



KfK 3067
November 1980

Ermittlung des Leistungsverhaltens von Timesharing-Systemen

D. Schriefer
Hauptabteilung Datenverarbeitung und Instrumentierung

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

Hauptabteilung Datenverarbeitung und Instrumentierung

KfK 3067

Ermittlung des Leistungsverhaltens von Timesharing-Systemen

D. Schriefer

Als Dissertation genehmigt von der Fakultät für Informatik
der Universität Karlsruhe

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
ISSN 0303-4003

ERMITTLUNG DES LEISTUNGSVERHALTENS VON TIMESHARING-SYSTEMEN

Zusammenfassung

Zur Beschreibung des Leistungsverhaltens von TS-Systemen werden betriebsbegleitende Messungen der leistungsbeschreibenden Kenngrößen (Kontrollvariable KV) durchgeführt. Die Menge der KVn wird nach der Grösse ihres Einflusses auf die benutzer-bezogene Leistungsgrösse System-Antwortzeit durch Anwendung statistischer Verfahren geordnet (multiple Regression, Clusteranalyse) oder eingeschränkt (schrittweise multiple Regression).

Die so ausgewählten, für den Normalbetrieb des TS-Systems signifikanten KVn werden zur Bildung eines TS-Benchmarks (TS-BM) verwendet. TS-BMs sichern die Reproduzierbarkeit der Arbeitslast. Sie werden modular aus Basis-Scripts B_1, \dots, B_M (hier: $M = 3$) aufgebaut und als Script-Produkte (Parallel-Betrieb mehrerer Basis-Scripts) oder Script-Polynome (Serial-Betrieb mehrerer S-Produkte, zur Erreichung mittlerer KV-Werte eines Betriebszyklus) als Quasi-, Pseudo- oder dynamische TS-Benchmarks gefahren. Die Lastanpassung erfolgt durch Auswahl bei den S-Polynomen und durch Parametrierung von Denkzeit (Transaktionsrate), I/O-Rate und CPU-Zeit-Verbrauch bei den S-Produkten. Der dynamische BM wird aus symmetrischen S-Polynomen H_{MN} , bzw. aus einer Folge von homogenen S-Produkten S_i : $H_{MN}(B_1, \dots, B_M) = \sum_i a_i S_i(B_1, \dots, B_M)$ aufgebaut. Die Homogenität der S-Produkte garantiert die näherungsweise Anpassung des künstlichen Funktionsspektrums an die Betriebslast.

Es zeigt sich, dass mit dem dynamischen BM die Systemleistungskurven, die Auslastungen und das Antwortzeitverhalten der Betriebslast existierender TS-Systeme (hier: IBM/370) im Rahmen der Messgenauigkeit nachgebildet werden können. Veränderungen des Leistungsverhaltens nach Veränderungen am DV-System (Hardware, Software, Systemparameter) können durch Messung der maximalen Systemleistung (Leistungsvermögen), des Leistungsquotienten und des Produktivitätsgewinns bewertet werden. Veränderungen der Betriebslast werden erkannt; dies führt sowohl zu Verfahren zur Anpassung der Benchmarks als auch zu wichtigen Entscheidungen für die Anwendungs- und Systemplanung in der RZ- und Unternehmensleitung.

Investigation Into the Performance of Timesharing Systems.

Abstract

In order to describe the performance of TS-systems continuous measurements of the performance-related quantities (controls variables, CV) are carried out. By using statistical methods the set of CVs is ordered (multiple regression, clusteranalysis) or reduced (stepwise multiple regression) according to their impact on the user-specific performance measure 'system response time'.

The thus selected CVs, which are significant for the operation of the timesharing system, are used for the construction of a TS-benchmark (TS-BM). TS-BMs ensure the reproducibility of the workload. They are generated modularly from basic scripts B_1, \dots, B_M ($M=3$). They are used as system-drivers in the forms of quasi-, pseudo- or dynamic benchmarks which consist of script-products (parallelity) or script-polynomials (seriality, to obtain average CV-values for a given operation cycle). Workload approximation is achieved through selection (S-polynomials) and through parameter tuning (think time, transaction rate, I/O-rate, CPU time consumption) when S-products are used. Dynamic benchmarks consist of symmetric S-polynomials H_{MN} or of series of homogeneous S-products S_i : $H_{MN}(B_1, \dots, B_M) = \sum_i a_i S_i(B_1, \dots, B_M)$. The homogeneity of S-products guarantees the real workload's approximate simulation of the functional spectrum.

It is shown that - using dynamic benchmarking - system performance diagrams, resource utilizations, and response time behavior of existing systems' (IBM/370) workload can be modeled to measurement precision. Performance changes due to system alterations (hardware, software, system parameters) can be evaluated by measuring the maximum system performance, performance ratios, and productivity gains. Workload changes are recognized automatically; this leads to adjustments of the TS-BMs as well as to important insights as a basis for management decisions for application and system planning.

Inhaltsverzeichnis

<u>1. Einleitung.</u>	Seite 1
1.1. Einführung, 1 - 4.	
1.2. Stand der Technik, 4 - 9.	
1.3. Ziele der Arbeit, 9 - 11.	
1.4. Ergebnisse, 11 - 15.	
<u>2. Kontrollvariable eines Datenverarbeitungssystems und ihre Messverfahren.</u>	Seite 16
2.1. Die Kontrollvariablen, 16 - 27.	
2.2. Makroskopische Perioden und Messungen, 28.	
2.3. Mikroskopische Messungen, 28 - 31.	
2.4. Betriebszyklen, 31 - 33.	
2.5. Auswahlkriterien repräsentativer Betriebszyklen, 33 - 36.	
<u>3. Klassifizierung der Betriebszustände eines Rechnersystems mit gemischten Betriebsmodi.</u>	Seite 37
3.1. Klassifizierung der Kontrollvariablen, 38 - 45.	
3.2. Zerlegung eines Betriebszyklus in verschiedene Betriebszustände, 45 - 51.	
<u>4. Systemleistung und Servicegrad.</u>	Seite 52
4.1. Der TS-Concurrency Level, 52 - 54.	
4.2. Systemleistung, 55 - 56.	
4.3. Das Leistungsvermögen, 56 - 59.	
4.4. Leistungsvergleiche, 60.	
4.5. Mittlere Antwortzeit und Servicerate, 61 - 63.	
4.6. Produktivitätsgewinn und Gewinnfunktion, 64 - 66.	

<u>5. Das Funktionsspektrum eines TS-Systems.</u>	Seite 67
<u>6. Simulation des Funktionsspektrums.</u>	Seite 71
6.1. Scripts und Benchmarks, 72 - 75.	
6.2. Dynamischer Benchmark, Pseudo- und Quasi-Benchmark, 75 - 76.	
6.3. Basis-Scripts, 77 - 81.	
6.4. Script-Potenzen, -Produkte und -Polynome, 81 - 83.	
6.5. Lastanpassung der Scripts, 84 - 108.	
<u>7. Leistungsbewertungsverfahren.</u>	Seite 109
7.1. Leistungsbewertungsstrategie, 109 - 110.	
7.2. Analyse der Betriebsdaten aus der Leistungsdatenbank, 110 - 113.	
7.3. Regressionsmodelle, 114 - 121.	
<u>8. Beispiele für den Einsatz der Verfahren in der betrieblichen Praxis.</u>	Seite 122
8.1. Übersicht über die Auswahl der Methoden, 122 - 125.	
8.2. Beurteilung von System-Veränderungen, 126 - 156.	
8.3. Benutzer-Verlustzeiten, 157 - 162.	
<u>9. Anhang.</u>	Seite 163
9.1. Systemkonfigurationen, 163 - 164.	
9.2. Leistungsdatenbank, 165 - 167.	
9.3. Kommandos der Basis-Scripts, 168 - 172.	
9.4. Abkürzungsverzeichnis, 173 - 175.	
9.5. Formelsammlung, 176 - 178.	
<u>10. Literaturverzeichnis.</u>	Seite 179

Übersicht über die Abbildungen.

Kapitel 1.

- 1.1. Zusammenhang zwischen Last und Leistung eines DV-Systems, S. 4

Kapitel 2.

- 2.1. Relative Summenhäufigkeit S_R (SRT) und ihre approximierte Normalverteilung P_R (SRT) für die System-Antwortzeit, S. 19
- 2.2. Relative Summenhäufigkeiten von Ankunfts-, Antwort- und Transaktionszykluszeit, S. 19
- 2.3. Die I/O-Rate SIO als Funktion des TS-Concurrency Levels TS-CCL, S. 24
- 2.4. Die TS-I/O-Rate TS-I/O als Funktion der Anzahl der laufenden TS-Sitzungen u , S. 24
- 2.5. Die TS-I/O-Rate TS-I/O als Funktion der Transaktionsrate TRX, S. 25
- 2.6. Der TS-Concurrency Level TS-CCL als Funktion der Anzahl der laufenden TS-Sitzungen u . S. 25
- 2.7. Die Systemantwortzeit SRT als Funktion des TS-Concurrency Levels TS-CCL, S. 26
- 2.8. Die I/O-Rate SIO als Funktion der Anzahl der laufenden TS-Sitzungen u , S. 26
- 2.9. Die Systemantwortzeit SRT als Funktion der I/O-Rate SIO, S. 27
- 2.10. Der vollständige Transaktionszyklus, S. 32
- 2.11. Betriebszyklen: Der TS-Concurrency-Level als Beispiel, S. 34

Kapitel 3.

- 3.1. Datenmatrix zur Klassifizierung der Kontrollvariablen, S. 41
- 3.2. Partitionen mit Längen $p > 20$, Teil-Dendogramm, S. 43
- 3.3. Separation verschiedener Betriebszustände innerhalb eines 24-stündigen Betriebszyklus, S. 49
- 3.4. Alternativ-Separation der Betriebszustände, S. 49
- 3.5. Messwerte der zur Cluster-Analyse verwendeten Kontrollvariablen, S. 50

Kapitel 4.

- 4.1. Transaktionszeiten, S. 54
- 4.2. Systemleistungskurven, S. 59
- 4.3. Verteilung der relativen Systemleistung, S. 59
- 4.4. Serviceraten, S. 62
- 4.5. Der TS-Concurrency-Level als Funktion der Anz. der Terminals u , S. 63
- 4.6. Die Antwortzeit SRT als Funktion der Anz. d. Terminals u , S. 63

Kapitel 6.

- 6.1. Übersicht über die Script-Strukturen und ihre Nomenklatur, S. 83
- 6.2. Bestimmung der Zielwerte für das Minimal-Produkt S_3 des dynamischen Benchmarks, S. 88.
- 6.3. Die Abhängigkeit der mittleren Transaktionsrate des Script-Produkts S_1 von der mittleren Denkzeit, S. 90.
- 6.4. Die Abhängigkeit der mittleren I/O-Rate des Script-Produkts S_1 von der mittleren Denkzeit, S. 91.
- 6.5. Die Abhängigkeit des mittleren CPU-Zeit-Verbrauchs des Script-Produkts S_1 von der mittleren Denkzeit, S. 92.
- 6.6. Ablaufdiagramm zur Lastanpassung des minimalen Script-Produkts für den dynamischen TS-Benchmark, S. 93.
- 6.7. Die Transaktionsrate als Funktion der Denkzeit. Messwerte zur Parametrierung des minimalen Produkts für den dynamischen TS-Benchmark, S. 97.
- 6.8. Die I/O-Rate als Funktion der Denkzeit. Messwerte zur Parametrierung des minimalen Produkts für den dynamischen TS-Benchmark, S. 98.
- 6.9. Ablaufdiagramm zur Parametrierung eines Script-Polynoms als Pseudo-Benchmark, S. 100.
- 6.10. Ablaufdiagramm zur approximativen Lösung der Last-Gleichungen, S. 101.
- 6.11. Schrittweise Erzeugung des optimalen Script-Polynoms, S. 108.

Kapitel 7.

- 7.1. Übersicht - Leistungsbewertungsstrategie für TS-Systeme, S. 111.
- 7.2. Betriebliche Systemleistung, S. 113.
- 7.3. Antwortzeit-Vergleich, Multipl. Regr.modell (Modell II), S. 118.
- 7.4. Kontrollvariablenwerte für das Regressionsmodell (Modell II), S. 119.

Kapitel 8.

- 8.1. Vergleich der Systemantwortzeiten des TS-Benchmarks mit den Betriebsdaten, S. 131.
- 8.2. Systemleistungskurven Benchmark und Betriebsdaten, S. 131.
- 8.3. Systemleistungskurven für S158 und S168, S. 136.
- 8.4. Extrapolation der Antwortzeiten für S158 und S168, S. 136.
- 8.5. Betriebs-Systemleistungskurven für S158 und S168, S. 139.
- 8.6. Summenhäufigkeit $P_R(1)$ der relativen Systemleistung, S. 139.
- 8.7. Antwortzeiten für S158 und S168, S. 140.
- 8.8. Ablauf des Pseudo-Benchmarks, S. 145.
- 8.9. I/O-Antwortzeiten als Funktion des Blockungsfaktors BF, S. 147.
- 8.10. Ausnutzung der Plattenspeicherkapazität als Funktion des BF, S. 147.
- 8.11. Hauptspeicher-Bedarf für I/O-Puffer als Funktion des BF, S. 148.
- 8.12. I/O-Antwortzeiten als Funktion der Dateigrösse, S. 148.
- 8.13. Antwortzeiten des QBM als Funktion der Platten-Hardware, S. 151.
- 8.14. TSO-Systemleistung als Fkt. des APL-CPU-Quantums SAPL, S. 154.
- 8.15. CPU-Auslastungen als Fkt. des APL-Quantums SAPL, S. 155.
- 8.16. System-Antwortzeit $SRT(SAPL;u)$, extrapoliert, S. 155.
- 8.17. Die Systemleistungskurven $L(\text{SYSTEM},SAPL;u)$, S. 156.
- 8.18. CPU-Auslastungen S168, S. 156.
- 8.19. Benutzer-Verlustzeiten, S. 157.
- 8.20. Mittlere Verlustzeit je Benutzer pro Messzeit-Intervall, S. 159.
- 8.21. Benutzerverteilung $u(t)$, S. 159.
- 8.22. Gesamt-Verlust, S. 162.
- 8.23. Verlust-Summenhäufigkeitsverteilung, S. 162.

Übersicht über die Tabellen.

Kapitel 2.

- 2.1. Kontrollvariable, S. 21.
- 2.2. Kontrollvariablen-Paare mit hoher Korrelation, S. 23
- 2.3. Parametrierung des SW-Monitors RMF für makroskopische Messungen, S. 28
- 2.4. Overhead-Vergleich bei makroskop. und mikroskop. Messungen, S. 30
- 2.5. Nachweis der Homogenität mehrerer Betriebszyklen, S. 36

Kapitel 3.

- 3.1. Summenhäufigkeiten der Kontrollvariablen bei Partitionen der Länge p, S. 45
- 3.2. Zuordnung der Betriebszustände zu den Zeitintervallen Kontrollvariablen CPU, SIO und TS-CCL, S. 47
- 3.3. Zuordnung der Betriebszustände zu den Zeitintervallen Kontrollvariablen CPU, TS-CPU und TS-IO, S. 47

Kapitel 5.

- 5.1. Einteilung der Funktionen nach Anwendungsklassen, S. 68.
- 5.2. Einteilung der Funktionen nach Lastarten, S. 70.

Kapitel 6.

- 6.1. Grobstruktur der Basis-Scripts, S. 79.
- 6.2. Zuordnung der Kommandos zu den Funktionsklassen in Abhängigkeit von den Parametern der Basis-Scripts, S. 84.
- 6.3. Anteile der Funktionsklassen bei der Betriebslast, S. 85.
- 6.4. Anteile der Funktionsklassen bei den homogenen Script-Produkten $S_{3j} = (T \ I \ C)^j$ mit den Parametern ..., S. 86

- 6.5. Polynombildende Script-Produkte, S. 104.
- 6.6. Abweichungen der Kontrollvariablenwerte beim optimalen Script-Polynom, 5%-Fehlergrenze, S. 106.
- 6.7. Abweichungen der Kontrollvariablenwerte beim optimalen Script-Polynom, 2%-Fehlergrenze, S. 107.

Kapitel 7.

- 7.1. Regressionsmodell 1. Grades mit 7 Regressoren (Modell I), S. 116.
- 7.2. Regressionsmodell 1. Grades mit 3 Regressoren (Modell II), S. 116.
- 7.3. Residuen-Analyse für Modell II, S. 118.
- 7.4. Regressionsmodell 2. Grades mit 3 Regressoren (Modell III), S. 120.
- 7.5. Regressionsmodell 1. Grades mit 3 Regressoren (Modell IV), S. 121.

Kapitel 8.

- 8.1. Verweilzeiten des Timesharing-Benchmarks, S. 128.
- 8.2. Vergleich der Systemleistungswerte des TS-Benchmarks mit den Betriebsdaten, S. 129.
- 8.3. Vergleich der Antwortzeiten des TS-Benchmarks und der Betriebsdaten, S. 129.
- 8.4. Verweilzeiten des TS-Benchmarks auf einer IBM/370-168 (S168), S. 133.
- 8.5. Der Leistungsquotient $Q(u)$, S. 135.
- 8.6. Antwortzeiten-Vergleich, S158 und S168, S. 135.
- 8.7. Verweilzeiten des Quasi-Benchmarks, S. 143.
- 8.8. System-Antwortzeiten des Quasi-Benchmarks, S. 143.

1. Einleitung.

1.1. Einführung.

Seit zu Beginn der 60-Jahre die ersten nicht-militärischen Einsätze von Datenverarbeitungsanlagen in Realzeit-Betriebsweise, dem Timesharing- oder Teilnehmerbetrieb, vorgestellt wurden ^{1,2}, hat diese Betriebsart eine rapide Entwicklung durchgemacht und sich in breiten Anwendungskreisen durchgesetzt.

Neben dem ursprünglichen Ziel seiner Entwickler, zur Verkürzung der Testphase bei Programmentwicklungen durch kurze Transaktionen (auch Interaktionen genannt) in ständigen Kontakt mit dem Computer zu bleiben bei gleichzeitiger Bedienung mehrerer Benutzer durch einen Rechner³ im Zeitmultiplexverfahren, haben sich Anwendungen für diese Computer-Betriebsweise entwickelt, die im direkten Zusammenhang mit diesem Ziel stehen (wie die Datenaufnahme oder -erfassung, -manipulation und Programmentwicklungssysteme) oder die indirekt an den in der Folge der Weiterentwicklung der Timesharing-Technologie entstandenen neuen Verfahren der Nachrichtenübertragung, der Verwaltung von Terminalnetzen sowie der transaktions-orientierten Betriebsmittelvergabe partizipierten. Heutige TS-Systeme werden in universeller Form als Betriebssystem-Komponenten (z.B. Timesharing Option TSO der IBM OS/VS2 (MVS), Conversational Monitor System CMS des IBM VM/370) oder zur oder zur Unterstützung spezieller Anwendungssysteme mit unterschiedlichem Betriebssystem-Integrationsgrad eingesetzt. Beispiele dafür sind DB/DC (Data Base/Data Communication) Systeme, die den Mehrfachzugriff auf Datenbanken über ein Terminalnetzwerk ermöglichen.

Zur Abwicklung seiner Anwendungen steht jedem Benutzer heutiger TS-Systeme ein Satz von Funktionen zur Verfügung ⁴, eine Menge

¹ CORBATO et al, An Experimental... (CTSS), <CORB62>,

² SHAW: The JOSS System, siehe <SHAW64>,

³ WILKES: Timesharing Computer Systems, <WILK68>,

⁴ HELLERMAN/CONROY: Computer System Performance, <HELL75>,

von Standard-Bausteinen, mit deren Hilfe er seine Anwendung realisiert. Die Qualität dieses Funktionssatzes beeinflusst häufig die Qualität des gesamten Systems entscheidend. Dabei handelt es sich um die folgenden Funktionstypen:

- * Bearbeitung von Programmen und Datenobjekten,
- * Sitzungskontrolle,
- * Unterstützung bei der Erkennung und Behebung von Programmier- und Bedienungsfehlern und
- * Ausführung ablauffähiger Programme.

Es sind jedoch nicht nur die Mächtigkeit dieser Funktionen, die zur raschen Verbreitung der Timesharing-Systeme beigetragen haben, sondern vor allem auch der einfache Zugang (durch Datenstationen) zur und der direkte Kontakt mit der DVA (Interaktivität). "Der 'Benutzer-unfreundlichkeit' in der Nutzung von DV-Systemen in Form des Stapelbetriebs wirkt der Teilnehmerbetrieb entgegen" <FISCHER⁵>. Dies psychologisch wichtige Moment entstammt der dem Menschen angemessenen Art, möglichst nur kleine Schritte einer Teststrategie zu planen und sein weiteres Verhalten erst nach dem Vorliegen von Teilergebnissen festzulegen.

Quantitative Vergleiche zur Produktivität von Programmierern haben schon SACKMAN⁶, GOLD⁷, SMITH⁸, SCHATZOFF⁹ u.a vorgelegt. Spätere Arbeiten über den Einsatz von interaktiven Systemen und ihre Auswirkungen auf die auf DV-Leistung angewiesenen Mitarbeiter von Fachabteilungen in Forschungseinrichtungen finden sich in Kapitel 8.3., sowie in¹⁰ und¹¹.

⁵ FISCHER, Teilnehmersysteme, <FISC73>,

⁶ SACKMAN: Timesharing versus Batch Processing, siehe <SACK67>,

⁷ GOLD: Methodology..., siehe <GOLD67>,

⁸ SMITH: A Comparision..., siehe <SMIT67>,

⁹ SCHATZOFF/TSAO/WIIG: An Experimental..., in <SCHA67>,

¹⁰ STREETER: Productivity of..., siehe <STRE75>,

¹¹ DOHERTY/KELISKY: Managing VM/CMS Systems..., in <DOHE79>.

Die grössere 'Benutzerfreundlichkeit' der TS-Systeme schlägt jedoch leicht in Unzufriedenheit und Frustration der Benutzer um, wenn ihre Erwartungen bezüglich des Servicegrades nicht mehr erfüllt werden. Nun darf es nicht allein das Ziel der Betriebsführung für ein TS-System sein, die subjektiven Erwartungen der Benutzer zu befriedigen. Der Psychologe BOIES¹² hat mit umfangreichem statistischen Material darauf hingewiesen, dass ein direkter Zusammenhang zwischen System-Bearbeitungszeiten der Funktionen (Antwortzeiten) des TS-Systems (in der BOIES-Studie IBM-TSS) und den darauf folgenden Benutzer-Bearbeitungszeiten (Denkzeiten, wengleich sie auch andere Zeitanteile mitenthalten, wie z.B. die Eingabezeit, s. Abb. 2.10., S. 32) besteht: Werden die Antwortzeiten zu lang, so benötigen die Benutzer an den Terminals, offenbar unkonzentriert und irritiert, längere Zeiten bis zum Beginn der folgenden Transaktion. Dies bedeutet, dass nicht nur längere System-Antwortzeiten mehr Benutzerzeit - und somit höhere Personalkosten - erfordern, sondern dies sogar einen Bremseffekt für die Produktivität mit sich bringt, wenn der Benutzer n i c h t auf die DVA angewiesen ist (s. Kap. 8.). Zum Aspekt der Erwartung eines bestimmten Servicegrades siehe auch CIESLOWSKI¹³.

Nicht nur der Betrieb des TS-Systems lässt Kosten entstehen, sondern Abweichungen von einem erwarteten oder geplanten Servicegrad verursachen gerade dort zusätzliche Kosten, wo TS-Systeme den Aufwand abgebaut haben, nämlich auf der Benutzerseite. Es ist hier besonders wichtig, die Erwartungen bezüglich des Servicegrades zu quantifizieren, um dieses Mass gemeinsam mit der Gesamtlast in eine Beziehung zu setzen zu den Komponenten der vorhandenen oder zukünftigen Hardware und Software.

Das Leistungsverhalten (performance) einer DVA hängt nicht nur von den physikalischen Leistungsparametern einzelner Hardwarekomponenten ab, sondern auch von der Anzahl und der Art der bearbeiteten Funktionen, von der Verwaltung des Terminalnetzes und von der

¹² BOIES: User Behavior..., siehe <BOIE74>,

¹³ CIESLOWSKI: Perceptions of Performance, <CIES78>,

Struktur des Betriebssystems, das die Zuteilung der System-Ressourcen besorgt ¹⁴.

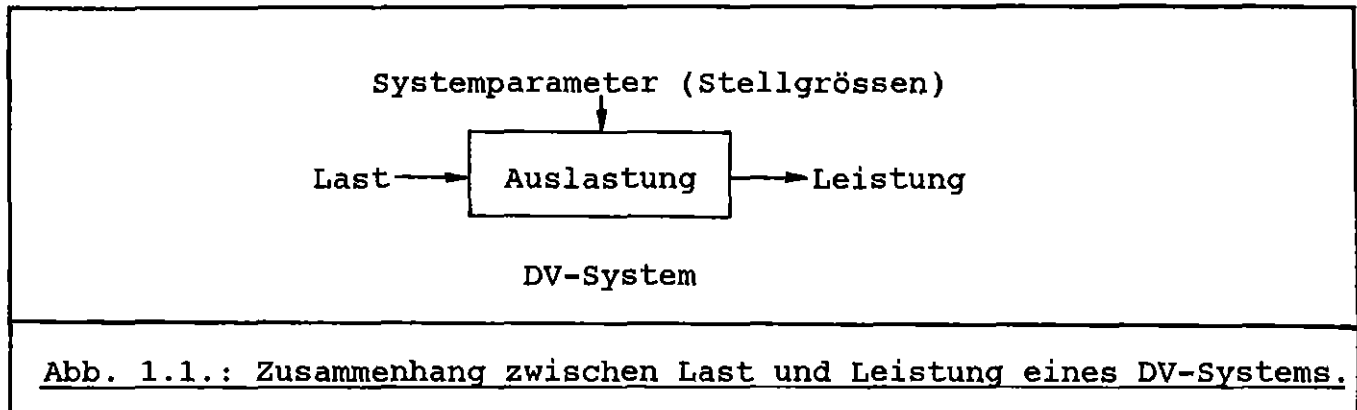


Abb. 1.1.: Zusammenhang zwischen Last und Leistung eines DV-Systems.

1.2. Stand der Technik.

Zum Vergleich des Leistungsverhaltens von DVA werden immer noch häufig nur die physikalischen Eigenschaften der Hardware benutzt^{15,16}. Für den Leistungsvergleich einzelner Komponenten liefert dies akzeptable Ergebnisse. Hier ist die interne Verarbeitungsgeschwindigkeit des Zentralprozessors (CPU), die sogenannte MIPS-Rate, anzuführen, die als Mass für das potentielle Leistungsvermögen der CPU dienen kann. In der Praxis wird das potentielle Leistungsvermögen jedoch nicht erreicht; daher spricht OSSWALD auch vom 'latenten' Leistungsvermögen¹⁶. Ähnliches gilt für Transferraten, Zugriffszeiten, Schreib- und Lesegeschwindigkeiten der angeschlossenen Kanäle und Speichermedien.

Die Antwortzeit stellt in einem Timesharing-System die wichtigste leistungsbeschreibende Grösse ^{17,18,19} dar, während sie für Batch-Systeme als die Verarbeitungszeit innerhalb der DVA nur einen geringen Teil an der gesamten Durchlaufzeit (turnaround time) angibt.

¹⁴ H. M. TEAGER im Vorwort zu <SCHE67>.

¹⁵ WINDFUHR: Methoden und Verfahren..., <Wind77>, S. 12.

¹⁶ OSSWALD: Leistungsvermögensanalyse..., <OSSW73>, S. 29 u. 46.

¹⁷ BUZEN: Fundamental Operational Laws..., <BUZE76>, S. 167-182.

¹⁸ HEIM: Methoden der Leistungsanalyse..., <HEIM79>, S. 70-86.

¹⁹ CLAPSON: Toward a Performance Science..., <CLAP76>, S. 308-315.

Die Antwortzeiten eines Timesharing-Systems sind im Normalbetrieb wegen der grossen Anzahl unterschiedlicher Transaktionen als statistisch verteilt anzusehen. Die Antwortzeit kann dann nur in der Form von Parametern ihrer Verteilung angegeben werden. Häufig werden Gesamtmittelwerte für vorgegebene Zeiträume verwendet; aussagekräftiger sind jedoch mehrere Fraktile (z.B. 0.50-, 0.75- und 0.95-Fraktile). Auch trägt eine Aufgliederung der Antwortzeiten nach Funktionsklassen zur besseren Leistungsbewertung bei (s. Kap. 5).

GRAEF²⁰ schlägt vor, von einem guten Leistungsverhalten zu sprechen, wenn bei einer vorgegebenen Soll-Antwortzeit 90% aller Antworten diese nicht überschreiten, 98% das Dreifache und der Rest das Zehnfache der Soll-Antwortzeit nicht überschreiten (diesem Vorschlag liegt eine angenäherte exponentielle Verteilung zugrunde).

FERRARI²¹ nennt die Leistungskennzahlen (OSSWALD) Performance-Indices und lässt in diesem Begriff subjektive Wertvorstellungen (siehe Seite 2) einfließen. Er klassifiziert die Indices nach Produktivität der DVA ('productivity'), nach Antwortzeitverhalten ('responsiveness') und Auslastung ('utilization') von Hardware und Software.

In dieser Arbeit wird in Anlehnung an eine Arbeit von BELL²² statt Leistungskennzahl oder Performance-Index die Bezeichnung Kontrollvariable verwendet werden (s. auch Kap. 2.). Von FERRARI stammt die Idee, namentlich zwischen primären und sekundären Kontrollvariablen zu unterscheiden. Die primären Kontrollvariablen sind die für eine Fragestellung relevanten, die sekundären sind entweder aus den primären abgeleitet oder werden für ihre Erzeugung benötigt, oder es handelt sich um solche Kontrollvariablen, deren Bedeutung für die untersuchte Fragestellung von untergeordneter Art ist²³.

²⁰ GRAEF/GREILLER/HECHT: Datenverarbeitung..., <GRAE70>, S. 42.

²¹ FERRARI: Comp. Syst. Perf. Eval., <FERR78>, S. 11 ff.

²² BELL: Performance Evaluation..., <BELL78>.

²³ FERRARI: Comp. Syst. Perf. Eval., <FERR78>, S. 17.

Die Kontrollvariablen müssen an der DVA gemessen oder aus Messwerten abgeleitet werden können.

Mit Hardware-Monitoren können solche Kontrollvariablen erfasst werden, die in Form von elektrischen Signalen zugänglich sind. Diese Messverfahren haben sich bei der Erkennung von ungleichmäßig ausgelasteten Hardware-Komponenten und bei der Behebung von I/O-Engpässen bewährt²⁴, zur Bestimmung der Leistung eines Gesamt-DV-Systems spielt ihr Einsatz jedoch keine Rolle²⁵. Software-Monitore²⁶ können Auskunft geben über tatsächliche Auslastungen, transferierte Datenmengen und Wartezeiten; ihr Einsatz ist jedoch stets systemspezifisch und erzeugt eine zusätzliche Mess-Last. Ausführliche Darstellungen zur Messtechnik mit wertenden Vergleichen finden sich bei SVOBODOVA²⁴, OSSWALD²⁷ und ELMENHORST²⁸.

Die Auswahl eines Messverfahrens hängt stets von den Zielen ab, die man mit einer Leistungsmessung erreichen will. Bei der Systembeurteilung während der Entwurfsphase oder bei der Suche nach Schwachstellen werden andere Verfahren eingesetzt als bei der Erstellung und Fortschreibung von Leistungsdaten zu planerischen Zwecken²⁹.

Zur Beurteilung des Leistungsverhaltens einer DVA ist neben der Kenntnis der die hardware- und softwarebeschreibenden Leistungsparameter auch eine quantitative Beschreibung der Arbeitslast^{30,31,32} notwendig. Nur mit einer qualifizierten und quantitativen Vorstellung über die Arbeitslast können die heute etablierten Verfahren des Benchmarking und der Simulation zu zuverlässigen Leistungsvorhersagen

²⁴ SVOBODOVA: Comp. Perf. Meas., <SVOB76>, S. 75-111.

²⁵ WINDFUHR: Methoden und Verfahren..., <WIND77>, S. 9.

²⁶ SCHRIEFER: Ein Stichprobenmonitor..., <SCHR76.2>.

²⁷ OSSWALD: Leistungsvermögensanalyse..., <OSSW73>, S. 81-90.

²⁸ ELMENHORST: Verfahren ..., <ELME74>, Kap. 2.5.

²⁹ ARIELY: Computer Measurement..., <ARIE74>.

³⁰ FERRARI: Comp. Syst. Perf. Eval. <FERR78>, Kap. 5, S. 211-295.

³¹ SREENIVASAN/KLEINMAN: On the Contruction..., <SREE74>.

³² BOARI/TOTH: Workload Definition..., <BOAR76>, S. 277-289.

oder Leistungsvergleichen herangezogen werden.

Simulationsmodelle lassen - sofern sie von den realen Gegebenheiten nicht zu stark abweichen, gute Voraussagen für die Betriebs- und Leistungsparameter erwarten. Ein hoher Aufwand an Zeit und Kosten, sowie Schwierigkeiten bei der Überprüfung des Gültigkeitsbereichs des Modells schränken ihre Anwendbarkeit jedoch ein. Es ist daher selbstverständlich, dass eine Nachbildung eines realen TS-Systems mit Hilfe von Simulationsmodellen nur dann zu rechtfertigen ist, wenn die hohen Kosten durch einen entsprechen grossen Nutzen mindestens ausgeglichen wird³³. Ein Simulationsmodell für das in dieser Arbeit hauptsächlich als Beispiel dienende IBM OS/VS2 MVS wird in ³⁴ vorgestellt.

Die Methode, mittels Benchmark-Programmen (siehe ^{35,36,37}) das Leistungsvermögen einer bestehenden Anlage zu untersuchen, hat sich in der jüngsten Vergangenheit stark verbreitet³⁸. Allerdings hatten daran insbesondere Batch-Systeme teil ^{39,40,41}.

Wenige Arbeiten auf dem Gebiete der Timesharing-Systeme ^{42,43} befassten sich mit sehr speziellen Fragestellungen, wobei i.a. synthetische Programme, sogenannte Scripts, verwendet wurden, die zwar funktionell angenähert, nicht jedoch lastmässig der natürlichen Last entsprachen. In ⁴⁴ wurden notwendige Tuning-(Leistungsverbesserungs-) Massnahmen ermittelt, um auf IBM-Systemen zusätzlich das Datenbanksystem

33 WINDFUHR: Methoden und Verfahren..., <WIND77>, S. 9.

34 CHIU/CHOW: A Performance Model..., <CHIU78>, S. 444 ff.

35 LITWILLER: Analysis..., <LITW68>.

36 BUCHHOLZ: A Synthetic Job..., <BUCH69>.

37 WOOD/FORMAN: Throughput Measurements..., <WOOD71>.

38 ZORN: Systemoptimierung..., <ZORN75>, S. 255-269.

39 MEAD/SCHWETMAN: Job Description..., <MEAD78>.

40 MÜLLER/KRIEGER/SCHÖNAUER: Der S.B.B., <MUEL74>.

41 SCHREIBER/THOMAS/WOLF: Beschreibung..., <SCHR74>.

42 TERPLAN: Workload Representation..., <TERP76>, S. 389-403.

43 MÜHLENBEIN: Ein Vorschlag..., <MUEH77.3>.

44 IBM: Host Systems Performance, <IBMW76>.

IMS betreiben zu können. ⁴⁵ enthält eine Kapazitätsstudie - ebenfalls in erster Linie für IMS, für die Timesharing-Aspekte wird auf zukünftige Arbeiten verwiesen. Mit festen Scripts wurden Vergleichsläufe für verschiedene Betriebssysteme oder verschiedene Zentraleinheiten durchgeführt ^{46,47,48}

Die Auswirkungen von Hauptspeicher-Engpässen (storage saturation) und daraus resultierender Paging-Raten auf die Antwortzeiten eines universellen TS-Systems hat BARD in ⁴⁹ untersucht. Ebenfalls BARD hat - aufbauend auf einer Klassifizierung der TS-Benutzer - Verfahren für die Beurteilung von Scheduling-Strategien angegeben ⁵⁰ und 1978 den Versuch unternommen, ein TS-System analytisch zu modellieren (IBM VM/370) ⁵¹. Er hat Beispiele für den praktischen Einsatz der warteschlangentheoretischen Ergebnisse gegeben ⁵². Schon SCHERR ⁵³ hat mit Warteschlangenmodellen TS-Systeme untersucht. Ihr Einsatz geht jedoch von einer Reihe von Annahmen aus, die in der betrieblichen Praxis nicht immer gegeben sind ⁵⁴.

Die besten, bisher veröffentlichten Ergebnisse zur Leistungsbeurteilung von TS-Systemen lassen sich mit Simulationsmodellen erzielen (siehe CHIU ⁵⁵ und MATERNA ⁵⁶). Jedoch ist der Zeit- und Kostenaufwand zur Erstellung solcher Modelle und die Beurteilung ihrer Güte enorm gross, sodass sich ihr Einsatz nur bei hinreichend grossen Projekten im Rechenzentrum rechtfertigen lässt.

-
- 45 ARTIS: Capacity Planning..., <ARTI78>.
 - 46 ALEXANDER/MERTENS/RUMLER: A Timesharing-Benchmark..., <ALEX76>.
 - 47 MERTENS: A Pragmatic Approach..., <MERT77>.
 - 48 MUEHLENBEIN: Ein Timesharing-Benchmark..., <MUEH77.1>.
 - 49 BARD: A Characterization..., <BARD76>, S. 35-55.
 - 50 BARD: The Modelling..., <BARD77>, S. 113-138.
 - 51 BARD: An Analytic Model..., <BARD78>, S. 498-508.
 - 52 BARD: Some Extensions..., <BARD79>.
 - 53 SCHERR: An Analysis..., <SCHE67>.
 - 54 WINDFUHR: Methoden und Verfahren..., <WIND77>, S. 58-60.
 - 55 CHIU/CHOW: A Performance Model..., <CHIU78>, S. 444 ff.
 - 56 MATERNA: Ein allgemeines Verfahren..., <MATE78>.

Umfassende Untersuchungen über systematische Zusammenhänge der Kontrollvariablen einer im TS-Betrieb fahrenden DVA, die eine permanente Leistungskontrolle und -bewertung eines TS-Systems ermöglichen und damit dem Rechenzentrums- und Unternehmensführung die Beurteilung kostenverursachender Aktivitäten erlauben, sind bisher nicht veröffentlicht. Eine Leistungsdefinition für Timesharing-Systeme, die gleichermaßen zur betriebsbegleitenden Leistungskontrolle und für die Systemplanung mit TS-Benchmarks herangezogen werden kann, muss in das unternehmensspezifische Berichtswesen integriert werden können. Eine solche Leistungsdefinition ist in der Literatur nicht bekannt.

In dieser Arbeit wird angestrebt, aufbauend auf bewährten Verfahren der Leistungsmessung, ein geeignetes Entscheidungsinstrumentarium für die Leistungsbewertung von Timesharing-Systemen zu schaffen, dessen Einsatz im Rechenzentrum wirtschaftlich ist und das den betrieblichen Realitäten eines Unternehmens Rechnung trägt. Seine Wirksamkeit wird anhand von Anwendungsbeispielen aus der Betriebspraxis dargelegt.

1.3. Ziele der Arbeit.

Die erarbeiteten Verfahren sollen es den Entscheidungsträgern des Rechenzentrums erlauben, bei konsequentem Einsatz der Verfahren

- eine laufendende Leistungskontrolle für das Timesharing-System ohne grossen Aufwand durchzuführen, Veränderungen des Lastprofils (also der Benutzung des Timesharing-Systems) zu erkennen und rechtzeitige Kapazitätsanpassungen zu veranlassen,
- Entscheidungsunterlagen für den Wirtschaftlichkeitsnachweis, soweit das Rechenzentrum dies tun kann, bereitzustellen und
- den Leistungsverbesserungsnachweis nach dem Auffinden, der Beurteilung und der Beseitigung von Schwachstellen des Systems zu erbringen.

Bei der Verfolgung dieser drei Ziele muss für alle auszuwählenden Verfahren stets von der Anwendbarkeit in der Betriebspraxis ausgegangen werden.

Die Frage nach der Zuverlässigkeit von System-Komponenten und die Beurteilung der Verfügbarkeit bei Timesharing-Systemen wurde in der vorliegenden Arbeit **n i c h t** behandelt. Wegen ihrer fundamentalen Bedeutung durch entstehende Verlustzeiten und Folge-Verlustzeiten, die sich aus dem Terminal-Benutzerverhalten ergeben, muss die Verfügbarkeit bei Timesharing-Systemen jedoch zukünftig mehr Beachtung finden.

Es werden alle Untersuchungen zur Leistungsbewertung aus der Sicht der Messbarkeit, der Leistungsanalyse, -verbesserung und -vorhersage unternommen:

A. Leistungsmessung: Zur Messung der leistungsrelevanten Kontrollvariablen soll ein festes Verfahren angegeben werden. Dieses Verfahren muss (weitgehend) unabhängig sein von Änderungen am Hardware- und Software-System, es soll aber auch unabhängig sein von Veränderungen der Last. Ebenso sollen die zu messenden Kontrollvariablen so festgelegt werden, dass sie für Timesharing-Systeme verschiedener Hersteller ihre Gültigkeit behalten.

B. Leistungsanalyse: Methoden zur Bestimmung der Zusammenhänge zwischen der Veränderung der last-bezogenen einerseits und der auslastungs-bezogenen Kontrollvariablen sollen formuliert und die Abhängigkeiten der Variablen untereinander dargestellt werden. Formale Methoden sollen es ermöglichen, Benutzungsintervalle mit gemischten Betriebsmodi des Systems zu separieren.

C. Leistungsverbesserungen und ihre Beurteilung: Verfahren zur Bewertung von Systemveränderungen sollen definiert werden (Leistungsvergleiche). Diese Verfahren sollen auch zur rechtzeitigen Erkennung von Engpässen am System verwendet werden können.

D. Leistungsvorhersage: Basierend auf den gewonnenen Kenntnissen

über die Veränderung der Last bei Timesharing-Systemen und ihrer Leistungsfähigkeit sollen Verfahren dargestellt werden, die Prognosen über Serviceverhalten und Auslastung bei zukünftigen Last- und Systemveränderungen zulassen.

Bei der Formulierung der Verfahren soll besonderes Gewicht auf die wirtschaftliche Einsetzbarkeit gelegt werden. Der für die Leistungsbewertung zu treffende Aufwand an Personal, Zeit und Kosten muss mit dem Nutzen vergleichbar und ihm gegenüber vertretbar sein. Diese Forderung ergibt sich direkt aus der Notwendigkeit der Anwendung der Verfahren in der betrieblichen Praxis des Rechenzentrums.

1.4. Ergebnisse.

Einem Grundgedanken von FERRARI folgend wird eine Hierarchie von Kontrollvariablen aufgestellt, deren kontinuierliche Beobachtung (betriebsbegleitende Messungen mit einem Software-Monitor) zur Aufstellung einer Leistungsdatenbank (Kap. 2.1., S. 22) geführt hat. Eine systematische Analyse der primären Kontrollvariablen-Werte der Leistungsdatenbank führt bereits zu einer Leistungsdefinition für Timesharing-Systeme (Kap. 7.2., S.110), die auf der Transaktionsrate und der System-Antwortzeit bei der Betriebslast beruht.

Zur Klassifizierung verschiedener Betriebsmodi eines Grossrechner-systems mit unterschiedlichen Lastarten (Timesharing, Batch u.a.) wird eine cluster-analytische Methode angegeben, um die Betriebsmodi anhand der Kontrollvariablen nach der Zeit zu separieren. Eine vorgeschaltete Cluster-Analyse über alle primären Kontrollvariablen ermöglicht es, den Aufwand für die Klassifizierung der Betriebsmodi durch eine Verringerung der Anzahl der Kontrollvariablen wesentlich zu reduzieren (Kap. 3., S. 37).

Eine Gliederung des Funktionsspektrums (Kommando-Vorrat des Timesharing-Systems) in transaktions-, I/O- und CPU-orientierte Kommandos wird unternommen (Kap. 5., S. 67). Es werden allgemeine Methoden zur Nachbildung der Betriebslast durch parametrierte Basis-Scripts vorgeschlagen, die einer solchen funktionellen Gliederung entsprechen (Kap.

6.3., S. 77). Die Begriffe Script-Produkt (paralleler Ablauf von Basis-Scripts), symbolisch:

$$SP = T^i I^j C^k,$$

wobei T, I und C drei Basis-Scripts bezeichnen, und Script-Polynom (sequentieller Ablauf von Script-Produkten), symbolisch:

$$P = \sum_r a_r SP(r),$$

werden eingeführt (Kap. 6.4., S. 81). Mit diesen Mischungen (Produkte, Polynome) der Basis-Scripts werden die Lastgrößen Transaktions-, I/O-Rate und CPU-Auslastung, sowie die Anzahl der Terminalsitzungen einer betrieblich relevanten Last im Mittel nachgebildet. Dabei bleiben die funktionsbezogenen Aspekte der einzelnen Basis-Scripts erhalten, sodass diese Methode auch die Approximation nur funktionell beschriebener zukünftiger oder künstlicher Lasten sichert.

Multiple Regressionsmodelle werden zur Vorhersage der Antwortzeiten untersucht. Regressionsmodelle verschiedenen Grades und mit unterschiedlichen Kontrollvariablen werden auf ihre Prognosequalität überprüft. Anhand von Betriebsdaten wird nachgewiesen, dass Modelle höheren Grades keinen signifikanten Gewinn für die Vorhersage der Service-Größe Antwortzeit erbringen. Prognosen auf dieser Basis müssen auf kurzzeitige Intervalle beschränkt bleiben (Kap. 7.3., S.120).

Leistungsmessung:

Standardisierte Messverfahren mit Software-Monitoren ermöglichen eine kontinuierliche Leistungskontrolle des Timesharing-Systems. Betriebsbegleitend werden die Mittelwerte der leistungsrelevanten Kontrollvariablen über 15-minütige Zeitintervalle mit einem zeit- und ereignis-getriebenen Software-Monitor aufgezeichnet (makroskopische Messungen, s. S. 28) und in regelmässigen Abständen in die Leistungsdatenbank übernommen. Messungen der Funktionsverteilung, der Verteilung der Zwischenankunfts-, System-Antwort- und Denkzeiten

werden wegen des höheren Aufwands nur sporadisch mit einem ereignisgetriebenen Software-Monitor durchgeführt (mikroskopische Messungen, s. S. 28). Sitzungs- und benutzer-bezogene Daten für die Abrechnung der Dienstleistungen werden regelmässig vom Accounting-System erfasst.

Leistungsanalyse:

Der Einsatz eines cluster-analytischen Verfahrens ermöglicht die Auswahl von drei signifikanten Kontrollvariablen, der CPU-Auslastung, der I/O-Rate und des Timesharing-Concurrency-Levels (Def. S. 55), die den Aufwand zur Klassifizierung verschiedener Betriebszustände anhand der Werte dieser Kontrollvariablen in einem Betriebszyklus wesentlich reduziert (Kap. 3., S. 38).

Die Einführung des Begriffes Systemleistung für Timesharing-Systeme (Kap.4.2., S. 55) erlaubt die Darstellung der Leistung bei steigenden Benutzerzahlen (Systemleistungskurve $L(u)$, s. S. 59). Für Systemvergleiche wird der Leistungsquotient Q (S. 60) definiert und es wird ein formelmässiger Zusammenhang zwischen Systemleistung und System-Antwortzeit hergestellt (Kap. 4.5., S. 61).

Leistungsverbesserung und -beurteilung, Vorhersage:

Hierfür werden Leistungsquotient und Gewinnfunktion (Kap. 4.6., S. 64) eingesetzt. Ergänzend zur Gewinnfunktion, die im Rechenzentrum aufgestellt wird und den Produktivitätsgewinn anhand der mittleren System-Antwortzeit berechnet, wird die von den Anwendern definierte, gemessene Benutzer-Verlustzeit-Funktion eingesetzt. Diese ist die Summe aller Wartezeiten, die für den Anwender über die geplanten und erwarteten System-Antwortzeiten hinausgehen. Die Verlustzeit-Funktion spiegelt die subjektive Benutzer-Zufriedenheit mit dem Timesharing-System wieder (Kap. 8.3., S. 157).

Zur Leistungsbeurteilung des Timesharing-Systems wird eine nach dem zu treffenden betrieblichen Aufwand beim Einsatz der Verfahren geordnete Hierarchie vorgeschlagen:

- A. Betriebsdaten-Analyse: Ausgehend von den in der Leistungsdatenbank gespeicherten Betriebsdaten werden die mittlere betriebliche Systemleistung $L_B(u)$ durch eine Regressionsparabel (2. Grades) und bei Systemvergleichen zusätzlich der betriebliche Leistungsquotient Q_B aus den Systemleistungen $L_B(u)$ der beiden zu vergleichenden Systeme bestimmt (Kap. 7.2., S.113, und Kap. 8.2.3., S. 135-139) und multiple Regressionsmodelle (1. Ordnung) mit den durch Cluster-Analyse ausgewählten Kontrollvariablen erstellt (Kap. 7.3., S.114). Hierbei wird gezeigt, dass Modelle höherer Ordnung den Mehraufwand nicht rechtfertigen.
- B. Quasi-Benchmark: (Kap.6.2., S. 75) Mit dem Quasi-Benchmark, dem Ablauf eines Basis-Scripts unter normaler Betriebslast, wird die Gewinnfunktion (Kap. 4.6., S. 64) für die durch das Basis-Script beschriebene Funktionsklasse beurteilt.

Diese beiden Stufen der Hierarchie (A. und B.) können ohne grossen Aufwand und ohne Betriebsunterbrechungen eingesetzt werden. Ihr Nachteil liegt darin, dass die den Beurteilungen zugrunde liegende Betriebslast im einzelnen nicht bekannt und nicht reproduzierbar ist.

- C. Pseudo-Benchmark: (Kap.6.2., S. 76) Beim Einsatz des Pseudo-Benchmarks werden die Lastzustände bereits durch ein Script-Polynom approximiert. Die durch den Pseudo-Benchmark erzeugte Last ist reproduzierbar; sie beruht auf einer künstlichen Last, deren CPU-Auslastung, I/O- und Transaktionsrate und Antwortzeit an einen mittleren oder beliebig punktuellen, betrieblich relevanten Lastzustand angenähert worden ist. Der Aufwand zur Parametrierung und zum Einsatz des Pseudo-Benchmarks wegen notwendiger Betriebsunterbrechungen wesentlich grösser als bei den o.a. Verfahren, jedoch kleiner als beim im folgenden angeführten dynamischen Benchmark. Dessen Einsatzgebiet liegt bei den Leistungsbeurteilungen anhand der Veränderungen von System-Antwortzeiten.

D. Dynamischer Benchmark: (Kap.6.2., S. 76) Der Einsatz eines dynamischen Timesharing-Benchmarks in der Form von Folgen lastmässig angenäherter homogener Script-Produkte oder Script-Polynome ermöglicht die Aufstellung der Systemleistungsfunktion $L(u)$ (Kap. 4.2., S. 55), der Darstellung der Abhängigkeit der Systemleistung von der Anzahl der gleichzeitig laufenden Terminalsitzungen (u) und damit der Bestimmung des Leistungsvermögens des Timesharing-Systems (maximale Systemleistung) bei der durch den Benchmark definierten Last, sowie den Vergleich zweier Systemleistungen und System-Antwortzeiten bei steigenden Terminalzahlen.

Damit ist der dynamische Timesharing-Benchmark dasjenige Verfahren, das Leistungsbewertungen mit der grössten Genauigkeit zulässt (im Vergleich zur Betriebsdaten-Analyse und zu Quasi- und Pseudo-Benchmark). Die Häufigkeit seines Einsatzes wird allerd durch den hohen betrieblichen Aufwand bei seiner Parametrierung und Durchführung eingeschränkt.

2. Kontrollvariable eines DV-Systems und ihre Messverfahren.

2.1. Die Kontrollvariablen.

Die Kontrollvariablen sind die für eine quantifizierende Beschreibung der verschiedenen leistungsbezogenen Funktionen eines DV-Systems bedeutsamen Masszahlen.

Die leistungsbezogenen Funktionen eines DV-Systems sind:

- * Funktionsumfang,
- * Arbeitslast,
- * Auslastung und
- * Servicegrad.

Diese Einteilung entspricht der von FERRARI¹; FERRARI klammert alle nicht oder nur sehr schwer zu quantifizierenden Grössen unter der Bezeichnung 'functional aspects' aus. Eine umfangreiche Liste von Leistungsmassgrössen mit detaillierten Literaturhinweisen findet man bei SVOBODOVA².

Unter dem Funktionsumfang werden alle funktionellen Fähigkeiten verstanden, die das DV-System den Anwendern anbietet, z.B. Batch-, Timesharing-, Realzeitbetrieb, Datenbanken und Kommunikationssysteme, aber auch die Möglichkeiten der Dateiverwaltung und Sprachübersetzung, der Editoren und der Funktionen zur Sitzungskontrolle.

Arbeitslast ist bestimmt durch die Menge der im DV-System eintreffenden Arbeitsaufträge. In Batch-Systemen ist dies die Anzahl der ankommenden Rechenaufträge in verschiedenen Anforderungsklassen, meist nach CPU-Zeit-Bedarf, Haupt- und Hilfsspeicher-Anforderungen, sowie Ein- und Ausgabemengen aufgegliedert. In Timesharing-Systemen werden

¹ FERRARI: Computer Systems Performance..., <FERR78>, Kap. 1.4.

² SVOBODOVA: Computer Performance Measurement..., <SVOB76>, Kap. 2.

alle Lastelemente (Transaktionen) in Form von Kommandos einer für dieses spezielle Timesharing-System realisierten Kommando-Sprache angenommen.

Die Betriebs- oder Arbeitslast des Timesharing-Systems wird durch die Anzahl und Art der über das Terminalnetz an das Rechnersystem gesendeten Transaktionen bestimmt. Es muss hier betont werden, dass ein Unterschied zwischen der Rate der ankommenden Transaktionen und der Rate der bearbeiteten Transaktion bestehen kann. (die Zahl der bearbeiteten Transaktionen pro Zeiteinheit bestimmt die Systemleistung, siehe Kap. 4.1.). Die zeitliche Verteilung der Ankünfte der Transaktionen bestimmt mit, wie schnell Transaktionen bearbeitet werden können. Diese Problematik ist von Batch-Systemen mit Multiprogramming-Betrieb bekannt.

Die Bearbeitung der Last verursacht eine Belegung der Hard- und Software-Komponenten des Systems. Der Anteil der Benutzungszeit eines Betriebsmittels an der Messzeit ist seine Auslastung.

Die Auslastung der Komponenten bestimmt den Servicegrad, der für den TS-Anwender die entscheidende Leistungsfunktion ist. Sind es im Batch die Durchlauf- oder die Jobzeit, so ist es im Timesharing allein die Antwortzeit³. Diese besitzt eine statistische Verteilung, die abhängig ist von der Ankunftsverteilung, der Art und Anzahl der Transaktionen, von Anzahl und Bedienzeit der Hardware-Komponenten, ferner von den Vergabe-Algorithmen für die Betriebsmittel. Misst man die Antwortzeiten SRT über ein Beobachtungsintervall T und berechnet die relativen Häufigkeiten der Antwortzeit, so lässt sich daraus die Wahrscheinlichkeit ableiten, mit der - unter Voraussetzung der Repräsentativität dieser für ein Beobachtungsintervall ermittelten Verteilung - die Antwortzeit SRT auch in Zukunft einen vorgegebenen Wert SRT_0 nicht überschreitet (siehe Abb. 2.1.).

³ OSSWALD: Leistungsvermögensanalyse..., <OSSW73>, S. 58-63.

Obwohl die Form der Verteilung der Antwortzeit (ebenso wie die der Zwischenankunftszeiten, 'inter-arrival times') wichtig für die Leistungsbewertung verschiedener Funktionen in gleichen Zeitintervallen und gleicher Funktionen zu verschiedenen Zeiten ist, wird in der Praxis vor allem mit dem arithmetischen Mittelwert der Antwortzeit gearbeitet, da die Antwortzeit-Verteilung nur durch aufwendige mikroskopische Messungen (siehe 2.3.) erfasst werden kann. Abb. 2.2. zeigt die gemessenen Summenhäufigkeiten der Zwischenankunfts-, der Antwort- und der Transaktionszykluszeiten (s. Abb. 2.10., S. 32).

Der Leistungsparameter 'Funktion', das sind die funktionellen Möglichkeiten eines Timesharing-Systems (S. 2 u.S. 16) wird bei einer detaillierten Leistungsanalyse i.a. nicht weiter berücksichtigt, da sich bei einem fest vorgegebenen System die Funktionen i.a. nicht ändern. Ausnahmen bilden die Neubeschaffung, der Entwurf und die Entwicklung eines DV-Systems. So wird auch in dieser Arbeit - ausser in den Kap. 5. zur Erfassung und Beschreibung und in Kap. 6. zur funktionellen Nachbildung der Reallast - der Begriff der Funktion in diesem Sinne (als qualitatives Mass) nicht weiter verwendet.

Die Kontrollvariablen⁴ eines Systems sind die leistungs- - bezogenen quantifizierenden Grössen für eine vorgegebene⁵ Fragestellung. FERRARI⁶ erwähnt die Verwendung von subjektiv wertenden Kontrollvariablen, rät aber zur Verwendung solcher technischer Grössen, die 'objektiver, möglicherweise quantifizierbar' sind. M. E. muss an die Kontrollvariablen unbedingt die Forderung der Messbarkeit (im technischen Sinne) gestellt werden.

Zusätzlich zur Messbarkeit ist jedoch noch die Forderung nach einem vertretbaren Aufwand (für die Messbarkeit) für den praktischen Einsatz unabdingbar. So werden Anforderungen an die Messdichte und

⁴ BELL: Performance Evaluation..., <BELL78>.

⁵ BELL/BOEHM/WATSON: Framework..., <BELL72>, S. 1141-1154.

BELL/BOEHM: Issues in Computer Performance..., <BELL75>, S. 4-39.

BELL: Managing Computer Performance..., <BELL76>, S. 21-27.

⁶ FERRARI: Computer Systems Performance..., <FERR78>, Kap. 1.4.

Abbildung 2.1.: Verteilungskurve der Systemantwortzeiten.

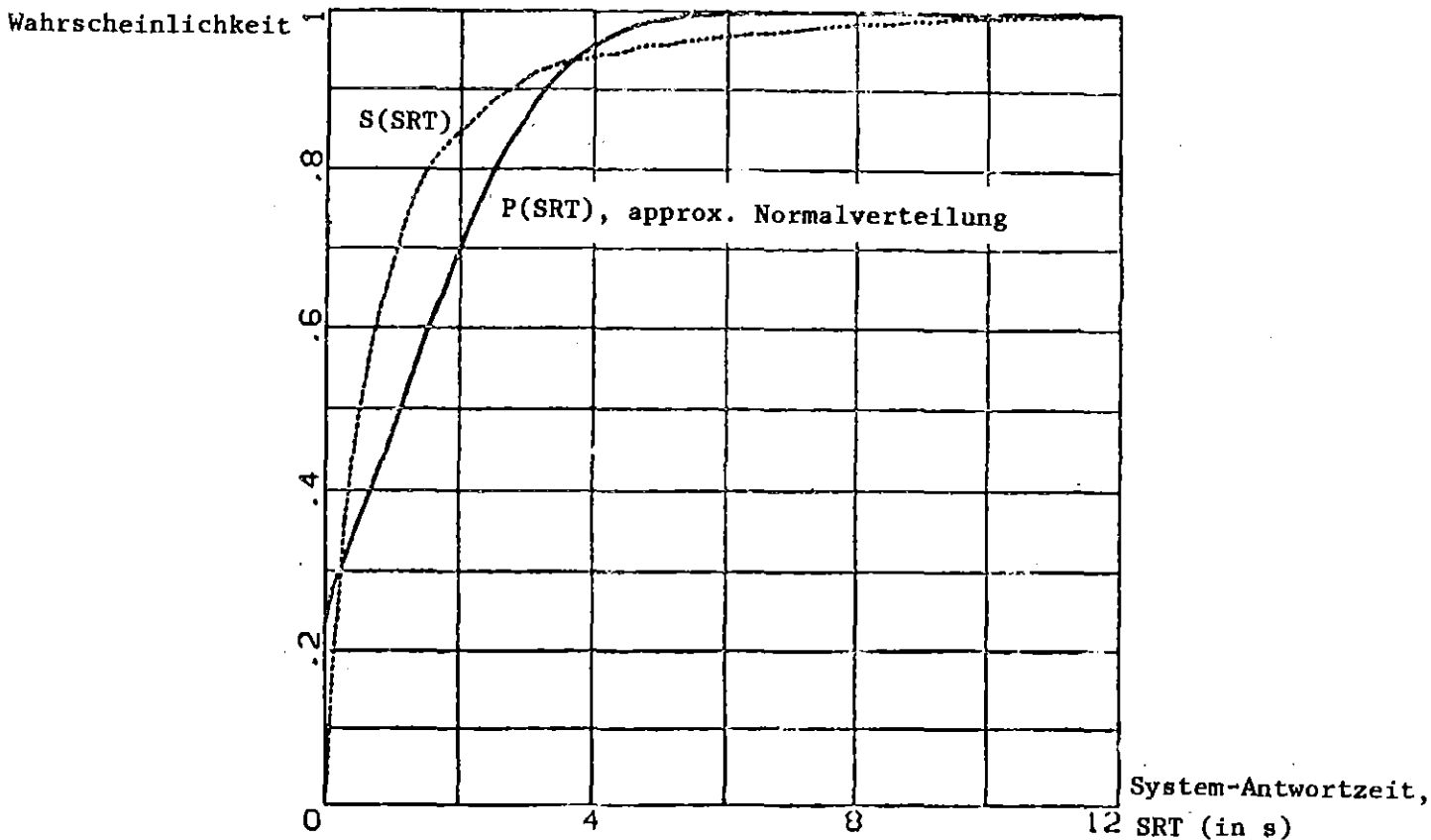
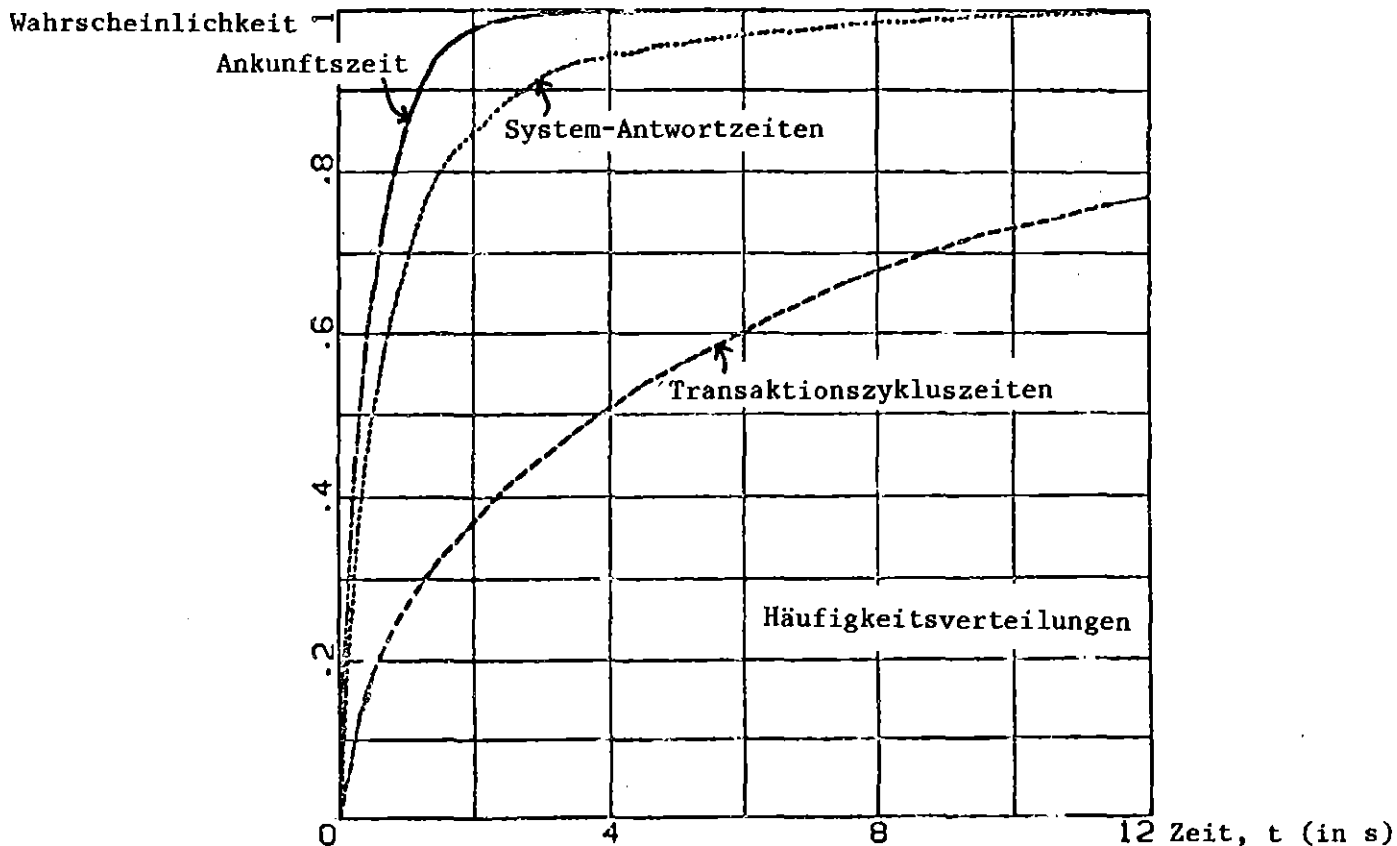


Abbildung 2.2.: Verteilungskurve der Ankunft-, Antwort-
Transaktionszykluszeiten.



Häufigkeit begrenzt werden. System(-Overhead)-, personal- und kosten- aufwendige (meist mikroskopische) Messungen müssen von der Messdauer und der Anzahl ihrer Wiederholungen eingeschränkt bleiben oder Spezialstudien vorbehalten bleiben (siehe Kap. 2.1. und 2.2.). Aus diesem Grunde wurden für die vorliegende Arbeit nur solche Kontrollvariablen in Betracht gezogen, die mit den betrieblich eingesetzten regulären Accounting-Verfahren⁷ oder mit einem im obigen Sinne einsetzbaren Messgerät, z.B. einem Standard-Software-Monitor⁸, erfasst werden.

Für mikroskopische Messungen (Kap. 2.3) können zusätzlich ereignisgetriebene Monitore, z.B.⁹, eingesetzt werden. Der System-Overhead, der mit ihrem Betrieb einhergeht, und die grossen und detaillierten Ausgabe-Datenmengen sowie ihre Handhabung lassen ihren Einsatz nur für kurze Zeitintervalle oder zur Klärung spezieller Fragen zu. Laut Literatur, z.B. in^{10, 11}, haben sich im praktischen Einsatz die in Tab. 2.1. angeführten Kontrollvariablen bereits bewährt.

In vielen TS-Systemen kann ein neuer Transaktionszyklus (s. S. 32) erst von einem Benutzer begonnen werden, wenn der vorhergehende abgeschlossen ist. Transaktionsankunftsrate r und Transaktions-(abfertigungs-)rate TRX sind dann, in nicht zu kleinen Zeitintervallen, gleich; dies soll im folgenden vorausgesetzt werden.

Für den Einsatz in der betrieblichen Praxis muss die Anzahl der Kontrollvariablen auf eine kleine Zahl beschränkt werden. Die Auswahl findet durch schrittweise multiple Regression¹² statt, mit deren Hilfe die signifikanten Einflussgrössen (Regressoren) aus den last- und auslastungsbezogenen KV bzgl. der Zielgrösse (Regressand) System-Antwortzeit SRT (vgl. Kap. 7.3., S. 114-121) ausgewählt werden. Mit seiner Hilfe

⁷ IBM: System Management Facility (SMF), <IBMO76.5>.

⁸ IBM: MVS Resource Measurement Fac. (RMF), <IBMO77.1>.

⁹ IBM: General Trace Fac. (GTF), in: Service Aids, <IBMO76.6>, S. 9-38.

¹⁰ FERRARI: Comp. Syst. Perf., <FERR78>.

¹¹ SVOBODOVA: Comp. Perf. Measurement ..., <SVOB76>, S. 16-22.

¹² DRAPER/SMITH: Applied Regression Analysis, <DRAP66>.

¹³ FERRARI: Comp. Syst. Perf., <FERR78>, Kap. 1.4.

wird eine Anzahl von KV definiert, die nach FERRARI¹³ die primären Kontrollvariablen genannt werden sollen. Es sind dies:

- die Anzahl (u) der (gleichzeitigen) Timesharing-Sitzungen (Summe der system-aktiven und der benutzer-aktiven Sitzungen),
- die Transaktionsrate (TRX, in 1/s),
- die Gesamtauslastung der CPU (CPU, in % der Messzeit),
- die Gesamt-I/O-Rate (SIO, in 1/s),
- die Auslastung der CPU durch Timesharing (TS-CPU, in %),
- die I/O-Rate durch Timesharing (TS-IO, in 1/s),
- der TS-Concurrency-Level (TS-CCL, Anz. der gleichzeitig vom System bearbeiteten Timesharing-Sitzungen, system-aktive Sitzungen, zur Definition vgl. Kap. 4.1., S. 52) und
- die System-Antwortzeit (SRT, in s),

Das Hauptspeicher-Referenz-Verhalten (Paging Rate, Hauptspeicher-auslastung) erscheint aufgrund homogener Speicheranforderungen der Last nicht explizit, sondern wird durch die KV SIO und TS-CCL eingebracht.

TABELLE 2.1.: Kontrollvariable.

A. Last-bezogen (Aktionsvariable, s. S. 22):

- u - Anzahl der (gleichzeitigen) TS-Sitzungen,
- r - Transaktionsankunftsrate (in 1/s),

B. Auslastungs-bezogen (Reaktionsvariable, s. S. 22):

- CPU - Gesamt-Auslastung der CPU (in % der Messzeit),
- SIO - Gesamt-I/O-Rate (in 1/s),
- TS-CPU - Auslastung der CPU durch TS (in %),
- TS-IO - I/O-Rate durch TS (in 1/s),
- TS-CCL - TS-Concurrency-Level (Anz. d. gleichzeitig vom System bearbeiteten TS-Sitzungen; Def. s.S. 52),
- PGR - Häufigkeit des Seitenwechsels (Pag. Rate, 1/s),
- RSP - Auslastung des realen Hauptspeichers (in %),
- HSP - Auslastung des Hilfsspeichers (in %),
- CIO_n - I/O-Rate des n-ten Datenkanals (in 1/s),
- DIO_m - I/O-Rate des m-ten on-line Datenträgers (in 1/s),
- CBY_n - Auslastung des n-ten Datenkanals (in %),
- DBY_m - Auslastung des m-ten on-line Datenträgers (in %),

C. Leistungs-bezogen:

- SRT - System-Antwortzeit (System Resp. Time, SRT, in s),
- TRX - Transaktionsrate (Durchsatz, in 1/s).

Die Werte der Kontrollvariablen werden von einem Software-Monitor betriebsbegleitend erfasst und gemeinsam mit den Accounting-Daten aufgezeichnet. Sie werden regelmässig zeitlich geordnet und in einer Leistungsdatenbank zu leistungsbezogenen Analysen bereitgehalten. Aus dieser Leistungsdatenbank (LDB) werden die Werte für die betriebslast-beschreibenden Variablen extrahiert, die die Zielwerte bei der Approximation durch künstliche Lasten (dynamischer Benchmark, Pseudo-Benchmark, Kap. 6.2., S. 75) darstellen. Die LDB dient jedoch auch der Sammlung von Leistungsdaten zur statistischen Analyse mit Regressionsmodellen (Kap. 7.3., S. 114) sowie der Bestimmung der betrieblichen Systemleistung $L_B(u)$ in Kap. 7.2., S. 110). Der Aufbau der Leistungsdatenbank ist im Anhang, Kap. 9.2., S. 165 ff.) beschrieben.

Für den RZ-Betrieb gewährleistet die Überwachung der Kontrollvariablenwerte dann die Einhaltung betrieblicher Vorgaben, wenn bei deren gehäufter Überschreitung Massnahmen zur Leistungsverbesserung eingeleitet werden. So muss eine über grössere Zeiträume ansteigende System-Antwortzeit unmittelbar Anstoss für die Suche nach antwortzeitverbessernden Massnahmen sein. Aber auch die Beobachtung von ungleichmässigen Auslastungen der einzelnen Datenkanäle kann Grund für eine System-Analyse sein, die z.B. die ungünstige Verteilung von Datenbeständen auf einzelne Datenträger ergibt.

Die Kontrollvariablen spiegeln Aktionen und System-Reaktionen wieder. Die Aktions-Variablen sind direkt ladungsabhängig (TRX, u), die Reaktions-Variablen sind nur indirekt ladungsabhängig (Auslastungen, TS-CCL, SRT werden durch die 'Reaktion' der DVA bestimmt). Stellgrössen sind (in Grenzen) ladungsunabhängig (z.B. Multiprogramming-Grad im OS/VS2 MVS).

Die Korrelationen zwischen den verschiedenen Kontrollvariablen sind unterschiedlich. Es existieren stark unkorrelierte Variablenpaare und es gibt solche Paare, deren Korrelationskoeffizient dicht bei 1 liegt. Bei den letzteren lassen die Messwerte im wesentlichen lineare Abhängigkeit erkennen. Hochkorrelierte Kontrollvariablenwerte eignen sich stets zur Bildung von Bewertungshypothesen, kausale Zusammenhänge lassen sich allein aus hohen Korrelationen **n i c h t** ableiten.

Alle Variablenpaare aus der Gruppe der primären Kontrollvariablen, mit hoher Korrelation (Korrelationskoeffizient $r > 0.75$ für eine einwöchige Messperiode) sind in Tabelle 2.2. aufgeführt.

Bemerkung: Dem Korrelationskoeffizienten $r_0 = 0.75$ entspricht das Bestimmtheitsmass $B_0 = 56.25\%$; dies bedeutet, dass sich 56.25% der Gesamtstreuung der Kontrollvariablen-Wertepaare durch die lineare Regression erklären lassen).

<u>TABELLE 2.2.: Kontrollvariablen-Paare mit hoher Korrelation</u>			
Kontrollvar.-Paar	Korr.-Koeff.	Best.-Mass	Abbildung
TS-CCL, SIO	0.92	85 %	2.3.
u, TS-IO	0.83	69 %	2.4.
TRX, TS-IO	0.83	69 %	2.5.
u, TS-CCL	0.82	67 %	2.6.
TS-CCL, SRT	0.79	62 %	2.7.
u, SIO	0.78	61 %	2.8.
SIO, SRT	0.78	61 %	2.9.

Messwerte aus der LDB, vom 1. - 8.3.1979, werktags von von 8.00-17.00 Uhr, Systemkonfiguration 4, S. 164.

Die linearen Regressionen zu diesen hoch-korrelierten Variablenpaaren sind in den Abb. 2.3. bis 2.9. zusammen mit ihren 95%-Vertrauensbereichen¹⁴ grafisch dargestellt. Als Messpunkte sind die über die Woche gemittelten Werte der Kontrollvariablen dargestellt (siehe 'Datenreduktion durch Zusammenfassung', Kap. 2.5.1., S. 33.).

¹⁴ - SACHS: Angewandte Statistik, <SACH78>, S. 342.

Abbildung 2.3.: I/O-Rate (SIO) und TS-Concurrency-Level (TS-CCL).

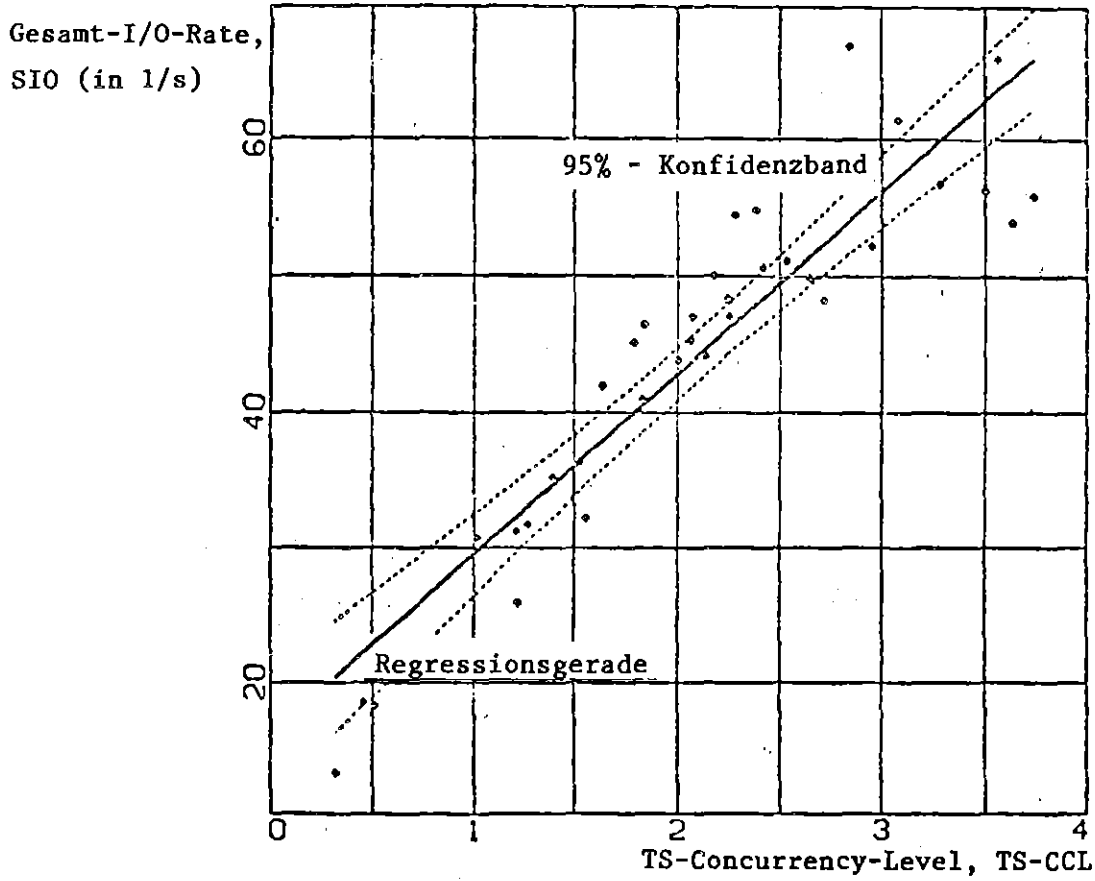


Abbildung 2.4.: TS-I/O-Rate (TS-IO) und Anzahl der Terminals (u).

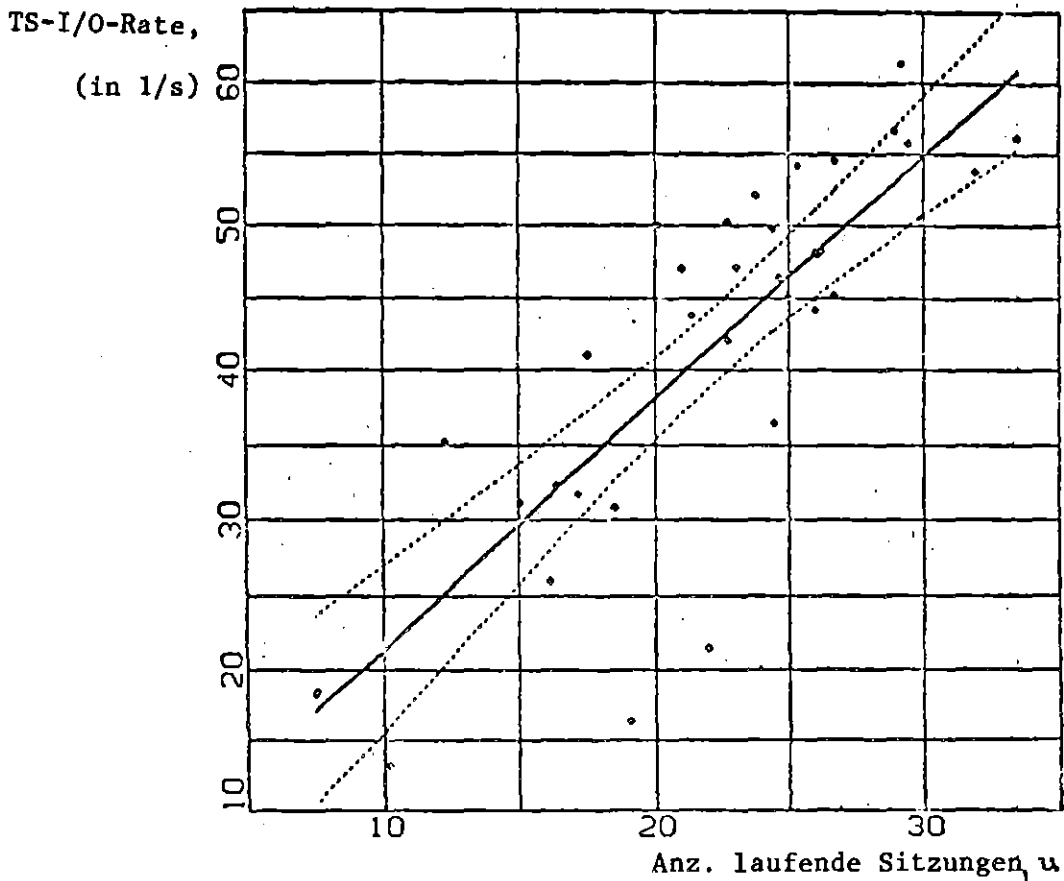


Abbildung 2.5.: TS-I/O-Rate (TS-IO) und Transaktionsrate (TRX).

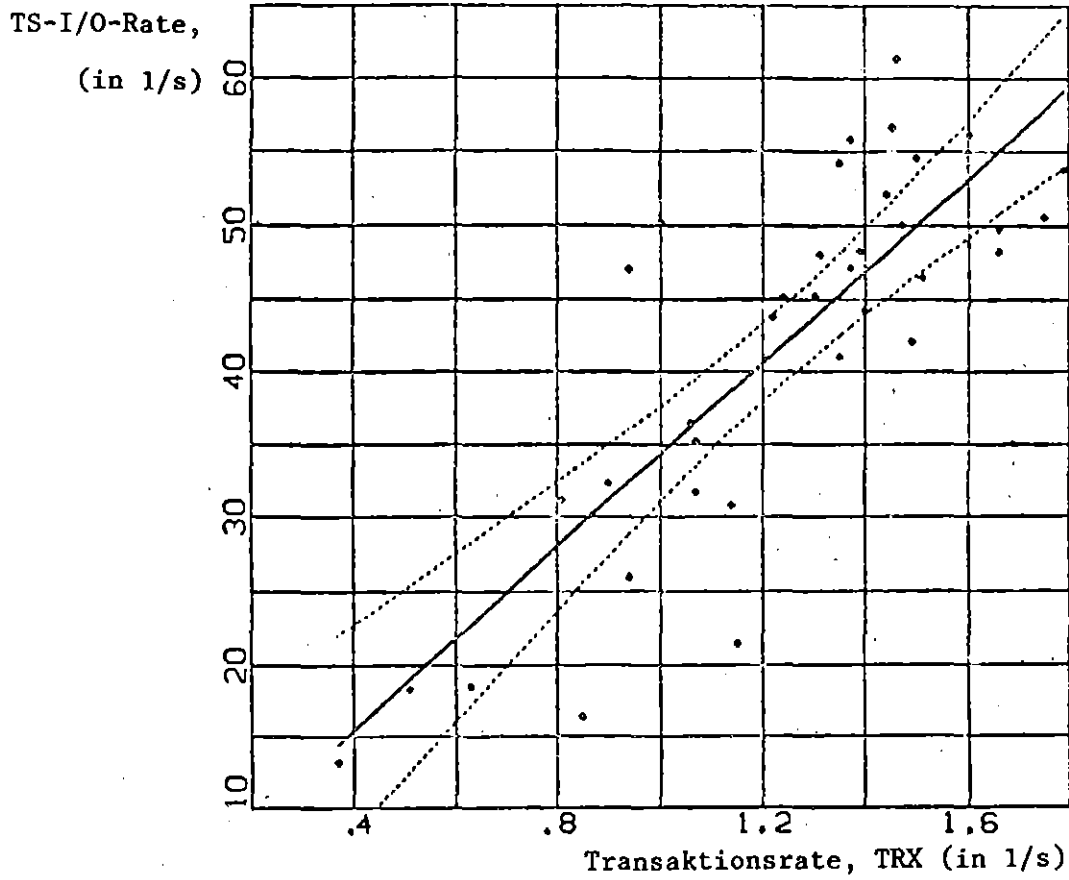


Abbildung 2.6.: TS-Concurrency-Level (TS-CCL) und Anz. der Sitzungen (u).
Sitzungen u.

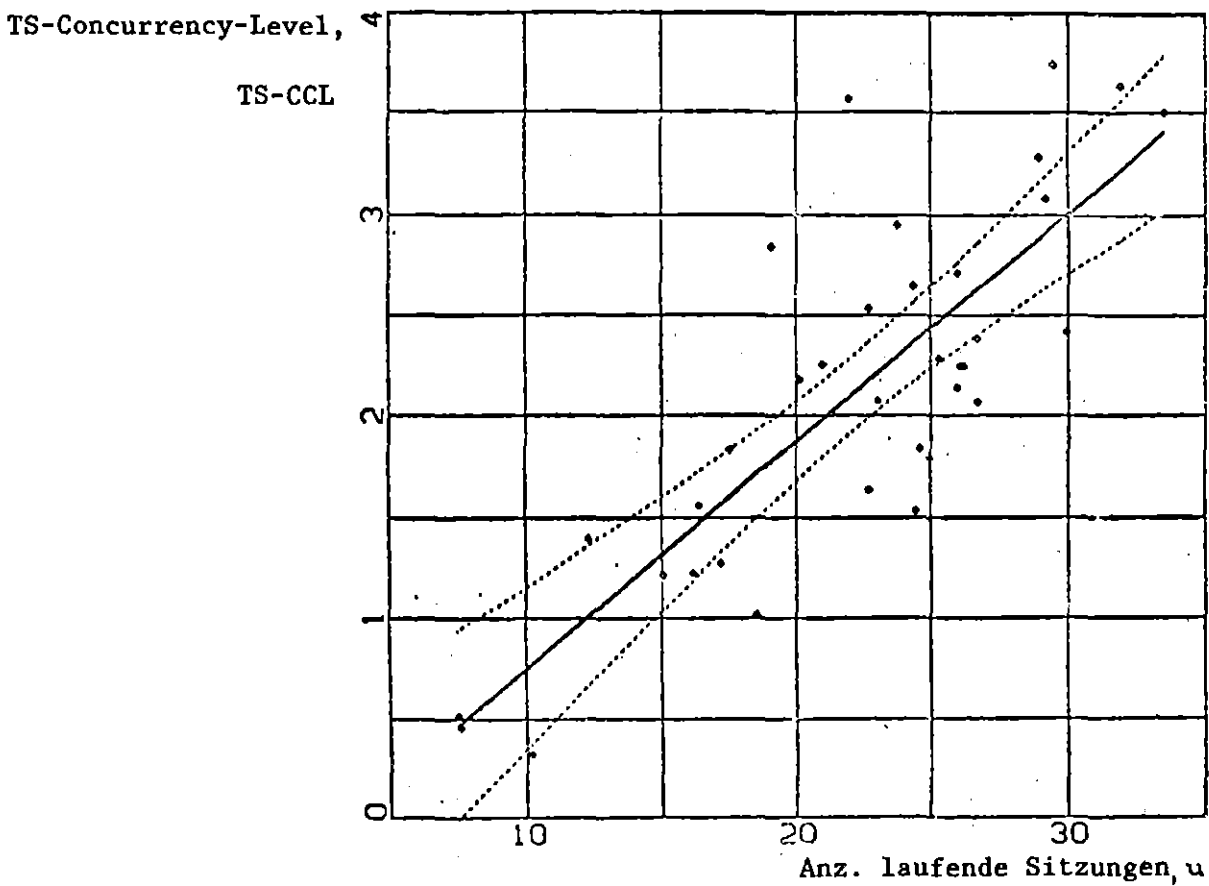


Abbildung 2.7. Antwortzeit (SRT) und TS-Concurrency-Level (TS-CCL).

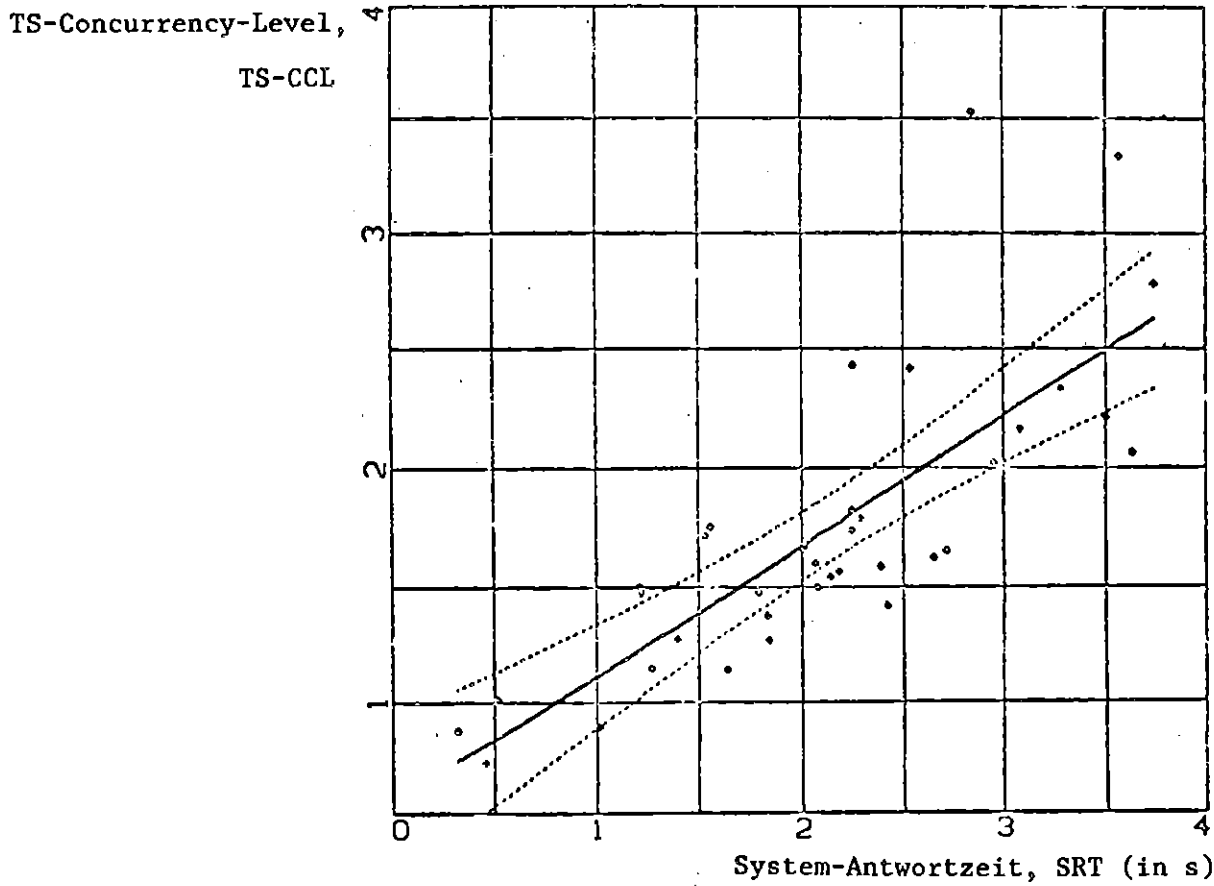


Abbildung 2.8.: I/O-Rate (SIO) und Anzahl der Terminals (u).

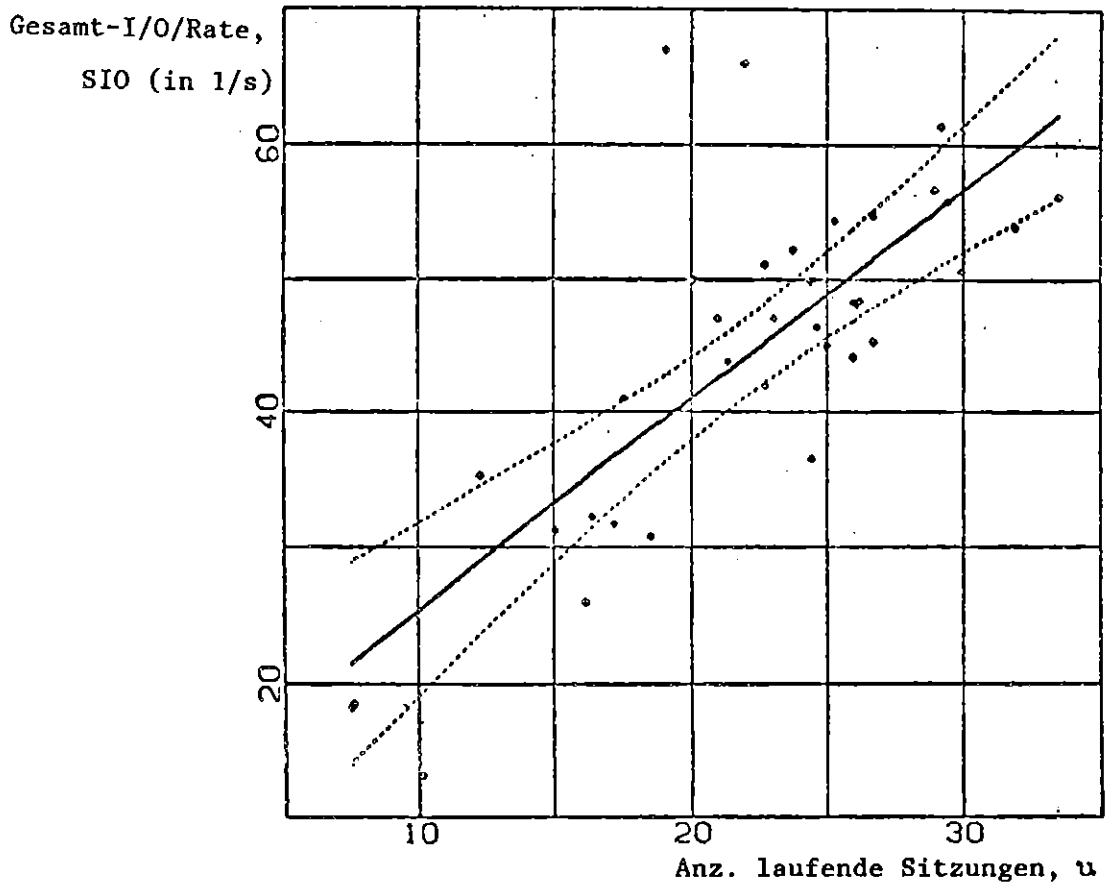
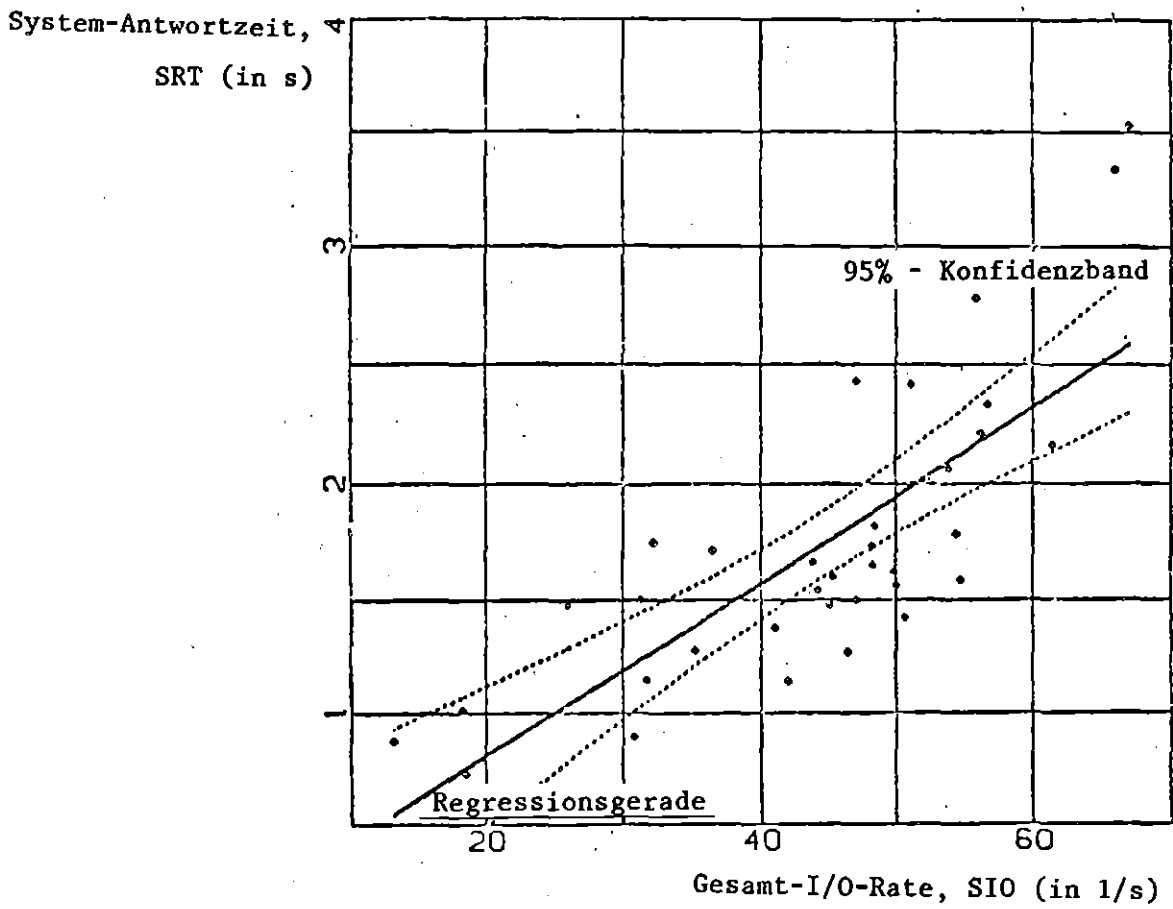


Abbildung 2.9.: Antwortzeit (SRT) und I/O-Rate (SIO):



2.2. Makroskopische Perioden und Messungen.

Alle Messungen, deren aufgezeichnete Werte keine Einzelereignisse mehr repräsentieren, sondern bereits reduzierte oder zusammengefasste Werte, werden makroskopische Messungen genannt. Hierfür werden in vorgegebenen Stichproben-Zeitabständen Messungen vorgenommen oder Ereignisse gezählt, aufaddiert und in besonderen Speichern gesammelt. Am Ende eines Berichtszeitraumes (Aufzeichnungsperiode) werden alle in diesem Zeitraum akkumulierten Werte zu einem Mittelwert (gfls. auch noch Standardabweichung) zusammengefasst, der diesen Zeitraum repräsentiert. Nur dieser Wert steht nach dem Aufzeichnungszeitraum noch zur Verfügung, die eigentlichen Messdaten werden im darauffolgenden Zeitraum mit den neuen Messwerten überschrieben.

Als ein Beispiel für makroskopische Messungen soll die Datenerfassung der Resource Measurement Facility (RMF)¹⁵ angeführt werden. Die Stichproben- und Aufzeichnungsintervalle sind parametrierbar (Tab. 2.3.). Der Betriebsoverhead beträgt bei den in Tab. 2.3. angegebenen Parametern weniger als 0.1% für den CPU-Zeit-Verbrauch und weniger als 0.1 I/O-Operationen pro Sekunde (s. Tabelle 2.4.).

Die Kontrollvariablenwerte der Leistungsdatenbank werden durch betriebsbegleitende makroskopische Messungen während der gesamten Betriebszeit des Timesharing-Systems erzeugt.

2.3. Mikroskopische Messungen.

Zur vollständigen Beschreibung der Verteilung des Funktionsspektrums des Timesharing-Systems oder zur Aufzeichnung der Antwortzeitverteilung müssen für die Analyse wesentliche Einzelereignisse erfasst werden. Dies hat immer dann zu geschehen, wenn die Aufzeichnung der makroskopische Messungen von der Auflösung her nicht ausreicht. Solche mikroskopischen Messungen werden mit ereignis-getriebenen Monitoren durchgeführt.

¹⁵ - IBM: OS/VS2 MVS RMF Ref. and User's Guide, <IBMO77.2>.

Ein (Software-) Monitor für mikroskopische Messungen muss

- * den Beginn einer Transaktion,
- * die eingegebenen Zeichen, mindestens jedoch
- * den Typ des aufgerufenen Kommandos,
- * die Art der Ausführung der Transaktion (z.B. Prompting wegen fehlender oder fehlerhafter Operanden, vorzeitigen Abbruch durch den Benutzer),
- * den Beginn und Ende der System-Bearbeitungszeit,
- * die Beginn und Ende der Ausgabe,
- * die Ausgabemenge und
- * das Ende der Transaktion

TABELLE 2.3.: Parametrierung des Software-Monitors Resource Measurement Facility (RMF)¹⁶ für makroskopische Messungen.

Steuerparameter	Bedeutung
CPU	COLLECT CPU ACTIVITY DATA
CHAN	COLLECT CHANNEL ACTIVITY DATA
DEVICE(DASD)	COLLECT DIRECT ACCESS DEVICE ACTIVITY DATA
DEVICE(TAPES)	COLLECT TAPE-DRIVE ACTIVITY DATA
ENQ(DETAIL)	COLLECT DETAILED ENQUEUE DATA
WKLD(PERIOD)	COLLECT WORKLOAD MANAGER DATA, REPORT AT PERIOD LEVEL
PAGESP	COLLECT PAGE-/SWAP-DATASET ACTIVITY DATA
PAGING	COLLECT PAGING/SWAPPING DATA
RECORD	RECORD DATA TO SMF FILES
CYCLE(1000)	USE 1000 MS AS A SAMPLING INTERVAL ^{*)}
INTERVAL(15M)	USE 15 MINS AS A REPORTING INTERVAL ^{*)}

*) Mit diesen Parametern erzeugt der Monitor pro Berichtsintervall 900 Stichproben, deren Mittelwert aufgezeichnet wird. Dieser weicht mit 90% Sicherheit nicht mehr als 3% vom tatsächlichen Mittelwert ab (siehe¹⁶, S. 38.). Vgl. auch in¹⁷, S. 84/85.

¹⁶ - IBM: OS/VS2 MVS RMF Ref. and Users Guide, <IBMO77.2>.

¹⁷ - OSSWALD: Leistungsvermögensanalyse, <OSSW73>.

aufzeichnen. Die Zeitspanne vom Ende der vorhergehenden Transaktion bis zum Beginn der neuen Transaktion heisst Benutzer-Antwortzeit oder Denkzeit (User response time). Die Benutzer-Antwortzeit, die System-Antwortzeit und die Ausgabezeit bilden einen Transaktionszyklus, ihre Summe ist die Transaktionszykluszeit. Die Darstellung eines vollständigen Zyklus findet sich in Abbildung 2.10., S. 32.

TABELLE 2.4.:

Overhead-Vergleich bei makroskopischen und mikroskopischen Messungen.

	makro. Mess. -RMF-	mikro. Mess. -GTF-
CPU-Zeit-Verbrauch	1.3 s (pro 15 min Berichtsint.)	12.5 s (pro 30 min Laufzeit)
	0.014 %	7 % (der Messzeit)
I/O-Rate	12 Datensätze (pro 15 min Berichtsint.)	700 Datensätze (pro 30 min Laufzeit)
	0.01 1/s	0.4 1/s
	0.02 %	0.8 % (der mittl. I/O- Rate v. 50 1/s)

Folgerung für den Einsatz der beiden verglichenen Monitore nach Tab. 2.4.: Die makroskopischen Messungen verursachen einen vernachlässigbaren CPU- und I/O-Overhead (0.014 bzw. 0.02 %); der CPU-Over-

head der mikroskopischen Messungen mit 7% liegt dagegen bereits in einem Bereich, der spürbare Auswirkungen auf das zu messende System hat. Dementsprechend können mikroskopische Messungen nur beschränkt eingesetzt werden.

Definition: Die System-Antwortzeit SRT (kurz: Antwortzeit) ist das Zeitintervall vom Ende der Eingabe bis zum Beginn der Ausgabe für diese Transaktion (erstes Zeichen, vgl. auch Kap. 4.5., S. 61).

Bemerkung: Die Ausgabezeit wird im Gegensatz zu anderen Autoren, z.B. in ¹⁸, nicht zur Antwortzeit gerechnet, da diese vor allem vom Terminaltyp und vom i.a. inhomogenen DV-Leitungsnetz abhängig ist.

Die in Abb. 2.2. (Häufigkeitsverteilungen der Zwischenankunftszeiten t_0 , der Antwortzeit SRT und der Transaktionszykluszeiten (CYT) dargestellten Messwerte wurden mit der General Trace Facility (GTF)¹⁹ gemessen und aufgezeichnet.

2.4. Betriebszyklen.

Für jede Art der Leistungsuntersuchung wird eine Analyse der durch die Aufgabenstellung eingegrenzten Kontrollvariablen Ausgangspunkt sein, da sich hier bereits Hinweise auf den weiteren Weg der Untersuchung ergeben (vgl. Kap. 7.2.). Je nach Untersuchungsziel wird auf verschiedene Zeiträume der in der Leistungsdatenbank aufgezeichneten Kontrollvariablen-Werte zurückgegriffen.

Arbeitslasten sind in der Praxis periodisch mit der Zeit²⁰. Ursache dafür sind die Produktionszyklen der Unternehmung und damit der Arbeitsrhythmus der DV-Benutzer in dieser Organisation²¹ - sofern sie freien

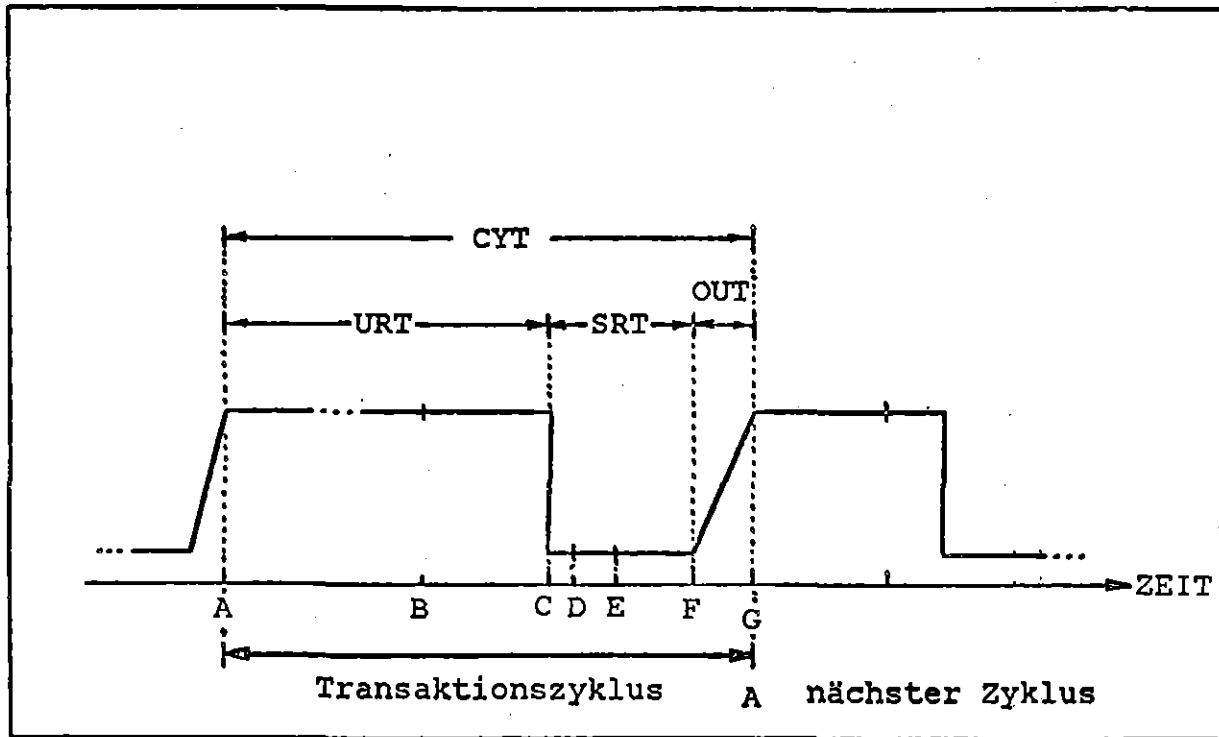
¹⁸ - MÜHLENBEIN: Softwarephysik..., <MUEH77.2>, S. 20.

¹⁹ - IBM: OS/VS2 MVS Service Aids, <IBMO76.5>, S. 9-38.

²⁰ - FERRARI: Computer Syst. Perf. Eval. <FERR76>, S. 222.

²¹ - SCHRIEFER: Complex Workloads ..., <SCHR77>, S. 4.

Abbildung 2.10.: Der vollständige Transaktionszyklus.



$CYT = G - A = \text{Zykluszeit,}$

$URT = C - A = \text{Benutzer-Antwortzeit, Denkzeit,}$

$SRT = F - C = \text{System-Antwortzeit,}$

$OUT = G - F = \text{Ausgabezeit,}$

$UTT = B - A = \text{(eigentliche) Denkzeit,}$

$INT = C - B = \text{Eingabezeit,}$

$RST = D - C = \text{Transaktion-Restore-Zeit,}$

$CRT = E - C = \text{Kommando-Erkennungszeit.}$

Zugang zu den DV-Systemen haben. Dies gilt insbesondere für ein sich als Dienstleistungsbetrieb verstehendes Rechenzentrum in einer Grossforschungsanlage. Die Periodizität zeigt sich sowohl in den aktiven Kontrollvariablen (siehe S. 22) u und TRX als auch in den reaktiven Variablen (wie CPU-Auslastung, I/O-Rate und System-Antwortzeit SRT. Diese zeitlichen Perioden sollen Betriebszyklen genannt werden. In Abb. 2.11. werden die Werte der Kontrollvariablen TS-CCL (TS-Concurrency-Level) über einen Zeitraum von 10 Tagen dargestellt, die Betriebszyklen (Arbeitstage) sind zu sehen.

Beispiele für Betriebszyklen sind die werktäglichen Hochlastzeitintervalle, die durch eine bestimmte Betriebsart bestimmte Betriebsphase (z. B. die 1. Schicht der Werktage) - siehe auch Kap. 3.5. -, ganze Werktage, wöchentliche, monatliche, viertel-, halb- (z.B. Hochschulsemester) und ganzjährige Perioden.

Die Auswahl des Betriebszyklus hängt von der Zielsetzung der Aufgabe ab. Beispiele werden in den Kapiteln 7. und 8. gegeben.

Zum Leistungsnachweis und für die Planung von Kapazitätsänderungen müssen Trendanalysen der durch grössere Betriebszyklen gegebenen Zeitreihen vorgenommen werden. Zur Schwachstellenanalyse werden kürzere Betriebszyklen herangezogen, z.B. einzelne Tage oder Wochen. Zur Datenreduktion durch Auswahl einzelner repräsentativer Betriebszyklen können entweder eine Anzahl gleichartiger Betriebszyklen über einen übergeordneten (= grösseren) Betriebszyklus gemittelt werden (s. unten) oder es werden die Daten eines Zyklus als repräsentativ angesehen (s. 2.5.2.). In beiden Fällen muss die Vernachlässigbarkeit des Trends innerhalb des Repräsentationsintervalls nachgewiesen werden (z.B. durch das in Kap. 2.5.2. angegebene Verfahren).

2.5. Auswahlkriterien repräsentativer Betriebszyklen.

2.5.1. Datenreduktion durch Zusammenfassung.

Die n Messwerte verschiedener Betriebszyklen werden durch Mittelwertbildung zusammengefasst:

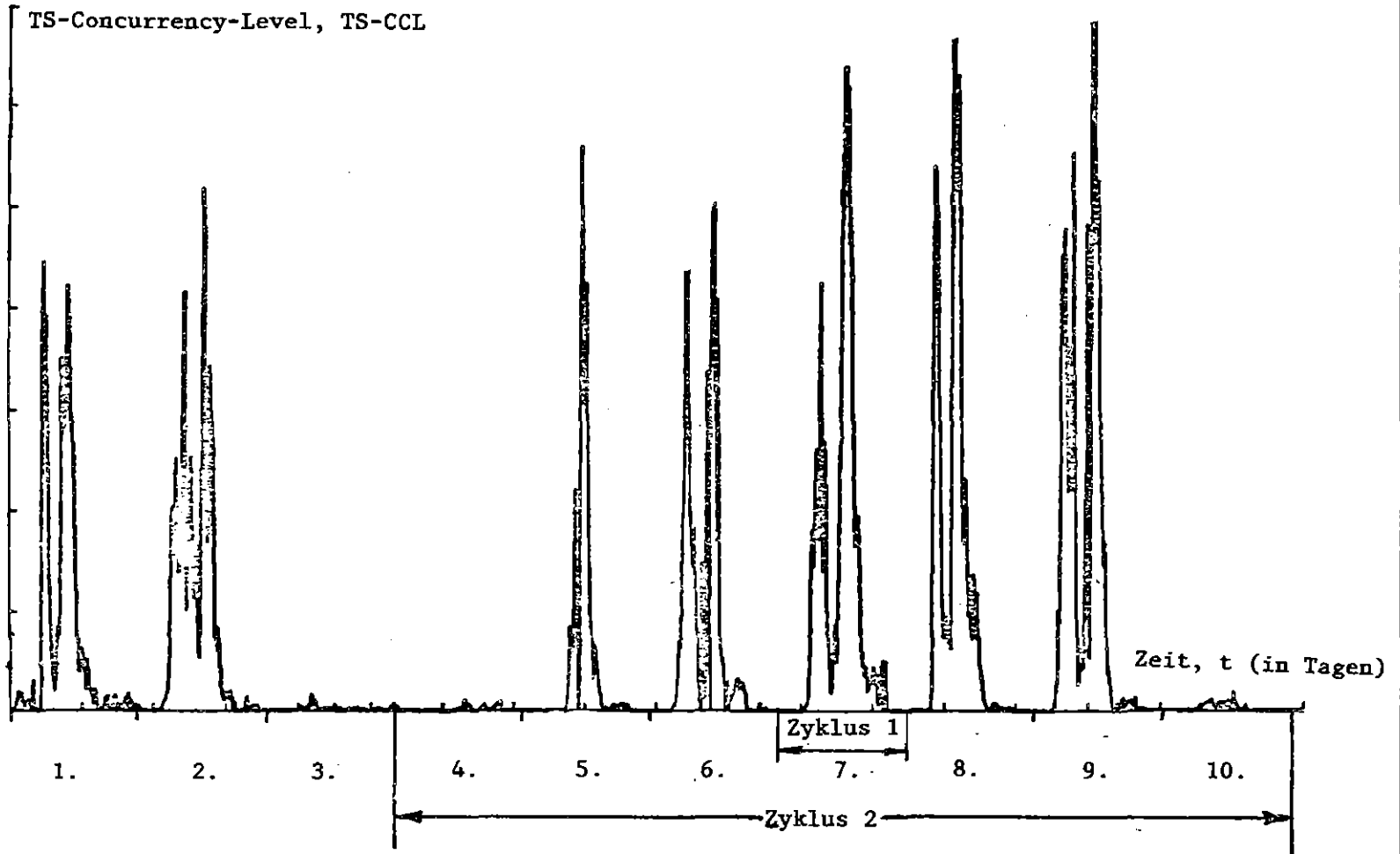


Abbildung 2.11.: Betriebszyklen: Der TS-Concurrency-Level als Beispiel.

$$KV(t_j) := (1/n) \sum_{i=1}^n KV(i, t_j),$$

mit $i = 1, \dots, n$ als Zyklusindex und $j = 1, \dots, m$ als Stichprobenindex. (Anwendungen siehe Kap. 8.2.). Fehlende Werte der Kontrollvariablen (z.B. wegen System-Neustart oder Wartung) während der Betriebszyklen müssen bei der Mittelwertbildung berücksichtigt werden.

2.5.2. Datenreduktion durch Auswahl.

Die Messwerte mehrerer Betriebszyklen können auch nur durch die Werte eines ausgewählten Betriebszyklus repräsentiert werden. Dieser Betriebszyklus soll möglichst vollständig sein, d.h. in der Aufzeichnung seiner Werte sollten keine Lücken vorhanden sein:

$$KV(t_j) := KV(i, t_j),$$

i ist der Index des ausgewählten Betriebszyklus.

Nachweis der Repräsentativität.

Die Verfahren zur Datenreduktion können dann angewendet werden, wenn ein Vergleich der Messwerte je zweier oder mehrerer Zyklen keinen Anhaltspunkt dafür liefert, dass die Messwerte aus verschiedenen Grundgesamtheiten stammen. Der Nachweis der Homogenität erfolgt anhand von Vier- oder Mehrfeldertafeln (siehe z.B.²²). Als Signifikanzniveau für die Homogenität wird stets die 5%-Grenze (also 95% Sicherheitswahrscheinlichkeit) verwendet. Ein Beispiel für den Homogenitätsvergleich findet sich in Tabelle 2.5.

²² - SACHS: Angewandte Statistik, <SACH78>, S. 358, S. 366.

TABELLE 2.5.: Nachweis der Homogenität mehrerer Betriebszyklen.

Name der Kontroll- Variablen	gemeins. Median ^{*)}		$\hat{\chi}^2$	
	Tag	1. Schicht	Tag	1. Schicht
CPU	29.9	82.2	0.00	3.56
I/O	16.3	45.0	0.75	0.00
TS-CPU	1.8	35.2	0.75	0.89
TS-I/O	0.8	44.3	0.75	0.22
TS-CCL	0.1	3.3	0.75	0.89
TRX	0.04	0.9	0.75	2.00
SRT	1.5	3.7	1.33	8.00
<u>Anzahl der Messungen</u>	192	72		

Alle Kontrollvariablen mit einem $\hat{\chi}^2$ -Wert bis zu $\chi^2 = 3.84$ sind mit 95% Sicherheitswahrscheinlichkeit für die beiden Messungen als homogen anzusehen.

*) Verglichen werden je 2 Betriebszyklen: 1. der gesamte Arbeitstag (24 Stdn.) des 16. und des 18.8.1978 und 2. deren 1. Schicht (8.00 bis 17.00 Uhr, 9 Stdn.), Systemkonfiguration 1 im Anhang, Kap. 9.2.

Der gemeinsame Median wird über die Daten aller Betriebszyklen zusammen gebildet (N.B.: Die Mediane der einzelnen Betriebszyklen weichen durchaus voneinander ab.).

3. Klassifizierung der Betriebszustände eines Rechnersystems mit gemischten Betriebsmodi.

Ein rund um die Uhr (d.h. 24 Stunden täglich, einschliesslich der Wochenenden) betriebenes Rechnersystem, das ausschliesslich Timesharing-Dienstleistungen anbietet, wird i.a. während der Nachtstunden und an Wochenenden nicht ausgelastet. Diese Aussage lässt sich schon anhand von Accounting-Daten erbringen. Wenn aus unternehmensspezifischen Gründen TS-Service jedoch durchgehend angeboten werden muss, so führt das aus wirtschaftlichen Gründen zu einem Mischbetrieb ausserhalb der TS-Hauptbetriebszeiten. Während dieser Mischbetriebszeiten werden z.B. auch Batchjobs auf der Timesharing-Anlage gefahren. Das Rechenzentrum kann aber auch aus anderen Gründen (z.B. konfigurationsbedingt) verschiedene Betriebszustände (TS überwiegend, Batch überwiegt) bevorzugen; in solchen Fällen muss das Timesharing mit anderen Betriebsarten koexistieren.

Ziel dieses Kapitels ist es, ein Konzept vorzulegen, mit dem allein aus der Kenntnis der Werte der Kontrollvariablen über einen Betriebszyklus eine Klassifizierung der verschiedenen Betriebszustände innerhalb dieses Betriebszyklus durchzuführen ist. Um die Klassifizierung vornehmen zu können, muss zunächst eine Klassifizierung der Kontrollvariablen hinsichtlich ihres Einflusses auf die Klassifizierung des Betriebszustands unternommen werden. Mit den ausgewählten Kontrollvariablen sollen sich die Betriebszustände eindeutig identifizieren lassen.

Bemerkung: Verwendet man nur e i n e Kontrollvariable, z.B. die Anzahl der laufenden Sitzungen u zur Klassifizierung der Timesharing-Dominanz während eines Zeitintervalls, so deckt sich dies nicht unbedingt mit der praktischen Erfahrung. Diese zeigt, dass erst die Untersuchung mehrerer Kontrollvariablen die Entscheidung ermöglicht, ob während eines Zyklus die eine oder die andere Art der Betriebsweise dominiert hat.

Beide Klassifizierungen werden nach dem Verfahren der wiederholten Cluster - Analyse vorgenommen.

3.1. Klassifizierung der Kontrollvariablen.

Methode: Cluster-Analyse (siehe DUDA/HART¹, ANDERBERG², HARTIGAN³, BOCK⁴) über alle Kombinationen der (primären) Kontrollvariablen (s. Kap. 2.1.).

Es wird eine vollständige hierarchische Cluster-Analyse über alle Messwerte des zu untersuchenden Betriebszyklus (Variablen der Cluster-Analyse) durchgeführt. Die Kombinationen sind die zu klassifizierenden Objekte. Eine Zerlegung des Gesamt-Clusters in Unter-Strukturen (Sub-Clusters) bis zu einer Länge p (Partitionen) ermöglicht eine Untersuchung der Auftretenshäufigkeit der einzelnen Kontrollvariablen (innerhalb der Kontrollvariablen-Kombinationen) in den Sub-Clusters. Kontrollvariablen mit der grössten Häufigkeit in einem Cluster dominieren dieses Cluster. Die Kontrollvariablen mit höchster Summenhäufigkeit bei gleicher Partitionslänge werden zur Klassifizierung der Betriebszustände verwendet.

Cluster-analytische Verfahren sind vereinzelt schon auf dem Gebiet der Leistungsbewertung eingesetzt worden, allerdings mit dem Ziel der Klassifizierung und der Konstruktion von Arbeitslasten oder deren Komponenten, so bei AGRAWALA/MOHR/BRYANT⁵, LANDAU⁶, FANGMEYER, GLODEN/LARISSE⁷, ARTIS⁸ und AGRAWALA/MOHR⁹.

3.1.1. Normierung der Kontrollvariablen.

Da die verschiedenen Kontrollvariablen grössenordnungsmässig

-
- 1 - DUDA/HART: Pattern Recognition..., <DUDA73>.
 - 2 - ANDERBERG: Cluster Analysis ..., <ANDE73>.
 - 3 - HARTIGAN: Cluster Algorithms, <HART75>.
 - 4 - BOCK: Automatische Klassifikation, <BOCK70>.
 - 5 - AGRAWALA/MOHR/BRYANT: An Approach to..., <AGRA76>, S. 18-32.
 - 6 - LANDAU: Clusteranalytische Untersuchungen..., <LAND76>, S. 273-277.
 - 7 - FANGMEYER/GLODEN/LARISSE: An Automatic..., <FANG77>, S. 427-733.
 - 8 - ARTIS: Capacity Planning..., <ARTI78>, S. 25-53.
 - 9 - AGRAWALA/MOHR: Predicting the Workload..., <AGRA78>, S. 465-471.

verschieden sind, wird eine Transformation oder Normierung^{10,11} der Werte durchgeführt. Es wird die Maximum-Normierung verwendet (Division durch den Maximalwert der Kontrollvariablen). Diese Normierung garantiert, dass alle transformierten Werte nicht grösser als 1 werden.

$$KV(j, t_i) := \frac{KV(j, t_i)}{\text{Max } KV(j, t_i)} \leq 1$$

(die t_i sind die Aufzeichnungsintervalle des zu untersuchenden Betriebszyklus ^{*})).

3.1.2. Die Kombinationen der Kontrollvariablen.

Zu den Messwerten der m ($=7$) originären, jedoch bereits normierten Kontrollvariablen werden alle Kombinationen aus diesen hinzugefügt. Alle möglichen Kombinationen der Kontrollvariablen werden deshalb verwendet, weil so die Einflüsse aller Kontrollvariablen aufeinander bewertet werden können und alle Kontrollvariablen dabei formal gleichberechtigt (d.h. in gleicher Anzahl) auftreten.

Diese können als alle aus den Kontrollvariablen gebildete, überhaupt mögliche Zustandsvektoren zur Beschreibung eines Messzeitpunktes t_i interpretiert werden. Das Verfahren erzeugt je j -te Klasse $K_j(m)$ weitere mehrdimensionale 'Kontrollvariable' ($j=2, \dots, m$). Dies sind:

$$K_0 = \sum_{j=1}^m K_j(m) = 2^m - 1 ,$$

^{*}) Zur Durchführung des Verfahrens wurden die Daten des 16.8.1978 ausgewählt, nachdem sie gemäss Kap. 2.5. auf Repräsentativität für den Monat August 1978 für alle verwendeten (7) Kontrollvariablen überprüft wurden (Syst.-Konf. 1, Kap. 9.2., S. 163).

¹⁰ - SPÄTH: Cluster-Analyse-Algorithmen, <SPAE75>, Kap. 2.

¹¹ - DURAN/ODELL: Clusteranalysis, <DURA74>, S. 5.

für $m = 7$ ist $K_0 = 127$ ($K_j(m)$ ist die Anzahl der Kombinationen von m Elementen zur j -ten Klasse ohne Wiederholung).

Die Kontrollmatrix $M_{ki}(j)$ der j -ten Klasse enthält alle aus den verschiedenen 'Kontrollvariablen' erzeugten, normierten Messwerte. Ihre Dimension ist $K_j(m) \times j \times n$ (n ist die Anz. d. Werte aus der LDB). $j=1$ entspricht den originären Kontrollvariablen (oder m ein-dimensionalen Zustandsvektoren). Alle m originären Kontrollvariablen sind mit der gleichen Häufigkeit vertreten.

Aus den Elementen der Kontrollmatrix $M_{ki}(j)$ gewinnt man für einen festen Messpunkt t_j die Projektion seines Zustandsvektors im m -dimension. Raum (gebildet durch alle m Kontrollvariablen) in den j -dimensionalen Unterraum. Der Betrag der Zeilenvektoren gibt den Abstand zum Koordinaten-Ursprung an:

$$s_{kij}(m) = \sqrt{\sum_{l=1_k}^{l_k+j-1} KV^2(l, t_i)}$$

mit $i=1, \dots, n$ (Messindex), $j=1, \dots, m$ (Klasse), $k=1, \dots, K_j(m)$ (Klassenindex) und 1_k ist der Beginn der k -ten Index-Permutation der Länge j .
Mit

$$A_{ijk} = (1/j) \cdot s_{kji}(m) = A_{ri},$$

mit $r=1, \dots, K_0$ (=127) Objekte im $i=1, \dots, n$ (=96) dim. Raum definiert. Alle A_{ri} (s. Abb. 3.1., S. 41) sind wegen der Normierung ≤ 1 .

3.1.3. Ähnlichkeitsmatrizen.

Durch die Cluster-Analyse soll eine Klassifizierung der Objekte aufgrund ihrer Ähnlichkeit vorgenommen werden. Objekte mit grösserer Ähnlichkeit liegen (im n -dim. Raum) dichter zusammen als Objekte mit geringerer Ähnlichkeit. Dazu wird hier als Ähnlichkeitsmass die

euklidische Norm verwendet:

$$d_{rs} = \sqrt{\sum_{i=1}^n (A_{ri} - A_{si})^2}$$

Abbildung 3.1.: Datenmatrix zur Klassifizierung der Kontr.-Var.

	1	...	i	...	n (=96)	Messzeiten
1						
...						
7						
8						
...						
28						
29						
...						
63						
64						
...						
98						
99						
...						
119						
120						
...						
126						
$K_0 = 127$						
Objekte						

mit r und $s=1, \dots, K_0$. In diesem Sinne bedeutet ein kleiner Abstand eine grosse Ähnlichkeit. In manchen Beispielen werden andere Ähnlichkeitsmasse verwendet¹².

Die Berechnung aller d_{jk} liefert die (quadratische) Ähnlichkeitsmatrix D . Sie legt die Ähnlichkeit zwischen alle K_0 Objekten fest.

Bemerkung: Wegen der Kommutativität der Ähnlichkeit ist die Ähnlichkeitsmatrix symmetrisch. Linearisiert man die obere Dreiecksmatrix von D zeilenweise, so wird der entstandene Vektor Ähnlichkeitsvektor genannt.

3.1.4. Hierarchische Cluster-Analyse.

Das hier angewendete agglomerative Verfahren der Clusterbildung wird z. B. bei SPÄTH¹³ beschrieben. Es besteht darin, die zwei Objekte mit grösster Ähnlichkeit zu einem Cluster zusammenzufassen und daraus ein neues Objekt (mit den Schwerpunktskoordinaten der beiden Einzelobjekte) zu bilden. Dieser Prozess wird wiederholt bis eine vollständige Hierarchie entstanden ist. Die niedrigste Stufe besteht stets aus allen originären Objekten (Terminale), die höchste Stufe definiert eine Punktvolke oder Cluster, die alle anderen Sub-Cluster enthält. Die niedrigste Stufe enthält die feinste Klasseneinteilung, sie zeigt jedoch keine Gruppierungen, die höchste Stufe enthält die grösste Klasseneinteilung, lässt jedoch keine Differenzierung mehr zu.

Eine Darstellung der Cluster-Hierarchie erfolgt mit dem Dendogramm (siehe ^{14,15}). In Abbildung 3.2. werden Teile des Dendogramms zur Auswahl der Kontrollvariablen gezeigt.

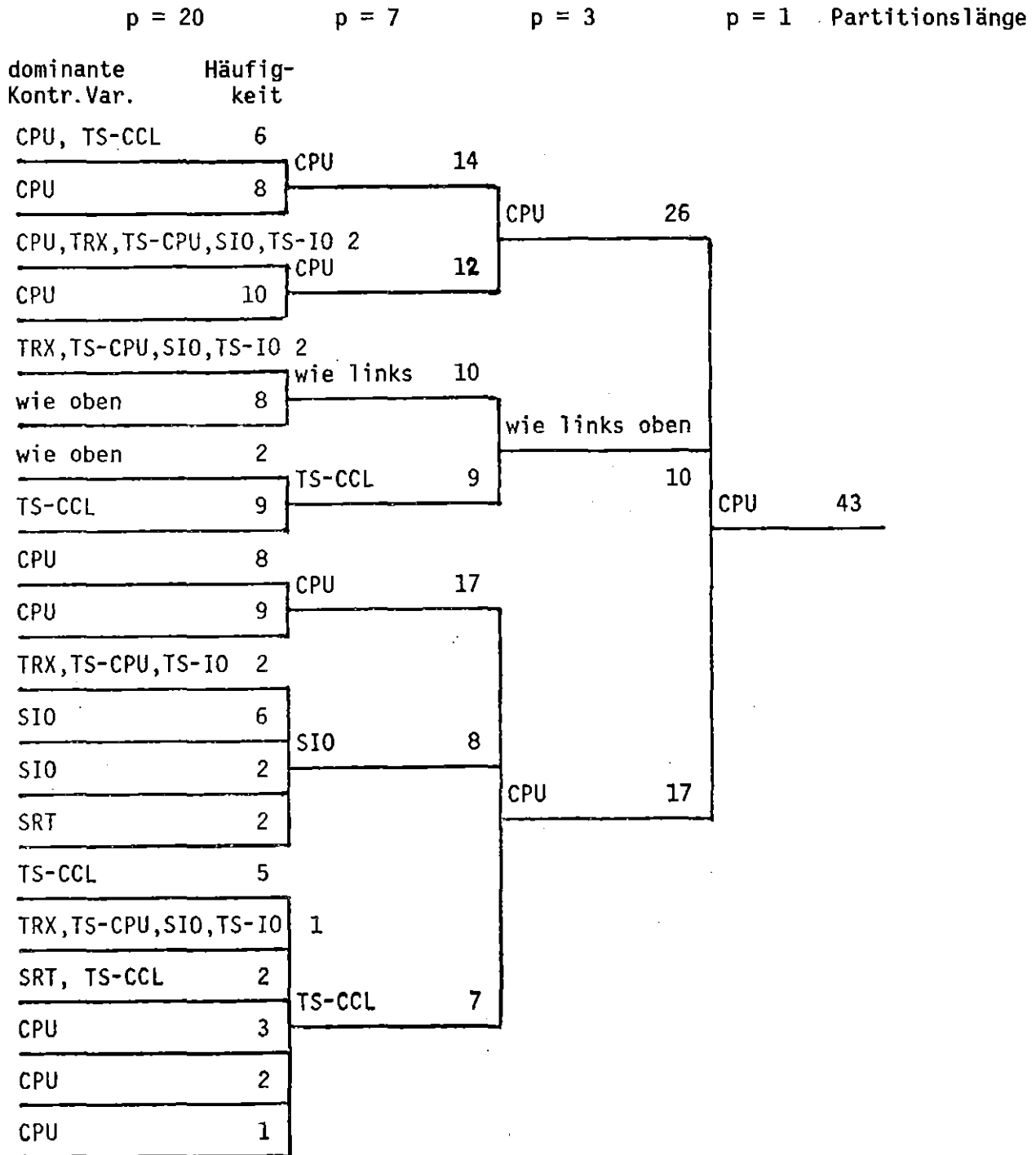
¹² - DURAN/ODELL: Clusteranalysis, <DURA74>, S. 5 und S. 19.

¹³ - SPÄTH: Cluster-Analyse-Algorithmen, <SPAE75>, S. 162.

¹⁴ - LANDAU: Clusteranalytische Untersuchungen, <LAND76>, S. 273-277.

¹⁵ - DURAN/ODELL: Clusteranalysis, <DURA74>, S. 73.

Abbildung 3.2.: Partitionen der Längen = 20, Teil-Dendogramm.



Erläuterungen zum Dendogramm auf S. 44.

3.1.5. Analyse der Cluster und Auswahl der Kontrollvariablen.

Eine Untersuchung der aus den Messdaten (s. S. 39) aufgebauten Cluster-Hierarchie soll bei einer Partition der Länge $p = 20$ beginnen^{*)} (s. Abb. 3.2., S. 43). Unter einer Partition der Länge p versteht man eine Auftrennung des Gesamt-Clusters ($p=1$), sodass es in p Einzel- oder Sub-Cluster zerfällt.

In Abb. 3.2. sind an den Endpunkten der Partition die in dem zugehörigen Sub-Cluster am häufigsten auftretenden Kontrollvariablen mit der Häufigkeit ihres Auftretens im Sub-Cluster angegeben. Die Auftretenshäufigkeit ist in Prozent der Anzahl der Terminale Endpunkte der Cluster) - hier alle $K_0 = 127$ KV-Kombinationen - angeführt.

In Tabelle 3.1. sind die Häufigkeiten des Auftretens der Kontrollvariablen in den Sub-Clustern für Partitionen mit $p = 20$ summiert. Anhand dieser Tabelle wird entschieden, welche Kontrollvariablen bei der folgenden Klassifizierung der Betriebszustände über die Zeit als Variable verwendet werden, und zwar nach der Reihenfolge der summierten Häufigkeiten.

Die dominierende Kontrollvariable ist die CPU (-Auslastung). Dies ergibt auch die Partition mit $p=1$. Die Partition mit $p=3$ liefert sofort noch 4 gleichwertige Variablen zusätzlich. Bei den Partitionen $p = 7$ und $p = 20$ zeigt sich etwas mehr Differenzierung. Es sind jeweils die gleichen drei Kontrollvariablen (CPU, SIO, TS-CCL, Tabelle 3.1.).

Die Kontrollvariablen CPU (CPU-Auslastung), SIO (I/O-Rate) und TS-CCL (TS-Concurrency-Level) zeigen bei Partitionen mittlerer Länge ($p=20$, $p=7$) eine gleichmässige Dominanz über die anderen Kontrollvariablen; sie besitzen die drei höchsten Summenhäufigkeiten in allen Sub-Clustern. Dies ist das Ergebnis zur Auswahl der die Betriebszustände eindeutig identifizierenden Kontrollvariablen, wenn man den folgenden cluster-analytischen Ansatz wegen des Rechenzeit-Aufwands auf drei Kon-

^{*)} Die Untersuchung einer feineren Cluster-Struktur (d.h. Partition $p > 20$) ist wegen ihrer hohen Detailstruktur sehr unübersichtlich und nicht sinnvoll. Im Extremfall $p = 127$ besteht die Struktur aus allen einzelnen Objekten, sie lässt keine Schlüsse mehr auf weitere Zusammenhänge zu.

trollvariable beschränken will. Höhere Ansätze würden nach der Tabelle 3.1. noch die Kontrollvariablen TS-CPU, TS-IO und TRX (gleichberechtigt bei Partitionen $p = 20, 7, 3$; bei $p = 3$ würde ausserdem TS-CCL entfallen).

TABELLE 3.1.: Summenhäufigkeiten der Kontrollvariablen.

Kontr.-Var.	Partitionslänge p			
	p = 20	p = 7	p = 3	p = 1
CPU	<u>49</u>	<u>43</u>	<u>43</u>	<u>43</u>
TS-CPU	17	10	<u>10</u>	-
SIO	<u>23</u>	<u>18</u>	<u>10</u>	-
TS-IO	17	10	<u>10</u>	-
TRX	17	10	<u>10</u>	-
TS-CCL	<u>22</u>	<u>16</u>	-	-
SRT	4	-	-	-

Alle Zahlen in % von K_0 . Die höchsten Summenhäufigkeiten sind unterstrichen.

3.2. Zerlegung eines Betriebszyklus in verschiedene Betriebszustände.

Methode: Cluster-Analyse über die Messzeitpunkte (Objekte) des Betriebszyklus mit den Werten der drei nach Kap. 3.1. ausgewählten Kontrollvariablen (CPU-Auslastung, I/O-Rate und TS-Concurrency-Level).

3.2.1. Normierung und Bestimmung der Ähnlichkeitsmatrix.

Die Normierung der Kontrollvariablen-Werte erfolgt wie in Kap. 3.1. durch die Maximum-Norm. Als Ähnlichkeitsmass wird wieder die euklidische Norm verwendet, diesmal allerdings nur im 3-dim. Raum der Kontrollvariablen ($KV(k, t_i) = CPU(t_i)$, $k=1$, $=SIO(t_i)$, $k=2$, $=TS-CCL(t_i)$, $k=3$)

$$d_{ij} = \sqrt{\sum_{k=1}^3 (KV(k, t_i) - KV(k, t_j))^2}$$

3.2.2. Analyse der Cluster.

Es ist zu untersuchen, in wie weit drei Kontrollvariable (S. 44) ausreichen, die Betriebszustände während des Tagesablaufs zu klassifizieren^{*)}. Mit einer Partition der Länge 3 wird der ganztägige Betriebszyklus in 3 Betriebszustände zerlegt, die zeitlich nicht zusammenhängend sein brauchen (s. auch die Bemerkung auf S. 51).

Ergebnis: Die verschiedenen Betriebszustände stimmen ungefähr mit den Betriebsschichten^{*)} überein. Eine Darstellung der Betriebszustände über der Zeit ist in Abb. 3.3. gegeben. Zum Vergleich und zur Interpretation sind in Abb. 3.5. die Werte der der Cluster-Analyse zugrunde liegenden Kontrollvariablenwerte grafisch dargestellt. Praktische Erfahrungen bestätigen die Ergebnisse der Cluster-Analyse.

Betriebszustand 1: Das System ist weitgehend leer; die Werte von CPU, SIO und TS-CCL sind sehr niedrig.

Betriebszustand 2: Die CPU-Auslastung ist hoch, TS-CCL und SIO sind niedrig; das System wird in einem Mischbetrieb gefahren.

Betriebszustand 3: Dieser Zustand wird durch hohe TS-CCL- und hohe SIO-Werte charakterisiert; der TS-Betrieb überwiegt.

Zu mehreren Wechseln des Betriebszustandes kommt es am frühen Morgen, hervorgerufen durch Wartungsarbeiten (Platten-Backups, Plattenspeicher-Kontingent-Verwaltung usw.). Ebenso zeigt die Abb. 3.3. Wechsel während der Normalschicht, diese werden verursacht durch das Absinken des TS-CCL. Die Zuordnung der Betriebszustände zu den Betriebsschichten ist aus Tab. 3.2. ersichtlich.

^{*)} - Bezeichnung der Betriebsschichten während des Tages:

A - Nachtschicht	(22.00-08.00 Uhr, 10 Stdn.),
B - Normalschicht	(08.00-16.30 Uhr, 8.5 Stdn.) und
C - Spätschicht	(16.30-22.00 Uhr, 5.5 Stdn).

TABELLE 3.2.: Zuordnung der Betriebszustände - Betriebsschichten.
und ihre Dauer in Stunden.

		Betriebszustand			Se.	%
		1	2	3		
Betriebs- schicht (Tageszeit)	A	<u>5.25</u>	4.75	0	10.00	42
	B	0.25	3.75	<u>4.50</u>	8.50	35
	C	0	<u>5.50</u>	0	5.50	23
Se.		5.50	14.00	4.50	24.00	100
%		23	58	19	100	-
<p>Bemerkung: Die jeweils unterstrichenen Zahlen geben den für den Betriebszeitraum dominierenden Betriebszustand an.</p>						

Folgerung: Im Hinblick auf den Personal- und Rechenzeit-Aufwand beim Einsatz cluster-analytischer Verfahren zur Klassifizierung der verschiedenen Betriebszustände und der relativ guten Übereinstimmung mit den praktischen Erfahrungswerten wird dieses Verfahren in der RZ-Praxis selten angewendet werden und die Erfahrungswerte an seine Stelle treten. M. E. reicht es i.a. aus, zur Beschreibung des überwiegenden Timesharing-Betriebs die Werte des TS-Concurrency Levels (TS-CCL) zu untersuchen. Ist eine noch gröbere Klasseneinteilung ausreichend, so ist die Verwendung der Betriebsschichten durchaus hinreichend. Dies wird durch die Ergebnisse im folgenden Kapitel 3.2.3. noch bestätigt. Seine Bedeutung liegt daher mehr in der Identifikation von Lastarten allein aufgrund von Kontrollvariablenwerten, wenn die Zuordnung von Betriebszuständen zu den Produktionszyklen der Unternehmung nicht so deutlich zutage tritt.

3.2.3. Alternativ-Separation der Betriebszustände.

Zusätzlich zu dem im Kap. 3.1. angegebenen Auswahlverfahren für die die Betriebszustände beschreibenden Kontrollvariablen CPU, SIO und TS-CCL wurden weitere KV-Gruppen untersucht, um die Stabilität des Auswahlverfahrens und der Betriebszustände zu untersuchen. Dazu gehörten auch die nach Tab. 3.1. optimalen KV-Gruppen mit 5 (p=3) und 6 (p=7,20) Elementen. Die Ergebnisse weichen so geringfügig von den in Kap. 3.2.2., Tabelle 3.2., angeführten Ergebnissen ab, sodass auf ihre Darstellung verzichtet werden soll. Verglichen werden jedoch die Resultate von S. 47 mit den Ergebnissen eines Versuchs mit den beiden die Last des TS-Systems beschreibenden KV TS-CPU und TS-IO, sowie der CPU-Ausl. (CPU). Die Ergebnisse dieses Versuchs sind in Tab. 3.3. zusammengestellt.

TABELLE 3.3.: Zuordnung der Betriebszustände - Betriebsschichten, Alternativ-Separation (KV = CPU, TS-CPU, TS-IO) und ihre Dauer in Stunden.

		Betriebszustand			Se.	%
		1	2	3		
Betriebs- schicht (Tageszeit)	A	<u>5.50</u>	4.50	0	10.00	42
	B	0	0	<u>8.50</u>	8.50	35
	C	0	<u>4.50</u>	1.00	5.50	23
Se.		5.50	9.00	9.50	24.00	100
%		23	58	19	100	-

Bemerkung: Die jeweils unterstrichenen Zahlen geben den für den Betriebszeitraum dominierenden Betriebszustand an.

Abbildung 3.3.: Separation verschiedener Betriebszustände innerhalb eines 24-stündigen Betriebszyklus.

KV = CPU, SIO, TS-CCL.

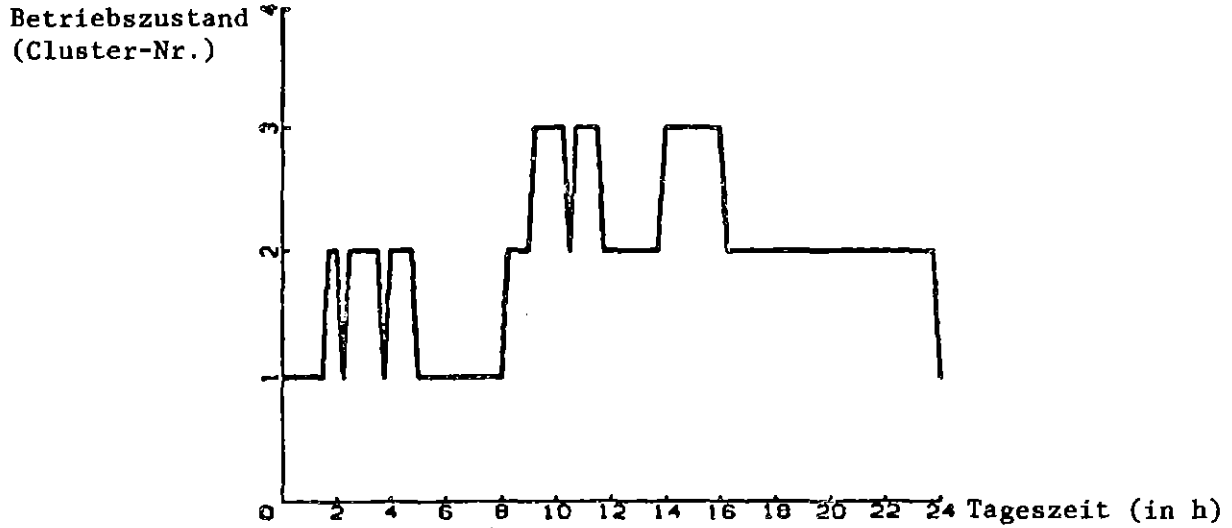


Abbildung 3.4.: Alternativ-Separation der Betriebszustände.

KV = CPU, TS-CPU, TS-IO.

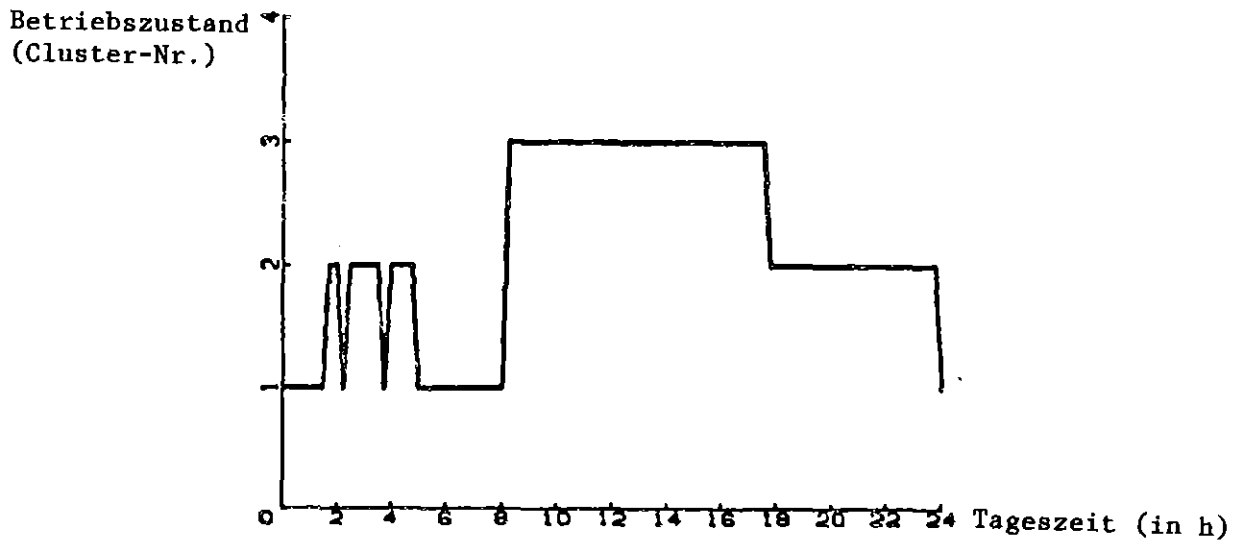
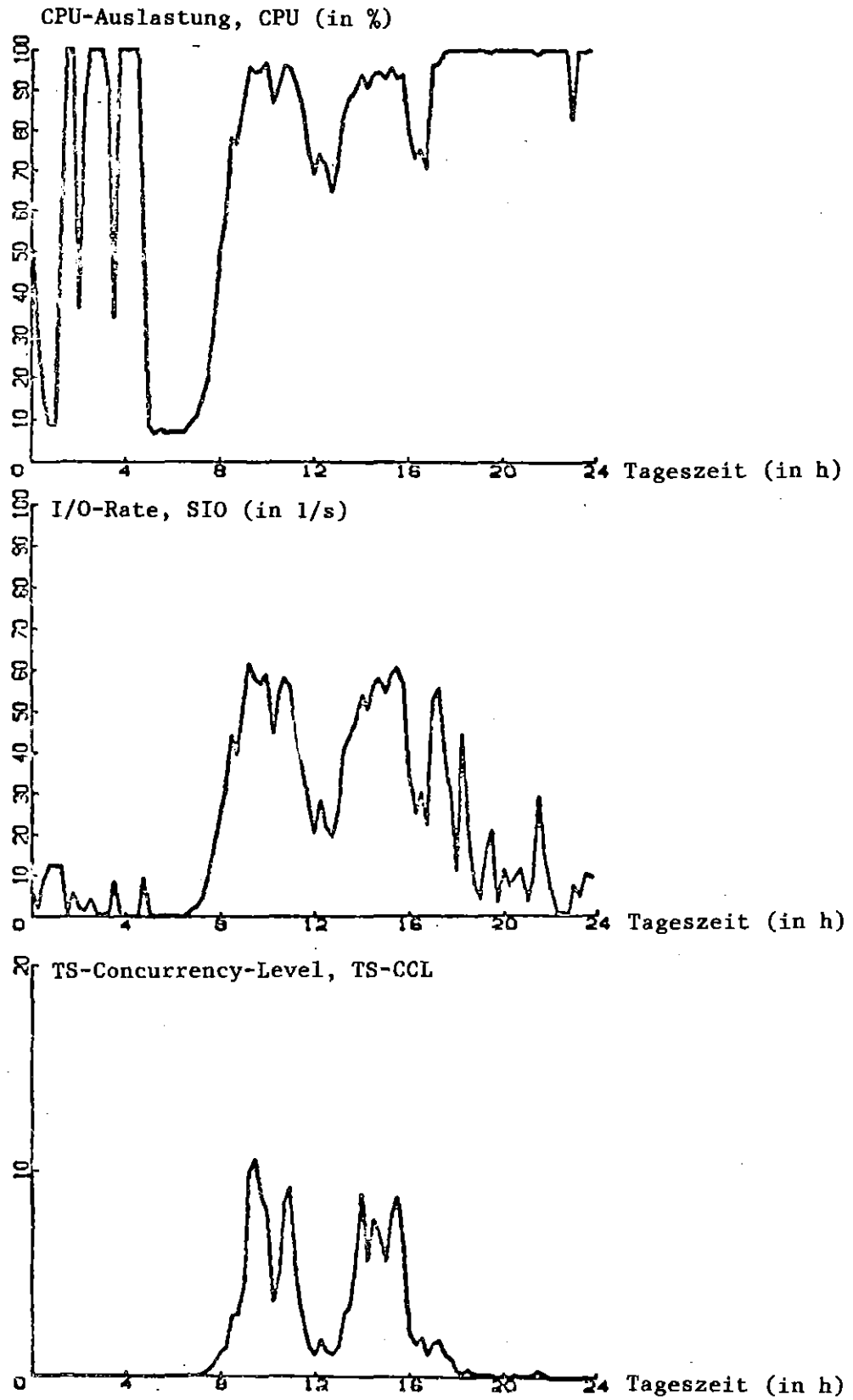


Abbildung 3.5.: Messwerte der zur Cluster-Analyse verwendeten Kontrollvariablen.



Ergebnis: Die Abweichungen beschränken sich auf die Wechsel während der Normalschicht, die in Abb. 3.4. nicht mehr vorhanden sind, und auf leichte Verschiebungen am Beginn und Ende des Betriebszustands 3, der sich geringfügig verlängert. Der Betriebszustand 3 zieht sich in das Betriebszeitintervall C hinein.

Wertung des Ergebnisses: Die Einführung der Kontrollvariablen TS-CPU und TS-IO führt zu einer Verringerung der 'Trennschärfe' bei der Klassenbildung. Dies muss nach dem Auswahl-Verfahren für die Kontrollvariablen (Kap. 3.1.) erwartet werden. Das Ergebnis ist eine Bestätigung des Auswahlverfahrens für die Kontrollvariablen. Es zeigt gleichzeitig die Stabilität der Betriebszustände gegenüber Veränderungen bei der Auswahl der Kontrollvariablen.

Bemerkung: Die Begrenzung der Anzahl der Betriebszustände auf 3 wurde nach empirischen Untersuchungen im Rechenzentrum des Kernforschungszentrums Karlsruhe festgelegt. Bei Bedarf der Unterscheidung von mehr als 3 verschiedenen Betriebszuständen kann die Partitionslänge p der Clusteranalyse von S. 46 auf einen grösseren Wert als 3 heraufgesetzt werden. Die Partitionslänge p entspricht der Anzahl der unterschiedlichen Betriebszustände.

4. Systemleistung und Servicegrad.

In diesem Kapitel werden einige für die Beschreibung des Leistungsverhaltens von TS-Systemen grundlegende Definitionen vorgenommen. Als systemorientierte Leistungsmasse werden die Systemleistung^{*)} L und das Leistungsvermögen (L_{\max} , maximale Systemleistung) und als Servicegradmass (benutzerorientiert) die System-Antwortzeit^{*)} (SRT) und die Service-rate (B) definiert und erläutert. Ein funktioneller Zusammenhang zwischen dem systemorientierten Leistungsbegriff L und dem Servicegrad wird angegeben; dabei zeigt sich, dass der TS-Concurrency-Level (TS-CCL) die Abhängigkeit des Servicegrades von der Systemleistung bestimmt und damit eine zentrale Rolle spielt.

Es werden messtechnische Verfahren angegeben, das Leistungsvermögen (bei vorgegebener Last) zu bestimmen. Das Leistungsvermögen lässt Leistungsvergleiche zwischen verschiedenen Systemen zu.

Die in diesem Kapitel verwendeten Leistungsvariablen werden im Gegensatz zu den Kontrollvariablen - mit Ausnahme der Variablen, die gleichzeitig Kontrollvariablen sind: TS-CCL, u, TRX und SRT - nicht direkt gemessen, sie sind abgeleitete Grössen. Diese Leistungsvariablen bilden die Grundlage für die Leistungsmessung und -beurteilung in der Praxis. Auf ihnen bauen die Leistungsbeurteilungen in den Kapiteln 7. und 8. auf.

4.1. Der Timesharing-Concurrency Level.

In einem TS-System werden während eines Zeitintervalls $T_M = t_1 - t_0$ von u laufenden Sitzungen ausgehende Transaktionen gemessen (Abb. 4.1.).

*) Die Systemleistung L und die System-Antwortzeit SRT hängen von der Last ab. Bei Messungen über grosse Betriebszyklen und bei reproduzierbaren Lasten (Benchmarks) können deren Schwankung vernachlässigt werden. Dagegen empfehlen sich für kurzfristige Leistungsanalysen, z.B. für system-interne Optimierungszwecke, lastunabhängige Leistungsmasse, wie 'Resource Time Units' und Dehnfaktoren (Vgl.: ZORN/GROSSE/HAAS: Leistungsoptimierung..., <ZORN80>, S. 595-602).

Es sei T_{ij} die Dauer der j -ten Transaktion des i -ten Benutzers in T_M .

Definition: (Gesamt-) Aktivzeit:

$$T_A = \sum_{i=1}^u \sum_{j=1}^{m_i} T_{ij} \quad (4.1)$$

m_i ist die Anzahl der in T_M beendeten Transaktionen des Benutzers i .

Definition: TS-Concurrency Level:

$$\text{TS-CCL} = \frac{\text{Aktivzeit}}{\text{Messzeit}} = \frac{T_A}{T_M} \quad (4.2.)$$

oder
$$T_A = T_M \cdot \text{TS-CCL} \quad (4.2.1.)$$

Es sei M die Anzahl aller in T_M beendeten Transaktionen:

$$M = \sum_{i=1}^u m_i \quad (4.2.2.)$$

Definition: Transaktionsrate:

$$\text{TRX} = M / T_M \quad (4.3.)$$

Wenn $D_i = \sum_j D_{ij}$ die (Gesamt-) Denkzeit des i -ten Benutzers in T_M ist, so gilt:

$$T_M = \sum_{j=1}^{m_i} T_{ij} + D_i \quad (4.4.)$$

für alle $i=1, \dots, n$ oder

$$u \cdot T_M = \sum_{i=1}^u \sum_{j=1}^{m_i} T_{ij} + \sum_{i=1}^u D_i$$

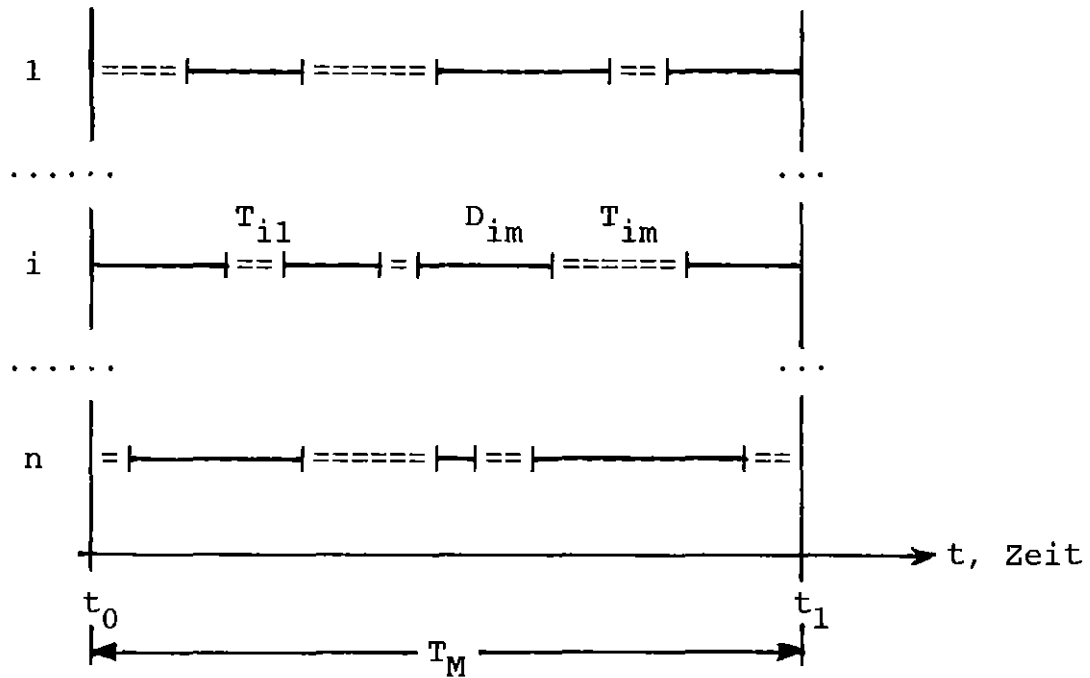
Es ist $T_D = \sum_i D_i$ die Denkzeit aller Benutzer in T_M ,
 also

$$(u - \text{TS-CCL}) \cdot T_M = T_D \quad (4.5.)$$

Die Formel (4.5.) besagt, dass bei festem T_M die Denkzeit T_D den Unterschied zwischen u und TS-CCL festlegt, u.a. auch $\text{TS-CCL} \rightarrow u$ für $T_D \rightarrow 0$ (s. Abb. 4.5., S. 63).

Abbildung 4.1.: Transaktionszeiten.

Benutzer



Aktivzeit: |=====|

Denkzeit: |—————|

4.2. Systemleistung.

Zur Leistungsbeschreibung einer Rechenanlage werden i.a. Leistungsgrößen einzelner Hardwarekomponenten herangezogen. OSSWALD stellt fest: "Von der Leistung eines Computer-Systems zu sprechen, ist jedoch ganz unmöglich. Es lässt sich physikalisch nicht nur eine Kennzahl für das angeben und mit dem Begriff Leistung belegen, was aus dem stochastischen Zusammenspiel Hardware-Software-Anwendungsprogramme in einer bestimmten Zeit herausgebracht wird."¹

In dieser Arbeit soll dieser Vorstellung nicht gefolgt werden, sondern neben Untersuchungen über den Einfluss der Betriebslast auf die Leistung des Systems der Versuch unternommen werden, mit einer reproduzierbaren Arbeitslast in der Form eines Scripts das Leistungsvermögen der Anlage festzustellen. Es wird damit dem Vorschlag FERRARIS² und anderer Autoren gefolgt, die eine künstliche reproduzierbare Arbeitslast zum Massstab der Leistungsvermögensbewertung machen (siehe Kap. 6.).

In dieser Arbeit soll - ausgehend von der physikalischen Leistungsdefinition - einem Ansatz von WINDFUHR³ für die Leistungsermittlung eines im Multiprogramming-Betrieb fahrenden Batch-Systems analog verfahren werden.

Für eine feste Arbeitslast (z.B. ein Script) eines Benutzers i wird bei u parallel laufenden Scripts die Verweilzeit T_i gemessen. Die Verweilzeit T_v aller u Sitzungen (oder Scripts) ist

$$T_v = \sum_{i=1}^u T_i \quad (4.6.)$$

-
- ¹ - OSSWALD: Leistungsvermögensanalyse..., <OSSW74>, S. 29.
² - FERRARI: Comp. Syst. Perf. Eval., <FERR74>, S. 300 ff.
³ - WINDFUHR: Methoden und Verfahren..., <WIND77>, S. 28.

Definition: mittlere Verweilzeit (eines Scripts):

$$T_S = T_V / u = (1/u) \cdot \sum_{i=1}^u T_i \quad (4.7.)$$

Definition: Systemleistung:

$$L = \frac{\text{Anz. d. Transaktionen}}{\text{Zeit}} = (1/T_S) \cdot \sum_{i=1}^u m_i$$

$$L = M / T_S \quad (4.8.)$$

$$L = Mu / T_V \quad (4.9.)$$

Bemerkung: Der Quotient M/T_S ist identisch mit der mittleren Transaktionsrate des Systems während T_S .

Die nach (4.8.) definierte Systemleistung steigt, wenn

- * die Anzahl der bearbeiteten Transaktionen M bei gleichbleibender Benutzerzahl und Verweilzeit steigt,
- * die Anzahl der Benutzer bei gleichbleibender Verweilzeit und gleicher Anzahl Transaktionen steigt oder
- * die Verweilzeit bei gleichbleibender Benutzer und Transaktionszahl sinkt.

4.3. Das Leistungsvermögen.

Als (System-) Leistungsvermögen soll die maximale Systemleistung bei einer gegebenen Arbeitslast bezeichnet werden. Laufen u Scripts parallel, so muss man feststellen, dass nach dem Überschreiten eines Wertes u_0 die Leistung L nicht weiter zu-, sondern abnimmt. Grund dafür ist der überproportionale Anstieg von T_V in Formel (4.9.). Das verstärkte Ansteigen von T_V ist die Folge des Auftretens einer Engpass-Ressource im System.

Der aus der Systemleistungskurve (siehe Abbildung 4.2.) abzulesende Wert u_0 ist der optimale Betriebswert für u bzgl. der Systemleistung. L_{\max} ist die maximale Systemleistung bei der vorgegebenen Last.

Bemerkung: Die Bestimmung der maximalen Systemleistung und des systemleistungs-optimalen Betriebswertes u_0 berücksichtigen keine Anforderungen an den Servicegrad des TS-Systems (siehe 4.5.).

Für die Ermittlung des System-Leistungsvermögens durch den Einsatz von u Scripts muss entweder ein system-internes Verfahren (zu messender Rechner treibt sich selbst) entwickelt werden oder ein externer Treiber, ein Rechner, der über die Kommunikations-Schnittstelle Terminals simuliert, muss verfügbar sein (Einsatz des dynamischen Benchmarks, s. Kap. 6.1. und 6.2., S. 72 ff.). Beide Verfahren sind aufwendig und werden deshalb in der Praxis selten angewendet. Ein Verfahren, das TS-CCL bezogen arbeitet und somit die Simulation vieler Terminals reduziert, wird ebenfalls in Kapitel 6. angegeben (Pseudo-Benchmark).

Bestimmung der maximalen Systemleistung in der Rechenzentrumspraxis.

Ein längerer Betriebszyklus (z.B. ein Monat) wird in kleinere Sub-Zyklen geteilt (z.B. Wochen oder Tage). Anhand der Werte der betriebsbegleitenden Messungen (Leistungsdatenbank) werden für jeden der Sub-Zyklen die Systemleistungskurven aufgetragen. Dazu wird die Formel (4.8.) verwendet. Liegen die Systemleistungskurven dicht beieinander, so kann auf eine gleichmässige Last während des Betriebszyklus geschlossen werden. Der Mittelwert der L_{\max} aller einzelnen Leistungskurven gibt einen Näherungswert für die maximale Systemleistung an (s. Abb. 4.2., S. 59, vgl. Kap. 7.2., S. 110).

Definition: relative Systemleistung:

$$l = L / L_{\max} \leq 1 \quad (4.10.)$$

oder $l' = 100 \cdot l \quad (\text{in } \%)$

Da das System in der Praxis nicht mit konstanter Last betrieben wird, kann das System auch nicht gleichmässig bei maximaler Systemleistung gefahren werden. Die Summenhäufigkeit⁴ $S_R(l)^*$ der relativen Systemleistung l (bzw. deren Normalverteilungsapproximation $P_R(l)$) gibt Auskunft über die Wahrscheinlichkeit, mit der ein bestimmter Leistungszustand gefahren wird.

Die Grösse

$$C_S = \sum_{i=1}^N S_R(l_i) \Delta l_i^* \quad (4.10.1)$$

$$= 0.05 \sum_{i=1}^{20} S_R(l_i)$$

(angenäherte Fläche unter $S_R(l)$), bzw.

$$C_P = \sum_{i=1}^N P_R(l_i) \Delta l_i^* \quad (4.10.2)$$

$$= 0.05 \sum_{i=1}^{20} P_R(l_i)$$

(angenäherte Fläche unter $P_R(l)$) soll als Kennzahl der freien Kapazität (Leistungsreserve) verwendet werden.

*) Die Summenhäufigkeit wird hier stets durch eine Gruppierung des Wertebereichs von 0 bis $l_{\max} = 1$ in $N = 20$ äquidistante Teile erzeugt.

⁴ - SACHS: Angewandte Statistik, <SACH78>, S. 47.

Abbildung 4.2.: Systemleistungskurven.

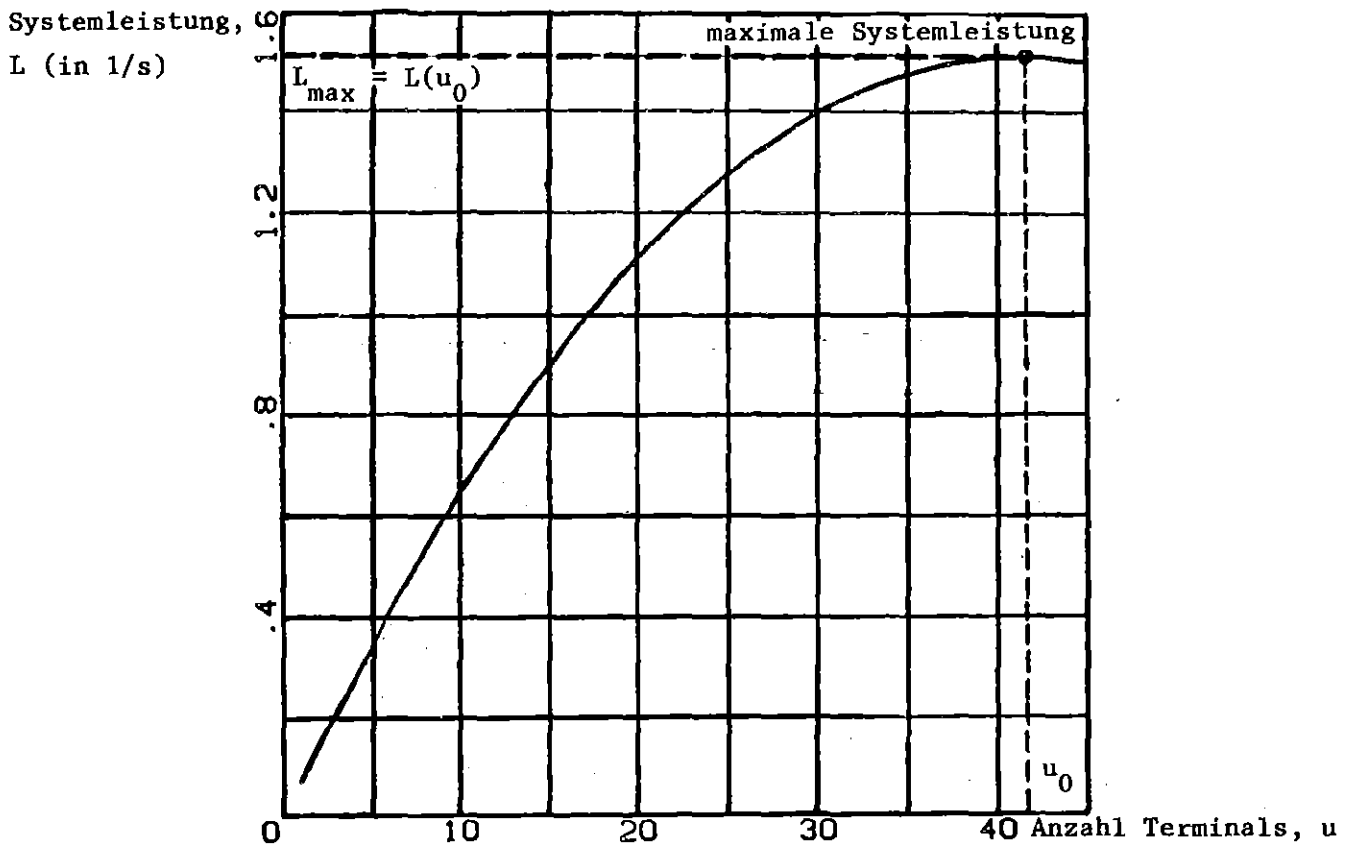
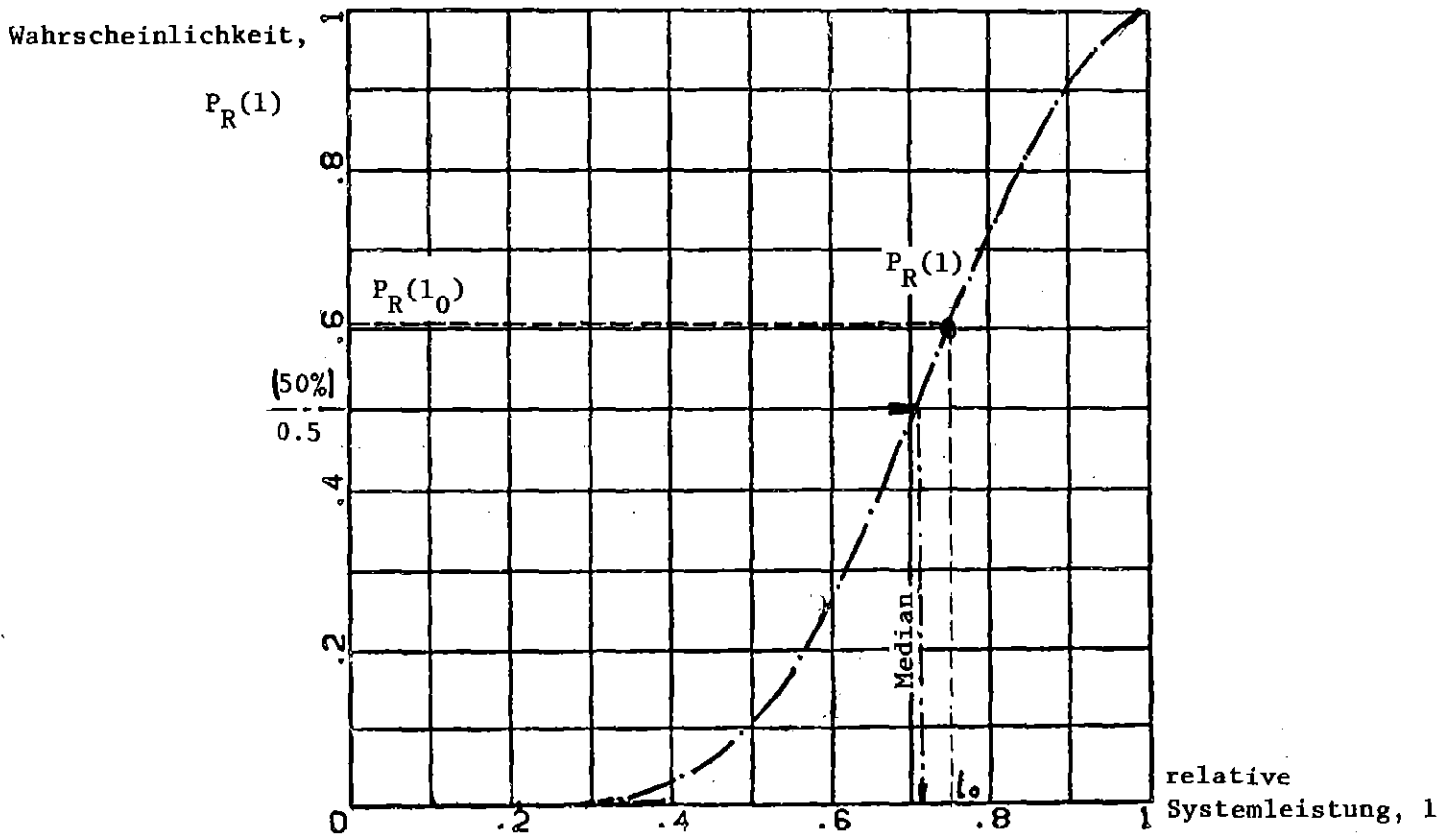


Abbildung 4.3.: Verteilung der relativen Systemleistung.



4.4. Leistungsvergleiche.

Leistungsvergleiche beziehen sich auf den Vergleich der maximalen Systemleistungen zweier Systeme.

Definition: Leistungsquotient:

$$Q = L_{\max,2} / L_{\max,1} \quad (4.11.)$$

Werden beide Leistungsvermögen durch Benchmarks ermittelt, so ist

$$Q = \frac{T_{v,1} \cdot M_2 \cdot u_2}{T_{v,2} \cdot M_1 \cdot u_1} \quad (4.12.)$$

Man beachte, dass bei einem Ansteigen von u und M auch T_v ansteigt. Wird auf den zu vergleichenden Systemen der gleiche dynamische Benchmark (Kap. 6.2., S. 75) gefahren, so ist $M_1 = M_2$ und $u_1 = u_2$ und es kann

$$Q(u) = L_2(u) / L_1(u) = T_{v,1} / T_{v,2} \quad (4.13.)$$

als Leistungsvergleichsfunktion bei verschiedenen u -Werten (z. B. beim Einsatz homogener Script-Produkte, s. Kap. 6.4., S. 81, als dynamischer Benchmark) verwendet werden (Beispiel in Kap. 8.2.2., S. 131).

Definition: relativer Leistungsquotient:

$$q = l_2/l_1 = (1/Q) \cdot L_2/L_1 \quad (4.14.)$$

Wird der relative Leistungsquotient mit gleichen Scripts ermittelt, so gelten $u_1 = u_2$ und $M_1 = M_2$. Es ist dann

$$q = (1/Q) \cdot T_{v,1} / T_{v,2},$$

mit den T_v als den Gesamtverweilzeiten für die Benchmarks.

4.5. Mittlere Antwortzeit und Servicerate.

Definition: Mittlere Antwortzeit:

$$\text{SRT} = \frac{\text{Aktivzeit}}{\text{Anzahl der Transaktionen}} = \frac{T_A}{M} \quad (4.15.)$$

Diese Formel ergibt sich auch aus der Gl. (4.1.), wenn man die Aktivzeit der j-ten Transaktion des i-ten Benutzers T_{ij} als die Antwortzeit der Transaktion interpretiert. Diese Zeit ist mit der in Kap. 2.3. gegebenen Definition der System-Antwortzeit identisch (Vgl. dort die Abb. 2.10., S.32). Nimmt man SRT als den Mittelwert aller Antwortzeiten im Messintervall T_0 , so ist

$$T_A = \text{SRT} \cdot M$$

oder

$$\frac{T_A}{T_0} = \text{SRT} \cdot \frac{M}{T_0}$$

und mit den Formeln (4.2.) und (4.8.)

$$\text{TS-CCL} = \text{SRT} \cdot L \quad (4.16.)$$

Die Gleichung (4.16.) stellt eine Beziehung her zwischen der mittleren Antwortzeit SRT und der Systemleistung L, sie gibt die funktionelle Abhängigkeit des Servicegrades (Antwortzeit) von der Systemleistung an.

Definition: Servicerate:

$$B = 1 / \text{SRT} \quad (4.17.)$$

(Reziprokwert der System-Antwortzeit) Bei steigender Antwortzeit SRT

sinkt die Servicerate B ab. Mit einer Leistungsverbesserung ist also ein Anstieg der Servicerate verbunden.

Aus der Definition der mittleren Antwortzeit (4.15.) und der Servicerate (4.17.) lässt sich die Bedeutung der Servicerate präzisieren:

$$B = \frac{M}{T_A} = \frac{\text{Anzahl der Transaktionen}}{\text{Aktivzeit}} \quad (4.18.)$$

Mit den Formeln (4.2.1.) und (4.8.) ergibt sich

$$B = L / \text{TS-CCL} \quad (4.19.)$$

oder $L = B \cdot \text{TS-CCL} \quad (4.19.1.)$

B ist der Anteil an der Systemleistung, der im Mittel auf jeden aktiven Benutzer entfällt (aus (4.19.)). Deshalb heisst B Servicerate. Bei konstanter Systemleistung und fallendem TS-Concurrency-Level (TS-CCL) steigt die Servicerate an. Bei TS-CCL = 1 steht jedem Benutzer (wegen $u \geq \text{TS-CCL}$) die Systemleistung $L(u)$ zur Verfügung.

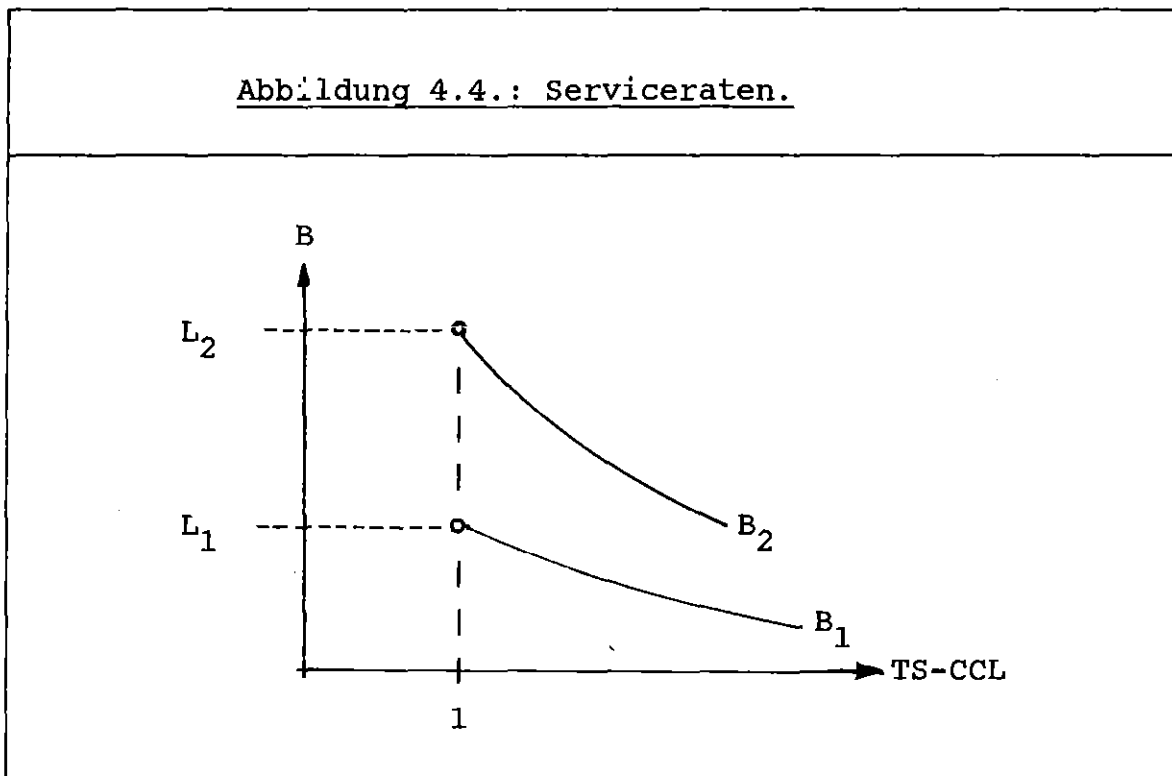


Abbildung 4.5.: Der TS-Concurrency-Level TS-CCL als Funktion von u.

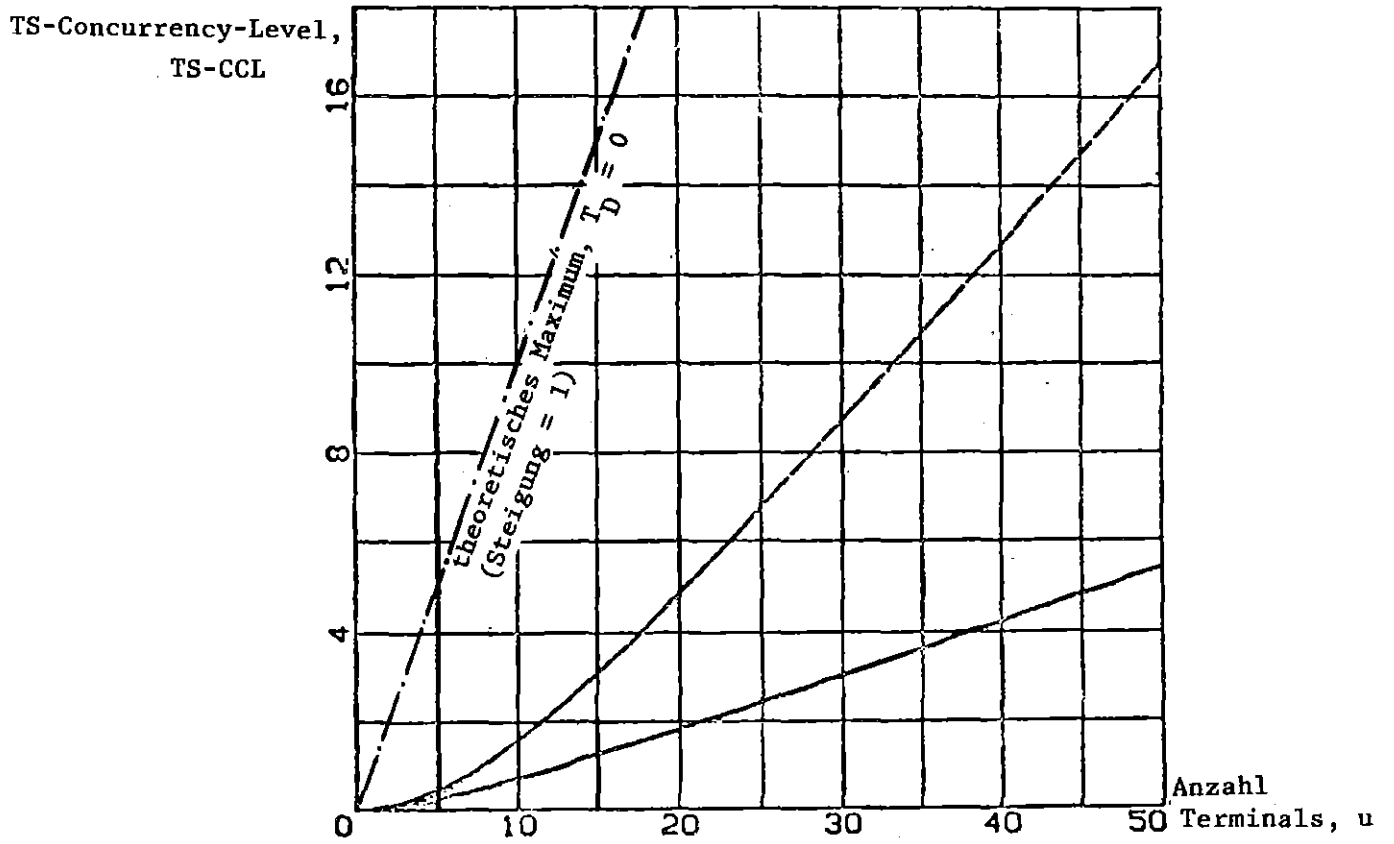
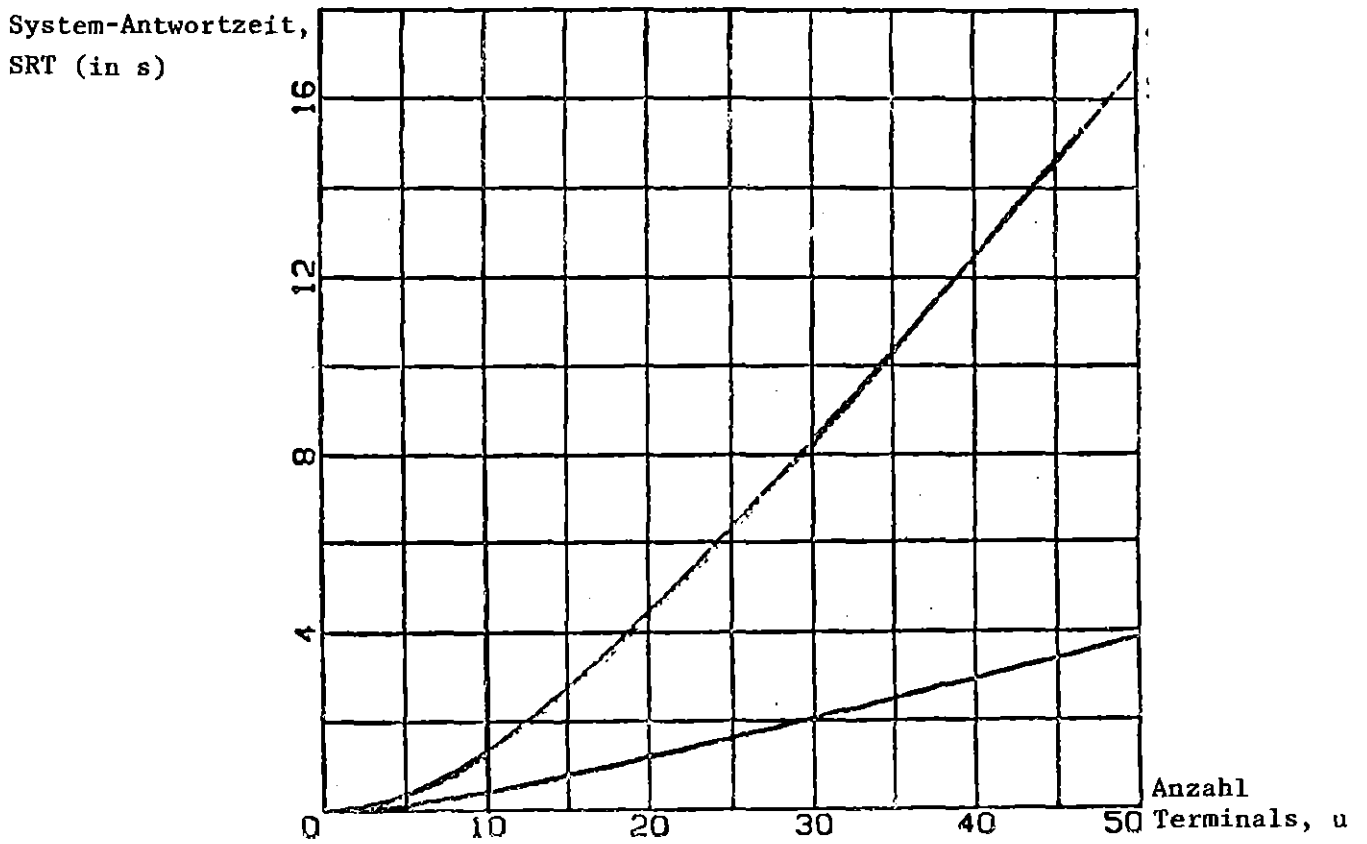


Abbildung 4.6.: Die Antwortzeit SRT als Funktion von u.



4.6. Produktivitätsgewinn und Gewinnfunktion.

Ohne Anknüpfungspunkte in der Literatur soll hier der Versuch gemacht werden, die durch Leistungsverbesserungen zu erzielende kleinere mittlere Antwortzeit durch ihre Auswirkungen auf die Benutzer-Produktivität zu quantifizieren. Dabei wird davon ausgegangen, dass die Produktivität grösser wird, wenn entweder der Benutzer zur Erfüllung gleicher Aufgaben weniger (Antwort-) Zeit aufwenden muss oder er bei gleicher Zeit mehr gleichartige Aufgaben erledigen kann.

Zum Vergleich verschiedener TS-Systeme (Index 1 und 2) auf der Basis ihrer mittleren Antwortzeiten SRT_1 und SRT_2 wird die Gewinnfunktion bestimmt:

Definition: Gewinnfunktion:

$$f(u_i) = \left(\frac{SRT_1(u_i) \cdot p_1(u_i)}{T_1} - \frac{SRT_2(u_i) \cdot p_2(u_i)}{T_2} \right) u_i \quad (4.20.)$$

mit $p_j(u_i)$ = relative Häufigkeit dafür, dass während der Messung j mit der Messzeit T_j : $i-1 < u_i \leq i$ ist.

Wird $T_1 = T_2 = T$ (gleicher Betriebszyklus), so ist

$$g(u_i) = T \cdot f(u_i)$$

der Produktivitätsgewinn für u_i Benutzer während T (in Mann-Sekunden, Ms oder Mann-Stunden Mh).

Definition: Gesamt-Produktivitätsgewinn:

$$F = \sum_{u_i=1}^{u_{\max}} f(u_i) \quad (4.21.)$$

F definiert den Produktivitätsgewinn für alle Benutzer auf der Basis der Vergleichsmessungen T_1 und T_2 .

Wählt man wieder $T_1 = T_2 = T$, so gibt

$$G = T \cdot F$$

den Gesamt-Produktionsgewinn (in Ms, Mh) für das Zeitintervall T an aller Benutzer während des Zeitintervalls T an.

$$F' = 100 \cdot F \quad (4.21.1)$$

definiert den relativen Produktivitätsgewinn (in %).

Der Produktivitätsgewinn wird am Produktionssystem (Betriebslast) bestimmt. $SRT_1(u)$ und $SRT_2(u)$ und die Verteilungen von u werden mit dem Monitor gemessen. Zweckmässigerweise werden die Messzeitintervalle T_j gleich gewählt, und zwar eine oder mehrere 1. Schichten oder Tage; der Produktivitätsgewinn kann dann einfach in Mann-Stunden pro Arbeitstag angegeben werden.

Mit dem Quasi-Benchmark (siehe Kap. 7.2.) wird der Gewinn für einzelne Funktionsklassen bestimmt (Beispiel in Kap. 8.2.6., S. 149).

$$f_k(u_i) = \left(\frac{SRT_{k,1}(u_i) \cdot p_1(u_i) \cdot r_{k,1}}{T_1} - \frac{SRT_{k,2}(u_i) \cdot p_2(u_i) \cdot r_{k,2}}{T_2} \right) \cdot u_i \quad (4.22.)$$

ist die Gewinnfunktion für Funktionen aus der Klasse k. Die $SRT_{k,j}$ werden durch die Antwortzeiten der Kommandos des Quasi-Benchmarks angegeben. $r_{k,j}$ ist der Anteil an der Benutzerverteilung $p_j(u_i)$, der Funktionen der Klasse k verwendet, beide Grössen müssen (mit einem Monitor) gemessen werden. Wegen der vollständigen Zerlegung des Funktionsspektrums ist stets:

$$\sum_k r_{k,j} = 1$$

Der Gesamt-Mittelwert der Antwortzeit (aller Klassen) für die Messung j bei u_i Benutzern, vgl. Formel (4.20.), ist:

$$\text{SRT}_j(u_i) = \frac{\sum_k \text{SRT}_{k,j}(u_i) \cdot r_{k,j}}{\sum_k r_{k,j}} \quad (4.24.)$$

wobei $\text{SRT}_{k,j}$ der Klassen-Mittelwert für die Klasse k ist. Dann ist mit (4.23.) und (4.24.):

$$f(u_i) = \sum_{k=1}^{k_{\max}} f_k(u_i), \quad (4.25.)$$

dies ist die Komposition der funktionsklassen-spezifischen Gewinne zur Gewinnfunktion nach Formel (4.20.). Analog gilt:

$$F_k = \sum_{u_i=1}^{u_{\max}} f_k(u_i) \quad (4.26.)$$

und

$$F = \sum_{k=1}^{k_{\max}} F_k = \sum_{k=1}^{k_{\max}} \sum_{u_i=1}^{u_{\max}} f_k(u_i) \quad (4.27.)$$

die Komposition des Gesamt-Produktivitätsgewinns.

Ein Beispiel zur Bestimmung des Produktivitätsgewinns wird in Kap. 8.2.6., S. 146-151, gegeben.

5. Das Funktionsspektrum eines Timesharing-Systems.

Universelle TS-Systeme dienen im Unternehmen dazu, Anwendern DV-Dienstleistungen an ihren Arbeitsplatz oder seiner Nähe zur Verfügung zu stellen. Dominieren nicht spezifische Anwendungen, die zum Einsatz eines dedizierten Dialogsystems geführt haben (z.B. Datenerfassung oder Auskunftsbearbeitung), so bieten TS-Systeme ein breites Spektrum von Nutzungs- und Anwendungsformen an. Dies reicht von der Stapelfernverarbeitung bis zur Prozessdatenerfassung bei der Art der Benutzung und von der Programmübersetzung, -test mit speziellen Hilfsfunktionen bis zur Vermittlung von Nachrichten ('electronic mail system') bei den Anwendungsfunktionen¹.

Die Funktionen ermöglichen dem Benutzer, dem TS-System verschiedene Arten von Arbeiten (als Transaktionen) in Auftrag zu geben. Das System führt die Funktion aus, nachdem der Benutzer die Art des Auftrages von einem Terminal aus, in der Form eines oder mehrerer Kommandos mit Argumenten versehen, an den Rechner übergeben hat. Elementare Funktionen werden durch die Eingabe von nur einem Kommando (mit Argumentliste) aufgerufen.

Die Kommandosprachen der TS-Systeme verschiedener Hersteller, sind i.a. sehr unterschiedlich, auch wenn die Funktionen vergleichbar sind.

Es ergeben sich mit der mangelnden Kompatibilität der Kommandosprache sofort Schwierigkeiten, eine für Systeme verschiedener Hersteller maschinenlesbare und interpretierbare Folge von Kommandos aufzustellen (TS-Benchmark oder Script, siehe Kap. 6.), und damit gleiche Funktionen mit gleichen Namen aufzurufen.

Die Anforderungen einzelner Funktionen an die Betriebsmittelvergabe sind stark variierend. Sie sind nicht nur von Funktion zu Funktion verschieden, sondern können allein durch die Veränderung eines Arguments beim Kommando-Aufruf wesentlich verändert werden².

¹ - FISCHER: Teilnehmerbetrieb, <FISC73>, S. 34.

² - DOHERTY: Measurement and Management..., <DOHE75>, S. 6.

Zur späteren Synthese (Kap. 6.) einer künstlichen Arbeitslast, die der realen entspricht, erfolgt eine Analyse der realen Arbeitslast.

Zur Durchführung müssen alle Transaktionen, die das System in einem Zeitintervall bearbeitet, aufgezeichnet und nach ihren Auftrenshäufigkeiten klassifiziert werden. Die Praxis hat gezeigt, dass dafür wenige Klassen ausreichen, die vom Anwendungsspektrum innerhalb der Unternehmensorganisation abhängen. Tabelle 5.1. zeigt eine in der Literatur (z. B. in ³) übliche Klasseneinteilung, die mit der in Kap. 1., S. 2, gegebenen funktionalen Beschreibung von TS-Systemen übereinstimmt (Fehlererkennung und -behebung, sowie die Programmausführung sind zusammengefasst). In ihr sind die Benutzungshäufigkeiten der Kommandos der einzelnen Klassen und deren Anteil an der akkumulierten System-Antwortzeit angegeben.

TABELLE 5.1.: Einteilung der Funktionen nach Anwendungsklassen.

	Anwendungsklassen			Anzahl bzw. Zeit
	Programmüb., und -ausf.	Datenver- waltung	Sitzungs- kontrolle	
Anz. Kommandos ^{4,5}	29 %	36 %	35 %	80
Benutz.-Häuf.	3 %	57 %	40 %	8327
akk. Antwortzt.	9 %	32 %	59 %	11.74 h
mittl. Antwortzt.	18.8 s	2.8 s	7.5	5.1 s

Messung: Software-Monitor General Trace Facility (GTF),
 Messzeit: 16. und 18.8.1978, insgesamt 183 min.,
 System: System-Konfiguration 1, s. Anhang, Kap. 9.2., S. 163.

³ - HELLERMAN/CONROY: Computer System Performance, <HELL75>, S. 236.
⁴ - IBM: OS/VS TSO Command Language Reference, <IBMO76.7>.
⁵ - ANTONCZYK/PIETSCHMANN/SCHRIEFER: TSO-Handbuch, <ANTO79>.

Änderungen oder Verschiebungen im Funktionsspektrum der Betriebslast können exogener (neue Projekte des Unternehmens werden auf das TS-System genommen) oder endogener Art (das Rechenzentrum verändert Betriebsparameter so, das bisher praktizierte Arbeitsmethoden der Benutzer teuer oder 'langsam' werden; neue Kommandos können bis dahin verwendete langsamere überflüssig machen).

Einem Experiment zur Nachbildung der realen Arbeitslast muss stets eine Untersuchung dieser Last vorausgehen. Stellt sich dabei eine Veränderung heraus, so muss die künstliche Arbeitslast angepasst werden.

Ziel der Klasseneinteilung für diese Arbeit ist es, die funktionellen Klassen jeweils durch benchmarkfähige Scripts zu repräsentieren (Kap. 6.) Daher wurde die folgende Einteilung vorgenommen, die die TS-Funktionen (Kommandos) nach der Art und gfls. dem Grad ihrer Lastauslösung gliedert. Als Last-Koordinaten wurden die Betriebsmittelanforderungen bzgl. CPU, I/O-Konfiguration und Speicher-Referenz gewählt, wobei aufgrund empirischer Untersuchungen im Rechenzentrum auf die letzte Koordinate verzichtet werden (vgl. Kap. 2.1., S. 20-21) konnte. Aus den Ergebnissen der Cluster-Analyse (S. 43-45) wurde die Notwendigkeit abgeleitet, eine CPU- und I/O-last-neutrale Funktionsklasse einzuführen. Sie soll die Kommandos repräsentieren, deren Systemanteil an der Antwortzeit gross ist. Diese Klasse soll transaktionsorientiert genannt werden. Ähnliche Einteilungen sind in der Literatur bekannt⁶.

Die CPU-orientierte Klasse (C-Klasse) besteht aus Kommandos, deren CPU-Lastkomponente überwiegt. Dazu gehören Kommandos der Programmübersetzung, der Programmier-testhilfen und die (CPU-orientierte) Programmausführung. Die I/O-orientierte Klasse (I-Klasse) umfasst Kommandos, die häufig Ein-/Ausgabe anstossen; dies sind z.B. Kopier-Kommandos für Dateien oder I/O-orientierte Programmausführungen. Die transaktionsorientierte Klasse (T-Klasse) sollen alle Kommandos zugezählt werden, deren explizite Betriebsmittelanforderungen gering sind, z.B. Unterkommandos des Text-Editors.

Diese Klasseneinteilung ist formal analog zu der in Kap. 6, S. 77,

⁶ - IBM: Host Systems Performance, <IBMO76>.

für die Auswahl von Basis-Scripts. Diese lastart-bezogene Einteilung eignet sich besonders als Grundlage für den Aufbau der funktionsklassen-orientierten Basis-Scripts, auf der die Erzeugung der betriebslast-nachbildenden Script-Produkte oder -Polynome (S. 81/82) für die Benchmark-Experimente beruht (Vgl. die funktionsklassenweise Lastanpassung in Kap. 6.5., S. 84-86).

Eine Liste mit der Funktionsklassen-Zugehörigkeit der in dieser Arbeit verwendeten Scripts befindet sich bei der Beschreibung der Basis-Scripts im Anhang, Kap. 9.3., S. 168 ff.).

Die Einteilung des Funktionsspektrums der Betriebslast gemäss dieser Lastart-Klassen ist in Tab. 5.2. wiedergegeben. Sie ist die Grundlage der Klassifizierung der Kommandos des TS-Systems für die vorliegende Arbeit. Die in Tabelle 5.2. angegebenen Anteile der einzelnen Klassen sind die Zielwerte für die funktionsklassenmässige Approximation der Benchmarklast in Kap. 6.2. (S. 84 ff.) und für das Beispiel in Kap. 8.2.1., S. 126, für den Einsatz des dynamischen TS-Benchmarks.

TABELLE 5.2.: Einteilung der Funktionen nach Lastarten.

	Funktionsklassen			Anzahl bzw. Zeit
	TRX- orientiert	I/O- orientiert	CPU- orientiert	
Anz. Kommandos	33 %	39 %	28 %	80
Benutz.-Häuf.	61 %	36 %	3 %	8327
akk. Antwortzt.	35 %	56 %	9 %	11.74 h
mittl. Antwortzt.	2.6 s	10.2 s	15.2	5.1 s

Messung: Software-Monitor General Trace Facility (GTF),
 Messzeit: 16. und 18.8.1978, insgesamt 183 min.,
 System: System-Konfiguration 1, s. Anhang, Kap. 9.2., S. 163.

6. Simulation des Funktionsspektrums.

Eine betriebsbegleitende Messung aller in Kapitel 2. vorgeschlagenen Kontrollvariablen ermöglicht eine Analyse und Beurteilung der Leistung des Timesharing-Systems unter seiner natürlichen Arbeitslast. Diese Last ist nicht reproduzierbar. Eine Wiederholung der verschiedenen Lastzustände tritt nur im statistischen Sinne ein und dann nur, wenn keine langfristigen Verschiebungen (Trends) auftreten.

Soll das Leistungsverhalten (Systemleistung und Servicerate) anhand reproduzierbarer Lasten ermittelt werden, so muss die natürliche Last (Betriebslast) durch eine künstliche Last nachgebildet werden¹.

Das Programmpaket, welches die künstliche Last erzeugt, heisst Benchmark². Das Fahren eines Benchmarks auf einer Rechenanlage ist das Benchmarking. Benchmarking erlaubt den Vergleich zweier Systeme unter gleichen und reproduzierbaren Last-Bedingungen. Zur Messung der Laufzeit von Benchmarks in Systemen mit Multiprogramming-Betrieb wird die mittlere Verweilzeit zur Bewertung herangezogen³. Dies ist für TS-Systeme stets der Fall (zur Definition der Verweilzeit vgl. Kap. 4.2.).

Ein Script besteht aus Folgen von Elementarfunktionen der Kommandosprache. Scripts werden zum Benchmarking auf TS-Systemen eingesetzt.

In diesem Kapitel soll die Strategie der Entwicklung mehrerer Scripts angegeben werden, das jedes für sich die Bewertung einzelner Funktionsklassen ermöglicht. Diese Scripts heissen Basis-Scripts. Durch Zusammensetzung mehrerer dieser Basis-Scripts werden die betrieblich relevanten Lasten nachgebildet. Verfahren zur Lastanpassung werden angegeben (siehe S. 84).

¹ FERRARI - Comp. Syst. Perf. Eval., <FERR78>, S. 221.

² JOSLIN: Computer Selection, <JOSL68>.

³ WINDFUHR - Methoden und Verfahren..., <WIND77>, S. 82.

6.1. Scripts und Benchmarks.

Zur Realisierung von Benchmarks auf Timesharing-Systemen werden Scripts herangezogen. Im Gegensatz zur Analyse der in der Leistungsdatenbank erfassten Kontrollvariablenwerte garantiert das Benchmarking eine kontrollierte Umgebung. Scripts sind anwendungsbezogene Funktionen, die in der Form von Kommando-Sequenzen des zu untersuchenden TS-Systems realisiert werden. Allerdings sind der Portabilität von Scripts sehr enge Grenzen gesetzt: Die Kommando-Sprachen der TS-Systeme verschiedener Hersteller sind so unterschiedlich, dass ablauffähige Scripts i.a. nur auf den Systemen eines Computer-Herstellers betrieben werden können.

Anwender-Scripts: Werden die Kommando-Folgen von Benutzern aufgezeichnet und später als Scripts verwendet, so sollen diese Anwender-Scripts heißen.

Traces: Traces sind die Aufzeichnungen aller relevanten Ereignisse des Systems. Wenn diese Aufzeichnungen später selbst zur Erzeugung von Treibern einer Arbeitslast verwendet werden (executable tr.), so erzeugen sie gleiche Verweilzeiten wie die reale Last. Solche Verfahrensweise ist in der Praxis i.a. zu zeitaufwendig, um eingesetzt zu werden.

Synthetische Scripts: Dies sind Kommando-Folgen, die nicht notwendigerweise bezug zu einzelnen realen Sitzungsabläufen haben. Sie sind durch Parameter an die natürliche Arbeitslast anzupassen. Synthetische Scripts werden verwendet, um

- Anlagen-Vergleiche mit einer Standard-Last vorzunehmen. Hierbei muss wiederum unterschieden werden zwischen Vergleichen von Anlagen, die mit der gleichen Betriebslast gefahren werden (Veränderungs-Vergleiche) und Vergleichen von Anlagen, die normalerweise verschiedene Betriebslasten fahren (Quervergleiche). Im letzten Fall muss darauf geachtet werden, dass Systemparameter speziell auf die Betriebslasten eingerichtet sind.
- im laufenden Betrieb Vergleichsmessungen zur Bewertung einzelner Funktionen oder Funktionsklassen vorzunehmen (durch einen Quasi-

Benchmark). Während der Verweil- oder Antwortzeitmessungen werden alle anderen Kontrollvariablen mitgemessen (zur Definition des Quasi-Benchmark siehe S. 75/76).

- Engpässe in einem System durch Erzeugung von Grenzbelastungen für einzelne Komponenten zu erkennen oder zu lokalisieren und die Ergebnisse der Engpass-Beseitigung zu verifizieren.

Den synthetischen Standard-Benchmarks entsprechende Scripts zur Durchführung von Quervergleichen von z.B. wissenschaftlichen interaktiven Systemen wurden schon in ⁴ (MULTICS) und ⁵ (UNIVAC) vorgeschlagen. Weitere Beispiele findet man in ^{6,7} (IBM) und ⁸ (SIEMENS).

Alle diese Beispiele (mit Ausnahme von ⁶) zeigen Scripts, die die Arbeitslasten des einzelnen Systems repräsentieren. Ein Vergleich z.B. des SALTZER/GINTELL-Ansatzes mit den Tabellen 5.1. und 5.2. (S. 68-70) zeigt, dass auch die funktionelle Portabilität keineswegs gegeben ist. Die in dieser Arbeit angegebene Methodik zeigt eine anlagenunabhängige Systematik, nach der ein TS-Benchmark aus Script-Produkten oder -Polynomen der Basis-Scripts (S. 77) aufgebaut werden kann.

Voraussetzung für den Einsatz von Benchmarks bei Timesharing-Anlagen ist die Möglichkeit, eine grosse Anzahl von Kommandos gleichzeitig für den Benchmark erzeugen zu können. Dafür bestehen verschiedene Realisierungen:

Manuelle Eingabe: Es werden u Terminals von u Personen besetzt, die eine vorgegebene Liste von Kommandos eingeben. Die Denkzeiten müssen von den Personen eingehalten werden. Dieses Verfahren ist organisatorisch so aufwendig, dass es in der Praxis kaum realisiert werden kann.

⁴ - SALTZER/GINTELL: The Instrumentation..., <SALT70>, S. 495-500.

⁵ - FOGEL/WINOGRAD: Einstein: An Internal..., <FOGE72>, S. 6-14.

⁶ - IBM: Host Systems Performance, <IBMO76>.

⁷ - ALEXANDER/MERTENS/RUMLER: A Timesharing Benchmark..., <ALEX76>.

⁸ - MÜHLENBEIN: Ein Timesharing Benchmark..., <MUEH77.1>.

Automatisierte Eingabe: Es werden u Terminals verwendet, von denen aus die bereits gespeicherten Scripts angestossen werden. Denkzeiten werden automatisch generiert. Die vorhandenen Datenübertragungswege werden benutzt. Zur Bedienung sind nur sehr wenige Personen erforderlich. Die Reproduzierbarkeit der Sitzungen ist grösser als bei manueller Eingabe, da die Variabilität der Eingaben durch verschiedene Personen entfällt. Zeitmessungen werden am System mit Monitoren durchgeführt.

Interne Treiber: ^{9,10} Das zu messende System erzeugt die Eingabe, diese wird jedoch nicht mehr über Terminals und die natürlichen Übertragungswege an das System gesendet, sondern nur noch durch Einordnen der Eingaben in die Systemwarteschlangen. Übertragungswege und Denkzeiten müssen simuliert werden. Zusätzlicher Overhead wird erzeugt.

Externe Treiber: An das zu messende System wird ein Treiber-System gekoppelt. Dieses sendet u Scripts an das zu messende System und unternimmt alle Zeitmessungen. Es wird kein zusätzlicher Overhead durch die Simulation von Denkzeit-, Übertragungsweg-Simulation oder Zeitmessungen erzeugt. Auslastungen müssen am zu messenden System aufgezeichnet werden. Der für das Treiber-System notwendige Hardware-Aufwand lässt dieses Verfahren z.Zt. noch selten zum Einsatz kommen. Es ist jedoch anzunehmen, dass mit fallenden Hardware-Preisen externe Treiber in Zukunft häufiger eingesetzt werden ^{11,12}.

Alle Benchmark-Experimente in dieser Arbeit wurden nach der Methode der automatisierten Eingabe gefahren.

Zur Steuerung der Denkzeit-Intervalle wird in den folgenden Scripts jeweils vor dem Aufruf eines Kommandos ein Prolog durchlaufen, der den Beginn des folgenden Kommandos um ein spezifiziertes Zeitintervall suspendiert. Die Erzeugung eines zyklischen oder periodischen Kommandosablaufs wird durch den Aufruf eines Zufallszahlengenerators

-
- 9 - FOGEL/WINOGRAD: Einstein: An Internal ..., <FOGE72>, S. 6-14.
10 - MÜHLENBEIN: Ein Timesharing Benchmark..., <MUEH77.1>.
11 - GREENBAUM: A Simulator of Multiple..., <GREE69>.
12 - IBM: TPNS, Teleprocessing Network Simulator, <IBMO76.8>.

vermieden. Dieser wählt gleichverteilt Denkzeit-Werte Z_0 aus dem symmetrisch um den Script-Parameter Z gelegenen Intervall $Z + Z/10$ aus.

6.2. Dynamischer Benchmark, Pseudo- und Quasi-Benchmark.

Timesharing-Benchmarks sollen je nach der Art ihrer Konstruktion und ihres Einsatzes als

Quasi-Benchmark (QBM),
Pseudo-Benchmarks (PBM) oder
dynamischer Benchmark (DBM)

bezeichnet werden. Dabei sind die Bezeichnungen 'Quasi-Benchmark' und 'dynamischer Benchmark' in Anlehnung an die Arbeit von WINDFUHR entstanden¹³, der sie in analoger Weise für die Leistungsermittlung von im Stapelbetrieb eingesetzten, industriell genutzten DVA benutzt hat. Pseudo- und dynamischer Benchmark sind Benchmarks i.e.S., während der Quasi-Benchmark parallel zur Betriebslast in das System gebracht wird.

Quasi-Benchmarks werden Scripts genannt, die - von einem Terminal gestartet - automatisch während der Betriebszeit ablaufen ("snooper terminals"¹⁴). Ihre Verweilzeit unter diesen (normlen Betriebs-) Bedingungen wird mit ihrer Verweilzeit als Benchmark verglichen. Da sich die Betriebslast bei Wiederholung eines Quasi-Benchmark-Experiments verändert hat, sind die Ergebnisse der Experimente *n i c h t* reproduzierbar; die sind erst nach mehrfacher Wiederholung statistisch verwertbar.

Quasi-Benchmarks erfordern nur einen geringen Aufwand, sie sind also sehr leicht einzusetzen. Eine mehrfache Wiederholung ist notwendig, um die mitlaufende Betriebslast statistisch mitbewertbar zu machen.

Zur Bestimmung der Gewinnfunktion und der funktionsklassenweisen Produktivitätsgewinne (vgl. Kap. 4.6.) werden Quasi-Benchmarks eingesetzt (vgl. Beispiel in Kap. 8.2.6.).

¹³ - WINDFUHR: Methoden und Verfahren..., <WIND77>, S. 24 und 79.

¹⁴ - HELLERMAN/CONROY: Computer Syst. Perf., <HELL75>, S. 247.

Pseudo-Benchmarks heissen Scripts, die mit angepasster Last, aber mit geringen Benutzerzahlen gefahren werden, um den Aufwand für das Benchmarking zu reduzieren. Sie geben nur einen (oder wenige benachbarte) Punkte am unteren Ende der Systemleistungskurve (s. S. 59) an. Um eine Lastanpassung zu erreichen, sind die Denkzeiten i.a. stark verringert (gegenüber dem dynamischen Benchmark, siehe im folgenden Abschnitt).

Mit Pseudo-Benchmarks werden Auslastungszustände der Systemkomponenten simuliert, *n i c h t* jedoch die realen Lastzustände (auf der Benutzerseite (wie Denkzeiten, Anzahl der Terminals)).

Mit dem dynamischen Benchmark wird die Systemleistungskurve (siehe Formeln 4.8. und 4.9., Kap. 4.) eines TS-Systems unter bekannten, durch Scripts definierte Last-Bedingungen ermittelt. Dies erfordert ein Benchmark-Experiment für jeden zu bestimmenden Punkt (u, L) der Leistungskurve.

Der dynamische Benchmark wird so parametrisiert (s. S. 85/86), dass eine Anpassung an die Verteilung der Funktionen (funktionsklassenweise Anpassung) und die Nachbildung der mittleren Betriebslasten gewährleistet ist. Im Gegensatz zum Pseudo-Benchmark bildet er auch alle von den Benutzern erzeugte Lastparameter (Denkzeiten, Anzahl der Terminals und Transaktionsraten usw.) nach.

Der dynamische Benchmark gibt Auskunft über den optimalen Betriebspunkt des TS-Systems, der maximalen Systemleistung $L_{\max} = L(u_0)$ und damit über die optimale Anzahl unterstützbarer Terminals. Ebenso lässt sich mit ihm das Antwortzeitverhalten $SRT(u)$ feststellen.

Er liefert damit die meisten und genauesten Aussagen zur Leistungsermittlung und -Beurteilung von TS-Systemen. Er hat gleichzeitig den Nachteil, bei der Erstellung und seiner Durchführung den grössten Aufwand mit sich zu bringen.

6.3. Basis-Scripts.

Definition: Basis-Scripts sind parametrierbare Scripts, die Lasten aus verschiedenen Funktionsklassen des TS-Systems nachbilden.

Da Basis-Scripts auch Funktionen anderer als nur der ihnen zugeordneten Funktionsklasse enthalten, werden die Argumente eines jeden Basis-Scripts so gewählt, dass die Funktion ihrer Klasse deutlich hervortritt. Für diese Arbeit war es ausreichend, drei verschiedene Basis-Scripts zu definieren. Jedes dieser Basis-Scripts repräsentiert eine Funktionsklasse gemäss Kap. 5. (s. S. 70).

Transaktionsorientiertes Basis-Script.

Ein transaktionsorientiertes Script ist eine Folge von Kommandos, die nur geringe Anforderungen an die Betriebsmittelvergabe stellt. (sog. 'triviale' Kommandos, vgl. Kap. 5.). Mit ihnen lassen sich in der Form eines synthetischen Scripts hohe Transaktionsraten erzeugen, wenn die Denkzeiten zwischen den Transaktionen klein gehalten werden.

I/O-orientiertes Basis-Script.

Ein Script soll I/O-orientiert heissen, wenn es besondere Anforderungen an das I/O-Subsystem stellt. Dabei können die I/O-Operationen auf ein einzelnes Gerät beschränkt bleiben, vom Betriebssystem nach den Betriebsmittelvergabe-Algorithmen innerhalb einer Gerätegruppe oder aus allen nach den Systemspezifikationen zulässigen Geräten ausgewählt werden. In den beiden letzten Fällen werden Teile des Betriebssystems mitbewertet, im ersten Fall wird die Leistung des Gerätes mit seinen Datenpfaden gemessen.

Bei dem I/O-orientierten Script soll die I/O-Menge (Anzahl der I/O-Operationen) parametrierbar sein. Dies muss im Hinblick auf die Strategie zur Zusammensetzung von Scripts zur Betriebslast-Nachbildung gefordert werden. Über diesen Parameter erfolgt die Fein-Anpassung der I/O-Rate der Basis-Scripts (siehe Kap. 6.5.) an den Zielwert.

CPU-orientiertes Basis-Script.

CPU- oder rechenzeit-orientierte Scripts sollen Kommandofolgen heissen, bei denen der CPU-Zeit-Verbrauch parametrisiert werden kann.

Bei der Nachbildung relevanter Lasten werden durch Mischung dieser drei Basis-Scripts (in der Form von Script-Potenzen, -Produkten und -Polynomen, siehe S. 81) Benchmarks gebildet, für die die Anpassung an die betrieblichen Lasten angestrebt wird.

Die Zerlegung der künstlichen Last in verschiedene Basis-Scripts ermöglicht es dem Experimentator, bei Veränderungen der Belastung eines Betriebsmittels durch die reale Last bei der darauffolgenden Anpassung der Scripts mit Veränderungen der Parameter nur eines Basis-Scripts auszukommen (Separierung der Last-Generatoren). Dazu müssen allerdings

- die mittlere I/O-Rate des I/O-orientierten Basis-Scripts (I/O-Treiber) deutlich grösser sein als bei den beiden anderen Basis-Scripts und
- der CPU-Verbrauch des CPU-orientierten Scripts (CPU-Treiber) deutlich höher liegen als bei den beiden andern Basis-Scripts.

Die Parameter der Basis-Scripts.

Die in dieser Arbeit verwendeten Basis-Scripts sollen mit

- T - für das transaktionsorientierte,
- I - für das I/O-orientierte und
- C - für das CPU-orientierte

Basis-Script bezeichnet werden. Alle drei Basis-Scripts besitzen die in der Tabelle 6.1., S. 79, angegebene Grobstruktur.

Die Beginn- und Endphasen ('Eröffnung' und 'Abschluss') aller drei Basis-Scripts enthalten die gleichen Kommandos. Sie dienen der Synchronisierung der Scripts, der Initialisierung von Variablen, der Zuordnung von Dateien und ihrer Freigabe. Die 'Vorbereitung' für den Aufruf des Script-Kerns ist Script-spezifisch; für das I-Script wird hier die Grösse und die physische Struktur der im Script-Kern bearbeiteten Datei festgelegt.

Tabelle 6.1. Grobstruktur der Basis-Scripts.

Phase	Anzahl der Kommandos in Script		
	T	I	C
Eröffnung	9	9	9
Vorbereitung	6	16	5
Kern	$(10+2t_T)n_T$	$6n_I$	$5n_C$
Abschluss	12	12	12

n_T, n_I, n_C sind die Durchlaufzahlen für die Script-Kerne; t_T steuert die Anzahl der im T-Kern aufgerufenen Editor-Unterkommandos.

Die Kerne der Basis-Scripts sind verschieden (s. Liste im Anhang, Kap. 9.3.). Die Anzahl der Kern-Durchläufe ist für jedes Script ein Parameter (n_T, n_I, n_C).

Der Kern von T besteht aus einer Einsprung in den Editor mit nachfolgenden Unterkommandos des Editors. Der Parameter t_T steuert die Anzahl der Editor-Unterkommandos im T-Kern ($=4+2t_T$).

Das I/O-orientierte Script I besitzt neben der Kern-Durchlaufzahl n_I den Parameter i_I ('I/O-intensity'), der in der Vorbereitungsphase von I die Grösse der im I-Kern n_I -fach zu kopierenden Datei festlegt. Die physische Struktur der Datei kann prinzipiell in der Vorbereitung von I verändert werden; sie wird für alle Anwendungen dieser Arbeit jedoch festgehalten (Blocklänge BS = 3120 Bytes, Satzlänge RS = 80 Bytes, feste Blockung, sequentielle Verarbeitung). i_I gibt die Grösse der Datei in Einheiten von 40 Bytes (halbe Satzlänge) an.

Daraus lässt sich die Grösse der Datei (DS) berechnen:

$$DS = i_I \cdot RS/2 = 40 i_I \quad (\text{in Bytes}).$$

Für die Anzahl der I/O-Operationen (Lesen und Schreiben) des Kopiervorgangs im Script-Kern von I ergibt sich:

$$IO = 2 DS / BF,$$

wobei BF der Blockungsfaktor für die Datei aufgerundet BS/RS ist.

Der CPU-Treiber C soll eine Steuerung des CPU-Zeit-Verbrauchs während des Benchmarking ermöglichen. Die Parametrierung erfolgt über den Parameter c_C , der die Anzahl der Ausführung von drei Programmschleifen eines Lademoduls steuert. Die Programmschleifen werden jeweils c_C^2 - mal nach dem Aufruf des CALL-Kommandos im Kern des C-Scripts (siehe Kap. 9.3., Seite 171) durchlaufen. Die Programmschleifen enthalten eine Addition, eine Multiplikation und eine Division mit Wertzuweisung, alle Operanden sind doppelt genau Gleitpunktzahlen.

Der Denkzeit-Parameter Z wird für jedes Basis-Script gleich angenommen.

Damit sind die drei Basis-Scripts T, I und C durch einen Satz von sieben Parametern beschrieben, wovon je zwei script-spezifisch sind und einer für alle Scripts durchgehend gleich:

$$\begin{aligned} T &= T(n_T, t_T, Z), \\ I &= I(n_I, i_I, Z) \text{ und} \\ C &= C(n_C, c_C, Z). \end{aligned}$$

Die Anpassung der Benchmark-Lasten an die betrieblich relevanten erfolgt durch

- parallelen und sequentiellen Ablauf der Basis-Scripts (Bildung von Script-Potenzen, -Produkten und -Polynomen),
- Anpassung der Parameter $n_T, n_I, n_C, t_T, i_I, c_C$ und Z beim dynamischen Benchmark oder
- Auswahl von 'Mischungen' der Basis-Scripts (Potenzen, Produkte oder Polynome) bei festgehaltenen Parametern n_T, \dots, Z für den Pseudo-Benchmark.

6.4. Script-Potenzen, -Produkte und -Polynome.

Definition: Parallel ablaufende Basis-Scripts heissen Script-Potenzen. Der Grad einer Script-Potenz ist die Anzahl der parallel ablaufenden Scripts.

Beispiel: I^5 bezeichnet eine Potenz, die aus 5 parallel ablaufenden I/O-orientierten Basis-Scripts besteht.

Definition: Parallel ablaufende Script-Potenzen heissen Script-Potenz-Produkte oder Script-Produkte. Der Grad eines Produkts ist die Summe der Grade der produktbildenden Potenzen.

Beispiel: I^5C^3 bezeichnet ein Produkt mit dem Grad 8, das aus 5 I-Scripts und 3 C-Scripts gebildet wird (alle 8 Scripts laufen parallel ab).

Definition: Script-Produkte sollen *homogen* heissen, wenn die sie bildenden Potenzen alle gleichen Grad besitzen.

$$\text{Beispiel: } T^3 I^3 C^3 = (T I C)^3 = S_9(T, I, C).$$

Script-Produkte garantieren die Verarbeitung von mehreren gleichzeitig ablaufenden Basis-Scripts. Für den dynamischen Benchmark wird der Grad der Produkte der Anzahl der laufenden Sitzungen entsprechen. Für Auslastungsuntersuchungen mit dem Pseudo-Benchmark sind Produkte geringeren Grades als für den dynamischen Benchmark ausreichend.

Definition: *Sequentiell* ablaufende Script-Produkte heissen Script-Polynome. Der höchste Grad der in ihm enthaltenen Produkte ist der Grad des Polynoms. Sind alle Produkte in einem Polynom homogen und bilden stets die gleichen Basis-Scripts die Produkte, so ist das Polynom symmetrisch (in T, I und C).

Beispiel: $P = 4 T^2 I^2 + 2 I^5 C^2$; dies bedeutet den 4-fachen Ablauf (hintereinander) des (homogenen) Produkts $T^2 I^2$, danach den 2-fachen Ablauf des (inhomogenen) Produkts $I^5 C^2$.

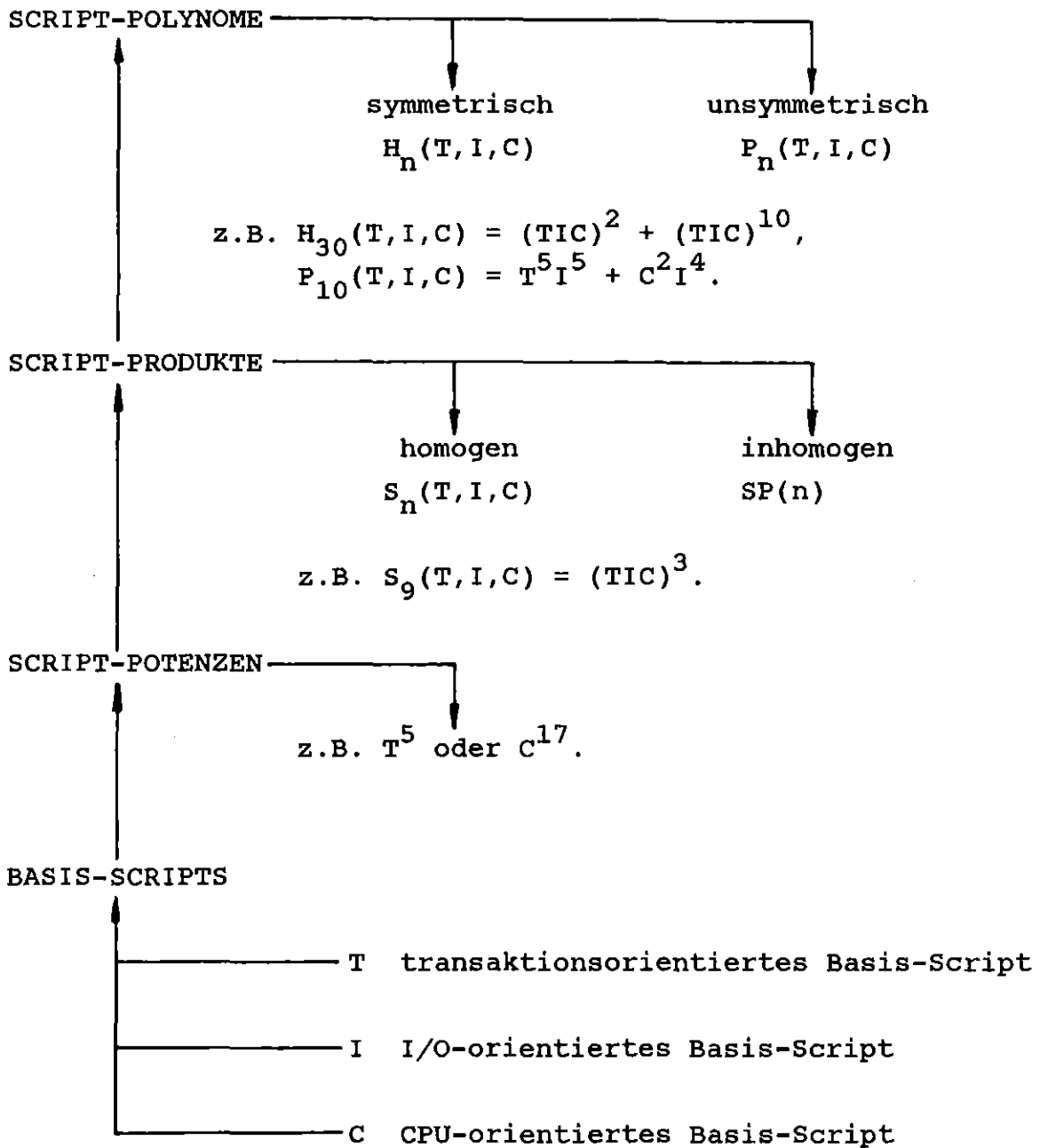
Beispiel: $H_{30} = T I C + (T I C)^3 + (T I C)^{10}$ ist ein symmetrisches Polynom 30. Grades.

Sequentiell ablaufende Scripts werden eingesetzt, um bei unveränderten Script-Parametern allein durch Aneinanderreihung der Scripts eine Lastanpassung zu gewährleisten.

Dynamische Benchmarks werden mit symmetrischen Script-Polynomen gefahren, sie werden zur Bestimmung der Systemleistung $L(u)$ herangezogen (Kap. 4.2., S. 56). Polynome (i.a. unsymmetrisch) werden zur vereinfachten Nachbildung von (statischen) Auslastungssituationen als Pseudo-Benchmark eingesetzt. Diese können nicht zur Ermittlung der Systemleistung unter realen Benutzerlasten verwendet werden.

Die Nomenklatur der Scripts wird in Abbildung 6.1. festgelegt.

Abbildung 6.1. Übersicht über die Script-Strukturen und ihre Nomenklatur.



6.5. Lastanpassung der Scripts.

6.5.1. Dynamischer Benchmark.

Für den dynamischen Benchmark sollen die Basis-Script-Parameter so bestimmt werden, dass die Verteilung der Kommandos gem. Tab. 5.2., S. 70, das Funktionsspektrum der natürlichen Betriebslast (funktionsklassenweise Anpassung) und der mittlere Betriebsmittelverbrauch approximiert werden.

Funktionsklassenweise Anpassung.

In den drei Basis-Scripts T, I und C sind die Parameter n_T , n_I , n_C und t_T eingeführt worden, um mit ihnen homogene Script-Produkte an die gemessene Benutzungshäufigkeit der Kommandos der Betriebslast anzupassen. Sie bestimmen die Durchlaufzahlen der Scriptkerne und der inneren Schleife des Transaktionstreibers (T-Script, s. S. 170). Tabelle 6.2. gibt die Anzahl der Kommandos der verschiedenen Funktionsklassen in Abhängigkeit von den Parametern an. Es ist zu beachten, dass die Script-Parameter i_I , c_C und Z nicht erscheinen.

<u>Tabelle 6.2. Zuordnung der Kommandos zu den Funktionsklassen in Abhängigkeit von den Parametern der Basis-Scripts.</u>				
Basis-Script	Funktionsklasse			
	TRX-orient.	I/O-orient.	CPU-orient.	Summe
T (trans-or.)	$9+n_T(4+2t_T)$	$18+6n_T$	0	$27+n_T(10+2t_T)$
I (I/O-or.)	9	$26+6n_I$	2	$37+6n_I$
C (CPU-or.)	8	$18+4n_C$	n_C	$26+5n_C$
Summe	$26+n_T(4+2t_T)$	$62+6(n_T+n_I)+4n_C$	$2+n_C$	$90+n_T(10+2t_T)+6n_I+5n_C$

Für den dynamischen Benchmark werden homogene Produkte, welche alle drei Basis-Scripts enthalten, eingesetzt, also

$$s_{3j}(T, I, C) = s_{3j} = T^j I^j C^j$$

$$s_{3j} = T^j(n_T, t_T, Z) I^j(n_I, i_I, Z) C^j(n_C, c_C, Z)$$

mit $j=1,2,3,\dots$

Nach Tabelle 5.2. besteht das Funktionsspektrum der Betriebslast aus den in Tabelle 6.3. angegebenen Anteilen der einzelnen Funktionsklassen.

<u>Tabelle 6.3. Anteile der Funktionsklassen bei der Betriebslast.</u>	
Funktionsklasse	Anteil in %
transaktions-orientiert	61
I/O-orientiert	36
CPU-orientiert	3

Die (homogenen) Produkte s_{3j} enthalten $90+n_T(10+2t_T)+6n_I+5n_C$ Kommandos, die sich nach Tab. 6.2. auf die verschiedenen Funktionsklassen verteilen. Es werden nun die Parameter n_T , t_T , n_I und n_C so bestimmt, dass die funktionsklassenweise Verteilung der Kommandos der Script-Produkte s_{3j} an die Verteilung in der Betriebslast angepasst werden.

Mit den gerundeten Ergebnissen

$$n_T = n_I = 20 \text{ und}$$

$$t_T = n_C = 10$$

erhält man für alle Produkte s_{3j} die funktionsklassenweise

Kommandoverteilung in Tabelle 6.4. Diese Anpassung liefert eine ausreichende Übereinstimmung mit den Messwerten in der Tabelle 6.3.

Tabelle 6.4. Anteile der Funktionsklassen bei den homogenen Script-Produkten $S_{3j} = (T I C)^j$ mit den Parametern $n_T = 20$, $n_I = 20$, $n_C = 10$, $t_T = 10$.

Funktionsklasse	Anteil in %	Abweichung in % von Tab. 6.3.
transaktions-orientert	59	-2
I/O-orientiert	39	+3
CPU-orientiert	2	-1

Anpassung des Betriebsmittelverbrauchs.

Für die als dynamischen Timesharing-Benchmark zu fahrenden homogene Script-Potenzen S_{3j} sind die Parameter n_T , n_I , n_C und t_T als Folge der funktionsklassenweisen Anpassung festgelegt. Daher ist

$$\begin{aligned}
 S_{3j} &= (T I C)^j \\
 &= (T(Z) I(i_I, Z) C(c_C, Z))^j
 \end{aligned}$$

Ziel dieses Abschnittes ist es, die Parameter i_I , c_C und Z so bestimmen, sodass die Last des dynamischen Benchmarks den Mittelwerten der Betriebslast (der Transaktionsrate (TRX), der I/O-Rate (TS-IO) und dem CPU-Zeit-Verbrauch (TS-CPU) bezogen auf die Anzahl der Terminals) entspricht.

Für ein Verfahren, das schnell zu einer guten Näherung führt, ist es wesentlich, den Einfluss von i_I und c_C weitgehend lokal zu halten, d.h. ihre Wirkung auf die anderen Scripts gering zu halten. Die Basis-Scripts wurden daher so festgelegt, dass i_I (Parameter zur I/O-Raten-Anpassung) nur im I-Script und c_C (Parameter zur CPU-Zeit-Verbrauch-Anpassung) nur im C-Script auftritt.

Die Transaktionsrate (TRX), die I/O-Rate (SIO) und der CPU-Zeit-Verbrauch (CPU) des dynamischen Benchmarks werden sehr stark vom Parameter Z (Denkzeit) beeinflusst. Dagegen sind Transaktions- und I/O-Rate sehr unempfindlich gegenüber Veränderungen von c_C ; Transaktionsrate und CPU-Zeit-Verbrauch sind unempfindlich gegenüber einer Variation von i_I (s. Abb. 6.3. und 6.5., S. 90 und S. 92). Die Abb. 6.4., S. 91, zeigt die angestrebte starke Abhängigkeit der I/O-Rate vom Parameter i_I .

Die Lösung des Näherungsproblems erfolgt durch einen Wechsel von Schätzung und Verifikation. Nach jeder Neu-Schätzung eines der Parameter Z , i_I oder c_C müssen durch eine Messung der Lastparameter des Scripts diese mit den Zielwerten verglichen werden. Die Anfangsschätzung muss aufgrund guter Systemkenntnisse oder aufbauend auf bereits vorhandenen Erfahrungswerten erfolgen.

Es seien die Funktionen

$$\text{TRX} = g_1(S_3) = g_1(i_I, c_C, Z)$$

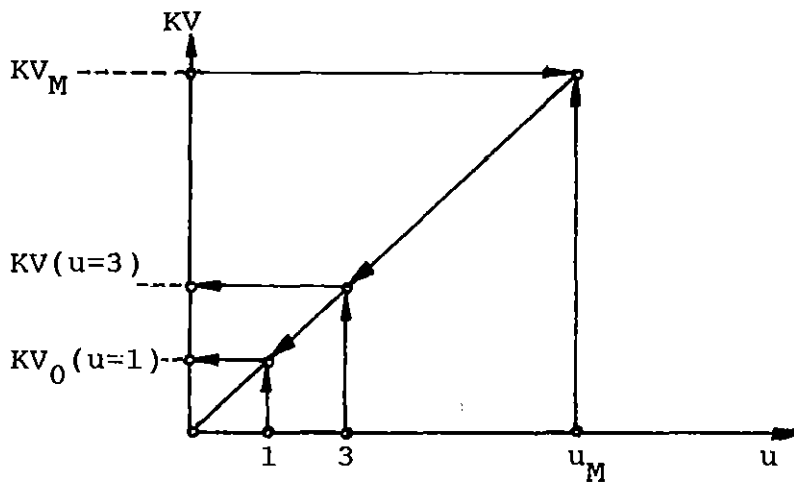
$$\text{SIO} = g_2(S_3) = g_2(i_I, c_C, Z)$$

$$\text{CPU} = g_3(S_3) = g_3(i_I, c_C, Z)$$

die Transaktionsrate (TRX), die I/O-Rate (SIO) und der CPU-Zeit-Verbrauch (CPU) des homogenen Produkts S_3 (also $j=1$) vom Grad 3 (alle Basis-Scripts treten genau einmal auf). Dies ist das homogene Produkt mit dem kleinsten Grad, das alle drei Basis-Scripts enthält. Es stellt für den dynamischen Benchmark die kleinste, technisch realisierbare Einheit dar. Dieses Script heisst Minimal-Produkt des dynamischen TS-Benchmarks.

Die Anpassung der Benchmarklast erfolgt durch die Anpassung der Lasten g_1, g_2, g_3 des Minimal-Produkts S_3 an die auf $u = 3$ umgerechneten Mittelwerte der Betriebslast (vgl. Abb. 6.2.).

Abbildung 6.2.: Bestimmung der Zielwerte für das Minimal-Produkt S_3 des dynamischen Benchmarks.



KV steht hier für alle drei Kontrollvariablen TRX, TS-IO, CPU. u_M und KV_M bezeichnen die Mittelwerte der Kontrollvariablen aus der Leistungsdatenbank.

$KV(u=3)$ ist der für das Minimal-Produkt S_3 anzustrebende Zielwert der Kontrollvariablen, $KV_0(u=1)$ ist Zielwert für das (hypothetische) Produkt S_1 .

Grund für dieses Vorgehen ist die Erfahrung, dass sich die Kontrollvariablen in Abhängigkeit von u mindestens im Bereich kleiner und mittlerer u -Werte linear verhalten (s. Kap. 2.1., Abb. 2.4., S. 24 für TS-IO und u , Abb. 2.8., S. 26 für SIO und u).

Zur einfacheren Darstellung werden alle Kontrollvariablen-Werte während der Anpassung auf $u = 1$ zurückgerechnet, d.h. für das Minimal-Produkt S_3 werden die Kontrollvariablen-Werte im Mittel pro Basis-Script berechnet (durch $u = 3$ dividiert). Die Lasten des so erzeugten - hypothetischen - Scripts 1. Grades (es wird mit S_1 bezeichnet) sind:

$$\text{TRX} = f_1(S_1) = f_1(i_I, c_C, Z) = g_1/3$$

$$\text{SIO} = f_2(S_1) = f_2(i_I, c_C, Z) = g_2/3$$

$$\text{CPU} = f_3(S_1) = f_3(i_I, c_C, Z) = g_3/3$$

Diese Werte sind durch Variation der Parameter i_I , c_C und Z an die Zielwerte TRX_0 , SIO_0 und CPU_0 (s. Abb. 6.2.), die aus der Leistungsdatenbank ermittelt wurden, anzupassen. Die Abbildung 6.6. (S. 93) gibt das allgemeine Verfahren dafür in Form eines Ablaufdiagramms an.

Abbildung 6.3.: Die Abhängigkeit der mittleren Transaktionsrate des Produkts S_1 von der mittleren Denkzeit.

$$\text{TRX} = f_1(i_I, c_C; Z)$$

Parameter $c_C = 750$ (fest),

Parameter $i_I = 1000, 2000, 4000, 8000$.

Transaktionsrate,

$\text{TRX} = f_1(i_I, c_C; Z)$ (in 1/s)

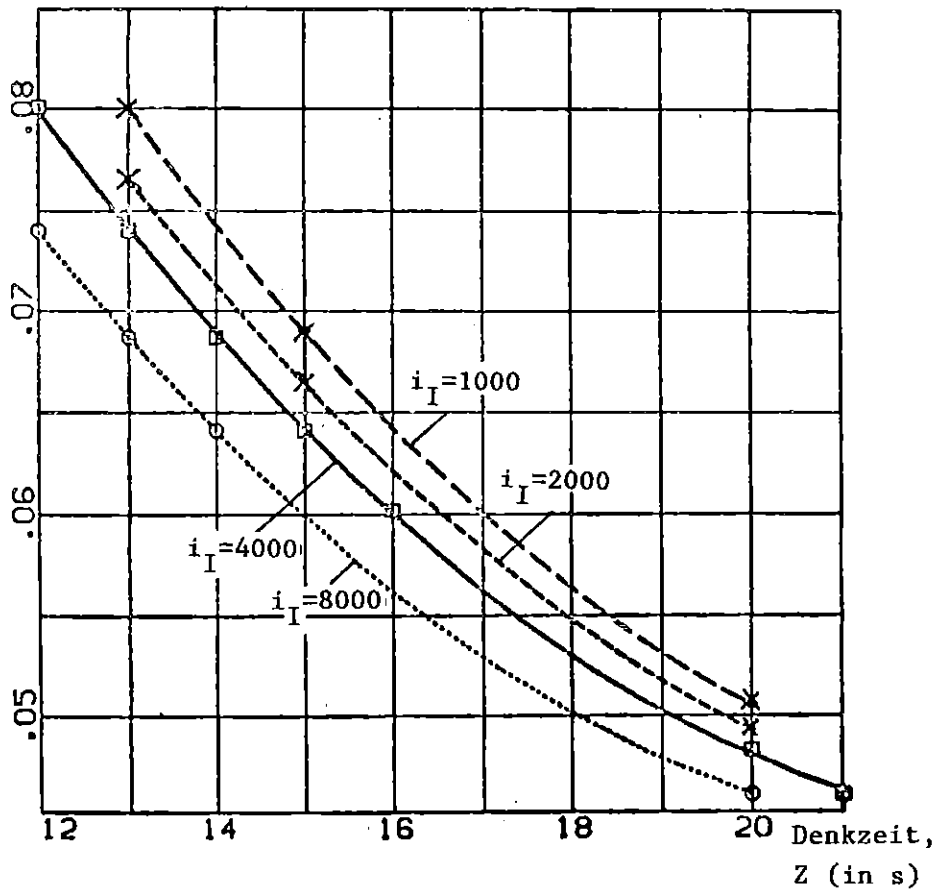


Abbildung 6.4.: Die Abhängigkeit der mittleren I/O-Rate des Produkts S_1 von der mittleren Denkzeit.

$$SIO = f_2(i_I, c_C; Z)$$

Parameter $c_C = 750$ (fest),

Parameter $i_I = 1000, 2000, 4000, 8000$.

I/O-Rate,
 $SIO = f_2(i_I, c_C; Z)$ (in 1/s)

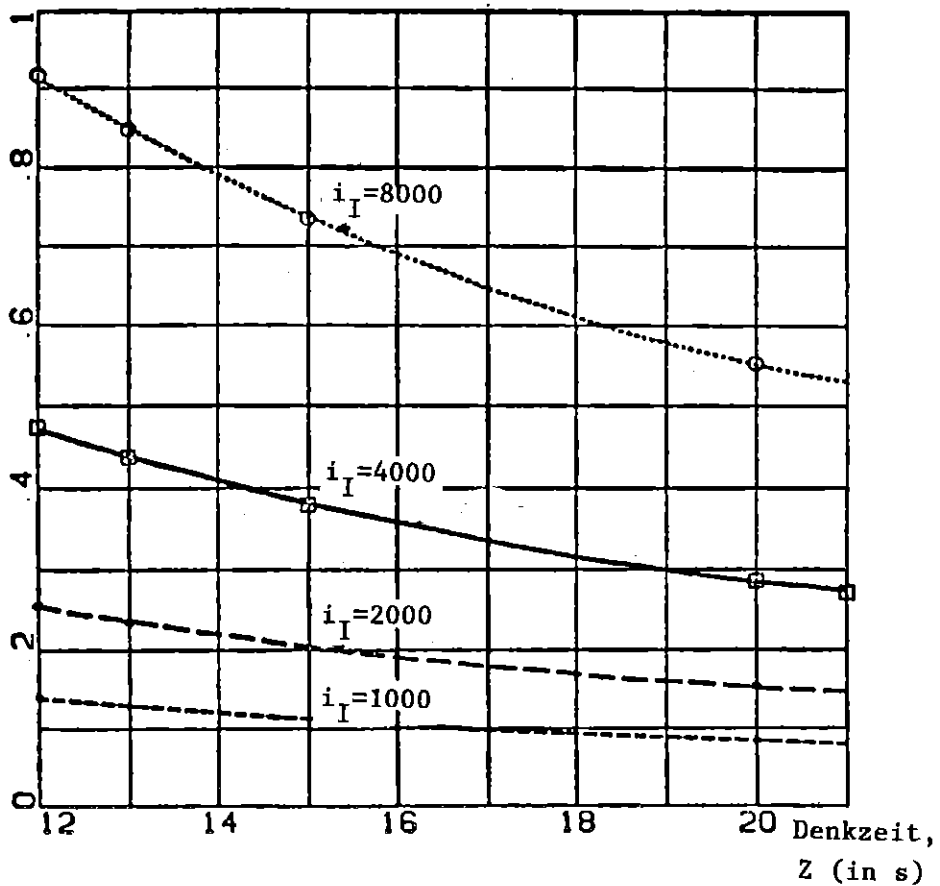


Abbildung 6.5.: Die Abhängigkeit des mittleren CPU-Zeit-Verbrauchs des Produkts S_1 von der mittleren Denkzeit.

$$\text{CPU} = f_3(i_I, c_C; Z)$$

Parameter $c_C = 750$ (fest),
Parameter $i_I = 1000, 2000, 4000, 8000$.

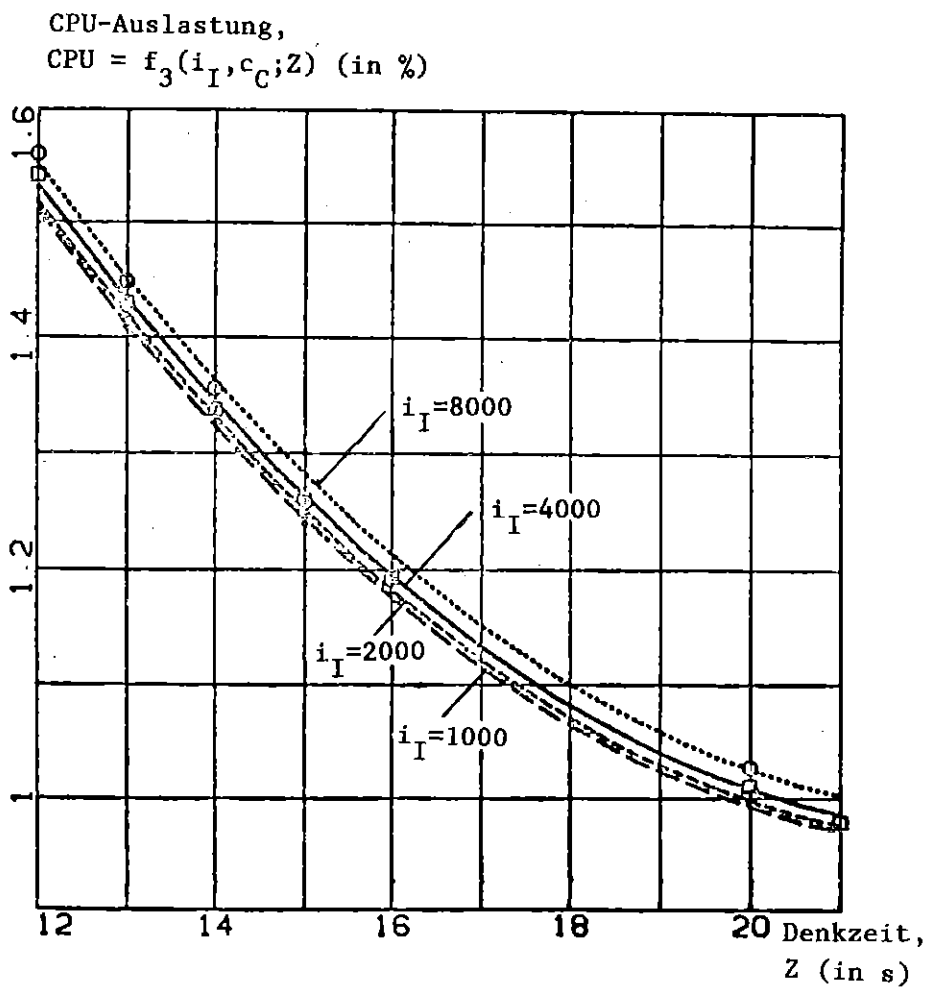
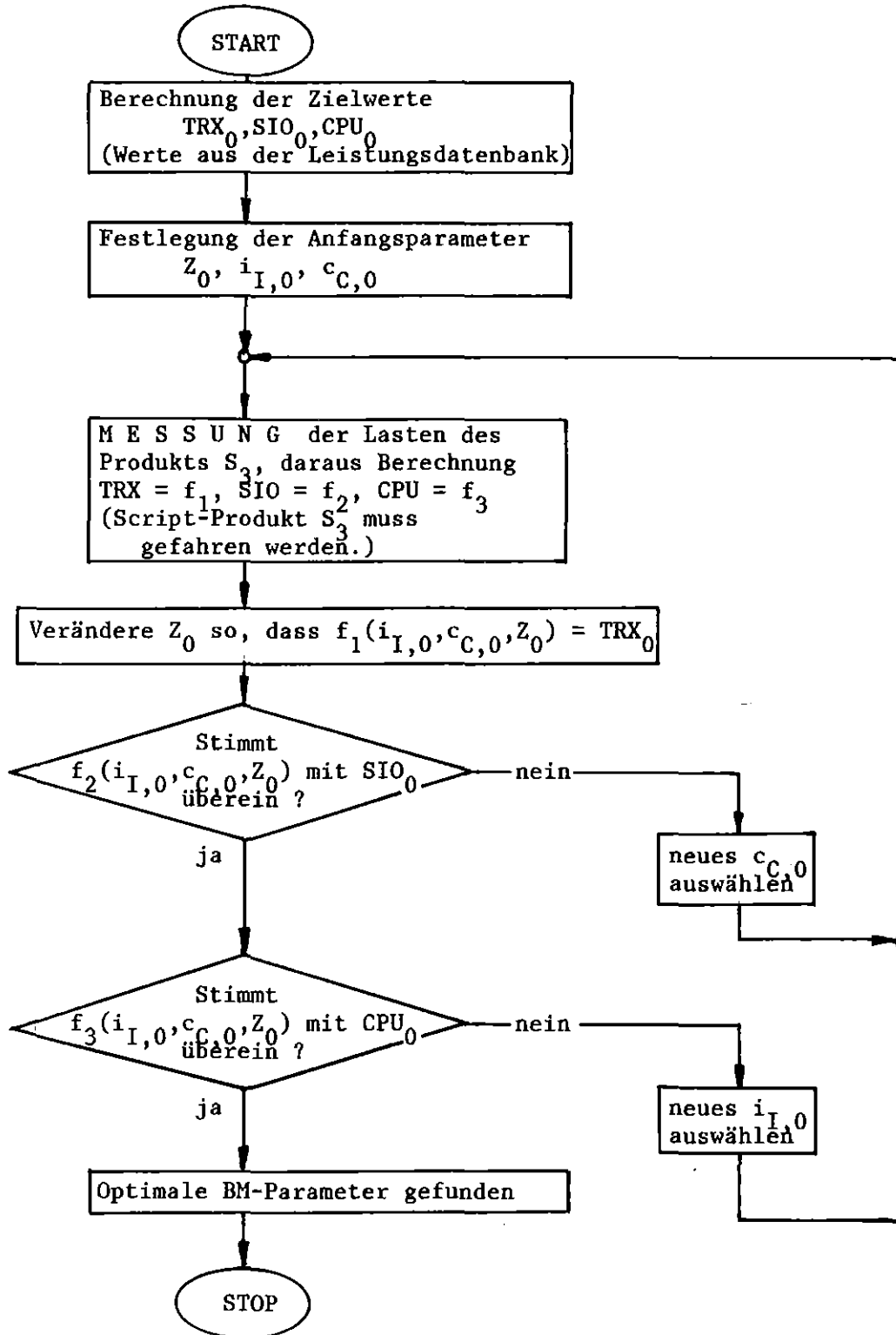


Abbildung 6.6.: Ablaufdiagramm zur Lastanpassung für den dynamischen Timesharing-Benchmark.



Beispiel für die Lastanpassung:

Für ein System IBM/370-158 mit 3 MB Hauptspeicher und einer Plattenperipherie von IBM 2314 (Benutzerdateien) und IBM 3330 (für System und Paging/Swapping), siehe System-Konfiguration 1 im Anhang, Kap. 9.1., ist der dynamische Timesharing-Benchmark an die Mittelwerte der während des August 1978 gemessenen Lastwerte anzupassen.

Lastwerte (aus der Leistungsdatenbank, auf $u = 1$ umgerechnet):

$$\text{TRX}_0 = 0.064 \quad 1/\text{s},$$

$$\text{SIO}_0 = 0.89 \quad 1/\text{s} \text{ und}$$

$$\text{CPU}_0 = 1.40 \quad \%$$

Anfangswerte der Parameter:

$$i_{I,0} = 4000 \quad (\text{Grösse der COPY-Datei}),$$

$$c_{C,0} = 750 \quad (\text{Anz. der Rechenschleifen}) \text{ und}$$

$$z_0 = 20 \quad (\text{mittlere Denkzeit}).$$

Die Messung des Minimal-Produkts S_3 für den Benchmark liefert die folgenden Lastwerte ($u=3$):

$$\text{TRX} = g_1(i_{I,0}=4000, c_{C,0}=750, z_0=20) = 0.16 \quad 1/\text{s},$$

$$\text{SIO} = g_2(i_{I,0}=4000, c_{C,0}=750, z_0=20) = 1.00 \quad 1/\text{s},$$

$$\text{CPU} = g_3(i_{I,0}=4000, c_{C,0}=750, z_0=20) = 3.20 \quad \%$$

oder für S_1 , das 'hypothetische' Script vom Grade 1, mit $u = 1$:

$$\text{TRX} = f_1(i_{I,0}=4000, c_{C,0}=750, Z_0=20) = g_1/3 = 0.05 \text{ 1/s,}$$

$$\text{SIO} = f_2(i_{I,0}=4000, c_{C,0}=750, Z_0=20) = g_2/3 = 0.3 \text{ 1/s und}$$

$$\text{CPU} = f_3(i_{I,0}=4000, c_{C,0}=750, Z_0=20) = g_3/3 = 1.1 \text{ \%}.$$

Die Abb. 6.7., S. 97, stellt die Abhängigkeit der Transaktionsrate von der mittleren Denkzeit Z dar. In dieser Abbildung sind auch die Werte der Messungen für $Z = 12, 13$ und 14 s aufgetragen.

1. Korrektur: Eine Anpassung der Transaktionsrate durch Veränderung der mittleren Denkzeit Z führt zu einem neuen Z_0 von 13 - 15 Sek. (Vgl. Abb. 6.7.). Obwohl $Z_0 = 15$ die Transaktionsrate bereits optimal approximiert, muss erst noch das Ergebnis der I/O-Anpassung abgewartet werden, bevor eine Entscheidung über die Wahl von Z_0 getroffen werden kann.

2. Korrektur: Anpassung der I/O-Rate durch Erhöhung der Dateigrösse des I/O-Treibers von $i_I = 4000$ auf 8000. Hier sind u. U. umfangreiche Experimente notwendig (d.h. mehrfache Veränderungen von i_I , auf deren Darstellung hier jedoch verzichtet wird). Abbildung 6.8. zeigt die Abhängigkeit der I/O-Rate SIO von der mittleren Denkzeit für $i_I = 4000$ (drei Experimente: $Z = 13, 14$ und 20 s, durchgezogene Linie) und $i_I = 8000$ für die vier Denkzeit-Mittelwerte $Z = 13, 14, 15$ und 16 Sekunden.

$\text{SIO}(Z=13) = 0.84 \text{ 1/s}$ und $\text{SIO}(Z>13) < 0.8 \text{ 1/s}$; die hier beste Anpassung lässt sich mit $Z_0 = 13$ erreichen, die Messung mit der Script-Potenz und einer mittleren Denkzeit von

$$Z_0 = 13 \text{ s}$$

ergibt eine mittlere I/O-Rate und einen CPU-Zeit-Verbrauch von

$$\begin{aligned} \text{SIO}(Z_0) &= 0.84 \text{ 1/s und} \\ \text{CPU}(Z_0) &= 1.45 \text{ \%}. \end{aligned}$$

Die Transaktionsrate beträgt jetzt

$$\text{TRX}(Z_0) = 0.073 \text{ 1/s}$$

(siehe Abb. 6.7., punktierte Linie). Würde man, Abb. 6.7. folgend, für $Z_0 = 14 \text{ s}$ wählen, so ergäbe sich für die I/O-Rate eine grössere Abweichung vom Zielwert SIO_0 , die Abweichung des CPU-Verbrauchs bleibt dabei für beide Z-Werte betragsmässig ungefähr gleich.

Zusammenfassung: Die Parametrierung des minimalen Script-Produkts S_3 , bestehend aus je einem TRX-, I/O- und CPU-Treiber zur Anpassung an die durch die Betriebsdaten definierten Zielwerte, ist wie folgt:

$$\begin{aligned} n_T &= 20, \quad t_T = 10, \quad n_I = 20, \quad n_C = 10, \\ Z_0 &= 13, \quad i_{I,0} = 8000, \quad c_{C,0} = 750. \end{aligned}$$

Die Messung des minimalen Produkts S_3 , auf S_1 zurückgerechnet, ergibt die folgenden Werte, mit den aufgeführten Abweichungen zu den Zielwerten:

$$\text{TRX} = f_1(Z_0, i_{I,0}, c_{C,0}) = 0.068 \text{ 1/s} \quad (-6.3 \%),$$

$$\text{SIO} = f_2(Z_0, i_{I,0}, c_{C,0}) = 0.84 \text{ 1/s} \quad (+5.6 \%) \text{ und}$$

$$\text{CPU} = f_3(Z_0, i_{I,0}, c_{C,0}) = 1.47 \text{ \%} \quad (-5.0 \%).$$

Dieses Ergebnis liefert einen mittleren absoluten Fehler von nur 5.6 %.

Mit dem hierdurch festgelegten dynamischen Benchmark wird in Kapitel 8.2. die Systemleistung $L(u)$ eines Timesharing-Systems ermittelt. Ebenfalls dort wird auch eine Leistungsvergleich zwischen verschiedenen Systemen durchgeführt.

Abbildung 6.7.: Die TRX-Rate als Funktion der Denkzeit. Messwerte zur Parametrierung des dynamischen TS-Benchmarks.

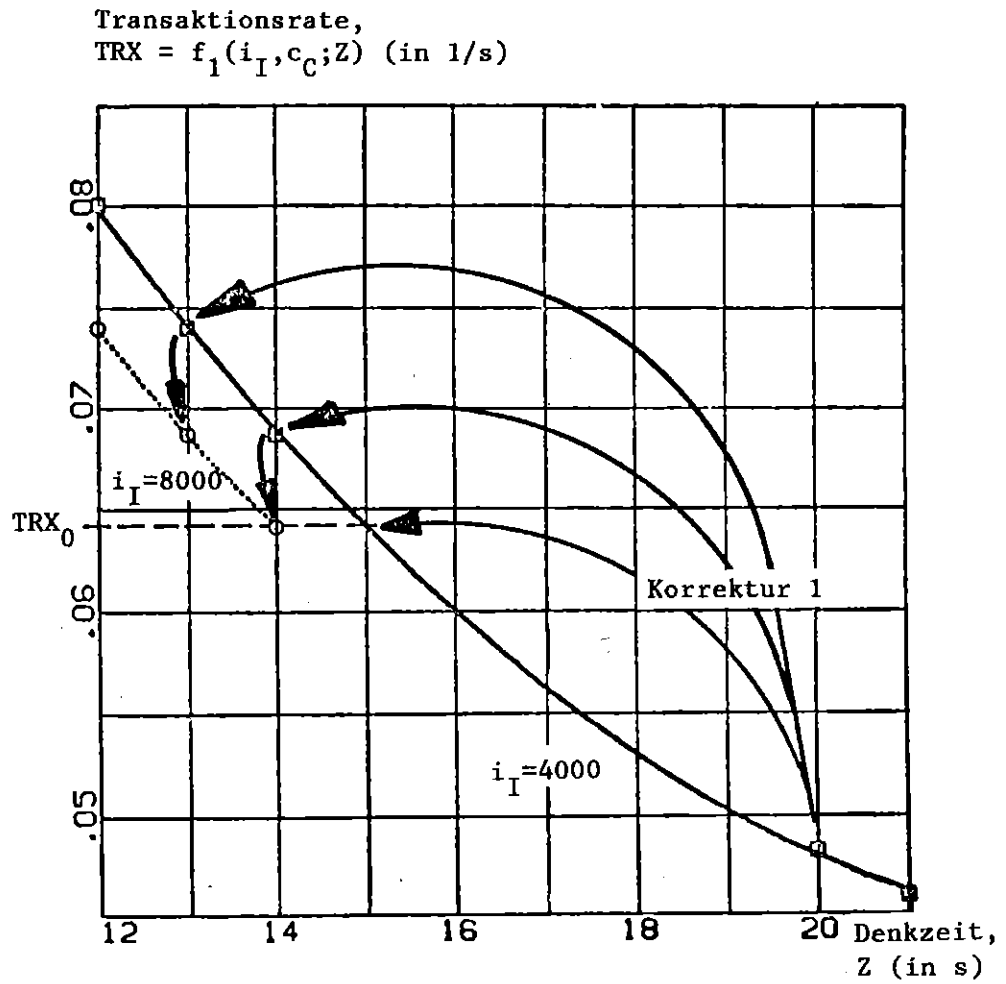
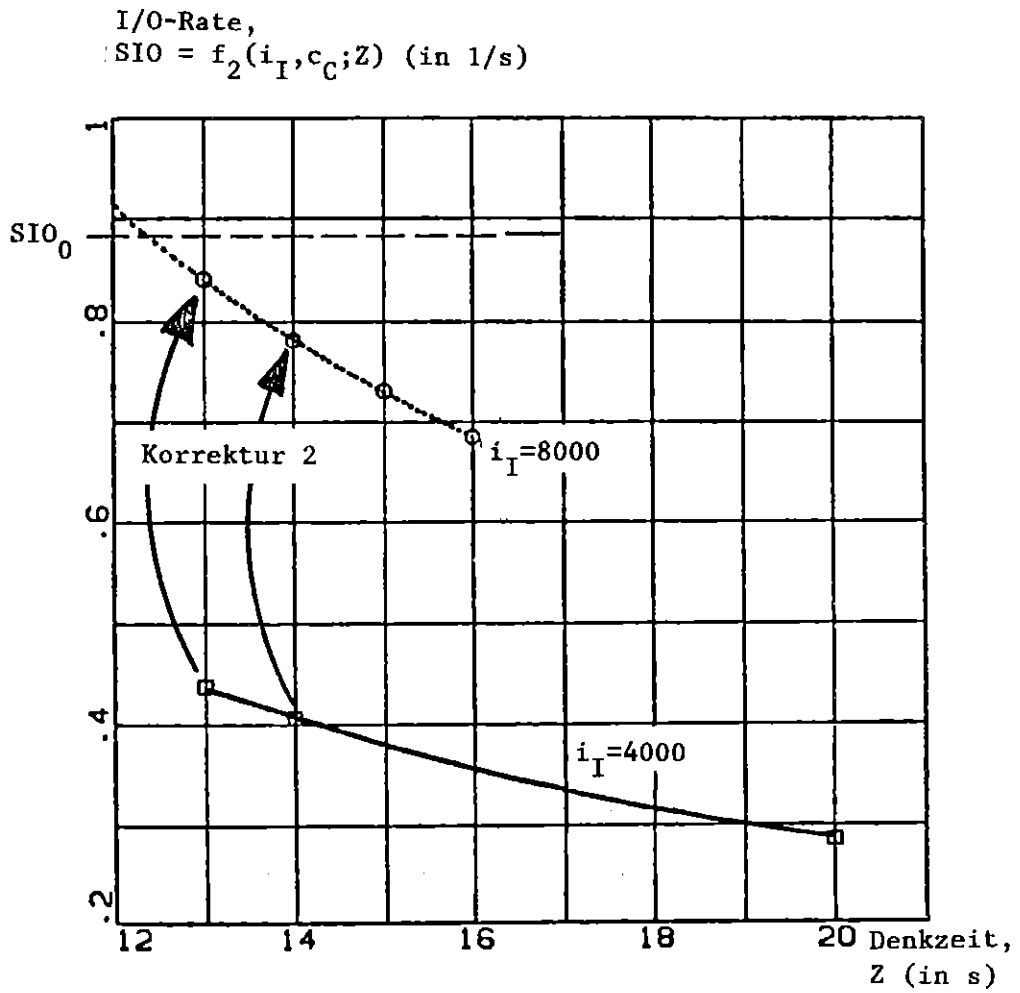


Abbildung 6.8.: Die I/O-Rate als Funktion der Denkzeit. Messwerte zur Parametrierung des dynamischen TS-Benchmarks.



6.5.2. Pseudo-Benchmarks.

Pseudo-Benchmarks dienen der Nachbildung von Auslastungssituationen; um die Verweilzeiten der künstlichen Last für den praktischen Einsatz klein zu halten, werden beim Pseudo-Benchmark die Denkzeiten gegenüber dem dynamischen Benchmark reduziert.

Im Gegensatz zum dynamischen Benchmark, bei welchem die Lasten des minimalen Script-Produkts an die Mittelwerte der Betriebslasten pro Benutzer angepasst werden, gilt es beim Pseudo-Benchmark, eine punktuelle Last (meist Mittelwerte vorgegebener Betriebszyklen, möglich sind jedoch auch andere Lasten, z.B. Hoch- oder Niedriglast-Betriebszustände) durch die Bildung eines Script-Polynoms aus einer Anzahl von Script-Produkten mit bekannten - gemessenen - Lastwerten anzunähern.

Wegen künstlich verkürzter Denkzeiten sind niedrige Grade der polynom-bildenden Script-Produkte ausreichend -, jedoch kann nicht mehr ohne weitere Untersuchungen eine Zuordnung zwischen Benutzerzahl und Systemleistung hergestellt werden. Die Bestimmung der maximalen Systemleistung kann wegen der stark reduzierten Anzahl der Benutzer nicht mit dem Pseudo-Benchmark durchgeführt werden.

Abb. 6.9. zeigt den Ablauf des Verfahrens zur Erstellung des optimalen Script-Polynoms. Nach der Last-Messung der Script-Produkte werden deren Last-Gleichungen aufgestellt. Ergebnis ist ein lineares Gleichungssystem der Form:

$$\begin{array}{l} a_{11}x_1 + \dots + a_{1j}x_j + \dots + a_{1n}x_n = b_1 \\ \dots \qquad \qquad \qquad \dots \\ a_{i1}x_1 + \dots + a_{ij}x_j + \dots + a_{in}x_n = b_i \\ \dots \qquad \qquad \qquad \dots \\ a_{m1}x_1 + \dots + a_{mj}x_j + \dots + a_{mn}x_n = b_m \end{array}$$

Abbildung 6.9.: Ablaufdiagramm zur Parametrierung eines Script-Polynoms als Pseudo-Benchmark.

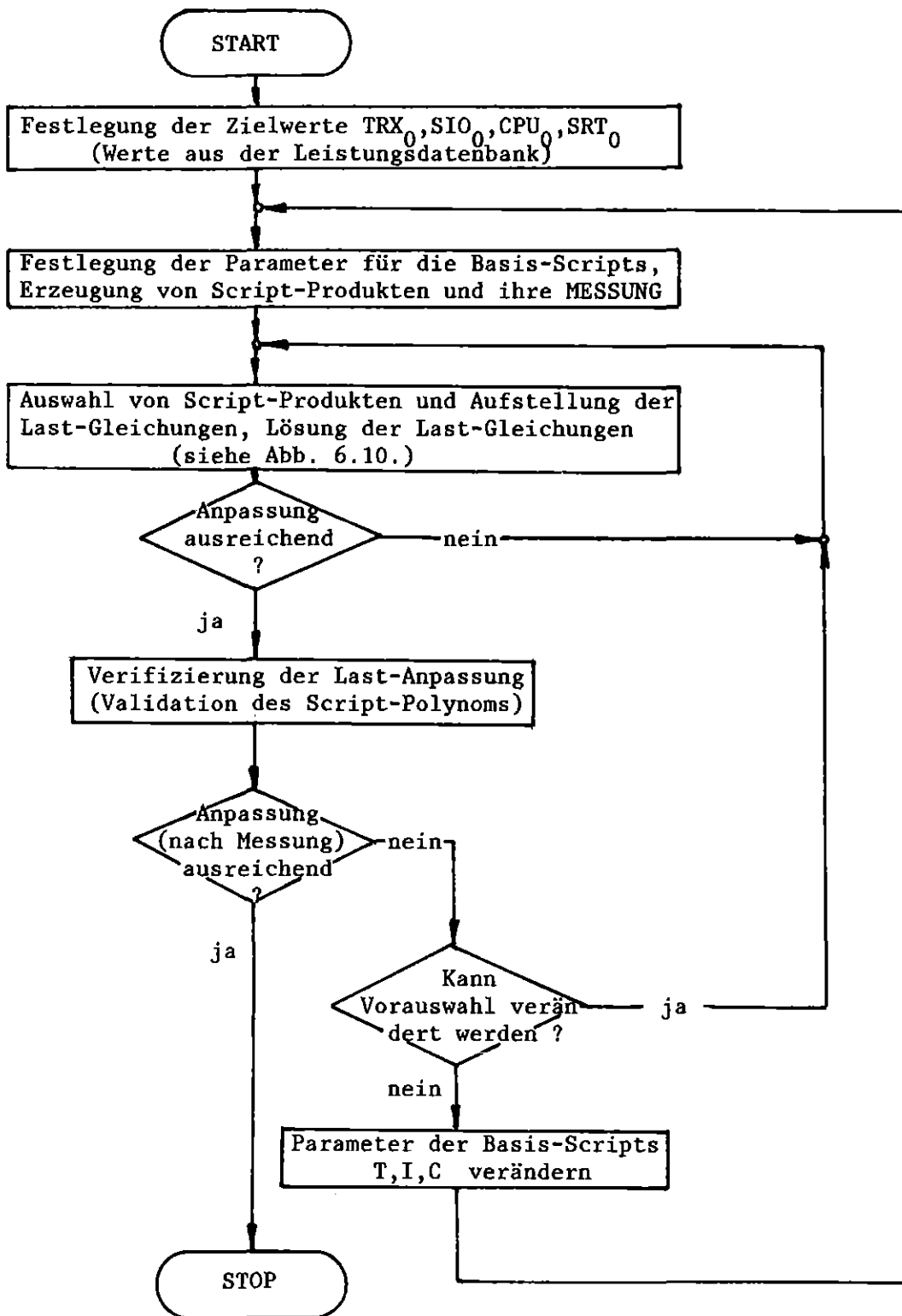
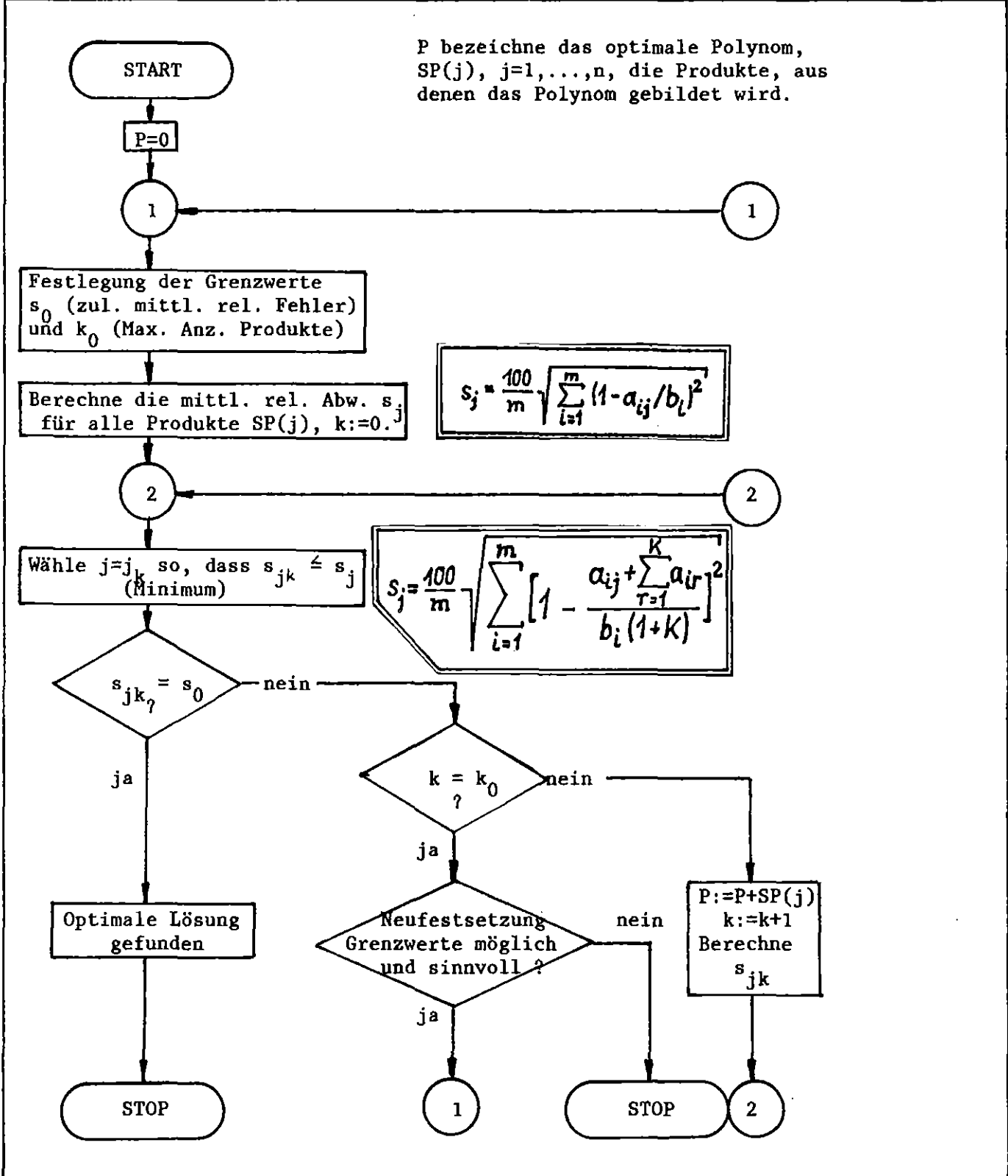


Abbildung 6.10.: Ablaufdiagramm zur approximativen Lösung der Last-Gleichungen.



wobei die a_{ij} , $i=1, \dots, m$, der gemessene Wert der Lastgrösse i des Produkts j ist, $j=1, \dots, n$ = Anzahl der verschiedenen Produkte. b_i , $i=1, \dots, m$, ist der Wert der Lastgrösse i des nachzubildenden Wertes des Betriebszyklus aus der Leistungsdatenbank.

Für die Lösungen x_j , $j=1, \dots, n$, welche die Koeffizienten der einzelnen Produkte im Polynom angeben, muss gelten:

x_j ganzzahlig und nicht negativ.

Lösungsverfahren für die Lastgleichungen.

Für den Einsatz der Pseudo-Benchmarks in der Praxis wurde ein Lösungsverfahren entwickelt, welches - ausgehend von jedem einzelnen Script-Produkt - a k k u m u l a t i v ein Script-Polynom bildet

$$P = \sum_{r=1}^n x_r SP(r),$$

P ist das zu bestimmende optimale Script-Polynom, die $SP(r)$ sind die dafür gebildeten Script-Produkte, x_r gibt jeweils die zu bestimmende Anzahl des Script-Produkts $SP(r)$ an.

Zur Begrenzung der Gesamtlaufzeit des Pseudo-Benchmarks wird nur eine Maximal-Anzahl von Produkten zur Polynom-Bildung zugelassen. Diese Anzahl beschränkt das Suchverfahren. Ebenso bricht das Verfahren bei Erreichen einer Approximationsgenauigkeitsschranke für den mittleren relativen Fehler ab. Das Verfahren ist durch zwei Ablaufdiagramme in den Abb. 6.9. und 6.10., S. 100/101, angegeben. Die Suche nach einer Anfangslösung entfällt hierbei; diese wird vom Verfahren geliefert.

Alle bisher aufgetretenen Anwendungen von Pseudo-Benchmarks in der Praxis sind nach diesem Verfahren gelöst worden.

Zwei naheliegende Verfahren, die Lösung der Last-Gleichungen durch Methoden der linearen ganzzahligen Optimierung¹⁵ zu finden und durch Anwendung des Lösungsalgorithmus für diophantische Gleichungen¹⁶, haben gezeigt, dass der damit verbundene, erhebliche Mehraufwand keinen wesentlichen Erfolg bringt. Das oben beschriebene Näherungsverfahren scheint im Hinblick auf die erreichbare Genauigkeit der Approximation und die Veränderlichkeit der zu approximierenden Last arbeitsökonomisch am günstigsten zu sein.

Ergebnisse der linearen (nicht-ganzzahligen) Optimierung liessen Rundungen zur Erzielung der Ganzzahligkeit (wegen zu grosser Rundungsfehler bei kleinen Ergebniswerten) nicht zu. Versuche mit ganzzahligen linearen Optimierungsverfahren^{17,18} ergaben keine zulässigen Lösungen im praktisch einsetzbaren Bereich (maximal 15 gleiche Script-Produkte pro polynombildendem Summand).

Die Theorie der diophantischen Gleichungen garantiert zwar ganzzahlige, nicht jedoch positive Lösungen bei Gleichungssystemen mit ganzzahligen, teilerfremden Zeilen-Koeffizienten. Bei unterbestimmten Gleichungssystemen ($n > m$) existieren die Lösungen x_j , $j=1, \dots, n$, mehrdeutig mit $n-m$ Lösungsscharparametern, mit deren Hilfe die Konstruktion von nicht-negativen Lösungen x_j versucht wurde. Bei der Durchführung des Verfahrens entwickelten sich die Lösungen sehr rasch zu grossen Zahlen, die ihren Einsatz zur Definition eines optimalen Pseudo-Benchmarks für den praktischen Einsatz wegen des hohen notwendigen Aufwands nicht zulassen.

15 - VAJDA: Mathematical Programming, <VAJD61>, S.191 ff.

16 - MORDELL: Diophantine Equations, <MORD69>, S. 30 ff.

17 - IBM: Mathematical Programming System Extended (MPSX)/370, MIP/370, Program Reference Manual, <IBMO75.2>.

18 - SPERRY UNIVAC: Functional Mathematical Programming System (FMPS), Kap. 8, MIP Operating Mode.

Beispiel für die Parametrierung eines Script-Polynoms als Pseudo-Benchmark.

Aus Basis-Scripts mit der Parametrierung $n_T=20$, $t_T=10$, $n_I=20$, und $n_C = 10$, sowie $i_I = 1000$, $c_C = 750$ und $Z = 1$ werden allein die in Tabelle 6.5. aufgeführten Script-Produkte erzeugt und ihre Lastwerte gemessen; die Ergebnisse (bzgl. der Kontrollvariablen TRX, SIO, CPU und SRT) sind der Tabelle 6.5. angegeben.

Tabelle 6.5. Polynombildende Script-Produkte und ihre Lastwerte.

Script-Produkt	Anz. d. Basis-Scripts T	Basis-Scripts I	Typ C	CPU- Ausl.	I/O- Rate	TRX- Rate.	Antwort- zeit SRT
SP(1)	1	0	0	22.44	12.36	4.03	.14
SP(2)	0	1	0	15.28	12.21	.76	.29
SP(3)	0	0	1	57.41	13.10	.40	3.05
SP(4)	2	0	0	35.34	14.69	6.71	.20
SP(5)	0	2	0	23.76	27.02	1.29	.56
SP(6)	0	0	2	72.57	25.22	.51	4.80
SP(7)	3	0	0	45.53	33.71	8.79	.23
SP(8)	0	3	0	28.94	38.93	1.59	.82
SP(9)	0	0	3	83.86	29.38	.60	6.38
SP(10)	4	0	0	48.82	47.85	9.35	.31
SP(11)	0	4	0	28.08	42.28	1.48	1.58
SP(12)	0	0	4	87.38	38.73	.63	8.33
SP(13)	5	0	0	52.31	38.23	9.97	.39
SP(14)	0	5	0	31.41	53.23	1.66	2.05
SP(15)	0	0	5	88.82	31.32	.64	10.66
SP(16)	1	1	0	32.99	25.47	4.21	.28
SP(17)	0	1	1	60.88	20.53	1.01	1.88
SP(18)	1	0	1	63.58	28.74	3.63	1.82
SP(19)	2	1	0	42.86	38.76	6.10	.45
SP(20)	0	2	1	60.95	21.78	1.38	1.85
SP(21)	1	0	2	73.54	19.30	2.65	3.83
SP(22)	1	1	1	67.54	38.76	3.69	1.68
SP(23)	4	1	0	56.10	46.61	8.17	.58
SP(24)	0	4	1	56.57	39.14	1.61	2.44
SP(25)	4	0	1	70.68	36.23	8.34	1.47
SP(26)	3	2	0	56.42	39.30	6.52	.81
SP(27)	2	0	3	82.04	47.04	3.60	4.94
SP(28)	1	2	2	78.83	47.52	2.76	3.49

Aus dieser Menge der Script-Produkte (Grad: maximal 5) sollen nach dem oben beschriebenen Verfahren Script-Polynome (s. S. 82) gebildet werden, die bzgl. der Kontrollvariablen TRX, SIO, CPU und SRT eine vorgegebene Last angenähert wiedergeben. Da Script-Polynome aus hintereinander ablaufenden Script-Potenzen gebildet werden, kann allein durch die Mittelwertbildung der Lastparameter der Produkte der Mittelwert der Lastparameter des Polynoms berechnet werden (System-Overhead ist bereits in den gemessenen Lastwerten der Script-Produkte enthalten).

Trotzdem muss das gefundene Script-Polynom wegen möglicher leichter Verschiebungen der Verweilzeiten einzelner Produkte nach der sequentiellen Verkettung erneut gefahren und die berechneten, theoretischen Werte durch Messungen verifiziert werden. Erfahrungen (bei allen durchgeführten Näherungen) zeigen, dass sich die gemessenen Werte des Polynoms gegenüber den aus den Produkten geringfügig nach oben veränderten (vgl. Tab. 6.6. u. 6.7., S. 106/107).

Es werden für die vier Kontrollvariablen ($m=4$) TRX, SIO, CPU und SRT die Lastgleichungen aufgestellt (gemäss S. 99), dazu werden die in Tabelle 6.5. angegebenen Werte für die a_{ij} , $i=1, \dots, m=4$, $j=1, \dots, n=28$, verwendet.

Als Zielwerte werden die Mittelwerte der KV-Daten aus der Leistungsdatenbank vom 19.- 23.2.1979 (IBM/370-168, 4 MB Hauptspeicher, IBM 3330-kompatibler Plattenspeicher, System-Konf. 3 des Anhangs, Kap. 9.1.):

$$\begin{aligned} \text{TRX}_P &= 1.2 \text{ 1/s,} \\ \text{SIO}_P &= 35 \text{ 1/s,} \\ \text{CPU}_P &= 28 \quad \% \quad \text{und} \\ \text{SRT}_P &= 1.4 \text{ s.} \end{aligned}$$

Die Zielwerte sind durch Linearkombination mit ganzzahligen und positiven Koeffizienten der Script-Produkten in Tab. 6.5. nach dem auf den Seiten 96-98 beschriebenen Verfahren anzunähern. Zur Abkürzung des Verfahrens werden alle diejenigen Produkte aus Tab. 6.5. eliminiert, deren TRX-Wert > 2 und deren SRT-Wert > 5 ist. Diese haben aufgrund der Messdaten keine Bedeutung bei der Nachbildung der realen Lasten. Dies schränkt die Anzahl der zulässigen Produkte von 28 auf 10 zur Bildung des Polynoms P ein:

$$P(s_0) = \sum_r x_r \text{SP}(r),$$

mit $r=2,3,5,6,8,11,14,17,20,24$; $\text{SP}(r)$ bezeichnet die Script-Produkte, x_r die zu bestimmenden ganzzahligen, positiven Koeffizienten (vgl. Tab. 6.5., S. 104).

Das Verfahren soll ein Script-Polynom liefern, dessen mittlere relative Abweichung s_j nicht grösser als ein Fehler von

$$s_0 = 5 \%$$

ist. Die Maximal-Anzahl (Abbruchkriterium für das Verfahren) der Produkte k_0 soll 10 nicht übersteigen (zur Beschränkung der Gesamtverweilzeit des Polynoms):

$$k_0 = 10.$$

Aus dieser Nebenbedingung ergibt sich sofort:

$$\sum_r x_r \leq k_0 = 10.$$

Abb. 6.11. zeigt die schrittweise Bestimmung des Polynoms. Der Anfangswert (erstes in das Polynom aufzunehmende Produkt) ergibt sich aus dem Ablaufdiagramm in Abb. 6.10. Das Ergebnis-Polynom $P(s_0=5\%)$ lautet:

$$P(5\%) = SP(2) + SP(5) + 4 SP(11) \quad (\text{mit } k=6)$$

oder
$$P_4(I, T, C) = I + I^2 + 4 I^4$$

in der Nomenklatur von S. 78/79. Dabei ist

$$I = I(n_T=20, t_T=10, n_I=20, i_I=2000, n_C=10, c_C=750, Z=1)$$

Für dieses Polynom sind die berechneten und die gemessenen Abweichungen gegenüber den Zielwerten in Tab. 6.6. angegeben.

Tabelle 6.6. Abweichungen der Kontrollvariablenwerte beim optimalen Script-Polynom, 5%- Fehlergrenze.

	berechnete Abweichg. (nach S. 104)	gemessene Abweichg.	Differenz
TRX	+ 11 %	+ 9 %	- 2 %
SIO	- 1 %	+ 5 %	+ 6 %
CPU	- 10 %	- 8 %	+ 2 %
SRT	- 15 %	- 13 %	+ 2 %

Wegen des noch stark vom Zielwert abweichenden Antwortzeitwert des P(5%) wird das Anpassungsverfahren bis zu einem Fehler von nur 2 % fortgesetzt. Schon nach drei Schritten (k=9) wird $s_0 = 2 \%$ erreicht; damit ist das Polynom

$$P(2\%) = SP(2) + SP(3) + SP(5) + SP(8) + 5 SP(11)$$

oder
$$P_4(I,T,C) = I + C + I^2 + I^3 + 5 I^4$$

mit
$$C = C(n_T=20, t_T=10, n_I=20, i_I=2000, n_C=10, c_C=750, Z=1)$$

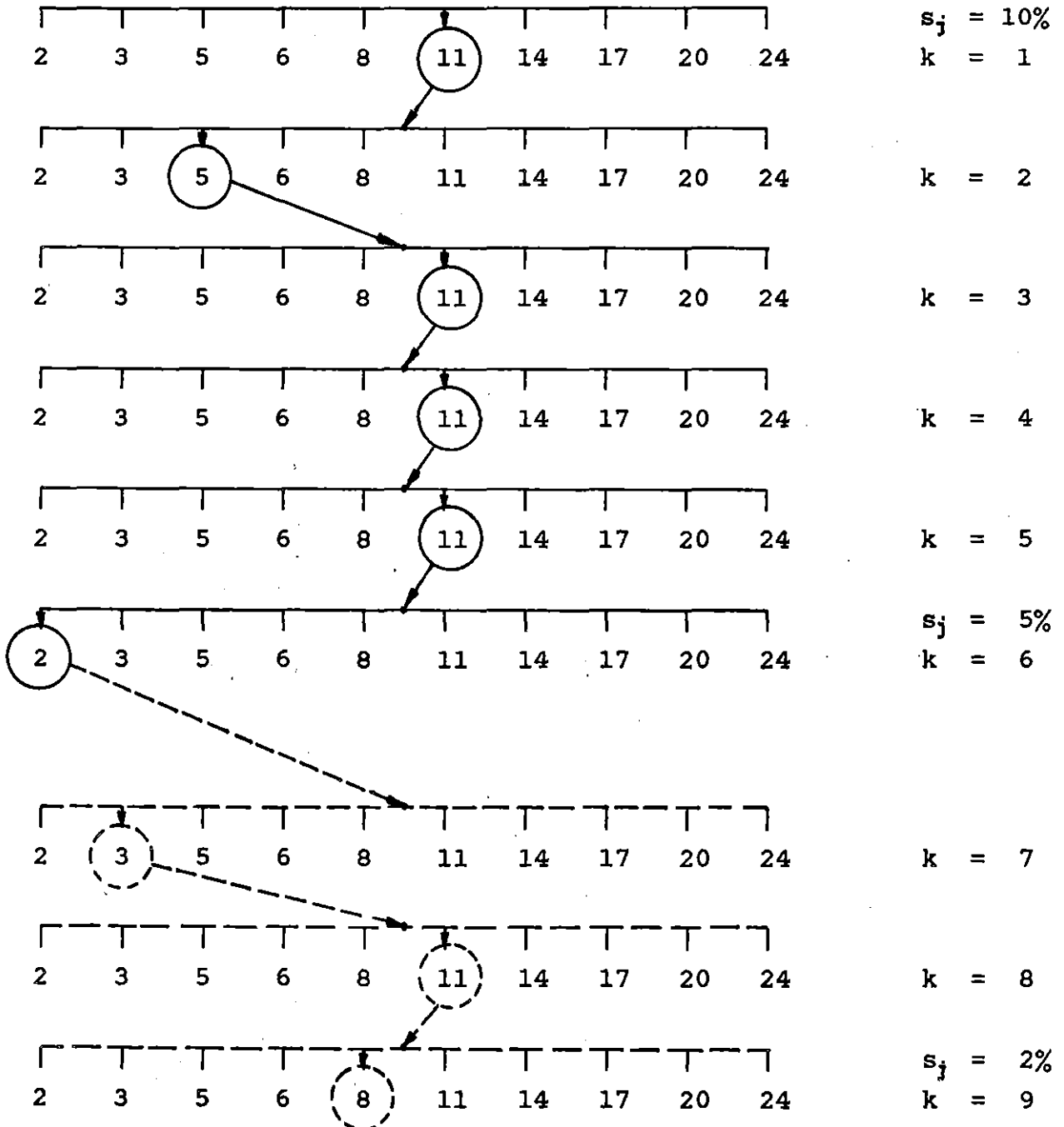
bestimmt. Tabelle 6.7. gibt die Abweichungen von den Zielwerten an. In Abb. 6.11. ist die Weiterführung bis $s_0 = 2 \%$ gestrichelt eingetragen.

<u>Tabelle 6.7. Abweichungen der Kontrollvariablenwerte beim optimalen Script-Polynom, 2%- Fehlergrenze.</u>			
	berechnete Abweichg. (nach S. 104)	gemessene Abweichg.	Differenz
TRX	+ 5 %	+ 2 %	- 3 %
SIO	- 4 %	+ 3 %	+ 7 %
CPU	+ 6 %	+ 7 %	+ 1 %
SRT	0 %	+ 3 %	+ 3%

Es zeigt sich, das beide Polynome P(5%) und P(2%) durch das I/O-orientierte Basis-Script I geprägt werden. Der durch das Script-Polynom $P(2\%) = P_4(T,I,C)$ wird für die Pseudo-Benchmarks-Experimente in Kap. 8.2. eingesetzt. Während der Messungen zur Bestimmung der Abweichungen für P(5%) und P(2%) wurden jeweils die hochgradigen Produkte zuerst gefahren.

Ein weiteres Ergebnis ist, dass bei kleiner Denkzeit Z, alle TRX-treibenden Produkte **n i c h t** Bestandteil des Polynoms werden. Grund dafür ist die wegen kleiner Denkzeiten ohnehin schon hohe Transaktionsrate.

Abbildung 6.11.: Schrittweise Erzeugung des optimalen Script-Polynoms.



7. Leistungsbewertungsverfahren.

7.1. Leistungsbewertungsstrategie.

Zur standardisierten Leistungsbewertung für ein Timesharing-System werden regelmässige Aktionen, die der Leistungskontrolle und dem Leistungsnachweis dienen, und solcher, die aufgrund besonderer, auffälliger Ergebnisse der Leistungskontrolle vorgenommen werden, zusammengefasst.

Die Leistungskontrolle, d.i. die Überwachung der mittl. max. Betriebs-Systemleistung, der relativen Betriebs-Systemleistung, sowie der Servicerate B und der mittleren Antwortzeit SRT, wird regelmässig, d.h. betriebsbegleitend, durch den Einsatz eines Software-Monitors durchgeführt. Die Führung der Leistungsdatenbank (Kap. 2.1., S. 22) geschieht automatisch nach festen Zeitintervallen. Dabei werden die relevanten Kontrollvariablenwerte überwacht. Bei der Überschreitung festgelegter Grenzwerte (control limits¹) setzt die Schwachstellen-Analyse ein.

Werden dabei auffällige Veränderungen beobachtet, z.B. ein Anstieg der Antwortzeit oder ein Abfall der Systemleistung, ohne dass dies nur zufällige Einzelwerte sind, so muss eine Leistungsdiagnose durchgeführt werden. Diese besteht in der Hypothesenfindung für die Ursache dieser Veränderung¹. Hypothesen zur Behebung (Leistungsverbesserung) werden nach vollzogener Veränderung der System-Konfiguration oder -Parameter mit Hilfe der Leistungskontrolle oder durch Fahren von Benchmarks oder Pseudo-Benchmarks verifiziert. In der Betriebspraxis muss darauf geachtet werden, dass der mit der Leistungsdiagnose und -verbesserung verbundene Aufwand durch die angewendeten Verfahren mit den erreichbaren Zielen im Einklang steht.

Bei grossen Veränderungen, die über Leistungsverbesserung durch Umkonfiguration oder Änderung von System-Parametern (Stellgrössen)

¹ - BELL/BOEHM/WATSON: Framework..., <BELL72>, S. 1141-1154.

hinausgehen, werden Quasi-Benchmarks oder der dynamische Timesharing-Benchmark zur Feststellung des Leistungsquotienten Q unter bekannten Lastbedingungen gefahren. Nach solchen Veränderungen sollten auch die Gewinn- und Benutzer-Verlustzeiten (Beispiele in Kap. 8.2.6. zur Gewinnfunktion und in Kap. 8.3. zur Benutzer-Verlustzeit) neu aufgestellt werden.

Diese vollständige Leistungsbewertungsstrategie (in Abb. 7.1., S.111) in der Form eines Ablaufdiagramms schematisch dargestellt) bildet eine Erweiterung der Vorgehensweise von WINDFUHR² für den Timesharing-Bereich.

7.2. Analyse der Betriebsdaten aus der Leistungsdatenbank.

Während zur Ermittlung des Leistungsvermögens und der Leistungskurve (Kap. 4.2.) Benchmarks eingesetzt werden, muss für die im Betrieb erreichte Systemleistung die statistische Verteilung der (gemessenen) Transaktionsraten herangezogen werden. Dies ergibt sich aus der Formel (4.8.), S. 56, wenn dort $T_S = T$ die Messzeit (i.a. einen Betriebszyklus) angibt. Die so ermittelte betriebliche Systemleistung soll mit L_B bezeichnet werden.

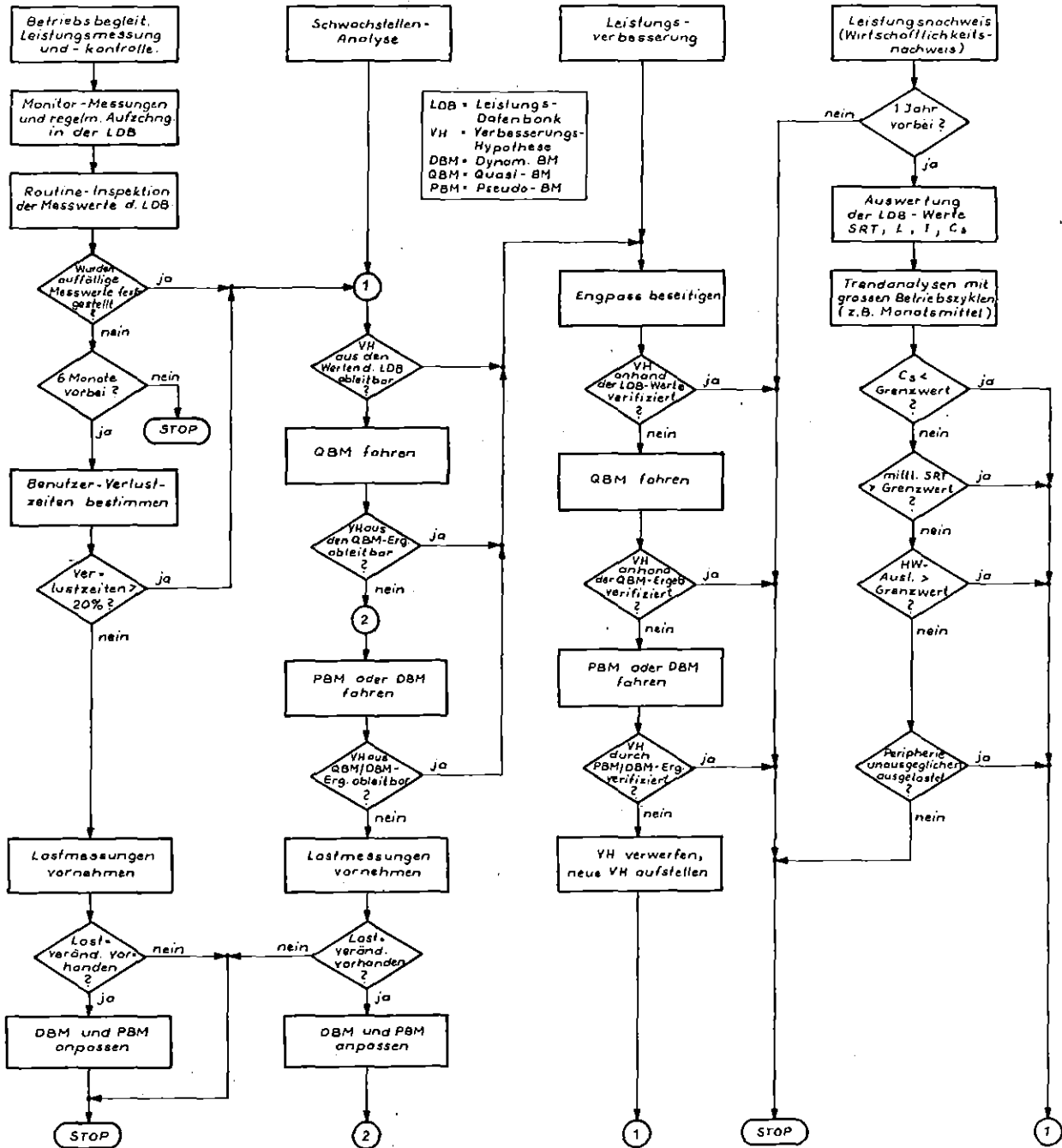
Eine Regressionsanalyse von L_B mit den (ebenfalls gemessenen) u -Werten (u = Anz. d. Terminalsitzungen) zeigt für steigende u -Werte einen zunächst fast linearen Anstieg von L_B , dann aber eine Abflachung, für die eine lineare Regression versagt. Die beobachtete Abflachung ist der Anlass für Versuche mit Regressionsparabeln (2. Grades) der Form

$$L_B(u) = c_1 u^2 + c_2 u,$$

die gute Ergebnisse erbracht haben. Die betriebliche Systemleistungs-

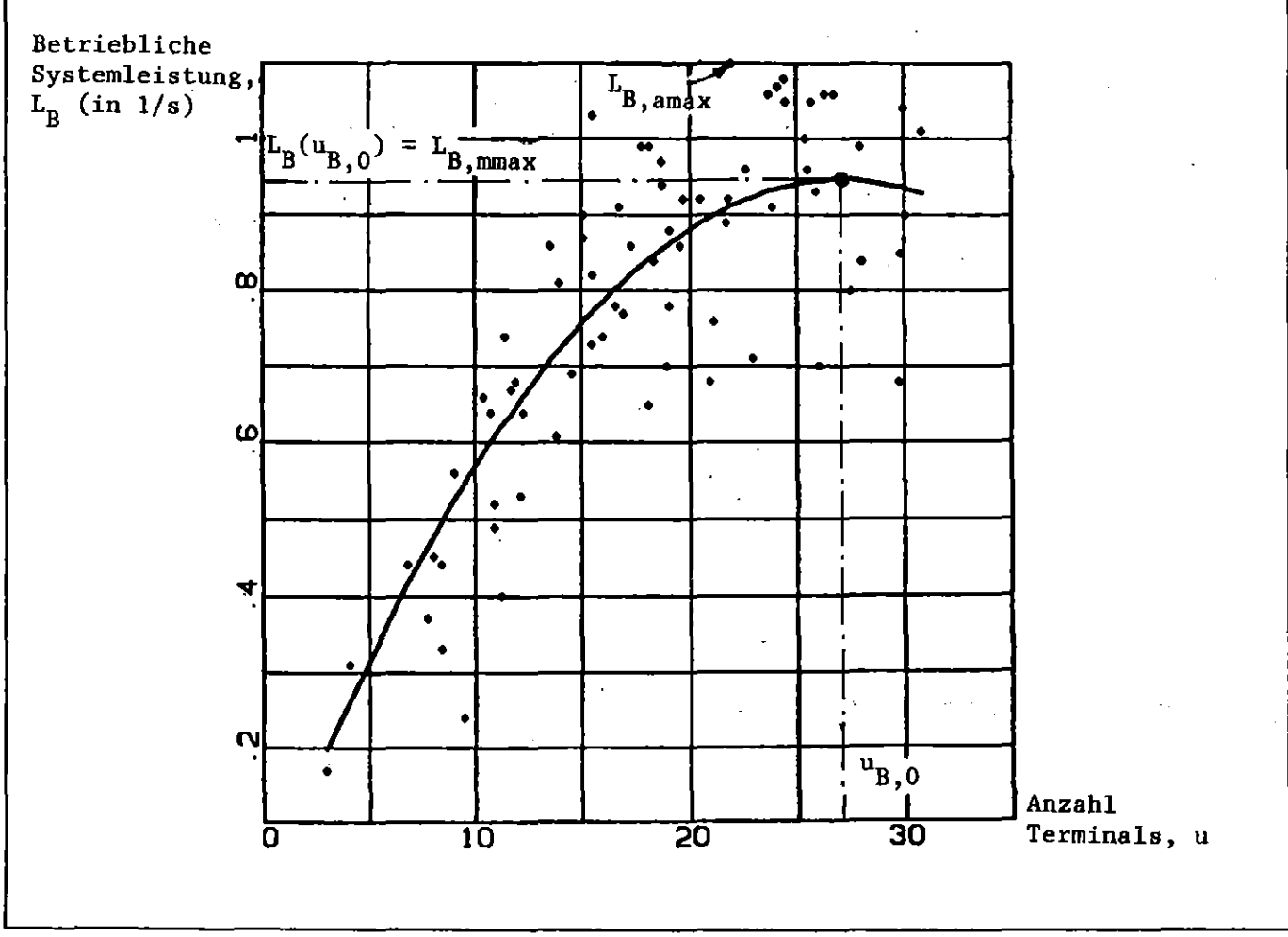
² - WINDFUHR: Methoden und Verfahren..., <WIND77>, S. 137/138.

Abb. 7.1.: Übersicht - Leistungsbewertungsstrategie für TS-Systeme.



kurve $L_B(u)$ erreicht ihren Maximalwert dort ($u = u_{B,0}$), wo sich eine Engpass-Ressource auf die Systemleistung auszuwirken beginnt (s. Abb. 7.2.).

Abbildung 7.2.: Betriebliche Systemleistungskurve.



Da es sich bei der Regression um eine nach der Methode der kleinsten Quadrate (Least Squares Fit) handelt, braucht der Maximalwert $L_B(u_{B,0})$ der betrieblichen Systemleistungskurve nicht identisch sein mit dem im Messzeitintervall T gemessenen Maximalwert der Transaktionsrate. Es soll daher unterschieden werden zwischen:

der maximalen (mittleren) Betriebs-Systemleistung

$$L_{B,mmax} = L_B(u_B, 0)$$

und

der maximalen abgegebenen Betriebs-Systemleistung $L_{B,amax}$, die der im Messzeitraum T maximalen Transaktionsrate entspricht.

In der Regel ist

$$L_{B,amax} > L_{B,mmax}$$

Da $L_{B,amax}$ als Einzelwert durch zufällige Schwankungen der Betriebslast entsteht, kann er für einen Leistungsvergleich verschiedener Systeme **n i c h t** in Betracht gezogen werden.

Definition: Leistungsquotient (Betrieb):

$$Q_B = L_{B,mmax,2} / L_{B,mmax,1}$$

Diese Formel entspricht (4.11.), Kap. 4.; sie gibt das Verhältnis zweier mittl. max. Betriebs-Systemleistungen an. Vergleiche der Leistungsquotienten nach dieser Formel (7.5.) werden in Kap. 8.2.3. gegeben.

Service rate und mittlere Antwortzeit werden, wie in den Formeln (4.14.) und (4.17.) definiert, verwendet, L wird jeweils durch $L_{B,mmax}$ ersetzt.

Weitere Verfahren zur Leistungsanalyse anhand der Betriebsdaten aus der Leistungsdatenbank werden durch die im folgenden Kapitel behandelten multiplen Regressionsmodelle angegeben, wobei ausschliesslich die Servicegrösse System-Antwortzeit die leistungsbeschreibende Rolle einnimmt.

7.3. Regressionsmodelle.

Es werden multiple Regressionsmodelle³ der Form

$$E \{ Y | X_1 = x_1, \dots, X_n = x_n \} = L(a_0, \dots, a_n; x_1, \dots, x_n)$$

erstellt, in welchem die Vorhersagegrösse Y durch die Servicegrösse SRT (System-Antwortzeit) als Funktion von gemessenen Kontrollvariablenwerten x_1, \dots, x_n mit Hilfe der Parameter a_0, \dots, a_n dargestellt wird. Ist die Funktion L linear in den $n+1$ Parametern a_i , so wird die Abhängigkeit von Y von den n Zufallsgrössen X_i als ein lineares Modell bezeichnet. Die Parameter a_0, a_1, \dots, a_n werden nach der Methode der kleinsten Fehlerquadrate (Least-Squares-Fit) bestimmt.

WALDBAUM⁴ und FRIEDMAN/WALDBAUM⁵ haben Modelle 2. Grades (die Zufallsvariablen X_1, \dots, X_n treten höchstens in Potenzen 2. Grades auf) zur Beurteilung von Tuning-Massnahmen an einem APL-System eingesetzt und die Verwendbarkeit dieses Verfahrens gezeigt. WALDBAUM weist besonders auf die Notwendigkeit der Einführung der quadratischen Terme hin; sein Verfahren der schrittweisen Eliminierung⁶ führt schliesslich sogar zum Verschwinden aller linearen Terme.

GOMAA^{5,6} erstellt und verwendet Modelle 1. Grades, um die Verweilzeiten von Batchjobs auf CYBER-Anlagen nachzubilden. MATERNA⁹ gibt ein Verfahren an, nach dem repräsentative Lasten eines Rechnersystems (DEC) nachgebildet werden können.

-
- ³ - SEBER: Linear Regression Analysis, <SEBE77>, S. 6.
⁴ - WALDBAUM: Evaluating Computing System Changes..., <WALD73>, S. 127-135.
⁵ - FRIEDMAN/WALDBAUM: Evaluating Syst. Changes Under an Uncontrolled Workload, <FRIE75>, S. 370-352.
⁶ - DRAPER/SMITH: Applied Regression Analysis, <DRAP66>, S. 171.
⁷ - GOMAA: Regression Models for the Evaluation..., <GOMA76>, S. 69-99.
⁸ - GOMAA: A Modelling Approach..., <GOMA77>, S. 171-199.
⁹ - MATERNA: Ein allgemeines Verfahren zur Konstruktion..., <MATE78>.

In dieser Arbeit werden multiple Regressionsmodelle verwendet, um die Servicegrösse System-Antwortzeit (SRT) in ihrer Abhängigkeit von mehreren anderen Kontrollvariablen statistisch darzustellen. Die unabhängigen Kontrollvariablen (Regressoren) werden mit dem Verfahren der schrittweisen Regression (forward selection procedure¹⁰) durch Abschätzung der Anteile (R_i^2 in Tabelle 7.1.) der verschiedenen Regressoren an der Approximation der Zielgrösse SRT ausgewählt. Die Modelle dienen zur kurzfristigen Vorhersage der System-Antwortzeit aus den Werten der Regressor-Variablen und zum Vergleich der Einflüsse der Kontrollvariablen auf die System-Antwortzeit bei unterschiedlichen Systemen.

Die Tabelle 7.1. zeigt, dass die Kontrollvariable TS-CCL (TS-Concurrency Level) den grössten Einfluss auf die Regression und damit auf die mit dem Modell (Modell I) geschätzten Werte der Regressionsvariablen SRT (Antwortzeit) besitzt. Die schrittweise Regression erfolgte über alle primären Kontrollvariablen (vgl. Kap. 2.1., S. 20/21) und die Paging-Rate. Das Modell ist linear in allen Regressoren, sein Bestimmtheitsmass beträgt $R^2 = 94 \%$, wovon allein 91 % auf die Kontrollvariable TS-CCL entfallen.

Eine Alternative zu dem in Tabelle 7.1. angegebenen Modell erhält man, wenn man von den in Kap. 3. mit cluster-analytischen Methoden bestimmten, der Trennung verschiedener Betriebsmodi dienenden, Kontrollvariablen TS-CCL (TS-Concurrency-Level), CPU (Gesamt-CPU-Auslastung) und SIO (Gesamt-I/O-Rate) ausgeht (Modell II). Dieses Modell schränkt den Aufwand für die schrittweise Regression ein (Tabelle 7.2.).

Die Resultate beider Modelle legen es nahe, auf die einfache Regression zwischen SRT und TS-CCL, bzw. deren Korrelation (Kap. 2.1., S. 27) zurückzugreifen. Gerade jedoch die dort mit anderen Messdaten

¹⁰ - DRAPER/SMITH: Applied Regression Analysis, <DRAP66>, Kap. 6.

TABELLE 7.1.: Modell I. *)

$$\text{Modell : } Y = a_0 + \sum a_i X_i$$

Var.-Name	symb. Name	Regr. Koeff.	Fehler-Abw.	t	R ² (X _i)
SRT	Y	a ₀ = 1.53	-	-	-
CPU	X ₁	a ₁ = 0.03	0.04	0.75	<0.5%
TS-CPU	X ₂	a ₂ = 0.01	0.05	0.10	<0.5%
SIO	X ₃	a ₃ = -3.43	4.92	-0.70	<0.5%
TS-IO	X ₄	a ₄ = 3.47	4.92	0.71	<0.5%
TRX	X ₅	a ₅ = -3.06	1.98	-1.54	3 %
TS-CCL	X ₆	a ₆ = 1.30	0.17	7.49	91 %
Paging-Rate	X ₇	a ₇ = -0.08	0.06	-1.36	<0.5%

Bestimmtheitsmass der Regression: R² = 94 %,

$$F_{\text{REG}} = 67.2 > F(7, 28; 95\%) = 2.36$$

TABELLE 7.2.: Modell II. *)

$$\text{Modell : } Y = a_0 + \sum a_i X_i$$

Var.-Name	symb. Name	Regr. Koeff.	Fehler-Abw.	t	R ² (X _i)
SRT	Y	a ₀ = 0.60	-	-	-
CPU	X ₁	a ₁ = 0.03	0.03	0.90	<0.5%
TS-CCL	X ₂	a ₂ = 1.11	0.13	8.28	91 %
SIO	X ₃	a ₃ = -0.06	0.03	1.71	1 %

Bestimmtheitsmass der Regression: R² = 92 %,

$$F_{\text{REG}} = 118.6 > F(3, 32; 95\%) = 2.90$$

*) beide Modelle (I und II) wurden unter Verwendung der Messdaten vom 16.8.1978, 8 - 17 Uhr (aus der Leistungsdatenbank) erzeugt; Systemkonfiguration 1 (lt. Anhang, Kap. 9.2., S. 163)

gefundene geringere Korrelation lässt Vorsicht bei einer zu raschen Eliminierung weiterer Einflussgrößen geboten sein (vgl. auch die Ergebnisse von Modell IV, Tabelle 7.5., S. 121). Die Modelle I und II lassen den geringen Einfluss aller Kontrollvariablen - ausgenommen TS-CCL und - in Modell I noch TRX, die Transaktionsrate - erkennen.

Die Prognosefähigkeit des Modells II soll anhand der Daten eines anderen Betriebszyklus abgeschätzt werden, der allerdings noch hinreichend dicht bei dem modellerzeugenden Zyklus liegt (um starke Betriebslastveränderungen oder Konfigurationsänderungen zu eliminieren):

Modell II (Erstellung und Validation):

$$SRT_j(t_i) = a_0 + a_1 CPU_j(t_i) + a_2 TS-CCL_j(t_i) + a_3 SIO_j(t_i)$$

der

$$SRT_{j,i} = a_0 + a_1 CPU_{j,i} + a_2 TS-CCL_{j,i} + a_3 SIO_{j,i}$$

mit $i=1, \dots, 36$ und $j=1, 2$. Die Messdaten stammen aus der Leistungsdatenbank und zwar für $j=1$, zur Erstellung des Modells, vom 16.8.1978, und für $j=2$, zur Validation des Modells, vom 18.8.1978. Beiden Messdatensätzen liegt die Systemkonfiguration 1 (s. S. 163) zugrunde.

Die Tabelle 7.3. (S. 118) gibt die Ergebnisse der Residuen-Analyse beider Messreihen für das Modell II an. Die Abb. 7.3. zeigt einen Vergleich der Antwortzeiten des Modells (Vorhersagewerte, $j=2$, gestrichelt) und die im Betrieb gemessenen Werte (Betriebsdaten vom 18.8.78, durchgezogene Linie). Aus Abb. 7.3. ist geht hervor, dass die Qualität der Vorhersage gut ist. Zum Vergleich sind sind die Werte der modellbildenden ($j=1$) Regressor-Variablen in Abb. 7.4. (S. 119) zusammen mit den Kontrollvariablenwerten für den Validationszyklus ($j=2$) angegeben.

TABELLE 7.3.: Residuen-Analyse für Modell II.

A. Residuen-Analyse der modellbildenden Messdaten (j=1, s.S. 117):

Fehlersumme : $3.86 \cdot 10^{-14}$,
 mittl. Fehlerquadrat : 0.76,
 max. Fehler : 2.42,
 (Messdaten vom 16.08.78, 8.00-17.00 Uhr).

B. Residuen-Analyse der Validations-Messdaten

Fehlersumme : -12.18,
 mittl. Fehlerquadrat : 0.94,
 max. Fehler : 3.58.
 (Messdaten vom 18.08.78, alle 15-min. Zeitintervalle, 8 - 17 Uhr).

Abbildung 7.3.: Vergleich der Antwortzeiten für Modell II (j=2) mit den gemessenen Betriebsdaten (s. S. 117).

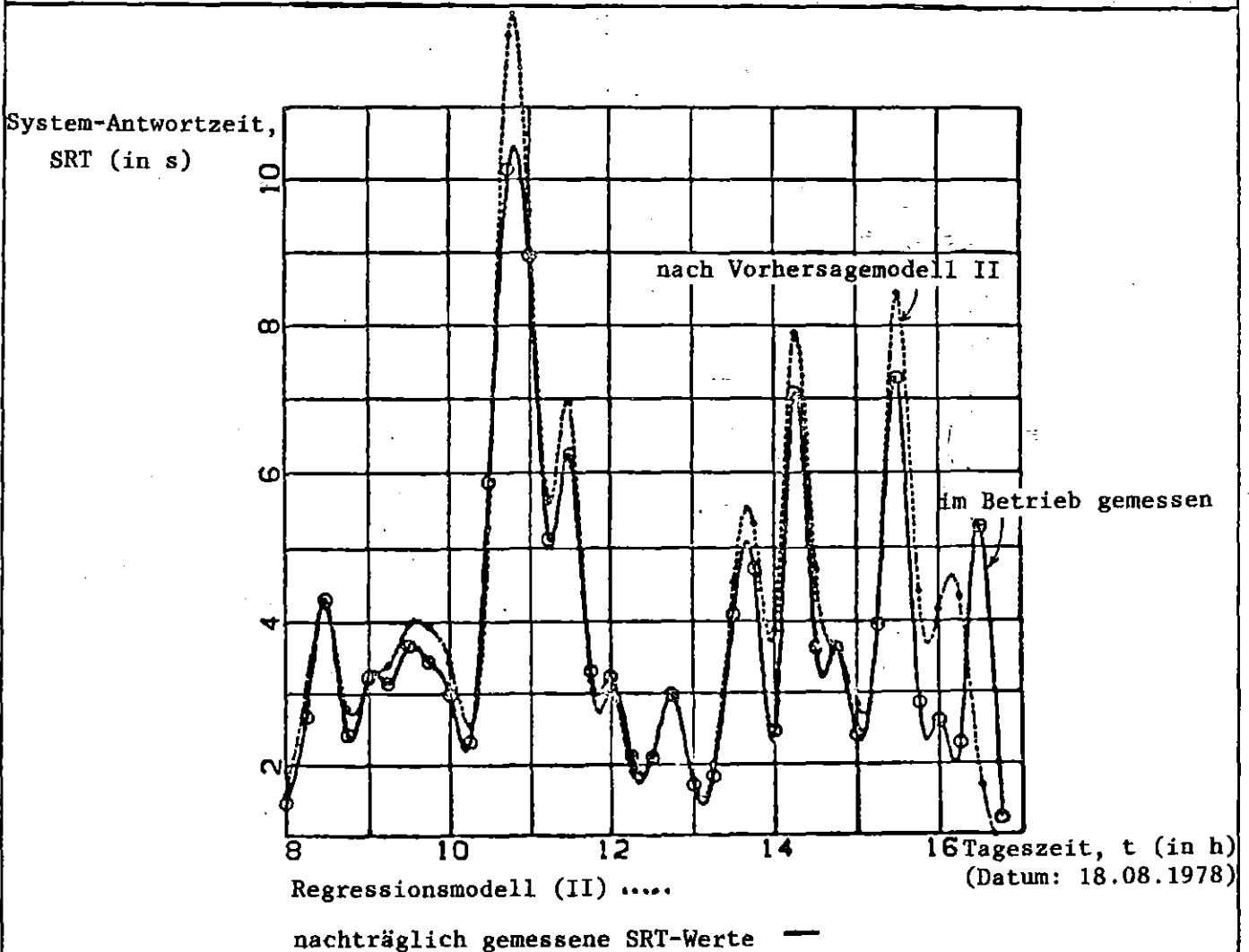
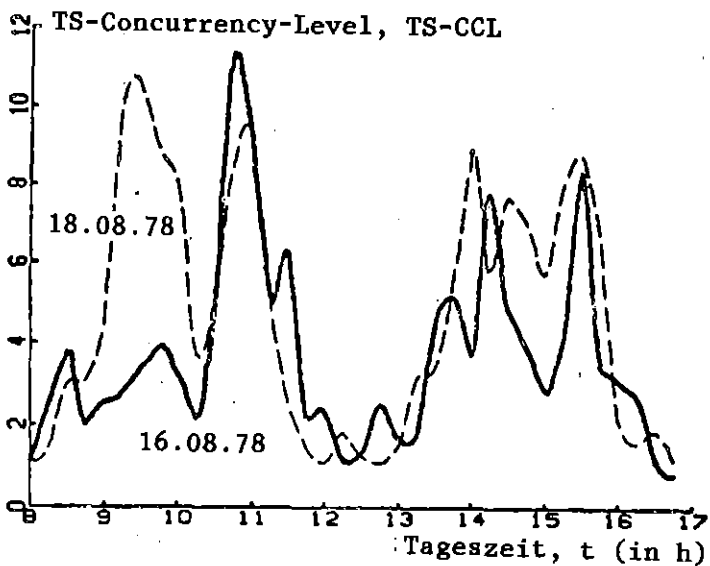
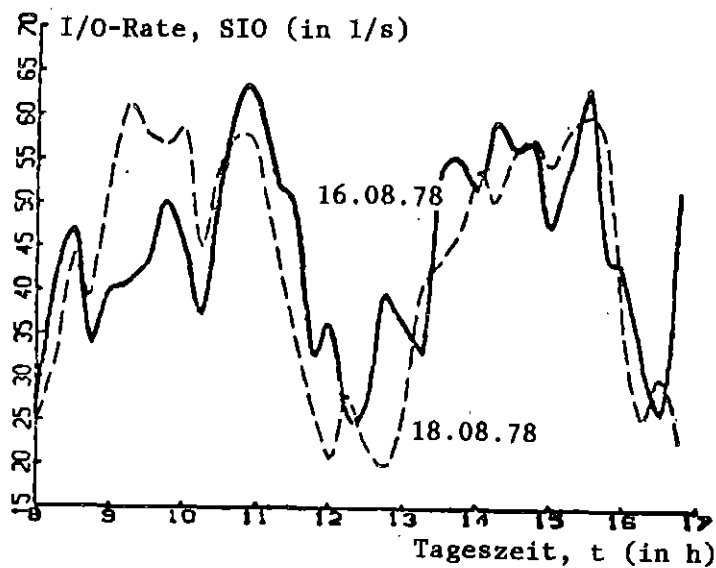
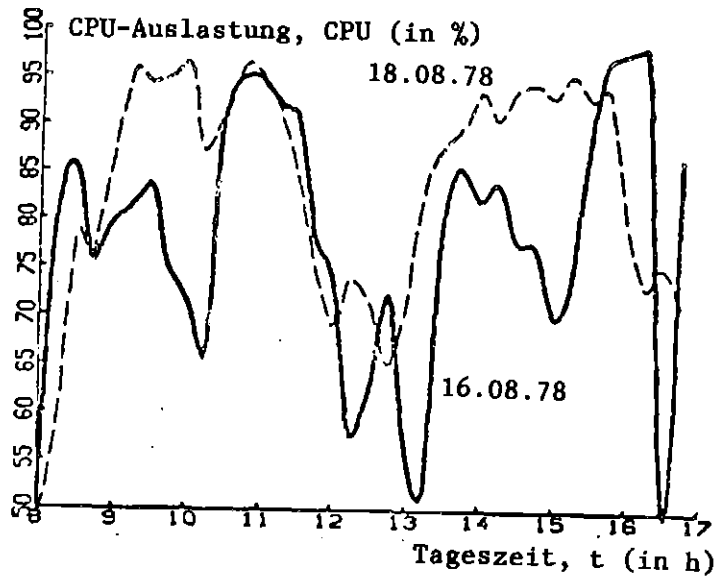


Abbildung 7.4.: Kontrollvariablenwerte für das Regressionsmodell II.



Modelle höheren Grades bringen keine wesentlichen Verbesserungen; das mit über 90% sehr hohe Bestimmtheitsmass kann mit einem Ansatz 2. Grades von 92 auf 94% erhöht werden, die F-Statistik für die Güte der Regression sinkt jedoch ab (s. Modell III, Tabelle 7.4.).

TABELLE 7.4.: Modell III.

$$\text{Modell : } Y = a_0 + \sum a_i X_i + \sum a_i X_i^2 + \sum \sum a_{ij} X_i X_j$$

Var.-Name	symb. Name	Regr. Koeff.	Fehler-Abw.	t	R ² (X _i)
SRT	Y	a ₀ = 11.95	-	-	-
CPU	X ₁	a ₁ = -0.35	0.40	-0.87	<0.5%
TS-CCL	X ₂	a ₂ = 5.35	3.18	1.68	91 %
SIO	X ₃	a ₃ = -0.29	0.53	-0.55	1 %
-	X ₁ ²	a ₄ = 0.00	0.01	0.46	<0.5%
-	X ₂ ²	a ₅ = 0.26	0.09	2.76	1 %
-	X ₃ ²	a ₆ = 0.00	0.01	-0.05	1 %
-	X ₁ X ₂	a ₇ = -0.06	0.04	-1.35	<0.5%
-	X ₂ X ₃	a ₈ = -0.04	0.05	-0.90	<0.5%
-	X ₃ X ₁	a ₉ = 0.01	0.01	0.41	<0.5%

Bestimmtheitsmass der Regression: R² = 95 %,

$$F_{\text{REG}} = 58.0 > F(9, 26; 95\%) = 2.27$$

Messdatenwerte wie bei Modell I und II.

Gültigkeitsbereich der Prädiktormodelle: Regressionsmodelle sollen nur innerhalb der Definitionsbereiche ihrer unabhängigen Variablen eingesetzt werden. Damit ist ihr Einsatz auf diesen Bereich eingeschränkt. Für Zeitintervalle mit geringen Abweichungen der Regressor-Kontrollvariablen von den modellbildenden Messdaten kann das gleiche Modell zur Vorhersage der Antwortzeit SRT verwendet werden.

Besteht jedoch der Vermutung, dass mehr als 10% der Messwerte aus dem für die Modellbildung zugrunde liegenden 4-Sigma-Bereichen herausfallen, so muss dies als Veränderung der Grundgesamtheiten der Regressorvariablen angesehen und damit auch als eine für das Modell wesentliche Systemveränderung angesehen werden (Toleranzgrenze von 94% bei unbekannter Verteilung¹¹). Eine Korrektur des bisherigen Modells ist notwendig. So sinkt der Einfluss des TS-CCL auf die Güte der Regression (im Vergleich der Modelle II, S. 116, und IV, s. unten), während der Einfluss der CPU (-Auslastung) um ca. 7% steigt; die Beeinflussung der Antwortzeit SRT durch den TS-CCL fällt auf unter 1/3 im Modell IV.

Regressionsmodelle stellen nur die beobachteten Werte der Kontrollvariablen in einen quantitativen Zusammenhang; sie lassen eine Veränderung erkennen, nicht aber ihre Ursache (vgl. Mod. II mit Mod. IV, gleiche KV, verschiedene Syst.-Konfigurationen).

TABELLE 7.5.: Modell IV.

$$\text{Modell : } Y = a_0 + \sum a_i X_i$$

Var.-Name	symb. Name	Regr. Koeff.	Fehler-Abw.	t	R ² (X _i)
SRT	Y	a ₀ = 0.17	-	-	-
CPU	X ₁	a ₁ = 0.02	0.01	2.03	7 %
TS-CCL	X ₂	a ₂ = 0.36	0.18	2.37	63 %
SIO	X ₃	a ₃ = 0.00	0.01	0.29	<0.5%

Bestimmtheitsmass der Regression: R² = 70 %,

$$F_{\text{REG}} = 24.42 > F(3,32;95\%) = 2.90$$

Messdaten aus der LDB vom 06.03.79, 8.00 - 17.00 Uhr, Systemkonfiguration 4; d.h. zwischen den den Modellen I und II zugrunde liegenden Messungen haben Konfigurationswechsel stattgefunden.

¹¹ - SACHS: Angewandte Statistik, <SACH74>, S. 219.

8. Beispiele für den Einsatz der Verfahren in der betrieblichen Praxis.

8.1. Übersicht über die Auswahl der Methoden.

Erstes Ziel von Leistungsbeurteilungen für TS-Systeme ist in der Praxis die laufende Kontrolle des Antwortzeitverhaltens. Dazu werden am System betriebsbegleitend Messungen durchgeführt. Den Zielsetzungen des Rechenzentrums gemäss werden Kontrollvariable festgelegt, deren Werte laufend und regelmässig gemessen werden (Kontrollvariablen, s. Kap. 2.1., S. 20/21) und die zur Analyse und für Vorhersagemodelle (Kap. 7.2., S. 112 und Kap. 7.3., S. 114 ff.) in der Leistungsdatenbank (Anhang, Kap. 9.2., S. 165-167) gesammelt werden.

Die Betriebsdaten der LDB zu analysieren und zu bewerten ist das für die Betriebspraxis am wenigsten aufwendige Verfahren. Dieses Verfahren ist nicht zuletzt deshalb so vorteilhaft, weil Analysen ohne Betriebsunterbrechungen und nachträglich durchzuführen sind. Wie die Ergebnisse von 8.2.1. und 8.2.3. zeigen, sind die Leistungsvergleiche anhand der LDB-Daten mit den aufgrund aufwendigerer Benchmark-Experimente gewonnener Leistungsdaten vergleichbar.

Der Einsatz des TS-Benchmarks oder des Pseudo-Benchmarks ist jedoch unumgänglich, wenn Veränderungen der Last beobachtet werden und deren Auswirkungen auf die Auslastungen beurteilt werden sollen. Hierzu muss die Last und ihre Veränderung genau beschrieben werden. Die Nachbildung der realen Last durch den dynamischen oder Pseudo-Benchmark kann immer nur angenähert geschehen. Für das Verfahren zur Approximation der Lasten müssen wegen notwendiger Anpassungsexperimente (Kalibrierung des BM) mehrfach Betriebsunterbrechungen in Kauf genommen werden. Auch die Experimente, die die Beurteilung der Veränderung ermöglichen, müssen mit dem System gefahren werden und verursachen längere Unterbrechungen des Betriebs.

Ohne Betriebsunterbrechungen können Quasi-Benchmarks (Script-Polynome 1. Grades) gefahren werden (Kap. 6.4., S. 81-83), da sie den natürlichen Betrieb als Last-Untergrund verwenden. In diesem Kapitel werden

Beispiele für die folgenden Aufgaben der Leistungsanalyse und -beurteilung gegeben:

Ermittlung des Leistungsvermögens eines TS-Systems unter Einsatz des dynamischen Timesharing-Benchmarks (in 8.2.1., S. 126),

Leistungsvergleich verschiedener Systeme unter Einsatz des dynamischen TS-Benchmarks (in 8.2.2., S. 130),

Leistungsvergleich verschiedener Systeme anhand der Kontrollvariablenwerte der Leistungsdatenbank (in 8.2.3., S. 135),

Leistungsvergleich bei unterschiedlicher Plattenkonfiguration unter Einsatz des Pseudo-Benchmarks (in 8.2.4., S. 141),

Bestimmung eines optimalen Systemparameters, der Auswahl eines Standard-Blockungsfaktors, unter Einsatz eines Pseudo-Benchmarks (in 8.2.5., S. 144),

Beurteilung einer Peripherie-änderung anhand eines Quasi-Benchmarks (in 8.2.6., S. 146) und

Bewertung von Veränderungen von Parametern in der Betriebssystem-Software (in 8.2.7., S. 152).

Bei der Leistungsbeurteilung werden unterschiedliche Ziele verfolgt: die Beschreibung des Leistungsvermögens eines TS-Systems in (8.2.1.) dient der Verifizierung des in Kap. 6.5. definierten dynamischen Benchmarks und ist als Vorarbeit anzusehen für den Leistungsvergleich in (8.2.2.) zweier verschiedener TS-Systeme.

Leistungsvergleiche dienen der Überprüfung von Mehraufwendungen an Investitionsmitteln bei Systemveränderungen. Sie ermöglichen der RZ-Betriebsführung klare Aussagen über die Erweiterung der Leistungsfähigkeit des Systems gegenüber der Unternehmensleitung und den Benutzern des Rechenzentrums. Eine Kontrolle des Leistungsverhaltens (System-

leistungskurve) ermöglicht eine Beurteilung der Leistungsbeschreibung des Herstellers.

In der folgenden Tabelle wird für alle Beispiele eine Übersicht gegeben über die hauptsächlich verfolgten Ziele, über die wichtigsten dort verwendeten Leistungsgrößen und über die eingesetzten Verfahren:

Ziel	Leistungsgröße	Verfahren	Abschnitt
Best. des Leistungsverhaltens	Systemleistungskurve $L(u)$, Leistungsvermögen L_{\max}	DBM	8.2.1.
Leistungsvergleich (Gesamtsysteme)	Systemleistungskurven, Leistungsquotient Q	DBM	8.2.2.
Leistungsvergleich (Gesamtsysteme)	betr. Systemleistungen, betr. Leistungsquotient	Betr.-daten-analyse (LDB)	8.2.3.
Leistungsvergleich (Teil-Systeme)	Verweilzeiten T_v , mittl. Antwortzeiten	PBM	8.2.4.
Leistungsvergleich (Teil-Systeme)	mittl. Antwortzeiten, Produktivitätsgewinn	QBM	8.2.6.
Auswahl opt. Syst. Parameter (Hardware)	mittl. Antwortzeit SRT	PBM	8.2.5.
Auswahl opt. Syst. Parameter (Software)	Systemleistung $L(u)$, Leistungsvermögen L_{\max}	DBM	8.2.7.

Die Frage nach der Erweiterbarkeit des Terminalnetzes, die sich dem RZ-Management permanent stellt, und der sich daraus ergebenden Frage nach der oberen Grenze für die gleichzeitig betreibbaren Terminals u_{\max} bei einer vorgegebenen oberen Grenze für die mittlere Antwortzeit SRT_{\max} (oder einer unteren Grenze

für die mittlere Service-Rate B_{\min}) lässt sich anhand der Formeln (4.15.) bis (4.19.), Kap. 4.5., mit Hilfe der betrieblichen Systemleistung (Kap. 7.2.) L_B näherungsweise bestimmen (Vgl. auch die Kap. 8.2.1. und 8.2.3.). Aus den Ergebnissen der Benchmark-Experimente lassen sich die Grenzwerte für die durch den Benchmark definierten Lasten ermitteln (bei Last-Veränderung).

Die rechtzeitige Planung von Kapazitätserweiterungen an Rechnerkern und Peripherie wird durch die Überwachung der relativen Systemleistung l und der Kennzahl C_S der Leistungsreserve (s. Def. auf S. 57/58, Kap. 4.3.) die aus den LDB-Daten bestimmt werden, gewährleistet.

Der Vergleich der Leistungen von Subsystemen, z.B. Plattenkonfigurationen erlaubt dem RZ-Management die Beurteilung der Leistung dieser Subsysteme, die zur Beurteilung herangezogenen Verfahren dürfen jedoch nicht so aufwendig sein, wie bei einer Gesamt-System-Beurteilung.

Im Kap. 8.3. wird eine im Ansatz bereits praktizierte Methode beschrieben, die subjektiven Erfahrungen der Benutzer und deren Einfluss auf den Fortgang der direkt vom TS-System abhängigen Arbeiten in den Anwendungsabteilungen zu erfassen. Die durch die Benutzer-Verlustzeiten entstandenen Kosten können von der Unternehmensleitung als Vergleichskriterien für Entscheidungen über den Miteinsatz im Rechenzentrum herangezogen werden.

Anmerkung: Für alle in den nachfolgenden Beispielen angeführten Systeme sind im Anhang, Kap. 9.1., S. 163-164, Hardware und Betriebssystem-Software beschrieben. Zur Erfassung der Messdaten wurden die beiden Software-Monitore (s. Kap. 2.1., S. 28-31) 'General Trace Facility, GTF' und 'Resource Measurement Facility, RMF' eingesetzt.

8.2. Beurteilung von System-Veränderungen.

8.2.1. Ermittlung des Leistungsvermögen eines Timesharing-Systems mit Hilfe des dynamischen Benchmarks.

Aufgabe: Aufbau eines Timesharing-Benchmarks zur Nachbildung der durch die Daten eines Betriebszyklus (hier: Monat August 1978) beschriebenen Timesharing-Last. Vergleich der Systemleistung $L(u)$ des dynamischen TS-Benchmarks mit der betriebl. Systemleistung $L_B(u)$ des Betriebszyklus, Vergleich der Antwortzeiten $SRT(u)$ und $SRT_B(u)$, sowie der CPU-Auslastungen und der I/O-Raten.

Beschreibung der experimentellen Umgebung:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 1 (s. Anhang, Kap. 9.1., S. 163).

Bemerkung: Es werden hier nur Betriebszustände ohne Batchbetrieb (vgl. Kap. 3.) untersucht. Zur Nachbildung eines TS/Batch-Mischbetriebs wird empfohlen, neben den Ergebnissen dieser Arbeit zusätzlich die Verfahren von WINDFUHR¹ einzusetzen. Der Einfluss des APL-Systems auf das Verhalten des zu beurteilenden TSO-Systems wird berücksichtigt. Aufgrund einer Stellgröße der System-Software wird gewährleistet, dass der Anteil der APL-Benutzer am CPU-Zeit-Verbrauch nach oben beschränkt (durch das APL-CPU-Quantum SAPL) bleibt. Für den dynamischen Benchmark wurde von einem Zielwert (Mittelwert aus der LDB) von 13% ausgegangen. Dies führte zu einer Festsetzung von SAPL = 10 (%) für den Betrieb des dynamischen TS-Benchmarks. Eine gesonderte Untersuchung des Einflusses dieses Systemparameters (SAPL) wird in Kap. 8.2.7., S. 152) durchgeführt. Die durch das APL-System initiierte I/O-Rate wird durch die Daten der LDB festgelegt, sie liegt unter 0.5 pro Sekunde und wird deshalb vernachlässigt werden. Der zusätzliche APL-Betrieb, der dem Normalbetrieb entspricht, wird durch eine Dialoganwendung von einem APL-Terminal aus realisiert.

¹ - WINDFUHR: Methoden und Verfahren ..., <WIND77>.

Durchführung: Der Aufbau des dynamischen Benchmarks erfolgt gemäss den in Kap. 6.5. (S. 84 ff.) angegebenen Verfahren; es werden danach aus den drei Basis-Scripts gebildete Folgen von homogenen Script-Produkten der Form

$$S_{3j}(T, I, C) = T^j(n_T, t_T, Z) I^j(n_I, i_I, Z) C^j(n_C, c_C, Z)$$

mit $j=1,2,3,\dots$ eingesetzt, deren Parameter n_T, \dots sowohl an die Funktionsklassenverteilung (S. 84-86) als auch an die Lastwerte (S. 94-98) angepasst worden sind. Es ergibt dies die Werte

$$n_T = 20, t_T = 10, n_I = 20, n_C = 10,$$

$$Z = 13, i_I = 8000 \text{ und } c_C = 750 \text{ (s. S. 96),}$$

die für alle eingesetzten Script-Produkte S_{3j} konstant gehalten werden.

Der DBM wird mit Produkten des Grades $3j = 3, 6, 15, 18$ und 21 gefahren; Zwischenwerte, z.B. $3j = 9, 12$, werden zur Aufwandreduktion eingespart. Der Grad der Script-Produkte definiert die Anzahl der aktiven Terminals $u = 3j$. Es wird also verwendet die Produkt-Folge:

$$\begin{aligned} S_3 &= T I C, \\ S_6 &= (T I C)^2, \\ S_{15} &= (T I C)^5, \\ S_{18} &= (T I C)^6, \\ S_{21} &= (T I C)^7, \end{aligned}$$

Da mehr als 21 Terminals für den Benchmark nach dem Verfahren der automatisierten Script-Eingabe (Kap. 6.1., S. 74) im RZ nicht zur Verfügung standen, konnten keine Produkte höheren Grades als 21 gefahren werden.

Ergebnis: Für die Werte $u = 3, 6, 15, 18, 21$ wurden die Summenverweilzeiten T_V (Kap. 4.2., S. 55) gemessen und daraus nach Formel (4.9.), S. 56, die Systemleistungswerte $L(u)$ berechnet (Tab. 8.1. und 8.2.). Die Tab. 8.1. enthält zusätzlich noch die mittleren Verweilzeiten $T_S(u)$; mit ihnen kann die mittl. Laufzeit der Produkte untereinander verglichen werden. So steigt der zeitliche Aufwand für den Einsatz des Produkts 21. Grades S_{21} gegenüber der Laufzeit von S_3 , dem Minimal-Produkt (S. 87) nur um rund 50 % an.

TABELLE 8.1.: Verweilzeiten des Timesharing-Benchmarks.

u (Pr.-Grad)	$T_V(u)$ (s)	$T_S(u)$ (min)
3	12028	67 (gerundet)
6	25248	70
15	76256	85
18	98710	91
21	121902	97

TABELLE 8.2.: Vergleich der Systemleistungswerte des TS-Benchmarks mit den Betriebsdaten.

u (Pr.-Grad)	$L(u)$ (1/s)	$L_B(u)$ (1/s)	$dL(u)$ $= L - L_B$	Abweichung
3	0.212	--	--	-
6	0.404	0.39	+ 0.014	+ 4 %
15	0.836	0.82	+ 0.016	+ 2 %
18	0.930	0.92	+ 0.010	+ 1 %
21	1.025	0.99	+ 0.035	+ 3 %

Tab. 8.2. vergleicht die mit Hilfe des DBM gemessenen Systemleistungswerte mit den betriebl. Systemleistungen $L_B(u)$ (S. 110), die durch eine Regression 2. Grades aus der TRX-Rate und der Anz. der Sitzungen u des zugrunde gelegten Betriebszyklus bestimmt werden. Die Abweichungen überschreiten nicht die 4 %-Grenze.

In Tab. 8.3. sind die für die verschiedenen Script-Produkte gemessenen TS-Concurrency-Levels (TS-CCL) angegeben, die daraus und nach Tab. 8.2. gem. Formel (4.16.) $SRT = TS-CCL / L$ und $B = 1 / SRT$ berechneten mittleren System-Antwortzeiten SRT und Serviceraten B, sowie die Mittelwerte der gemessenen Antwortzeit SRT_{gem} und die aus den Betriebsdaten ermittelte mittl. Antwortzeit SRT_B (durch Regression auf die u -Werte abgebildet) angegeben.

TABELLE 8.3.: Vergleich der Antwortzeiten des TS-Benchmarks mit den Betriebsdaten.

u	TS-CCL	B (1/s)	SRT (s)	SRT_{gem} (s)	SRT_B (s)	Abweich.
3	0.21	1.00	1.00	0.91	0.7	30 %
6	0.51	0.78	1.28	1.25	1.0	22 %
15	2.37	0.35	2.82	2.77	2.4	15 %
18	3.90	0.24	4.19	4.15	4.0	5 %
21	5.02	0.20	4.92	4.90	5.2	6 %

Die hier festzustellenden stärkeren Abweichungen haben ihre Ursache in den grösseren Schwankungen der Betriebsdaten-SRT, deren Grund wiederum in der im Normalbetrieb sehr ungleichmässigen APL-Benutzung liegen.

Zum Vergleich der CPU-Auslastung und der I/O-Raten von Benchmark und Normalbetrieb sei auf Kap. 8.2.7., S. 152 ff. verwiesen.

Bestimmung der maximalen Systemleistung durch den DBM: Aus den o.a. installationsbedingten Gründen (S. 129/130) konnte die maximale Systemleistung L_{\max} nicht durch 'Hochsetzen' von u erreicht werden; es muss daher aus den vorliegenden Daten durch den Verlauf der Systemleistungskurve bestimmt werden. Dazu wird in Analogie zur betriebl. Systemleistung (S. 110 u. 112) eine Approximationsparabel (2. Grades) verwendet, deren Maximalwert das Leistungsvermögen des Systems festlegt. Es ergibt sich für den dynamischen Benchmark:

$$\begin{aligned} L_{\max} &= 1.14 \quad (1/s) \quad \text{bei} \\ u_0 &= 30.98 \quad \approx 31 \end{aligned}$$

und aus den Betriebsdaten der LDB:

$$\begin{aligned} L_{B,\max} &= 1.08 \quad (1/s) \quad \text{bei} \quad (-5.26 \%) \\ u_{B,0} &= 29.84 \quad \approx 30 \quad (-3.68 \%). \end{aligned}$$

(Kap. 7.2., S. 113). Der Verlauf der beiden Systemleistungskurven $L(u)$ des Benchmarks und $L_B(u)$ der LDB-Werte sind in Abbildung 8.2. angegeben.

Wertung: Die realen Lastbedingungen lassen sich mit Hilfe des vorgelegten Benchmarks gut nachbilden. Für alle durchgeführten Experimente war die Abweichung geringer als 5%. Damit ist die Betriebslast-Nachbildung durch den TS-Benchmark möglich und zur Leistungsbewertung sinnvoll. Während der Experimente können die Antwortzeiten direkt gemessen oder durch Messung anderer Kontrollvariablen (z.B. TS-CCL) indirekt bestimmt werden. Nachteil des Benchmarks sind seine langen mittleren Verweilzeiten von über 1-1.5 Stunden pro Script-Produkt.

Bemerkung: Bei Extrapolationen der Systemleistung auf u -Werte, die wesentlich grösser sind (ca. 50%) als die mit dem dyn. TS-Benchmark gemessenen, muss die steigende Ungenauigkeit bei der Extrapolation berücksichtigt werden. In der Praxis jedoch wird ein so starker Anstieg ohnehin nur dann realisiert werden, wenn er mit der Übernahme neuer Anwendungsgebiete verbunden ist. Damit muss eine Neu-Bestimmung der (zukünftigen) Last erfolgen.

Abbildung 8.1.: Vergleich der betrieblichen Antwortzeiten mit den Benchmark-Antwortzeiten.

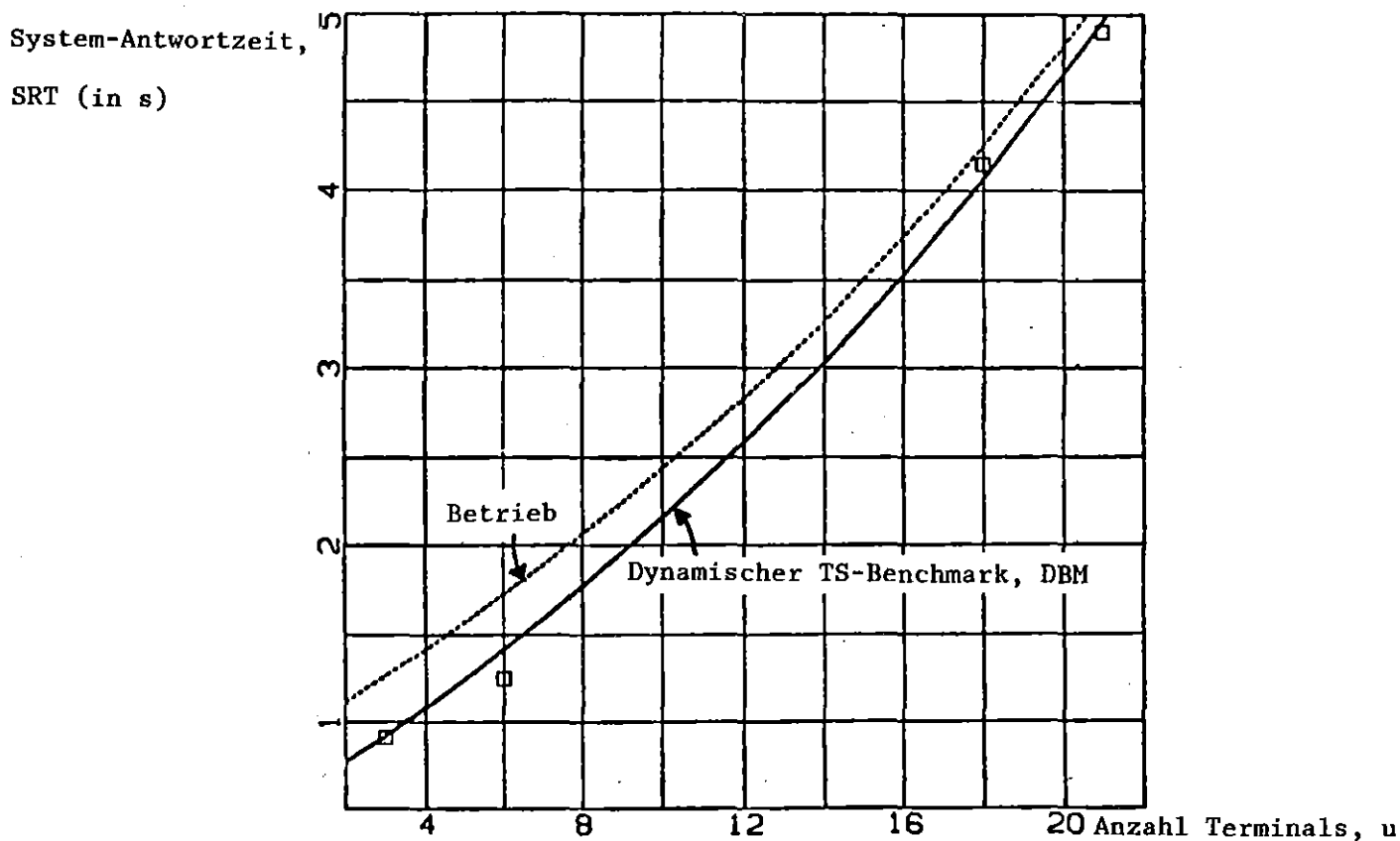
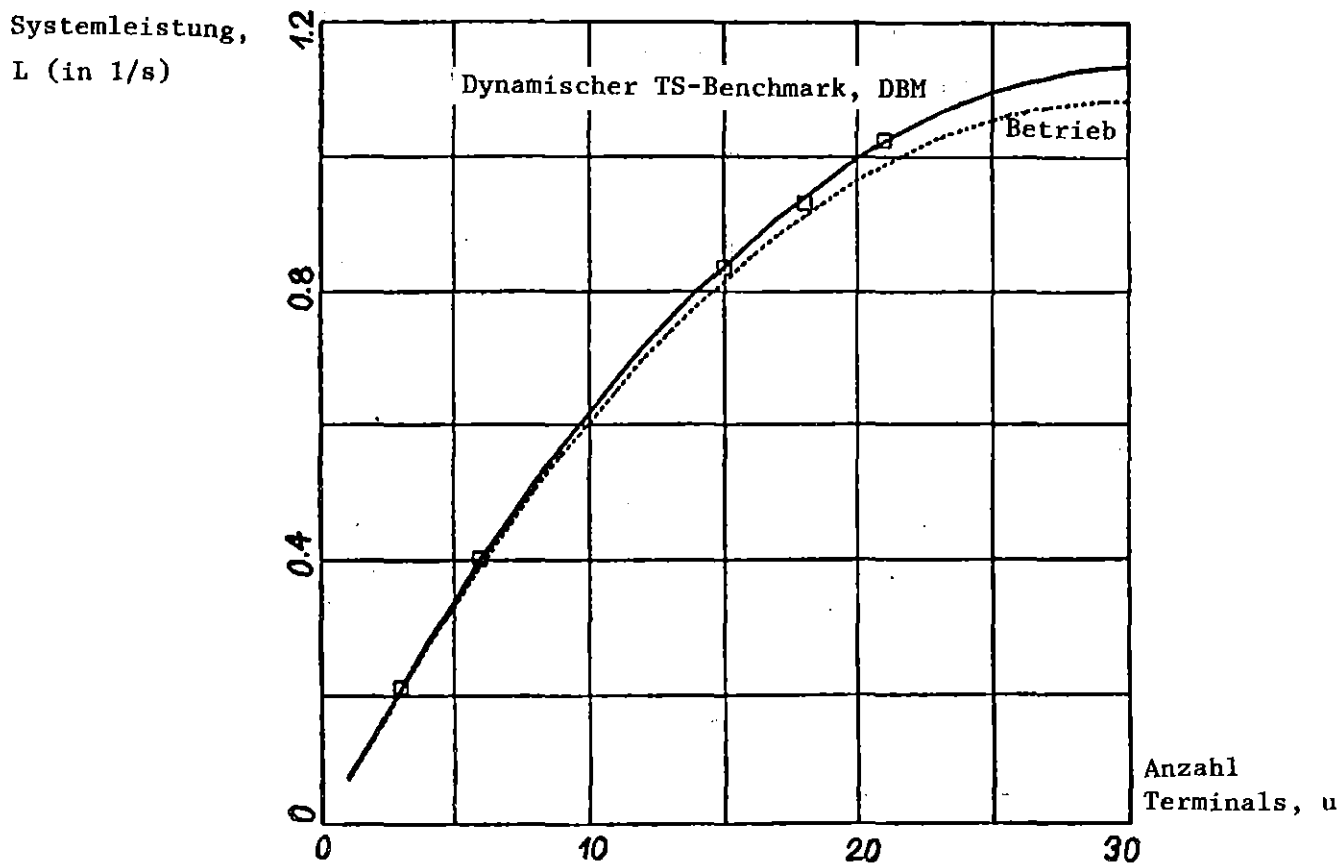


Abbildung 8.2.: Die Systemleistungskurven $L(u)$ und $L_p(u)$.



8.2.2. Leistungsvergleich verschiedener TS-Systeme mit Hilfe des dynamischen Timesharing-Benchmarks.

Aufgabe: Vergleich zweier sich durch die Hardware-Konfiguration unterscheidender Systeme durch Bestimmung der Systemleistungskurven $L(\text{System}, u)$, der Leistungsvermögen $L_{\max}(\text{System})$ und Ermittlung des Leistungsquotienten Q .

Beschreibung der experimentellen Umgebung:

System S158 (wie in Kap. 8.2.1.):

Systembeschreibung (Hardware/Software):

Systemkonfiguration 1 (s. Anhang, Kap. 9.1., S. 163).

System S168:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 3 (s. Anhang, Kap. 9.1., S. 164).

Durchführung: Zur Bestimmung der Systemleistungskurven $L(S158, u)$ und $L(S168, u)$ wird der in Kap. 8.2.1. entwickelte Benchmark eingesetzt. Da am System S168 ebenfalls APL mitbetrieben wird, so wird das APL-Quantum von 10% (Vgl. 8.2.1.) vorläufig weiterverwendet, obwohl dies bei schnellerer CPU ein grösseres absolutes CPU-Zeit-Quantum (in Sekunden) darstellt. Der DBM wird - wie in 8.2.1. - mit homogenen Script-Produkten der Grade $u = 3, 6, 18$ und 21 gefahren:

$$S_3(T, I, C) = T I C, \dots, S_{21}(T, I, C) = (T I C)^7.$$

Ergebnis: In Tabelle 8.4. sind die Gesamtverweilzeit $T_v(S168, u)$, sowie die daraus berechneten Systemleistungswerte $L(S168, u)$ und die mittlere Verweilzeit $T(S168, u)$ angegeben. Die entsprechenden Werte für das System S158 sind den Tab. 8.1. und 8.2. zu entnehmen.

Nach der Formel (4.12.) ist

$$Q = \frac{M(S168,u) \cdot u \cdot T_V(S158,u)}{M(S158,u) \cdot u \cdot T_V(S168,u)} = \frac{L(S168,u)}{L(S158,u)}$$

TABELLE 8.4.: Verweilzeiten des Timesharing-Benchmarks - S168.

u	$T_V(u)$ (s)	L(u) (1/s)	$T_S(u)$ (min)
3	11905	0.221	66 (gerundet)
6	24870	0.422	69
18	90890	1.01	84
21	108652	1.15	86

Da $M(S158,u) = M(S168,u)$ für gleiche Script-Produkte (und damit auch für gleiches u) ist, gilt:

$$Q = \frac{T_V(S158,u)}{T_V(S168,u)} ;$$

(nach Formel 4.13., S. 60). Dabei ist $T_V(S158,u) = T_V(u)$ aus Tab. 8.1. Der Leistungsquotient Q als Funktion von u ist in Tab. 8.5. tabelliert:

u	Q(u)
3	1.042
6	1.045
18	1.086
21	1.122

TABELLE 8.5.: Der Leistungsquotient Q(u).

Der Anstieg von $Q(u)$ zeigt für steigende u -Werte eine steigende Leistungsverbesserung, von 4.2 % bei $u = 3$ auf 12.2 % bei $u = 21$. Die Abb. 8.3. (S. 136) gibt die beiden unterschiedlichen Systemleistungen wieder, aus ihr sieht man den stärkeren Anstieg von $L(S168, u)$. Analog zum Vorgehen auf S. 130 (Kap. 8.2.1.) wird das Leistungsvermögen $L_{\max}(S168)$ aus den vorliegenden Daten des Benchmarks ermittelt. Die maximalen Systemleistungen von S158 und S168 sind:

$$\begin{aligned} L_{\max}(S158) &= 1.14 \quad (1/s) \quad \text{und} \\ L_{\max}(S168) &= 1.42 \quad (1/s) \quad \text{oder} + 24.6 \% \quad \text{und} \\ u_0(S158) &= 30.98 \quad (= 31) \quad \text{und} \\ u_0(S168) &= 37.85 \quad (= 38) \quad \text{oder} + 22.2 \%. \end{aligned}$$

Hieraus ergibt sich nach Formel (4.11.), S. 60, der Leistungsquotient

$$Q = \frac{L_{\max}(S168)}{L_{\max}(S158)} = 1.25$$

Eine Mehrleistung des S168 gegenüber dem S158 um 25 % wird durch den TS-Benchmark ausgewiesen. Tabelle 8.6. enthält die aus dem TS-Concurrency-Level (TS-CCL) und der Systemleistung L (gem. Formel 4.16., S. 61) berechneten und gemessenen Antwortzeiten SRT und SRT_{gem} . Abb. 8.4. stellt die Interpolation und die Extrapolation für die System-Antwortzeit $SRT(\text{System}, u)$ dar. Sie werden nach der Formel (4.16.) aus $L(\text{System}, u)$ und den gemessenen TS-CCL-Werten berechnet. Die Beantwortung der Frage 'Wieviel mehr Terminals können mit der veränderten Konfiguration betrieben werden, wenn noch eine mittlere Antwortzeit von nicht mehr 20 Sek. eingehalten werden soll?' ist nach Bestimmung des Faktors

$$\frac{u(S168, SRT \leq 20 \text{ s})}{u(S158, SRT \leq 20 \text{ s})} = \frac{63}{42} = 1.50$$

möglich, nämlich ca. 50% mehr.

TABELLE 8.6.: Antwortzeiten-Vergleich, S158 und S168.

u	SRT(S158)	SRT _{gem} (S158)	SRT(S168)	SRT _{gem} (S168)	Veränd., gem.
3	1.00	0.91	0.48	0.49	46 %
6	1.28	1.25	0.53	0.52	58 %
18	4.19	4.15	1.44	1.42	66 %
21	4.92	4.90	1.73	1.69	66 %

8.2.3. Leistungsvergleich verschiedener TS-Systeme anhand der Daten aus der Leistungsdatenbank.

Aufgabe: 1. Die in Kap. 8.2.2. mit Hilfe des TS-Benchmarks gelöste Aufgabe des Leistungsvergleichs soll mittels der in der Leistungsdatenbank (LDB) gespeicherten Datenwerte durchgeführt werden.

Dazu sind die Betriebs-Systemleistungskurven $L_B(\text{System}; u)$ festzustellen; $L_{B, \text{max}}(\text{System})$ und Q_B sind zu berechnen (vgl. Kap. 7.4.). Darüberhinaus sind relative Systemleistung $l(\text{System}; u)$ (siehe Formel 4.10.), die Kapazitätskennzahlen $C_S(\text{System})$, sowie die Servicerate B und die Antwortzeit SRT zu bestimmen.

2. Die Ergebnisse sind mit den Resultaten von Kap. 8.2.2. zu vergleichen, um die hier eingesetzte Methode des Leistungsvergleichs qualitativ einzuordnen.

Abbildung 8.3.: Systemleistungskurven für S158 und S168.

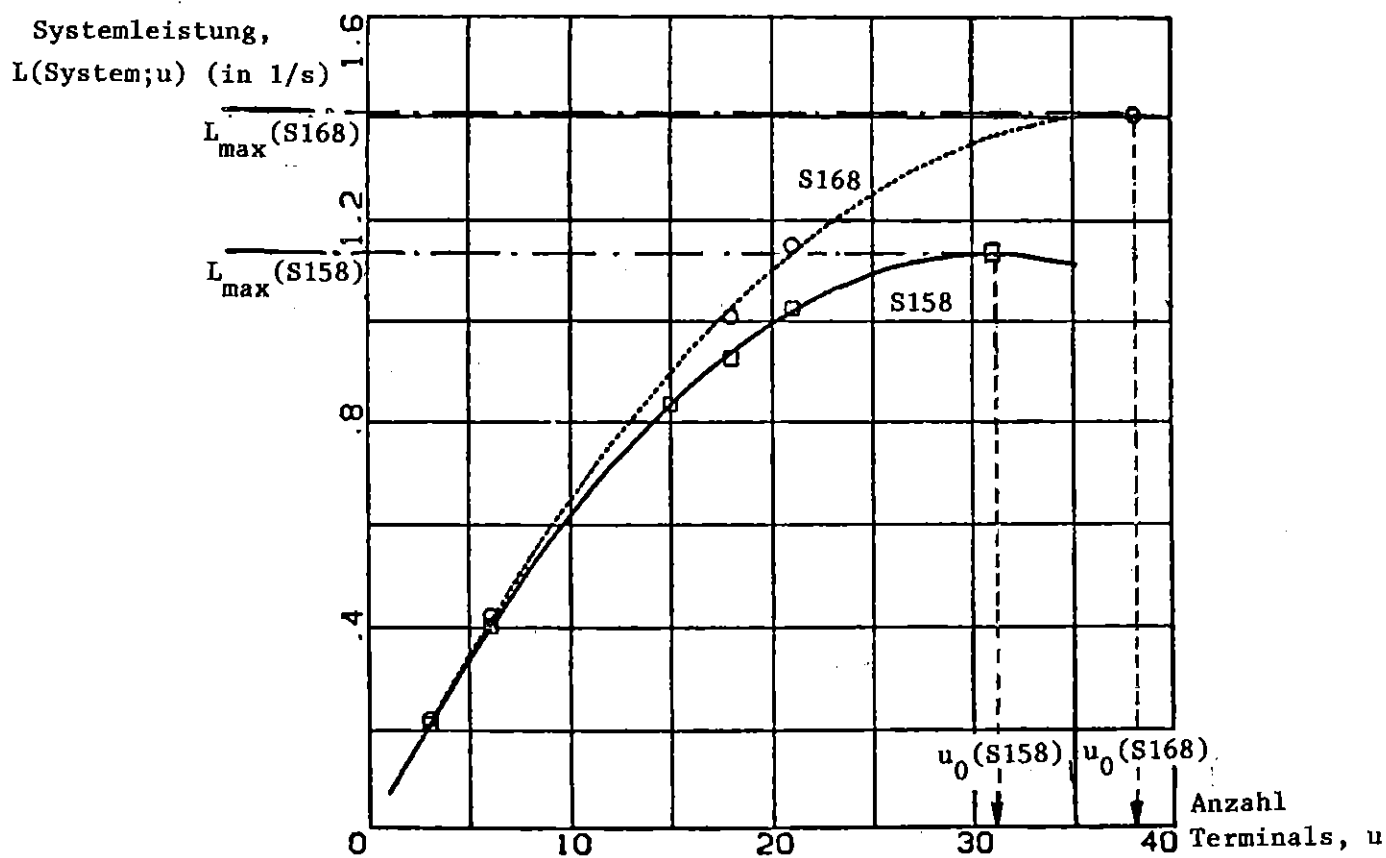
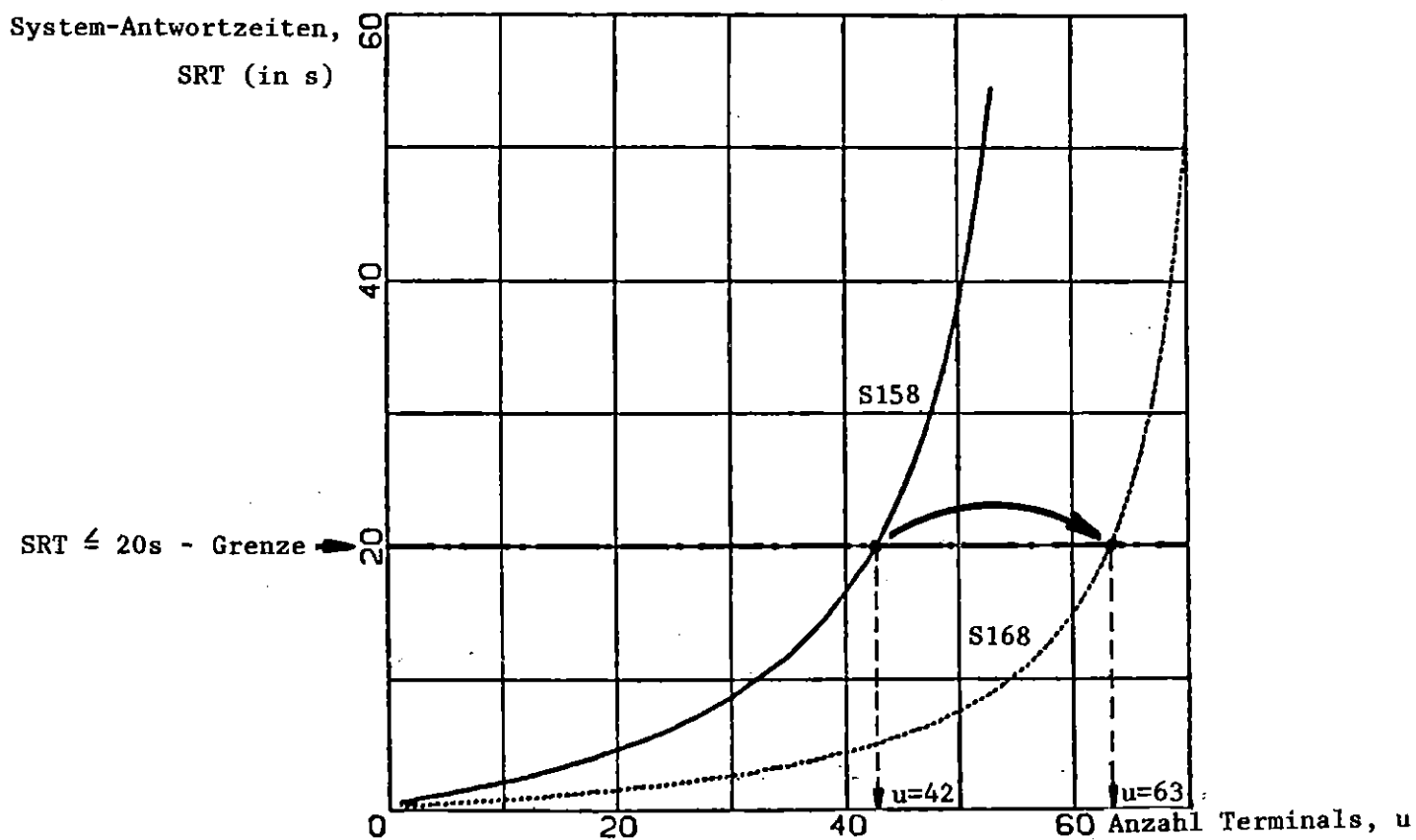


Abbildung 8.4.: Extrapolation der Antwortzeiten für S158 und S168.



Beschreibung der Umgebung:

System S158: wie in Kap. 8.2.2.;

Systembeschreibung (Hardware/Software):

Systemkonfiguration 1 (s. Anhang, Kap. 9.1., S. 163).

System S168: wie in Kap. 8.2.2.;

Systembeschreibung (Hardware/Software):

Systemkonfiguration 3 (s. Anhang, Kap. 9.1., S. 164).

Bemerkung: Als Vergleichs-Zyklen sollen neben den zur Benchmark-Erzeugung verwendeten Daten des August 1978 (für S158) die Daten aus dem März 1979 (für S168) verwendet werden, da das System S168 erst Anfang Februar 1979 in Betrieb genommen wurde.

Durchführung der Auswertung: Auswahl eines repräsentativen Zyklus aus den Betriebsdaten für S158 (18. Aug. 1978, 1. Schicht) und für S168 (2. März 1979, 1. Schicht) gemäss Kap. 2.5.3. Mit Mehrfeldertafeln werden die folgenden Kontrollvariablen jeweils für beide Vergleichs-Zyklen auf Homogenität, d.h. hier Repräsentativität, geprüft: u, TRX, TS-CCL und SRT.

Ergebnis: 1. Abb. 8.5.(S. 139) zeigt die beiden betrieblichen Systemleistungskurven $L_B(S158;u)$ und $L_B(S168;u)$; sie werden durch die Messwerte der Transaktionsrate definiert. Es ergeben sich daraus:

$$\begin{aligned} L_{B, \text{mmax}}(S158) &= 1.29 \quad (1/s) \text{ und} \\ L_{B, \text{mmax}}(S168) &= 1.46 \quad (1/s) \text{ mit} \end{aligned}$$

$$\begin{aligned} u_{B,0}(S158) &= 37 \quad \text{und} \\ u_{B,0}(S168) &= 42 \quad , \end{aligned}$$

sowie $Q_B = 1.13.$

Der Leistungsquotient Q wird hier deutlich kleiner ausgewiesen als in Kap. 8.2.2. ($Q=1.25$).

Die relative Systemleistung l :

$$l(\text{System};u) = \text{TRX}(\text{System};u)/L_{B,mmax}(\text{System})$$

wird zur Bestimmung der relativen Summenhäufigkeit $S_R(l)$ von berechnet. $S_R(l)$ und $P_R(l)$ (normalverteilte Approximation von $S_R(l)$) sind für beide Systeme in Abb. 8.6. angegeben. $S_R(l)$ erreicht für $l = 1$ jeweils nicht den Wert 1, Grund dafür sind die $L_{B,mmax}$ überschreitenden Werte der Transaktionsrate TRX an. Es sind $L_{B,amax}(S158)=1.40$ und $L_{B,amax}(S168)=1.56$ (s. Kap. 7.2., S. 113).

Die Kennzahl der 'freien Kapazität' $C_S(\text{System})$ wird nach Formel (4.10.1.), S. 58, berechnet. Es sind:

$$\begin{aligned} C_S(S158) &= 0.352 \quad \text{und} \\ C_S(S168) &= 0.338 \quad . \end{aligned}$$

Die mittleren Antwortzeiten (aus der LDB) sind in Abbildung 8.7. als Punkte eingetragen (empirische Werte). Die berechneten Werte der Antwortzeit SRT sind als Linien eingezeichnet:

$$\text{SRT}(\text{System};u) = \frac{\text{TS-CCL}(\text{System};u)}{\text{TRX}(\text{System};u)}$$

Die berechneten SRT-Werte sind bis zu $u=60$ extrapoliert. Die Punkte der stärksten Krümmung (Scheitelpunkte) liegen bei $u=50$ (S158) und $u=70$ (S168); jenseits dieser Punkte beträgt der mittlere Anstieg der Kurven mehr als 1, an diesen Punkten wird die Systemsättigung erreicht.

Abbildung 8.5. Betriebs-Systemleistungskurven für S158 und S168.

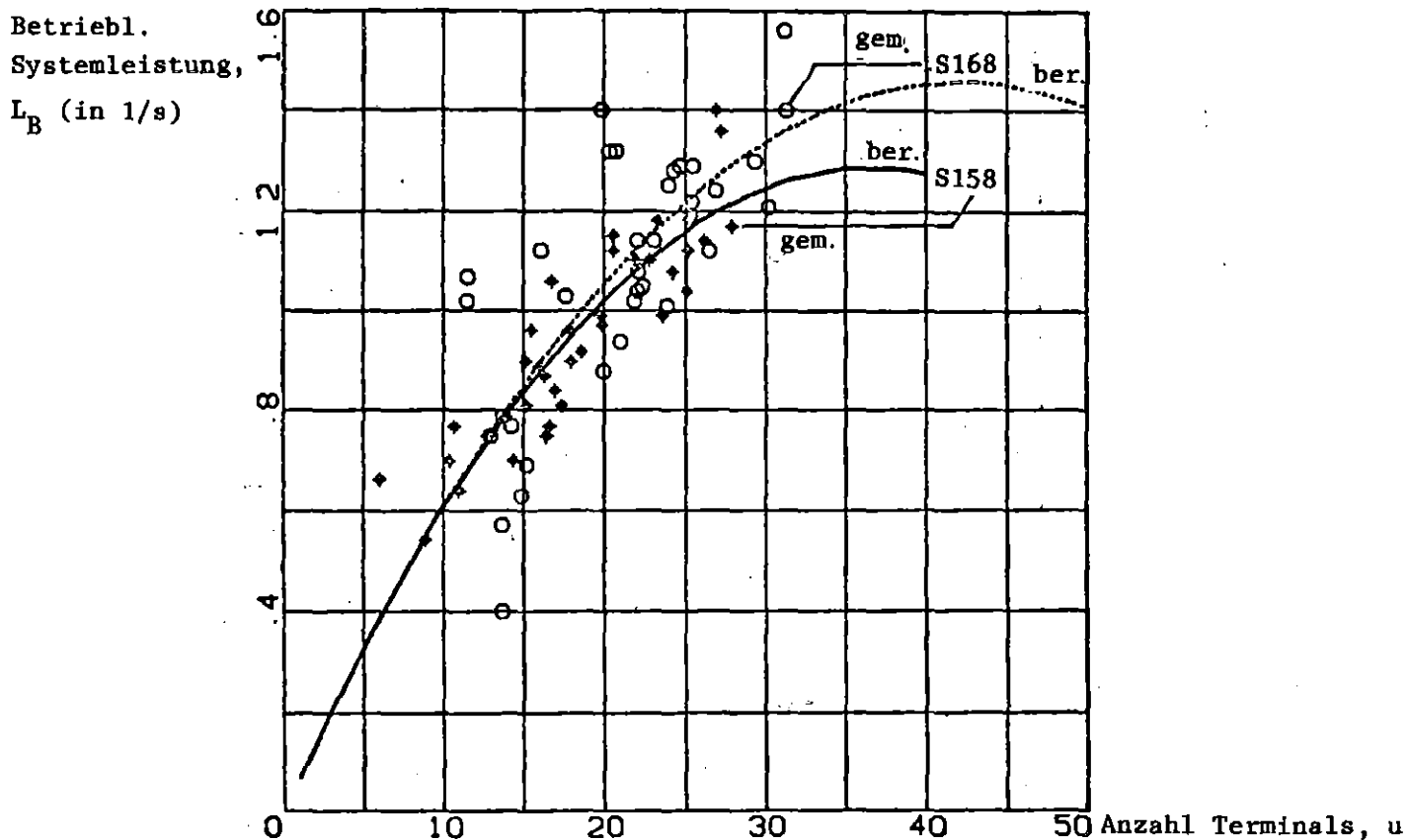


Abbildung 8.6. Summenhäufigkeit $S_R(l)$ der relativen Systemleistung.

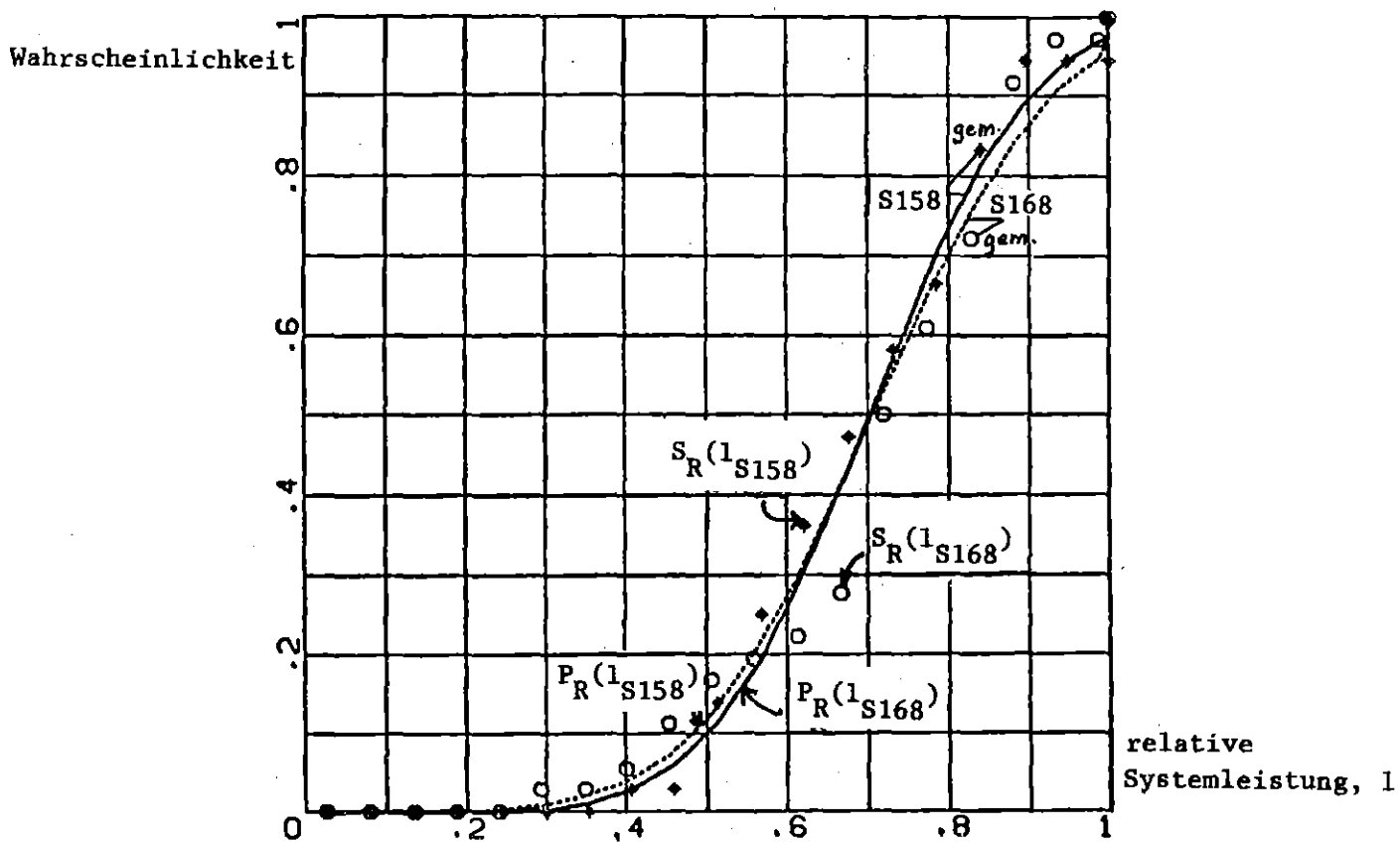
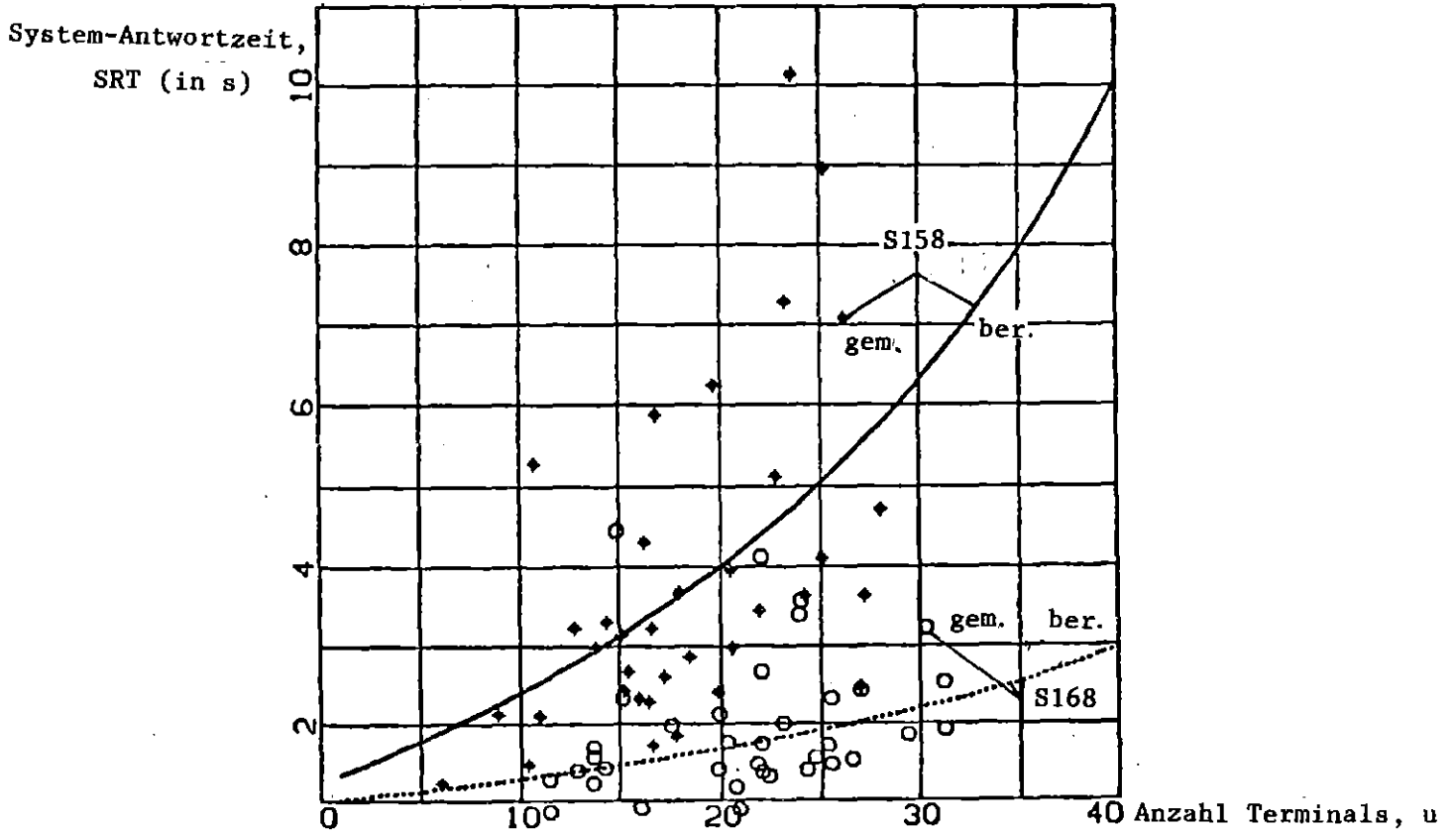


Abbildung 8.7.: Antwortzeiten für S158 und S168.



Ergebnis: 2. Ein Vergleich der Leistungskurven von $L_B(\text{System};u)$ mit $L(\text{System};u)$ aus Kap. 8.2.2. zeigt folgende Abweichungen:

System	$L_{B,mmax}(\text{System})$	$L_{max}(\text{System})$	$L_{B,mmax} - L_{max}$	Abweichung
S158	1.29 1/s	1.14 1/s	0.15	+ 13 %
S168	1.46 1/s	1.42 1/s	0.04	+ 3 %

Wertung: Die in der o.a. Tabelle angegebenen Abweichungen zeigen, dass die in der LDB gespeicherten Betriebsdaten gute Näherungen für die Systemleistungskurven ergeben; jedoch muss die sehr geringe Abweichung für S168 auf eine zufällige sehr gute Übereinstimmung der Betriebslast des ausgewählten Zyklus mit der Benchmarklast zurückgeführt werden.

8.2.4. Leistungsvergleich anhand der Verweil- und System-Antwortzeiten bei unterschiedlicher Platten-Konfiguration - Pseudo-Benchmark.

Aufgabe: Die Leistungsverbesserung durch eine Magnetplattenperipherie-Änderung ist anhand der veränderten System-Antwortzeiten zu quantifizieren.

Beschreibung der experimentellen Umgebung:

System PIBM:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 3 (s. Anhang, Kap. 9.1., S. 164).

System PCDC:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 5 (s. Anhang, Kap. 9.1., S. 164).

Hardware wie bei System PIBM, ausgenommen:

Benutzer-Datei-Residenz auf CDC 33801- Platten im
IBM 3330-II-Modus (18 ms mittl. Zugriffszeit);

Verfahren: Einsatz eines Pseudo-Benchmarks (PBM), der nach dem Parametrierungsverfahren aus Kap. 6.5., S. 99-102, festgelegt wird. Die Parametrierung des PBM selbst ist in Kap. 6.5., S. 104 ff., für die Systemkonfiguration 3 durchgeführt. Der PBM wird hier eingesetzt, weil dieser mit geringerem Aufwand zu fahren ist als der dynamische BM. Die nur mit dem DBM zu messende Systemleistung L wird zur Bewertung nicht herangezogen (s. Aufgabenstellung).

Es wird das Script-Polynom $P(5\%)$ (von S. 106) verwendet:

$$P(5\%) = P_4(T, I, C) = SP(2) + SP(5) + SP(11)$$

Eine Lastanpassung mit 5% mittlerem relativem Fehler ist gewährleistet (S. 106, Kap. 6.5.).

Durchführung: Bezeichnet man die nach Kap. 6.5., S. 104, parametrisierten Basis-Scripts mit T , I und C , so bedeuten die Script-Produkte

$$SP(2) = T^0 I^1 C^0,$$

$$SP(5) = T^0 I^2 C^0 \text{ und}$$

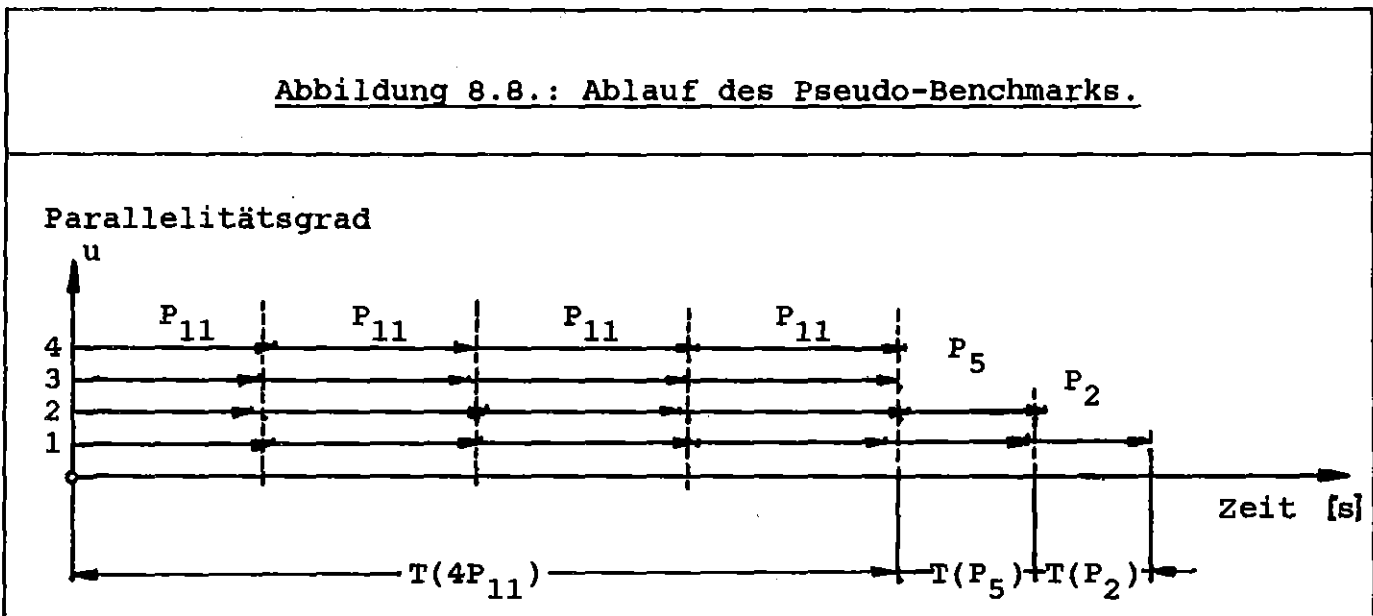
$$SP(11) = T^0 I^4 C^0.$$

Der Exponent des Basis-Scripts gibt den zu fahrenden Parallelitätsgrad an. Das Script-Polynom $P(5\%)$ hat dann die Darstellung:

$$P(5\%) = T^0 I^1 C^0 + T^0 I^2 C^0 + 4 T^0 I^4 C^0.$$

Gefahren werden die Script-Potenzen in absteigender Folge ihrer Grade, also 4mal P_{11} , 1mal P_5 und 1mal P_2 (Abb. 8.8.).

Abbildung 8.8.: Ablauf des Pseudo-Benchmarks.



Ergebnis: Die für den Pseudo-Benchmark gemessenen Verweilzeiten für die beiden zu vergleichenden Systeme sind in der Tabelle 8.7. aufgeführt.

<u>Tabelle 8.7. Verweilzeiten des Pseudo-Benchmarks.</u>					
i	k	T(PIBM, kP _i) s	T(PCDC, kP _i) s	T s	Abweichung %
2	1	203	198	5	2.5
5	1	250	241	9	3.6
11	4	1620	1501	119	7.4
Summe:		2073	1940	133	6.4

<u>Tabelle 8.8. System-Antwortzeiten des Pseudo-Benchmarks.</u>					
i	k	T(PIBM, kP _i) s	T(PCDC, kP _i) s	T s	Abweichung %
2	1	0.30	0.28	0.02	6.7
5	1	0.60	0.53	0.07	11.7
11	4	1.60	1.39	0.21	13.1
Mittel:		1.22	1.06	0.16	13.1

Tabelle 8.8. gibt die während des Ablaufs des Pseudo-Benchmarks (per GTF) gemessenen System-Antwortzeiten (SRT) an.

Wertung: Das hier eingesetzte Script-Polynom als PBM lässt keine Variation der Anzahl der Benutzer zu. Es ermöglicht daher nur eine Bewertung der Laufzeitveränderung anhand des

statischen Lastprofils, das durch den PBM definiert wird. Die Senkung der Verweilzeit um 6.4 % und der Antwortzeiten um 13.1 % kann hier nur zur Bewertung der Veränderung unter diesen, nach Kap. 6.5., S. 104, mittleren Lastbedingungen herangezogen werden.

Die Veränderung der Benutzer-Datei-Konfiguration von Platten mit 30 ms mittlerer Zugriffszeit auf solche mit 18 ms mittlerer Zugriffszeit erbringt bei der durch den PBM definierten mittleren Last nur einen reduzierten 'Gewinn' von 13.1 % bei den System-Antwortzeiten.

8.2.5. Auswahl eines Standard-Blockungsfaktors: Anwendung eines Basis-Scripts als Pseudo-Benchmark.

Aufgabe: Für den installationsspezifischen Standard-Blockungsfaktor bei Dateien mit Sätzen fester Länge, der immer dann zur Anwendung kommt, wenn der TS-Benutzer keine eigene Spezifikation abgibt, ist ein optimaler Wert durch den Einsatz eines Pseudo-Benchmarks unter folgenden Randbedingungen zu finden:

- A) die Antwortzeit soll nicht (wesentlich) vom Blockungsfaktor beeinflusst werden,
- B) die Speicherkapazität der Magnetplatten soll maximal ausgenutzt werden und
- C) unter der Annahme von 4 ständig im Hauptspeicher gehaltenen I/O-Puffern für Sätze von 80 Bytes sollen bei dem festzulegenden Standardwert nicht mehr als 5% des (installationsbedingt) dem TS-Benutzer zur Verfügung stehenden Hauptspeichers von 320K belegt werden.

Bemerkung: Die Festlegung des Standard-Blockungsfaktors (BF) findet nach einem Wechsel der Platten-Konfiguration statt. Gleichzeitig mit der Parameterauswahl wird die Antwortzeit zwischen beiden Konfigurationen verglichen.

Beschreibung der experimentellen Umgebung:

Zur Feststellung der Antwortzeiten:

System P2314:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 1 (s. Anhang, Kap. 9.1., S. 163).

System P3330:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 3 (s. Anhang, Kap. 9.1., S. 164).

Für das System P3330 ist der Standardwert des BF zu ermitteln.

Verfahren und Durchführung: Da es sich hier um die Beurteilung der I/O-Konfiguration handelt, wurde für den einzusetzenden Pseudo-Benchmark ein eine künstliche "Stress-Last" (die nicht notwendigerweise der natürlichen Last entspricht) erzeugender I/O-Treiber (Basis-Script) mit den folgenden Parametern (Kap. 6.3., S. 78-81) ausgewählt:

$$n_I = 20, i_I = 1000 \text{ und } Z = 1.$$

Gemessen wird nur die Antwortzeit der klassenspezifischen Hauptfunktion des Scripts, eines COPY-Kommandos.

Abbildung 8.9. zeigt die mit dem Pseudo-Benchmark gemessenen Antwortzeiten, und zwar für das System I (P2314) und für das System II (P3330). Abbildung 8.10. gibt die Ausnutzung des Plattenspeicherplatzes in % für das System II (fest geblockte Dateien mit einer Satzlänge von 80 Bytes) Alle nicht eingetragenen Werte liegen unterhalb des nächstgrösseren eingezeichneten Auslastungswertes. Abbildung 8.11. zeigt den Haupt-

speicherbedarf für die I/O-Puffer unter der Annahme C). Abb. 8.12. gibt die Anhängigkeit der Antwortzeit von der Grösse der bearbeiteten Datei bei festem Blockungsfaktor $BF=39$ wieder.

Ergebnis: Nach Abbildung 8.9. muss der auszuwählende BF-Wert > 21 (dem bisherigen Standardwert, System I) werden. Für Werte > 30 nimmt $SRT(P3330;BF)$ kaum noch ab. Die Abbildung 8.10. lässt als speicherplatz-optimalen Wertevorrat nur wenige diskrete Werte von BF zu ($BF = 39, 53, 80$ und 162) für die fest Satzlänge von 80 Bytes zu, wobei mit der Grösse auch die Ausnutzung der Platten zunimmt.

Die Abbildung 8.11 schränkt gemeinsam mit der Forderung C) BF nach oben ein; die Grenze von 5% des verfügbaren Hauptspeichers von 320 Kbytes wird schon bei einem Blockungsfaktor von 39 erreicht. $BF = 53, 80, 162$ kommen daher nicht mehr in Betracht, obwohl sie eine noch bessere Plattenspeicher-Ausnutzung erlauben würden.

Für den bisherigen Standard-Blockungsfaktor von 21 sinkt die Antwortzeit $SRT(P3330;BF=21)$ von 12 auf 9 s (für $SRT(P2314;21)$) um 25%, für den neuen Standardwert $BF=39$ sogar um ca. 50%.

Die Antwortzeit-Reduzierung allein durch Vergrösserung des Blockungsfaktors beträgt ca. 44% (Vergleich von $SRT(P3330;21)$ mit $SRT(P3330;39)$, Abbildung 8.9.).

Das Antwortzeitverhalten hängt dagegen nur sehr wenig von der Grösse der Datei ab (Abbildung 8.12.).

8.2.6. Bewertung einer Peripherie-Änderung anhand der Antwortzeiten und des Produktivitätsgewinns.

Aufgabe: Für drei verschiedene Plattenkonfigurationen ist anhand der mittleren Antwortzeiten I/O-orientierter Kommandos und daraus berechneter Produktivitätsgewinne (Kap. 4.6., Formel 4.26., S. 66) eine Bewertung der Konfigurationen durchzuführen.

Abbildung 8.9.: I/O-Antwortzeit als Funktion des Blockungsfaktors.

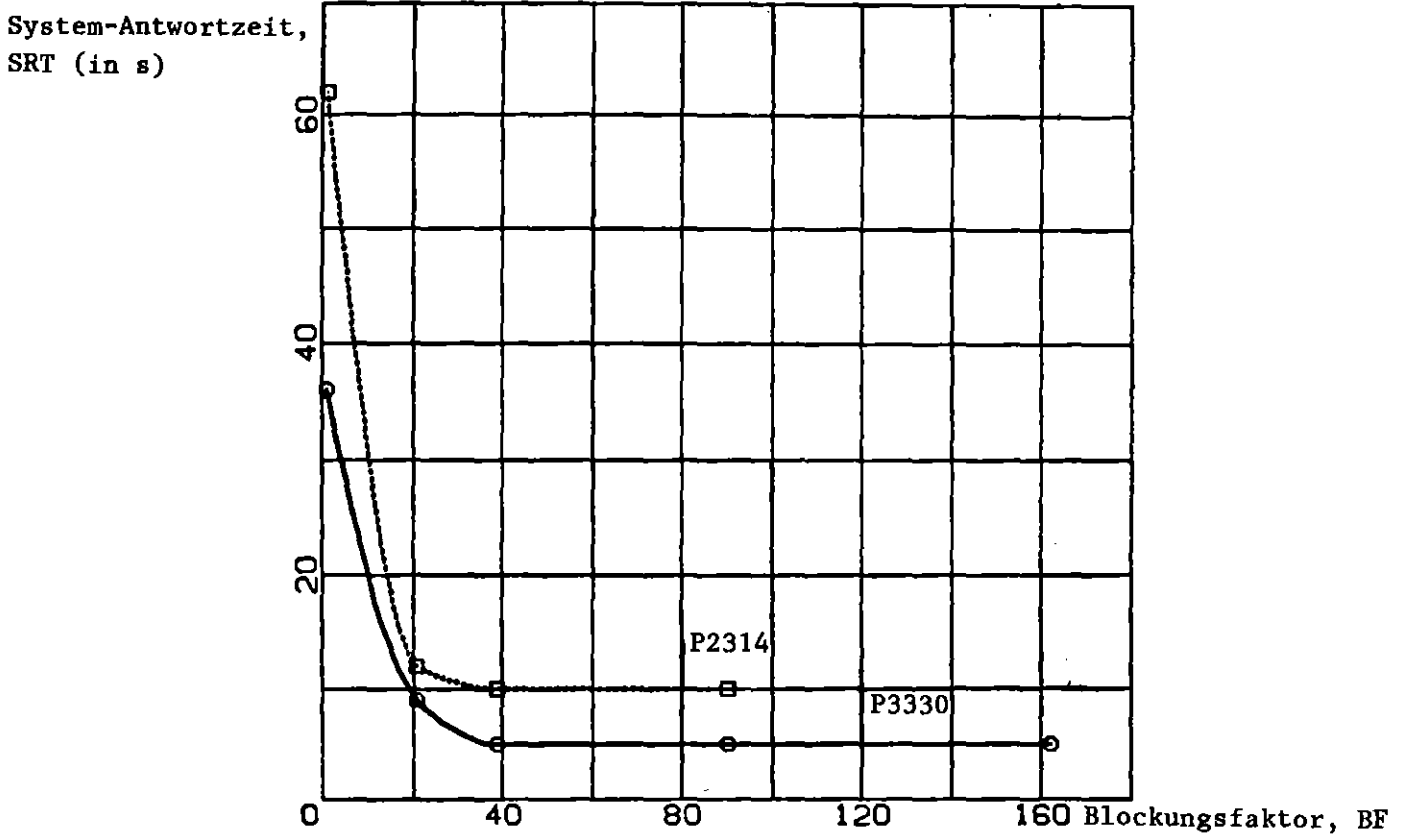


Abbildung 8.10.: Ausnutzung der Plattenspeicherkapazität als Funktion des Blockungsfaktors.

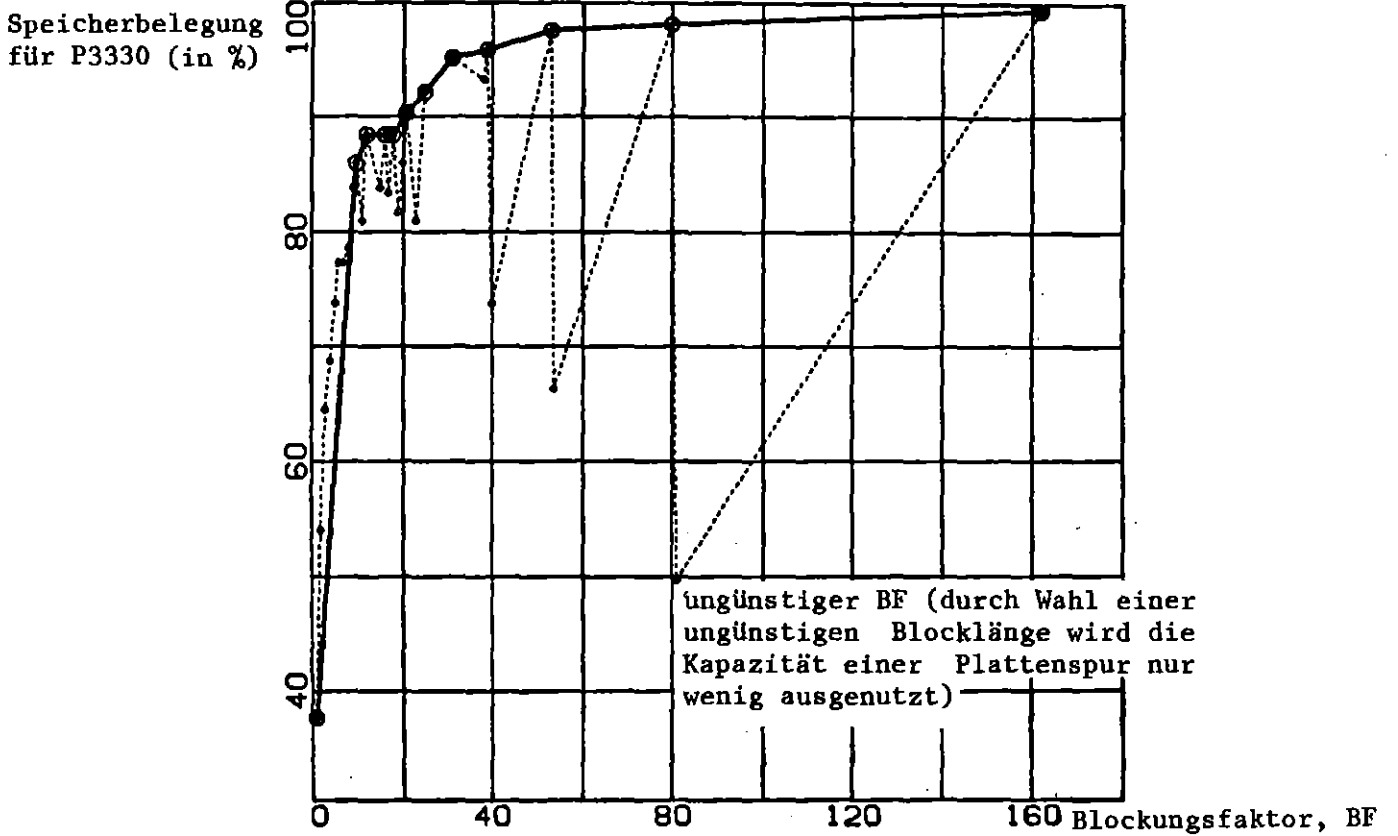


Abbildung 8.11.: Hauptspeicherbedarf für I/O-Puffer in Abhängigkeit vom Blockungsfaktor.

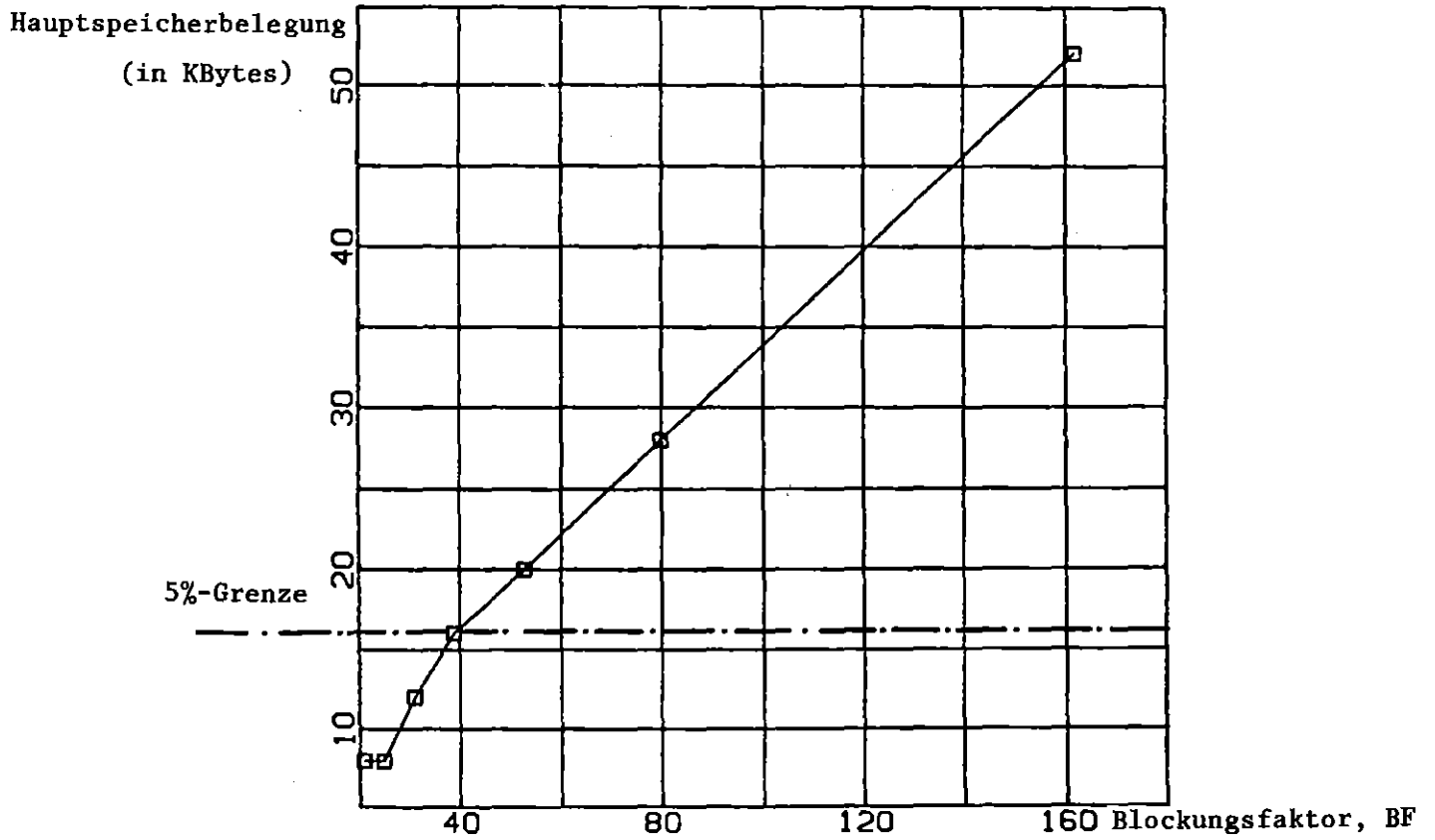
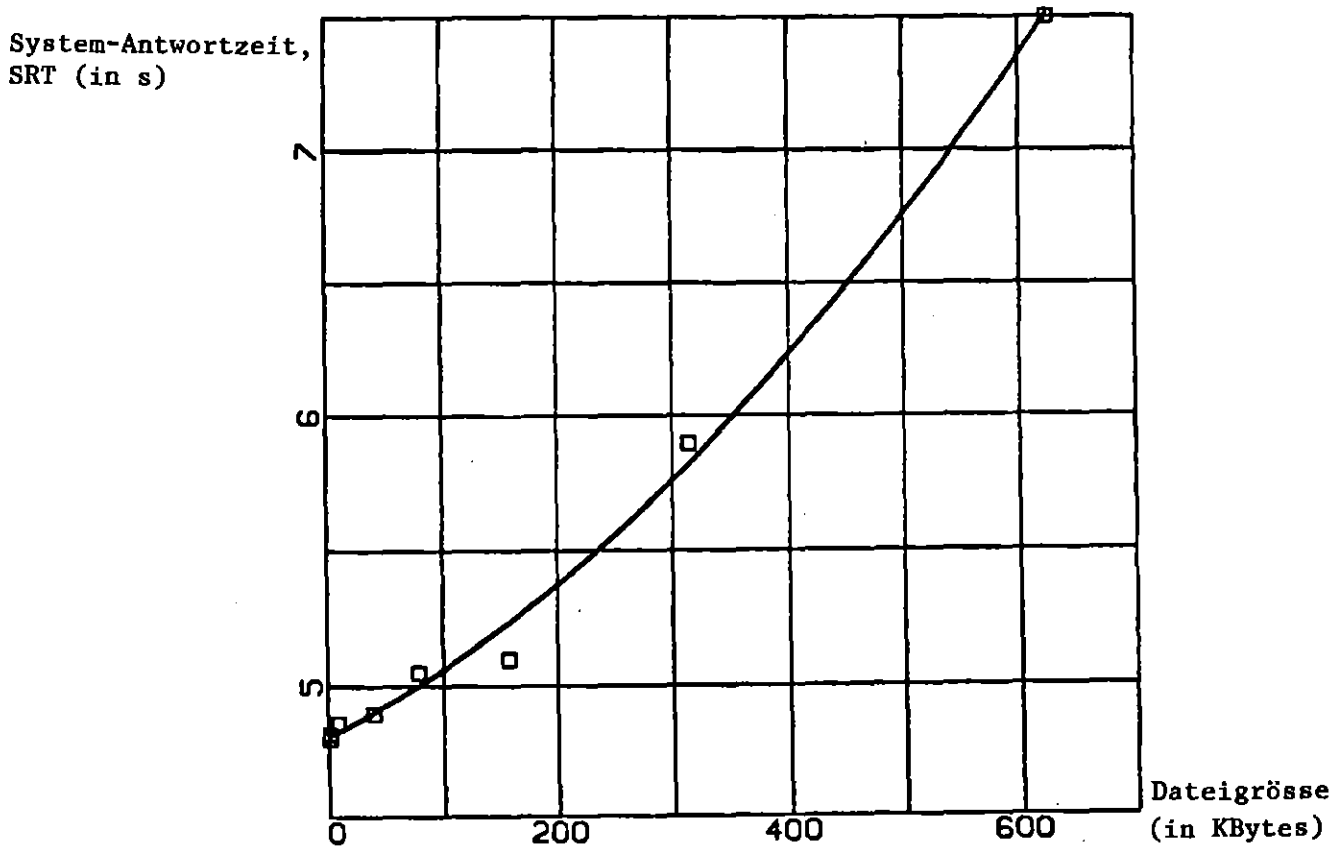


Abbildung 8.12.: I/O-Antwortzeiten in Abhängigkeit von der Dateigrösse.



Beschreibung der experimentellen Umgebung:

System P2314:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 1 (s. Anhang, Kap. 9.1., S. 163).

System P3330:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 3 (s. Anhang, Kap. 9.1., S. 164).

System P33801:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 5 (s. Anhang, Kap. 9.1., S. 164).

Verfahren: Für die vorliegende Aufgabe bietet sich Pseudo- und Quasi-BM als mögliche Verfahren an. Da der QBM ohne Betriebsunterbrechungen zu fahren ist, wird er hier vorgezogen. Mit Hilfe eines I/O-orientierten Basis-Scripts $I(n_I=20, i_I=1000, Z=1)$ wird der Quasi-Benchmark als zusätzliche 'Stress-Last' der I/O-Konfiguration während des Normalbetriebs (1. Tagesschicht) gefahren. Die Antwortzeiten der COPY-Funktion des Treibers I werden gemessen und aufgezeichnet. Die Verteilung der Benutzer des TS-Systems über die Tageszeit wird mit der Resource Measurement Facility RMF (siehe Kap. 2.) gemessen. Zur Bestimmung des Anteils der I/O-orientierten Kommandos des betrieblichen Transaktionsstroms wird an allen Messtagen jeweils für 30 Minuten die gesamte Arbeitslast aufgezeichnet und daraus der mittlere Anteil der I/O-orientierten Funktionen bestimmt.

Mit Ausnahme der "Übergangskonfiguration" System = P3330 ist die Benutzerverteilung über die Laufzeit des Pseudo-BM und die Funktionsklassenverteilung aller Kommandos festzustellen. Daraus erfolgt die Berechnung des Produktivitätsgewinns $F_{I/O}$ für die I/O-orientierten Kommandos beim Übergang von Plattenkonfigurationen P2314 zu P33801.

Durchführung: System P2314 : 30 Messungen, 22.-25.8.1978, jeweils zwischen 9.00 und 17.00 Uhr (Vgl. auch die Klassifizierung der Betriebsmodi in Kap. 3.2.);

System P3330 : Übergangskonfiguration, daher nur 11 Messungen, 24. und 27.10.1978, jeweils zwischen 9.00 und 17.00 Uhr;

System P33801 : 32 Messungen, 8.-23.5.1979, jeweils zwischen 9.00 und 17.00 Uhr.

Alle QBM-Laufzeiten wurden auf Auffälligkeiten anhand des System-Konsol-Reports (Platten-Ausfall, Software-Störungen usw.) überprüft, Messungen während solcher Störungen eliminiert.

Ergebnis: Die Antwortzeiten der drei Experimente sind in Abbildung 8.13. dargestellt. Bei einer Benutzerzahl von $u = 20$ betragen die mit dem QBM gemessenen Antwortzeiten

$$\begin{aligned} \text{SRT}(\text{P2314}; u=20) &= 15 \text{ s,} \\ \text{SRT}(\text{P3330}; u=20) &= 10 \text{ s und} \\ \text{SRT}(\text{P33801}; u=20) &= 7.5 \text{ s,} \end{aligned}$$

also Antwortzeitreduktionen von 33% von P2314 zu P3330 und 25 % von P3330 zu P33801, bzw. 50 % von P2314 zu P33801.

Der Produktivitätsgewinn für die I/O-orientierten Kommandos nach Formel (4.26., S. 66) bewertet den Übergang von der I/O-Konfiguration P2314 zu P33801:

$$F_{I/O} = \sum_{u_i=1}^{u_{\max}=28} u_i \left(\frac{\text{SRT}(\text{P2314}; u_i) p(\text{P2314}; u_i) r(\text{P2314})}{T(\text{P2314})} - \frac{\text{SRT}(\text{P33801}; u_i) p(\text{P33801}; u_i) r(\text{P33801})}{T(\text{P33801})} \right)$$

Dabei sind $\text{SRT}(\text{System}; u)$ die Antwortzeiten aus Abbildung 8.13., $p(\text{System}; u)$ die relativen Häufigkeiten, mit der u Benutzer während der Messperiode $T(\text{System})$ für $\text{System} = \text{P2314}, \text{P33801}$ angeschaltet waren (Messung durch RMF) und $r(\text{System})$ ist der Anteil der I/O-orientierten Kommandos (gem. Kap. 5.) des mit

der General Trace Facility GTF aufgezeichneten TRX-Stromes:

$$r(P2314) = 0.52 \quad \text{und}$$

$$r(P33801) = 0.57$$

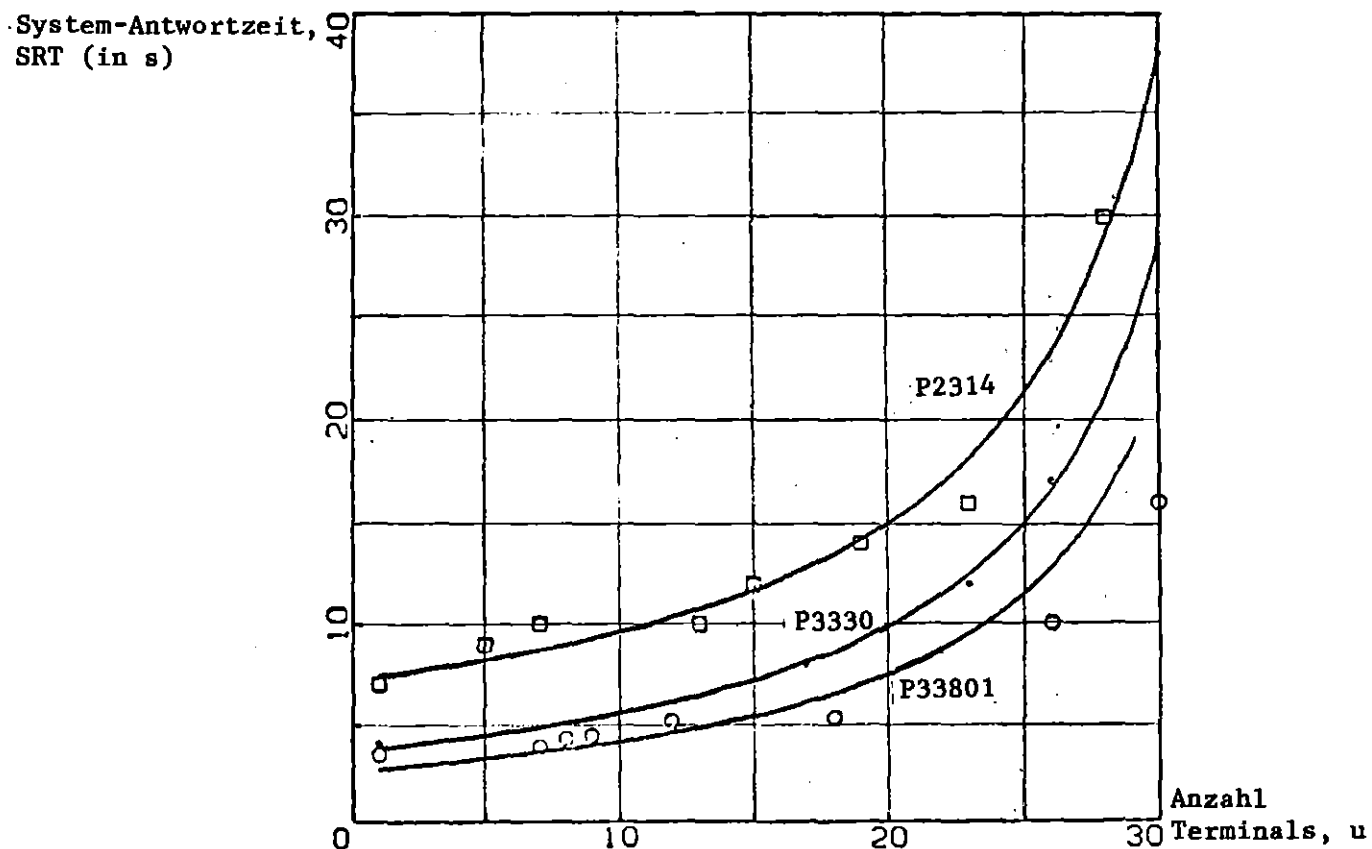
(bei 8 Stichproben zu 30 min bei P2314 mit insgesamt 14487 erfassten Kommandos und bei 6 Stichproben zu je 30 min bei P33801 mit insgesamt 14042 erfassten Kommandos).

Es ergibt sich der Produktivitätsgewinn von:

$$F_{I/O} = 0.17,$$

oder 17%, für die I/O-orientierten Funktionen beim Übergang von P2314 auf P33801.

Abbildung 8.13.: Antwortzeiten des Quasi-Benchmarks in Abhängigkeit von der Magnetplatten-Hardware.



8.2.7. Beurteilung von Veränderungen von Stellgrößen der Betriebs-System-Software.

Aufgabe: Mittels des TS-Benchmarks ist der Einfluss des APL-Benutzer-CPU-Quantums, eines Installationsparameters der System-Software, auf die TS-Systemleistung $L(\text{SAPL};u)$, die System-Antwortzeit $\text{SRT}(\text{SAPL};u)$ und auf die CPU-Auslastung $\text{CPU}(\text{SAPL};u)$ zu beurteilen.

Beschreibung der experimentellen Umgebung:

System:

Systembeschreibung (Hardware/Software):

Systemkonfiguration 1 (s. Anhang, Kap. 9.1., S. 163).

APL läuft auf derselben Zentraleinheit mit höherer Priorität, jedoch wird APL in seiner Ausführung unterbrochen, wenn während einer 100 ms grossen Zeitscheibe von APL mehr als der Grenzwert SAPL an anteiliger CPU-Zeit verbraucht worden ist.

Durchführung: Zur Quantifizierung des Einflusses des SAPL-Wertes wird der Parameter verändert: $\text{SAPL} = 0, 10$ und 20% . Für den Versuch $\text{SAPL} = 0$ wird das APL-System nicht betrieben.

Der in Kap. 8.2.1. beschriebene dynamische TS-Benchmark wird als Folge von homogenen Script-Produkten:

$$S_3 = T I C,$$

...

$$S_{21} = (T I C)^7$$

jeweils allen Parameter-Werten $\text{SAPL} = 0, 10$ und 20 gefahren. Für die Experimente mit $\text{SAPL} = 10$ und 20 wird ein APL-Programm gestartet, welches unbeschränkt CPU-Zeit verbraucht, diese wird nur durch den eingestellten Parameter SAPL eingeschränkt.

Ergebnisse: Die Funktionen $L(\text{SAPL};u)$ sind für $\text{SAPL} = 0, 10$ und 20 und $u = 3, 6, 18$ und 21 in der Abb. 8.14. dargestellt. Die maximalen Systemleistungen betragen

$$\begin{aligned}L_{\max}(\text{SAPL}=20) &= 0.88, \\L_{\max}(\text{SAPL}=10) &= 1.14 \text{ und} \\L_{\max}(\text{SAPL}=00) &= 1.38.\end{aligned}$$

Dies entspricht einem Verlust von 36.2 % bei $\text{SAPL}=20$ und von 17.4 % bei $\text{SAPL}=10$ bei der Systemleistung gegenüber dem Benchmark-Experiment ohne APL ($\text{SAPL}=0$). Zum Vergleich ist in Abb. 8.14. die in Kap. 8.2.1. für den Normalbetrieb bestimmte Systemleistungskurve $L_B(u)$, mit $L_{B,\text{mmax}} = 1.08$, eingezeichnet. Der Maximalwert der Betriebskurve unterschreitet den Wert $L_{\max}(\text{SAPL}=0)$ um 21.7 %.

Die CPU-Auslastungen (Abb. 8.15.) (Gesamt und nur TSO) gehen für $\text{SAPL} = 20$ stark auseinander. Obwohl eine Gesamtauslastung von 85 % erreicht wird, erhält TSO nur noch etwa 37 % CPU-Zeit. APL und System-Overhead zusammen vergrößern ihren Anteil auf über 40 %, während bei $u=0$ (kein Benchmark-Script) dieser Anteil nur 28 % beträgt ($\text{SAPL} = 20$).

Der TS-CPU-Overhead steigt ebenfalls mit zunehmender Script-Anzahl u an, jedoch nur von etwa 5 % ($u=3$) auf 13 % bei $u=21$ ($\text{SAPL} = 0$).

Die System-Antwortzeit $\text{SRT}(\text{SAPL};u)$ für die Benchmark-Versuche ist in Abbildung 8.16. dargestellt. Sie wird aus Formel (4.16.) nach Extrapolation von $L(u)$ und TS-CCL ermittelt.

Bemerkung: Zum Vergleich wurde das Experiment mit dem System S168 aus Kap. 8.2.2. (IBM/370-168 CPU, Systemkonfiguration 3) wiederholt. Abb. 8.17. zeigt die Systemleistungen $L(\text{System},\text{SAPL};u)$ für System = S158 und S168 gemeinsam für alle drei SAPL-Werte. Es ist zu sehen,

dass $L(S168, SAPL=10)$ nur sehr wenig kleiner ist als $L(S158, SAPL=0)$; mit dem System S168 und $SAPL=20$ lässt sich auch eine leicht verbesserte Systemleistung erbringen als mit dem System S158 und $SAPL=10$.

In Abbildung 8.18. ist die CPU-Auslastung (Gesamt und durch TSO) für das System S168 angegeben. Hier verlaufen die TS-Anteile fast gleich, offensichtlich wird der CPU-Verbrauch durch das Timesharing von SAPL nicht wesentlich beeinflusst.

Abbildung 8.14.: TS-Systemleistung als Funktion des APL-CPU-Quantums.

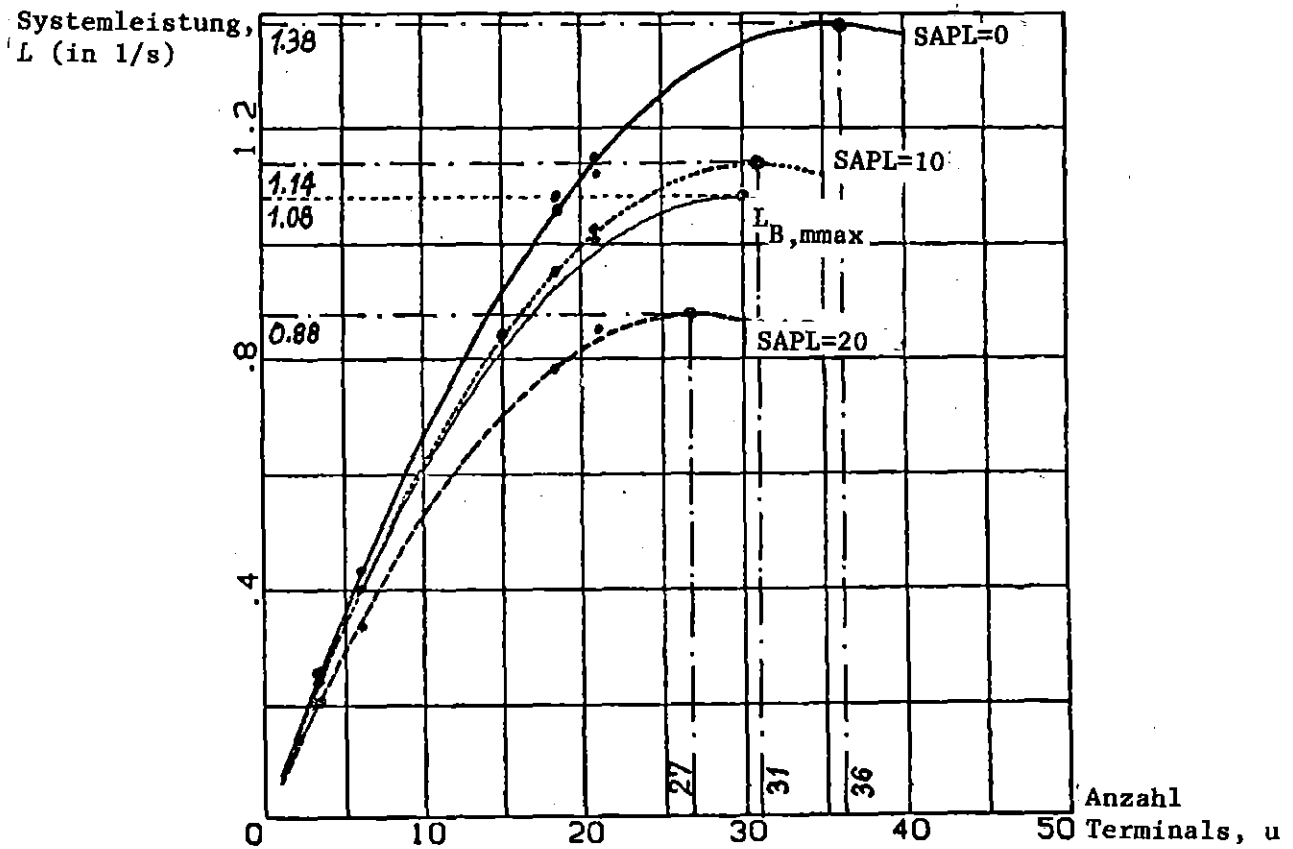


Abbildung 8.15.: CPU-Auslastungen als Funktion des APL-CPU-Quantums SAPL.

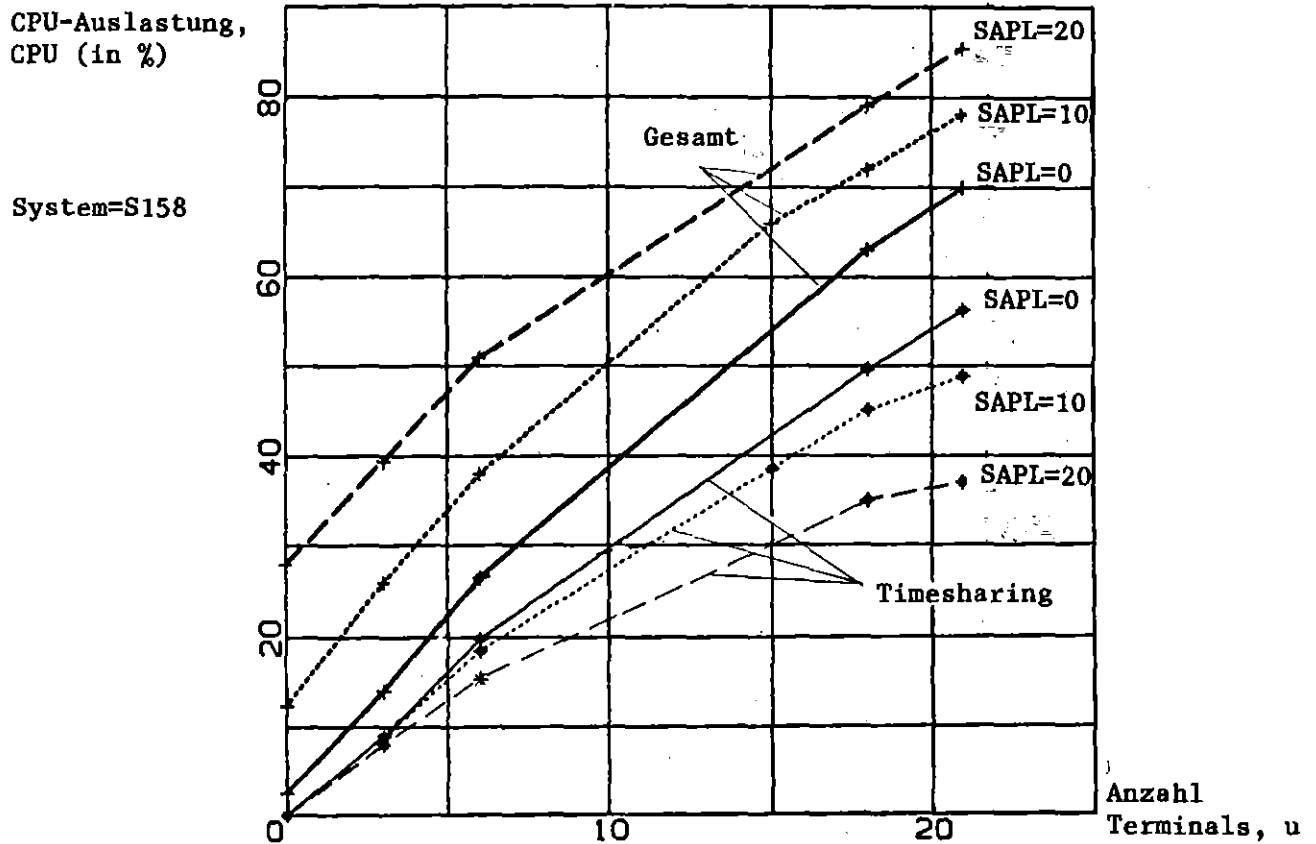


Abbildung 8.16.: System-Antwortzeiten SRT(SAPL;u), extrapoliert.

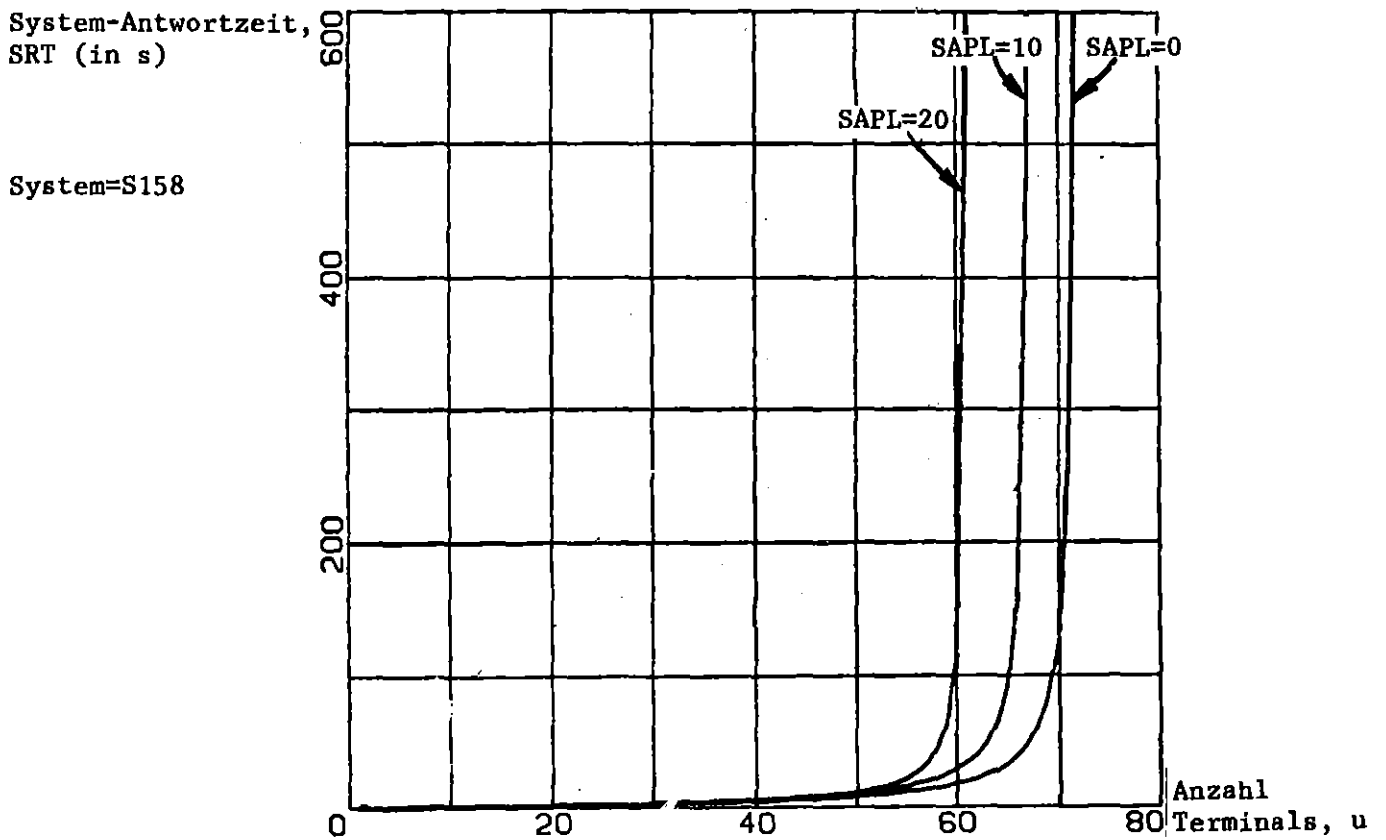


Abbildung 8.17.:

Die Systemleistungskurven
 $L(\text{System}, \text{SAPL}; u)$

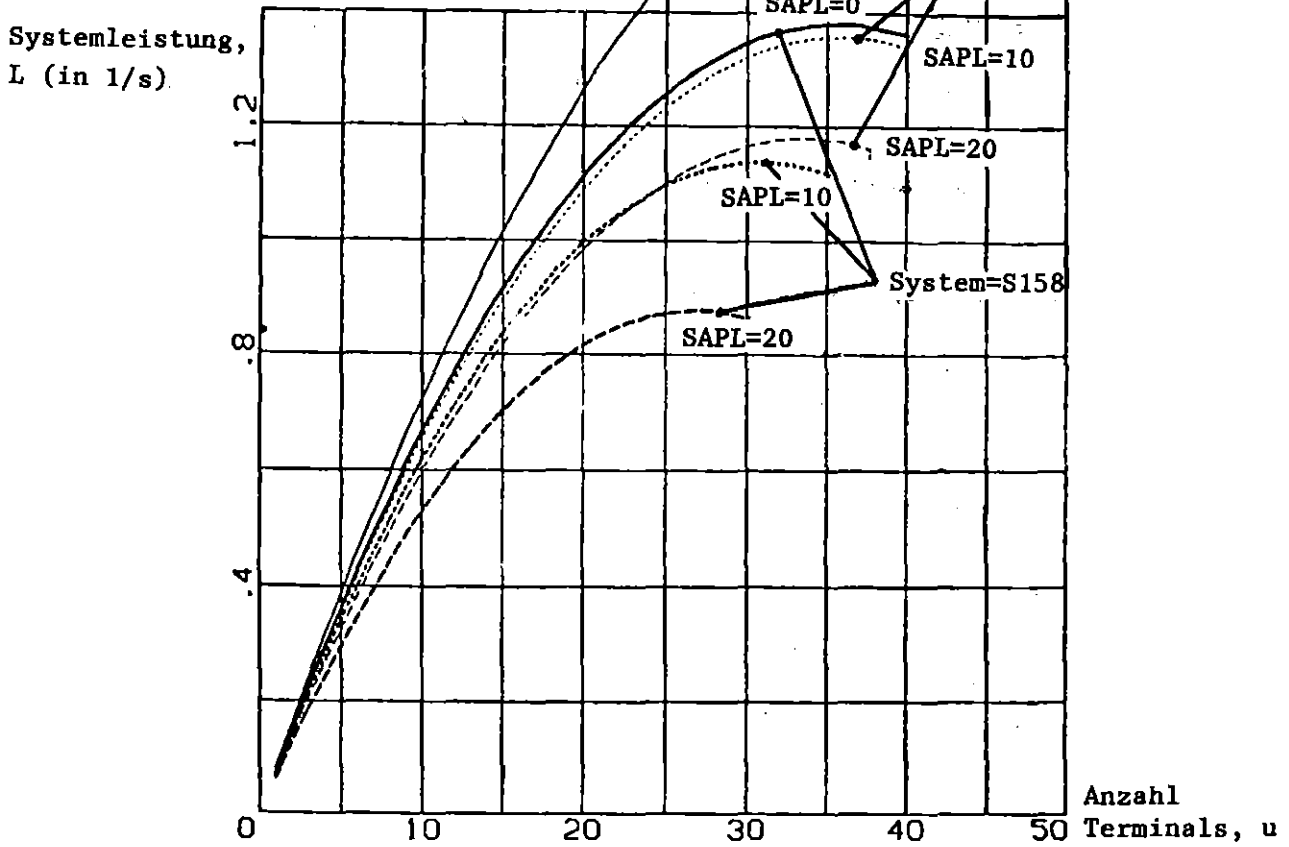
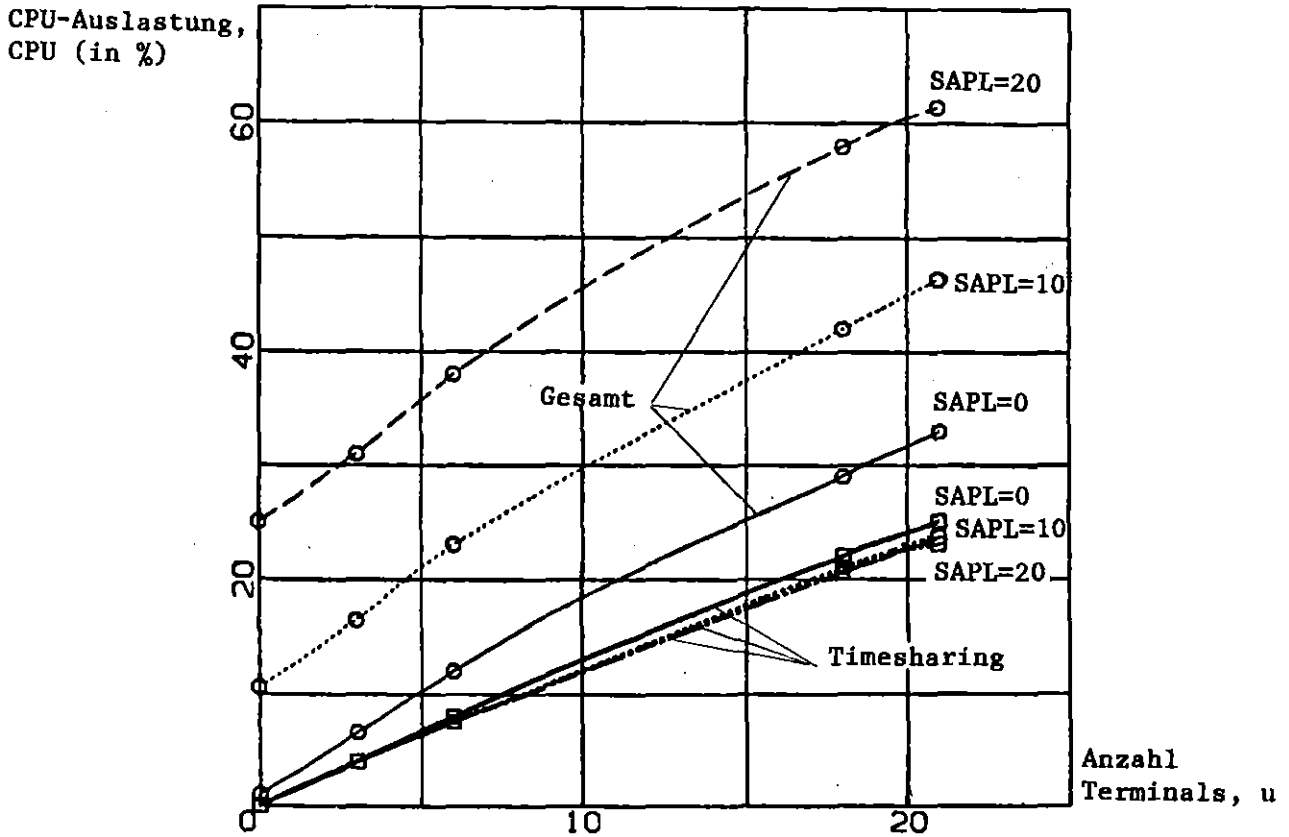


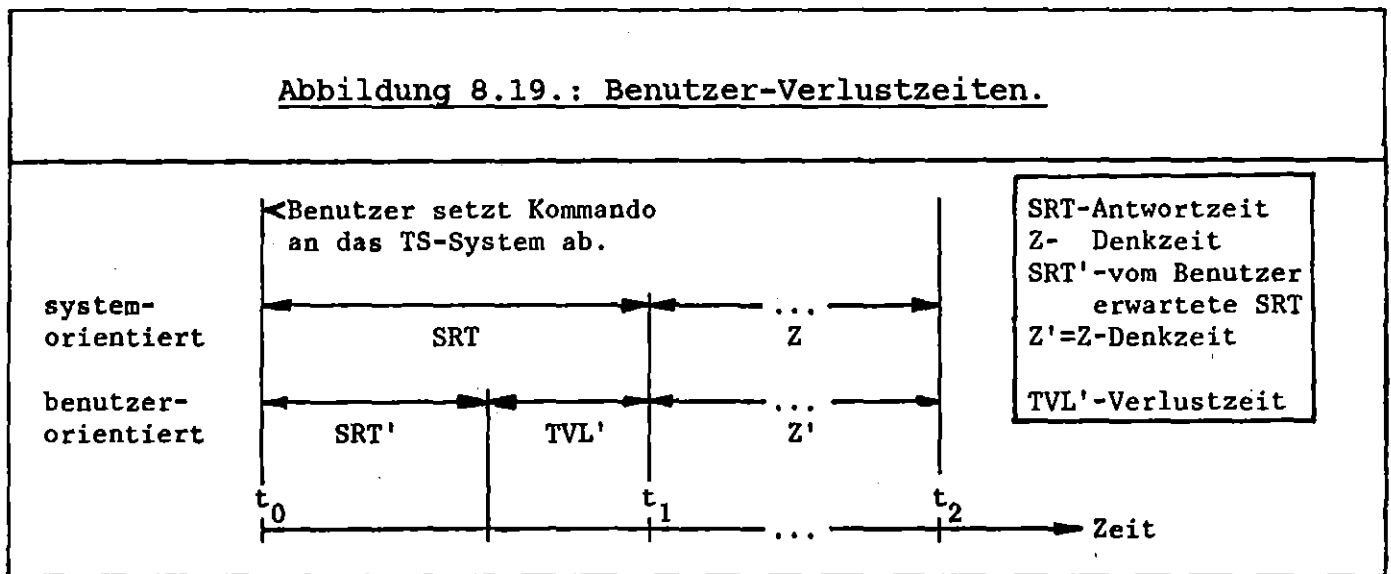
Abbildung 8.18.: CPU-Auslastungen, S168.



8.3. Benutzer-Verlustzeiten.

Zur Quantifizierung der Auswirkungen des Antwortzeitverhaltens des Timesharing-Systems TSO auf den Fortgang von stark DV-orientierten Forschungs- und Entwicklungsarbeiten wurden gemeinsam mit dem Rechenzentrum in Bereichen, die intensiv DV-Anwendungen betreiben, Verfahren entwickelt, welche die subjektiven Erfahrungen dieser Benutzer festhalten. "Die Anwendung diese quantitativen Ergebnisse soll eine Planungsgrundlage für den Mitteleinsatz im DV-Bereich darstellen²."

Die an dem Verfahren beteiligten Benutzer werden aufgefordert, Wege- und Wartezeiten zu den Terminals festzuhalten, ebenso die Zeiten, die nach Ansicht dieser Benutzer als Antwortzeiten auf ihre Kommando-eingaben nicht vertretbar sind und ihren Arbeitsrhythmus negativ beeinflussen (siehe ³). Diese Zeiten werden (Benutzer-) Verlustzeiten (TVL') genannt. Sie sind nicht identisch mit den System-Antwortzeiten SRT (siehe Abb. 8.19.).



² - ADI: Gemeinsame Studie: "Einfluss der DV-Kapazität auf das Erreichen von Zielen der F&E-Arbeiten im KfK", ADI/INR/IRE, vom 4.12.78.

³ - BOIES: User Behavior..., <BOIE74>, S. 16/17.

Die von den Anwendern erwartete System-Antwortzeit 'SRT' lässt sich nach Feststellung der Benutzer geplant überbrücken, z.B. durch Inspektion von Druckausgaben, Fehlerlisten, Planung und Entscheidung über die Fortsetzung ihres Dialogs am Terminal, während das Zeitintervall 'TVL' zu nicht-planbaren Unterbrechungen des Arbeitsflusses und damit zu Zeitverlusten für ihre Forschungs- und Entwicklungsarbeiten führen.

Die vorliegenden Daten entstammen zwei Vergleichsmessungen, zwischen denen sowohl die I/O-Peripherie als auch die Zentraleinheit verändert wurden (siehe Kap. 8.2.2.).

Messung 1: 9. - 15.12.78 (1 Woche, je 9 Stunden täglich, Wochenenden eingeschlossen),
14 beteiligte Mitarbeiter aus verschiedenen Anwendungsgebieten (alle mit naturwissenschaftl. oder technischer Ausbildung und mehrjähriger DV-Erfahrung) des KfK, mit 162 Sitzungen und 98 Stunden Sitzungsdauer ⁴,

System: IBM/370-158 mit IBM 2314 Plattenperipherie (Systemkonfiguration 2, S. 163).

Messung 2: 5. - 11.03.79 (1 Woche, je 9 Stunden täglich, Wochenenden eingeschlossen),
11 beteiligte Mitarbeiter aus verschiedenen Anwendungsgebieten des KfK, mit 91 Sitzungen und 51.5 Stunden Sitzungsdauer ⁵,

System: IBM/370-168 mit CDC 3380 Plattenperipherie (Systemkonfiguration 5, S. 164).

⁴ - MAREK, SCHLECHTENDAHL: Auswertung der TSO-Benutzerstatistik, IRE-6/496/1979 (KfK int.) vom 27.3.79.

⁵ - MAREK, SCHLECHTENDAHL: Auswertung der TSO-Benutzerstatistik, IRE-6/514/1979 (KfK int.) vom 4.7.79.

Die mit Hilfe von Schachuhren und Selbstaufschreibung vorgenommenen Aufzeichnungen der Verlustzeiten wurden einschliesslich der Wochenenden während einer vollen Woche durchgeführt; die Mittelwerte über alle 7 Wochentage sind in Abb. 8.20., die entsprechende Benutzer-Anschaltverteilung (ebenfalls über alle 7 Wochentage gemittelt) ist in Abb. 8.21. grafisch dargestellt. Die mittlere Benutzer-Anschaltverteilung wird für die Hochrechnung auf die Gesamt-Verlustzeit verwendet, obwohl aufgrund der verschiedenen DV-Erfahrung und unterschiedlicher Arbeitsgebiete anderer Benutzer hier abweichende Verlustzeiten erwartet werden könnten.

Den Wert für die Gesamt-Verlustzeit findet man aus:

$$V = \int_{t_0}^{t_{18}} TVL'(t) u(t) dt ,$$

bzw. aus der Summe:

$$V = \sum_{i=1}^{18} TVL'(T_i) u(T_i) (T_{i+1} - T_i) \quad (\#)$$

für die Messperiode 1:

$$\begin{aligned} V_1 &= 727 \quad \text{M min (Mann-Minuten),} \\ &= 12.12 \text{ Mh} \quad \text{oder 19.2\%,} \end{aligned}$$

(bezogen auf 63 erbrachte Mh) und für die Messperiode 2:

$$\begin{aligned} V_2 &= 150 \quad \text{M min ,} \\ &= 2.50 \text{ Mh} \quad \text{oder 4.0\%.} \end{aligned}$$

Nach Feststellung der Anwendungsabteilungen sind Werte der Verlustzeitfunktion TVL' von 20-25% oder 12-15 min pro Stunde gerade noch akzeptabel.

Abbildung 8.20.: Mittlere Verlustzeit je Benutzer (pro Messzeitint.)

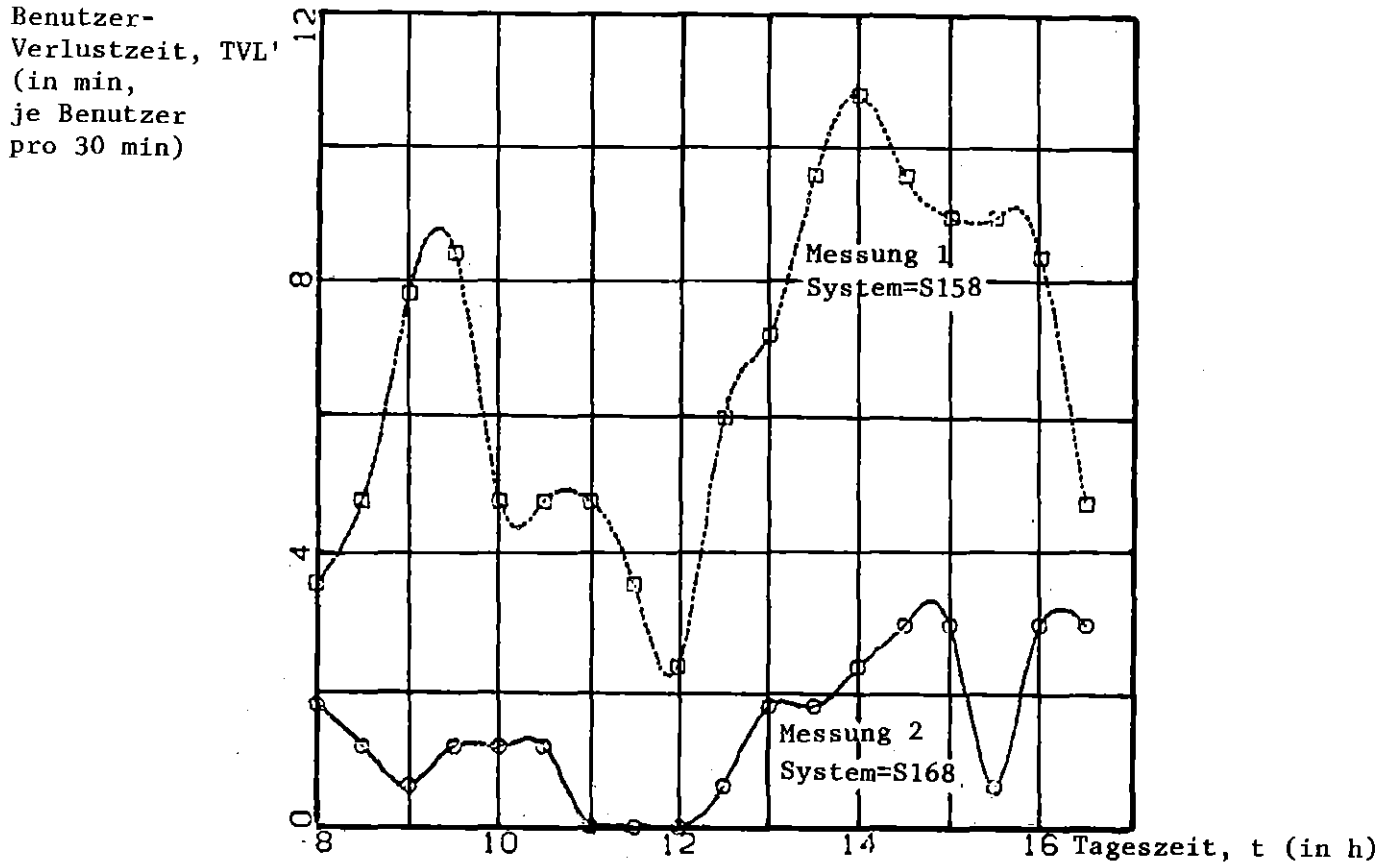
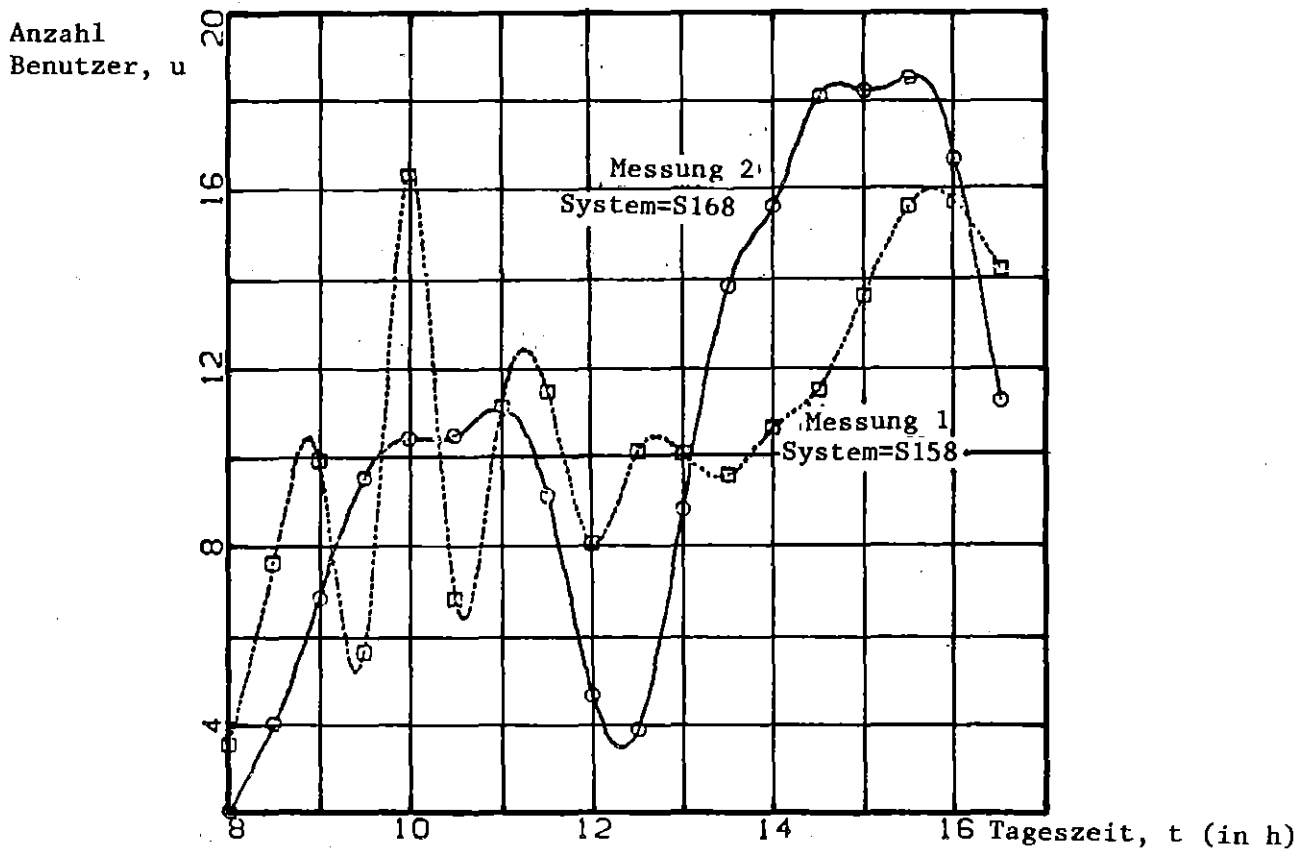


Abbildung 8.21.: Benutzerverteilung $u(t)$.



Die Differenz von V_1 und V_2 ergibt die Verlustzeit-Reduktion; sie ist ebenfalls ein Mass für die Leistungsverbesserung des Time-sharing-Systems, hier beträgt sie 11.62 Mh oder 15.2%.

Die Abbildung 8.22. zeigt die Verteilung über den Tag für die Verluste in Mann-Stunden. Die dargestellte Kurve entspricht den Summanden

$$V(T_i) = TVL'(T_i) u(T_i) (T_{i+1} - T_i)$$

in der Formel (#) mit $i = 1, \dots, 18$ ($T_1 = 8$ Uhr und $T_{19} = 16.30$ Uhr).

Die Wahrscheinlichkeit, dass die anzutreffende Verlustzeit nicht grösser als TVL' ist, wird in der Abbildung 8.23 dargestellt. P_1 gibt die Kurve für die Messung 1, P_2 für die Messung 2 wieder. Zugrunde liegen die gleichen Messdaten wie in Abbildung 8.20.. P_1 sagt u.a. aus, dass die Wahrscheinlichkeit, bei der neuen Konfiguration Verlustzeiten von weniger als 4 Minuten (pro Stunde Anschaltzeit) anzutreffen, fast schon 1 ist. Dagegen war bei der früheren Konfiguration mit einer Verlustzeit von 6 Minuten noch nicht der Median der Wahrscheinlichkeitsverteilung erreicht, m.a.W. die Wahrscheinlichkeit, Verlustzeiten von über 6 Minuten (pro Stunde) oder 20% vorzufinden, betrug 0.5.

Weder auf einen Vergleich der Kosten, die durch die Systemänderungen entstanden sind, mit den durch die Verlustzeit-Reduktion eingesparten, noch auf die Relevanz des früheren Erreichens der (der Anwenderabteilung) vorgegebenen Ziele kann hier näher eingegangen werden, da dies eine unternehmenspolitische Aufgabe ist. Die der Verlustzeit-Rechnung entsprechenden systemseitige Beurteilung der Veränderungen wurde bereits in Kap. 8.2.2. gegeben.

Abbildung 8.22.: Gesamt-Verlust.

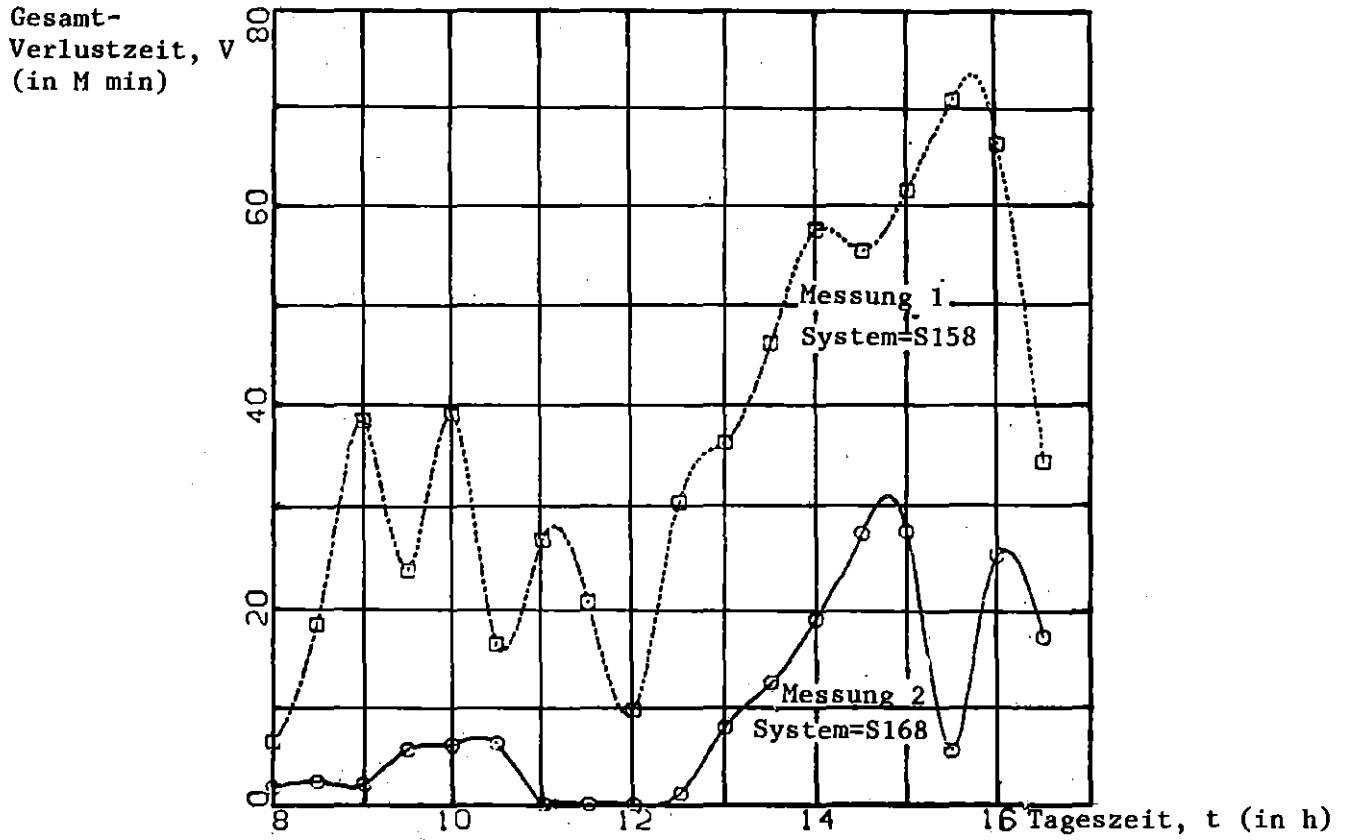
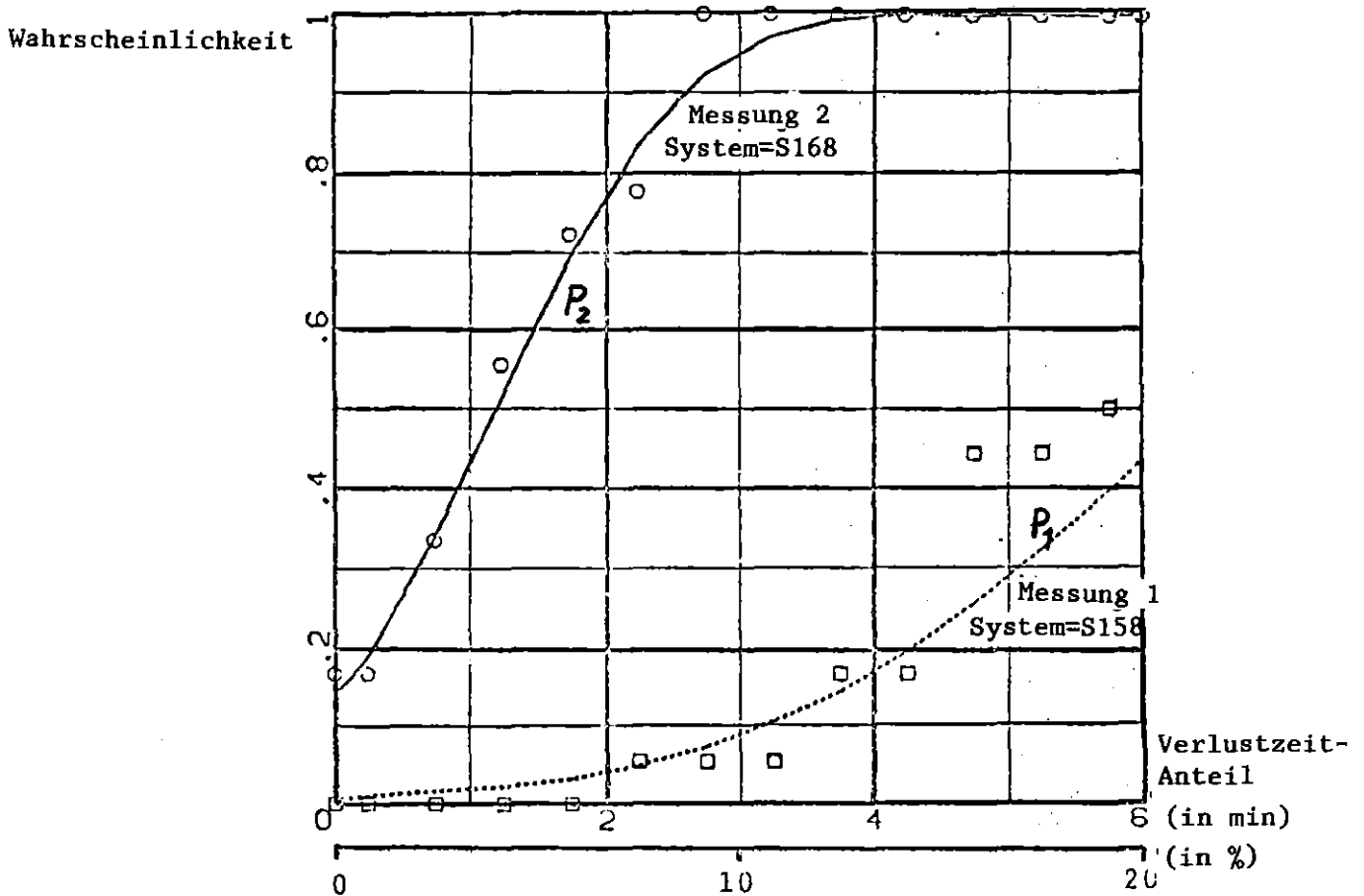


Abbildung 8.23.: Verlust-Summenverteilung.



9. Anhang.

9.1. Systemkonfigurationen.

Dieser Abschnitt gibt einen Überblick über die in der vorliegenden Arbeit angeführten Systemkonfigurationen.

Systemkonfiguration 1:

A. Hardware:

Zentraleinheit (CPU) : IBM/370-158 ¹, 115 ns Zykluszeit,
Pufferspeicher: 8 KB,
Hauptspeicher: 3 MB, 690 - 1035 ns Zykluszeit,
Datenkanäle: 4 BMPX, 1 MPX,
Datenträger: System: BASF 6230 *),
Paging/Swapping: IBM 2305 und BASF 6230,
temp. Benutzerdateien: BASF 6230,
perm. Benutzerdateien: IBM 2314,
Anzahl der installierten Terminals: 71;

B. Software:

Basis-Betriebssystem: OS/VS2 (MVS), Release 3.7D ²,
Komponenten: JES/3, TCAM, TSO, APL;

Systemkonfiguration 2:

A. Hardware:

Zentraleinheit (CPU) : IBM/370-158,
Pufferspeicher: 8 KB,
Hauptspeicher: 3 MB,
Datenkanäle: 4 BMPX, 1 MPX,
Datenträger: System: BASF 6230,
Paging/Swapping: IBM 2305 und BASF 6230,
temp. Benutzerdateien: BASF 6230,
perm. Benutzerdateien: CDC 33302,
Anzahl der installierten Terminals: 76;

B. Software:

Basis-Betriebssystem: OS/VS2 (MVS), Release 3.7D,
Komponenten: JES/3, TCAM, TSO, APL;

*) Mittlere Zugriffszeiten der Plattenspeicher:
IBM 2314: 70 ms, IBM 3330, BASF 6230, CDC 33302: 30 ms,
CDC 33801: 18 ms, IBM 2305 (Festkopf): 5 ms.

¹ - IBM: A Guide ... Model 158, <IBMO74.1>, S. 22.

² - IBM: System/370 Summary, <IBMO76.2>, Sect. 4, S. 4-21 ff.

Systemkonfiguration 3:

A. Hardware:

Zentraleinheit (CPU) : IBM/370-168/1³, 80 ns Zykluszeit,
Pufferspeicher: 16 KB,
Hauptspeicher: 4 MB,
Datenkanäle: 3 BMPX, 1 MPX, 3 Selektor-Kanäle,
Datenträger: System: BASF 6230,
Paging/Swapping: IBM 2305 und BASF 6230,
temp. Benutzerdateien: BASF 6230,
perm. Benutzerdateien: CDC 33302,
Anzahl der installierten Terminals: 76;

B. Software:

Basis-Betriebssystem: OS/VS2 (MVS), Release 3.7G,
Komponenten: JES/3, TCAM, TSO, APL;

Systemkonfiguration 4:

A. Hardware:

Zentraleinheit (CPU) : IBM/370-168/1,
Pufferspeicher: 16 KB,
Hauptspeicher: 4 MB,
Datenkanäle: 4 BMPX, 1 MPX, 2 Selektor-Kanäle,
Datenträger: System: BASF 6230,
Paging/Swapping: IBM 2305 und BASF 6230,
temp. Benutzerdateien: BASF 6230,
perm. Benutzerdateien: CDC 33302,
Anzahl der installierten Terminals: 76;

B. Software:

Basis-Betriebssystem: OS/VS2 (MVS), Release 3.7G,
Komponenten: JES/3, TCAM, TSO, APL;

Systemkonfiguration 5:

A. Hardware:

Zentraleinheit (CPU) : IBM/370-168/1,
Pufferspeicher: 16 KB,
Hauptspeicher: 4 MB,
Datenkanäle: 4 BMPX, 1 MPX, 2 Selektor-Kanäle,
Datenträger: System: CDC 33302,
Paging/Swapping: IBM 2305 und CDC 33801,
temp. Benutzerdateien: CDC 33801,
perm. Benutzerdateien: CDC 33801,
Anzahl der installierten Terminals: 92;

B. Software:

Basis-Betriebssystem: OS/VS2 (MVS), Release 3.7G,
Komponenten: JES/3, TCAM, TSO, APL;

³ - IBM: A Guide ... Model 168, <IBMO74.2>, S. 22 ff.

9.2. Die Leistungsdatenbank.

Die Leistungsdatenbank (LDB) enthält Aufzeichnungen der leistungsrelevanten Daten (Kontrollvariablen-Werte) des Timesharing-Systems, die während des Normalbetriebes von dem betriebsbegleitend mitlaufenden Software-Monitor Resource Measurement Facility (RMF) als Mittelwerte über 15 Minuten-Intervalle an die System Management Facility (SMF) gegeben werden und von dieser gemeinsam mit den Accounting-Daten aufgezeichnet werden.

Die Leistungsdaten werden mindestens einmal monatlich von den Accounting-Daten getrennt und chronologisch sortiert in der LDB abgespeichert. Die LDB wird jeden Monat um sechs Dateien, der Monats-LDB, erweitert. Die Grösse einer Monats-LDB (6 Dateien) beträgt maximal etwa 2.4 Megabytes.

Übersicht über den Zustand der LDB

		Datei 1	Datei 2	Datei 3	Datei 4	Datei 5	Datei 6
	N a m e	PRIMAR	TASKST	STRPAG	SUSTAT	CHANNL	DEVICE
Monat 1	Dez. 77
Monat 2	Jan. 78
...
Monat 13	Dez. 78
Monat 14	Jan. 79
...
Monat 25	Dez. 79

Die Monats-LDB besteht physisch aus sechs Dateien: PRIMAR, TASKST, STRPAG, SUSTAT, CHANNL und DEVICE.

PRIMAR: Diese Datei enthält neben der Auslastung der Zentraleinheit Auszüge aus den folgenden Dateien. Sie enthält a l l e in Kapitel 2. definierten primären Kontrollvariablen.

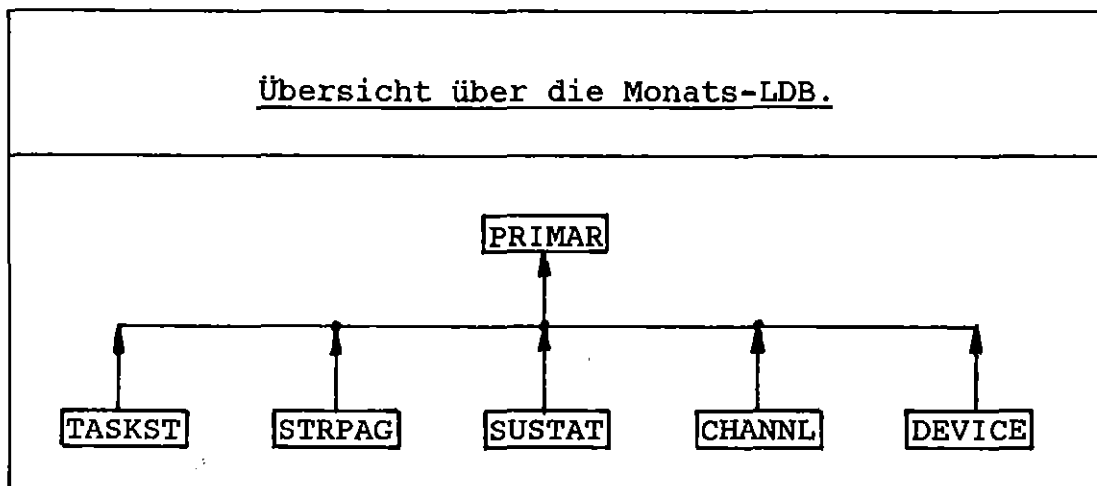
TASKST: Diese Datei enthält die Verteilung der Anzahlen der TS-Benutzer im Aufzeichnungsintervall (15 Minuten) und anderer im System laufender Aufträge (Tast Statistik).

STRPAG: Diese Datei enthält die Auslastung des Haupt- und Hilfsspeichers sowie die Paging-Raten, aufgegliedert nach den verschiedenen Arten des Paging.

SUSTAT: Diese Datei enthält die Verteilung der Service Units durch den SRM des MVS auf die verschiedenen Auftragsgruppen.

CHANNL: Diese Datei enthält die an den Datenkanälen gemessenen I/O-Raten und Belegzeiten.

DEVICE: Diese Datei enthält - für 5 ausgewählte Datenträger - Namen, Adresse, I/O-Raten, Kontrolleinheiten- und Geräte-Belegzeiten, sowie die Anzahl der wartenden Anforderungen (queued requests).



Jeder Eintrag in einer Datei der LDB repräsentiert ein 15-Minuten-Intervall des Normalbetriebes. Er wird mit einem Kopfeintrag, der gleichzeitig das Ordnungskriterium ist (Datum, Uhrzeit), als Satz in die entsprechende Datei beschrieben.

Die Dateien PRIMAR, TASKST, STRPAG, SUSTAT und CHANNL besitzen Sätze der Länge 120 Bytes, Blöcke der Länge 3000 Bytes; die Datei DEVICE hat eine Satzlänge von 245 Bytes und eine Blockgröße von 3185 Bytes. Alle Dateien sind fest geblockt.

Für diese Arbeit wurden nur Werte aus der Datei PRIMAR herangezogen; die folgende Abbildung gibt eine Detaildarstellung eines Satzes mit Format und Bedeutung der einzelnen Datenfelder.

Spalte von bis	Feldname	Format	Bedeutung (DSNAME=PRIMAR) (LRECL=120, BLKSIZE=3000, RECFM=FB)
1 8	STRT_DAT	A(8)	DD.MM.YY - Start-Datum
9 18	STRT_TME	P'Z999999'	_HHMMSSTTT - Start-Zeit.
19 23	INTERVAL	F(5,0)	Dauer der Messperiode (s).
24 30	CPU_BUSY_SEC_PERC	F(7,2)	mittl. Gesamt-CPU-Auslastung in %.
31 37	RYL	F(7,2)	mittl. IN-READY Level.
38 44	MPL	F(7,2)	mittl. Multiprogramming Level.
45 51	TSO_USERS	F(7,2)	mittl. Anz. TSO-Tasks (Benutzer).
52 58	P_RATES(5)	F(7,2)	mittl. Gesamt-Paging-Rate (P/s).
59 65	FRAMES	F(7,2)	mittl. Realspeicherauslastung in %.
66 72	SLOTS	F(7,2)	mittl. Hilfsspeicherauslastung (%).
73 78	RATE_TRX	F(6,2)	mittl. TRX-Rate (1/s).
79 86	R_TIME	F(8,2)	mittl. Antwortzeit (s).
87 94	TSO_CPU	F(8,2)	mittl. CPU-Verbrauch durch TSO in %
95 100	TSO_CCL	F(6,2)	mittl. TSO-Aktivitätsgrad.
101 108	SUM_SIOS	F(8,2)	mittl. Gesamt-I/O-Rate (1/s).
109 116	TSO_SIOS	F(8,2)	mittl. TSO-I/O-Rate (1/s).

9.3. Kommandos der Basis-Scripts.

Die Kommandos der in den in Kap. 6.3. definierten Basis-Scripts sollen hier zusammen mit ihrer Funktionsklassenzugehörigkeit angegeben werden. Die in Kap. 6.3. eingeführte Zerlegung in Eröffnung, Vorbereitung, Kern und Abschluss bleibt erhalten. Die Kerne, die, durch die Parameter n_T (im T-Script), n_I (im I-Script) und n_C (im C-Script) gesteuert, wiederholt werden, sind vollständig abgedruckt.

Basis-Scripts, Eröffnungs- und Abschlussphase (für alle Scripts einheitlich):

Phase	Kommando	Anzahl	Funktionsklasse
Eröffnung	ALLOCATE	2	I/O
	FREE	1	I/O
	SENDMESS	1	I/O
	SEND	1	I/O
	TIME	1	trans.
	TTIME	1	I/O
	USER	1	trans.
	VTIME	1	I/O
Abschluss	ALLOCATE	1	I/O
	END	1	trans.
	FREE	1	I/O
	OLDPROC	1	I/O
	PRMW	1	I/O
	SENDMESS	1	I/O
	SEND	1	I/O
	TIME	2	trans.
	TTIME	1	I/O
	USER	1	trans.
	VTIME	1	I/O

Basis-Scripts, Vorbereitungsphase, nach Scripttyp getrennt:

Phase	Kommando	Anzahl	Funktionsklasse
Vorbereitung T-Script	ALLOCATE	2	I/O
	COPY	1	I/O
	DELETE	1	I/O
	TIME	1	trans.
	TTIME	1	I/O
	VTIME	1	I/O
Vorbereitung I-Script	ALLCTERM	1	trans.
	ALLOCATE	3	I/O
	ATTRIB	1	trans.
	CALL	2	CPU
	DELETE	1	I/O
	FREETERM	1	trans.
	TIME	3	trans.
	TTIME	2	I/O
	VTIME	2	I/O
Vorbereitung C-Script	ALLOCATE	1	I/O
	FREE	1	I/O
	TIME	1	trans.
	TTIME	1	I/O
	VTIME	1	I/O

Transaktionsorientiertes Basis-Script (T-Script), Kern:

Phase	Kommando	Anzahl	Funktionsklasse
Kern	DOWN	t_T	trans.
	EDIT	1	I/O
	END	1	trans.
	TDF	1	I/O
	TOP	1	trans.
	TTIME*)	2	I/O
	UP	t_T	trans.
	VERIFY	2	trans.
	VTIME	2	I/O

Es folgt eine vollständige Liste des T-Kerns:

```

/*****
/*
/*          START BM KERNEL
/*          TRANSAKTION DRIVER
/*****

AGAIN: SET CTIME=&SYSTIME.          /* MAJOR LOOP
/* CONTROLLED BY PARM NT
VTIME >EDT &TEST. MEAN(&Z.)        /* WRITE ELAPSED TIME, STATUS
PUTFILE DT
SET A=&A.+1
EDIT AAAA.DATA NONUM              /* ENTER EDIT D.S.
VERIFY OFF                         /* PUT ECHO FLAG OFF
TOP                                 /* MOVE L.C. TO TOP OF D.S.
SET B=0

AGIN: SET B=&B.+1                   /* MINOR LOOP, TRIV. CMDS.
/* CONTROLLED BY PARM TT
DOWN 1                             /* MOVE L.C. ONE DOWN
UP 1                                /* MOVE L.C. ONE UP
IF &B. LT &TT. THEN GOTO AGIN

VERIFY ON                           /* PUT ECHO FLAG ON
END NOSAVE                          /* END EDIT - NO SAVE
VTIME EDT< &TEST. MEAN(&Z.)        /* WRITE ELAPSED TIME, STATUS
PUTFILE DT

TDF &SYSTIME. &CTIME.               /* CALC. DELTA TIME
SET TTIME=&TTIME.+&SDIFF.
WRITENR &SDIFF.,
IF &A. LT &NT. THEN GOTO AGAIN

/*****
/*
/*          END BM KERNEL
/*****

```

*) TTIME erscheint nicht in der nachfolgenden Liste, da es von VTIME aufgerufen wird.

I/O-orientiertes Basis-Script (I-Script), Kern:

Phase	Kommando	Anzahl	Funktionsklasse
Kern	COPY	1	I/O
	TDF	1	I/O
	TTIME *)	2	I/O
	VTIME	2	I/O

Es folgt eine vollständige Liste des I-Kerns:

```

/*****
/*          START BM KERNEL          */
/*          I/O - DRIVER              */
/*****

AGAIN: SET A = &A. + 1                /* MAJOR LOOP          */
                                           /* CONTROLLED BY PARM NI . */
VTIME >CPY &TEST. MEAN(&Z.)          /*WRITE ELAPSED TIME, STATUS*/
PUTFILE DT
SET C1TIME = &SYSTIME

COPY &DSN2. &DSN3. NONUM             /* COPY SECOND. D.S.INTO TEMP.* /
                                           /* SIZE AND PHYS. ATTR. DET.  */
                                           /* DURING PREP. STAGE II.    */
                                           /* SIZE ACCORD. TO PARM II . */

SET C2TIME = &SYSTIME
VTIME CPY< &TEST. MEAN(&Z.)          /* WRITE ELAPSED TIME, STATUS */
PUTFILE DT
TDF &C2TIME. &C1TIME.                /* CALC. DELTA TIME      */
SET TTIME = &TTIME. + &SDIFF.
WRITENR &SDIFF.,
IF &A. LT &NI. THEN GOTO AGAIN

/*****
/*          END   BM KERNEL          */
/*****

```

*) TTIME erscheint nicht in der nachfolgenden Liste, da es von VTIME aufgerufen wird.

CPU-orientiertes Basis-Script (C-Script), Kern:

Phase	Kommando	Anzahl	Funktionsklasse
Kern	RUN/CALL	1	CPU
	STIME	1	I/O
	TDF	1	I/O
	TTIME *)	1	I/O
	VTIME	1	I/O

Es folgt eine vollständige Liste des C-Kerns:

```

/*****/
/*          START BM KERNEL          */
/*          CPU - DRIVER              */
/*****/

AGAIN: SET CTIME=&SYSTIME.           /* MAJOR LOOP          */
/* CONTROLLED BY PARM NC .          */
VTIME >RUN &TEST. MEAN(&Z.)         /* WRITE ELAPSED TIME, STATUS */
PUTFILE DT
SET A=&A.+1

IF &RUN. = RUN THEN RUN 'TSO557.SRU.PLI(BM16)' '/&CC.' /* RUN */
ELSE CALL 'TSO557.SRU.LOAD(BM16)' '/&CC.' /* CALL */

/* CPU TIME CONTR. BY PARM CC */
VTIME RUN< &TEST. MEAN(&Z.)         /* WRITE ELAPSED TIME, STATUS */
PUTFILE DT
TDF &SYSTIME. &CTIME.               /* CALC. DELTA TIME    */
SET TTIME=&TTIME.+&SDIFF.
WRITENR &SDIFF.,
IF &A. < &NC. THEN GOTO AGAIN

/*****/
/*          END BM KERNEL          */
/*****/

```

*) TTIME erscheint nicht in der nachfolgenden Liste, da es von VTIME aufgerufen wird.

9.4. Abkürzungsverzeichnis.

Im folgenden findet sich ein alphabetische geordnetes Verzeichnis aller in der vorliegenden Arbeit verwendeten Abkürzungen.

APL	A Programming Language, Programmiersprache u. -system,
B	Servicerate in 1/s,
BF	Blockungsfaktor,
BM	Benchmark,
BMPX	Block Multiplexor Channel, Block-Multiplex-Kanal,
BS	Blocklänge (block size) der COPY-Datei im I-Script,
C	CPU-orientiertes Basis-Script,
c_C	Schleifenparameter für C-Script,
CCL	Concurrency-Level,
CMS	Conversational Monitor System (des IBM VM/370),
C_P	Kennzahl freie Kapazität, basierend auf $P_R(1)$,
CPU	Central Processing Unit, Zentraleinheit, in der Arbeit jedoch auch zur Bezeichnung der Kontrollvariablen CPU-Zeitverbrauch in % (Gesamt), verwendet,
CRT	Command Recognition Time, Kommando-Erkennungszeit,
C_S	Kennzahl freie Kapazität, basierend auf $S_R(1)$,
CYT	Transaktionszykluszeit in s,
D	mittlere Denkzeit (Warteschlangenmodelle),
DB/DC	Data Base/Data Communication,
DBM	dynamischer Benchmark,
DS	Grösse der COPY-Datei im I-Script,
DVA	Datenverarbeitungsanlage,
GTF	General Trace Facility, Software-Monitor,
F	Produktivitätsgewinn in Ms (Mann-Sekunden),
$f_k(u)$, $f(u)$	Gewinnfunktion,
$H_n(T, I, C)$	symmetrisches Script-Polynom n-ten Grades, enthält Basis-Scripts T, I und C,
I	I/O-orientiertes Basis-Script,
i_I	Grösse der I-Datei,
IMS	Information Management System,
INT	Input Time, Eingabezeit in s,

JES	Job Entry System (des OS/VS2 MVS),
KV	Kontrollvariable,
L	Systemleistung in 1/s,
l	relative Systemleistung,
L_B	Betriebs-Systemleistung (in 1/s) aus der LDB,
$L_{B,amax}$	absolute maximale Betriebs-Systemleistung in 1/s,
$L_{B,mmax}$	mittlere maximale Betriebs-Systemleistung in 1/s,
LDB	Leistungsdatenbank,
L_{max}	maximale Systemleistung, Leistungsvermögen des TSS,
M	Summe aller Transaktionen während T_M ,
MIPS	'million instructions per second',
MPL	Multiprogramming-Level,
MPX	Multiplexor-Channel, Multiplex-Kanal,
MVS	Multiple Virtual Storage (IBM OS/VS 2),
n_C	Durchlaufzahl Kern des C-Scripts,
n_I	Durchlaufzahl Kern des I-Scripts,
n_T	Durchlaufzahl Kern des T-Scripts,
OS/VS	Operating System/Virtual Storage (IBM),
OUT	Output Time, Ausgabezeit in s,
$P(KV)$	Normalverteilungsapproximation von $S(KV)$,
PBM	Pseudo-Benchmark,
$P_n(T, I, C)$	Script-Polynom n-ten Grades, enthält Basis-Scripts T, I und C,
$P_R(KV)$	Normalverteilungsapproximation von $S_R(KV)$,
Q	Leistungsquotient zweier TSS,
q	relativer Leistungsquotient zweier TSS,
Q_B	Betriebs-Leistungsquotient zweier TSS,
QBM	Quasi-Benchmark,
RMF	Resource Measurement Facility, Software-Monitor,
RS	Satzlänge (record size) der COPY-Datei im I-Script,
RST	Restore Time, Aktivierungszeit in s,
RYL	Ready-Level,
$S(KV)$	Summenhäufigkeit der Kontrollvariablen 'KV', bezogen auf 20 äquidistante Klassen zwischen 0 und dem Maximalwert KV_{max} ,
$S_R(KV)$	relative Summenhäufigkeit, sonst wie $S(KV)$,

SP(r)	r-tes Script-Produkt (bei der Erzeugung eines Script-Polynoms),
SG	Servicegrad,
SIO	I/O-Rate in 1/s (Gesamt),
SMF	System Management Facility,
$S_n(T, I, C)$	homogenes Script-Produkt n-ten Grades, enthält Basis-Scripts T, I und C,
SRM	System Resource Manager (des IBM OS/VS2 MVS),
SRT	System-Antwortzeit des TSS in s,
T	Transaktionsorientiertes Basis-Script,
T_A	Aktivzeit in s,
T_D	Denkzeit in s,
TCAM	Telecommunication Access Method, Message-Handler für TSO,
T_M	Messzeit in s,
TRX	Transaktionsrate des TSS in 1/s,
TS	Timesharing,
T_S	mittlere Script-Laufzeit in s,
T'_S	mittlere Bedienzeit (Warteschlangenmodelle),
TS-CCL	TS-Concurrency-Level,
TS-CPU	CPU-Zeitverbrauch in % (TSS),
TS-IO	I/O-Rate in 1/s (TSS),
TSO	Time Sharing Option (des IBM OS/VS),
TSS	Timesharing System,
t_T	Durchlaufzahl Editor-Unterkommandos, T-Script,
T_V	Script-Verweilzeit in s,
TVL'	Verlustzeit (in min),
u	Anzahl der aktiven Terminals,
URT	User Response Time, Benutzer-Antwortzeit in s,
UTT	User Think Time, Denkzeit in s,
V	(Gesamt-) Benutzer-Verlustzeit (in Mh),
VM/370	Virtual Machine Facility/370 (IBM)
WL	Workload,
Z	Denkzeit, Parameter der Basis-Scripts T, I und C,

9.5. Formelsammlung.

Aktivzeit:

$$T_A = \sum_{i=1}^u \sum_{j=1}^{m_i} T_{ij} \quad (4.1)$$

TS-Concurrency Level:

$$\text{TS-CCL} = \frac{\text{Aktivzeit}}{\text{Messzeit}} = \frac{T_A}{T_M} \quad (4.2.)$$

$$T_A = T_M \cdot \text{TS-CCL} \quad (4.2.1.)$$

Anzahl der Transaktionen in der Messzeit:

$$M = \sum_{i=1}^u m_i \quad (4.2.2.)$$

Transaktionsrate:

$$\text{TRX} = M / T_M \quad (4.3.)$$

$$T_M = \sum_{j=1}^{m_i} T_{ij} + D_i \quad (4.4.)$$

$$(u - \text{TS-CCL})T_M = T_D \quad (4.5.)$$

Gesamtverweilzeit, Benchmark:

$$T_V = \sum_{i=1}^u T_i \quad (4.6.)$$

Mittlere Verweilzeit (eines Scripts):

$$T_S = T_V / u = (1/u) \cdot \sum_{i=1}^u T_i \quad (4.7.)$$

Systemleistung:

$$L = M / T_S \quad (4.8.)$$

$$L = Mu / T_V \quad (4.9.)$$

Relative Systemleistung:

$$l = L / L_{\max} \quad (4.10.)$$

Kennzahl der freien Kapazität:

$$C_S = \sum_{i=1}^N s_R(l_i) \Delta l_i \quad (4.10.1)$$

$$C_P = \sum_{i=1}^N p_R(l_i) \Delta l_i \quad (4.10.2)$$

Leistungsquotient:

$$Q = L_{\max,2} / L_{\max,1} \quad (4.11.)$$

$$Q = \frac{T_{v,1} \cdot M_2 \cdot u_2}{T_{v,2} \cdot M_1 \cdot u_1} \quad (4.12.)$$

$$Q(u) = L_2(u) / L_1(u) = T_{v,1} / T_{v,2} \quad (4.13.)$$

Relativer Leistungsquotient:

$$q = l_2/l_1 = (1/Q) \cdot L_2/L_1 \quad (4.14.)$$

Mittlere Antwortzeit:

$$SRT = \frac{\text{Aktivzeit}}{\text{Anzahl der Transaktionen}} = \frac{T_A}{M} \quad (4.15.)$$

$$TS-CCL = SRT \cdot L \quad (4.16.)$$

Servicerate:

$$B = 1 / SRT \quad (4.17.)$$

$$B = \frac{M}{T_A} = \frac{\text{Anzahl der Transaktionen}}{\text{Aktivzeit}} \quad (4.18.)$$

$$B = L / TS-CCL \quad (4.19.)$$

Gewinnfunktion:

$$f(u_i) = \left(\frac{SRT_1(u_i) \cdot p_1(u_i)}{T_1} - \frac{SRT_2(u_i) \cdot p_2(u_i)}{T_2} \right) u_i \quad (4.20.)$$

$$f_k(u_i) = \left(\frac{SRT_{k,1}(u_i) \cdot p_1(u_i) \cdot r_{k,1}}{T_1} - \frac{SRT_{k,2}(u_i) \cdot p_2(u_i) \cdot r_{k,2}}{T_2} \right) u_i \quad (4.22.)$$

Komposition der funktionsklassen-spezifischen Gewinne:

$$f(u_i) = \sum_{k=1}^k \max f_k(u_i), \quad (4.25.)$$

Gesamt-Produktivitätsgewinn:

$$F = \sum_{u_i=1}^u \max f(u_i) \quad (4.21.)$$

$$F_k = \sum_{u_i=1}^u \max f_k(u_i) \quad (4.26.)$$

Komposition des Gesamt-Produktivitätsgewinns:

$$F = \sum_{k=1}^k \max F_k = \sum_{k=1}^k \max \sum_{u_i=1}^u \max f_k(u_i) \quad (4.27.)$$

Gesamt-Benutzerverlustzeit:

$$V = \sum_{i=1}^n TVL'(T_i) \cdot u(T_i) \cdot (T_{i+1} - T_i)$$

10. Literaturverzeichnis.

- <AGRA76> A. K. AGRAWALA, J. M. MOHR, R. M. BRYANT: An Approach to the Workload Characterization Problem, IEEE Computer, June 1976, pp. 18-32.
- <AGRA78> A. K. AGRAWALA, J. M. MOHR: Predicting the Workload of a Computer System, AFIPS NCC, 1978, pp. 465-471.
- <ALEX76> R. ALEXANDER, B. MERTENS, R. D. RUMLER: A Timesharing Benchmark on an IBM/370-168, KFA Jülich, Rep.-Nr. 1280, März 1976.
- <ANDE73> M. R. ANDERBERG: Cluster Analysis for Applications, Academic Press, 1973.
- <ANTO79> R. ANTONCZYK, R. PIETSCHMANN, D. SCHRIEFER: TSO-Handbuch, 4. Auflage, Okt. 1979, Kernforschungszentrum Karlsruhe.
- <ARIE74> G. A. ARIELY (Ed.): Computer Measurement and Evaluation Handbook, Martin Marietta Corp., Data Systems, Baltimore, Nov. 1974.
- <ARTI78> P. ARTIS: Capacity Planning for MVS Computer Systems, Proc. of the Int. Conf. On Perf. of Comp. Inst. (ICPCI), Ed.: D. Ferrari, North Holland 1978, pp. 25-53.
- <BARD76> Y. BARD: A Characterization of VM/370 Workloads, Proc. of Modelling and Perf. Eval. of Computer Systems, Hrsg. Beilner/Gelenbe: North Holland, 1976, pp. 35-55.
- <BARD77> Y. BARD: The Modelling of Some Scheduling Strategies for an Interactive Computer System, in: Computer Performance, Hrsg. Chandy/Reiser, North Holland, 1977, pp. 113-138.
- <BARD78> Y. BARD: An Analytic Model of the VM/370 System, IBM Journ. of Research and Development, Vol. 22, 1978, pp. 498-508.
- <BARD79> Y. BARD: Some Extensions to Multiclass Queuing Network Analysis, 4th Symp. Model. and Perf. Eval. of Comp. Systems, Wien 1979, Vorabdruck.
- <BELL72> T. E. BELL, B. W. BOEHM, R. A. WATSON: Framework and Initial Phases for Computer Performance Improvement, AFIPS FJCC, 1972, Vol. 2, pp. 1141-1154.
- <BELL75> T. E. BELL, B. W. BOEHM: Issues in Computer Performance

- Evaluation: Some Consensus, Some Divergence,
ACM Performance Evaluation Review, July 1975, pp. 4-39.
- <BELL76> T. E. BELL: Managing Computer Performance with Control Limits, ACM Perf. Eval. Review, 1976, pp. 21-27.
- <BELL78> T. E. BELL: Performance Evaluation of Computer Systems, SEAS Spring Technical Meeting, Bern, April 1978.
- <BOAR76> M. BOARI, P. TOTH: Workload Definition and Performance Evaluation of a Large Computing System, Online-EUROCOMP Conf. on Computer Performance Evaluation, 1979, pp. 277-289.
- <BOCK70> H. BOCK: Automatische Klassifikation, Springer, 1970.
- <BOIE74> S. J. BOIES: User Behavior on an Interactive Computer System, IBM System Journal, 13, No. 1, 1974, pp. 2-18.
- <BOYS75> J. W. BOYSE, D. R. WARN: A Straightforward Model for Computer Performance Prediction, Comp. Surveys, 7, No. 2, Juni 1975, pp. 73-93.
- <BUCH69> W. BUCHHOLZ: A Synthetic Job for Measuring System Performance, IBM Systems Journal, 1969, pp. 309-318.
- <BUZE76> J. P. BUZEN: Fundamental Operational Laws of Computer System Performance, Acta Informatica, 7, 1976, pp. 167-182.
- <CHIU78> W. W. CHIU, M. W. CHOW: A Performance Model of MVS, IBM Systems Journal, 17, No. 4, 1978, pp. 444-462.
- <CIES78> R. J. CIESLOWSKI: Perceptions of Performance, AFIPS NCC, 1978, pp. 433-437.
- <CONT78> Control Data Corp.: CDC 33801 and 33502 Disk Memory Systems, Man. No. 83322210, 1978.
- <CORB62> F. J. CORBATO, M. MERWIN-DAGGETT, R. C. DALEY: An Experimental Timesharing System, (CTSS), AFIPS Conf. Proc. SJCC, 1962, pp. 335-344.
Diophantine Analysis, Chelsea 1966, Kap. 2, pp. 41-99.
- <DOHE75> W. J. DOHERTY: Measurement and Management of Interactive Computing, SHARE Annual Conf., Los Angeles, 1975.
- <DOHE79> W. J. DOHERTY, R. P. KELISKY: Managing VM/CMS Systems for User Effectiveness, IBM Systems Journal, 18, No. 1, 1979, pp. 143-163.
- <DRAP66> N. DRAPER, H. SMITH: Applied Regression Analysis, John Wiley, 1966.

- <DUDA73> R. O. DUDA, P. E. HART: Pattern Recognition and Scene Analysis, John Wiley, 1973.
- <DURA74> B. S. DURAN, P. L. ODELL: Cluster Analysis, Springer, 1974, (Lecture Notes in Economics and Mathematical Sciences).
- <ELME74> W. ELMENHORST: Verfahren zur Leistungsvermögensanalyse von Rechenanlagen, Diplomarbeit Universität Karlsruhe, 1974.
- <FANG77> H. FANGMEYER, R. GLODEN, J. LARISSE: An Automatic Clustering Technique Applied to Workload Analysis and System Tuning, Proc. of the Conf. Modelling and Performance Evaluation of Computer Systems, Ispra 1976, Hrsg. H. Beilner, E. Gelenbe, North Holland 1977, pp. 727-433.
- <FERR78> D. FERRARI: Computer Systems Performance Evaluation, Prentice-Hall, 1978.
- <FISC73> U. E. Fischer: Teilnehmerbetrieb, Verlagsgesellschaft Rudolf Müller, 1973.
- <FOGE72> FOGEL, WINOGRAD: EINSTEIN: An Internal Driver in a Time-Sharing Environment, Operating Systems Review, 6, No. 3, Oct. 1972, pp. 6-14.
- <FRIE75> H. P. FRIEDMAN, G. WALDBAUM: Evaluating System Changes Under Uncontrolled Workloads: A Case Study, IBM Systems Journal, 14, No. 4, 1975, pp. 340-352.
- <GOLD67> M. GOLD: Methodology for Evaluating Timeshared Computer Usage, Ph. D. Thesis, MIT, 1967.
- <GOMA76> H. GOMAA: Regression Models for the Evaluation of Computer System Performance, ONLINE-EUROCOMP Conf. 1976, pp. 69-99.
- <GOMA77> H. GOMAA: A Modelling Approach to the Evaluation of Computer System Performance, Proc. of the Conf. on Modelling and Performance Evaluation of Computer Systems, Ispra 1976, Eds. H. Beilner, E. Gelenbe, North Holland, 1977, pp. 171-199.
- <GRAE70> M. GRAEF, R. GREILLER, G. HECHT: Datenverarbeitung im Realzeitbetrieb, Oldenbourg Verlag, 1970.
- <GREE69> H. J. GREENBAUM: A Simulator of Multiple Interactive Users to Drive a Timesharing Computer System, Techn. Rep. TR-58, MIT-MAC, Jan. 1969.
- <GREN71> U. GRENANDER, R. F. TSAO: Quantative Methods for

Evaluating Computer System Performance, Review and Proposals, IBM Research Report RC3478, 1971.

- <HELL75> H. HELLERMAN, T. F. CONROY: Computer Systems Performance, McGraw-Hill, 1975.
- <HART75> J. A. HARTIGAN: Clustering Algorithms, John Wiley, 1975.
- <IBMO73> IBM: IBM System/370 Model 168 Functional Characteristics, 1973, GA22-7010.
- <IBMO74.1> IBM: A Guide to the IBM System/370 Model 158, 1974, GC20-1754.
- <IBMO74.2> IBM: A Guide to the IBM System/370 Model 168, 1974, GC20-1755.
- <IBMO75> IBM: OS/VS2 MVS, Initialization and Tuning Guide, 1975, GC28-0681.
- <IBMO75.2> IBM: Mathematical Programming System Extended (MPSX), Mixed Integer Programming (MIP), PP-No. 5740-XM3, Program Reference Manual, SH19-1099, 1975.
- <IBMO76.1> IBM: JES/3 Monitoring Facility (JMF), IUP-No. 57-PHR, 1976, SH20-1881.
- <IBMO76.2> IBM: IBM System/370 System Summary, 1976, GA22-7001.
- <IBMO76.3> IBM: OS PL/1 Optimizing Compiler, Programmer's Guide, 1976, SC33-0006.
- <IBMO76.4> IBM: OS PL/1 Checkout and Optimizing Compilers, Language Reference Manual, 1976, GC33-0009.
- <IBMO76.5> IBM: OS/VS2 SPL: Service Aids, 1976, GC28-0674.
- <IBMO76.6> IBM: OS/VS2 SPL: System Management Facility (SMF), 1976, GC28-0754.
- <IBMO76.7> IBM: OS/VS2 TSO Command Language Reference, 1976, GC28-0646.
- <IBMO76.8> IBM: Teleprocessing Network Simulator (TPNS), PP-No. 5740-XT4, Progr. Ref. Manual, SH20-1823, und Language Guide, SH20-1825, 1976.
- <IBMO76.9> IBM: Reference Manual for IBM 3330 Series Disk Storage, 1976, GA26-1615.
- <IBMO77.1> IBM: OS/VS2 MVS Resource Measurement Facility (RMF), PP-No. 5740-XY4, General Information Manual, 1977, GC28-0921.

- <IBMO77.2> IBM: OS/VS2 MVS Resource Measurement Facility (RMF), PP-No. 5740-XY4, Reference and Users Guide, 1977, GC28-0922.
- <IBMO77.3> IBM: OS/VS2 MVS Performance Notebook, 1977, GC28-0886.
- <IBMW76> IBM: Host Systems Performance, WTSC Data Systems Technical Report, Nov. 1976, ZZ10-9843.
- <JOSL68> JOSLIN: Computer Selection, Addison-Wesley, 1968.
- <KLEI76> L. KLEINROCK: Queueing Systems, Vol. II, Computer Appl., Wiley, 1976.
- <LAND76> K. LANDAU: Clusteranalytische Untersuchungen zur Computer-Arbeitslast, Elektronische Rechenanlagen, 18, Heft 6, 1976, pp. 273-277.
- <LITW68> R. J. LITWILLER: Analysis of I/O-Activity and Performance of the Technical Jobstream and Derivatives under OS/360 MVT. IBM Poughkeepsie, Technical Report TR00.1986, März 1968.
- <MATE78> W. MATERNA: Ein allgemeines Verfahren zur Konstruktion von repräsentativen Lastmodellen, Universität Hamburg, Bericht-Nr.: FBI-HH-B-43/78, 1978.
- <MEAD78> R. L. MEAD, H. D. SCHWETMAN: Job Scripts - A Workload Description Based on System Event Data, AFIPS NCC, 1978, pp. 457-464.
- <MERT77> B. MERTENS: A Pragmatic Approach to Quantative Performance Evaluation, Proc. Measuring, Modelling, and Evaluating Computer Systems, Bonn 1977, Hrsg. Beilner/Gelenbe, North Holland, 1977, pp. 1-17.
- <MOOR71> C. G. MOORE: Network Models for Large-Scale Timesharing Systems, Tech. Rep. 71/1, Univ. of Michigan, April 1971.
- <MORD69> L. J. MORDELL: Diophantine Equations, Academic Press, 1969.
- <MUEH77.1> H. MÜHLENBEIN: Ein Timesharing-Benchmark auf einer SIEMENS 4004/151 und 7748, RZ77.207, GMD-Verlag, 1977.
- <MUEH77.2> H. MÜHLENBEIN: Zum Begriff der Leistung von Dialogsystemen: Das Dehnungsgesetz und die Antwortzeit bei steigender Belastung, RZ77.208, GMD-Verlag, 1978; auch in: GMD-Spiegel 4/77, pp.16-28.

- <MUEH77.3> H. MÜHLENBEIN: Ein Vorschlag zur Leistungsbeschreibung von interaktiven DV-Systemen, GI-Workshop Organisation von Rechenzentren, Göttingen 1977, GMD-Verlag.
- <MUEL74> P. MÜLLER, H. KRIEGER, W. SCHÖNAUER: Der synthetische Standard-Benchmark (SSB), Universität Karlsruhe, 1974.
- <OSSW73> B. OSSWALD: Leistungsvermögensanalyse, Toeche-Mittler, 1973.
- <SACH78> L. SACHS: Angewandte Statistik, Springer, 1978.
- <SACK68> H. SACKMAN: Time-Sharing Versus Batch Processing: The Experimental Evidence, AFIPS SJCC, 1968, pp. 1-10.
- <SALT70> SALTZER, GINTELL: The Instrumentation of MULTICS, CACM, 13, No. 8, Aug. 1970, pp. 495-500.
- <SCHA67> M. SCHATZOFF, R. TSAO, R. WIIG: An Experimental Comparison of Timesharing and Batch Processing, CACM, 10, No. 5, 1967, pp. 261-265.
- <SCHE67> A. L. SCHERR: An Analysis of Time-Shared Computer Systems, MIT Press, 1967.
- <SCHR74> H. SCHREIBER, B. THOMAS, F. WOLF: Beschreibung eines synthetischen Jobmix für verschiedene Benchmark-Tests, NTG-GI Jahrestagung, Braunschweig 1974, pp. 218-232.
- <SCHR76.1> A. SCHREINER (Hrsg.): Betrieb von Rechenzentren, 1976.
- <SCHR76.2> D. SCHRIEFER: Ein Stichprobenmonitor für Terminalsysteme, KFK-Bericht 2248, 1976.
- <SCHR77> D. SCHRIEFER: Complex Workloads and System Capacity, IBM Research Report RC6608, 1977.
- <SEBE77> G. A. F. SEBER: Linear Regression Analysis, John Wiley, 1977.
- <SHAW64> C. J. SHAW: The JOSS System, Datamation, 10, No. 11, Nov. 1964, pp. 32-36.
- <SMIT67> L. B. SMITH: A Comparison of Batch Processing and Instant Turnaround, CACM, 10, No. 8, 1967, pp. 495-500.
- <SPAETH75> H. SPAETH: Cluster-Analyse-Algorithmen zur Objektklassifizierung und Datenreduktion, Oldenbourg Verlag, 1975.
- <SREE74> K. SREENIVASAN, A. J. KLEINMAN: On the Construction of a Representative Synthetic Workload, CACM 1974, pp. 127-133.
- <STRE75> D. N. STREETER: Productivity of Computer-Dependent Workers, IBM System Journal, 14, No. 4, 1975, pp. 292-305.

- <SVOB76> L. SVOBODOVA: Computer Performance Measurement and Evaluation Methods: Analysis and Applications, American Elsevier, 1976.
- <TERP76> K. TERPLAN: Workload Representation in Mixed Environment, Online-EUROCOMP Conf. on Computer Performance Evaluation, 1976, pp. 389-403.
- <VAJD61> S. VAJDA: Mathematical Programming, Addison-Wesley, 1961.
- <WALD73> G. WALDBAUM: Evaluating Computing System Changes by Means of Regression Models, First Annual SIGME Symposium on Measurement and Evaluation, 1973, pp. 127 - 135.
- <WILK68> M. V. WILKES: Timesharing Computer Systems, American Elsevier, 1968.
- <WIND77> M. WINDFUHR: Methoden und Verfahren zur Leistungsermittlung von Datenverarbeitungsanlagen für den standardisierten Einsatz in Unternehmen, Dissertation, Universität Karlsruhe, Mai 1977.
- <WOOD71> D. C. WOOD, E. H. FORMAN: Throughput Measurements Using a Synthetic Job Stream, AFIPS FJCC 1971, pp. 51 - 55.
- <ZORN75> W. ZORN: Systemoptimierung mit Hilfe von synthetischen Belastungsprofilen, in <SCHR76.1>, pp. 255-269.
- <ZORN80> W. ZORN, W.-H. GROSSE, M. HAAS: Leistungsoptimierung von DV-Systemen, Proc. Operations Research, Nr. 9, 1980, pp. 595-602.