in Cloud Computing. With the possibility to host software services in combination with infrastructure services, software providers do not even have to worry about the datacenter any longer. As Armbrust et al. (2009) sum it up: "Analogously to how SaaS allows the user to offload some problems to the SaaS provider, the SaaS provider can now offload some of his problems to the Cloud Computing [infrastructure] provider".

Foster and Tuecke (2005) outlined that Utility, On-Demand and Grid Computing have a big overlap in fostering the idea of IT in general as a service. The underlying idea is to provide physical resources and other services (such as platforms or software) over the Internet via some standardized protocols. Cloud Computing seems to be a promising paradigm towards this vision of computing as a utility like electricity. It leverages technologies like virtualization, Web 2.0 software tools and paradigms to provide access to physical resources, development platforms and business software environments through a Web browser, which was not done before in such a way.

2.2 Traditional Revenue Management

The term Revenue Management is most commonly used for the theory and practice of maximizing expected revenues by opening and closing different fare classes or dynamically adjusting prices for services. The development of the scientific research

in this discipline started after the deregulation of the American airline industry in 1978, which relaxed restrictions over standardized prices and profitability targets (Belobaba, 1987). Secomandi et al. (2002) define Revenue Management as the marriage of "OR/MS (operations research/management science), statistics, economics, software development, and consulting to manage demand for a firm's resource inventory with the goal of maximizing revenue (or profit)".

Netessine and Shumsky (2002) define Yield Management as a part of Revenue Management although boundaries between both are often ambiguous. However, most authors perceive the term Yield Management as the predecessor for what is nowadays called Revenue Management. To develop industry-independent models for revenue optimization, Weatherford and Bodily (1992) introduced the term Perishable Asset Revenue Management. This term relates to one of the main characteristics of Revenue Management: The perishability of the offered services. For example, a Cloud Computing provider cannot save up resources at a certain point in time in order to sell them later. Instead, an unsold service becomes worthless without creating any revenue (Netessine and Shumsky, 2002). The resources are time-dependent and have to be consumed at a given time. Otherwise, the resources for this certain timeslot are not available any more. Throughout this thesis, the term Revenue Management will be used synonymously with Yield Management and Perishable Asset Revenue Management.

Revenue Management techniques assume that Cloud service consumers can be classified into different segments. The key advantage of Revenue Management systems is the possibility to extract consumers' willingness to pay for the identified market segments by offering services with different levels of restrictions and charging those at different prices. The preferences are distinguished by different needs at different points in time and by varied willingness to pay. Shen and Su (2007) model the consumer heterogeneity along two dimensions: their willingness to wait for a better offer and their willingness to pay (i.e. their reservation price). Impatient consumers, who need a service soon, will be willing to pay more or they will increase their reservation price over time. The unpredictable behavior of consumers characterizes the high complexity of the appropriate application of Revenue Management tools. Consequently, from the service provider point of view the main topic in Revenue Management theory is to find the right combination of consumers buying different kinds of services to provide the highest possible revenue. Therefore, a choice has to be made between offering a service for sale or protecting it and waiting for a more profitable consumer. If vendors decide to protect a service for future demand, they take the risk of ending up without selling the service (Goldman et al., 2002; Netessine and Shumsky, 2002).

When applying Revenue Management models, there are numerous types especially for the airline industry that show an increase in the firm's revenue, some considering single leg models (e.g. Belobaba (1987) or Gosavi et al. (2002)) and others considering network models (Gallego and van Ryzin, 1997; Bertsimas and Popescu, 2003). *Single leg models* represent only one resource in the optimization model. Thus, more complex dependencies like several airline routes or multiple resources consumed by various services are not taken into account. The latter case is modeled in the *Network Capacity Control (NCC)*. All these models are based on several assumptions especially about the behavior of the consumer, e.g. a consumer will not switch between the service offers or react to price changes. *Customer Choice models* try to identify and formalize consumers' behavior to improve the Revenue Management models according to their booking limits and prices. Further aspects of Revenue Management like price-based control, overbooking, forecasting, passenger diversion (buy up) or degradation costs are disregarded. Interested readers for these topics are referred to the papers of e.g. Elmaghraby and Keskinocak (2003), Gosavi et al. (2002), Subramanian et al. (1999), Botimer and Belobaba (1999) or Zhao and Zheng (2001). Figure 2.2 outlines the dependency between the different areas. This thesis will focus on capacity control and customer segmentation or customer choice models, respectively.

2.2.1 Capacity Control for Single Leg

The problem of optimally allocating resources to several classes of demand is simplified by restricting the number of used resources to one. This single resource might be, for example, seats in an airplane or, in the context of IT, an Internet service provider who offers Internet access. In the latter case only the usage of bandwidth as resource is considered. Littlewood (1972) suggested that in a simple two fare model, discount bookings should be accepted as long as the expected future revenue for the remaining full fare services is lower than discount sales.

Let n be the capacity of maximum customers or connections for an Internet service provider, where customers can be distinguished by discount and full fare classes. The protection level[30] for full fare services is denoted by q_i and each unit is sold for a certain price. The provider can decide whether to accept or reject a request for the discount fare class. In case of acceptance the provider will gain a revenue of r_2 for the discount fare class (r_1 for the full fare class). The demand for the full fare class depends on the expected demand. To accept a discount request the expected marginal value for the full fare class must be lower than the price for the discount fare class. Formally, $r_2 \geq r_1 \cdot P(D_1 \geq q_1)$, where D_1 is the expected

[30]The amount of seats which are explicitly reserved for a certain fare class.

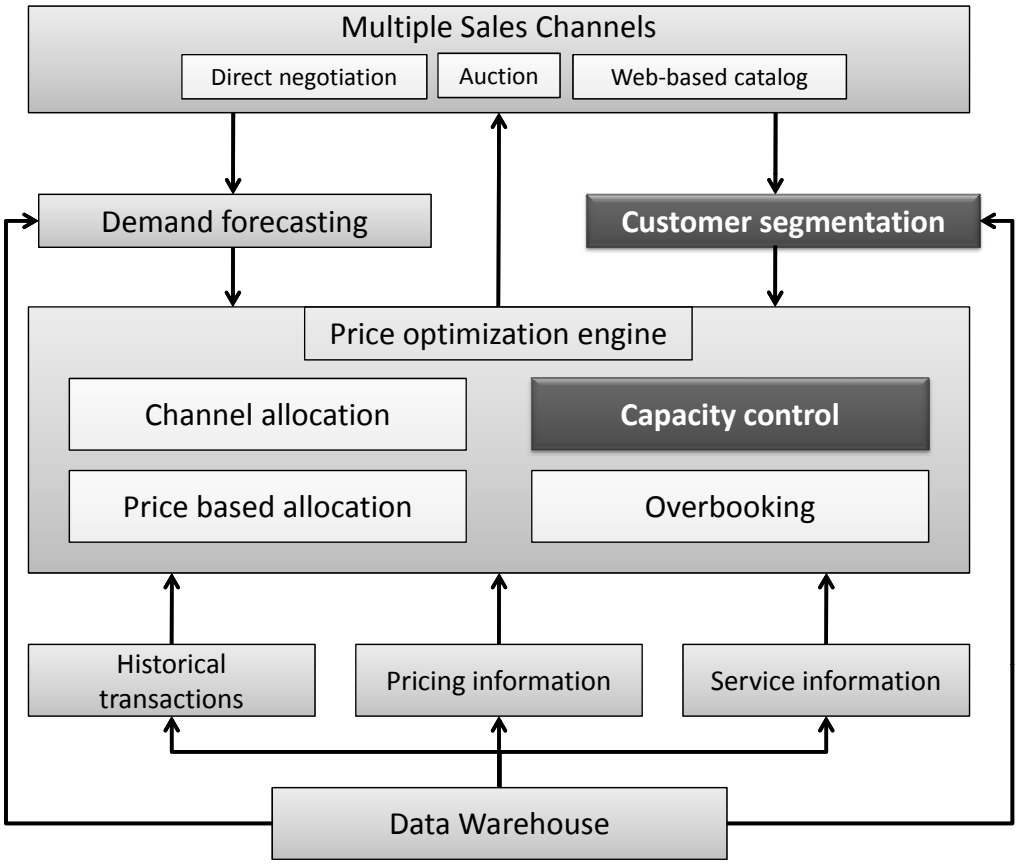


Figure 2.2: Revenue Management process flow in the e-Business context (adapted from Talluri and van Ryzin (2004b) and Bichler et al. (2002)).

demand for the full fare class. Consequently, the optimal protection level satisfies $r_2 < r_1 \cdot P(D_1 \geq q_1)$ and $r_2 \geq r_1 \cdot P(D_1 \geq q_1 + 1)$ (Netessine and Shumsky, 2002; Talluri and van Ryzin, 2004b). There are several extensions of this simple two-class to a n-class model. One of these general approaches was introduced by Belobaba (1987) and is called Expected Marginal Seat Revenue (EMSR). Although the model does not create optimal revenue in the general case especially when revenues of different classes are close together, it is easy to implement and therefore widely used in airline Revenue Management (Gosavi et al., 2002).

There are three common types of availability control that are distinguished. The first one is to strictly split the given capacity into fare classes with independent booking limits. But with uncertain demand a lot of revenue might be lost due to rejected full fare booking requests although capacity from other classes are still available. To overcome this problem, virtual nesting defines the booking limits q_i for strictly monotonically decreasing revenue in fare classes i as $q_i = n - q_{i+1}$ with capacity n. In comparison to the first type, it simply enables that higher-valued fare classes i can be extended by decreasing the capacity of lower valued by one, i.e. $i + 1$ (Smith and Penn, 1988; Talluri and van Ryzin, 2004b). The third and very popular method is to set a bid price that represents the lower bound for the price of a booking request.

If the price of the service is below the bid price the request will be rejected. Otherwise, it will be accepted. The bid price has to be updated after every sale to result in near optimal solution and generally follows a monotonically non-decreasing function for a decreasing remaining capacity. Accordingly, low valued classes are closed first and higher valued ones are reserved for short term bookings (Smith and Penn, 1988; Williamson, 1992).

2.2.2 Network Model of Capacity Control

The single leg model as described in the previous section only takes one resource[31] into account. However, complex services are built on resources or basic services with different capacity limitations. The demand for one complex service can change the requested units for the basic services or resources simultaneously. This interdependency has to be jointly coordinated to make precise decisions as to accept or reject consumer requests.

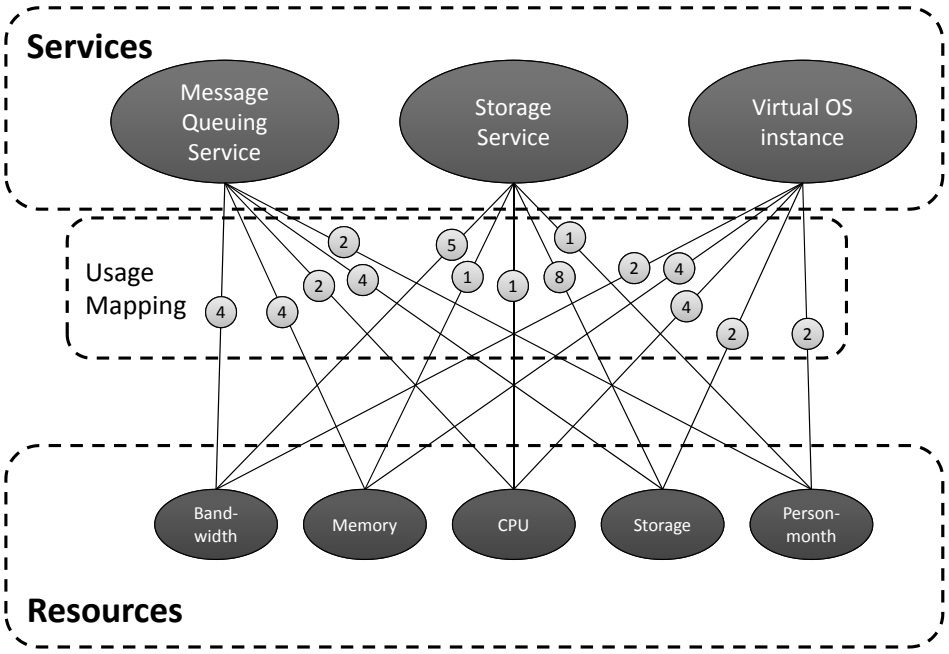


Figure 2.3: An example for the dependencies between services and resources on the IaaS layer.

Compared to the single leg problem the Network Capacity Control (NCC) problems are better suited for real-life situations by taking the resource dependencies between the offered services into account. Services use different kind of resources, e.g. the storage service in Figure 2.3 consumes five units of bandwidth. The amount of each resource consumed by the services define the usage mapping. In general, the change on the number of users of one service affects the resource usage and the

[31] e.g. bandwidth of an Internet connection or one origin-destination route for airlines

availability of the resources for other services. The application of network models implies a more accurate estimate of the expected revenue. However, NCC problems are more complicated than single leg controls, require more effort concerning implementation, and are methodologically challenging. They arise in industries where customers demand bundles of resources. Glover et al. (1982) were one of the first to theoretically describe the problem of optimizing profit in the scheduled airline industry taking into account different fare classes, demand and flight-segment capacities. By considering different services at the same time the problem becomes far more complex. But there is a high probability for significant revenue enhancement. For example, a hotel might consider every room and night as a single service. Because in general a one night booking creates less profit then a multiple night stay, the use of a multiple resources approach would be likely to raise profits. Availability control can be applied from the single resource problem. In particular, bid-price control is easy to adapt by setting bid prices for every resource and accumulate them for the requested service.

The major task of network capacity controls is to support optimal accept/reject decisions on dynamically arriving requests for complex services. Services differ in their prices, and can also differ in their resource demands depending on the application context. A successful establishment of such a system can yield high potential profit. Compared to the single-resource case, optimization is much more complex and exact optimization in many cases is not feasible (Williamson, 1992). NCC is very intuitive, but Talluri and van Ryzin (1998) also show that it is not generally optimal. However, bid-price control achieves good results for most situations and is gaining popularity against virtual nesting controls (Talluri and van Ryzin, 2004b). Furthermore, virtual nesting control uses a process called indexing, which clusters each complex service to a virtual class. This process inherently introduces noise into the existing data and affects the forecast as well. Therefore, this thesis will focus on the bid-price approach of NCC and this approach will be discussed in detail in Chapter 4.

2.2.3 Customer Choice Model

Currently, Cloud services and offers are based on a trial and error principle. Amazon, as one of the first, successfully started offering virtual machines for a relatively low price. Microsoft and others followed with different kind of services. From a theoretical point of view, the profitability of such offers are not well-known. It seems that the prices and service design are chosen rather arbitrarily. Customer choice and market response models are part of Revenue Management and offer researchers the ability to analyze and incorporate consumers' behavior in a structured manner. General Revenue Management models for quantity- and price-based control usu-

ally simplify the scenario regarding the reaction of consumers to price changes or rejected requests. This approach allows certain properties of the model to be proven analytically, but neglects the complex interaction with the consumer of a service.

However, the preferences of consumers are of great importance. Consumers may decide to wait until a service becomes cheaper or expect a certain quality level of the offered service. They may switch or upgrade to another service if the requested service is not available at that moment (Talluri and van Ryzin, 2004b). Their willingness to pay may be lower than a Cloud service provider assumes. Or the consumers are ready to pay more for the service than the offered price (Belobaba, 1987; Anderson et al., 1992). Hence, the understanding of consumers' preferences is directly linked to the decision making process consumers go through. Uncertainty about the consumers has a great impact on providers' strategy. If services can be reserved in advance, Ng (2008) identifies that consumers' behavior can be stochastic, probabilistic and deterministic. The time of arrival is not known to the provider and is thus a stochastic process. The process of a consumer making the choice whether to buy, not to buy or to wait for buying a service later can be described probabilistically. The behavior of a consumer can be influenced by pricing policies and service design, which is the deterministic component in the model from Ng (2008). The goal is to determine the most significant attributes that identify the best choice of a group of alternatives and how the offered services can be designed. Especially in Cloud Computing, providers have a high degree of flexibility to design various services individually for a certain group of consumers.

Ben-Akiva and Lerman (1985) present a framework for choice theories that consists of five steps: in step one the choice problem has to be defined. After defining the problem the different possible alternatives have to be generated. In the third step the attributes of the alternatives are evaluated. Subsequently, a choice is made according to the decision rules that apply. Step five is the implementation of the choice. It is important to note that individuals can diverge from these procedures and follow their habits, their intuition or imitate a trendsetter.

Four elements are part of the procedures that form the specific theory of choice. These are *decision makers*, *alternatives*, *attributes of alternatives*, and *decision rules*. The decision maker is any individual or group who faces different choice situations. Each decision maker behaves differently and has a different willingness to pay as a result of individual preferences. Considering the individual's unique preference a set of choices is derived from the universal set of choices which contains all possible combinations of attributes. The individual set includes the alternatives that are known and feasible to the decision maker. Feasibility is defined by constraints like physical and time availability, financial issues, informational constraints. The decision rule describes the mechanism used by the decision maker to identify the

preferred choice. There are four categories of decision rules which are described below:

- **Dominance:** The preferred alternative has at least one attribute that is better than all other alternatives and is not worse for all others. It is very unlikely that in real world scenarios one alternative is the cheapest, performing best and most comfortable. However, this procedure can be used to sort out redundant alternatives. Furthermore, an indifference threshold value can be defined, when small changes in the attribute levels are not considered as significant for the overall utility (e.g. the decision maker is indifferent between the availability of 25MB/s or 35MB/s of bandwidth.)

- **Satisfaction:** For each attribute there exists a level of satisfaction that has to be reached to take the associated alternative into consideration. Alternatives that do not fulfill the desired level of aspiration are not taken into consideration. Although this does not necessarily lead to a distinctive choice, the combination with other rules can have a big impact on the final decision.

- **Lexicographic rules:** If attributes are ranked by importance, the decision maker can choose the alternative that is most attractive to him. If more than one alternative is left by deciding on the most important attribute, the decision maker goes on with the second most important attribute and continues until a final choice is made.

- **Utility:** Utility is defined as the single objective function which expresses the attraction of an alternative. In economics, ordinal and cardinal utilities are distinguished. Ordinal utility captures only ranking while cardinal utility also captures the strength of preferences numerically.

The Cloud service consumer as the decision maker can also combine these rules. A common way is to combine the lexicographic rules with the satisfaction, which is known as the elimination by aspects (Tversky, 1972). Given this orientation on the consumers' decision making process, the question is: How can the consumers' preferences be analyzed in order to offer the revenue maximizing set of alternatives? There exist two ways to analyze consumers' preferences: *composition* and *decomposition* (Hahn, 1997). The compositional approach lets consumers value the attributes of each alternative separately and indirectly. The respective part-worth utilities are then composed into a total utility. In contrast, an overall judgment is split into its part-worth utilities in the decompositional approach. Therefore, either multivariate methods or linear optimization is used. In the empirical study described in detail in Chapter 3 the decompositional approach is performed, as this approach leads to

a better estimation of consumers' preferences compared to the compositional approach (Akaah and Korgaonkar, 1983). The information about individual utilities has to be aggregated to allow vendors to segment the market and structure the offers with respect to each segments' preferences.

2.3 Research Approach

The previous two sections highlighted the key characteristics of Cloud Computing and Revenue Management. This section describes the intersection of the research domain Cloud Computing and the research methodologies used in this thesis in detail. The similarities and differences between the traditional Revenue Management and the peculiarities of Cloud Computing will be outlined. Subsequently, the methods applied to analyze the application and the effectiveness of Revenue Management to Cloud Computing will be discussed and research questions will be derived.

2.3.1 Towards Revenue Managed Clouds

The first paper analyzing Revenue Management concepts for on-demand IT scenarios was published by Dube et al. (2005). In the suggested model one resource is offered at different prices. By assuming that the customer behavior follows a logit model, the authors analyzed an optimization model for a small number of price classes and provided numerical results. Although the authors state that "in an on-demand operating environment, customers and jobs, or service requests arrive at random", the behavior of price sensitive customers can be influenced by offers different prices for the same product, which in turn reduces the randomness (Bitran and Caldentey, 2003; Wilson, 1995).
Nonetheless, a certain degree of uncertainty still exists due to the unpredictable customer behavior and unpredictable events. Customers may cancel their resource reservations or may not show up. Overbooking strategies developed for the airline sector, for instance Coughlan (1999), can be used to accept more reservations than the actual available capacity, which can effectively minimize losses of revenue caused by customer no-shows. The benefits of overbooking for shared hosting platforms was emphasized by Urgaonkar et al. (2002) as well. They did not optimize the revenue by classifying different services, but only the throughput rate. Cancelations and no-shows reduce the efficiency of resource usage. Sulistio et al. (2008) analyzed how overbooking strategies can be applied to maximize revenue. Different prices were charged for one resource and three overbooking policies were implemented and compared via simulation.
Nair and Bapna (2001) introduced Revenue Management concepts for a similar ap-

plication domain, namely for an Internet Service Provider. The provider has to decide whether to accept an incoming customer request or to reject it. The application domain is different from Cloud Computing as it does not take advance reservation and resource-service dependency into account. Customers can get an internet access only instantly.

Revenue Management deals with complex decision problems. Therefore, the question arises under which conditions the application of Revenue Management methods is beneficial for a company. First of all, Revenue Management is a framework to optimize resource allocation and pricing from a providers' perspective. According to Talluri and van Ryzin (2004b), there are numerous conditions, which add complexity to a company's decision making processes, and therefore motivate the use of Revenue Management techniques to support this process. The requirements of Revenue Management can be distinguished into three categories: *service properties*, *consumer behavior* and *policy & design* (Figure 2.4). The services offered by the provider should be perishable and inflexible in production. The addressed consumers should have a heterogenic demand characteristic and their consumption should be unpredictable to a certain extent. Furthermore, the business model from the provider has to take overbooking for a better resource utilization into account and the services have to be able to be broken down to the consumed basic services or resources. A service-resource mapping should be feasible and the services should compete for the scarce resources. The conditions for applying Revenue Management to Cloud Computing are discussed below.

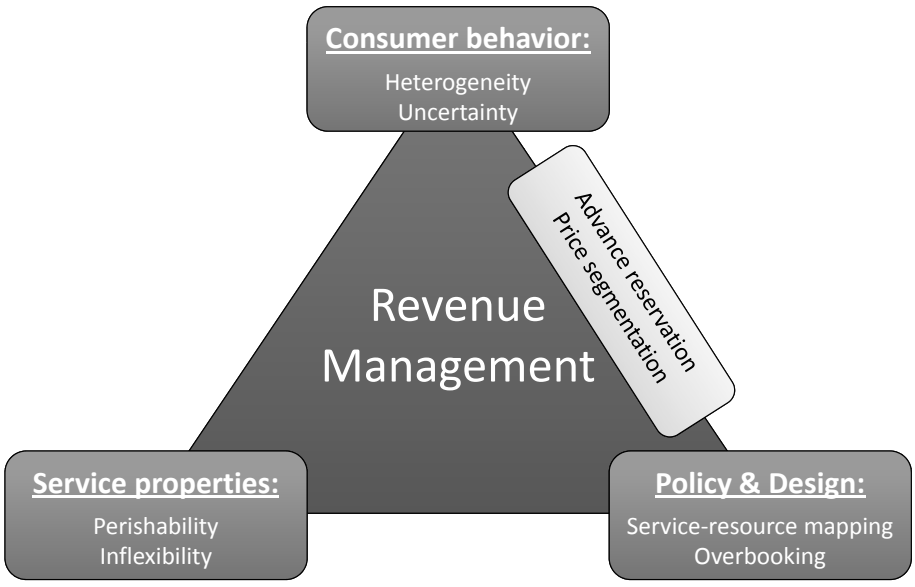


Figure 2.4: Requirements for applying Revenue Management.

Perishability:
As discussed in Section 2.1.3 for Cloud services, the perishability of the offered services is one of the main characteristics that have to be fulfilled for an appropriate use of Revenue Management (Netessine and Shumsky, 2002). For the traditional industries, this means that every unused seat in an airplane or every vacant room in a hotel signifies that potential revenue could not be realized. This property can be directly transferred to a Cloud Computing environment: As long as enough resources are left to provide at least one service[32], revenue is lost. Cloud services are perishable with time, i.e. resources can be utilized at a given time or they perish.

Inflexibility of production:
The production of some services is inflexible, that is, variations in demand cannot be counterbalanced easily by simply adjusting the supply. This implies delays, higher fixed costs, more capacity constraints, or even higher switching costs. These factors complicate the possibilities of correctly reacting on fluctuations in demand as well as decisions concerning supply. These factors make the application of Revenue Management methodologies useful. The airline sector is extremely inflexible in production. A flight from an origin to a destination is limited by the number of seats on the flight, and the total cost of the flight is fixed, independent of how many passengers are on the plane. Hence, it is beneficial for an airline to sell as many seats on a flight as possible in order to reduce the unit cost per seat. Therefore, the correlating service offerings are considered as limited and without the possibility of an extension in a specific time horizon (Weatherford and Bodily, 1992).
In the case of Cloud services, the production of additional capacity is much more flexible. Paleologo (2004) states that it is possible to increase the capacity of physical resources by integrating additional hardware into a present system. However, this highly depends on the hardware, the infrastructure, the architecture, and the required support of applications. Therefore, production and supply of Cloud services can be less inflexible, but this is not the general case.
Moreover, physical limitations still apply from a provider's perspective, which lead to higher inflexibility (Bailey, 2008). A survey by IDC Research, Inc. revealed that limiting factors can be either the space of data centers or power and cooling issues. Although the user currently gets the impression of unlimited resource usage by one provider, the physical constraint can be an obstacle. The deployment of new servers to extend an existing server farm can last up to 24 days on average (Lin et al., 2009). Another example is that providers such as Terremark[33] have limited the number of available VMs to 60 per customer. This restriction allows Terremark to better

[32]Note that every service requires one or more physical resources (e.g. CPU, memory, storage etc.) for creation, provision and consumption
[33]Terremark (`http://www.terremark.com/`)

forecast utilization and allocate capacity. Consequently, the available resources can be considered as fixed for a certain time horizon to effectively implement Revenue Management techniques.

Same resources for multiple services (service-resource mapping):
For Cloud Computing there is the desired advantage of flexible services. Compared to an airplane, where every seat may be used to satisfy any element of a certain set of final services, the resources of a Cloud Computing center are also able to provide different service offerings. Therefore, several resource bundles may be defined as Cloud services, e.g. containing different amounts of CPU power and storage. It can be seen that the flexibility of the initial capacity is given and also copious quantities of possible resource combinations are enabled. However, leaving the possibility open for every feasible combination, the problem of computing an optimal pricing strategy becomes more expensive (Bitran and Caldentey, 2003). From a technical perspective, virtualization technologies foster the implementation of multiple services, e.g. multiple operating system environments, on a single physical machine. The definition of resources is not limited to physical resources. Resources in the service-resource mapping can also represent basic services. Complex services invoke these pre-existing basic services to support a business process. Hence, a mapping can be defined between the complex services and the basic services. If the basic services are created and provided by a third-party, they can be limited in their capacity by determining different prices for different amounts. These prices will affect the revenue of a complex service provider.

Overbooking:
Due to cancelations and not arriving customers, so called no-shows, airlines began to overbook flights. The problem is to find the optimal relation between real seats and additionally sold seats that do not exist. This procedure stimulated to put more focus on forecasting of customer behavior (McGill and van Ryzin, 1999). Every service provider using overbooking has to define a service level that a certain amount of requests cannot be satisfied (Rothstein, 1971). The Cloud Computing overbooking procedure would be to sell more of the computing and storage capacity than the computing center has. In this case, not every costumer will exploit its reserved resources completely. Overbooking of Cloud Computing resources allows more flexibility. Urgaonkar et al. (2002) show that the usage of overbooking techniques can increase utilization drastically: Already an overbooking rate of just 1% may increase the utilization of the entire data center by a factor of two without losing meaningful availability guarantees. However, these guarantees are one way to provide a suggestive limitation for the overbooking practice. Especially for Cloud services, where overbooking can theoretically be used limitlessly by scaling down the resources for

every instance, service guarantees have to be established to ensure a certain service quality for the user. Therefore, a SLA is introduced which defines a minimum availability for the offered service, measured in percentage. Finally, for not fulfilling a SLA, economic penalties for the provider have to be defined. Amazon as one of the first virtual instance providers offered a 99.95% availability of their operating system. In April 2009 3Tera was one of the first providers committing to a 99.999% availability (only 6 minutes of downtime per year). FlexiScale has outnumbered this offer by a 100% availability guarantee.

Advance reservation:
The ability to reserve future services, like booking a room for a special day in the future and ordering a rental car on the same day, is naturally integrated in every industry that traditionally uses Revenue Management. In contrast, this naturalness is not directly rejected in Cloud Computing services. Currently, there are only a few Cloud Computing providers offering this possibility (e.g. RenderRocket[34] or the Cloud toolkit OpenNebula[35] in combination with Haizea[36]). This useful feature has some relevant advantages that are outlined by Boss et al. (2007): On the one hand, it gives the user the possibility to ensure a prospective computation demand by reserving the required services for the desired time. On the other hand, it provides the seller with the ability to easily discriminate customers not only by their valuation but also by varying terms in the SLA. Otherwise, there would be exclusively immediate bookings. Furthermore, it seems to be a good supporting feature to smoothen the demand for services, and thus the resource utilization. Hence, it has to become relevant in the near future.
An example comes from academia, where advance reservation led to better planning and scheduling of computing instances. The North Carolina State University in Raleigh, USA, piloted a virtual computing lab[37] system in fall 2004. It delivered computing lab applications and virtual machines with Windows and Linux to remote users via a Cloud Computing infrastructure. This infrastructure could be exerted for lectures to show students different applications. Especially, course instructors at the university had the ability to reserve computing instances for courses at a certain time. The pre-configured instances could be loaded for the requested time period, e.g. for two hours from 2pm to 4pm, and the number of instances could be reserved (obviously depending on the number of students on the course). Starting with 700 reservations per semester at the beginning in 2004, the number of reservations rapidly increased to 80,000. In 2007 the number of reservations was at its peak with 600 concurrent bookings (Schaffer et al., 2009). This scenario is rele-

[34]RenderRocket (`http://www.renderrocket.com/`)
[35]OpenNebula (`http://www.opennebula.org/`)
[36]Haizea (`http://haizea.cs.uchicago.edu/`)
[37]Virtual computing lab of the University in Raleigh (`http://vcl.ncsu.edu/`)

vant for commercial providers as well.
From a consumer perspective, for advance reservation, the consumers must accurately know how long they need the services or have a good estimate of their demand in order to efficiently use the reserved service. Providers benefit from a better demand identification and forecasting. They can schedule the resource usage efficiently for the different kinds of offered services.

Fluctuations in demand and uncertainty:
Changes in demand are natural in many industries. In the case of demand for airline tickets there are strong variations, i.e. some customers travel only on certain weekdays or depending on the season. This leads to additional uncertainty about future demand and makes the demand-management decisions more complex. Consequently, the company is exposed to a certain risk of poor decision-making.
The demand for Cloud services also varies depending on a company's changing business requirements for Information and Communication Technology. A famous example is Animoto[38], which had to scale from 50 instances of Amazon's EC2 to 3500 instances within three days. As mentioned above, this is practically impossible without Cloud services. It is also hard to predict how long applications will run in different environments (Barker et al., 2009). Thus, applications and especially complex workflow systems with various dependencies between computing jobs induce a big uncertainty in runtime. Another issue is outage in the Cloud, which is a key part of the SLA (Leavitt, 2009). Providers have proven not to be perfect. Figure 2.5 depicts some of the outages announced in the news between January 2008 and June 2009. A provider whose systems are still online can face an unpredictable demand for his services during the outages of other providers. For example, the six hours outage of Salesforce already reduces the availability of the service per year to 99.93% or 99.2% for the outage of FlexiScale in 2008.

Heterogeneity of customers:
It has already been mentioned that it is a very common method for companies using Revenue Management to enforce price discrimination. Of course, perfect price discrimination, also known as first degree price discrimination, is extremely difficult to establish, especially when a seller faces at least one competitor. Accordingly, the offered services have to be simultaneously differentiated by adding or restricting certain features to reach the desired customers in every price class. In most cases, the demand for these price classes differ particularly in their valuation for special service features and price sensitivity (Talluri and van Ryzin, 2004b). For example, Amazon started the EC2 service in 2006 with a pay-per-use model. In March 2009

[38] Animoto (`http://www.animoto.com/`)

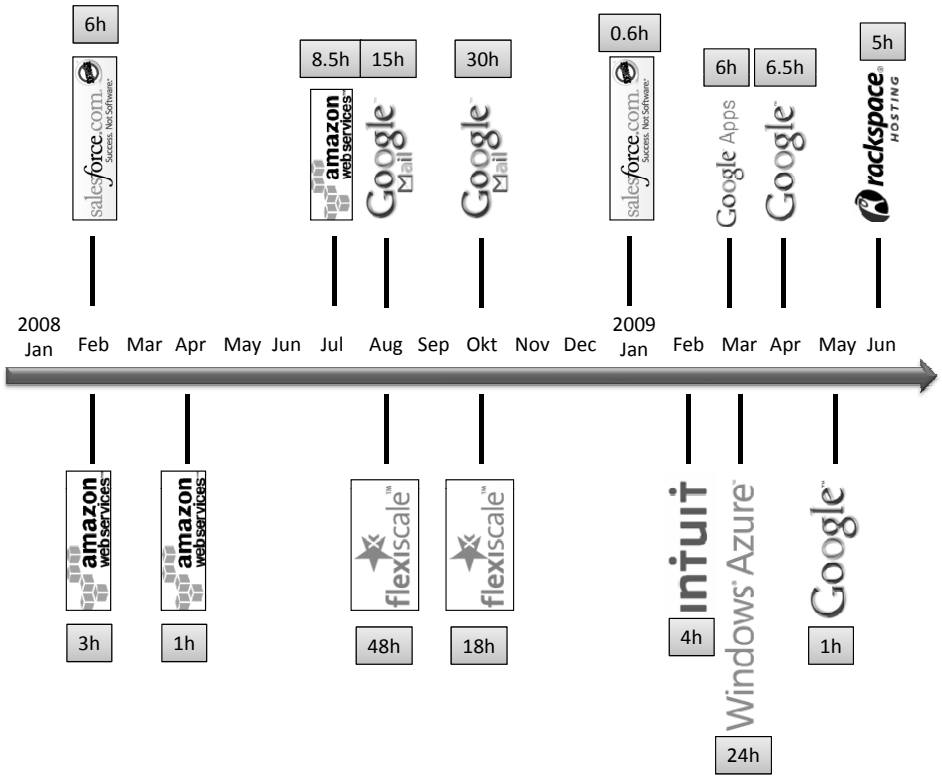


Figure 2.5: Outage examples from January 2008 to June 2009

they introduced "Reserved Instances". Consumer pay a one-time fee and can use the instances at a lower rate per hour than the standard on-demand price for a finite time. In December 2009 Amazon announced a new model called Spot Instances. Every consumer submits a bid and if the bid is higher than the current market price, his instance will run. Otherwise, it will be frozen and restarted at a later point in time, if the market price falls again below his bid. Hence, Amazon offers different pricing models for the same service. However, the third service comprises uncertainty when the instance will run, but the price is usually much lower than an on-demand instance.

To establish service differentiation by Internet Service Providers, Nair and Bapna (2001) proposed to use quality of service as the basis for a segmentation. This suggestion can be directly adapted to Cloud Computing: For example, a service level might indicate the minimum percentage of availability, which is defined in a SLA. Additionally, different needs for resource combinations between price segments might be identified and with advance reservation, restrictions can be additively adopted for reservation changes.

Price segmentation:

In the Internet context McKnight and Bailey (1997) already emphasized the necessity of pricing models, in particular, flat-rate models, capacity-based pricing and usage-

based pricing. Different pricing models can separate the demand for certain service. Users with unpredictable usage and few peaks in their usage behavior may prefer a usage-based pricing model, if they will gain less utility from a flat-rate model (depending on the price for flat-rates). Users with high-demand are better off with a flat-rate model. Different prices for the pricing models can also result in price discrimination. McAfee (2008) identified three conditions when price discrimination may apply: "consumers differ in their demands for a given good or service, a firm has market power, and the firm can prevent or limit arbitrage". Currently, Cloud service offers are heterogeneous between the providers. All pricing models and different kind of service levels characterize the different prices for Cloud services. The goal is to serve different kind of consumer needs, if consumer show a heterogenic demand. Hence, the price segmentation is executed by the provider, but it highly depends on the consumers' acceptance and needs.

Overall, the requirements from Revenue Management seem to fit to the Cloud Computing domain. However, some research questions are still to be answered. In particular, the consumer behavior in Clouds has been very convoluted. No attempt has yet been made to analyze how consumers select the offered service and which attribute of the services they value most. The required analysis for the sample question and further questions will be approached by appropriate research methods as described in the subsequent section.

2.3.2 Applied Methodologies and Research Questions

Research in the Information Systems discipline is a contentious issue in the community. According to Hevner et al. (2004), two paradigms dominate the research area: behavioral science and design science. The goal of behavioral science is to develop and justify theories for understanding organizational and social interactions by designing, implementing and analyzing Information Systems. The outcome is a more efficient and effective way of managing information in an organization or in a community. Design science follows the engineering approach. The artifacts created in this context seek to create innovative methods and tools to solve an existing problem (Simon, 1996). Artifacts comprise four phases: *constructs* (vocabulary and symbols), *models* (abstraction and representations), *methods* (algorithms and practices) and *instantiations* (implementation and prototype systems). Hevner et al. (2004) argue that both paradigms are complementary and Information Systems research can contribute to solving problems in the productive application of IT. Technology and human interaction with this technology are inseparable. However, Hevner et al. (2004) do not take into consideration how the interaction with other disciplines affect the research methods in the Information Systems discipline. Furthermore, research in Information Systems can also be executed on a conceptual

level without the instantiation phase of the artifact. The definition of Banker and Kauffman (2004) provide a more comprehensive view on the research streams in Information Systems. They identify the following five streams:

- Decision support and design science
- Value of information
- Human-computer systems design
- Information Systems organization and strategy
- Economics of Information Systems and Information Technology

The relevant stream for this thesis is the *decision support and design science stream*. Related disciplines in this stream are operations research, computer science, economics, marketing and strategic management (Banker and Kauffman, 2004). Revenue Management is an interdisciplinary approach taking economics, statistics, software tools and strategic decisions into account to maximize the revenue (Secomandi et al., 2002). Talluri and van Ryzin (2004b) stress the relevance of marketing methods for Revenue Management to identify consumer behavior in order to apply the mathematical models to practice in a better way. Hence, there is a great concurrence between these two research methodologies.

This thesis is divided into two parts. Chapter 3 analyzes the heterogenic character of Cloud service consumers. Its goal is to elicit the preferences from consumers for certain Cloud services at the infrastructure level. This approach gives a first indication which attributes are valued most by consumers. Preferences can be revealed by marketing methods. Talluri and van Ryzin (2004b) and Ng (2005) suggest conjoint analysis as an appropriate method to identify preferences in the Revenue Management context. As mentioned in the previous section, Revenue Management is applicable from a provider's perspective. But it has to be analyzed whether the Revenue Management framework is applicable from the consumer's perspective as well. It is not clear if consumers would endorse price discrimination or different kind of service levels with different prices. Moreover, the price sensitivity of Cloud consumers is an important topic. It is interesting to know the condition under which price discrimination is accepted and what valuation for a certain service is appropriate. Following research questions will be answered in Chapter 3:

RQ 2.1. *Are revenue management models applicable to Clouds so that consumers would accept the pricing and capacity management policies of the providers?*

RQ 2.2. *Under which conditions does a Cloud user accept price discrimination?*

RQ 2.3. *Which service attributes of an Infrastructure-as-a-Service (IaaS) offer are most valuable to the customer?*

The second part of this thesis (Chapter 4) concentrates on the scenario of scarce resources and how consumer requests in such a scenario can be accepted or rejected in a heuristic manner. Cloud providers will face a stochastic demand with a frequent number of incoming requests over a finite time horizon. Requests will appear for different kinds of offered services by the provider. The provider must decide whether to accept a customer at a certain point in time without rejecting demand for higher-valued services or face under-utilization of the resources at the end of the period. Different kinds of algorithms already exist and the proposed heuristic considers a scenario, where updates of bid prices cannot be performed after each request. Thus, the heuristic automatically updates the bid prices according to predefined parameters. These parameters are determined offline via a genetic algorithm before the entire time period starts. Banker and Kauffman (2004) indicate that simulation is a common research method to analyze and solve the identified problem in the Information Systems discipline. Genetic algorithms have been proven to perform well for simulation based optimization (Holland, 1975). Hence, the challenge is to find a self-updating bid price algorithm, which suits the Revenue Management scenario:

RQ 2.4. *How accurately can a simple linear function approximate well known algorithms for bid price calculation without reoptimization between two or more timeslots, taking the assumptions and requirements from Revenue Management in general and from Cloud Computing into account?*

Simulation as a research method is applied here for two purposes. First, the parameters for CBPP are determined via simulation-based optimization to identify the optimal values. Second, simulation help to understand complex interaction between various parameters in different kind of settings. Hence, data for statistical analysis is created via simulation in order to reveal the dependencies.

Currently, Cloud service providers are mainly technology-driven. This thesis focuses on consumer requirements and analyzes if providers can apply Revenue Management methods to the Cloud. While Chapter 3 identifies the preferences of consumers, Chapter 4 assumes that these preferences are already considered in the design of the services. Consequently, prices and services are set and the consumer will request these services, where the proposed heuristic comes into play.

Part II

Model Design, Implementation and Evaluation

Chapter 3

Customer Choice in Clouds

What business thinks it produces is not of first importance. What the customer thinks he is buying, what he considers value, is decisive. And what the customer buys and considers value is never a product. It is always utility, that is, what a product does for him.

[Peter Drucker, 1974]

In this chapter the preferences of Cloud service consumers will be analyzed and the applicability of Revenue Management concepts will be discussed. A survey was conducted to analyze their preferences. After the introduction, the related work about previous surveys are discussed and the conjoint analysis method is described in detail. The steps for setting up a conjoint analysis are presented. To understand the customer choice models from Revenue Management in detail, several approaches are compared and important aspects for the questions are derived. In Section 3.3.1 the descriptive questions for the conducted survey are determined and the research questions are outlined. The survey design as well as the stepwise development of the conjoint analysis are explained in section 3.4. Section 3.5 describes the results from the survey.

3.1 Introduction

Since the beginning of trade, vendors have faced the problem of how to price their goods and what quantity to produce in order to fulfill the demand. For example, the vendor could be a farmer who has to decide on the amount of fruits he harvests on a day-to-day basis and on the prices he sells his fruits at the market. Even from this very simple example one can understand the difficulty of the problem. There is uncertainty about the future market demand as exogenous factors influence customers' buying behavior and customers' valuation of the product. This uncertainty

of customers' valuation leads to the problem of setting prices, not to deter potential buyers on the one hand and not to loose potential profits on the other hand. Moreover, customers act strategically. In the previous example, this could be a customer waiting for the main harvest period before satisfying his demand. At that time the supply is high and it is not possible for the seller to postpone his sales because otherwise the fruit would get spoiled. In almost the same manner the seller can delay when he harvests his crop and offers his products at a time of low supplies and high demand.

In this sense the problem of Revenue Management is a very old idea. In fact, the innovation introduced by Revenue Management does not lie not in the decision management process itself, but rather, as Talluri and van Ryzin (2004b) define, "in the method of decision making - a technologically sophisticated, detailed, and intensely operational approach to make demand-management decisions". This raises the questions whether Revenue Management is applicable to Cloud Computing and how this can be done? These questions were already answered partially in Section 2.3. However, only the requirements for the applicability were analyzed. Furthermore, it is interesting to investigate, whether customers for Cloud services will accept Revenue Management methods like dynamic pricing strategies of providers in practice. Revenue Management concepts like accept/reject policies, dynamic pricing or advance reservation are not always accepted by the consumers. Consumers in some domains are not used to dynamic prices and occasionally deny it. In 2000, Amazon for example tried to introduce dynamic pricing for the online store. Customer complained about the frequent price changes for certain products and Amazon stopped the price changes (Weiss and Mehrotra, 2001). Today, Amazon is still changing the price, but it is not as frequent as in 2000. Therefore, a survey as a part of this thesis was conducted to understand the customers' perception of Revenue Management methods to avoid an unsuccessful adoption like Amazon. The design of sophisticated services and computation of the appropriate price requires an interpretation of users' preferences and requirements (Chellappa and Gupta, 2002).

3.2 Related Work

This section is divided into three parts. At first, previous surveys are presented to give an overview of the results achieved before the survey conducted in the work at hand. Then, the advantages of the chosen method to elicit the customers opinion, namely conjoint analysis, are outlined. The theory for the customer choice analysis was derived from the literature related to Revenue Management, which is described in Section 3.2.3.

3.2.1 Previous Surveys

Most Cloud service offers are mainly driven by companies creating innovative services. Since the term Cloud Computing has been coined, a bunch of new services have arisen or were rebranded to Cloud services. However, the introduction of these offers were not based on a thorough market analysis. Software services are not very costly like other products (e.g. pharmaceutical) and thus can be tested on a trial an error basis. Google is a successful company, which has focused on web-based services. Even applications like Microsoft Office have been transferred to the web by Google (Google Docs[1]) and it has been improved step by step, although its functionality has neither outperformed nor superseded Microsoft Office. An examination of the demand is completely missing. There are several concerns from the customer side not to use such Cloud services. The company Hosting.com[2] identified that 64% of the participants are afraid of the security risks in using Cloud services (Hosting.com, 2009). Data is often an intangible asset and companies avoid to outsource these data to a third party company. Especially the banking sector possesses sensitive data for evaluating future investments. The transfer of data as well as the storage on a server should be protected against intruders into the system or electronic eavesdropper. In another survey conducted in 2008 by CIO.com[3], 59% of the participants supported the statement about security concerns (McLaughlin, 2008). The survey from Avanade determined that as much as 72% have more confidence in internal than external computing resources (Leipold et al., 2009). The main reasons are the lack of security and control.

The outsourcing of data to a Cloud provider is coupled with the loss of control over the data. Both studies pinpoint this issue as a major concern besides security. The outsourcing company does not know how the data is managed and who is using or distributing the data. Since the data transfer depends on the Internet, the servers hosting the data cannot be completely cut off from the outside world and only hosted inside the company. Furthermore, in case of a downtime of the servers, the company depends on the performance of the provider, i.e. how quick he can solve the problem. An in-house solution would give the choice to ask different maintenance providers to tackle this issue instead of relying on one partner. In urgent cases more money can be spent to prioritize this issue.

Although these obstacles persist, users still see the benefits of Cloud services. According to Hosting.com (2009) the primary reason to switch to Cloud Computing is cost saving (34%) followed by high availability (17%) and performance as well as pay-per-use pricing model (each 12%). In the ongoing research from IDC analyst

[1]Google Docs (`http://docs.google.com`)
[2]Hosting.com (`http://www.hosting.com`)
[3]CIO.com (`http://www.cio.com`)

Gens (2008), it transpired that customers prefer the ease of deployment of software component (63%) and also the pay-per-use model (61%). Cost savings are mentioned in third place (57%). McLaughlin (2008), on the other hand, ranks scalability as the most important factor. Secondly, saving hardware and maintenance cost are listed. Although the answers of all studies seem to be heterogeneous, the cost seems to be the main driver for moving to the Cloud. However, the Cloud is not a solution considered by all companies. In fact, 75% of the participants of a German IDC survey did not consider using Cloud services yet (IDC Research, Inc., 2009). Avanade says that 80% of the companies, who do not use Cloud services, are not planning to use them in the future.

All of the above mentioned surveys were conducted by companies. Most of the participants were Chief Information Officers (CIOs) and not Cloud service users. Currently, there is no scientific survey available for Cloud Computing usage. Moreover, this chapter focuses on understanding the application and design of Revenue Management models to the Cloud. Thus, it is interesting to know from a customer or Cloud service user perspective, if concepts like advance reservation of computing resources or price discrimination will be accepted.

3.2.2 Conjoint Analysis

Conjoint analysis is known as a multivariate data analysis method (like the regression or variance analysis) to understand the dependency among the examined parameters of a product or service (Green and Rao, 1971; Green and Srinivasan, 1990). It is often used in marketing research to evaluate a product before it is introduced to the market. It is one of several methods to elicit preferences from customers. A utility of a service for every customer can be derived from the evaluation of the customers' perception for different combinations of service attributes. Fishburn (1967) lists 24 different methods for empirical determination of additive utility functions. These can be distinguished into two classes of models. One way is to investigate the utility of attributes and their characteristics separately. Another way is to derive the utility by using holistic statistical methods. The traditional conjoint analysis is part of the second group of methods using a decompositional approach. It is a widely accepted method for identifying customer preferences (Hahn, 1997; Kaul and Rao, 1995). In general, two approaches dominate this research area (Figure 3.1):

- The compositional approach allows the participants to evaluate the attributes separately. Afterwards, the attributes are combined to give a total valuation of a product or service. The total valuation is often determined by a simple linear function adding up the attributes' utility.

- In the decompositional approach the user is faced with different services distinguished by their attributes and he has submitted a valuation for these services. Afterwards, the total valuation is broken down into every single attribute by multivariate or psychometric analysis.

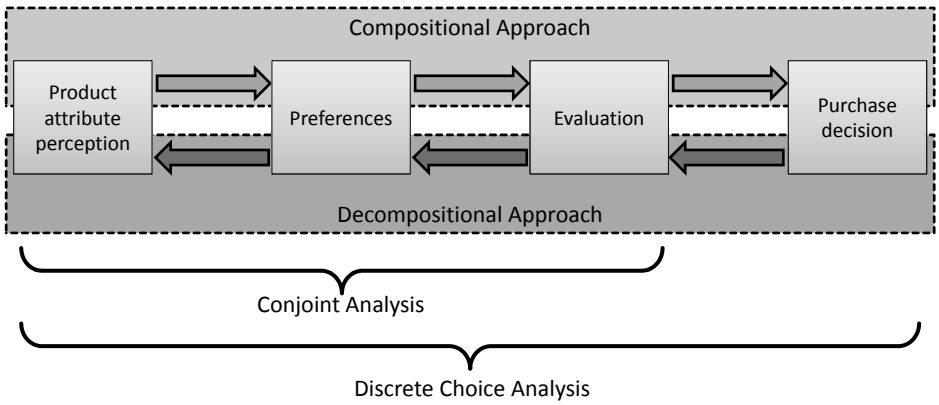


Figure 3.1: Comparison of compositional and decompositional models (Hahn, 1997)

Both approaches can lead to completely different results for the same surveyed domain (Green and Srinivasan, 1990). Examples for compositional approaches are the Self-Explicated method (Srinivasan, 1988), the Analytical Hierarchy Process by Saaty (1980) or the Multi-Attributive Value Theory (Keeney, 1969). The disadvantage of these methods is the isolated evaluation of every single attribute, which does not reflect a realistic scenario. Furthermore, it is hard for the participant to provide a valuation for a single attribute, since she often does not know or is not able to express it explicitly. Instead she can only know her benefit from the entire service. In the decompositional approach the participant faces a more realistic scenario by ranking the entire service (Akaah and Korgaonkar, 1983). Hence, this thesis focuses on the decompositional methods, in particular the conjoint analysis, and thus disregards the compositional approaches.

Historically, holistic ratings were seen as the central attribute of conjoint analyses (Green and Srinivasan, 1978), but that condition has been weakened lately resulting in a non-existent clear definition (Green et al., 1997). However, Teichert (2001) considers simulated decision making, decompositional approach, model specification and experimental design as the four key characteristics. Meanwhile, two additional characteristics indicate problems of the method: utility evaluation and individual approach. Since there is no linkage between the observed utility and the values of interest (e.g. market share), the inquirer has to transform the outcomes of the utility evaluation into choice decisions. Comparisons between utility evaluations of different individuals are questionable because of the individual approach of conjoint analyses. Therefore, conjoint analysis focuses on the analysis of individual utility functions and the aggregation is done in a second step.

One significant advantage of conjoint analysis compared to other methods for identifying preferences is the fact that the buying decision is explicitly modeled and thereby the decision process of customers is captured better compared to directly asking for their preferences (Teichert, 2001). In his paper, Neslin (1981) demonstrates the benefit of using statistical methods, i.e the Analysis Of Variance (ANOVA) model, for linking features to perceptions compared to self-stated methods. The author's conclusion is that statistical methods are analytically rigorous and allow to examine more complex interaction effects. Hsee (1996) outlines that a direct evaluation of preferences is impossible for attributes that are dependent with regards to their content, indicating the need for a decompositional approach like conjoint analysis.

The general process of executing a conjoint analysis is not unambiguously defined, but Green and Srinivasan (1990) propose a widely accepted and applied procedure which is divided into six steps (Table 3.1). At first, the relevant attributes of the service have to be identified and the parameter value must be reduced to a few. Then, the preference model has to be determined to understand the dependencies of the attributes and their impact on the total utility. Thirdly, the choice of an appropriate survey model is important to estimate the amount and quality of data. The collection of the data is done in the fourth step. Afterwards, the evaluation requires an estimation method to analyze the utility values of each attribute. Finally, the utility values are aggregated for each participant to a total utility for creating an overall ranking of the attributes (Hahn, 1997). Since conjoint analysis is a multivariate approach, the independent variables are the attributes and the parameter values, whereas the preferences for the fictive good describes the dependency. Moreover, the preference of a single customer is of no value, if it is not representative for a group of customers. Thus, the aggregation of the individual valuation is more relevant for the service designer (Backhaus et al., 2008).

The goal of the conjoint analysis is to analyze the preferences of the customers and to identify the utility of every single attribute via a linear-additive model (Teichert, 2001). In general, conjoint analysis comprises different approaches. Three well-known approaches are the *traditional conjoint analysis,* the *choice-based variant* and the *adaptive conjoint analysis.* In the traditional version, the participants have to evaluate all possible combinations of the parameter values. The drawback of the traditional method is the restriction of the number of attributes. To avoid too many combinations of the different levels, the adaptive conjoint analysis first explores the important attributes in a compositional way and then uses a reduced set of important attributes and levels for the decompositional survey. Hence, it allows to take more attributes into account. For a small set of attributes the adaptive conjoint analysis does not provide significant improvements (Teichert, 2001).

Table 3.1: Steps for operating a conjoint analysis (Green and Srinivasan, 1990)

Step	Methods
1. Preference Model	Vector model, ideal-point model, part-worth function model, mixed model
2. Data collection method	Two-factor-at-a-time (trade-off analysis), Full-profile
3. Stimulus set construction	Fractional factorial design, random sampling from multivariate distribution, Pareto-optimal designs
4. Stimulus presentation	Verbal description (multiple cue stimulus card), paragraph description, pictorial or three-dimensional model representation, physical products
5. Measurement scale for the dependent variable	Rating scale, Rank order, paired comparison, constant-sum paired comparisons, graded paired comparisons, category assignment
6. Estimation method	Metric methods (multiple regression); Nonmetric methods (LINMAP, MONANOVA, PREFMAP, Johnson's nonmetric algorithm); Choice-probability-based methods (logit, probit)

Choice-based conjoint analysis has the advantage to directly elicit the choice behavior of the participants. Its binary choice structure does not require an underlying solid transformation model like the traditional approach. The user has to decide, whether to buy a sample product A or B (with their specified attributes) or none of them. However, the binary choices come along with the loss of information, since the utility cannot be derived from the binary choices. Hence, the information is more useful on an aggregated level than for individuals, e.g. to determine market share or sales volume. The aggregation process disregards the heterogeneity of the customer characteristics. The traditional conjoint analysis emphasizes the investigation of the individual utility function and thus aggregates the data in a second step via a transformation model.

The results of a conjoint analysis are often used to forecast the utility of the participants based on the estimated utility parameter. Hahn (1997) ascertained no significant difference of the forecast quality between the traditional conjoint analysis and the choice-based conjoint analysis, if a probabilistic estimation method is chosen. Moore et al. (1998), however, depict a difference between both methods and point out that the traditional conjoint analysis performs even better.

Since this thesis focuses on deriving the utility for every individual (a heterogenic character of the respondents are expected) a choice-based approach is not meaningful due to the above mentioned disadvantages (Teichert, 2001).

3.2.3 Customer Choice Models

The process of structural decision making can be divided into three steps. In the first step the focus is on the observation of historical demand and the estimation of customers' utility functions. This information is aggregated to generate a demand forecast and define the customer segmentation. The goal of all structural decisions is to clarify what to sell and how to do it. Pak (2005) outlines these decisions on a strategic level comprising the bundling of services, the differentiation of services to target customer segments, the selection of appropriate sales methods and the design of the general price structure over a longer period of time. In contrast, day-to-day layered price and quantity decisions are made on an operational level. These involve decisions such as: what price to charge at a certain point in time, when to give a discount, what part of the total capacity to reserve for each customer segment and whether to accept or reject a specific sales offer. The structural sales decisions obviously have a substantial impact on the day-to-day price and quantity decisions. Although structural decisions are sometimes considered to be marketing decisions, it is essential to integrate customers' preferences and historical demand patterns into operational decision making and Revenue Management models (Boyd and Bilegan, 2003; Pak, 2005). Classical Revenue Management models fail to account for strategic customer behavior as they assume myopic customers (Belobaba, 1989; Gallego and van Ryzin, 1994). However, when customers act strategically, Revenue Management models need to incorporate such behavior into revenue maximization. The survey in this thesis aims at identifying customers' preferences at the structural level and at exploring how customers make their decisions on buying Cloud Computing services at the operational level. An example for the operational level decision would be customers deciding after comparing competitive offers or strategically delaying their purchase in expectation of better prices in the future. Shen and Su (2007) present a complete overview on recent research focusing on customer behavior. They distinguish between strategic customer behavior with intertemporal effects and customer choice behavior for multi-product settings. Strategic customer behavior includes models on dynamic pricing, capacity rationing and valuation uncertainty as well as, from a consumers' perspective, customer response to dynamic pricing. Models of customer choice behavior consider demand dependencies resulting from customers' selection of a set of services as well as from substitutable or complementary effects across services. In what follows the important aspects of

these models will be explained in detail. An overview in tabular form can be found in the appendix A.3.

Dynamic Pricing under Strategic Customer Behavior:
Su (2007) describes a model of dynamic pricing taking endogenous intertemporal demand into account. Having a monopoly, the seller who has a finite inventory over a finite time horizon, maximizes his revenue by adjusting prices dynamically. Customers meanwhile can decide to buy the service, stay in the market and wait for a lower price, or exit the market. The population is heterogeneous along two dimensions having different willingness to pay and willingness to wait. The composition determines the optimal pricing policy. Hence, Su (2007) derives four distinct groups of customers: patient-high-types, impatient-high-types, patient-low-types and impatient-low-types. For example, the impatient-low-type, on the one hand, is not willing to wait for a lower price and on the other hand, is not ready to pay a high price. Furthermore, Su (2007) determines optimal pricing policies for cases when each customer segment dominates the market. The findings are summarized in a framework considering markups and markdowns. Markdowns should be applied when impatient-high-type customers and patient-low-type customers dominate the market, while markups increase revenues when patient-high-type customers and impatient-low-type customers are present. Meanwhile, the benefits of strategic behavior are that demand is not immediately lost, as it possibly leads to sales after markdowns and that there is increased competition for availability at lower prices. This leads to higher reservation prices and generated purchases.

In a series of papers, Levin et al. (2006) first present a stochastic, game-theoretical dynamic pricing model and later Levina et al. (2009) incorporated demand learning into their model. Customers' so called "degree of strategicity" is consistent with a discount factor which takes the value one, when customers are myopic and there are no discounts on future purchases and zero, when customers disregard future purchases (Levin et al., 2006). They demonstrate the existence of a unique subgame-perfect equilibrium pricing policy, providing equilibrium optimality conditions for both customer and seller. Vendors who do not account for strategic behavior and do not use the strategic equilibrium pricing policy, receive lower total revenues. The model proposed in Levina et al. (2009) develops "an adaptive procedure that permits learning of consumer response through observation of sales over successive planning horizons". They show the robustness of the learning approach by modeling a simulation-based technique to determine optimal prices.

In contrast to the previous models Yin et al. (2009) study the effect of inventory information on strategic customers. They consider a model with two types of customers and two in-store display formats to analyze effects on the retailer's profit and

order quantity. All customers are either myopic or strategic and arrive in a Poisson process. In their case, the retailer can display all available units or one unit at a time. As a result, they prove analytically that firms earn higher expected profits and order more, by displaying one unit at a time on the sales floor, when all customers act strategically.

Capacity Rationing:
In another group of papers the interaction between pricing and rationing is studied. When facing strategic customers, rationing can be considered as a strategic tool for firms along dynamic pricing. Thereby these models investigate the likeliness that customers purchase earlier at higher prices.

Gallego et al. (2008) suggest a two-period model with deterministic demand. As the remaining capacity is not visible, customers need to interact with the firm to estimate the capacity. In that model customers update their expectations in each period with respect to the outcomes of previous periods. This process converges to an equilibrium point. The conclusion is that it is beneficial for a firm to ration capacity by disposing of excess units instead of training customers to wait for sales when all customers are strategic.

In the context of supply chain management, Su and Zhang (2008) apply a model of customer behavior and outline positive effects of quantity and price commitment for the firm's revenue. The idea is derived from the news vendor problem where leftover units have to be sold at a lower price. To overcome customers' strategic behavior – waiting for price reductions – the seller commits to a certain capacity level or price implemented through contractual arrangements. This leads to a better overall performance in decentralized supply chains.

Valuation Uncertainty:
Valuation uncertainty relates to the fact that customers strategically delay purchases, because they anticipate better deals in the future (see *Dynamic Pricing under Strategic Customer Behavior*). Another reason for valuation uncertainty is the lack of information or the need to wait until more details are released. There is only a small amount of work available in Revenue Management for valuation uncertainty. The most important one for the survey is from Koenigsberg et al. (2008) who based on the empirical analysis of easyJet's pricing patterns study stated: "the conditions under which offering a last-minute deal is optimal under the single price policy". They found that, especially when customers are uncertain whether firms offer last-minute deals, it is beneficial to sell the remaining capacity at lower cost.

Customer Response to Dynamic Pricing:
It is important to understand strategic customer behavior in revenue management. Several models were proposed to explain how customers respond to firms' pricing strategies. They can either wait for special offers or accept prices, if they fall below a threshold value.

In their model, Anderson and Wilson (2003) assume that firms set prices dynamically according to Belobaba's EMSR rule (Belobaba, 1989) or the optimal pricing policy suggested by Gallego and van Ryzin (1994). Both are widely accepted in practice. Furthermore, they assume that firms have a fixed capacity and set protection limits to restrict the number of units sold contributing only low-revenue. High-revenue generating customers are expected to arrive later. The authors then calculate the probability that at the end of this procedure, part of the resource remains unsold. They show that if this probability is high enough, high-revenue customers may wait for last-minute discounts. In addition, they perform numerical studies to investigate the effects of this strategic waiting behavior on the firm's revenue.

Zhou et al. (2005) model the case of a single strategic customer facing the optimal pricing policy by Gallego and van Ryzin (1994). They find that the customer should immediately buy if the price is below a threshold depending on the his valuation at a time. Subsequently, they investigate the case with multiple customers and show that strategic customer behavior benefits the seller. The reason is that customers are not immediately gone, if a service is not available. They are open to other service offers or return in a later point in time, when the price is lower.

Choice from a Set of Services:
In the context of airline Revenue Management, customer choice models investigate how customers choose between firms' different services. Thereby, firms have to decide which service to make available to customers and what price to charge. Talluri and van Ryzin (2004a) introduced a Revenue Management model under a discrete choice model of customer behavior where the supplier has to choose at each point in time which set of services to offer. The customers then choose an option from that subset including the no-purchase option. The latter option can motivate the customer to buy this service or a similar one from another Cloud provider.

The papers from Zhang and Cooper (2005, 2009) consider a framework of parallel flights with separate inventory and customer choice behavior. In their first paper, they formulate the problem of parallel flights with similar origin and destination as a Markov decision process. Furthermore, they derive computable bounds for the value function, and simulation-based procedures to obtain good policies. In their second paper they focus on pricing decision rather than on quantity decisions. They show that policies, motivated by bounds for the value function, dominate pooling

heuristics for symmetric problems. The approach can also be transferred to larger problems.

Substitution and Complementarity across Services:
Substitution and complementary effects are modeled using multi-dimensional demand functions having customers who choose among different services. Apart from these effects, methods like cross-selling or up-selling are also covered.

Based on the observation that offering complementary services in e-commerce settings has become very popular, Netessine et al. (2006) propose a model where cross-selling in the dynamic settings is identified as an opportunity complementary to single-service Revenue Management. In their setting, they consider a firm that manages a set of services and faces stochastic customer arrivals. The goal of their model is to select complementary services from the set and to define the price of such a package to maximize revenues. They find that their model is most effective when service inventory is approximately equal to expected demand.

Aydin and Ziya (2008) investigate the practice of up-selling by introducing a promotional service. The promotional service is offered at a possibly discounted price only if a customer purchases a different good. They investigate how the discount depends "on the inventory levels, [the remaining] time, type of pricing policy in use, and the relationship between the customers' reservation prices for the promotional product[4]" (Aydin and Ziya, 2008). The results show inter alia that under dynamic pricing the up-selling decision is independent of the level of inventory and the remaining time.

These six categories of customer choice models build the basis for designing the questions. Moreover, the classification from Su (2007) is required to classify the customers according to their willingness to pay and willingness to wait.

3.3 Model Formulation

The survey as well as this section are divided into two parts. The first part is based on descriptive questions. The theoretical motivation to asking these questions is outlined. Although the related work of customer choice in Revenue Management was analyzed in the previous section, the following section incorporates literature beyond Revenue Management and relates the descriptive questions in the survey with the theoretical background derived from the literature. In the second part, the appropriate hypotheses are generated to answer the Research Questions 2.1 and 2.3.

[4]The concept also applies for services.

3.3.1 Theoretical Background for the Survey Questions

As Armbrust et al. (2009) outline in their article, moving business operations to Cloud Computing providers is mainly a question of cost effectiveness. Therefore, users have to compare the Capital Expenditure (CAPEX) and Operating Expenditure (OPEX) when owning a datacenter to the OPEX for using Cloud services. CAPEX refers to the acquisition and installation costs while OPEX refers to the variable costs that originate during the operation. In addition, customers have different prices in mind and respond differently to dynamic pricing (Anderson and Wilson, 2003; Zhou et al., 2005). Thus, at first, the participants were asked about the appropriate standard CPU hour (**Questions 1 & 2**). This gave insights about the estimation of the users' valuation. Furthermore, in **Question 10** the participants had to reveal their expectation about the future development of price and quality of service.

The economic benefits for Cloud users increase in cases of underutilization of their own datacenter. In other words, Cloud Computing is extremely beneficial to those who need an elastic computing resource due to a volatile utilization of its own resources. For Cloud users this means shifting the risks of under- or over-provisioning to the Cloud provider. In addition to the shifting of risks and cost effectiveness, moving data processing to the Cloud leads to advantages in maintenance. However, Cloud providers consequently face the problem of handling the volatile demand of numerous users. Thus, from a provider's perspective the question is, whether consumers can somehow predict their usage behavior or if it is totally random (**Question 4**). In conjunction with Questions 1 & 2 it can be revealed, if uncertain customers can be convinced by a last-minute offer (Koenigsberg et al., 2008).

Currently, there is no dominant pricing policy that would help to control the utilization level of resources for Cloud providers. Pricing strategies currently applied throughout the industry also fail to maximize revenue and to control volatile behavior of the customer. Weinhardt et al. (2009) survey the applied pricing models and find that most Cloud providers use metered pricing to charge for their services on posted price basis. Apart from the pay-per-use policy, subscription models are applied as well. Although other dynamic pricing policies may result in more efficient allocations and prices (Lai, 2005), customers prefer more practical pricing policies that are easy to understand (Dasilva, 2000; Fishburn and Odlyzko, 1999). In a market that some consider purely competitive (Carr, 2005), meaning that a commodity or public utility is traded, the question remains how firms can differentiate themselves without getting locked into a ruinous price war. Netessine and Shumsky (2005) name it in the context of Revenue Management "the horizontal competition" among providers and analyze how horizontal competition affects the decision

about the capacity allocation for airlines in a two-player game. As Dixit et al. (2008) outline, the key is to use IT enhanced pricing strategies to determine customers' willingness to pay in a way that customers' perceive pricing strategies as fair. It would be interesting to know, whether the Cloud service consumer would change their provider depending on prices or for any other reason and if the consumer is interested in comparing prices at all (**Question 7, 8 & 9**).

Varian et al. (2004) strengthen the point from Dixit et al. (2008) by outlining the possibilities of differential pricing for information goods in contrary to physical goods. He suggests to reduce the speed of operation for lower valued services or to provide less capabilities. Apart from differential pricing, Bakos and Brynjolfsson (1999) suggest bundling of goods to increase the profitability. Netessine et al. (2006) outlined the benefits of cross-selling products. In the context of capacity rationing in supply chains Su and Zhang (2008) proposed contractual agreements to overcome strategic customer behavior. This raises three further questions in the Cloud Computing context:

1. Would consumers accept differentiated pricing, which would help providers to segment customers into separate classes according to their willingness to pay to increase revenue (**Question 11i**)?
2. Would consumers accept a subscription model (**Question 11v**)?
3. Do consumers prefer bundling of products over individual offers (**Question 3**)?

Maglaras and Zeevi (2005) theoretically analyze the quality of service for best-effort services vs. guaranteed services regarding bandwidth. Users can choose between these two services, which have non-substitutable characteristics. These characteristics are usually defined in SLAs. A question derived from this case is that if the consumer accepts a lower price for a lower performance level (**Question 5**).

According to Su (2007) markdowns should be applied for impatient-high-type customers and patient-low-type customers while markups increase revenues in presence of patient-high-type customers and impatient-low-type customers As outlined in previous section this leads to higher reservation prices and generate purchases. Furthermore, Nair and Bapna (2001) suggested a decision model for Internet providers. They stated that "request and service happen simultaneously. This is not the case in airlines and hotels, where the request is made at one time (making the reservation) and the capacity is used up at another (the flight taking off or the hotel room [getting] occupied)". However, the virtual lab in Raleigh requires advance reservation for computing resources and labs for the course instructors (see Section 2.3.1). Consequently, request and service happen at different points in time. Hence,

it would be interesting to know what types of customer a Cloud provider faces and especially if they accept future purchases (**Question 6**). This will foster strategic behavior (Levin et al., 2006).

In Clouds, providers hide information from their customers about the total capacity they possess. Of course, the customers would act strategically based on these information to receive a better price (Gallego et al., 2008; Yin et al., 2009). However, currently providers with a fixed price model like Linode[5] reveal this information to the customer. Thus, the question of how consumers evaluate such information is important (**Question 11iv**).

Moreover, it was interesting to see, if common sales practice would apply for Cloud Computing as well. Cross-selling is well-known from Amazon's bookstore. Companies have to think about how the bundles are created and how to price the bundles, which is a combinatorial problem. The challenge is to bundle and price the services in a revenue maximizing way (Netessine et al., 2006). The question is, if customers would generally pay attention to such complementary offers (**Question 11ii**). Furthermore, loyalty programs are widely applied in the airline and retail business Aydin and Ziya (2008). Customers can receive an upgrade for staying with the same provider and using his service for a certain period of time. Providers have to plan such offers in advance to consider it in the quantity and pricing decisions. How do customers value such an offer (**Question 11iii**)? These models indicate the variety of modeling approaches of customer behavior in Revenue Management. Based on the authors' findings the survey aims to test the options of transferring these models to Cloud services. Therefore, the topics of interest are the importance of prices and their willingness to change providers as well as the importance of prices when changing provider. Furthermore, users' opinion towards multi-service offers and the willingness to receive complementary offers are investigated. Another important aspect is users' valuation of the possibility to receive reduction for advanced booking or best effort services. Currently, the common opinion is that Cloud services are mostly unpredictable. However, the experience from the scientific driven Grid Computing domain is that users tend to increase usage of Grid resources few weeks before conferences or important events to run simulations or analysis on particle physics data. Therefore, the ability to predict one's usage behavior was inquired. An overview of all the questions can be found in the appendix A.1.

Table 3.2: Impact of consumer behavior, provider offer and trading object factors on the Revenue Management requirements

	Predefined factors of the services	**Service design influenced by**	
		the consumer	**the provider**
Heterogeneity		X	
Demand uncertainty		X	
Inflexibility	X		
Perishability	X		
Advance reservation		X	X
Price segmentation		X	X
Multiple service offers			X
Overbooking			X

3.3.2 Hypotheses

First and foremost, the motivation for this survey is to elicit the opinion of the consumers about typical Revenue Management characteristics (e.g. price discrimination), when Revenue Management methods are introduced for Cloud scenarios. In Section 2.3.1, the requirements for a successful application were derived from the existing literature. These characteristics are influenced by consumers' behavior, providers' offer and the trading object itself. Although Revenue Management techniques are implemented and controlled by the provider, consumers can deny offers or pricing policies like the dynamic pricing approach of Amazon in 2000[6]. For example, customers may not accept price discrimination or they cannot predict their behavior. Hence, advance reservation cannot be applied successfully in the Cloud. Requirements like inflexibility or perishability are predefined by the service itself. They cannot be changed, neither by the provider nor by the consumer. However, heterogeneity and demand uncertainty are determined to a certain extent by customers' behavior and their ability to predict their service usage (see Table 3.2). Presumably, certain groups of consumers will have different kinds of demand characteristics. This leads to the following hypothesis:

Hypothesis 3.1. *The usage frequency of computing services depends on the role of the user.*

It is assumed to have a group of consumers, who are able to predict their demand in advance. Though, it cannot be concluded that these consumers would also book

[5]Linode (`http://www.linode.com`)
[6]see Section 3.1 or Weiss and Mehrotra (2001)

in advance. From the provider's perspective, predictable demand is beneficial, since he can reserve and allocate the required resources and calculate his profit. Thus, consumers, who can predict their behavior and book in advance, clearly put the provider in a favorable position:

Hypothesis 3.2. *Users, who are able to predict their behavior, also book the resources in advance.*

Companies are able to offer different kinds of services by making only small or restrictive changes to a service and keeping the underlying good or resource consumed by these services as is. This is also known as versioning (Varian, 2002). For example, services can be categorized into free and premium products, where the free version only offers a low bandwidth access to the storage and the premium one yields a high connection speed. The consumers' demand is segmented by the price. Thus, it would be interesting to know, whether users who ask for a lower price also accept a lower performance level of their requested service:

Hypothesis 3.3. *Users, who support price discrimination, also prefer best effort services.*

The RQ 2.1 (see Section 2.3) can be answered by analyzing hypotheses 3.1, 3.2 and 3.3.

Moreover, another RQ 2.2 focuses on the conditions, when a consumer would accept price discrimination. In the survey, three aspects were analyzed to answer this question. Hypothesis 3.3 already mentions the condition of best effort service usage in conjunction with price discrimination. The other two aspects are whether user favoring price discrimination would also book in advance or prefer a subscription model. If consumers support price discrimination and do not book in advance, the discrimination has to focus on differentiating the services itself by offering different support level or limited software enhancements for the low price segments. Otherwise, price discrimination can be mainly based on the intertemporal dynamic pricing strategy. Subscription models as the second aspect can also be offered for different price segments. If the participants do not necessarily support price discrimination and subscription model, then they do not necessarily expect a lower price for a subscription model than for an instance hour. However, it is common in practice to receive a discount for buying a large amount of service in advance. For example, Amazon offers so-called *Reserved Instances* to pay a one-time fee valid for one year in order to receive a discount of 70% for each CPU hour of a Linux instance.

Hypothesis 3.4. *Users, who support price discrimination, also prefer to book in advance.*

Hypothesis 3.5. *Users, who support price discrimination, also prefer a subscription model.*

Subsequently, consumers have different preferences according to the offered services. In particular, the attributes defining the services and their parameter value have an influence on the consumer decision for a service. It is necessary for a provider to know, which attributes have a greater value than others in order to offer the appropriately designed services. The conjoint analysis in this survey focuses on the IaaS level and tries to identify the crucial attributes in order to answer RQ 2.3.

3.4 Customer Choice Survey

This section commences with the explanation why this specific survey design was chosen. Subsequently, the steps for conducting a conjoint analysis are operationalized as described in Section 3.2.2. After determining the Conjoint analysis design, the choice sets are selected and their relevance for the analysis is motivated.

3.4.1 Survey Design

Preparing an empirical analysis to test theoretical assumptions is a well approved method in marketing research. There are three approaches to collect empirical information: oral interview, written survey or telephone interview. With the advent of the Internet, online surveying has become a fourth accepted option. As the target group of this survey is actively creating the future of the Internet addressing them through an online survey is a logical approach. In order to reach the highest possible number of participants and meanwhile offer customers an anonymous participation, the survey is hosted on a website with open access instead of an invitation based approach.

The Web-based approach includes several advantages in processing a survey. The respondents cannot be biased by the relationship with the interviewer. This leads to more objective results. Likewise, the survey data is directly stored into a database avoiding the error-prone transfer of results. This automation not only cuts the cost of data processing but also allows great scalability in the number of participants with very little additional costs. The subjective perception of an online survey is shorter compared to traditional approaches leading to higher quality in data due to less exhaustion (Decker, 2001).

However, there are some disadvantages when conducting a Web-based survey. These can either be technical problems, like displaying errors in different browsers and the response time of the Website, or problems related to the participants' iden-

tity. Homburg and Krohmer (2003) warn about the fact that anonymous participants may not be part of the target group and thereby lead to less reliable results. Especially when offering monetary incentives or vouchers the primary goal of respondents might be to get these prizes, which may eventually lead to multiple participation. To ensure that there are no such effects, there will be no monetary incentives for the participants. In the underlying survey, those specifying their email address receive a summary on the outcomes of the survey in order to validate that the majority of participants is included in the target group. At this point, it should be noted that due to the self-selection bias the group of people being studied depends on the participants' decision to take part. This can lead to a distortion of results, if answers from the non-participating group of people differed from those who participated. Nonetheless, the advantages of this surveying method outweigh the obstacles and the fact that the target groups' primary area of work is the Internet supports the conduction of a Web-based survey (Welker et al., 2004).

Oppenheim (2000) outlines a detailed summary of how to develop the survey design. Bearing this in mind, the survey had a subtle design (e.g. moderate colors) and no unnecessary graphics to speed up the load time. The questions were short and explanations were only added where needed. In order to minimize the flop rate, the survey only contained the relevant number of questions (Lütters, 2004). The processing time for the 26 questions was approximately 10 to 15 minutes. The website consisted of five pages which can be found in the appendix A.

At the beginning, a welcome screen was shown where the goal of the survey was outlined and the privacy policies were explained. Subsequently, the questions were split into three groups. While the first group of questions focused on customer behavior, the second group of questions aimed at identifying customers' preferences using a conjoint analysis. A third group of questions was related to sociodemographic factors like age, residence, industry and role of the respondents. After answering all questions and submitting the data a notification was given to show that the survey has been completed.

Before the roll-out of the survey, a pre-test was performed with a group of people clarifying any misunderstandings. Based on these findings the questionnaire was adopted as suggested by DeShazo and Fermo (2002). Target group for this survey were users of Cloud services. To reach this target group 750 users were directly addressed via email and an undefined number was addressed indirectly via postings in Cloud Computing related user groups and in social networks like Xing[7] and Facebook[8] or via blog[9]. The e-mail receivers were selected from related research

[7] Xing (`http://www.xing.com`)
[8] Facebook (`http://www.facebook.com`)
[9] Cloudytimes Blog (`http://www.cloudytimes.com`)

projects like D-Grid[10], a German initiative which fosters the development of Grids in Germany and active user groups with their focus on Cloud Computing[11]. The invitation included a link directing recipients to the website containing the questions. The survey was conducted in February 2009 in line with the four to eight week period suggested by Werner and Stephan (1998) for data collection of online surveys. Within this period satisfying 65 data sets were collected.

3.4.2 Preferences, Stimuli and Data Collection Method

When designing a conjoint analysis, two factors have to be taken into consideration. First, a definition of the stimuli and second the number of stimuli. A stimuli is defined as a combination of different product characteristics which is presented to respondents for evaluation (Green and Srinivasan, 1978). There are two types of stimuli that are common in conjoint analysis design. Using the two-factor method or so called "trade-off analysis" by Johnson (1974) respondents have to rate all combinations of two attributes at a time. That means that for every pair of attributes a trade-off-matrix is developed so that respondents have to rate $\binom{n}{2}$ trade-off-matrices surveying n attributes. The full-profile method considers combinations of a single levels of each attribute. The number of stimuli thereby depends on the number of attributes and the respective levels. For n attributes with m_1 levels for the first attribute, m_2 levels for the second attribute,..., and m_n levels for the n-th attribute, there exist $m_1 \times m_2 \times ... \times m_n$ different stimuli in total.

In terms of expenditure of time and requirements towards the respondents, the trade-off analysis has an advantage over the full-profile method. The evaluation of two attributes together with their respective levels simplifies the survey design and needs less explanations to the respondents. With higher numbers of attributes and levels, the number of stimuli grows faster using the full-profile method. The full-profile method offers a set of alternatives with all attributes and their parameter values reflecting a more realistic scenario for the participants. The unrealistic decision scenario and the empirically validated inferiority make the trade-off method unattractive and thus the full-profile method is chosen for this survey (Segal, 1982; Safizadeh, 1989).

3.4.2.1 Evaluation of Stimuli

For the conjoint analysis, a 7-point Likert-type scale is used to evaluate the sample products. This type of rating scale is called semantic differential and was introduced by Osgood et al. (1957). The concept of semantic differential is a multi-item measure

[10] D-Grid (`http://www.d-grid.de`)
[11] e.g. Google Groups `http://groups.google.com/group/cloud-computing`

that allows a relatively direct indication of attitude. Although nowadays it is used in various contexts, it was originally designed to measure the meaning of a concept as it consists of bipolar evaluative adjective pairs, such as good-bad (Ajzen, 2005). The 7-point scale is one of the most common formats. Malhotra and Peterson (2005) and Dawes (2008) found that there is little difference in using either a 5-point, 7-point or 10-point scale. Other common methods beside the rating scale like the dollar metric or the constant-sum method are discussed by Green and Srinivasan (1978). All these methods are based on metric scales whereas pairwise comparison and rank-based scale are non-metric approaches. Wittink et al. (1994) state that rating scale and rank-based scale are most often used in practice and research. However, metric scales provide more information about the relation between different service rating than the other. Hence, the rating of two different items can provide the information, how strong one item is preferred Likert scales produce usually ordinal data, although an assumption of symmetry of response level by offering odd number of levels allows to interpret the results on an interval scale. Hence, the 7-point Likert scale is appropriate for the conjoint analysis.

3.4.2.2 Estimation of Part-worth Utilities

The data gathered through the empirical study is then processed in the conjoint analysis to calculate the part-worth utilities. From those values, the metric total utility and the relative importance of the different attributes are derived.

A part-worth utility β is estimated for every level and these single part-worth utilities are then linked to determine the total utility y for a distinct stimulus. The determination is based on an additive model. Equation (3.1) defines the general formulation of the additive utility model used for conjoint analysis. It follows the idea to construct a total utility y_{jk} from the part-worth utilities β_{jm} so that the empirical observations are represented as accurately as possible.

$$y_k = \mu + \sum_{j=1}^{J} \sum_{m=1}^{M_j} \beta_{jm} \cdot x_{jmk} \tag{3.1}$$

with

$$\begin{aligned} y_k &= \text{estimated total utility for stimulus } k \\ \beta_{jm} &= \text{part-worth utility for level } m \text{ of attribute } j \\ x_{jmk} &= \begin{cases} 1 & \text{, if stimuli } k \text{ consists of attribute } m \text{ with level } k \\ 0 & \text{, otherwise} \end{cases} \end{aligned}$$

The variable μ equals the average score of the scale or, in other words, the average

utility from which the attributes positively or negatively deviate. The metric analysis of variances is based on the central assumption that the distance between the different ratings of stimuli is considered equidistant. Hence, the additive model like in equation 3.1 can be applied. In case of ordinal scaled data the monotonous analysis of variance leads to the appropriate evaluation of the dataset (Backhaus et al., 2008).

The part-worth utilities are Ordinary Least Squares (OLS) estimates being calculated by minimizing the squared distance between the empirical observation and the estimated utility as in equation 3.2. Just as well the regression analysis of p-values on the dummy variables x_{jm} in equation (3.1) can be conducted.

$$\min_{\beta} \sum_{k=1}^{K} (p_k - y_k)^2 \tag{3.2}$$

To calculate the part-worth utilities in general, the analysis of variances is used, which estimates the part-worth utilities based on the metric OLS algorithm. The OLS method is widely applied in practice (Green and Krieger, 1993). Carmone et al. (1978) and Darmon and Rouziès (1994) confirm that OLS has been widely accepted for metric scaled stimuli. Even for non metric scales OLS has been successfully applied (Wittink et al., 1994; Carmone et al., 1978). Moreover, OLS in combination with a 7 point scale leads to the most accurate results compared to a 4 point scale and slightly better than 9 point and 11 point scales (Darmon and Rouziès, 1999).

3.4.2.3 Aggregation of Part-worth Utilities

The estimation of the part-worth utilities can be executed for every individual. A generic view on the preferences comprising all individuals requires an aggregation of the utilities. There are two approaches to obtain aggregated results from a conjoint analysis. The first results from the individual conjoint analyses are normalized for every individual. These normalized part-worth utilities are then summed up. The other possibility is to conduct a conjoint analysis that delivers aggregated part-worth utilities over a population.

The idea behind the first approach is to identify the minimum between each part-worth utility and the smallest part-worth utility of its attributes. The following equation formulates this relationship:

$$\beta^*_{jm} = \beta_{jm} - \beta^{\min}_{j} \tag{3.3}$$

with

β_{jm} = part-worth utility for attribute j and level m
β^*_{jm} = minimal part-worth utility for attribute j

These so called transformed part-worth utilities are then used to calculate the normalized part-worth utilities $\widehat{\beta}_{jm}$. It is important to identify the combination of levels with the highest part-worth utilities. The sum of these part-worth utilities is the maximum value of the domain and is set to $\widehat{\beta}_{jm} = 1$ in order to normalize the part-worth utilities.

$$\widehat{\beta}_{jm} = \frac{\beta^*_{jm}}{\sum_{j=1}^{J} \max_m \left\{\beta^*_{jm}\right\}} \tag{3.4}$$

The absolute values thereby only allow to draw conclusions in terms of importance of attribute levels for the total utility of a stimulus. One cannot conclude that if one attribute has significantly higher part-worth utilities than the other, this determines a change of preferences. Furthermore, the size of the spread between the different part-worth utilities of one attribute influences the choice decision. In case of a huge spread, a change from a low level to a significantly higher level will affect the total utility and may change the choice. To measure this relative importance of each attribute, w_j has to be calculated as follows (Backhaus et al., 2008):

$$w_j = \frac{\max_m \{\beta_{jm}\} - \min_m \{\beta_{jm}\}}{\sum_{j=1}^{J} \max_m \{\beta_{jm}\} - \min_m \{\beta_{jm}\}} \tag{3.5}$$

The second approach requires the assumption that the individuals are understood as replications of the attribute design to receive aggregated results from a conjoint analysis. To use the methods presented in section 3.4.2.1 for calculation of the conjoint analysis, the index k has to be adjusted by the number of participants N.

$$k = N \cdot \prod_{j=1}^{J} M_j \tag{3.6}$$

The variable J thereby represents the number of attributes and M_j the number of levels of attribute j. k represents every individual instead of every stimulus (see equation (3.1)). Although the calculated part-worth utilities using both methods

may vary, the relative importance leads to the same results. According to Backhaus et al. (2008) in most cases researchers are primarily interested in the average utility of their customers and therefore, the aggregated approach provides them with satisfying results.

3.4.3 Choice Set for the Conjoint Analysis

The design of the choice set is crucial to the outcome of the empirical results. In particular, a survey with a high number of attributes and choices can lead to a more realistic situation a participant is confronted with. Furthermore, it can increase the quality of the answers, because more details are asked. However, this complexity of the questionnaire can result in a large amount of information. A participant faces a lots of choices and feels overstrained. Consequently, it can result in a negative impact on his answers. He will simplify his answering strategy and may not always reply truthfully (Mazzotta and Opaluch, 1995). Another possibility is that the overwhelming amount of information cannot be considered in detail by the participant. His answer may not represent his opinion, since he adopted heuristic decision rules. Hence, it may undermine the reliability of the data (DeShazo and Fermo, 2002).

Several studies scrutinized the impact of the number of attributes and alternatives on the results. DeShazo and Fermo (2002) tested four to seven attributes with two to seven alternatives and in another set nine attributes with six to nine alternatives. An increasing number of attributes will have a negative impact. Lee and Lee (2004) analyzed nine and 18 attributes and had similar results. However, they used 18 and 27 alternatives, respectively. The number of alternatives did not have a significant impact on the results. According to DeShazo and Fermo (2002), more alternatives have even a positive effect, although they analyzed a smaller set than Lee and Lee (2004). Hence, in this thesis seven attributes are considered. The set of alternatives is determined to be 18. Furthermore, the number of levels vary between two and three to avoid the number-of-levels effect (Currim et al., 1981), which induces participants to prefer attributes with higher number of levels more than attributes with lower number of levels. Hence, this configuration allows to receive reliable results.

To receive significant results, Backhaus et al. (2008) outline seven aspects, which have to be taken into consideration when defining attributes and their levels. The following three principles relate to the selection of attributes.

- **Relevance:** The attributes have to be relevant meaning that they influence customers' decision making in the buying process.

- **Interference:** The vendor has to be able to control the attributes that are analyzed within the conjoint analysis. In other words, variation of the respective attributes should be possible within the design process of the product.
- **Independence:** The utility of one attribute shall not be influenced by the characteristics of other attributes. Respondents shall not consider characteristics of different attributes as dependent on another.

Furthermore, there are four principles that need to be considered when selecting the characteristics or levels.

- **Feasibility:** The vendor has to have the knowledge to produce and deliver the investigated characteristics and bundles.
- **Compensatory relationship:** Based on the assumption that the total utility is calculated by accumulating all part-worth utilities, this means a low part-worth utility of one attribute can be counterbalanced or compensated by a high part-worth utility of another attribute.
- **Knock-out criterion:** It is of great importance that no characteristic is seen as a knock-out criterion which would contradict the compensative nature of the characteristics.
- **Limit:** Since the complexity of the survey grows exponentially with the number of characteristics, it is advisable to limit the number of attributes and levels.

Table 3.3 summarizes the attributes and respective characteristics being selected for the conjoint analysis. They are selected by studying the service design of successful vendors like Amazon Web Services and interviewing experts from Sun's Asia Pacific Science & Technology Center[12]. Subsequently, a definition for each of the attributes is provided:

- **Price:** For the following sample products there are three different prices ($0.70, $1.10 and $2.00). The prices are on an hourly basis and include CPU, memory, data transfer, storage and the subsequently presented characteristics. The prices stem from various providers like Amazon Web Services, 3Tera or Flexiscale offering services with different kind of characteristics.
- **Performance:** The survey distinguishes between guaranteed-performance and best-effort service types. Guaranteed performance means a guaranteed high service level and thereby high performance. A best-effort service has a lower priority and jobs may have to wait before being executed. This attribute was

[12]SunŠs Asia Pacific Science & Technology Center (`http://apstc.sun.com.sg/`)

Table 3.3: Attributes and their levels

Attributes	Levels	Attributes	Levels
Price	$2.00	Operating System	Both
	$1.10		Windows
	$0.70		Linux
Performance	Guaranteed	Availability	99.95%
	Best effort		99.75%
			99.50%
Support Level	Phone	Value-added	Load Balancing
	Email		Firewall
	Documentation		None
Start-up Time	Instant		
	Prolongated		

motivated by the work of Maglaras and Zeevi (2005) for Revenue Management. Furthermore, so far most providers did not specify SLAs, but only offered best effort services.

- **Support Level:** Products include different levels of support (phone, email and documentation). Documentation is provided on the Website if products include documentation only. It has an impact on the cost for the provider. A phone supports requires a call center infrastructure. For example, 37signals[13] has only email, blog and documentation support for his web-based collaboration software unlike AppNexus[14], who offer a 24/7 phone support.

- **Start-up Time:** Start-up Time defines the period between 'booking'/registering and set-up of an instance. This survey distinguishes between instant start-up, which means within minutes the instance is ready for calculations, and prolongated start-up. In the latter case, there is a significant time delay between 'booking' and set-up. Companies get the flexibility to delay the virtual machine start up in order to favor high class customers, when resources are scarce.

- **Operating System:** Within this survey, it is distinguished between instances provided with Windows or Linux only and an environment where a choice between both operating systems during set-up and operation is possible. In

[13] 37signals (http://www.37signals.com)
[14] AppNexus (http://www.appnexus.com)

the beginning Amazon's EC2 service had only Linux instances in its service portfolio. Since October 2008, a Microsoft Windows Server is available. However, a strict requirement can be the request for both operating systems, since some software development teams need both for testing and compatibility purposes.

- **Availability:** There are three different availability-levels: a) 99.95% equals less than 4.5 hours downtime per year b) 99.75% equals less than 22 hours downtime per year c) 99.50% equals less than 44 hours downtime per year. Outages of provider sites are a well known problem (see Section 2.3.1). Flexiscale is among the first companies giving their customers a 100% available IaaS. The EC2 service from Amazon promises at least 99.95% with a limited 10% refund of the Service bill, if the availability drops below 99.95%. The reason for limiting the refund is determined by their outage experience of EC2 and S3 discussed in Section 2.3.1.

- **Value-added:** Value-added services can be a Firewall, Load Balancing or none. The motivation stems from the service offer of 3Tera, where virtual machines can be preconfigured with Firewalls or specification of Firewalls can be suggested. FlexiScale[15] allows to add a firewall for £0.10 extra per hour. Load Balancing allows workload-intensive applications to shift jobs from one machine to another automatically or by predefined rules. Hence, resources can be utilized more efficiently, when consumers have multi-CPU applications. Linode and AppNexus offer Load Balancing over several virtual instances.

Except for price and availability, all attributes are regarded as discrete. Price and availability are considered as linear. Lower price and higher availability are valued as the more favorable options, respectively. The participant will receive the highest benefit of each attribute, if the service is for free or if 100% availability can be guaranteed. For example, Flexiscale offers a 100% SLA of their IaaS and until June 2009 Mor.ph[16] allowed to create a free account and use their application server by uploading Java Web applications.

The previous section emphasized the advantages of the full-profile method over the trade-off method. As a consequence, the number of stimuli for the attributes presented above would be $(3 \times 2 \times 3 \times 2 \times 3 \times 3 \times 3) = 972$ for the full-profile method. Hence, the necessity arises to reduce the complete set of stimuli to a reduced design. This reduced design represents the complete design but decreases the number of stimuli to allow a focused expenditure of time. A methodology to create an asymmetrically reduced design[17] was proposed by Addelman (1962) and Street

[15] FlexiScale (`http://www.flexiscale.com`)
[16] Mor.ph (`http://www.mor.ph`)
[17] it is asymmetric, because the number of levels are not equal for every attribute

et al. (2005). 22 profile cards were derived including four holdout cards, which are used to check the robustness of the individual utility functions. This is in line with Green and Srinivasan (1978) who outline that a sample should consider not more than 20–30 stimuli. A complete list of the profile cards can be found in the appendix A.2.

3.5 Results & Implications

The results from the survey were analyzed in two different ways. The following section covers the descriptive results derived directly from the data. In Section 3.5.2 the data is analyzed via statistical methods to corroborate or discard the hypotheses from Section 3.3.2.

3.5.1 Descriptive Results

Figure 3.2 illustrates the composition of the respondents who took part in the survey. The composition is split into industry distribution, global distribution and age. As these questions were not obligatory, the 'no answer' option is included. The topic of Cloud Computing still is technology driven with focus on developing standards and not on addressing Chief Information Officers' needs. Other surveys have already analyzed opinions from Chief Information Officers regarding a strategic view on Cloud Computing (see Section 3.2.1). However, the high share of IT workers (55.4%) and researchers (21.5%) in the population validate to conclude that the results drawn in the following are based on professionals' opinions and valuations. The accuracy of the answers is further emphasized by the age distribution with the greatest share of the population being between the age of 26 and 45 (66.2%) and another 21.5% being older than 45. Another important aspect to mention is the global distribution of the population with 38.5% working in the Americas, 30.8% in Europe, 13.8% in Asia and in Australia as well as 3.1% from the Middle East. For classification reasons, the user type is of importance. While developers (38.5%) form the biggest group of participants, corporate (enterprise and SME) users with 35.4% are almost equally present. Whereas scientific and end users are underrepresented with 15.4% and 10.8% respectively.

The participants' predominant areas of application[18] are application hosting (64.6%), Web hosting (38.5%), and high performance computing (33.8%). As Figure 3.2 shows most of the users work with IaaS providers (55.4%) while 29.2% predominantly use PaaS offers. The small share of participants who primarily works with

[18] at most three answers were possible

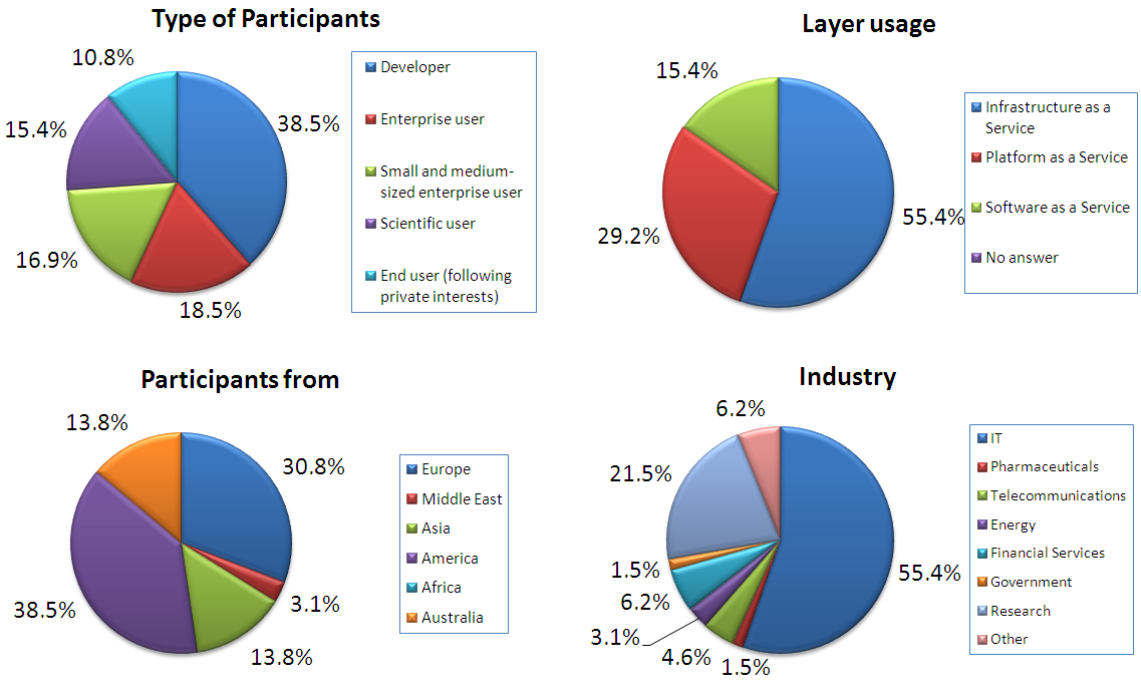


Figure 3.2: Characteristics of the participants in the survey

SaaS solutions (15.4%) again indicates the high skill level of the survey population and thereby the robustness of the results.

In the following, the results from the questions considering choice and change behavior are evaluated. A topic of great interest is the change behavior of Cloud Computing customers. Three questions address this topic. Question 7 asks if customers compare offers from different providers before registering for a service. The next question investigates if customers have already changed their provider and the reason for changing, if the question was answered in the affirmative. Question 9 explicitly asks for the price reduction that would motivate a customer to change his provider if everything else remains the same. Although at least 72% of the respondents[19] state that they compare different offers before selecting a Cloud service provider, the actual change behavior shows different results. Out of all 65 respondents 37 (56.9%) have not changed their provider yet. Especially end users (90.0%) and scientific users (71.4%) display a low change rate. When comparing the usage frequency with the change behavior, it shows that only those who use Cloud services daily actively switch providers (64.3%). Those who only use Cloud services rarely, meaning less than monthly, stick with the same provider (85.7%). The reasons to change the provider are in the first instance better prices, better service, and

[19] At least 72% compare Content Delivery Network (CDN) offerings and at most 88% compare storage solution offerings.

better performance. Users who have not changed their service provider state three major concerns: missing compatibility/SLAs and as a result the complexity of migration, missing trust in other providers, and migration costs.

The results from Question 9 show that the majority of respondents is only willing to change the provider when the price is at least 25% lower than the price of their current provider (53.8%). Only 23.0% of the users are willing to change their provider when others offer a price reduction of up to 10%, and 15.4% of the respondents would not change their provider for a better price. Interestingly, the groups of respondents who would only change their provider for a significant reduction of 25% to 50% have a median price, considered appropriate, of US-$0.25 and US-$0.15 for one standard instance per hour. In contrast, the groups of respondents who would change their provider for a price reduction of at least 10% or less have median prices of US-$0.70 (at least 10% reduction) and US-$0.50 (in any case), respectively. The group of respondents that would not change the provider for a better price has a median price of US-$1. Only 21.5% of the participants, however, have changed the provider since using Cloud services. More than 50% have never gained experience with another provider (see Figure 3.3).

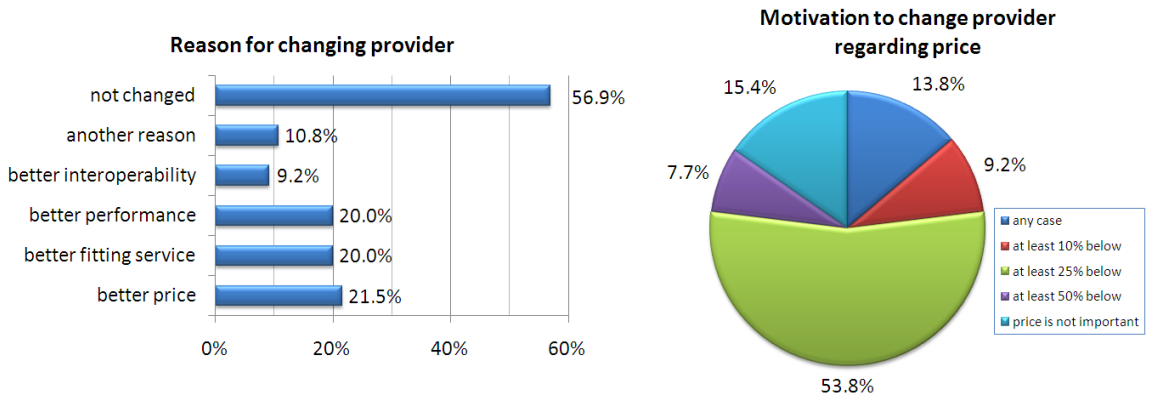
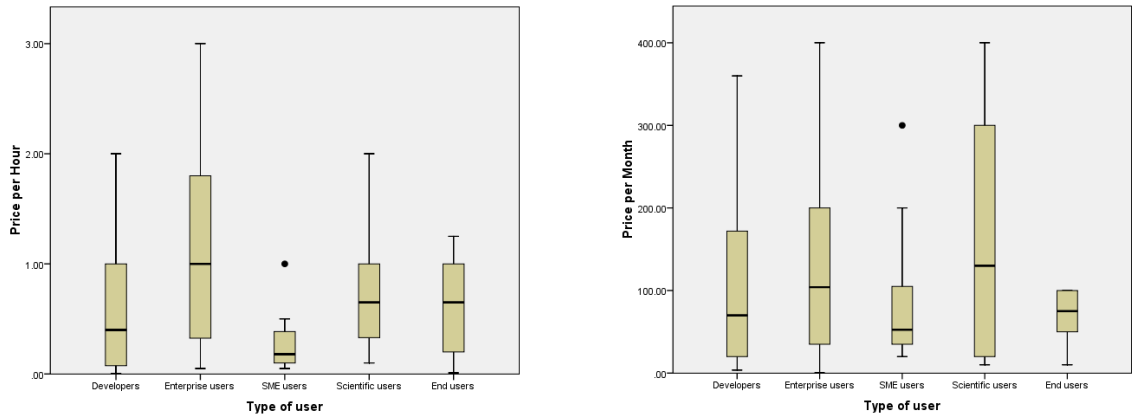


Figure 3.3: Reasons for changing current provider

As expected, the appropriate price for one standard instance per hour differs between the different user groups. Figure 3.4a shows that median value for SME users and developers is the lowest within the population with values of US-$0.18 and US-$0.45. Especially for SME users the variance of 0.081 indicates that the value is accurate. By eliminating the outliers, all values above US-$5 or at least four times the average value, the results for the group of developers become more robust and the median value is US-$0.40. In contrast, the willingness to pay for enterprise and end users is considerably higher. Their median values are US-$1 and US-$0.65. The currently used pricing policy of metered pricing for every part of the service finds great acceptance within the population. More than three quarters (76.9%) prefer to customize their services instead of buying bundles that have a fixed configuration

of resources and services. Comparing the estimated hour price to the monthly price, 78.4% of the participants value the hour price higher compared to the monthly fee assuming that they would use the service 24 hours a day for 30 days. Hence, either they expect a discount on the monthly fee or they would not use it 24 hours a day.

Expectations of future development of prices and quality of service are derived from Question 10. The overwhelming majority of 84.6% expects prices to decrease and meanwhile an increase in quality of service. 11.3% anticipate an increase both in price and quality of service. Only two participants (3.2%) expect a decrease both in prices and quality of service.



(a) Expected price for one hour

(b) Expected price for one month

Figure 3.4: Expected price clustered by each user group for hourly and monthly prices

Apart from all pricing issues, Question 11 focuses on other aspects that influence customers' valuation. Respondents are asked to rate the introduction of a system utilization monitor, differential pricing, a subscription model, a loyalty program, and complementary offerings on a 5-point Likert scale ranging from 'strongly agree' to 'strongly disagree'. The system utilization monitor shows the utilization of the resources running a job or application and could display a light in green for low utilization, in yellow for moderate utilization, and in red for high utilization. The respondents rate the introduction of such a function very positively as 90.8% either agree or strongly agree. 54 participants confirm the introduction of differential or tiered pricing would be beneficial. Differential pricing in this case denotes to different pricing in peak and off-peak periods. The introduction of a subscription model implies the booking of a certain number of server instances permanently at a lower price, and additional instances later depending on workload. The responses show an almost equally positive rating as compared to differential pricing with 52 participants agree or strongly agree on the introduction and the median value is 2 ("agree"). The introduction of a loyalty program like those offered by airlines giving redemptions in terms of higher performance or discounts attracts less acceptance, although still positive. 40 respondents support the introduction, 20 nei-

ther agree nor disagree on the introduction and 5 even disagree. The introduction of complementary offers, referring to promotions offered to complement used services is positively regarded, though receiving the least acceptance. 31 respondents who agree or strongly agree to the introduction face 11 respondents who disagree or strongly disagree to the introduction.

The idea of revenue management is to offer discounts when customers are willing to take restrictions. Based on that assumption, Question 4 checks for the ability to predict usage behavior and Question 6 asks for the willingness of customers to book a time slot in order to receive price reductions. 52.3% of the respondents are able to predict their usage behavior, 12.3% of the respondents are able to predict their behavior one week in advance and at least 23.1% are able to estimate their resource requirements two days ahead. The group of those that is willing to receive discounts and meanwhile can predict its usage behavior consists of 40.0% of the population. 12.3% of the population can predict their behavior but has no interest in receiving discounts. The group of users that is willing to receive discounts consists of 39 (78.5%) respondents. The possibility to receive discounts for a best effort service, meaning jobs or applications of other privileged users get higher priority, is favored by 56.9%.

Further interesting results are derived from the comments sent in by some of the respondents. They give insight into their own definitions of Cloud Computing calling it rather a mash-up that "involves leveraging services that you could not provide yourself relating to a specific widely needed resource", like Amazon or Google Maps. Meanwhile, one respondent is skeptical about generalized computing services because of three reasons. First, packaged software is always highly customizable, like SAP and solutions that are useful for different businesses, would therefore need to have numerous complex interfaces. The second obstacle that is mentioned refers to performance and reliability with Cloud services involving higher latency compared to in-house solutions. A third point is the fact that differentiating business objects and developing innovations is very difficult using commodity application software. The respondent concludes by predicting a spread of specific services like Google Maps but no success for generalized Cloud services.

In contrast, another respondent outlines the benefits of Google's App Engine that "have abstracted the hardware and system administration" leaving no need for a system administrator so that the company can focus on developing software. Being an SME company they could not afford to employ staff with the high level of expertise on system administration that Google has and compared to Amazon Web Services the basic version of the App Engine is for free, although they would be willing to pay more because of the benefits they receive.

3.5.2 Inductive Results

The goal of the conjoint analysis is to identify the most important level of each factor and the most important factor (*RQ* 2.3). The distribution for the aggregated "relative importance" levels of the seven factors outlines what determines the decision process. Operating system clearly ranks first as the most important factor followed by price (see table 3.4). With values of 24.3% and 18.9% these two factors have significantly higher influence on the decision making process than support which is ranked third with 14.6% followed by value added services with 13.0%. Performance (11.1%), availability (9.5%) and start-up time (8.8%) have less influence on the participants' choice. The correlations between the observed and estimated preferences indicate a good estimation model. Pearson's R which estimates the correlation between the metric total utility and the empirical observation, is significantly high with $Pearson's\ R = 0.96$.

Apart from the relative importance, the part-worth utilities of the different factors show interesting results (see table 3.4). The optimal product offers both operating systems, at the lowest price, with phone support, no value added services, a guaranteed performance level at the highest availability rate, and is ready to use immediately after start-up. The total utility U^* for the optimal Cloud service is calculated as follows[20] (see equation (3.1)):

$$U^* = 5.03 + 0.53 - 0.57 + 0.25 + 0.57 + 0.29 - 0.22 + 0.16 = 6.04^{21}$$

The results for the respective factor levels explain customers' preferences in detail. For the factor "operating system", users prefer to have a system running on Linux, if not both operating systems are offered, while a Windows only instance has a negative utility. This valuation changes when the population only consists of enterprise users. They refuse Linux only based instances even stronger than those running on Windows only. Thereby higher prices are considered worse than lower prices. Unlike in other cases prices are not regarded as indicators for the quality of the product. Results for the factor support level indicate that all users value phone support as the most important support level. Email support also has a positive utility and documentation only receives a negative utility. While value added services receive a negative utility from the total population, the group of scientists differs from the average user because load balancing has a strong positive utility for them.

[20] Note that this approach considers empirical value simultaneously (also known as combined conjoint analysis). Hence, the information loss is lower on an aggregated level, but the path-worth utility can significantly differ from aggregated values of every individuals. However, the relative importance remains almost the same (Backhaus et al., 2008).

[21] Please note that due to rounding errors the total value does not exactly deviate by one from the average utility.

Table 3.4: Utility estimates and standard error rates for each level

Factors	Relative importance	Levels	Utility Estimates	Std. Error
Operating System	24.30%	Both	0.53	0.13
		Windows	-0.63	0.13
		Linux	0.10	0.13
Price	18.90%	$ 0.70	-0.57	0.12
		$ 1.10	-0.90	0.18
		$ 2.00	-1.63	0.34
Support Level	14.30%	Phone	0.25	0.13
		Email	0.09	0.13
		Documentation	-0.34	0.13
Value-added Service	13.00%	Firewall	-0.60	0.13
		Load Balancing	0.03	0.13
		None	0.57	0.13
Performance	11.10%	Guaranteed	0.29	0.10
		Best effort	-0.29	0.10
Availability	9.50%	99.95%	-0.22	0.11
		99.75%	-0.45	0.22
		99.50%	-0.67	0.36
Start-up time	8.80%	Instant	0.16	0.10
		Prolongated	-0.16	0.10
Constant		μ	5.03	0.32

Developers also value load balancing positively although they prefer not to have any value added services. As expected, the population of respondents values a higher availability more than a lower availability.

The applicability of Revenue Management models in the Cloud from the customers' perspective was summarized in *RQ 2.1*, if Revenue Management models are applicable for Clouds. The questions from the survey (except the conjoint analysis) were mostly nominally scaled. Thus, the three hypotheses were tested via a chi-square test[22]. The goal of Hypothesis 3.1 is to find out whether the usage frequency depends on the role of the user. Revenue Management models cluster users into different groups in order to provide a certain price for a certain group of users. According to the chi-square test, business users and developers use these

[22]For more information on Pearson's chi-square test see e.g. Cowan (1998)

services more often than scientific or end users (Pearson's chi-square test, p-value = 0.004<0.01). This allows a provider to offer high price and low price segments of services and differentiate them by determining better conditions for more frequent consumers. However, it cannot be concluded whether more frequent consumers are able to predict their requirements better or not (Pearson's chi-square test, p-value = 0.746).

Table 3.5: Hypotheses results

H1	The usage frequency of computing services depends on the role of the user.	✓
H2	Users, who are able to predict their behavior, also book the resources in advance.	✓
H3	Users, who support price discrimination, also prefer best effort services.	✓
H4	Users, who support price discrimination, also prefer to book in advance.	X
H5	Users, who support price discrimination, also prefer a subscription model.	X

Another positive result was derived for the second Hypothesis 3.2. Consumers, who are able to predict their service usage also accept booking in advance. Otherwise, advance reservation will not make any sense, if consumers do not accept it, although they were able to do so. The test shows that there is a relationship between both questions (Question 4 and Question 6) according to Pearson's chi-square test with a p-value=0.034<0.05. Hence, the null hypothesis (independence of both answers) can be rejected.

Furthermore, price discrimination is essential to price services according to consumers' needs (Hypothesis 3.3). In particular, if scientific users would prefer low bandwidth for a lower price, then low fare classes and high fare classes increase the revenue of the provider. The null hypothesis states that price discrimination and different levels of service quality are dependent, i.e. consumers, who prefer price discrimination seem to prefer different service levels as well. The null hypothesis is rejected. Hence, there is a relationship between these two parameters (Pearson's chi-square test, p-value = 0.016<0.05).

Besides the above mentioned context, price discrimination can be tested according to booking in advance (Hypothesis 3.4). A user who prefers price discrimination will also take the chance to reserve instances beforehand. This hypothesis cannot be confirmed (p=0.60>0.05). However, the participants seem to significantly favor a subscription model with advance reservation (p=0.047<0.05). Hence, a subscription model combined with a prolongated start-up time or comprising a condition to receive the instance with delay for a lower price would be beneficial to the provider, since he can sell the on-demand services for a higher price.

Inferentially, would a supporter of price discrimination also choose a subscription model in order to receive a lower price (Hypothesis 3.5)? The chi-square test does not identify a significant relationship between subscription model and price discrimination (p=0.21>0.05). Therefore, no statement can be derived from this result. In the context of *RQ* 2.2, only price discrimination and different service levels have a strong dependency.

The classification of consumers into certain groups to apply price discrimination is an important aspect in Revenue Management. Su (2007) classified the consumers into four different categories (see Section 3.2.3). An impatient customer may not want to wait too long and has a higher valuation for a service than a patient customer, who chooses to wait longer and pay a lower price. Thus, a clear preference classification will ensure the success of Revenue Management. It is assumed that impatient high-type customers will have a higher willingness to pay and cannot predict their behavior, whereas patient low-type customers will request a lower price and can predict his usage. High class customers are determined by their estimated value for one computing instance for one hour. They are selected from all participants with a valuation above the third quartile ($\geq$ \$1.00). Table 3.6 illustrates the distribution over the classes. Interestingly, one third of the participants need resources often instantly and are not ready to pay a high price (low class, impatient customers). 58% of the customers are patient high-type or impatient low-type. When these two customer groups dominate the market, Su (2007) recommends to increase the price over time, which is in line with the common practice in Revenue Management. Additionally, this table outlines the heterogeneity of the customer behavior. Intertemporal price discrimination are only applicable for a heterogenic customer group (Stokey, 1979).

Table 3.6: Crosstabulation based on participants' valuation for a one hour instance and their usage predictability

	High class	Low class	Total
Patient	16	18	34
Impatient	9	22	31
Total	25	40	65

3.5.3 Implication

Summarizing the results, three major findings can be listed. At first, the conjoint analysis revealed that price is an important factor, but it is dominated by the Operating System offered by the provider (RQ 2.3). The IaaS layer is based on the chosen

Operating System and the current software running on top of them do not seem to be flexible enough. Users of IaaS seem to prefer a well-known platform instead of comparing the availability rate of different providers or between the services of one provider. Contradicting the results from previous surveys, performance and availability are not the most important factors of an IaaS. Service sellers should consider designing Operation System, the service price and the support level crucially by offering different kind of factor levels for the heterogenic customer classes. The heterogeneity allows application of price discrimination for Cloud services.

Secondly, consumers are open towards price discrimination and more than 50% would book time slots in advance or accept best effort services for discounts (RQ 2.2). Revenue Management models can be applied to the Cloud from a consumer perspective. Important characteristics of Revenue Management models like price discrimination and advance reservation will be accepted in line with other factors like service level differentiation. However, price discrimination is not supported in conjunction with all Revenue Management aspects (Hypothesis 3.3, 3.4 and 3.5). This characteristic also underlines the heterogenic customer classes.

Last but not least, comments from the respondents also indicate what other surveys point out: changing providers is very complex, and missing standards make it harder to transfer operations from one provider to the other (Gens, 2008). Even though prices influence the first choice decision, migrating operations to another provider for a better price is only an option for non-critical operations and for significantly (at least 25%) lower prices.

Providers of Cloud services should crucially analyze the market situation. Although consumers seem to accept price discrimination under certain condition, McAfee (2008) states that three conditions have to be fulfilled to successfully apply price discrimination. Besides a heterogeneous consumer behavior, Cloud service provider needs market power and he has to prevent or limit arbitrage options. While the last two arguments depend on the market situation or on the service design of the provider respectively, the survey results confirm the heterogeneity of the consumer behavior. All these factors are important to successfully apply Revenue Management models to the Cloud (RQ 2.1). Furthermore, operating system and price have a great impact on the provider selection process of the consumer. Thus, offering various Linux or Windows versions (or even other operating systems) can result in a higher market share. However, a provider should consider other additional services like phone support as well to distinguish his service from the competitors' offers.

Chapter 4

Capacity Management in Clouds

The problem of seat inventory control is complicated by the high uncertainty in the demands and the diverse arrival patterns of the requests for the various fare classes.

[Lee and Hersh, 1993]

The goal of this chapter is to define and evaluate sa heuristic called Customized Bid-Price Policy (CBPP) to calculate bid prices in the Revenue Management context for Cloud services. Section 4.2 outlines the existing approaches for the bid price calculation. In Section 4.3 an example for applying bid price models to Cloud services is given and the drawbacks of existing approaches are discussed. The heuristic is evaluated via simulation. Simulation configuration and the hypotheses are described in Section 4.4. The results are summarized in Section 4.5.

4.1 Introduction

In the Cloud Computing market, Cloud service providers face dynamic and unpredictable consumer behavior. The methodology, through which prices are set in a dynamic environment by providers, can influence the demand behavior of price sensitive consumers (Bitran and Caldentey, 2003). Consequently, consumers with a low valuation for a service would use it during cheaper periods. Business consumers are willing to pay a higher amount for its usage (see Chapter 3). By identifying the right price for a customer and a requested service at a certain point in time, providers can achieve higher revenues (Kimes, 1989). However, in some settings, it is difficult to change prices over time or the service is designed to be offered for a fixed price. For example, Amazon currently offers its Elastic Compute Cloud[1] service at a fixed price of $0.085 for a CPU hour without frequently changing the prices. Price changes can

[1] A small Linux instance (`http://aws.amazon.com/ec2`)

be realized with specific markets like auctions. Historically, Amazon started with their on-demand fixed price model in 2006. In March 2009, they announced the "reserved instances" payment model. It comprises of a one-time fee valid for a certain time period and allows access to instances for a much lower price per hour than the on-demand model. This model is beneficial to the consumer, who uses the service quite frequently. In December 2009, Amazon presented the "spot instance" model, where prices change dynamically over time. If the spot price (i.e. the current market price) decreases below the consumer's bid, the instance of the consumer will be started and the consumer pays the current spot price. If the price increases above his bid, the running instance will be stopped until the price decreases again or users cancel their bids. However, it is not clear how the spot price is determined by Amazon. One option would be to apply Revenue Management methods to support the identification of appropriate prices for Cloud services. Generally, Amazon aims to satisfy different kinds of demands of the heterogenic consumer preferences by offering three different pricing mechanisms.

Another important aspect in Revenue Management is the comprehensive view of prices in conjunction with advance reservation (see Section 2.3). Advance reservation is already applied to Cloud services. For example, RenderRocket[2] offers on-demand access to rendering software to create animated movies or to render architectural and industrial designs. They charge users "on-demand hourly" with rates starting from $1.50 per server hour. But they also offer advance reservation of servers on a daily, weekly or monthly basis. This offer allows a guaranteed priority on reserved instances and an on-demand scalable system.

Although advance reservation and the analysis of consumer behavior can reduce uncertainty in demand, there are still unpredictable occurrences. An example of unpredictable service requests is the problem Animoto[3] faced in spring 2008. Animoto offers to create customized web-based videos automatically by uploading images and music. It needs substantial computing power for video processing. In spring 2008, Animoto had an unpredictable demand for their service, when 750,000 people signed up for it within three days. However, the existing infrastructure was not able to manage it. The instant availability of Amazon's EC2 allowed them to add up to an additional 3,500 virtual instances to satisfy the spike in demand[4].

Furthermore, Cloud services are characterized by various properties defined in SLAs, which enable the implementation of price discrimination strategies for distinguishing similar services based on their SLA attributes (e.g. higher throughput or

[2] http://www.renderrocket.com

[3] http://www.animoto.com

[4] NY Times article: http://www.nytimes.com/2008/05/25/technology/25proto.html?_r=3&adxnnl=1&oref=slogin&ref=business&pagewanted=print&adxnnlx=1212768774-L4fMNfgaHc/0lDK5wKcevQ, last accessed on 03.03.2010

lower availability rate). From a provider's point of view, offering various kinds of services based on advance reservation and on-demand instances can lead to uncertain demand requests. When demand outweighs supply, a provider has to decide, whether to accept an incoming request or reject it in favor of a request arriving later for a service with higher revenue. In the case of Amazon's Spot instances the price can be calculated by analyzing several scenarios with different price settings and virtual resource limitation in order to determine the appropriate pricing strategy.

In this chapter, a decision concept for a provider is presented to accept or reject incoming requests for services in order to increase revenue in a scarce resource market. A provider offers several Cloud services, which use the same resources from the provider's resource pool. The goal for a provider is to sell the most expensive services to the paying customer(Phillips, 2005). When a consumer requests a service with low revenue, the provider has the possibility to accept this request or to wait for a prospective customer asking for the high valued services. Different decision rules well known from Revenue Management for the Airline Industry are analyzed to understand how to apply Revenue Management concepts to Cloud Computing. This contribution comprises of a more efficient decision rule called *Customized Bid-Price Policy (CBPP)*, which is based on the bid-price control concept introduced briefly in Section 2.2.1 and 2.2.2. Its efficiency is analyzed via simulation-based optimization in Section 4.5.

4.2 Related Work

Each offered service represents a booking class, which has a fixed price. The provider has to decide, if a service request should be accepted or rejected. Thus, a limit defining how many requests are operable for each booking class has to be identified, which is known as capacity control (see Section 2.2.1 and 2.2.2). Nested booking limits allows the prevention of bookings for services with higher revenue being rejected in favor of bookings with lower revenue. They define how much capacity is reserved for a certain booking class. Every service has limited access to resources like CPU, memory, storage, or bandwidth. Due to multiple resources a nested booking limit control must be defined for each resource. This is called virtual nesting control (Williamson, 1992; Smith and Penn, 1988). It is difficult to forecast demand appropriately for virtual classes. The requirement of mapping services to virtual classes also increases complexity (Talluri and van Ryzin, 2004b). Furthermore, the assumption that demand for low-class services occurs earlier than for high-class services is common in Revenue Management (Gallego and van Ryzin, 1994). If demand arrives in a strictly high-to-low order the providers simply need to accept consumer requests in a 'first-come first-served' order to maximize their revenue. On the other

hand, when demand is stochastic, the strict low-to-high order is also less appropriate. For a more realistic scenario, the assumption must be made that the demand for low-class services is more likely to arrive earlier and the demand for high-class service is more likely to arrive later in time (Kimms and Mueller-Bungart, 2007).

Bid prices are interpreted as an approximation of the opportunity cost of reducing the resource capacities, which are needed to satisfy incoming service requests (Bertsimas and Popescu, 2003). Möller et al. (2008) describe bid prices as monetary values of a single capacity unit for a resource. The sum of the resource demands of a request weighted with the corresponding resource bid prices defines the bid price of a service. If this sum exceeds the revenue yielded by the sale of one unit of the respective service, the request is rejected, otherwise it is accepted (Williamson, 1992). Regular updates of bid price values are necessary to guarantee a continuous precision of the bid prices. Less accurate bid prices can lead to accept/reject decisions of minor value. Continuously updated bid prices are based on the current booking situation at a certain point in time t. That is, if a large amount of capacity has already been sold, the bid prices turn out to be higher.
Bichler and Setzer (2007) propose an admission control for media on-demand services, e.g. a media streaming service. The authors compare an adaptive admission control based on a Deterministic Linear Programming (DLP) model (Williamson, 1992) with static admission rules, and point out the benefits of the adaptive method. Their results show, that the adaptive DLP control rejects early service requests, and thereby is able to accept high-revenue service requests arriving later.
Bid prices vary depending on the current state of the system. The availability of revenue information enables a bid price control to accept more requests which yield higher revenues. This is a major advantage over class-based controls as protection levels, because these controls either accept a class of requests or reject it (Talluri and van Ryzin, 2004b) . If a class is under-utilized, these capacities cannot be automatically allocated to other classes with high demand unless special policies are defined.

4.3 Optimization Approach

The decision of accepting or denying a request depends on the applied policy or heuristics. Capacity control comprise heuristic approaches for the original dynamic programming problem. A Bellman equation defines the optimal policy for accepting or rejecting requests. Since the speed of computation matters (especially for large resource/product settings), bid-price control is an approximation method to quickly update the policies after the arrival of new requests. It provides a good estimate, but not always an optimal solution. Especially in the Network Capacity Control (NCC)

setting, the calculation of the optimum increases exponentially with the number of resources m and services n (Talluri and van Ryzin, 2004b).

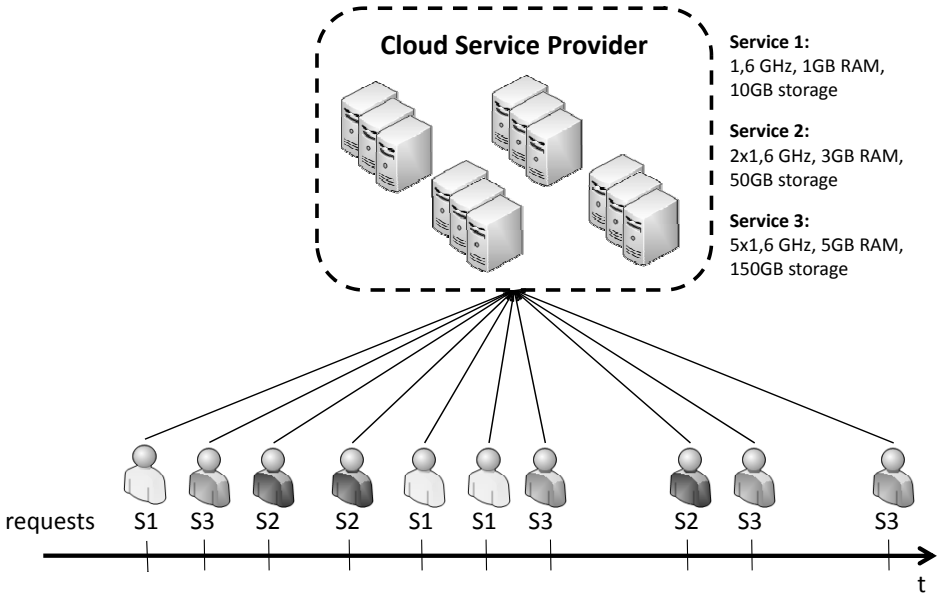


Figure 4.1: Incoming requests for different services in different timeslots

Cloud services in the Revenue Management context depend on various parameters. A service provider offers different kinds of services. Each service is based on physical resources to generate the services. These services are requested at different points in time. The decision when to accept or reject a request may change during the time horizon for the same service, since the calculated bid price for every resource depends on the current utilization of the resource. To model the request for Cloud services, probabilistic models such as Poisson processes are used to estimate the demand. The parameters, their interrelation and the induced assumptions are described below:

Time: A customer has the possibility to book a service within a booking period T. T corresponds to the total amount of time remaining to book services and is finite and countable. Furthermore, time is discrete with a point in time $t \in \{T, T-1, \ldots, 1\}$ (Adelman, 2007). Requests for services arrive at discrete points in time t. Customers book services in advance according to their computational needs (Figure 4.1). In practice, the booking period in the Cloud service domain is considerably smaller than in airline Revenue Management. This contributes to a more spontaneous setting with customers booking services according to the more agile business environment and changing requirements for Cloud services. The duration of a booked service is fixed. For example, the duration can be set to one hour (e.g. Amazon EC2), and the customers have to plan how long they will need the service by reserv-

ing multiple units of these service units. The availability of services depends on the resources possessed by the provider.

Resources: The provider has to manage and control m different resources $h \in \{1,\dots,m\}$. In the present context, the term resources stands for computing capacities. Examples for resources are CPU power, memory, storage or bandwidth. Every resource needs to be quantifiable, and must be dividable into discrete segments. For example, the allocation could be done as follows: One unit of resource $h = 1$ represents one (virtual) computing unit, one unit of resource $h = 2$ consists of 500 MB of memory, one unit of resource $h = 3$ corresponds to 1 GB of storage space, and one unit of resource $h = 4$ conforms to 1 GB data traffic in both directions. Every resource h is restricted by a finite amount of capacity c_h. The amount of capacity already reserved at time t for previous customer requests is denoted by $\bar{c}_{ht}$.

Services: Resources are necessary to provide Cloud services consuming these resources. Resources are allocated to n different classes with each service $i \in \{1,\dots,n\}$. Each service class represents a (virtual) computing environment that can be used by the customer for computational purposes. The definition of the services depends on the defined setting $m \times n$ (resources $\times$ services), e.g. setting 3×3 considers three resources and three services. Classes can be distinguished by their resource usage or by the different prices. For example, a service class consuming the highest amount of storage can be denoted as a "high storage" service. The service, which consumes the highest amount of the most valuable resource, is likely to cost more than a low-fare service with less valuable resources. However, the design of the classes highly depends on the strategic goal of the provider and on the consumer preferences for the offered services.

Formally, the matrix $\mathbf{A}$ describes the mapping between the resources and services. An element a_{hi} represents the usage of resource h by one unit of service i. $\mathbf{A}_i$ shows all resources consumed by service i and $\mathbf{A}^h$ the services using resource h. The resource consumptions of the different services are expressed by matrix $\mathbf{A}_i$. The price of service i is denoted with r_i, and depends on the amount of resources required by the service. r_i is the revenue yielded by the sale of one unit of service i. Prices are fixed over a predefined booking period.

Demand: Demand can be modeled in a variety of ways (McGill and van Ryzin, 1999). For instance, Bitran and Mondschein (1997) model demand as a time-dependent Poisson process with arrival rate λ_t. The demand between the different classes of services is assumed to be independent, which is a common assumption in Revenue Management (Belobaba, 1989; Williamson, 1992; Gallego and van Ryzin,

1994; Bertsimas and Popescu, 2003). A consumer, who desires a cheap service, will not book a higher class and, hence, a more expensive service. However, a consumer with a high valuation of the service may prefer a low-fare service. Currently, most Revenue Management literature assume independency between fare classes. Consequently, users will not switch between different fare classes. Furthermore, the arrival of group requests is not taken into account. It is assumed that at most one service request can arrive per discrete unit of time t (Talluri and van Ryzin, 1998).

Another question of Revenue Management in practice is how to perform an appropriate forecast of demand. Forecasts are a complicated statistical concept addressing the uncertainty of possible future outcomes. They aim to predict future demand, or to give an estimate of the probability distribution of demand. Forecasting is usually based on historical sales data (Chen and Kachani, 2007). However, it is assumed that providers have certain demand information from past booking periods. Therefore, they are able to perform almost accurate demand forecast. Fluctuations in demand will be handled by passing through different demand scenarios in the simulation. Some models for capacity control in Revenue Management consider the fact that some customers do not show up or cancel their bookings. Depending on the company and the fare classes booked by the customers, in some cases a refunding - in part or even in full - is possible. Concepts, which incorporate customer no-shows, refunding, as well as overbooking are not considered in the bid-price model for Cloud services. Furthermore, demand in this context is not influenced by the supply of other provider companies. Competition between providers is not examined. Aspects concerning strategic customer behavior, such as willingness to pay vs. willingness to wait, are not examined (Su, 2007). The focus of this thesis is on a provider, who is exposed to incoming customer requests for the offered services in a market with a scarce amount of resources.

Arrival process: D_{iT} represents the amount of requests for service i arriving in the complete booking period T. D_{iT} can be subdivided into the expected demand arriving between the current point in time t and the end of the booking period, denoted as D_{it} (demand-to-come), and the demand prior to the beginning of the booking period until t as $\hat{D}_{it}$(see Figure 4.2).

A request for service i at time t arrives with probability p_{it}, and thus, the arrival of a request for service i at time t is a random variable X_t with $X_t = (\{0,i\} \,|i \in \{1,\ldots,n\})$, i.e. $X_t = 0$ if no request comes in at t. T is finite and countable, and thus, the arrival process of requests by the customers is a time-discrete stochastic process X, which is a sequence of random variables X_t. The demand for service class i arrived from T until t is described by $\hat{D}_{it}$. If a request for service i occurs in

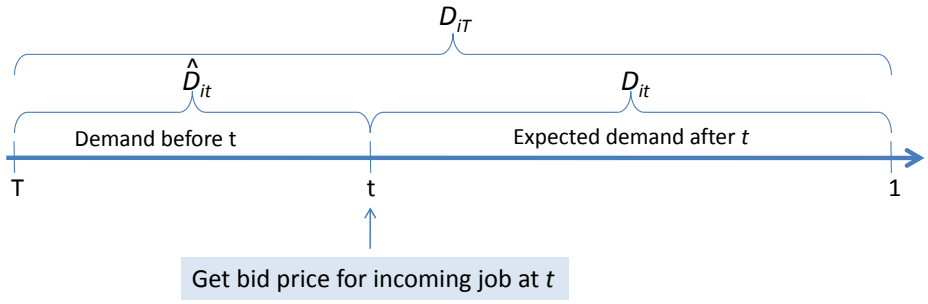


Figure 4.2: Demand definition in a finite time

time slot t, the demand arrived until t changes from its previous value $\hat{D}_{it}$ at time $t+1$ to its new value $\hat{D}_{it}+1$ at time t (note that the time runs backwards).

In Revenue Management it is often assumed that the so-called low-fare customers book earlier than the high-fare customers (Belobaba, 1989). This assumption is also valid in this chapter. The probability of arrival of a low-class request (service $i=1$) is high at the beginning of the booking period, and decreases over time. On contrary, the probability of arrival of a high-class request is low at the beginning of the booking period, but increases in time. In the present setting the provider offers several services. The low-before-high restriction is always held for the cheapest and for the most expensive service, although due to the stochastic nature, there is a small probability that high-fare requests can arrive in the early stage as well.

4.3.1 Bid Price Control

The dynamic programming problem was heuristically approached by several authors. In this section, three different models are presented, of which two of them serve as a benchmark for the simulation results in Section 4.5. The Randomized Linear Programming (RLP) model is not considered, since the improvement to the deterministic version was on the one hand not significant in the Cloud context and on the other hand aggregated values with different demand scenarios were inherently incorporated in the simulation.

4.3.1.1 Deterministic Linear Programming Model

The network model for bid-prices assumes expected demand information and excludes the stochastic nature of the demand (Glover et al., 1982; Williamson, 1992). Based on demand forecasts the expected aggregate demand-to-come D_{it} for the remaining booking periods is calculated, and it is assumed that the demand is equal to its mean values. An approximation for the objective-value function V is obtained by:

$$Max.\ V(x) = \sum_{i=1}^{n} r_i \cdot x_i \tag{4.1}$$

$$s.\,t. \sum_{i \in A^h} a_{hi} \cdot x_i \leq c_h - \bar{c}_{ht}\ \forall h \in \{1,\ldots,m\} \tag{4.2}$$

$$0 \leq x_i \leq D_{it}\ \forall i \in \{1,\ldots,n\} \tag{4.3}$$

The condition for accepting a request is: The revenue r_i yielded through the sale of service i must be greater than or equal to the sum of the resource consumptions of service i weighted with the corresponding bid-prices (Talluri and van Ryzin, 1998):

$$r_i \geq \sum_{h \in A_i} a_{hi} \cdot \pi_{ht}. \tag{4.4}$$

Additionally, it must be ensured that there is still sufficient capacity of every resource available to satisfy the request. This is expressed by the condition:

$$a_{hi} \leq c_h - \bar{c}_{ht}\ \ \forall h \in A_i \tag{4.5}$$

(4.1) is the objective function, which maximizes the total revenue. The total revenue results from the sum of the prices r_i charged for each service multiplied by the number of units of each service sold in the booking period x_i. Constraint (4.2) ensures that enough capacity of each resource is available to satisfy the need for capacity by the number of allocated units of the services. Constraint (4.3) guarantees that the number of services sold are not below zero and do not exceed the expected demand-to-come.

The solution vector of the primal problem is discarded, and the variables of the optimal solution of the dual problem are used as bid-prices (Talluri and van Ryzin, 1998). If the constraint (4.2) is linearly dependent for all resources h, then only one solution for the dual prices exists. Otherwise, according to Paris (1981), the optimal solution can have multiple optimal dual bid-price vectors. The Deterministic Linear Programming (DLP) can either be solved at the beginning of the booking period with the given demand forecast by using static bid-prices or, as a more dynamic approach, by recalculating the bid-prices at certain data collection points during the booking period. The former approach has only one bid price over the entire horizon and thus deterministically accepts or rejects certain services, e.g. all service $i = 1$ are rejects, while service $i = 2$ and $i = 3$ are accepted. The latter is advantageous

in order to keep up a certain precision of the bid-prices. The main benefit of the DLP model is that it can be solved efficiently, which makes it popular for practical applications. Its performance strongly depends on the size of the network as well as on the reliability of the demand forecasts. However, this model does not imply any uncertainty in demand. Furthermore, the accuracy of bid-price values in the DLP model depends on how frequent these recalculations are performed. The most frequent calculation of bid-prices is carried out by recalculating bid-prices each time a request occurs. Another drawback of the DLP model is that the dual variables of resources can be zero. Hence, capacity for a resource is higher than the mean demand (Talluri and van Ryzin, 1998). Consequently, too low bid-prices can be the result. A simple numerical example is provided to illustrate the problem: consider a network with two resources and two services with the following resource usage (matrix A_i) and prices r_i (Table 4.1):

Table 4.1: Dual variables example of the DLP problem

	Services	$i=1$	$i=2$
Resources	$h=1$	2	4
	$h=2$	4	2
	Prices r_i	4.00	5.50

The expected total demand for service $i=1$ is $D_{1T}=5$, and for service $i=2$ $D_{2T}=8$. Furthermore, assume that the available capacities of both resources are $c_1=c_2=40$. Given this information, the dual problem looks as follows:

$$Min.\ V(y) = 40y_1 + 40y_2 + 5s_1 + 8s_2; \tag{4.6}$$

$$s.\ t.\ 2y_1 + 4y_2 + s_1 \geq 4.00; \tag{4.7}$$

$$4y_1 + 2y_2 + s_2 \geq 5.50; \tag{4.8}$$

$$y_1, y_2, s_1, s_2 \geq 0 \tag{4.9}$$

The total demands for the services can be converted into the total demands per resource by calculating: $2 \cdot 5 + 4 \cdot 8 = 42$ for resource $h=1$ and $4 \cdot 5 + 2 \cdot 8 = 36$ for resource $h=2$. Obviously, the total demand for resource $h=1$ exceeds the available capacity $c_1=40$ of resource ($42 > 40$). Contrarily, the total demand for resource

$h = 2$ is below the capacity of resource $h = 2$ ($36 < 40$). Solving the dual problem leads to the solution vector with the optimal dual variables $y_1 = 1.375$ and $y_2 = 0.0$. For resource $h = 2$ the capacity is higher than the demand for the resource, and thus, the optimal dual variable y_2 is zero. Using such low values as bid-prices can result in inefficient outcomes in terms of revenue performance as well as resource utilization, because condition (4.4) can be fulfilled for too many of the low-revenue requests. Hence, low-fare services are sold too frequently and less capacity can be reserved for the 'later arriving' high-fare requests yielding more profit.

4.3.1.2 Randomized Linear Programming Model

The Randomized Linear Programming (RLP) model induces stochastic information. The expected demand as in the DLP case is replaced by a random demand vector D (Smith and Penn, 1988). For instance, Gallego and van Ryzin (1994) model the demand as a Poisson process. The probability distribution of the demand for each service is used to generate different scenarios of demand-to-come D_{it}. The optimal solution of this problem represents a random variable, which provides the approximation to the objective-value function V. According to Talluri and van Ryzin (1999) the application of RLP leads to a significantly higher revenue than DLP. However, in various test settings in this thesis, no significant difference to DLP could be identified for Cloud services and thus this approach is disregarded.

4.3.1.3 Certainty Equivalent Control

Another approach called certainty equivalent control extends the concept of bid-prices and directly calculates an approximation of the opportunity cost for every service and not for every resource (Bertsimas and Popescu, 2003). For this purpose it solves two instances of the DLP problem described above: The first instance solves the initial DLP problem (4.1) and the second instance subtracts the amount of resources demanded by the request from the remaining capacity of the resource

$$c_h - \bar{c}_{ht} - a_{hi}, \forall h \in \{1, \ldots, m\}. \tag{4.10}$$

The approximation of the opportunity cost of service i is then obtained by subtracting the objective function value of instance 2 ($V'(x)$) from the objective function value of instance 1 ($V(x)$). This approximation does not depend on the optimal dual variables. Thus, the drawback of multiple optimal dual variables of the linear programming model is eliminated. The Certainty Equivalent Control (CEC) policy requires forecasts for the total demand for each service, as well as forecasts for the expected demand-to-come (D_{it}). The main advantage of the CEC policy arises from

the numerous and periodic updates of the approximation of the opportunity costs, thereby guaranteeing a certain accuracy. Since CEC is based on the DLP problem, it shares the same disadvantage of only incorporating expected demand and not considering uncertainty of the demand process. However, Bertsimas and Popescu (2003) have proven that CEC outperforms DLP.

4.3.2 Customized Bid Price Policy

A different NCC approach for calculating bid prices was proposed by Klein (2007). The idea of this approach is to use simple linear additive functions to calculate bid prices every time a request occurs. The functions are based on parameters, which can easily be kept on track during the booking period, such as the current amount of reserved capacity $\bar{c}_{ht}$ as well as the expected demand-to-come D_{it}. It uses a continuous time model with a booking period represented by the time interval $[T;0]$. The concept further involves the determination of coefficients (control variables) via simulation-based optimization. The control variables are used for calibrating the bid-price functions adequately, which are evaluated offline before the time horizon. In the following, the basics of the Self-Adjusting Bid-Price (SABP) concept are explained. Subsequently the customized concept named Customized Bid-Price Policy (CBPP) is presented, which uses a genetic algorithm for calibrating the control variables. Additionally, Section 4.5 outlines a thorough statistical analysis of the benchmark.

In SABP two different linear bid-price functions are explained (Klein, 2007). One is time-oriented, and involves the time remaining to sell the services, and the other one is resource-oriented. However, the focus here is on the resource-oriented alternative, because it takes into account information about the future resource demands of incoming requests. It can describe the current booking situation more accurately, and is better suited to evaluate resources in Clouds. Klein (2007) has also proved that the resource-oriented version outperforms the time-oriented approach. Bid prices are calculated for each resource h every time a request occurs. The variable π_{ht} denotes the bid price of resource h at time t. In the case of an arrival of a request for service class i at time t, the bid-price control decides whether to accept or to reject the request.

The components of the bid-price function describe the current booking situation. Consequently, if many requests arrive, the computations of bid prices take place more frequently. Therefore, the accept/reject decisions directly affect the values of future bid prices. The formula of the bid-price function for a resource h at time t is:

$$\pi_{ht} = \overline{\pi}_h + \alpha_h \cdot \bar{c}_{ht} - \beta_h \cdot u_{ht}. \tag{4.11}$$

$\overline{\pi}_h$ denotes the base bid price, which is a control variable and provides the basis for the bid price calculation. It is determined by simulation-based optimization. Naturally, the value of $\overline{\pi}_h$ influences the bid prices. Therefore, the question arises sa to how the base bid price is set. Klein (2007) calculates it by creating a random number and multiplying it with the minimum bid price of resource h. The minimum bid price is the value at which, if it is exceeded, requests for at least one service class i are no longer accepted, and is computed by $\pi_h^{min} = min\left\{r_i/a_{hi} | i \in A^h\right\}$.

The bid-price function further consists of two parts: The first part of (4.11) $(+\alpha_h \cdot \bar{c}_{ht})$ is responsible for the increase of the respective bid price over time. The amount of reserved capacity of resource h at time t $(\bar{c}_{ht})$ can only be natural numbers counting the number of resources that are already allocated. If a request is accepted, the bid price of a resource increases by the value $\bar{c}_{ht}$ multiplied with α_h. Thus, the bid price of resource h increases only through the acceptance of incoming requests. This corresponds to the fact that available resources get less due to sales and hence become more expensive.

The second part of the formula $(-\beta_h \cdot u_{ht})$ decreases the bid price for every request made. A decrease is required to avoid free capacity leading to a revenue loss. If the function had no decreasing part, the bid price would increase monotonically after every acceptance. From a certain point in time every future request will be rejected, and no more sales could take place, although free capacity is still available. u_{ht} is the capacity required to satisfy the demand for service $i \in A^h$ until t. The demand until t for a service $i \in A^h$ can be calculated by $D_{iT} - D_{it} = \hat{D}_{it}$. It requires forecasts of the total expected demand per service i (D_{iT}), as well as forecasts of the expected demand-to-come (D_{it}) for every point in time t until the end of the booking period. u_{ht} is calculated by

$$\sum_{i \in A^h} a_{hi} \cdot (D_{iT} - D_{it}). \tag{4.12}$$

For every service, its demand for resource h (a_{hi}) is multiplied by the expected demand until t $(D_{iT} - D_{it})$, and the sum of these services is taken. The values of u_{ht} increase over time as more demand is realized. This leads to a decrease in the bid-price function when additional demand arrives over time. The two parts of the resource-oriented bid-price function ensure that the total value of a bid price (π_{ht}) only increases, if requests are accepted and decrease otherwise. The increase amplifies if the amount of reserved capacity $(\bar{c}_{ht})$ is high. The parameters α_h and β_h are control variables required for calibration purposes in the simulation-based optimization. They have a strong impact on the accept/reject decisions. For instance, very high values for coefficient α_h and very low values for coefficient β_h lead to more

frequent reject decisions, because the increase of the bid-price function turns out to be too high. This would imply losses in revenues due to rare sales. In the opposite case, very low values of α_h and very high values of β_h result in rather low bid prices, and thus, it can happen, that capacity is mostly sold to low-fare classes leading to potentially lost revenues, which could be yielded by high-revenue requests arriving later. Because of these reasons, promising values for the control variables are obtained via simulation-based optimization.

The Customized Bid-Price Policy (CBPP) approach is based on the resource-oriented bid-price function of SABP. The bid price of resource h at time t is calculated by the formula:

$$\pi_{ht} = \overline{\pi}_h + \alpha_h \cdot \frac{\bar{c}_{ht}}{c_h} - \beta_h \cdot \frac{u_{ht}}{U_{hT}} \tag{4.13}$$

The control variables $\overline{\pi}_h$, α_h, and β_h are determined via a genetic algorithm, which is described below. Again, the bid price calculation is based on the base bid price $\overline{\pi}_h$. In CBPP policy, the genetic algorithm uses the minimum bid price (π_h^{min}) as upper bound for the base bid price. Thereby, it is guaranteed that the base bid price does not turn out to be too high, which would lead to rejects of requests for all service classes.

The customized bid-price function is also based on two parts. The first part ($+\alpha_h \cdot \frac{\bar{c}_{ht}}{c_h}$) increases the bid price π_{ht} if requests are accepted. The amount of reserved capacity of resource h at time t ($\bar{c}_{ht}$) is divided by the total capacity of resource h (c_h), and hence can only take values in $[0;1]$. An acceptance of a request for service i leads to an increase of the bid price of resource h to the amount of the delta of the value $\frac{\bar{c}_{ht}}{c_h}$. The increase of the bid price also depends on the value of α_h.

The decreasing part ($-\beta_h \cdot \frac{u_{ht}}{U_{hT}}$) lowers the bid price every time a request is made. As explained above, u_{ht} corresponds to the amount of capacity needed to satisfy the demand until t for the services $i \in A^h$.

U_{hT} is the capacity of resources that is required to satisfy the total expected demand of the complete booking period calculated by $\sum_{i \in A^h} a_{hi} \cdot D_{iT}$. U_{hT} can also be denoted as the total resource demand. It requires a forecast of the total demand for service class i in the booking period. The quotient $\frac{u_{ht}}{U_{hT}}$ is always in $[0;1]$, and increases over time as more demand is realized. Hence, the bid price decreases with incoming requests. Reasonable values for the control variables $\overline{\pi}_h$, α_h, and β_h are obtained by a genetic algorithm in order to minimize inefficient outcomes.

The main advantage of SABP and CBPP is the very frequent recalculation of bid prices with minimum computational effort. Thereby, information about the current

booking situation is always considered, and the bid prices exhibit a certain precision. Klein (2007) states that this approach is robust to errors in the forecast. If the realized demand is less than the forecasted one, less capacity units may be reserved, and more capacity is available to satisfy incoming requests. Thus, the increasing part of the bid-price function is lower, and more requests with lower revenues can be accepted. On the other hand, if the realized demand is higher, the bid price increases strongly as more capacity is reserved due to the acceptance of requests. Furthermore, a strict fragmentation of the services into only two subsets (S_1: Accepted, i.e. $r_i \geq \sum_{h \in A_i} a_{hi} \cdot \pi_{ht}$; S_2: Rejected, i.e. $r_i < \sum_{h \in A_i} a_{hi} \cdot \pi_{ht}$) is avoided. This fragmentation arises in models, which do not perform a frequent recalculation of bid prices. Given static bid prices, it is fixed, which service classes can be accepted and which not. Through permanently updating the bid prices a certain accuracy is guaranteed, and a strict fragmentation cannot occur. Moreover, the offline optimization of the control variables allows to save time during the period, when requests arrive. A sum of the current bid prices for every resource has to be calculated, while the bid prices are updated automatically over time.

Although the concept shows some robustness against forecast errors as mentioned above, it does not imply stochastic information about demand. This assumption is not very realistic, and is not considered in other deterministic models as well, such as DLP and CEC. The computational effort before the incoming request period, which is required for finding appropriate values for the control variables $\overline{\pi}_h, \alpha_h$, and β_h is significantly high. It mainly comes from the necessity for simulation-based optimization. Another weakness of the CBPP model is the large amount of forecast values needed, which requires a lot of memory. For each point in time and for each service offered, the expected demand-to-come ($\overline{D}_{it}$) must be stored in order to be able to calculate the demand until t and the resource demands (u_{ht}) for each incoming request.

Note that the SABP concept of Klein (2007) has been developed for managing resources in the context of airlines. The resource consumption of airline services, which are seats in certain booking classes on certain flights at certain dates, is very different to the resource usage by services in Clouds. To demonstrate this, consider the following example. An airline offers different types of booking classes (e.g. economy class and business class) on each of its flight. Depending on the booking classes and on the flights offered, the airline can offer a certain number of services. However, in this case the resource consumptions by the different services usually are all one unit of a single resource. Table 4.2 represents a service-resource mapping for Cloud services with $m = 3$ resources and $n = 3$ services.

It can be easily observed that the resource demands (a_{hi}) in the two areas of application show a major difference. Matrix A_i in case of services in Clouds ex-

Table 4.2: Resource usage by services in Clouds

Service i	$i=1$	$i=2$	$i=3$
CPU	2	4	8
Memory	2	8	4
Storage	8	2	4

hibits significantly higher natural numbers than in case of airline services, which usually consume one seat on a single-leg flight. Even if an airline offers multi-leg services, that is, if someone buys a sequence of flights over multiple destinations, the resource usage per flight is always one seat. Note that group bookings are not considered in these models. Group bookings require a different demand modeling, since the amount of services in group bookings vary over time. The service resource mapping is fixed over the entire time horizon. Hence, accepting one booking request in the context of services in Clouds means a considerably larger change in capacity compared to the acceptance of a booking request for an airline ticket, but is not comparable to the group booking problem. Furthermore, accepting service with different kinds of resource demand can significantly affect the yield. It is possible to accept services which have a lower revenue, but need abounded resources, while high-fare services have to be rejected due to resource scarcity.

Depending on the type of service requested by the consumer, the capacity of multiple resources may be decreased in case of an acceptance, which applies to all three services in this example (table 4.2). Therefore, a stronger competition for multiple resources takes place between services in Clouds. Hence, the usage of the original bid-price function (4.11) would lead to very high values of the increasing and decreasing part of the function in the setting of Cloud services, and thus, some bid price values can turn out to be inappropriate. This is avoided by using relative values in (4.13).

4.3.3 Non Optimal Outcome

In certain cases bid-price control can fail to produce an optimal decision (Talluri and van Ryzin, 1998). In the context of services in Clouds, an illustration of these cases is provided below:

Example one:
Assume that a provider offers three resources and three services. The resource consumptions of the services are given in table 4.3.

Table 4.3: Non-optimality example 1 - service-resource mapping and service prices

	$i=1$	$i=2$	$i=3$
$h=1$	2	4	8
$h=2$	2	8	4
$h=3$	8	2	4
Price r_i	15.00	17.00	23.00

Due to low resource availability only one request for all services can be accepted. It is assumed that a request for service i occurs with different probability p_i at time t. There are three possible options for which type of service to accept at $t=2$. One possible acceptance option may arise at $t=1$: A request for service $i=3$ occurs with a probability of p_3 of 0.8, and a probability of 0.2 for no request to arrive (Table 4.4). Remember that only one request can arrive per time slot.

Table 4.4: Non-optimality example 1 - available capacity for acceptance of one more request

	Request for i	$i=1$	$i=2$	$i=3$	Option 1	Option 2	Option 3
$t=2$	p_i	0.4	0.4	0.2	$i=1$	$i=2$	$i=3$
$t=1$	p_i	0	0	0.8	-	-	$i=3$

The amount of available capacity is 8 units of each resource. Therefore, at $t=2$ an acceptance of a request for service i leads to the values of available capacity at $t=1$ as outlined in table 4.5.

Table 4.5: Non-optimality example 1 - capacity left at $t=1$ after acceptance of service i

Acceptance of service i	$i=1$	$i=2$	$i=3$
$h=1$	6	4	0
$h=2$	6	0	4
$h=3$	0	6	4

Obviously, an acceptance of any service request at $t=2$ leads to a lack of capacity for one resource in the next timeslot $t=1$ and therefore no incoming request could be accepted. An optimal decision in this case is to reject incoming requests for the services with lower revenues ($i=1$ and $i=2$) at $t=2$, and to only accept a request for the high-class service $i=3$ (if it occurs). The corresponding bid-price control requires the following conditions, where bid prices are weighted with consumptions of the respective resources:

- Condition 1: $r_1 < \sum_{h \in A_{i=1}} a_{h1} \cdot \pi_{ht}$ with $\sum_{h \in A_{i=1}} a_{h1} \cdot \pi_{ht} = 2 \cdot \pi_{1t} + 2 \cdot \pi_{2t} + 8 \cdot \pi_{3t}$
- Condition 2: $r_2 < \sum_{h \in A_{i=2}} a_{h2} \cdot \pi_{ht}$ with $\sum_{h \in A_{i=2}} a_{h2} \cdot \pi_{ht} = 4 \cdot \pi_{1t} + 8 \cdot \pi_{2t} + 2 \cdot \pi_{3t}$
- Condition 3: $r_3 >= \sum_{h \in A_{i=3}} a_{h3} \cdot \pi_{ht}$ with $\sum_{h \in A_{i=3}} a_{h3} \cdot \pi_{ht} = 8 \cdot \pi_{1t} + 4 \cdot \pi_{2t} + 4 \cdot \pi_{3t}$

Table 4.6 contains the respective resource consumptions (A^h) and a set of bid prices in period $t = 2$, for which conditions one and two hold, but condition three is violated.

Table 4.6: Non-optimality example 1 - set of bid prices

	π_{ht}	$i = 1$	$i = 2$	$i = 3$
$h = 1$	1.75	2	2	8
$h = 2$	1.55	4	8	2
$h = 3$	1.25	8	4	4
r_i		15	17	23
$\sum_{h \in A_i} a_{hi} \cdot \pi_{ht}$		16.6	21.9	25.2

For the given revenues per service all three conditions cannot be kept. There is a conflict in the requirement that $\sum_{h \in A_{i=1}} a_{h1} \cdot \pi_{ht}$ must be above $r_1 = 15.00$ as well as $\sum_{h \in A_{i=2}} a_{h2} \cdot \pi_{ht}$ above $r_2 = 17.00$, but at the same time $\sum_{h \in A_{i=3}} a_{h3} \cdot \pi_{ht}$ must be below $r_3 = 23.00$ with the same bid prices π_{ht}.

As table 4.6 shows, the set of bid prices would lead to the rejection of all service requests at time slot $t = 2$, because the revenue of each service r_i is below the sum consisting of the resource consumption weighted with the bid price for all resources (e.g. 23 < 25.2). However, the optimal policy would be to accept service $i = 3$ and to reject the other two services. This example demonstrates the non-optimality of bid-price control in the Cloud context, although it shows some difference compared to the airline case provided by Talluri and van Ryzin (1998).

An interesting aspect in the given example is that through lowering the bid price π_{1t} by the amount of 0.3, the bid-price policy would produce the optimal decision, i.e. reject requests for services $i = \{1,2\}$ and accept a request for service $i = 3$. This can be explained by the highest amount of consumption of resource $h = 1$ by service $i = 3$ compared to the other service classes. Hence, bid price π_{1t} has a strong impact on the calculation of $\sum_{h \in A_{i=3}} a_{h3} \cdot \pi_{ht}$, and lowering the value of π_{1t} leads to a smaller total sum ($\sum_{h \in A_{i=3}} a_{h3} \cdot \pi_{ht}$) below price r_3.

From the observations above, the question is raised, whether bid-price control is able to compute bid prices which can lead to optimal decisions in the above cases. Another important aspect is that these situations can only occur, if demand for the services shows a certain mix of different incoming requests, i.e. there is no order of low-fare requests arriving strictly before high-fare requests (see Section 4.3). From a theoretical point of view bid-price control can produce non-optimal decisions. However, the previous example showed that there is also the possibility of optimal decisions. Bid prices depend on the service prices, the realized demand, the reserved capacity as well as on future requests. Regular updates of bid prices is a key feature to achieve accurate results. If updates are performed less frequently, the bid prices are more static, and the potential of producing an optimal decision is decreased.

The previous example only considered a small fraction of time slots. Examples for non-optimal bid prices are neither restricted to the end of the booking period nor to the amounts of available capacity close to the total capacity limits. Two main reasons exist for the potential non-optimality of bid-price controls. First, bid prices have some kind of marginal costs property and cannot reasonably evaluate large changes in capacity (Domschke and Klein, 2004), which are realized by selling a service, in relation to the availability of the services. Another reason for the non-optimality is that the relation between the amount of capacity used to satisfy a request and the opportunity cost of selling a service may be non-linear. That is, if a sale of a service $i = 1$ has the same opportunity cost as the sale of service $i = 2$ and service $i = 3$, and the resource demand by the different services is different as well, it is not possible to express this relation by a linear interrelation. However, bid-price control assumes to have a linear relationship between the resources for one service.

In the application context of Cloud Computing a sale of a service means a considerably larger change in capacity compared to the airline context (Section 4.3.2). It is important to note that in case of different resource consumptions by the Cloud services the complexity of the interdependency between resources and services increases. The bid price of resource h is weighted with the amount of consumption of this resource by the respective service class. Moreover, it is common that Cloud services consume several resources at once. In case of an incoming request for a low-revenue service in time slot t at a scarce level of available capacity, and when the probability of a high class request in next time period is high, a decision rule could be to explicitly set the bid prices in period t to high values, so that low-fare requests are rejected. Thereby, capacity is reserved for the potentially high-revenue request in period $t - 1$. This suggestion is based on the internal provider policies, e.g. preferring “gold customers” over “silver customers” and is not modeled explicitly in the approach of this thesis. However, it can be integrated in this model by limiting the available capacity for a certain time Δt.

Table 4.7: Non-optimality example 2 - requests arriving

Time	No.	Request for i	Revenue r_i
$t = 8$	1	$i = 1$	15.00
$t = 7$	2	$i = 2$	17.00
$t = 6$	3	$i = 1$	15.00
$t = 5$	4	$i = 3$	23.00
$t = 4$	5	$i = 2$	17.00
$t = 3$	6	$i = 1$	15.00
$t = 2$	7	$i = 3$	23.00
$t = 1$	8	$i = 2$	17.00
$t = 0$	9	$i = 3$	23.00

Example two:
This example outlines how mixed strategies of accepting different kind of Cloud services can increase revenue. Instead of only accepting the most beneficial services, a low-fare service could have more impact on revenue. Due to the larger change in capacity, this scenario plays a major role in Clouds. In this example, sufficient capacity is available for accepting three or four service requests (depending on the type of service requests). The same values of revenues and resource demands as in the previous example are used. Table 4.7 contains an ex-post view on the arrival of different requests at time t and subsequent time slots. Arrival probabilities are not considered. The values of available capacities at time t are for $c_1 = 24$, for $c_2 = 20$, and for $c_3 = 20$.

In this situation the provider can sell the remaining capacity in multiple ways. Possible acceptance strategies are summarized in table 4.8. Strategy no. 2 yields the highest revenue in this example, by accepting one $i = 1$ request, one $i = 2$ request, and two $i = 3$ requests. This strategy further results in the best utilization, i.e. lowest amount of unused capacity of each resource. It does not depend on which of the requests are accepted, e.g. for $i = 1$ the capacity control can accept request no. 1, 3, or 6.

Due to the multiple options of allocating capacity and due to the uncertainty of demand, i.e. at time t it is not clear, which types of requests will occur in which order. Given the remaining amounts of available capacities, the outcome can easily become non-optimal. For instance, if the provider accepts the first three incoming requests (no. 1, 2, and 3) in a first-come-first-serve manner, a high-fare request cannot be accepted due to insufficient capacity of resource $h = 3$. But a request for service

$i = 2$ can still be accepted, e.g. no. 5 or 8. Then, the maximum possible revenue is 64 (similar to strategy no. 5).

Strategy no. 6 (Table 4.8) provides an example of a rather poor result concerning revenue performance: Through the admission of requests no. 1 and 3 (or also 3 and 6) as well as request no. 8 the capacity for an additional acceptance of a service-2 requests remains unused and the revenue output is only 47. Such cases may appear when the bid prices between two certain time slots are too high. Strategy 7 also results in unused capacity, because no more requests occur after $t - 9$.

Table 4.8: Non-optimality example 2 - possible strategies in example 2

Strategy	Admissions per i			Admit requests	Unused capacities			Revenue
	$i = 1$	$i = 2$	$i = 3$	with no.	$h = 1$	$h = 2$	$h = 3$	
1	0	0	3	4, 7, 9	0	8	8	69
2	1	1	2	1, 2, 4, 7	2	2	2	80
3	0	2	1	2, 4, 5	8	0	12	57
4	2	0	1	3, 6, 9	12	12	0	53
5	2	2	0	2, 3, 5, 6	12	0	0	64
6	2	1	0	1, 3, 8	16	8	2	47
7	1	1	1	3, 5, 9	10	6	6	55

The explained aspects demonstrate that even if the bid-price control accepts three high-class requests ($i = 3$), the revenue and utilization is poorer compared to the best strategy no. 2. In terms of utilization and revenue performance, a mixed acceptance strategy of service requests performs better in some cases than accepting only high-fare requests. Naturally, it strongly depends on the service configurations, the number of services offered as well as on the prices of the services. In addition, the considerations clarify that there is a conflict between reserving capacity for requests for high-revenue services arriving later and the risk of achieving poor utilizations and revenues. This is due to the sale of too many low-revenue services or due to significant fluctuations in high-fare demand.

Since demand modeling implies uncertainty, the described scenarios are very hard to discover during the booking period. It is possible that a multitude of such situations occur. Therefore, the use of an appropriate and accurate forecasting method is important in practice. Furthermore, the determination of the optimal allocation of services can only be performed ex-post by trying all possible combinations of acceptance of the different requests occurred in the booking period (Talluri and van Ryzin, 2004b). Although there are circumstances, under which bid-price controls produce non-optimal decisions, Talluri and van Ryzin (1998) state that when

capacity and the volumes of sales are large, bid-price controls perform asymptotically optimal.

4.4 Simulation Environment

CBPP was implemented in Java to evaluate the outcome of the algorithms and to understand the dependency of the several parameters. The simulation was executed on a Windows 7 machine with an Intel Centrino Core 2 Duo CPU and 2GB RAM in a Java Runtime Environment 1.6. The OR-Objects[5] package for Java was used to solve the DLP or CEC instances. The genetic algorithm was implemented via the Java Genetic Algorithms Package[6]).

Furthermore, several assumptions, which are common in the Revenue Management context, also apply in this simulation. It is assumed that the provider has certain demand information from past booking periods, and is able to perform a more or less accurate demand forecast. Fluctuations in demand will be handled by generating different demand scenarios during the simulation. Customers, who do not use the booked services, do not get a refund. Hence, if consumers book a service they will have to pay for it. Cancelations and no-shows are not considered in the model. Demand for each service is independent and can be modeled by a probability distribution (Weatherford and Belobaba, 2002).

Several parameters describe the setting of the simulation. This thesis focuses especially on the interdependency between the bound values for the genetic algorithm parameters, the price variation, the service-resource mapping and their impact on Demand-to-Capacity ratio (DCR), the bid price for each resource and the achieved revenue. Changes in demand and capacity will obviously influence the DCR and thus the outcome of the algorithm. The impact of these parameters will be described in the next two subsections, since they are necessary to evaluate CBPP in Section 4.5.

4.4.1 Genetic Algorithm

Genetic algorithms, introduced in 1975 by John Holland (Holland, 1975), are used as a tool for search and optimization. They belong to the class of evolutionary algorithms defined in the 1960s by Rechenberg (1973). Genetic algorithms are optimization concepts, which search a solution space of a given problem for reasonable solution values. When searching for good or optimal solutions, a genetic algorithm does not create all potential solutions at the beginning. Rather, it works by regard-

[5]OR-Objects Java package 1.2.4 (`http://opsresearch.com/`)
[6]Java Genetic Algorithms Package 3.4.3 (`http://jgap.sourceforge.net`)

ing only a small part of the solution space. Given this part of the solution space, a simulation of an evolution is performed using the survival of the fittest strategy. Individuals who are fitter than other individuals, have a higher probability of surviving and of still being persistent in the next generation in the evolution process. The term fitness has to be adapted to certain circumstances or to an environment (Goldberg, 1989). Before describing the structure of the genetic algorithm, some common terminology from biology is introduced (Mitchell, 1998).

Chromosomes are the carriers of the genes. They contain a set of genes. A gene is a unit within a DNA double-strand molecule. Genetic information about a certain characteristic is encoded in the gene. Each gene represents a certain attribute of an individual and is located at a certain position (locus) within the chromosome. The possible values or settings of a gene are denoted as alleles. An individual in this context contains one chromosome, thus, the term chromosome refers to the term individual here. A population consists of a collection of chromosomes or individuals.

A genetic algorithm consists of the following parts, which have to be defined depending on the application context and the problem to solve (Mitchell, 1998):

- An appropriate chromosome has to be defined and the number of genes in the chromosome as well as the gene values have to be determined. A definition of the bounds for the genes is essential.
- Several genetic operators describe the evolution process. It is essential to define which methods to be used for recombination and evolution of the genes. There are three basic types of genetic operators: The selection operator, the crossover operator, and the mutation operator.
- A random initial population has to be created. Before starting the evolution process, an initial population according to the defined chromosome representation has to be created.
- A fitness function has to be defined, which is responsible for calculating the fitness of each chromosome within the population. The term fitness depends on the application context. It can be measured in positive real numbers, and high fitness values mean that the given chromosome is well adapted to certain circumstances.
- Furthermore, it has to be specified when to stop the evolution process. For instance, a maximum number of evolution steps can be defined.

As mentioned before, the objective of the genetic algorithm is to find adequate values for the control variables of the customized bid price function (4.13). The bid price function contains three control variables: The base bid price $\overline{\pi}_h$ as well

as α_h and β_h. All the parameters are calculated for each single resource. Hence, the genetic algorithm has to find three adequate values for each resource, which leads to a total number of genes in a chromosome of $3 \cdot m$. Moreover, the genetic algorithm must simulate a complete booking period T, which includes the demand expected for the given sales period based on the forecast by the provider. The application of the genetic algorithm in CBPP is described below:

Chromosome representation:
The choice of an appropriate chromosome representation, which defines how many genes the chromosome contains, and which values the genes are allowed to obtain, is a key component. The control variables $(\pi_h, \alpha_h, \beta_h)$ in (4.13) are used as genes within the genetic algorithm and are optimized in the evolution process. The size of the chromosome depends on the number of resources used by the provider to configure the offered services. Another important issue is how to specify the numerical range in which the values of the control variables can lie. The genes containing the values of the base bid prices are allowed to take values in the interval $\left[0; \pi_h^{min}\right]$, which guarantees that the base bid price is always below the minimum bid price. Thereby, the production of counterproductive values is avoided, as in the case when the base bid price exceeds the price of service i ($\overline{\pi}_h > r_i$), which would lead to rejecting requests for all services i already at the beginning of the booking period.

Genetic Operator:
Based on the three basic genetic operators "selection operator", "crossover operator" and "mutation operator", extended versions such as "averaging crossover operator" or "ranged swapping mutation operator" were created to receive better results. After testing all these operators integrated in JGAP, two of them were selected for the evaluation, namely "swapping mutation operator" and "two way mutation operator", since they outperformed the others. While the crossover operator randomly chooses a position in a chromosome and changes the subsequence of genes after that locus with the subsequence of genes of another chromosome at the same locus, the swapping mutation operator does not alter (mutate) the genes of a chromosome. It selects a start position in a chromosome, and swaps the genes after that position. The two way mutation operator works in two steps. In step one it assumes that every gene in a chromosome has a different effect on the fitness value if it is mutated. Therefore, it dynamically adapts the mutation rate for different genes, and selects a gene for mutation with a higher probability to least affect the fitness value. After the selection of a gene, step two continues with a traditional mutation. It randomly mutates the genes of a chromosome. It goes through all chromosomes in the population, and mutates a gene with a certain probability, which is called the

mutation rate. If a gene of a chromosome is mutated, it is a candidate chromosome for the natural selection process.

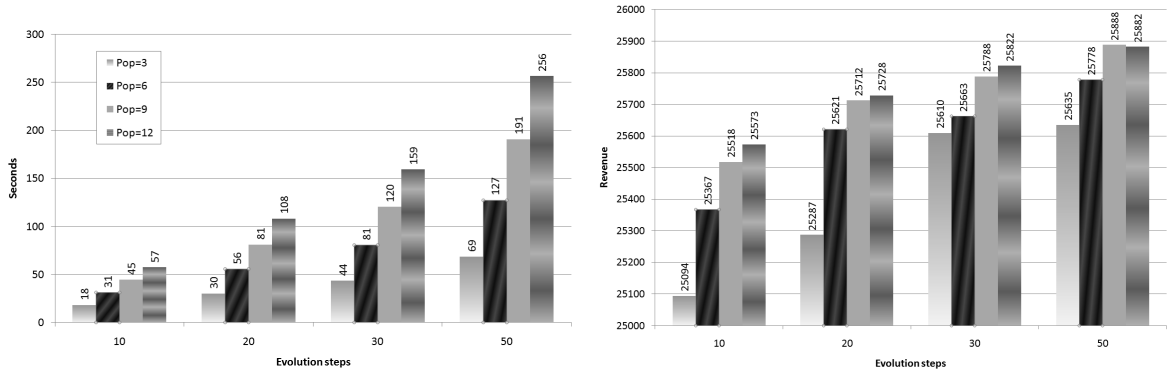
Initial population:
Given the numerical bounds of the gene values and the genetic operators, an initial population is created randomly depending on the population size specified in the genetic algorithm. Based on this random input, the optimal parameter values are determined during the evolution steps. A bigger population size would lead to higher computation time, but it could also lead to better results.

Fitness function:
The fitness function calculates the fitness of each chromosome. Fitness in this context is the revenue yielded with a set of gene values based on the expected demand. The fitness function runs through the complete forecast, and simulates accept and reject decisions using the gene values of the chromosome to be evaluated as control variables. In doing so, the potential revenue based on the accept/reject decisions is determined.

Evolution steps:
Through various simulation settings it was identified that 20 evolution steps already lead to remarkable results without a large computation effort. An analysis of the runtime is provided below.

The genetic algorithm needs more computational power than DLP and CEC to identify the best gene values for an optimal decision. Naturally, the runtime is dependent on the population size (POP), the number of variables (genes) to search for and the number of evolution steps (ES). The number of genes are static and represented by the variables α, β and $\overline{\pi}_h$. The impact of the values for these variables are analyzed below. A higher population size leads to significantly more computation time. For example, the runtime for a setting with a population size = 9 and 30 evolution steps is almost 4 times longer than for a population size of 12 with the same amount of evolution steps on average (Figure 4.3a).

An increase of the evolution steps will result in a higher runtime as well. A doubling of the number of evolution steps leads almost to a doubling of the runtime of the genetic algorithm. It is important to state that a longer evolution time does not necessarily lead to a better solution. For example, in Figure 4.3b the setting with POP = 9 and ES = 30 gains higher revenue than the settings with POP = 6 and ES = 50. Furthermore, the runtime is slightly better (Figure 4.3a). Moreover, a rise in the population size leads to a longer runtime of the genetic algorithm, but it also adds diversity to the population, which in turn increases the probability of finding a



(a) Runtime comparison

(b) Revenue comparison

Figure 4.3: Runtime and revenue analysis with different number of population and evolution steps for a 4x5 setting.

better solution more quickly in terms of number of evolution steps. However, even in this case, the performance will not monotonically increase with higher population size. For ES = 50 the setting with POP = 9 performs slightly better than with POP = 12 on average despite the latter setting having a 34% longer runtime. The difference between the setting with the lowest revenue (POP = 3; ES = 10) and the setting with the highest revenue (POP = 9; ES = 50) is about 3% increase in revenue, which is significant in the Revenue Management context (van Ryzin and Vulcano, 2008). Since runtime and revenue are based on average values analyzed over 50 different runs, the outcome can strongly vary in these runs. Thus, an analysis on the standard deviation for every setting reveals that the deviation decreases with the increase of evolution steps (Table 4.9). Hence, it is more likely to receive a good revenue with more evolution steps and with a higher population size.

Table 4.9: Standard deviation for all settings

	Pop=3	Pop=6	Pop=9	Pop=12
ES=10	323,1	179,8	143,1	140,6
ES=20	232,4	146,0	135,2	129,4
ES=30	151,7	142,8	117,5	114,0
ES=50	128,9	121,6	107,1	94,3

After the definition of the chromosomes, operators, population size, the fitness function and the evolution steps, the genetic algorithm is executed. The genetic algorithm performs the following steps[7]:

[7]The pseudocode of a genetic algorithm is available in chapter 4 in the book from (Schöneburg et al. (1995)).

1. At the beginning an initial population is created based on the chromosome representation as described above. This generation is numbered as generation zero.
2. In the second step, the fitness value of all chromosomes in the population is calculated by the fitness function, which runs through the expected demand scenario using the gene values as control variables in the bid-price function.
3. After steps 1 and 2, the evolution process is started. In each evolution phase the selection operator randomly selects pairs or bigger subgroups of the population of chromosomes for reproduction.
4. The chromosomes selected by the selection function are reproduced, and then they are recombined (crossover) or mutated by the genetic operators defined.
5. Subsequently, some chromosomes of the current population are replaced by the new altered chromosomes thereby creating a new generation.
6. Given the new generation, the generation enumerator is incremented by one.
7. Steps 2 to 6 are repeated until the maximum number of allowed evolutions is reached.
8. When the evolution process is terminated, the fittest chromosome, i.e. the set of control variables with the highest potential revenue, is taken as input for the self-adjusting bid prices function.

In general, it is reasonable to use a higher population size and a larger number of evolution steps in order to increase the probability of reaching a near-optimal solution, provided that the runtime is reasonable as well. The outcome of the simulation depends on the simulation setting (e.g. service-resource mapping or pricing).

4.4.2 Simulation Process and Hypotheses

The previous two sections are embedded in a larger simulation process. This process can be subdivided into three phases: Demand modeling, offline calculation and online calculation (see Figure 4.4). Compared to online algorithms like DLP and CEC, CBPP requires an offline calculation phase to determine the appropriate values for the control variables.

Demand modeling defines the incoming request according to a certain probability function. A standard approach is to use a non-homogeneous Poisson process (Talluri and van Ryzin, 1999). Although a discrete time model is assumed for the demand modeling and the Poisson process is based on continuous time model, this demand data can be applied for discrete time simulation as well (Bertsimas and Popescu, 2003; Subramanian et al., 1999). Several studies have chosen a Beta distribution in combination with a Gamma distribution to model the demand data.

The Gamma distribution is used for the expected demand of a service and the Beta density function for the distribution of the incoming services over the time horizon (Bertsimas and de Boer, 2005; Weatherford et al., 1993). This thesis adapts this approach. The flexible structure of the Beta density function enables a flexible modeling of the demand over time for each service. Hence, the probability of each service class can have its peaks at different points in time during the entire period, which is required to have a certain mix of different services over time and not to follow a strict low-before-high order (see Section 4.3). Kimms and Mueller-Bungart (2007) provide a thorough literature review about on-demand data assumptions for Revenue Management and describe the demand modeling in detail.

Not only the demand has a certain probability function, but the prices and the service-resource mapping as well. Prices are selected from a uniform distribution between $[10;50]$ and every value $a_h i$ for the service-resource mapping between $[0;10]$. These values were necessary to appropriately distinguish between each service setting and to reduce homogeneity in the price set. The simulation-based optimization is commonly based on forecast (van Ryzin and Vulcano, 2008; Gosavi et al., 2007). Hence, forecast values for every timeslot have to be determined by using the same probability function as for the demand. Then, the actual demand is created and it can be compared to the forecast values in every timeslot. However, an analysis on forecast errors is not considered in this thesis.

The implementation of the simulation separates the demand generation from the actual simulation according to Frank et al. (2008). The determination of the control variables is done by the genetic algorithm in the *offline calculation phase*. The fitness function is equal to the revenue achieved by the algorithm. A higher revenue means a higher fitness value of the chosen chromosome. The genetic operators are crucial to identify good parameter values during the evolution steps. The initial population size and the evolution steps have to be defined as well. For the simulation the population size was set to 9 and the evolution steps to 25. This setting does not always provide the best results (see Section 4.4.1), but it is an average benchmark for a trade-off with an acceptable runtime and acceptable volatility of the revenue results. Hence, results can be improved by selecting a bigger population size and a higher number of evolution steps. After the parameters are set, the genetic algorithm has to be executed. The outcome is the achieved revenue and values for the parameters α, β and $\overline{\pi}_h$. These values were determined based on the forecasted demand, the current price setting, the predefined service-resource mapping and the predefined bounds for the control variables. They will be used in the online phase.

In the *online phase* at most one request arrives per timeslot. CBPP analyzes, if the request can be accepted or rejected based on the control variables determined in the offline phase. If the bid price is equal or below the fixed price for the service, the

service will be accepted; otherwise, it will be rejected. After each timeslot (whether a request has arrived or not) the bid price for every resource will be updated according to α, β and $\overline{\pi}_h$.

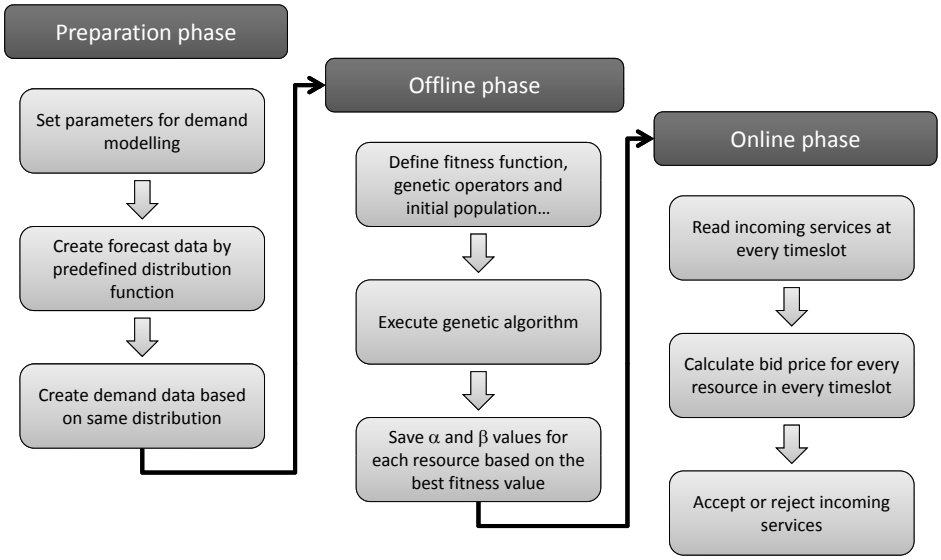


Figure 4.4: The three simulation phase

Klein (2007) has outlined the difficulty of updating bid prices during incoming requests for online algorithms. Thus, DLP or CEC algorithms are not executed every time a request occurs. One reason is the runtime of these algorithms are too long to execute between two requests. Although software like CPLEX[8] can handle very complex scenarios (Bixby, 2002), the execution of a sum for CBPP is always faster than a linear programming problem. The complex calculation of CBPP is done offline, while the update of the bid prices is performed online during the incoming requests. Moreover, requests can arrive within milliseconds, which requires a simple automatically updating algorithm. This thesis analyzes the behavior of CBPP statistically and provides an evaluation of its performance in the Cloud context to answer the RQ 2.4:

How accurately can a simple linear function approximate well known algorithms for bid price calculation without reoptimization between two or more timeslots, taking the assumptions and requirements from Revenue Management in general and from Cloud Computing into account?

The linear optimization function of CBPP was defined in Section 4.3.2. It is an offline algorithm and the relevant parameters are optimized before the time horizon of incoming requests. Then, the bid prices are automatically updated, whenever a request occurs. If online algorithms like DLP or CEC cannot be updated at every incoming request, CBPP can be applied to automatically adapt the current bid price. This thesis analyzes different variations of the online algorithms. At first,

[8]CPLEX (`http://www.cplex.com/`)

both algorithms are updated in every timeslot (DLP-D and CEC-D). Then, a quasi-static version (DLP-S and CEC-S) is analyzed, which only permits to be updated DLP and CEC in certain timeslots (e.g. every second, third or tenth timeslot). The performance of CBPP is measured by the revenue achieved in the simulation. Several aspects have to be analyzed when comparing the online and offline algorithms. Obviously, the revenue of CBPP has to be higher than the online algorithms for different kinds of fare class settings:

Hypothesis 4.1. *The revenue yielded with CBPP is higher than the revenue obtained with CEC-S.*

Hypothesis 4.1 will give a general idea of the performance of CBPP. A more detailed analysis is necessary to understand, if the revenue is differing between the fare class settings. This should not be the case to derive a general statement for Hypothesis 4.1. Otherwise, a sensitivity analysis is necessary to understand, which variables affect the outcome. For example, the choice of the fixed prices can influence the accept/reject decision significantly. However, it not obvious how the selection of a certain price will affect the revenue outcome of CBPP:

Hypothesis 4.2. *The revenue yielded with CBPP does not vary among different prices for the same service-resource mapping.*

Furthermore, the control variables for the genetic algorithm, namely α, β and $\overline{\pi}_h$, have an impact on the accuracy of the genetic algorithm and thus on the outcome. Every control variable requires an upper and a lower bound. These bounds narrow down the possible values of the control variables for the genetic algorithm. If the spread is too small, the optimization function is less flexible, which results in lower revenue. A large spread provides too many options from which the genetic algorithm can select. It reduces the probability of finding the near-optimal solution unless the population size and evolution steps are high enough. Therefore, the impact of the control variables has to be analyzed:

Hypothesis 4.3. *The fine-granular selection of upper bounds does not influence the revenue.*

In Revenue Management the parameter Demand-to-Capacity ratio (DCR) plays an important role in the understanding of how bid price decisions may change, when demand changes (Weatherford and Belobaba, 2002). DCR defines the ratio between the total incoming requests and the total capacity of each resource. A ratio of one means the total demand can be satisfied. $DCR > 1$ represent an excess in demand and in case of $DCR \leq 1$, resources are not used at the end of the time

period. The modeling of DCR in the simulation will influence the revenue. A very high demand (e.g. $DCR > 2$) for certain or all services may result in a minor impact of the bid price control. If there are too many high-fare service requests (vs. available capacity), all of them can be accepted and the highest revenue is yielded. If capacity is still available, other services can be accepted, if it is possible to backfill the available capacity. The bid-price control will reject most of the demand for low-fare services unless some of them are necessary for backfilling. Furthermore, a service provider, who faces such a high demand, tends to shift resources in order to satisfy this demand. Since this thesis analyzes a scarce market, the case of $DCR \leq 1$ is disregarded. The optimal strategy in this case would be to accept all services.

The assumptions of Revenue Management mentioned in the RQ 2.4 refer to the demand modeling applied from the Revenue Management context, since no data profile of current demand data in Clouds is available yet. In the simulation the demand is modeled according to Kimms and Mueller-Bungart (2007).

4.5 Results & Implications

In this section the results from the simulation are presented. The goal of the simulation is to understand the dependency between the parameters, such as the genetic parameters, the base bid price or the number of services and prices, in the simulation and their impact on the revenue. This analysis is required to answer the RQ 2.4. In Section 4.5.1, the simulation results are analyzed statistically to accept or reject the hypotheses outlined in Section 4.4.2. Subsequently, a sensitivity analysis has been carried out to scrutinize the impact of the parameters on the revenue (Section 4.5.2).

4.5.1 Statistical Results

The CBPP algorithm provides a heuristic approach to automatically and efficiently update the bid prices between two or more timeslots in order to maximize the revenue. It comprises three simulation phases to identify good bid prices. In these phases different parameters can influence the outcome of the simulation. Hence, three hypotheses were defined to understand the dependency between the parameters and the simulation outcome (see Section 4.4.2).

In general, the overall performance of CBPP is the most important aspect. Figure 4.5 shows an excerpt of 50 runs of the simulation. Here it can be seen graphically that the performance of DLP-S and CEC-S do not yield as much revenue as the continuous updates variant of their algorithms. CEC-D is quite close to optimum (100%) in every run. DLP-D is not always achieving optimal values and in some

cases also performs worse than its quasi-static version, CEC-S and CBPP. Cooper (2002) already stated that in some cases frequent reoptimization of DLP can decrease the revenue. However, he also emphasized that in general a reoptimization is beneficial. This simulation includes some of these cases. Even CBPP underperforms compared to all other algorithms in some cases. In this specific case, the genetic parameters α and β were not set properly. Bad values were the result of the offline phase, which had an impact on the revenue in the online phase. In Figure 4.5 the forecast and the genetic parameters were created for 10 demand scenarios. The outcome of every algorithm is based on the same demand scenario data and the same forecast data. In the last 10 scenarios (runs 40-49), the genetic parameters performed worse in some scenarios compared to the others, since the incoming demand was differing sometimes quite strongly from the forecast. The goal of CBPP is to automatically update the price, when only CEC-S or DLP-S can be applied. Consequently, the outcome of CBPP will be compared with the quasi-static versions. It is obvious from Figure 4.5 that no algorithm is always superior to all the others. Hence, a statistical analysis is required to validate the preeminence of CBPP over the quasi-static algorithms.

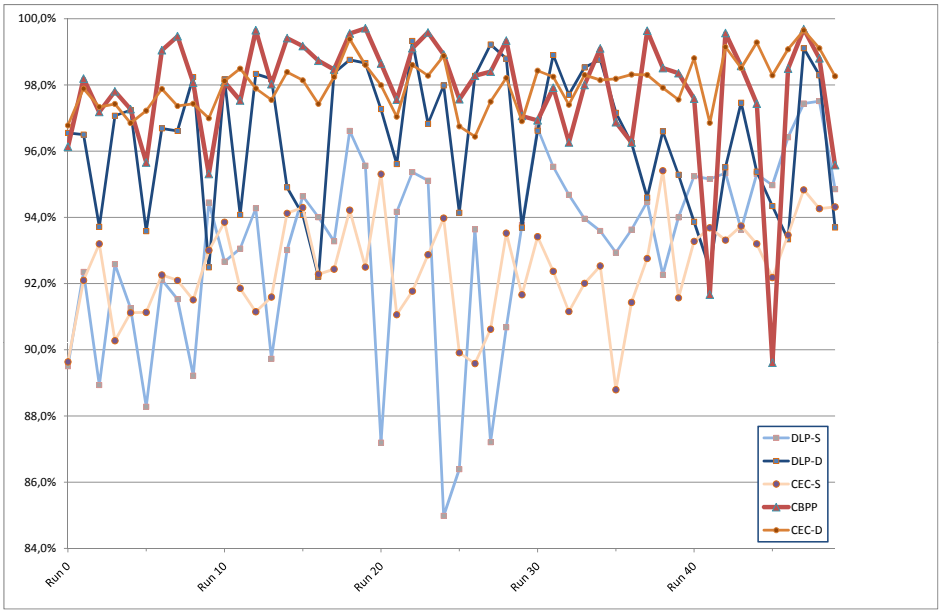


Figure 4.5: Revenue for 50 different runs

Hypothesis 4.1 states that CBPP outperforms CEC-S. According to van Ryzin and Vulcano (2008) an increase of 1% to 3% is already significant in Revenue Management. To corroborate the hypothesis, a one-tailed one sample t-test was executed. CBPP has to show at least 2% higher revenue than CEC-S to gain significant results ($H_0 : \mu \leq 2\%$ and $H_1 : \mu > 2\%$). Different service-resource mappings were chosen to show the significance over several settings. The service-resource mapping was created by a uniform distribution (see Section 4.4.2). Consider that in certain settings,

CBPP can also underperform compared to the static algorithms. It depends on the prices between high and low valued services as well as on the resource usage of each service. Thus, 50 different service-resource mapping with different prices were created randomly. Each mapping consists of 20 different forecast dataset and 100 demand dataset. One forecast dataset was used for five different demand scenarios. The data analysis is based on 5000 revenue outcomes for one comparison (e.g. 3x3 with DCR=1.2), which is sufficient for a sound statistical analysis. Moreover, various DCR settings can influence the revenue outcome. Thus, the different settings for DCR vary between 1.1 and 1.8. (see Section 4.4.2). The DCR is usually set for every resource individually.

Tables 4.10 and 4.11 show the difference between CBPP and CEC-S in yielded revenue. Almost all theses cases are highly significant ($p < 0.001$) or significant ($p < 0.01$). However, in the settings with few resources and services (e.g. 3x3) no significant increase can be proven except for DCR=1.8. In most cases the revenue increases with higher DCR. A DCR above 2 decreases the revenue significantly. Comparing the outcome between DLP-S and CEC-S, DLP-S seems to outperform CEC-S in the small settings. For 3x3 the revenue difference between CBPP and CEC-S is always higher than between CBPP and DLP-S. DLP-S performs well for small settings. Even for DLP-S, there is no significant revenue increase for the 3x3 setting, if CBPP is applied. In general, the small setting provides a worst case scenario. Since the runtime is short, heuristics are not relevant for these cases. Furthermore, such small settings are very uncommon (see Section 2.3.1). For larger settings CBPP can increase the revenue significantly.

Table 4.10: Results for Hypothesis 4.1: Revenue difference between CBPP and CEC-S (* denotes significance at the level of $p = 0.05$, ** at $p = 0.01$, and *** at $p = 0.001$.)

	DCR=1.1	DCR=1.2	DCR=1.4	DCR=1.6	DCR=1.8
3x3	1.7%	1.6%	2.2%	2.2%	3.5%***
4x4	2.8%**	4.0%***	5.2%***	4.7%***	5.2%***
4x5	2.9%***	3.5%***	6.4%***	5.5%***	6.4%***
4x7	2.5%	4.5%***	6.5%***	6.6%***	6.9%***
4x10	3.1%***	3.8%***	4.6%***	4.8%***	5.2%***
6x8	3.0%***	4.2%***	5.6%***	6.3%***	6.6%***
10x10	3.2%***	3.7%***	4.9%***	4.6%***	5.1%***

The analysis above focuses on the general outcome of CBPP. Different settings for DCR, for genetic parameters and for service-resource mapping were chosen. The

Table 4.11: Results for Hypothesis 4.1: Revenue difference between CBPP and DLP-S (* denotes significance at the level of $p = 0.05$, ** at $p = 0.01$, and *** at $p = 0.001$.)

	DCR=1.1	DCR=1.2	DCR=1.4	DCR=1.6	DCR=1.8
3x3	0.8%	1.2%	1.4%	0.9%	2.1%
4x4	2.8%**	4.4%***	5.0%***	6.5%***	6.3%***
4x5	2.2%	3.8%***	5.4%***	5.9%***	6.9%***
4x7	3.3%***	4.7%***	5.8%***	6.3%***	7.3%***
4x10	2.7%**	3.6%***	4.2%***	5.1%***	6.8%***
6x8	2.9%***	4.6%***	5.2%***	5.3%***	6.4%***
10x10	3.0%***	5.1%***	7.3%***	8.6%***	7.9%***

dependency between the various parameters of CBPP have to be scrutinized. For the following hypotheses an ANOVA[9] was conducted.

Hypothesis 4.2 is a negated phrase. It is assumed that small changes in the price of a service do not have an impact on the revenue (alternative hypothesis). In other words, applying CBPP to different price setting will lead to similar results in all runs. Tables 4.12 and 4.13 show the results for a 4x4 setting, when only small increase or decrease of one specific price changed the outcome of CBPP and CEC-S or DLP-S, respectively. There is a significant difference, since certain services are accepted by one service more than by others. If prices change, CBPP seems not always to outperform other algorithms, because in some cases the improvement is less than 1%. For example, price changes for $i = 4$ does not significantly differ from the original setting. However, a marginal difference in the price of $i = 2$ had a big impact on the outcome of CBPP, while the price for CEC-S was quite stable compared to the original setting. Consequently, prices for different Cloud services do not have the same impact on the outcome, but they can have an impact. Thus, a sensitivity analysis is required to understand the dependency.

As outlined in Section 4.4.1 the parameters of CBPP influence the revenue. In particular, the upper bounds are in focus of this analysis after presenting the impact of the population size and the evolution steps. Again, the hypothesis is formulated in a negated way. The results in Table 4.14 show that at least for one case there is a significant deviation in the revenue over all runs. Especially, for bounds with low values, there is a high variance in the outcome and therefore the probability of receiving near-optimal solution decreases. The objective function is not flexible enough to identify good revenue results. Hence, in this case a sensitivity analysis

[9] A Levene test proved the homoscedasticity of the statistical population.

Table 4.12: Results for Hypothesis 4.2: small price changes of a specific service affect revenue difference between CBPP and CEC-S

	i=1	i=2	i=3
i=1	X	X	X
i=2	p<0.01**	X	X
i=3	p<0.01**	p=0.43	X
i=4	p=0.13	p=0.02*	p=0.4

Table 4.13: Results for Hypothesis 4.2: small price changes of a specific service affect revenue difference between CBPP and DLP-S

	j=1	j=2	j=3
j=1	X	X	X
j=2	p<0.01**	X	X
j=3	p<0.01**	p=0.27	X
j=4	p=0.07*	p<0.01**	p=0.2

is necessary as well to understand why a certain bounds should be preferred. The bounds should be neither too low nor too high. Low bounds will not enable a correct adaptation of α and β. Too high bounds with a low population size result in scattered values for these parameters and consequently prevent the identification of near-optimal solutions. The results in table 4.14 are created for a 4x4 setting with DCR=1.4 and a population size of 9. Similar results were achieved for other fare classes and settings as well.

Table 4.14: Results for Hypothesis 4.3: Upper bounds affect revenue

	$\alpha_U = 2$, $\beta_U = 2$	$\alpha_U = 3$, $\beta_U = 3$	$\alpha_U = 5$, $\beta_U = 5$
$\alpha_U = 3$, $\beta_U = 3$	p<0.001***	X	X
$\alpha_U = 5$, $\beta_U = 5$	p<0.01**	p=0.06*	X
$\alpha_U = 8$, $\beta_U = 8$	p<0.01**	p=0.08*	p=0.12

The results of Hypotheses 4.2 and 4.3 emphasize the necessity for further analysis to determine how these variables affect the revenue. Thus, a sensitivity analysis is presented in the following section to explore the dependency and to elicit the cases, where CBPP performs worse than other algorithms.

4.5.2 Sensitivity Analysis

The sensitivity analysis is necessary to understand the impact of different parameters in the simulation setting. A common and reasonable approach is to fix all parameters except the examined one. The main parameter is the bid price. The bid price is calculated by the algorithms and thus influenced by many other parameters. The goal is to identify the best bid prices to approximate the optimum accurately.

The genetic upper bounds define the search space to receive good values for α_h and β_h of every resource h. These parameters have been analyzed by selecting seven representative bound settings for each service-resource mapping (Table B.1 in the appendix). The average revenue of CBPP, the standard deviation over all simulation runs and the revenue difference to CEC-S are used as a metric to understand the dependency. For the price range chosen for this simulation (see Section 4.4.2), a bound for $\alpha_U = \beta_U = 10$ yields the highest revenue with the lowest deviation. As assumed low bounds do not have a high standard deviation, but the revenue is lower than in the best case. High values for α_h and β_h can decrease the revenue on average compared to $\alpha_U = \beta_U = 10$, since the standard deviation is higher than in the best case. These statements are valid for all the tested settings. In general, there is a difference of about 1% in revenue between the worst case and the best case. According to van Ryzin and Vulcano (2008), this is already a significant delta. The volatility of the revenue can increase up to 40% between the best case (here usually $\alpha_U = \beta_U = 10$) and the worst case ($\alpha_U = \beta_U = 20$). A countermeasure for avoiding volatility for high bounds is to increase the population size and the evolution steps to achieve better values on average. Then, the probability of finding appropriate values is higher, but the runtime increases as well (see Section 4.4.1).

Providers in some domains change prices frequently. They are forced to do this due to internal (e.g. increasing cost) or external (e.g. competition) factors. However, price variation has a great impact on the decision policy and the revenue. It is important to understand the dependency between prices and other factors. In the previous section, it was shown that small price changes already lead to significant revenue gain or loss. A first step is to incrementally vary the prices for each service and to monitor when the revenue significantly drops. The scenario presented in table 4.15 builds the basis for the analysis by fixing the service-resource mapping and defining certain prices for the services. The data was created randomly according to

Table 4.15: Service-resource mapping and price in the basic scenario

	i=1	i=2	i=3	i=4
h=1	2	4	8	6
h=2	7	5	4	4
h=3	4	9	4	3
h=4	4	3	4	9
Price	16.2	18.9	20.5	33

Section 4.4.2. One of these prices is slightly increased or decreased, while the others are kept fixed. This approach enables to explicitly analyze one price change for one service at a time and isolates other influential parameters.

Table 4.16 gives an overview on the price changes. Note that this table represents 28 variations, since only one price is changed at a time to determine the effects of changes accurately. The prices are changed in small steps to understand the dependency of the price setting and the prices of each service do not overlap with the other services. The critical cases are shown in bold in the table. In these cases, there is a significant jump in the revenue between two selected variation. For example, the price change from 16 (variation 21) to 17 (variation 25) drops the revenue by 4.5%. With a low price for service $i = 1$ the revenue is not changing significantly. A lower price than the randomly chosen price 16.2 does not have an impact on the revenue difference between CBPP and CEC-S[10]. However, a higher price decreases the value by more than 4%. Here, the price for service $i = 1$ was increased to 17 (variation 25).

Comparing the outcome from CBPP and CEC-S the revenue of CBPP is similar in both cases. Variation 25 leads to a slightly higher outcome by 1% due to the higher price of service $i = 1$. CEC-S shows a significant increase in the revenue (in some settings up to 8%). Both algorithms are based on forecasts. While CBPP is relatively robust against forecast errors, CEC-S had problems to compensate this loss. In variation 21, 10% fewer requests for service $i = 4$ arrived than expected. Since, CEC-S has reserved capacity for the expensive service $i = 4$, other service requests were rejected. Especially, in the beginning of the period, more service $i = 1$ were rejected in variation 21 than in variation 25. A total bid price of 16.4 at the beginning resulted in declining service $i = 1$ requests during the first quarter of the period, when most of the requests for service $i = 1$ arrive. The higher price for service $i = 1$ in variation 25 caused better bid prices and thus a better acceptance

[10]The comparison with DLP-S has been disregarded, since it showed a similar behavior to CEC-S for the first analysis. The analysis in the remaining part of this section is related to the CBPP algorithm and does not depend on the performance of CEC-S or DLP-S.

Table 4.16: Price variations for randomly selected scenarios

	$i=1$		$i=2$		$i=3$		$i=4$	
basic price	**16.2**	**rev diff**	**18.9**	**rev diff**	**20.5**	**rev diff**	**33**	**rev diff**
variation 1-4	11	5.4%	16.5	-0.2%	**21**	**4.9%**	23	4.5%
variation 5-8	12	5.3%	17	0.3%	**22**	**1.5%**	24	4.9%
variation 9-12	13	4.5%	17.5	0.4%	23	1%	35	4.7%
variation 13-16	14	4.5%	**18**	**0.4%**	24	0.6%	36	4.2%
variation 17-20	15	4.6%	**18.5**	**5.0%**	25	0.9%	37	4.4%
variation 21-24	**16**	**5.1%**	19.5	4.9%	26	0.6%	38	4.3%
variation 25-28	**17**	**0.6%**	20	4.8%	27	0.7%	39	4.3%

rate. While in variation 21 the scarce resource was $h = 1$, in variation 25 resource $h = 4$ was the bottleneck (Table 4.17). Resource $h = 4$ mainly used by service $i = 4$ and resource $h = 1$ by service $i = 3$, which reflects the lower acceptance rate of service $i = 3$ in order to accept more service $i = 4$ requests. Both services compete for resource $h = 1$. Hence, this resource is rarely available in both cases. CBPP generally accepted less service $i = 4$ requests and focused more on service $i = 1$ and $i = 2$. Although the revenue discrepancy was not significant in variation 25, CBPP yielded a higher revenue in both cases than CEC-S[11]. For variation 3 and 7 similar interpretations apply.

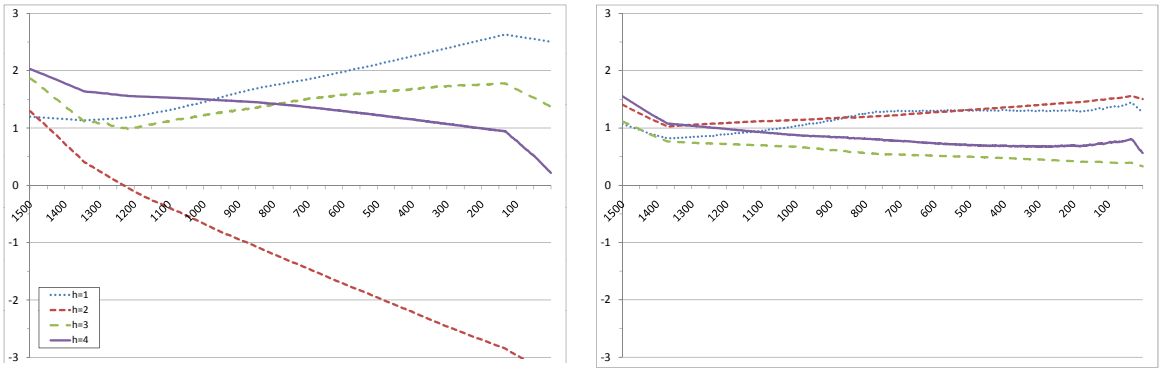
Table 4.17: Capacity occupation for specific scenarios in variation 21 and 25 at the end of the period

	Variation 21	Variation 25
$h = 1$	99.9%	99.7%
$h = 2$	75.5%	84.6%
$h = 3$	85.4%	81.1%
$h = 4$	91.2%	99.9%
average	88.8%	91.3%

The case of service $i = 2$ outlines another important aspect of CBPP. Variation 14 and 18 are distinguished by the prices for service $i = 2$ with a discrepancy of 0.5. This small change leads to an average decrease of revenue of more than 4% (Table 4.16).

[11] Although the analyzed scenarios for comparing both cases were selected randomly, it was considered to have similar request scenarios and forecasts, if available, to isolate the scenario from other effects.

Analyzing several runs, a typical pattern recurs. Figure 4.6 depicts the development of the bid prices for each resource in the 4x4 setting for CBPP. The resource $h = 2$ plays a crucial role for the outcome of CBPP. It is the most important resource for service $i = 1$ with a resource consumption of 7 (Table 4.15). Although service $i = 2$ needs more capacity from resource $h = 3$, it still requires the second highest amount of resource $h = 2$. The price of service $i = 2$ in variation 14 is closer to the price of service $i = 1$ than in variation 18. To accept service $i = 2$ the bid prices on the resource level have to be adapted appropriately. In variation 18 the bid prices are set almost equally and behave in a similar way over time (Figure 4.6b). Variation 14 shows a difference in the curve progression. The bid price for resource $h = 2$ decreases continuously, since service $i = 2$ needs to be accepted, while other bid prices increase (Figure 4.6a). This indicates a low α and a high β for resource $h = 2$, while the others have a slightly higher α than β. Hence, more service $i = 2$ requests can be accepted. However, due to the proximity of the prices for service $i = 1$ and $i = 2$, more service $i = 1$ requests are accepted in variation 14 than in variation 18. Consequently, less capacity is available for service $i = 3$ and $i = 4$. A decrease in the price for service $i = 1$ from 16.2 to 15 and keeping the other prices as in variation 14 increases the revenue by 2%. Thus, prices should not be set too close to each other in order to improve the ability of the genetic algorithm to find appropriate bid prices for the resources. This is a disadvantage of the linear dependency of the bid prices. They cannot be adapted flexibly, since they depend on the incoming request and the decision of the previous time slots.



(a) Resource bid prices of variation 14 with price for $i = 2$: 18 **(b)** Resource bid prices of variation 18 with price for $i = 2$: 18.5

Figure 4.6: Bid price development over time for each resource (4x4 setting).

4.5.3 Implication

Providers can implement CBPP to manage revenue and resource utilization more efficiently, when bid prices cannot be updated between two or more incoming requests. Usually, the service-resource mapping is relatively fixed for their settings.

They have a portfolio of standard services. Thus, changes in the service-resource mapping is only useful, when providers often renew their portfolio by introducing new innovative services. The results above have proved that CBPP can outperform standard algorithms by up to 6% on average (RQ 2.4). However, it highly depends on the applied scenario. First of all, the provider has to be aware of the dependencies between services and resources. They must be able to technically map services to the consumed resources. The service-resource mapping has a great impact on the revenue. There is no general policy for which service-resource mapping CBPP performs best, since it depends on the price and the incoming demand.

The analysis of runtime in Section 4.4.1 provides a worst case scenario, where the algorithm is executed on a single machine. Genetic algorithms are capable of generating parallel threads to improve the runtime significantly. Hence, the runtime of CBPP will be much better, if the algorithm is executed in a cluster or Cloud. Furthermore, the revenue can be improved by selecting a higher population size and evolution steps, since the runtime can be reduced due to parallelization.

Cloud service prices are set by providers. They have to consider the current market prices and the willingness to pay of the consumers for their services. Changes in prices will change the bid price for every resource as well and thus influence the accept/reject decision of the provider. In some cases, CEC-S can outperform CBPP, if the prices are set too close. Providers can adjust the prices or the resource consumption of the services to avoid this issue. Service prices can also help to calculate the optimal reservation prices, if services are auctioned. For example, Amazon has three different pricing models. By defining the service price internally as reservation price, Amazon can calculate how many services may be offered as Spot instances, Reservation instances or On-Demand instances.

Demand is often a reaction on the offered Cloud services, the necessity for these services and their prices. In the simulation, it was assumed that the demand follows a Non Homogeneous Poisson Process (NHPP) and expensive services are likely to be requested later in time than inexpensive services. The provider has to analyze the incoming requests and model the forecasts accurately. Since service-resource mapping often cannot be changed instantly and demand can only be affected indirectly, other parameters play a more important role. Capacity for Cloud services is usually relatively fixed (see discussion in Section 2.3.1). Providers can virtually limit the capacity for certain services to guarantee a promised service level. This approach allows them to calculate the risk of accepting a predefined amount of service and still fulfill the SLA. Obviously, the virtual limit can be extended based on previous experience or if providers are risk takers.
The DCR depends on the demand and the capacity. By setting different virtual lim-

its, this parameter can limit the resource usage to make it scarce, which will have an impact on the service and thus its price, because accept/reject policies change.

Part III

Finale

Chapter 5

Conclusion and Outlook

In a Long Tail economy, it's more expensive to evaluate than to release.

[Chris Anderson, 2004]

The thesis is divided into two self-contained chapters with their contributions. The results in these chapters are summarized in Section 5.1 and their interrelation are explained to understand the integrated view and their impact in general (Section 5.2). There are still questions and opportunities left for future work. These are outlined in Section 5.3.

5.1 Summary of Contributions

Cloud Computing is a paradigm of providing infrastructure, platform and software as services over the Internet. Instant service request from consumers can be satisfied on-demand and the service are highly scalable. The provision of such services challenge Cloud service providers to manage their computing resources efficiently and also to price their services appropriately. Both aspects are best handled by Revenue Management methods and tools.

The objective of this thesis is to analyze the interrelation of Revenue Management and Cloud Computing. At first, the properties of Revenue Management methods and Cloud Computing were outlined and compared with each other (Chapter 2). Eight characteristics are relevant for Revenue Management. The offered Cloud services possess inherent Revenue Management characteristics such as perishability and inflexibility (Section 2.3.1). Cloud service providers use IT hardware to provide these services. On the one hand, they have to manage the utilization of these resources and, on the other hand, they have to fulfill the promised service levels. In case of scarce resources and uncertain demand, overbooking strategies

enable the increase of the resource utilization. From the consumers' perspective, Cloud Computing promises high flexibility by outsourcing IT resources and their maintenance. This, however can lead to volatile demand and uncertain consumer behavior for a Cloud service provider. Furthermore, consumers have different kinds of demands. Some may prefer a guaranteed service while others focus on low-price services. Hence, the heterogenic consumers and the problem of uncertainty can be addressed by introducing advance reservation or price segmentation for Cloud services.

All these characteristics fit to the Cloud Computing paradigm, however, consumers' preferences have not been analyzed in the Cloud context from a Revenue Management perspective. Thus, a survey was conducted in this thesis to identify how consumers would react to the introduction of typical Revenue Management methods like advance reservation (Chapter 3). This survey comprised of three parts. At first, general questions like frequency of Cloud service usage were asked to classify the participants into different user categories. In a second step, more specific questions were necessary to analyze whether Revenue Management methods are accepted by consumers. The theoretical motivation for these questions stems from the Revenue Management literature about customer choice theory. The relevant papers were categorized to identify similar research questions. Then, important questions were derived and embedded in the survey. The third part analyzed the consumer preferences via a conjoint analysis. This research method enables the derivation of the utility of a service for every customer from the evaluation of the customers' perception for different combinations of the predefined service attributes.

Three research questions have been answered by this survey. Initially, the applicability of certain provider policies were examined (*Research Question 2.1*). Consumers were asked how they would react to advance reservation and price discrimination. Furthermore, the dependency on their usage frequency of IaaS was analyzed. The answers were quite positive. Consumers are open to price discrimination, when the offered services have different characteristics. Even advance reservation was appreciated by the survey participants. Thus, Revenue Management methods are applicable to Cloud services.
Price discrimination plays an important role in Revenue Management, since customer segmentation and different service characteristics enable the increase of revenue through different prices. Different survey questions were analyzed via a Chi-square test for the *Research Question 2.2*. Consumers' profit from different prices according to the service levels. They are ready to buy services with lower or no guaranteed service level for a lower price. However, there were no significant results between price discrimination and booking in advance. Hence, it is not obvious whether a service with an advance reservation option should have a different price to an on-demand service. Maybe consumers would buy both services for the

same price or even pay a higher price for the advance reservation. For example, the provider for rendering software RenderRocket asks a higher price for advance reservation services with a guaranteed service level.
The conjoint analysis revealed the preferences of the consumers to answer *Research Question 2.3*, which attributes are most important for the consumers. The operating system is an essential part in the IaaS offer when consumers decide between various offers. Both Windows and Linux should be available. Linux is more attractive than Windows. The operating system is followed by the price. Obviously, a lower price is preferred to a higher price. Furthermore, the consumers benefit from a good support level with direct contact to the help desk via phone. This support makes them feel that they have more control by directly influencing the recovery process in the case of failure. The survey from Avanade also substantiates the fact that loss of control is the biggest fear of the Cloud users (Leipold et al., 2009).
An IaaS provider will increase revenue by introducing price discrimination and various service offers for heterogeneous customers. Operating systems and price have a great impact on the provider selection process of the consumer. Thus, offering various Linux or Windows versions (or even other operating systems) can result in a higher market share. However, a provider should consider other additional services like phone support as well to distinguish the offered services from the competitors' offers.

After knowing the consumer preferences, the provider has to design the Cloud services accordingly. These services are offered to a variety of consumers with certain service level conditions. To fulfill these service levels, systems are often run redundantly. The provider faces the problem of utilizing the hardware resources and to meet the SLA. In case of scarce resources the provider has to decide when to accept or reject certain service requests. In another way of interpretation, he can compare different scenarios with different prices assuming to have almost perfect knowledge about the demand. When requests for these Cloud services appear rapidly, the provider has to decide in an automated way when to accept or reject the requests.
Chapter 4 suggests an automated update of bid prices called Customized Bid-Price Policy (CBPP) as a heuristic to efficiently make these decisions. It performs better than well-known algorithms, if an update of the bid prices in not possible after every incoming request or in every timeslot, respectively. While the alternative algorithms are calculated online, CBPP estimates the price decision for every timeslot offline and then uses an additive approach to determine the bid price during the incoming requests. The revenue of CBPP is, in most of the scenarios, better than standard algorithm with a less frequent update. It can increase the revenue by up to 20%, but it can also perform 10% worse than other algorithms. Several parameters affect the revenue outcome, which have not been scrutinized in the literature before.

The impact of price changes were analyzed to understand its effect on the bid price. A small difference between the prices for each service can lead to substantially bad results for CBPP compared to the quasi-static CEC-S algorithm. However, the results for CBPP are stable, while price changes may influence the CEC-S outcome. Small price differences are less error-prone and thus increase the revenue of CEC-S. One assumption is to have a Non Homogeneous Poisson Process (NHPP) to model the demand. Another aspect is the Demand-to-Capacity ratio (DCR) value, which is assumed to be between one and two. The service-resource mapping was modeled by a random distribution. It can be concluded that an offline algorithm with an additive function for the online update can increase the revenue under the above mentioned conditions (*Research Question* 2.4).

5.2 Integrated View of the Results

Revenue Management encompasses many aspects of a service providers' decision process how and when to allocate a service to a certain consumer. This thesis covered two aspects of the entire process discussed in Chapter 3 and 4. Cloud services have to be consumed electronically over a network. Cloud consumers can accept a posted price offer or participate in an auction. Some companies like 3Tera allow only a direct negotiation without a posted-price offer. Consumers have different demand characteristics and ask for various additional services from a simple phone support to complex pre-configured several virtual machines with workflow management application. By identifying the consumers' preferences, a provider can efficiently design basic and complex services to satisfy the demand. The preference analysis can be done via survey (Chapter 3) or by analyzing historical customer data (if available). Historical data and current demand can also help to accurately forecast the demand for certain services at certain points in time. An accurate forecast will lead to higher revenue, since the (hardware) resources can be utilized efficiently. Moreover, both consumer preferences and forecasting are necessary to provide the price optimization engine with the right information at the right time. Current prices are based on the market situation, the demand, the resource availability and strategic decisions of the provider. For example, it is common practice in the travel industries that the prices increase over time, while in the field of fashion, the retailing prices decrease (Su, 2007). Amazon.com is currently the most dominant Cloud service provider and they seem to be open to various pricing policies. Their portfolio comprises posted-price offers, subscription models and dynamic pricing. It is likely that Cloud services offers will have many facets.

From a provider's perspective, these aspects have to be taken into consideration in order to efficiently price the offered services. Capacity control algorithms like

CBPP enable to determine different demand scenarios and calculate the appropriate price according to the demand. This, however, assumes that each provider has profound knowledge about the price sensitivity of the customers, i.e. the provider knows how the customers will react, if prices are increased or decreased by a certain percentage. Often, the provider has imperfect information and thus has to approximate the optimal decision supported by historical data and experience. Revenue Management provide tools and methods to achieve a near optimum solution. Though, it is often based on various assumptions (see Chapter 2). A Cloud service provider has to conceive these assumptions to apply these methods efficiently in practice.

5.3 Future Work

The current developments in the market for Cloud Computing services have not led to a mature stage yet. Technical and economical challenges remain to be solved. Several surveys have already emphasized the fear of Cloud customers to loose control of the hardware management and thus to be unable to react quickly in emergencies. Even the data management and the access to the data is convoluted, since a consumer does not know who has access to the data. This is in line with the security concerns, when data transfer and data storage are not 100% secured by encryption methods. Currently, there is no guarantee of who is responsible, if data security is violated. SLAs often do not take this aspect into consideration. Gens (2008) identified that security, performance and availability are the biggest challenges. Developing reliable security mechanisms for Cloud services will help to reduce the fear of the consumers. This approach can be enhanced by establishing international laws to enforce minimum requirements for Cloud service SLAs.
Comments from the respondents of the conducted survey also corroborate what other surveys pointed out: changing providers is very complex, and missing standards make it hard to transfer operations from one provider to the other. Sunk cost at one provider lead to a lock-in effect for the consumer (Varian et al., 2004). Although prices influence the first choice decision, migrating operations to another provider for a better price is only an option for non-critical operations and significantly lower prices (at least 25%) according to the conducted survey in this thesis. The initiative "Open Cloud Manifesto" comprises currently over 175 companies including major companies like IBM and SAP amongst others. Though, Amazon and Google as the current big players in the Cloud are, however, missing. It indicates that these companies, which are already successfully offering Cloud services, are not interested in abolishing the lock-in effect. Consumers will definitely benefit from such an initiative, if they agree on standards. One step towards lowering

the switching costs is enabled by the company CloudSwitch[1]. They offer a service to migrate virtual machines created with VMWare[2] to Amazon EC2. This service saves the time of configuring an EC2 instance from scratch.
The tools from Revenue Management described in this thesis are only applicable, if a service-resource mapping is possible. Providers have to at least estimate the resource consumption of each service. Virtualization technologies help to allocate a certain amount of resource for certain virtual instance on the IaaS layer. An in-depth control of the service consumption will help to manage the hardware resources efficiently, which has to be analyzed technically.

From an economic point of view, price determination and Cloud service design need a profound knowledge about the consumer preferences. The conducted survey was a first step towards understanding these preferences. However, each parameter analyzed in this survey has to be examined more closely. Detailed knowledge on how changes in the availability rate may have an impact on the customer's choice is valuable information for the providers. They can calculate their revenue better by considering the outages, the willingness of a customer to take a certain outage rate into account and to derive the penalty rates for their SLA violation. Cloud Computing is currently in an evolving stage. The survey in this thesis provides a snapshot from the consumer preferences in 2009. More continuous surveys are necessary to evaluate the historical and dynamical development of the service offers. Furthermore, finite price changes over time as proposed by Bitran and Mondschein (1997) are interesting approaches to understand how often prices are allowed to vary. Since the Cloud Computing market is still in its infancy, there is no common price structure or pricing policy. Pay-as-you-go, subscription model and capacity pricing are common and widely used on all layers (infrastructure, platform and software as a service). Another interesting aspect is to analyze consumer preferences on other layers beside the infrastructure level as well. A first approach on SaaS was done by Köhler et al. (2010).

The proposed heuristic CBPP is based on an additive function to calculate the bid prices. The disadvantage of the additive function is its dependency on the bid prices in each timeslot, which makes the function less flexible. Perhaps non-linear function may approximate the bid prices more accurately. For example, some resources may have a higher impact on the service value than other resources. An adaptive non-linear function can put a heavier weight on selected resources to approximate the bid prices in a better way.
Moreover, this approach defines the time horizon as finite. The provider limits his view to a certain time period. If CBPP is applied for reservation and the usage will start at the end of the time horizon, this approach is appropriate. An interesting

[1] http://www.cloudswitch.com
[2] http://www.vmware.com

extension would be a continuous usage case. CBPP can be applied to the continuous case with certain limitation, i.e. the customer instantly using this service will use it until the end of the time horizon. Though, the used resources will not be released for further offers. An extension would take the duration of usage of a Cloud service into account, which provides a more realistic scenario.

Part IV

Appendix

Appendix A

Customer Choice Survey

A.1 Questionnaire

Figure A.1: Websurvey Questionnaire Part 1

Identifying Customers' Preferences in Cloud Computing

0% 100%

Valuation of Cloud Computing Services

The following questions focus on the valuation of cloud computing services.

*__Q1:__ What price is appropriate for using the following standard computing instance for **one hour**?

We define a standard computing instance as follows:
2 standard server CPUs (x86) with 3 GHz
24 - 32 GB of memory
500 GB of instance storage
64-bit platform

Only numbers may be entered in this field

Please enter your **true valuation** in US-$!

*__Q2:__ What price is appropriate for using the following standard computing instance for **one month**?

We define a standard computing instance as follows:
2 standard server CPUs (x86) with 3 GHz
24 - 32 GB of memory
500 GB of instance storage
64-bit platform

Only numbers may be entered in this field

Please enter your **true valuation** in US-$!

*__Q3:__ Do you prefer to customize services according your needs or do you prefer services offering bundles?

A customized service allows you to pick exactly the characteristics you need but the configuration is more complex (e.g. Amazon Web Services). Bundles have the advantage of simpler configuration but you may purchase service characteristics you do not need (e.g. flat rate).

Choose one of the following answers

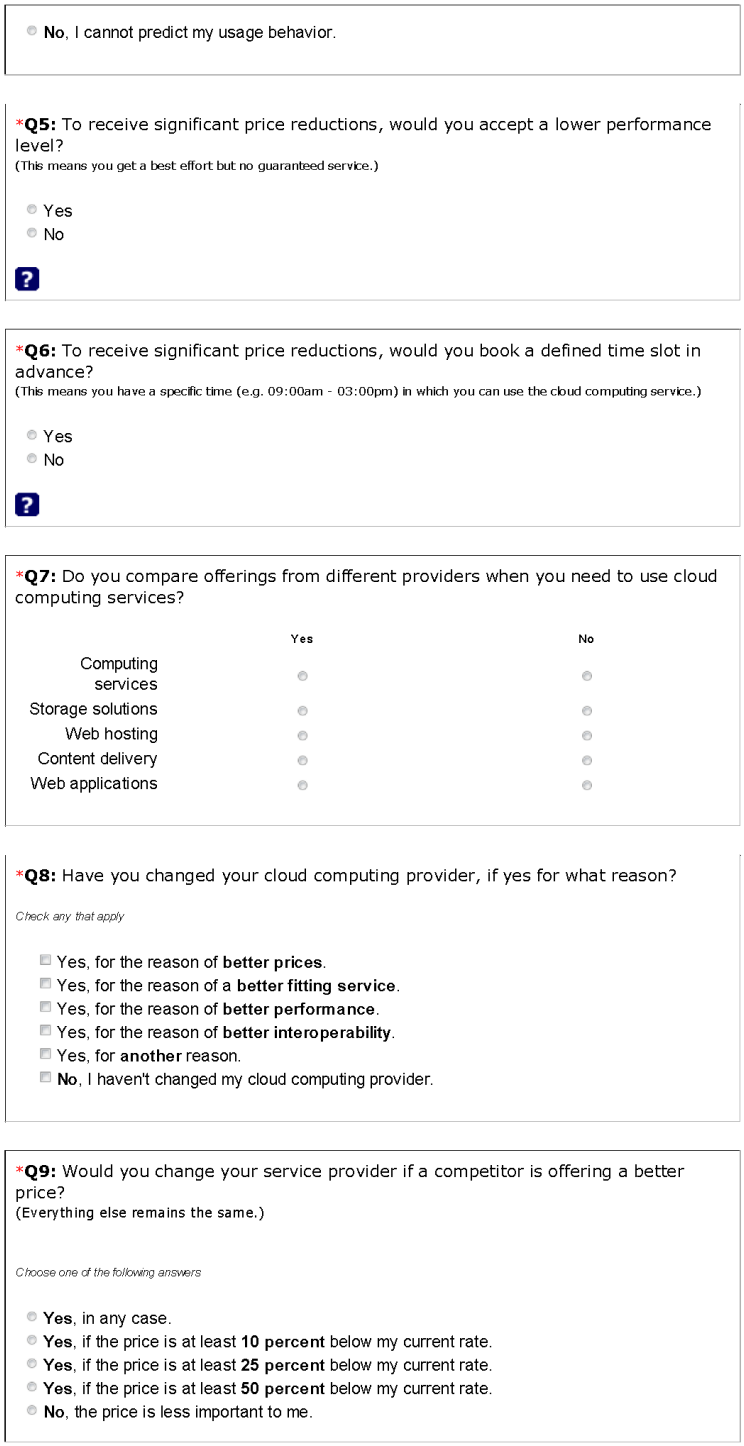
- I prefer to **customize** products and services.
- I prefer products and services offered as **bundles**.

*__Q4:__ Can you predict your usage behavior for cloud computing services?

Choose one of the following answers

- **Yes**, I can predict it at least **half a day** before usage.
- **Yes**, I can predict it at least **one day** before usage.
- **Yes**, I can predict it at least **two days** before usage.
- **Yes**, I can predict it at least **one week** before usage.

Figure A.2: Websurvey Questionnaire Part 2



○ **No**, I cannot predict my usage behavior.

*Q5: To receive significant price reductions, would you accept a lower performance level?
(This means you get a best effort but no guaranteed service.)

○ Yes
○ No

*Q6: To receive significant price reductions, would you book a defined time slot in advance?
(This means you have a specific time (e.g. 09:00am - 03:00pm) in which you can use the cloud computing service.)

○ Yes
○ No

*Q7: Do you compare offerings from different providers when you need to use cloud computing services?

	Yes	No
Computing services	○	○
Storage solutions	○	○
Web hosting	○	○
Content delivery	○	○
Web applications	○	○

*Q8: Have you changed your cloud computing provider, if yes for what reason?

Check any that apply

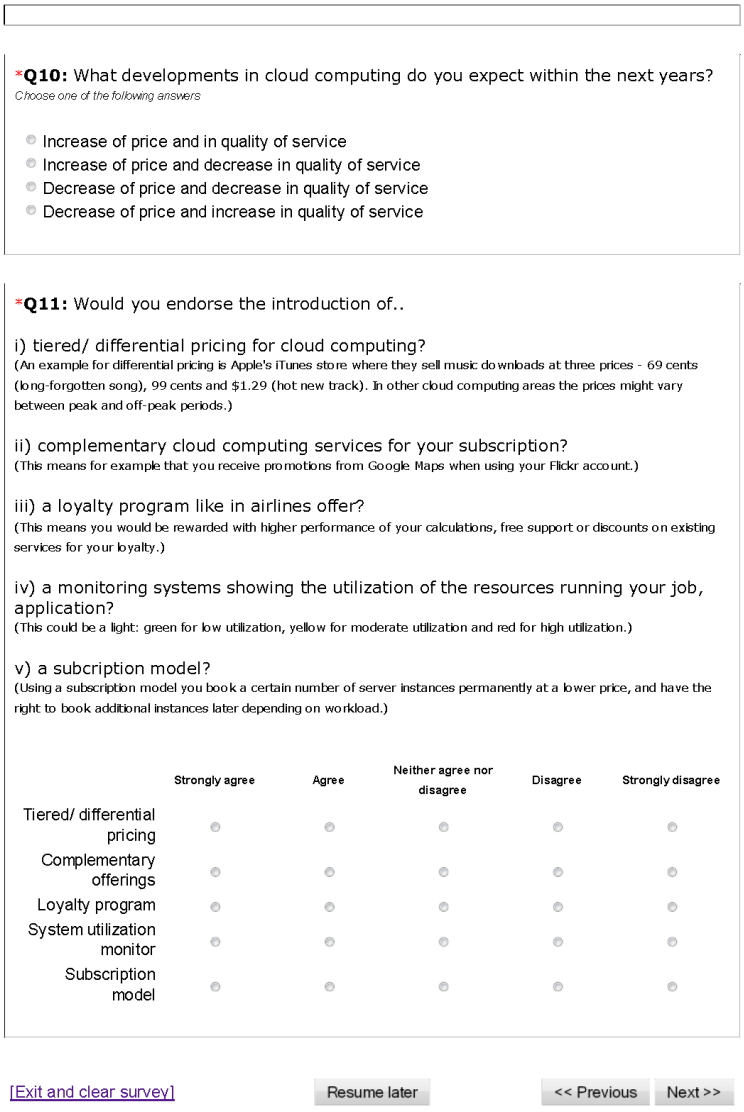
☐ Yes, for the reason of **better prices**.
☐ Yes, for the reason of a **better fitting service**.
☐ Yes, for the reason of **better performance**.
☐ Yes, for the reason of **better interoperability**.
☐ Yes, for **another** reason.
☐ **No**, I haven't changed my cloud computing provider.

*Q9: Would you change your service provider if a competitor is offering a better price?
(Everything else remains the same.)

Choose one of the following answers

○ **Yes**, in any case.
○ **Yes**, if the price is at least **10 percent** below my current rate.
○ **Yes**, if the price is at least **25 percent** below my current rate.
○ **Yes**, if the price is at least **50 percent** below my current rate.
○ **No**, the price is less important to me.

Figure A.3: Websurvey Questionnaire Part 3



*Q10: What developments in cloud computing do you expect within the next years?
Choose one of the following answers

- Increase of price and in quality of service
- Increase of price and decrease in quality of service
- Decrease of price and decrease in quality of service
- Decrease of price and increase in quality of service

*Q11: Would you endorse the introduction of..

i) tiered/ differential pricing for cloud computing?
(An example for differential pricing is Apple's iTunes store where they sell music downloads at three prices - 69 cents (long-forgotten song), 99 cents and $1.29 (hot new track). In other cloud computing areas the prices might vary between peak and off-peak periods.)

ii) complementary cloud computing services for your subscription?
(This means for example that you receive promotions from Google Maps when using your Flickr account.)

iii) a loyalty program like in airlines offer?
(This means you would be rewarded with higher performance of your calculations, free support or discounts on existing services for your loyalty.)

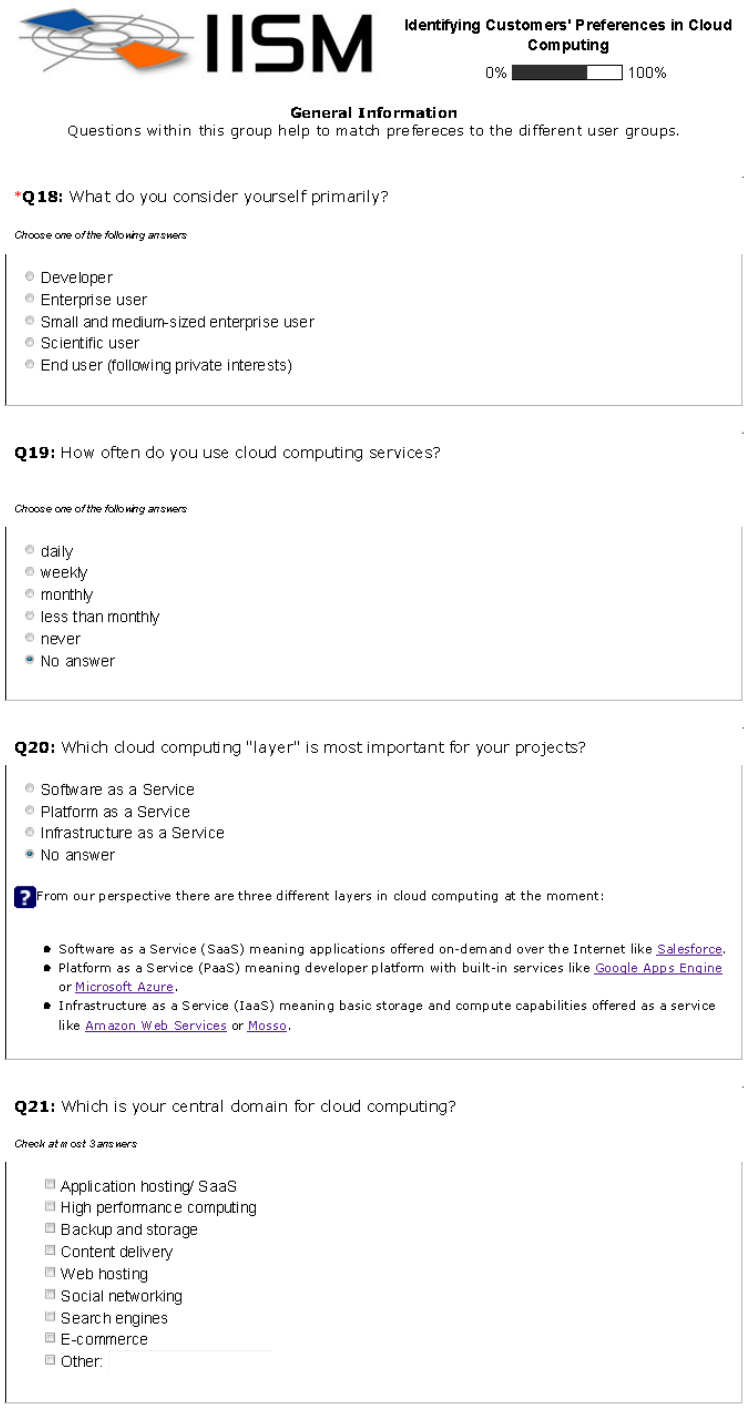
iv) a monitoring systems showing the utilization of the resources running your job, application?
(This could be a light: green for low utilization, yellow for moderate utilization and red for high utilization.)

v) a subcription model?
(Using a subscription model you book a certain number of server instances permanently at a lower price, and have the right to book additional instances later depending on workload.)

	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
Tiered/ differential pricing					
Complementary offerings					
Loyalty program					
System utilization monitor					
Subscription model					

[Exit and clear survey] Resume later << Previous Next >>

Figure A.4: Websurvey Questionnaire Part 4



IISM

Identifying Customers' Preferences in Cloud Computing

0% 100%

General Information

Questions within this group help to match prefereces to the different user groups.

*__Q18:__ What do you consider yourself primarily?

Choose one of the following answers

- Developer
- Enterprise user
- Small and medium-sized enterprise user
- Scientific user
- End user (following private interests)

Q19: How often do you use cloud computing services?

Choose one of the following answers

- daily
- weekly
- monthly
- less than monthly
- never
- No answer

Q20: Which cloud computing "layer" is most important for your projects?

- Software as a Service
- Platform as a Service
- Infrastructure as a Service
- No answer

From our perspective there are three different layers in cloud computing at the moment:

- Software as a Service (SaaS) meaning applications offered on-demand over the Internet like Salesforce.
- Platform as a Service (PaaS) meaning developer platform with built-in services like Google Apps Engine or Microsoft Azure.
- Infrastructure as a Service (IaaS) meaning basic storage and compute capabilities offered as a service like Amazon Web Services or Mosso.

Q21: Which is your central domain for cloud computing?

Check at most 3 answers

- Application hosting/ SaaS
- High performance computing
- Backup and storage
- Content delivery
- Web hosting
- Social networking
- Search engines
- E-commerce
- Other:

Figure A.5: Websurvey Questionnaire Part 5

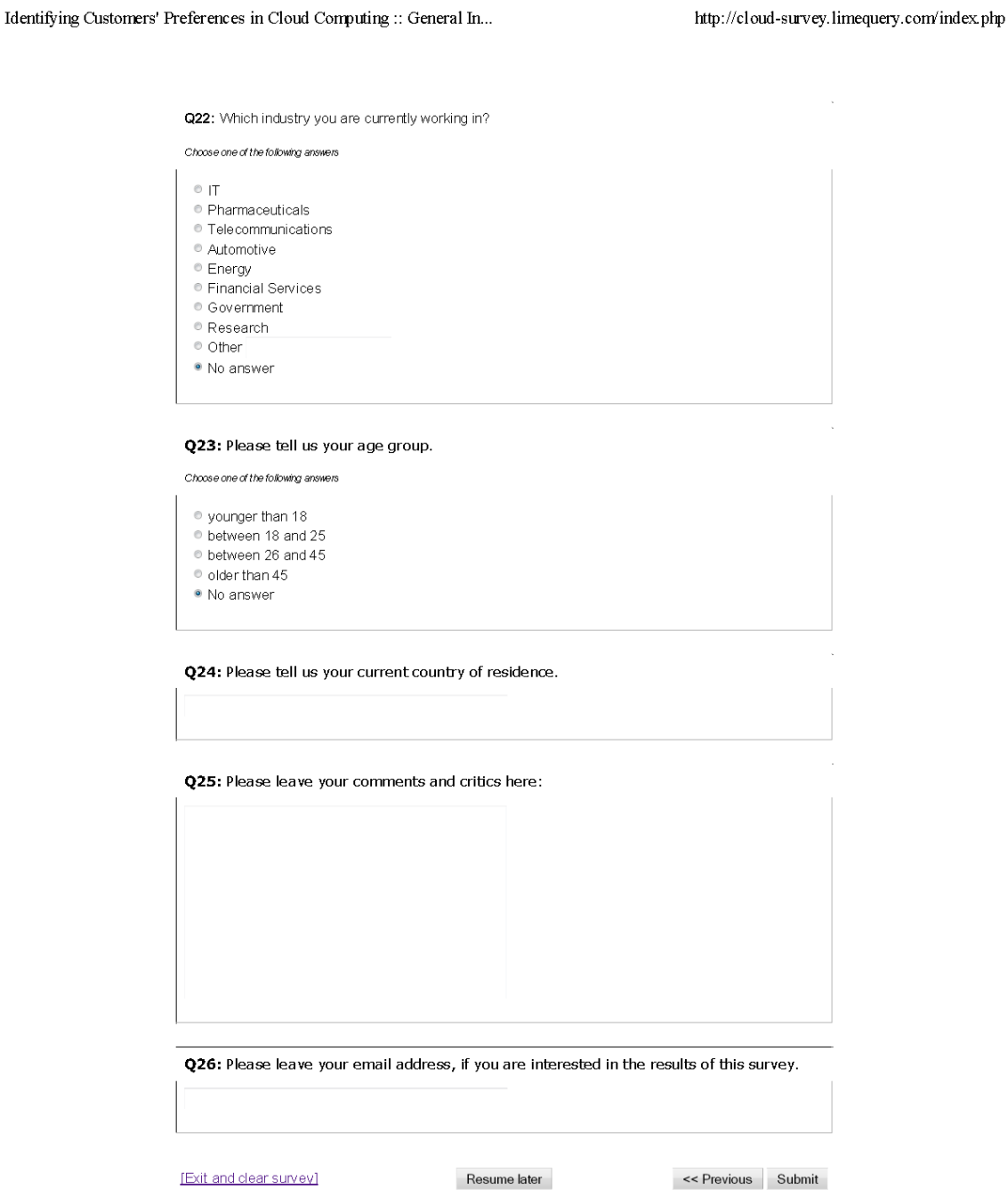


Figure A.6: Websurvey Questionnaire Part 6

A.2 Profile Cards

IISM

Identifying Customers' Preferences in Cloud Computing

0% 100%

Product Samples

The product samples consist of seven characteristics. Those characteristics and their respective specifications are described below.

1. **Price:** For the following sample products there are three different prices **($0.70, $1.10 and $2.00)**.The prices are on an hourly basis and include CPU (2.0 GHz CPU - 4 virtual cores), memory (15 GB), data transfer (max. 10 GB), storage (1500 GB) and the subsequently presented characteristics.
2. **Performance:** Within this survey we distinguish between **guaranteed-performance** and **best-effort** service types. Guaranteed performance means a guaranteed high service level and thereby high performance. A best-effort service has a lower priority and jobs may have to wait before being executed
3. **Support Level:** Products offer different levels of support **(phone, email and documentation)**. Support via phone means you can contact support staff by phone. If the product offers only email support this implies support staff can only be contacted via email. Documentation is provided on the website if products include documentation.
4. **Start-up Time:** Start-up Time defines the period between 'booking'/registering and set-up of an instance. This survey distinguishes between **instant** start-up which means within minutes the instance is ready for calculations and **prolongated** start-up. In this case there is a significant interval between 'booking' and set-up.
5. **Operating System:** Within this survey we distinguish between instances provided with **Windows** or **Linux** only and an environment where you can choose between **both** operating systems during set-up.
6. **Availability:** There are three different availability-levels:
   99.95% equals less than 4.5 hours downtime per year
   99.75% equals less than 22 hours downtime per year
   99.50% equals less than 44 hours downtime per year
7. **Value-added:** Value-added services can be a **Firewall**, **Load Balancing** or **none** at all.

*__Q12:__ Please rate the following products according your preferences:

	Price:	Performance:	Support Level:	Start-up Time:	Operating System:	Availability:	Value-added Services:
Product 1	*$0.70*	*Guaranteed*	*Email*	*Instant*	*Both*	*99.95%*	*None*
Product 2	*$1.10*	*Best Effort*	*Documentation*	*Instant*	*Linux*	*99.95%*	*None*
Product 3	*$2.00*	*Guaranteed*	*Documentation*	*Instant*	*Linux*	*99.50%*	*Load Balancing*
Product 4	*$0.70*	*Guaranteed*	*Documentation*	*Prolongated*	*Windows*	*99.95%*	*Firewall*

	7 - I would definitely buy this product.	6	5	4	3	2	1 - I would definitely not buy this product.
Product 1	○	○	○	○	○	○	○
Product 2	○	○	○	○	○	○	○
Product 3	○	○	○	○	○	○	○
Product 4	○	○	○	○	○	○	○

*__Q13:__ Please rate the following products according your preferences:

	Price:	Performance:	Support Level:	Start-up Time:	Operating System:	Availability:	Value-added Services:
Product 5	*$2.00*	*Guaranteed*	*Documentation*	*Prolongated*	*Both*	*99.50%*	*None*

Figure A.7: Websurvey Conjoint Analysis Part 1

Product 6	*$2.00*	*Best Effort*	*Email*	*Prolongated*	*Windows*	*99.75%*	*None*
Product 7	*$1.10*	*Guaranteed*	*Email*	*Prolongated*	*Linux*	*99.95%*	*Load Balancing*
Product 8	*$0.70*	*Best Effort*	*Phone*	*Prolongated*	*Linux*	*99.75%*	*Firewall*

	7 - I would definitely buy this product.	6	5	4	3	2	1 - I would definitely not buy this product.
Product 5	○	○	○	○	○	○	○
Product 6	○	○	○	○	○	○	○
Product 7	○	○	○	○	○	○	○
Product 8	○	○	○	○	○	○	○

*__Q14:__ Please rate the following products according your preferences:

	Price:	Performance:	Support Level:	Start-up Time:	Operating System:	Availability:	Value-added Services:
Product 9	*$1.10*	*Best Effort*	*Documentation*	*Instant*	*Both*	*99.75%*	*Firewall*
Product 10	*$2.00*	*Guaranteed*	*Phone*	*Instant*	*Both*	*99.95%*	*Firewall*
Product 11	*$0.70*	*Best Effort*	*Email*	*Instant*	*Both*	*99.50%*	*Load Balancing*
Product 12	*$1.10*	*Guaranteed*	*Email*	*Instant*	*Windows*	*99.50%*	*Firewall*

	7 - I would definitely buy this product.	6	5	4	3	2	1 - I would definitely not buy this product.
Product 9	○	○	○	○	○	○	○
Product 10	○	○	○	○	○	○	○
Product 11	○	○	○	○	○	○	○
Product 12	○	○	○	○	○	○	○

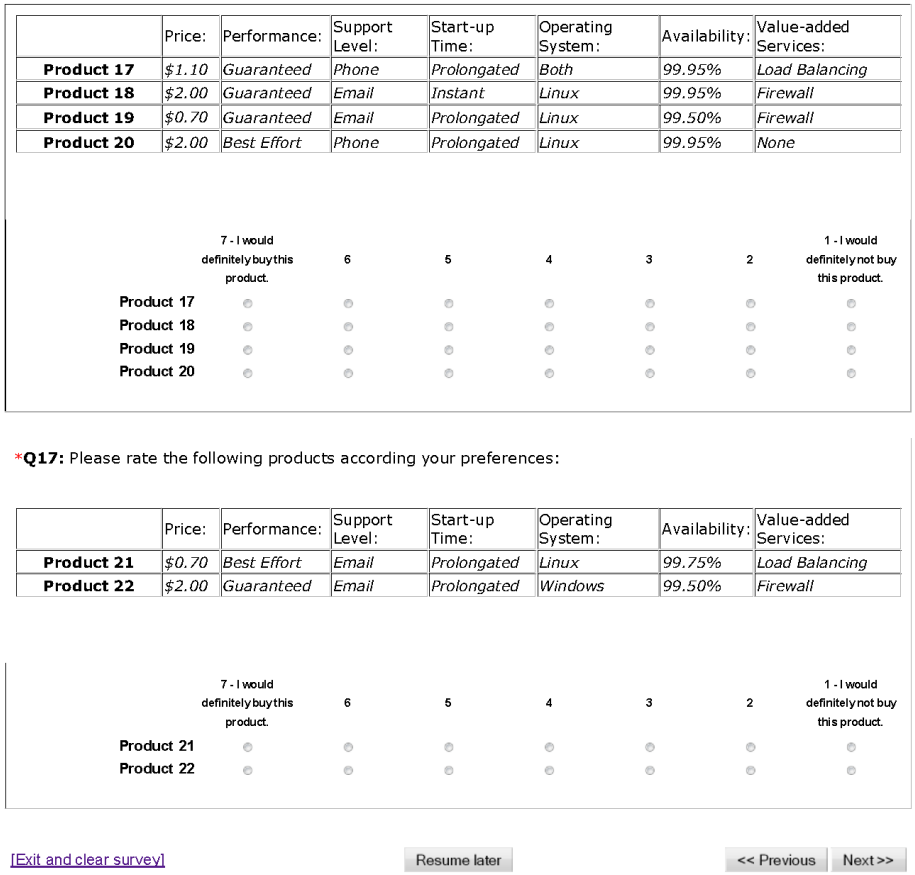
*__Q15:__ Please rate the following products according your preferences:

	Price:	Performance:	Support Level:	Start-up Time:	Operating System:	Availability:	Value-added Services:
Product 13	*$0.70*	*Guaranteed*	*Phone*	*Instant*	*Linux*	*99.75%*	*None*
Product 14	*$2.00*	*Best Effort*	*Phone*	*Instant*	*Windows*	*99.95%*	*Load Balancing*
Product 15	*$0.70*	*Guaranteed*	*Documentation*	*Instant*	*Windows*	*99.75%*	*Load Balancing*
Product 16	*$1.10*	*Guaranteed*	*Phone*	*Instant*	*Windows*	*99.50%*	*None*

	7 - I would definitely buy this product.	6	5	4	3	2	1 - I would definitely not buy this product.
Product 13	○	○	○	○	○	○	○
Product 14	○	○	○	○	○	○	○
Product 15	○	○	○	○	○	○	○
Product 16	○	○	○	○	○	○	○

*__Q16:__ Please rate the following products according your preferences:

Figure A.8: Websurvey Conjoint Analysis Part 2



	Price:	Performance:	Support Level:	Start-up Time:	Operating System:	Availability:	Value-added Services:
Product 17	*$1.10*	*Guaranteed*	*Phone*	*Prolongated*	*Both*	*99.95%*	*Load Balancing*
Product 18	*$2.00*	*Guaranteed*	*Email*	*Instant*	*Linux*	*99.95%*	*Firewall*
Product 19	*$0.70*	*Guaranteed*	*Email*	*Prolongated*	*Linux*	*99.50%*	*Firewall*
Product 20	*$2.00*	*Best Effort*	*Phone*	*Prolongated*	*Linux*	*99.95%*	*None*

	7 - I would definitely buy this product.	6	5	4	3	2	1 - I would definitely not buy this product.
Product 17	○	○	○	○	○	○	○
Product 18	○	○	○	○	○	○	○
Product 19	○	○	○	○	○	○	○
Product 20	○	○	○	○	○	○	○

*__Q17:__ Please rate the following products according your preferences:

	Price:	Performance:	Support Level:	Start-up Time:	Operating System:	Availability:	Value-added Services:
Product 21	*$0.70*	*Best Effort*	*Email*	*Prolongated*	*Linux*	*99.75%*	*Load Balancing*
Product 22	*$2.00*	*Guaranteed*	*Email*	*Prolongated*	*Windows*	*99.50%*	*Firewall*

	7 - I would definitely buy this product.	6	5	4	3	2	1 - I would definitely not buy this product.
Product 21	○	○	○	○	○	○	○
Product 22	○	○	○	○	○	○	○

[Exit and clear survey] Resume later << Previous Next >>

Figure A.9: Websurvey Conjoint Analysis Part 3

A.3 Theoretical Models Overview

see next two pages

Table A.1: Overview of the related customer choice models from Revenue Management (part 1/2)

	Clusters					
	Dynamic Pricing			**Capacity rationing**		**Valuation uncertainty**
Description	**Su(2007)**	**Levina et al. (2009), Levin et al. (2006)**	**Yin and Tang(2006)**	**Gallego et al. (2008)**	**Su and Zhang (2005)**	**Koenigsberg et al. (2005)**
Types of customers	Strategic-high-type; Strategic-low-type; Myopic-high-type; Myopic-low-type	(2009) Customers have random valuation, volatile and vary over time (2006) Customers believe prices follow certain stochastic process	All customers are either strategic or myopic	Customers know their willingness-to-pay for periods 1, 2 ...; but discounted offers change their future expectations and buying behavior	Customers are myopic or strategic; myopic if their willingness to pay is greater than the market price, strategic if their willingness to pay is below the market price	Two types of customers: those with higher valuation and those with lower valuation;
Focus	Prices freely increase and decrease	(2009) Discount factor as 'degree of strategicity' (2006) Incorporation of demand learning by firms	Effect of inventory information on strategic customer behavior	Markdown management problem where vendor has to determine optimal sequence of discounts	Strategic customer behavior on supply change performance	Conditions under which offering last-minute deals is optimal under a single price policy
Outcome	Increasing prices optimal with strategic-high-type and myopic-low-type customers	(2009) Derivation of dynamic equilibrium (2006) Simulation-based technique to find optimal prices	Increased revenue when displaying one unit at a time	All customers are strategic: single price policy myopic and strategic customers: two-price markdown policy	Vendor's profits can improve through quantity or price commitment in decentralized supply chains	If customers are uncertain, firms will offer last-minute deals; for an intermediate capacity level, uncertainty with respect to the arrival of high valuation customers leads to last-minute discount offers

Table A.2: Overview of the related customer choice models from Revenue Management (part 2/2)

	Clusters					
	Customer response to dynamic pricing		**Choice from a set of products**		**Substitution and complementaries**	
Description	**Anderson and Wilson (2003)**	**Zhou et al. (2005)**	**Talluri and van Ryzin (2004)**	**Zhang and Cooper (2005, 2009)**	**Netessine et al. (2006)**	**Aydin and Ziya (2007)**
Types of customers	Strategic customers	Customers follow a price-sensitive Poisson process	Customers follow an independent demand model	Customers choose dynamically according to an random choice model	One customer per period	Target and non-target customers
Focus	Customers are able to understand the revenue approach by firms and act strategically to counter the price and quantity decisions	Study of the threshold purchasing policy for the strategic customer	Decision with subset of products to offer at each point in time	Parallel flights with separate inventories	Cross-selling in the dynamic setting as an opportunity complementary to single-product revenue management	Promotional product tha need to be sold after a certain deadline
Outcome	Customers may decide to wait in case low fare classes are closed; this may have serious revenue implications particularly for low demand flights	Both customer and firm can benefit from strategic customer behavior in practice by adopting a target price purchasing system	The more capacity available the further the optimal set is along the sequence	Separable bounds are equal if different flights have equal or different fares	Most effective when product inventory is approximately equal to expected demand	Under dynamic pricing, up-selling decision does not depend on inventory level or remaining time

Appendix B

Genetic Algorithm

Table B.1: Analysis of upper bound settings on outcome of CBPP

		$\alpha_U = 2.$ $\beta_U = 2$	$\alpha_U = 3.$ $\beta_U = 3$	$\alpha_U = 5.$ $\beta_U = 5$	$\alpha_U = 8.$ $\beta_U = 8$	$\alpha_U = 10.$ $\beta_U = 10$	$\alpha_U = 15.$ $\beta_U = 15$	$\alpha_U = 20$ $\beta_U = 20$
3x3	avg rev	19192	19239	19255	19288	19334	19306	19275
	std dev	619	633	642	605	499	676	712
	rev diff	1.7%	1.9%	2.0%	2.2%	2.4%	2.3%	2.1%
4x4	avg rev	18735	18794	18869	18884	18937	18896	18837
	std dev	658	678	696	649	565	704	748
	rev diff	3.8%	4.2%	4.6%	4.7%	5.0%	4.7%	4.4%
4x5	avg rev	23692	23735	23804	23840	23908	23876	23842
	std dev	628	665	689	625	537	688	734
	rev diff	4.8%	5.0%	5.3%	5.5%	5.8%	5.6%	5.5%
4x7	avg rev	26116	26175	26221	26354	26429	26401	26366
	std dev	318	321	375	287	245	354	394
	rev diff	5.7%	5.9%	6.1%	6.6%	7.0%	6.8%	6.7%
4x10	avg rev	21489	21510	21577	21625	21683	21611	21556
	std dev	315	316	332	267	232	327	368
	rev diff	4.1%	4.2%	4.5%	4.8%	5.1%	4.7%	4.4%
6x8	avg rev	22173	22263	22324	22409	22472	22426	22384
	std dev	439	431	401	358	296	376	447
	rev diff	5.2%	5.6%	5.9%	6.3%	6.6%	6.4%	6.2%
10x10	avg rev	22209	22267	22312	22370	22438	22393	22361
	std dev	525	506	483	459	384	470	521
	rev diff	3.8%	4.1%	4.3%	4.6%	4.9%	4.7%	4.6%

Bibliography

Adabala, S., Chadha, V., Chawla, P., Figueiredo, R., Fortes, J., Krsul, I., Matsunaga, A., Tsugawa, M., Zhang, J., Zhao, M., Zhu, L., and Zhu, X. (2005). From virtualized resources to virtual computing grids: the in-vigo system. *Future Generation Computer Systems*, 21(6):896–909.

Addelman, S. (1962). Orthogonal main-effect plans for asymmetrical factorial experiments. *Technometrics*, 4(1):21–46.

Adelman, D. (2007). Dynamic bid prices in revenue management. *Operations Research*, 55(4):647–661.

Ajzen, I. (2005). *Attitudes, Personality, and Behavior*. Open University Press, Maidenhead, England, second edition.

Akaah, I. and Korgaonkar, P. (1983). An empirical comparison of the predictive validity of self-explicated, huber-hybrid, traditional conjoint, and hybrid conjoint models. *Journal of Marketing Research*, 20(2):187–197.

Albaugh, V. and Madduri, H. (2004). The utility metering service of the universal management infrastructure. *IBM Syst. J.*, 43(1):179–189.

Alkadi, I. and Alkadi, G. (2006). Grid computing: The past, now, and future. *Human Systems Management*, 25(3):161–166.

Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2004). *Web services: concepts, architectures and applications*. Springer Verlag, Heidelberg, Germany.

Anandasivam, A., Best, P., and See, S. (2010). Customers' Preferences for Infrastructure Cloud Services. In *Proceedings of the 12th IEEE Conference on Commerce and Enterprise Computing*.

Anandasivam, A. and Neumann, D. (2009). Managing revenue in grids. In *Hawaii International Conference on System Sciences*, pages 1–10, Los Alamitos, CA, USA. IEEE Computer Society.

Anandasivam, A. and Premm, M. (2009). Bid price control and dynamic pricing in clouds. In *17th European Conference on Information Systems (ECIS 2009), Verona, Italy*, pages 328–Ű341.

Anandasivam, A. and Weinhardt, C. (2010). Towards an efficient decision policy for Cloud service providers. In *Proceedings of the International Conference on Information Systems (ICIS), Saint Louis, USA*.

Anderson, C. and Wilson, J. (2003). Wait or Buy? The Strategic Consumer: Pricing and Profit Implications. *Journal of the Operational Research Society*, 54(3):299–306.

Anderson, E. and Sullivan, M. (1993). The Antecedents and Consequences of Customer Satisfaction for Firms. *Marketing Science*, 12(2):125–143.

Anderson, S. P., de Palma, A., and Thisse, J.-F. (1992). *Discrete Choice Theory of Product Differentiation*. MIT Press, Cambridge, USA.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing. Technical report, EECS Department, University of California, Berkeley.

Austin, D., Barbir, A., Ferris, C., and Garg, S. (2004a). Web Services Architecture Requirements. Technical report, W3C Working Group. http://www.w3.org/TR/wsa-reqs/.

Austin, J., Jackson, T., Fletcher, M., Jessop, M., Cowley, P., and Lobner, P. (2004b). Predictive maintenance: Distributed aircraft engine diagnostics. In Foster, I. and Kesselman, C., editors, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann [u.a.], Amsterdam; Heidelberg [u.a.], 2. edition.

Aydin, G. and Ziya, S. (2008). Pricing promotional products under upselling. *Manufacturing & Service Operations Management*, 10(3):360–376.

Backhaus, K., Erichson, B., Plinke, W., and Weiber, R. (2008). *Multivariate Analysemethoden*. Springer, Berlin, Heidelberg, twelfth edition.

Bailey, M. (2008). Building, planning, and operating the next-generation datacenter. Technical report, IDC Research, Inc.

Baker, M., Buyya, R., and Hyde, D. (1999). Cluster computing: A high-performance contender. *Computer*, 32(7):79–83.

Bakos, Y. and Brynjolfsson, E. (1999). Bundling information goods: Pricing, profits, and efficiency,. *Management Science*, 45(12):1613–1630.

Balakrishnan, P., Selvi, S., and Britto, G. (2008). Service level agreement based grid scheduling. In *IEEE International Conference on Web Services*, pages 203–210.

Banker, R. and Kauffman, R. (2004). The Evolution of Research on Information Systems: A Fiftieth-Year Survey of the Literature in "Management Science". *Management Science*, 50(3):281–298.

Barker, K. J., Davis, K., Hoisie, A., Kerbyson, D. J., Lang, M., Pakin, S., and Sancho, J. C. (2009). Using performance modeling to design large-scale systems. *Computer*, 42(11):42–49.

Bell, G. and Gray, J. (2002). What's next in high-performance computing? *Communication of the ACM*, 45(2):91–95.

Belobaba, P. P. (1987). Airline Yield Management An Overview of Seat Inventory Control. *Transportation Science*, 21(2):63.

Belobaba, P. P. (1989). Application of a Probabilistic Decision Model to Airline Seat Inventory Control. *Operations Research*, 37(2):183–197.

Bemer, R. (1957). How to consider a computer. *Automatic Control Magazine*, pages 66–69.

Ben-Akiva, M. and Lerman, S. (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press Series in Transportation Studies. MIT Press, Cambridge, USA.

Bertsimas, D. and de Boer, S. (2005). Simulation-based booking limits for airline revenue management. *Operations Research*, 53(1):90–106.

Bertsimas, D. and Popescu, I. (2003). Revenue Management in a Dynamic Network Environment. *Transportation Science*, 37(3):257–277.

Bichler, M., Kalagnanam, J., Katircioglu, K., King, A. J., Lawrence, R. D., Lee, H. S., Lin, G. Y., and Lu, Y. (2002). Applications of flexible pricing in business-to-business electronic commerce. *IBM Systems Journal*, 41(2):287–302.

Bichler, M. and Setzer, T. (2007). Admission control for media on demand services. *Service Oriented Computing and Applications*, 1(1):65–73.

Bitner, M., Faranda, W., Hubbert, A., and Zeithaml, V. (1997). Customer contributions and roles in service delivery. *International Journal of Service Industry Management*, 8(3):193–205.

Bitran, G. and Caldentey, R. (2003). An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, 5(3):203–229.

Bitran, G. R. and Mondschein, S. V. (1997). Periodic pricing of seasonal products in retailing. *Management Science*, 43(1):64–79.

Bixby, R. (2002). Solving Real-World Linear Programs: A Decade and More of Progress. *Operations Research*, 50(1):3–15.

Blau, B. (2009). *Coordination in Service Value Networks*. PhD thesis, University of Karlsruhe, Karlsruhe, Germany.

Bonorden, O., Gehweiler, J., and Meyer auf der Heide, F. (2006). A web computing environment for parallel algorithms in java. *Scalable Computing: Practice and Experience*, 7(2):1–14.

Boss, G., Malladi, P., Quan, D., Legregni, L., and Hall, H. (2007). Cloud computing. Technical report, IBM Corporation. http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud%_computing_wp_final_8Oct.pdf.

Botimer, T. and Belobaba, P. (1999). Airline pricing and fare product differentiation: A new theoretical framework. *The Journal of the Operational Research Society*, 50(11):1085–1097. (private-note)difference between price discrimination and product differentiation.

Boyd, A. E. and Bilegan, I. C. (2003). Revenue management and e-commerce. *Management Science*, 49(10):1363–1386.

Briscoe, G. and Marinos, A. (2009). Digital ecosystems in the clouds: towards community cloud computing. In Ieee, editor, *3rd IEEE International Conference on Digital Ecosystems and Technologies (DEST 2009)*, pages 103–108, New York, USA. Institute of Electrical and Electronics Engineers (IEEE).

Buyya, R., Yeo, C., and Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *10th IEEE International Conference on High Performance Computing and Communications*, pages 5–13, Washington DC, USA. IEEE Computer Society.

Carmone, F., Green, P., and Jain, A. (1978). Robustness of conjoint analysis: some Monte Carlo results. *Journal of Marketing Research*, 15:300–303.

Carr, N. (2005). The End of Corporate Computing. *MIT Sloan Management Review*, 46(3):67–73.

Carroll, W. and Grimes, R. (1995). Evolutionary Change in Product Management: Experiences in the Car Rental Industry. *Interfaces*, 25(5):84–104.

Chellappa, R. K. and Gupta, A. (2002). Managing computing resources in active intranets. *International Journal of Network Management*, 12(2):117–128.

Chen, C. and Kachani, S. (2007). Forecasting and optimisation for hotel revenue management. *Journal of Revenue and Pricing Management*, 6(3):163–174.

Cooper, W. L. (2002). Asymptotic Behavior of an Allocation Policy for Revenue Management. *Operations Research*, 50(4):720–727.

Coughlan, J. (1999). Airline overbooking in the multi-class case. *Journal of the Operational Research Society*, 50(11):1098–1103.

Cowan, G. (1998). *Statistical data analysis*. Oxford University Press, Oxford, UK.

Currie, W. L. (2004). Value creation from the application service provider e-business model: the experience of four firms. *Journal of Enterprise Information Management*, 17(2):117–130.

Currim, I. S., Weinberg, C. B., and Wittink, D. R. (1981). Design of subscription programs for a performing arts series. *The Journal of Consumer Research*, 8(1):67–75.

Darmon, R. and Rouziès, D. (1994). Reliability and internal validity of conjoint estimated utility functions under error-free versus error-full conditions. *International Journal of Research in Marketing*, 11:465–476.

Darmon, R. and Rouziès, D. (1999). Internal validity of conjoint analysis under alternative measurement procedures. *Journal of Business Research*, 46(1):67–81.

Dasilva, L. A. (2000). Pricing for qos enabled networks: A survey. *IEEE Communications Surveys & Tutorials*, 3(2):14–20.

Dawes, J. (2008). Do data characteristics change according to the number of scale points used? *International Journal of Market Research*, 50(1):61–77.

Decker, D. (2001). *Marktforschung mit dem Internet:Einsatzmöglichkeiten, Grenzen und Entwicklungspotenziale*. Tectum Verlag, Marburg, Germany.

Desai, B. and Currie, W. (2003). Application service providers: a model in evolution. In *ICEC '03: Proceedings of the 5th International Conference on Electronic Commerce*, pages 174–180.

DeShazo, J. and Fermo, G. (2002). Designing Choice Sets for Stated Preference Methods: The Effects of Complexity on Choice Consistency. *Journal of Environmental Economics and Management*, 44(1):123–143.

Dixit, A., Whipple, T., Zinkhan, G., and Gailey, E. (2008). A Taxonomy of Information Technology-enhanced Pricing Strategies. *Journal of Business Research*, 61(4):275–283.

Domschke, W. and Klein, R. (2004). Bestimmung von Opportunitätskosten am Beispiel des Produktionscontrolling. *Zeitschrift für Planung und Unternehmenssteuerung*, 15:275–294.

Dube, P., Hayel, Y., and Wynter, L. (2005). Yield management for IT resources on demand: analysis and validation of a new paradigm for managing computing centres. *Journal of Revenue and Pricing Management*, 4(1):24–38.

Elmaghraby, W. and Keskinocak, P. (2003). Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Science*, 49(10):1287–1309.

Figueiredo, R., Dinda, P., and Fortes, J. (2003). A Case for Grid Computing on Virtual Machines. In *Proceedings 23rd International Conference on Distributed Computing Systems*, pages 550–559.

Fishburn, P. C. (1967). Methods of estimating additive utilities. *Management Science*, 13(7):435–453.

Fishburn, P. C. and Odlyzko, A. M. (1999). Competitive pricing of information goods: Subscription pricing versus pay-per-use. *Economic Theory*, 13(2):447–470.

Foster, I. (2002). What is the Grid? A Three Point Checklist. *Grid Today*, 1(6):22–25.

Foster, I., Freeman, T., Keahy, K., Scheftner, D., Sotomayer, B., and Zhang, X. (2006). Virtual clusters for grid communities. In *Sixth IEEE International Symposium on Cluster Computing and the Grid,*, pages 513–520.

Foster, I. and Kesselman, C. (1998). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., San Francisco, USA, first edition.

Foster, I. and Tuecke, S. (2005). Describing the elephant: The different faces of it as service. *Queue*, 3(6):26–29.

Foster, I., Zhao, Y., Raicu, I., and Lu, S. (2008). Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop GCE '08*, pages 1–10.

Fox, A., Gribble, S. D., Chawathe, Y., Brewer, E. A., and Gauthier, P. (1997). Cluster-based scalable network services. *ACM SIGOPS Operating Systems Review*, 31(5):78–91.

Frank, M., Friedemann, M., and Schroder, A. (2008). Principles for simulations in revenue management. *Journal of Revenue and Pricing Management*, 7(1):7–16.

Gallego, G., Phillips, R., and Sahin (2008). Strategic management of distressed inventory. *Production and Operations Management*, 17(4):402–415.

Gallego, G. and van Ryzin, G. J. (1994). Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40(8):999–1020.

Gallego, G. and van Ryzin, G. J. (1997). A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research*, 45(1):24–41.

Gallouj, F. and Weinstein, O. (1997). Innovation in services. *Research Policy*, 26(4-5).

Gens, F. (2008). IDC on the Cloud. IDC Research, Inc., Survey report. `http://blogs.idc.com/ie/?p=189`, accessed on September 11, 2009.

Glover, F., Glover, R., Lorenzo, J., and McMillan, C. (1982). The passenger mix problem in the scheduled airlines. *Interfaces*, 12:73–79.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, USA, 1 edition.

Goldman, P., Freling, R., Pak, K., and Piersma, N. (2002). Models and techniques for hotel revenue management using a rolling horizon. *Journal of Revenue and Pricing Management*, 1(3).

Gosavi, A., Bandla, N., and Das, T. (2002). A reinforcement learning approach to a single leg airline revenue management problem with multiple fare classes and overbooking. *IIE Transactions*, 34(9):729–742.

Gosavi, A., Ozkaya, E., and Kahraman, A. F. (2007). Simulation optimization for revenue management of airlines with cancellations and overbooking. *OR Spectrum*, 29(1):21–38.

Green, P. E. and Krieger, A. M. (1993). *Conjoint analysis with product-positioning applications*, volume 5 of *Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam, The Netherlands.

Green, P. E., Krieger, A. M., and Vavra, T. G. (1997). Evaluating new products. *Marketing Research*, 9(4):12–21.

Green, P. E. and Rao, V. R. (1971). Conjoint measurement for quantifying judgmental data. *Journal of Marketing Research*, 8:355–363.

Green, P. E. and Srinivasan, V. (1978). Conjoint analysis in consumer research: Issues and outlook. *The Journal of Consumer Research*, 5(2):103–123.

Green, P. E. and Srinivasan, V. (1990). Conjoint analysis in marketing: New developments with implications for research and practice. *Journal of Marketing*, 54(4):3–19.

Hahn, C. (1997). *Conjoint- und Discrete Choice-Analyse als Verfahren zur Abbildung von Präferenzstrukturen und Produktauswahlentscheidungen*. Betriebswirtschaftliche Schriftenreihe. LIT Verlag, Münster, Germany.

Heskett, J. (1990). *Service Breakthroughs: Changing the Rules of the Game*. The Free Press, New York, USA.

Hevner, A., March, S., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

Hill, P. (1999). Tangibles, intangibles and services: A new taxonomy for the classification of output. *The Canadian Journal of Economics*, 32(2):426–446.

Hoffman, D. (2003). Marketing + MIS = e-service. *Communications of the ACM*, 46(6):53–55.

Holland, J. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, USA.

Homburg, C. and Krohmer, H. (2003). *Marketingmanagement: Strategie - Instrumente - Umsetzung - Unternehmensführung*. Gabler Verlag, Wiesbaden, Germany.

Hosting.com (2009). 2009 Cloud Computing Trends Report. Technical report, Hosting.com. `http://www.hosting.com/cloudhosting/ebook/`.

Hsee, C. K. (1996). The evaluability hypothesis: An explanation for preference reversals between joint and separate evaluations of alternatives. *Organizational Behavior and Human Decision Processes*, 67(3):247–257.

IDC Research, Inc. (2009). IDC-Studie: Cloud Computing in Deutschland ist noch nicht angekommen. IDC Corporate Headquarters, Survey report. `http://www.idc.com/germany/press/presse_cloudcomp.jsp`, accessed on September 11, 2009.

Ivanov, I. (2008). *Utility Computing: Reality and Beyond*, volume 23 of *Communications in Computer and Information Science*, pages 16–29. Springer, Berlin, Germany.

Johnson, R. M. (1974). Trade-off analysis of consumer values. *Journal of Marketing Research*, 11(2):121–127.

Judd, R. (1964). The case for redefining services. *The Journal of Marketing*, 28(1):58–59.

Kaul, A. and Rao, V. R. (1995). Research for product positioning and design decisions: An integrative review. *International Journal of Research in Marketing*, 12(4):293–320.

Keeney, R. (1969). *Multidimensional utility functions: theory, assessment, and application.* PhD thesis, Massachusetts Institute of Technology, MIT, Cambridge.

Kimes, S. E. (1989). Yield management: A tool for capacity-constrained service firms. *Journal of Operations Management*, 8(4):348–363.

Kimms, A. and Mueller-Bungart, M. (2007). Simulation of stochastic demand data streams for network revenue management problems. *OR Spectrum*, 29(1):5–20.

Klein, R. (2007). Network capacity control using self-adjusting bid-prices. *OR Spectrum*, 29(1):39–60.

Knight, W. (2006). Unlocking the grid. *Engineering & Technology*, 1(3):42–45.

Koenigsberg, O., Muller, E., and Vilcassim, N. (2008). easyjet pricing strategy: Should low-fare airlines offer last-minute deals? *Quantitative Marketing and Economics*, 6(3):279–297.

Köhler, P., Anandasivam, A., MA, D., and Weinhardt, C. (2010). Customer Heterogeneity and Tariff Biases in Cloud Computing. In *Proceedings of the International Conference on Information Systems (ICIS), Saint Louis, USA*.

Lai, K. (2005). Markets are dead, long live markets. *ACM SIGecom Exchanges*, 5(4):1–10.

Leavitt, N. (2009). Is cloud computing really ready for prime time? *IEEE Computer*, 42(1):15–20.

Lee, B. K. and Lee, W. N. (2004). The effect of information overload on consumer choice quality in an on-line environment. *Psychology and Marketing*, 21(3):159–183.

Leff, A., Rayfield, J., and Dias, D. (2003). Service-level agreements and commercial grids. *IEEE Internet Computing*, 7(4):44–50.

Leipold, M., Otte, C., and Ziegler, M. (2009). Globale Avanade-Studie zeigt: Sicherheitsbedenken beim Cloud Computing bremsen Einzug der Technologie in Unternehmen - trotz wirtschaftlicher Vorteile. Technical report, Avanade Inc.

Levin, Y., McGill, J., and Nediak, M. (2006). Optimal dynamic pricing of perishable items by a monopolist facing strategic consumers. Technical report, Queen's University. To appear in Production and Operations Management.

Levina, T., Levin, Y., McGill, J., and Nediak, M. (2009). Dynamic pricing with online learning and strategic consumers: An application of the aggregating algorithm. *Operations Research*, 57(2):327–341.

Lin, G., Fu, D., Zhu, J., and Dasmalchi, G. (2009). Cloud Computing: IT as a Service. *IT Professional*, 11(2):10–13.

Littlewood, K. (1972). Forecasting and control of passenger bookings. Technical report.

Lovelock, C. and Wirtz, J. (2001). *Services Marketing: People, Technology, Strategy*. Pearson Prentice Hall, New Jersey, USA.

Lütters, H. (2004). *Online-Marktforschung*. Gabler Verlag, Wiesbaden, Germany.

Ma, D. and Seidmann, A. (2008). The Pricing Strategy Analysis for the "Software-as-a-Service" Business Model. In Altmann, J., Neumann, D., and Fahringer, T., editors, *Grid Economics and Business Models*, Lecture Notes in Computer Science, chapter 8, pages 103–112. Springer Berlin Heidelberg, Berlin, Heidelberg.

Maglaras, C. and Zeevi, A. (2005). Pricing and design of differentiated services: Approximate analysis and structural insights. *Operations Research*, 53(2):242–262.

Maglio, P., Srinivasan, S., Kreulen, J., and Spohrer, J. (2006). Service systems, service scientists, ssme, and innovation. *Communications of the ACM*, 49(7):81–85.

Malhotra, N. K. and Peterson, M. (2005). *Basic marketing research : A decision-making approach*. Pearson Education, Upper Saddle River, NJ, USA, second edition.

Mazzotta, M. and Opaluch, J. (1995). Decision making when choices are complex: A test of Heiner's hypothesis. *Land Economics*, 71:500–515.

McAfee, R. P. (2008). Price discrimination. In *Issues in Competition Law and Policy*, volume 1, chapter 20, pages 465–484.

McGill, J. and van Ryzin, G. (1999). Revenue management: Research overview and prospects. *Transportation Science*, 33(2):233–256.

McKnight, L. and Bailey, J. (1997). Internet economics: When constituencies collide in cyberspace. *IEEE Internet Computing*, 1(6):30–37.

McLaughlin, L. (2008). Cloud Computing Survey: IT Leaders See Big Promise, Have Big Security Questions. Technical report, CIO.com. `http://www.cio.com/article/455832/Cloud_Computing_Survey_IT_Leaders_See_Big_Promise_Have_Big_Security_Questions`.

Mell, P. and Grance, T. (2009). The nist definition of cloud computing (v15). Technical report, National Institute of Standards and Technology. `http://www.csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc`, accessed on October 23, 2009.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, USA.

Möller, A., Römisch, W., and Weber, K. (2008). Airline network revenue management by multistage stochastic programming. *Computational Management Science*, 5(4):355–377.

Moore, W., Gray-Lee, J., and Louviere, J. (1998). A cross-validity comparison of conjoint analysis and choice models at different levels of aggregation. *Marketing Letters*, 9(2):195–207.

Motahari-Nezhad, H. R., Stephenson, B., and Singha, S. (2009). Outsourcing business to cloud computing services: Opportunities and challenges. *IEEE IT Professional, Special Issue on Cloud Computing*, 11(2).

Nair, S. and Bapna, R. (2001). An application of yield management for Internet Service Providers. *Naval Research Logistics*, 48(5):348–362.

Neslin, S. A. (1981). Linking product features to perceptions: Self-stated versus statistically revealed importance weights. *Journal of Marketing Research*, 18(1):80–86.

Netessine, S., Savin, S., and Xiao, W. (2006). Revenue management through dynamic cross selling in e-commerce retailing. *Operations Research*, 54(5):893–913.

Netessine, S. and Shumsky, R. (2002). Introduction to the theory and practice of yield management. *INFORMS Transactions on Education*, 3(1):34–44.

Netessine, S. and Shumsky, R. A. (2005). Revenue management games: Horizontal and vertical competition. *Management Science*, 51(5):813–831.

Neumann, D. (2007). *Economic Models and Algorithms for Grid Systems*. PhD thesis, University of Karlsruhe, Karlsruhe, Germany.

Ng, I. (2005). Differentiation, self-selection and revenue management. *Journal of Revenue and Pricing Management*, 5(1):2–9.

Ng, I. C. L. (2008). *The pricing and revenue management of services*. Routledge, London, England.

Oppenheim, A. N. (2000). *Questionnaire Design, Interviewing and Attitude Measurement*. Leicester University Press, Leicester, England, second edition.

Osgood, C., Suci, G., and Tannenbaum, P. (1957). *The measurement of meaning*. University of Illinois Press, Urbana.

Pak, K. (2005). *Revenue Management: New Features and Models*. PhD thesis, Erasmus University Rotterdam, Rotterdam, Netherlands.

Paleologo, G. (2004). Price-at-risk: A methodology for pricing utility computing services. *IBM Systems Journal*, 43(1):20–31.

Papazoglou, M. (2008). *Web services: principles and technology*. Prentice Hall, New Jersey, USA.

Paris, Q. (1981). Multiple optimal solutions in linear programming models. *American Journal of Agricultural Economics*, 63(4):724–727.

Parkhill, D. F. (1966). *The challenge of the computer utility*. Addison-Wesley Professional, USA.

Phillips, R. (2005). *Pricing and Revenue Optimization*. Stanford Business Books, Stanford, USA.

Rathmell, J. (1966). What is meant by services? *The Journal of Marketing*, 30(4):32–36.

Rechenberg, I. (1973). *Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann Holzboog, Berlin, Germany.

Rothstein, M. (1971). An Airline Overbooking Model. *Transportation Science*, 5(2):180–192.

Rust, R. and Kannan, P. (2003). E-service: a new paradigm for business in the electronic environment. *Communications of the ACM*, 46(6):36–42.

Saaty, T. L. (1980). The analytic hierarchy process, planning, piority setting, resource allocation. *European Journal of Operational Research*, 74(3):426–447.

Safizadeh, M. H. (1989). The internal validity of the trade-off method of conjoint analysis. *Decision Sciences*, 20(3):451–461.

Sarmenta, L. (2001). *Volunteer Computing*. PhD thesis, Cambridge, USA.

Schaffer, H., Averitt, S., Hoit, M., Peeler, A., Sills, E., and Vouk, M. (2009). Ncsu's virtual computing lab: A cloud computing solution. *Computer*, 42(7):94–97.

Schöneburg, E., Heinzmann, F., and Feddersen, S. (1995). *Genetische Algorithmen und Evolutionsstrategien: eine Einführung in Theorie und Praxis der simulierten Evolution*. Addison-Wesley, USA.

Schumpeter, J. A. (1935). The analysis of economic change. *The Review of Economics and Statistics*, 17(4):2–10.

Secomandi, N., Abbott, K., Atan, T., and Boyd, E. A. (2002). From Revenue Management Concepts to Software Systems. *Interfaces*, 32(2):1–11.

Segal, M. (1982). Reliability of conjoint analysis: Contrasting data collection procedures. *Journal of Marketing Research*, 19(1):139–143.

Shen, Z. J. M. and Su, X. (2007). Customer Behavior Modeling in Revenue Management and Auctions: A Review and New Research Opportunities. *Production and Operations Management*, 16(6):713–728.

Sheryl (2003). Revenue management: A retrospective. *Cornell Hotel and Restaurant Administration Quarterly*, 44(5-6):131.

Simon, H. (1996). *Sciences of the Artificial*. MIT Press, Cambridge, USA.

Smith, B. and Penn, C. (1988). Analysis of alternative origin-destination control strategies. In *Proceedings of the 28th Annual AGIFORS Symposium*, pages 123–144.

Srinivasan, V. (1988). A conjunctive-compensatory approach to the self-explication of multiattributed preference. *Decision Sciences*, 19(2):295–395.

Stahel, W. R. (1997). The service economy: 'wealth without resource consumption'? *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 355(1728):1309–1319.

Staten, J. (2008). Is Cloud Computing Ready For The Enterprise? Technical report, Forrester Research, Inc., Cambridge, USA.

Stoesser, J. (2009). *Market-Based Scheduling in Distributed Computing Systems*. PhD thesis, University of Karlsruhe, Karlsruhe, Germany.

Stokey, N. L. (1979). Intertemporal price discrimination. *The Quarterly Journal of Economics*, 93(3):355–371.

Street, D. J., Burgess, L., and Louviere, J. J. (2005). Quick and easy choice sets: Constructing optimal and nearly optimal stated choice experiments. *International Journal of Research in Marketing*, 22(4):459–470.

Su, X. (2007). Intertemporal pricing with strategic customer behavior. *Management Science*, 53(5):726–741.

Su, X. and Zhang, F. (2008). Strategic customer behavior, commitment, and supply chain performance. *Management Science*, 54(10):1759–1773.

Subramanian, J., Jr, and Lautenbacher, C. J. (1999). Airline Yield Management with Overbooking, Cancellations, and No-Shows. *Transportation Science*, 33(2):147–167.

Sulistio, A., Kim, K. H., and Buyya, R. (2008). Managing Cancellations and No-Shows of Reservations with Overbooking to Increase Resource Revenue. In *Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, pages 267–276. IEEE Computer Society Washington, DC, USA.

Talluri, K. and van Ryzin, G. (2004a). Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33.

Talluri, K. T. and van Ryzin, G. J. (1998). An Analysis of Bid-Price Controls for Network Revenue Management. *Management Science*, 44(11):1577–1593.

Talluri, K. T. and van Ryzin, G. J. (1999). A randomized linear programming method for computing network bid prices. *Transportation Science*, 33(2):207–216.

Talluri, K. T. and van Ryzin, G. J. (2004b). *The Theory and Practice of Revenue Management*. Springer, Berlin, Germany.

Teichert, T. (2001). *Nutzenschätzung in Conjoint-Analysen*. Deutscher Universitäts-Verlag, Wiesbaden, Germany.

Tversky, A. (1972). Elimination by Aspects: A Theory of Choice. *Psychological Review*, 79:281–299.

Urgaonkar, B., Shenoy, P., and Roscoe, T. (2002). Resource overbooking and application profiling in shared hosting platforms. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*.

van Ryzin, G. and Vulcano, G. (2008). Computing Virtual Nesting Controls for Network Revenue Management Under Customer Choice Behavior. *Manufacturing and Service Operations Management*, 10(3):448–467.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. (2009). A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1):50–55.

Vargo, S. and Lusch, R. (2004). Evolving to a new dominant logic for marketing. *Journal of Marketing*, 68(1):1–17.

Varian, H. (2002). Market structure in the network age. In Brynjolfsson, E. and Kahin, B., editors, *Understanding the Digital Economy: Data, Tools, and Research*, pages 137–150. The MIT Press.

Varian, H. R., Farrell, J., and Shapiro, C. (2004). *The Economics of Information Technology*. Cambridge University Press, Cambridge, USA.

Verma, R., Fitzsimmons, J., Heineke, J., and Davis, M. (2002). New issues and opportunities in service design research. *Journal of Operations Management*, 20(2).

Vinod, B. (2004). Unlocking the value of revenue management in the hotel industry. *Journal of Revenue and Pricing Management*, 3(2):178–190.

Vouk, M. A. (2008). Cloud computing - issues, research and implementations. *Journal of Computing and Information Technology*, 16(4).

Wang, L., Tao, J., Kunze, M., Castellanos, A. C., Kramer, D., and Karl, W. (2008). Scientific cloud computing: Early definition and experience. In *High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on*, pages 825–830.

Weatherford, L., Bodily, S., and Pfeifer, P. (1993). Modeling the Customer Arrival Process and Comparing Decision Rules in Perishable Asset Revenue Management Situations. *Transportation Science*, 27(3):239–251.

Weatherford, L. R. and Belobaba, P. P. (2002). Revenue impacts of fare input and demand forecast accuracy in airline yield management. *The Journal of the Operational Research Society*, 53(8):811–821.

Weatherford, L. R. and Bodily, S. E. (1992). A taxonomy and research overview of perishable-asset revenue management: yield management, overbooking, and pricing. *Operations Research*, 40(5):831–844.

Weinhardt, C., Anandasivam, A., Blau, B., and Stoesser, J. (2009). Business models in the service world. *IEEE IT Professional, Special Issue on Cloud Computing*, 11(2):28–33.

Weiss, A. (2007). Computing in the clouds. *netWorker*, 11(4):16–25.

Weiss, R. and Mehrotra, A. (2001). Online dynamic pricing: Efficiency, equity and the future of e-commerce. *Virginia Journal Of Law And Technology*, 6(2).

Welker, M., Werner, A., and Scholz, J. (2004). *Online-Research: Markt- und Sozialforschung mit dem Internet*. Dpunkt Verlag, Heidelberg, Germany.

Werner, A. and Stephan, R. (1998). *Marketing-Instrument Internet*. Dpunkt Verlag, Heidelberg, Germany.

Werthimer, D., Cobb, J., Lebofsky, M., Anderson, D., and Korpela, E. (2001). Seti@home—massively distributed computing for seti. *Computing in Science and Engineering*, 3(1):78–83.

Williamson, E. (1992). *Airline network seat control*. PhD thesis, Massachusetts Institute of Technology, Cambridge, USA.

Wilson, R. (1995). Nonlinear pricing and mechanism design. In Amman, H. M., Kendrick, D. A., and Rust, J., editors, *Handbook of Computational Economics (Vol. 1)*, pages 253–294. Elsevier.

Wittink, D., Vriens, M., and Burhenne, W. (1994). Commercial use of conjoint analysis in europe: Results and critical reflections. *Journal of Research in Marketing*, 11:41–52.

Won, K. (2009). Cloud computing: Today and tomorrow. *Journal of Object Technology*, 8(1):65–72.

Yin, R., Aviv, Y., Pazgal, A., and Tang, C. S. (2009). Optimal Markdown Pricing: Implications of Inventory Display Formats in the Presence of Strategic Customers. *Management Science*, 55(8):1391–1408.

Youseff, L., Butrico, M., and Da Silva, D. (2008). Toward a unified ontology of cloud computing. In *Grid Computing Environments Workshop, 2008. GCE '08*, pages 1–10.

Zhang, D. and Cooper, W. L. (2005). Revenue management for parallel flights with customer-choice behavior. *Operations Research*, 53(3):415–431.

Zhang, D. and Cooper, W. L. (2009). Pricing substitutable flights in airline revenue management. *European Journal of Operational Research*, 197(3):848–861.

Zhao, W. and Zheng, Y. S. (2001). A Dynamic Model for Airline Seat Allocation with Passenger Diversion and No-Shows. *Transportation Science*, 35(1):80–98.

Zhou, Y. P., Fan, M., and Cho, M. (2005). On the threshold purchasing behavior of customers facing dynamically priced perishable products. Technical report, University of Washington.

Declaration about the thesis

Erklärung

(gemäß §4, Abs. 4 der Promotionsordnung vom 15.08.2006)

Ich versichere wahrheitsgemäß, die Dissertation bis auf die in der Abhandlung angegebene Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und genau kenntlich gemacht zu haben, was aus Arbeiten anderer und aus eigenen Veröffentlichungen unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, 15.Mai 2010 _____

Arun Anandasivam