



---

**Benchmark-Tests mit der  
IBM DB2 Digital Library  
unter Einsatz des  
Java Access Layers**

Udo Willke  
Werner Vogelpohl



# Übersicht



- 
- Testprogramm
  - Überlegungen zum Datenmodell
  - Übersicht der eingesetzten Hardware
  - identifizierte Einflußfaktoren
    - DL-Connect
    - Indexklassen
    - Performance der DL-Client-Rechner
  - Ergebnisse der Tests
  - Fix IR40907
  - weitere Messungen



---

### Untersuchung der Laufzeit von parametrischen Queries

1. Definition von DL-Queries verschiedener Art ("Testsuite")  
→ Parameter: Indexklasse, Kriterium, maximale Anzahl der Ergebnisse
2. Verarbeitung der Queries in einer **Java Application** unter Benutzung der Access Layer Klassen
3. Registrierung und Auswertung der Laufzeiten
4. Einzeltests und Tests mit 3, 6 und 12 gleichzeitigen DL-Clients

zusätzlich diskutieren: Standortbestimmung: diese Messungen / IBM-Studie  
IBM-Stude: Schwerpunkt auf Dimensionierung der Server-Systeme (Frage: Welchen Rechner brauche ich, um ein vorgegebenes Arbeitsvolumen, ausgedrückt in LSFR's, bewältigen zu können). Such-Operationen werden gar nicht berücksichtigt, sondern primär "create folder", "create item", "store / retrieve item" und "workbasket" - Operationen. Außerdem wird versucht, das Benutzerverhalten zu simulieren durch Einbau von Wartezeiten in die Testskripten. Weiterer Unterschied : Die Implementierungssprache ist C++, in unserem Fall ist es Java. Ergebnisse sind daher nicht unbedingt sehr vergleichbar.

Unsere Untersuchung konzentriert sich auf Laufzeitmessungen und Antwortzeiten, zunächst nur bei Anfragen an den Library Server, da sie sich in unserem Projekt als problematisch herausgestellt haben. Die bereits bekannten Performanceprobleme beim "Folder Manager" werden, wie vereinbart, gar nicht untersucht.



## Überlegungen zum Datenmodell



Nur 4 Indexklassen  
enthalten eine größere  
Menge an Items

→ Nur auf diesen  
Indexklassen machen  
parametrische Queries  
im Rahmen eines  
Performancetests Sinn

Indexklasse	Anzahl Items	max. Items
BWObject	4521	ca. 340
BWPerson	ca. 9000	ca. 730
BWObjectInstance	ca. 13500	ca. 2200
BWPart	ca. 50000+	ca. 1600
BWFormat	19	n/a
BWSubject	54	n/a
BWLanguage	1	n/a
BWCoverage	7	n/a
BWCreator	n/a	n/a
BWContributor	0	n/a
BWRelation		
BWValue		
BWAttribute		

Schätzung zur Anzahl der in jeder Indexklasse enthaltenen Items

Eine genaue Bestimmung der Zahlen lassen die DL-APIs nicht zu, da im DL-Client ein Laufzeitfehler auftritt, wenn die Zahl der Ergebnis-Items die in der Spalte "max Items" aufgeführte Zahl übersteigt. Dies muß bei den Tests berücksichtigt werden.

Trotz der Überlegungen zur Verteilung der DL-Items über die Indexklassen nehmen wir an, daß die zugrundeliegende Datenbank geeignete Suchstrategien besitzt, um die Suchzeit (weitgehend) unabhängig von der Anzahl der Items in der Indexklasse zu machen.



## Beispiele für Queries: "Suite 5"



```
BWObjectInstance:InstanceNumber > "3":0
BWObject:DatePublished < "1997":0
BWObjectInstance:Application == "text/plain":0
BWPerson:CompleteName LIKE "prechelt%":0
BWPart:PartNumber == "200":0
BWObjectInstance:GeneratedBy <> NULL:0
BWObjectInstance:GeneratedBy == "Acrobat Distiller 3.01":0
BWObject:(DatePublishedString == "1998-4-8") AND (DCType ==
"phdthesis"):0
BWPart:PartSize > "20000000":0
BWPart:PartSize BETWEEN "15000000" "20000000":0
BWPerson:CompleteName LIKE "schoenauer%":0
BWObjectInstance:InstanceNumber > "3":0
BWPerson:CompleteName LIKE "%peter%":0
...
```

Auszug aus der Testsuite 5, die nur wenige Ergebnis-Items in der jeweiligen Indexklasse generiert. Max.Results ist a priori nicht limitiert, das Anfragekriterium liefert jedoch "automatisch" wenige Ergebnisse zurück.

Problematisch an den VVV-Daten, daß sie nicht mit sehr differenzierter Metainformation ausgestattet sind. Die Anfragekriterien müssen daher verhältnismäßig einfach gehalten werden.

Um Caching-Effekte zu vermeiden, wurde darauf geachtet möglichst keine Systematik in der Abfolge der Queries zu haben.



## Java Code



```
private static long runQuery(TheObjectFactory Factory, String[] s)
    throws DKException, UnknownNameException {

    long t0, t1;
    int max_res = (new Integer(s[2])).intValue();

    Factory.setMaxResults(max_res);

    t0 = System.currentTimeMillis();
    TheFolder f = Factory.doQuery(s[0], s[1]);
    t1 = System.currentTimeMillis();

    System.out.println(s[0] + ":" + s[1] + ":" + f.count() + ":" + (t1 - t0));
    return (t1 - t0);
}
```

---

```
s[0] = <Indexklasse>, s[1]= <Kriterium>, s[2] = <Max. Results Parameter>
```

---

Private Methode der Java-Testklasse, die wir, der Einfachheit halber, als Java Anwendung gestaltet haben.

Tests mit Java Servlets wären viel schwerer handhabbar gewesen (Installation eines Java-fähigen HTTP-Servers, Problematik des Multi-Threading beim Test-Servlet, Auswertung der Ergebnisse)

Ablauf des Programmes:

- Einlesen der Testqueries
- kontinuierliches Abarbeiten der Testqueries über die Methode runQuery
- Protokollierung des Ergebnisses pro Query (Anzahl Anzahl Treffer, Laufzeit) nach stdout
- Rückgabe der Laufzeit der doQuery()-Methode zur Main()-Methode, um die Gesamtlaufzeit der Testsuite berechnen zu können



## Kennzahlen des DL-Server-Systems



Knoten	Prozessor	Taktfrequenz	RAM
dl001 (Object Server)	Power2	135 MHz	256 MB
dl002 (Library Server)	Power2	77 Mhz	256 MB
dl003/dl004 (Oracle DB's)	Power2	77 MHz	256 MB

**Plattenlaufwerke:**

4 x 18 GB über  
SSA - Controller

**Datenbestand:**

4521 BWOjects,  
d.h. ~ 6 x 751 VVV-  
Dokumente

**Grundkonfiguration:** 10 Library-Server Child-Prozesse

Das zentrale Server-System besteht aus insgesamt 4 SP/2 - Knoten, die als Library-Server, Object-Server und für die zugehörigen Oracle-Datenbanken genutzt werden

Die Kenngrößen des DL-Serversystems wurden über die gesamten Tests konstant gehalten.



## Übersicht der DL-Client-Systeme



Rechnername	Standort	OS	Prozessor	Taktfrequenz	RAM
<b>ubkaaixa</b> RS/6000 43P Model 150	UB	AIX 4.3.2	PowerPC	375 MHz	256 MB
<b>ubkaaixg</b> RS/6000 - F 50	UB	AIX 4.3.1	PowerPC ( 2 x )	166 MHz	256 MB
<b>ubkaps114</b> PC - System	UB	Win NT 4.0	Pentium II	300 MHz	128 MB
<b>vogelpohl</b> PC - System	RZ	Win NT 4.0	Pentium II	266 MHz	128 MB
<b>rzhelder</b> PC - System	RZ	Win NT 4.0	Pentium III	500 MHz	256 MB
<b>dl001</b> SP/2 Knoten	RZ	AIX 4.2.1	Power2	135 MHz	256 MB

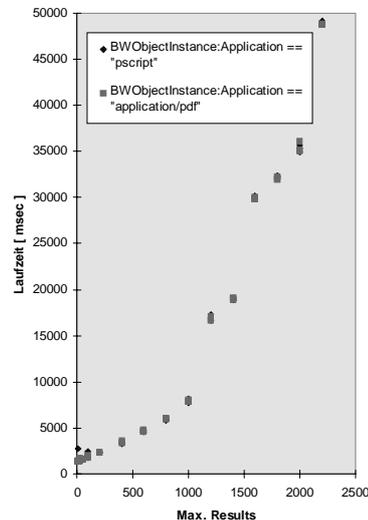
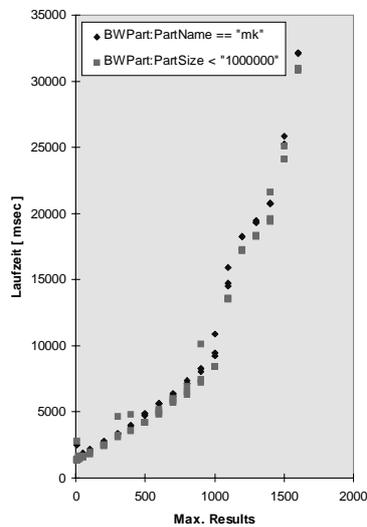
Kenngrößen der DL-Client-Systeme:

3 Rechner an der UB, 3 Rechner am RZ

3 AIX-RS/600-Systeme, 3 Win NT-PC-Systeme



## Einflußfaktor DL-Connect



14. Februar 2000

© Universitätsbibliothek Karlsruhe

9

Verhalten der Laufzeiten wenn für jede Query ein neuer DL-Connect ausgeführt wird.

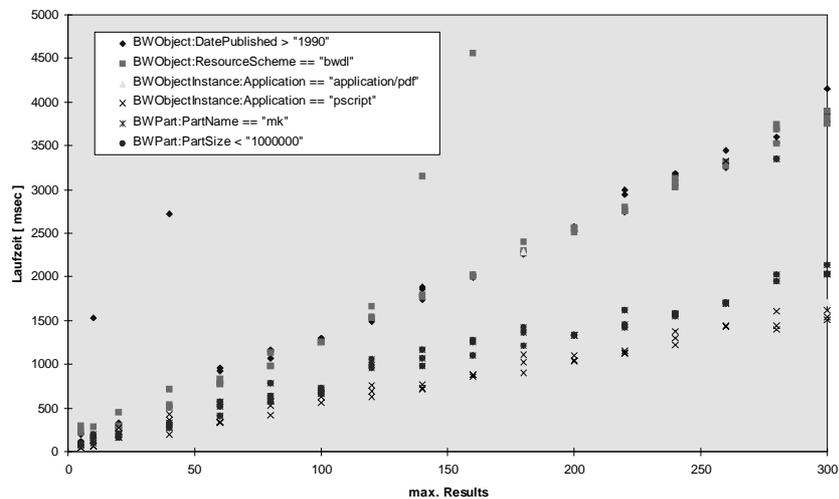
Treppenstruktur bei den Antwortzeiten kann nicht einfach erklärt werden, da die zum Test eingesetzte Java Klasse die Zeitmarken unmittelbar vor und nach der doQuery()-Methode ermittelt, und DL-Connect, bzw. Laden der JVM nicht berücksichtigt wird. Unklar ist ob für die beobachteten Artefakte evtl. die JVM verantwortlich ist. Es ist jedoch denkbar, daß das erneute Laden der JVM für jede Query einen Zusatzaufwand bei der Ressourcenallozierung bzw. bei der Garbage Collection nach sich zieht.

Wichtig: Meßpunkte schneiden y-Achse bei  $t = 1,4$  sec.

Meßpunkte der Tests mit nur 1 x DL-Connect verlaufen linear und schneiden y-Achse bei (+/-)  $t = 0$  sec



## Einflußfaktor Indexklasse



14. Februar 2000

© Universitätsbibliothek Karlsruhe

10

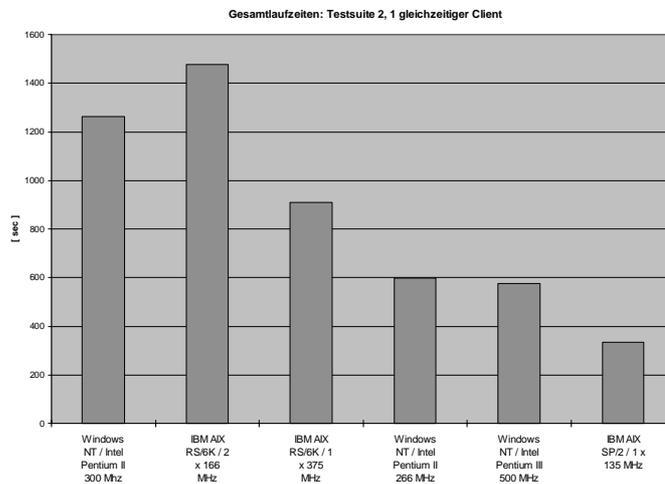
Vergleich der Laufzeiten einer Testsuite für verschiedene Indexklassen (mit verbesserter Java Klasse)

Tendenziell sind die Laufzeiten für Queries auf BWPart und BWOBJECTInstance kürzer als für BWOBJECT.

Der Unterschied kann für 300 Items ca. 2 Sekunden pro Query ausmachen



## Einflußfaktor DL-Client-Rechner



14. Februar 2000

© Universitätsbibliothek Karlsruhe

11

Schaubild: Gesamtlaufzeit der Testsuite 2 für die verschiedenen eingesetzten DL-Client-Systeme. Die drei linken Balken zeigen die Ergebnisse für die Rechner der UB, die drei rechten die Ergebnisse für die Rechner des RZ

Folgende Schlußfolgerungen können gezogen werden:

- 1) Die "Entfernung" des Library-Servers im Netzwerk scheint eine Rolle zu spielen
- 2) Die Rechenleistung der DL-Client-Rechner hat Einfluß auf die Laufzeit der "Testsuiten"
- 3) Unix-Systeme zeigen tendenziell bessere Ergebnisse als Win-NT-Systeme



## Rechnerlast unter Windows NT



“Testsuite 2”

BWPart:PartSize < "1000000", max. Results : 5 ... 1600 Items

BWPart:PartName == "mk", max. Results : 5 ... 1600 Items

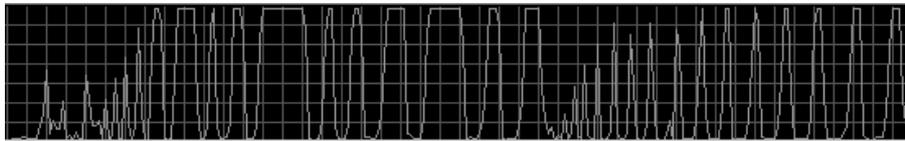


Illustration: Zunehmende Last auf einem Win-NT Client-Rechner, wenn der Max.Results Parameter sukzessiv erhöht wird, in Übereinstimmung mit der Schlußfolgerung aus dem vorhergehenden Schaubild

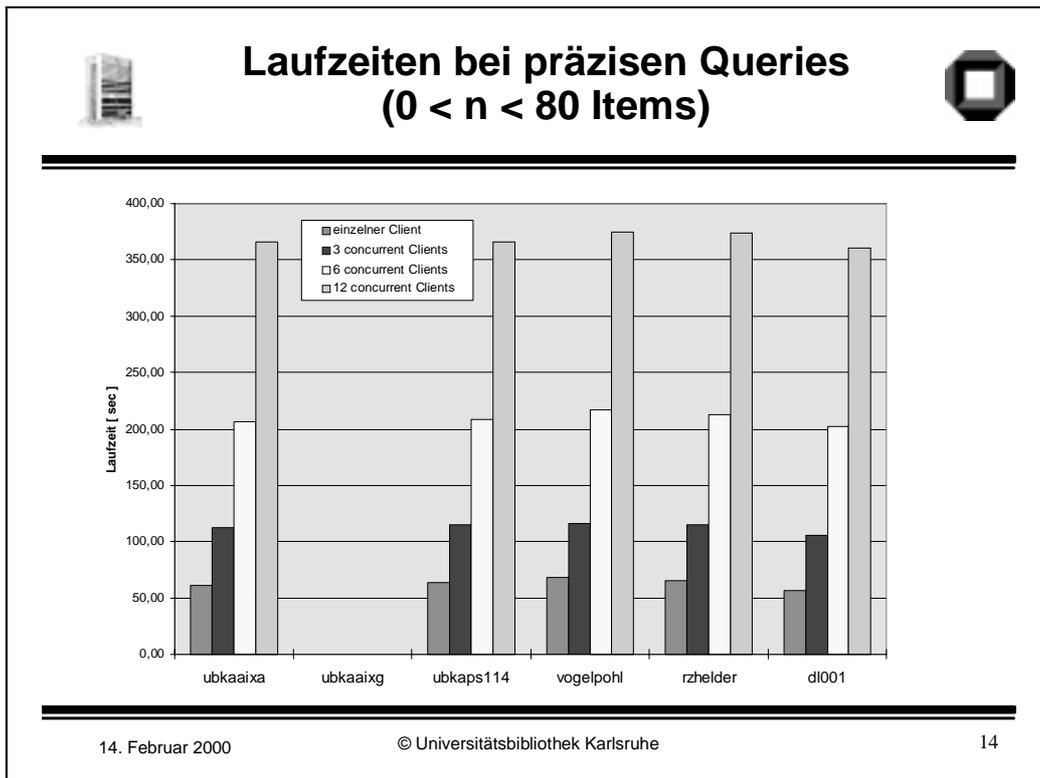


## Fallunterscheidung



- Queries, die nur wenige Items als Ergebnis haben (  $0 < n < 80$  )
- Queries, die “sehr viele” Items für die gegebene Indexklasse als Ergebnis haben (  $340 < n < 2200$  )
  - **BWObject** - 340 Items
  - **BWObjectInstance** - 2200 Items

Mit den in den Vortests gewonnenen Erfahrungen hat es sich als sinnvoll erwiesen, die Fallunterscheidung “kurz” Queries und “lange” Queries durchzuführen, da damit zwei unterschiedliche Betriebsarten des Library Server Systems provoziert werden können und auch die Ergebnisse für die Clients davon abhängen.



### 1. Fall: “präzise” Queries

Die gesamte Laufzeiten einer Testsuite für einzelnen Clients unterscheiden sich kaum. Die von den DL-Clients gestellten Anfragen werden vom Library-Server “reihum” bedient. Die Rechenleistung der DL-Clients spielt bei diesem Betriebszustand offenbar keine Rolle

Auf der Server-Seite beobachtet man sehr hohe Last ( 8 ) auf dem Oracle-Konten, jedoch nur sehr geringe Last auf dem Library - Server (unter 1)

Interpretation: Die Anfragen der DL-Clients an den Library-Server werden von diesem an die Oracle Datenbank weitergereicht. Sie erledigt in diesem Fall die Hauptarbeit

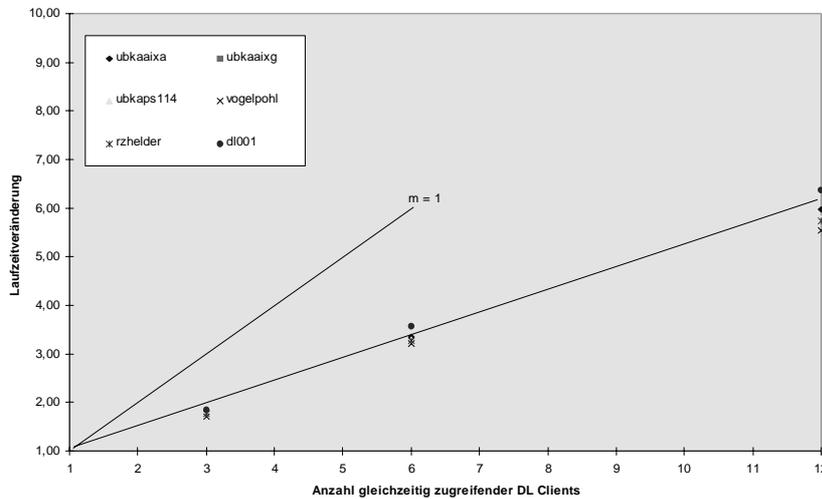
“3 gleichzeitige Clients” beschreibt die Situation, wenn jeweils die drei Client-Rechner an einem Standort (UB oder RZ) aktiv sind.

“6 gleichzeitige Clients” beschreibt die Situation, wenn alle sechs Rechner am Test teilnehmen

“12 gleichzeitige Clients” werden erreicht, wenn jeweils zwei JVM’s (Java Virtual Machine) pro Client-Rechner gestartet werden.



## Skalierungsverhalten bei präzisen Queries ( $0 < n < 80$ Items)



14. Februar 2000

© Universitätsbibliothek Karlsruhe

15

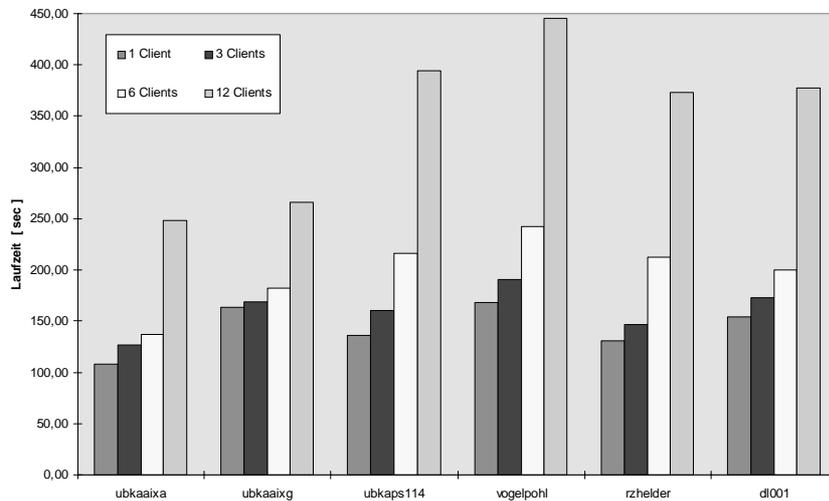
Darstellung der relativen Veränderung der Laufzeiten für die verschiedenen DL-Client-Systeme bei 3, 6 und 12 gleichzeitig auf den Library Server zugreifenden Clients.

Für jeden Client-Rechner werden die relative Veränderung der kumulierten Laufzeit für die gewählte Testsuite bei 3, 6 und 12 Clients zur Laufzeit bei nur einem Client berechnet

Das Ergebnis zeigt, daß die relativen Änderungen der Laufzeit in etwa linear mit der Anzahl der gleichzeitig aktiven Clients ansteigen. Die Hilfslinie mit der Steigung  $m=1$  zeigt die Annahme für das erwartete Verhalten des Server-Systems, das zu 100 % ausgelastet ist. Hier dürfte die Verdopplung der Anzahl der DL-Clients nur zu einer Verdopplung der Gesamtlaufzeit führen, also  $m=1$



## Laufzeitverhalten bei "unpräzisen" Queries (340 < n < 2200 Items)



14. Februar 2000

© Universitätsbibliothek Karlsruhe

16

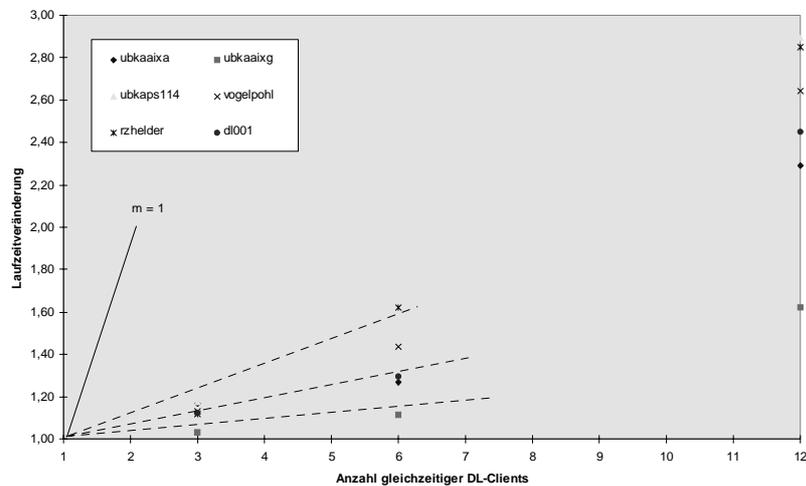
### 2. Fall: unpräzise Queries

In diesem Fall hat Rechenleistung der DL-Clients Einfluß auf die Gesamtlaufzeit der Testsuite

Serverseitig beobachten wir hohe Last (3 - 4) auf dem Library - Server, jedoch nur moderate Last auf dem zugehörigen Oracle-Knoten



## Skalierungsverhalten bei "unpräzisen" Queries (340 < n < 2200 Items)



14. Februar 2000

© Universitätsbibliothek Karlsruhe

17

Auswertung durch Normieren der Laufzeiten

Linie mit  $m=1$  zeigt wieder die Annahme zum Verhalten des Server-Systems bei 100 % Auslastung: Verdoppelung der Anzahl der Clients führt nur zur Verdoppelung der Laufzeit für die Anfragen

Steigung für die einzelnen Clients zunächst geringer als im Fall der präzisen Queries, dann jedoch Übergang zu höheren Steigungen. Dies zeigt wahrscheinlich den Beginn der Library Server - Saturation an

Unix - Systeme zeigen tendenziell bessere Performance als Win-NT-Systeme



## Netzwerkkommunikation mit dem Library Server



Query	Anzahl Treffer	Anzahl Pakete	Anzahl Bytes
BWPerson:CompleteName LIKE "schoenauer%":0	1	73	70.764
BWObject:ResourceScheme == "bwdl":340	340	1894	2.826.680

### Abschätzung:

Summe der Längen der Datenfelder in der Indexklasse x Anzahl der  
Treffer

BWPerson: 3.405 Bytes x 1 Treffer = **3.405 Bytes**

BWObject: 7.198 Bytes x 340 Treffer = **2.447.320 Bytes**

Die bisherigen Auswertungen der Laufzeittests lassen die Vermutung aufkommen, die Antwortzeiten würden nicht nur durch die Rechen- und Transaktionsleistung von DL-Client und -Server beeinflusst, sondern auch durch die Netzwerkkommunikation. Diese wurde daher mit dem unter AIX verfügbaren Tool "iptrace" untersucht, und die vom Library-Server kommenden Datenpakete analysiert.

Man sieht, daß gerade bei Queries auf "großen" Indexklassen, d.h. solche deren Gesamtlänge der Datenfelder sehr groß ist, eine erhebliche Datenübertragung über das Netzwerk ausgelöst wird.

Die gute Übereinstimmung zwischen der Abschätzung für das Datenvolumen bei BWObject mit den aus "iptrace" gewonnenen Daten läßt den Schluß zu, daß alle Datenfelder dieser Indexklasse für jeden gefundenen Item vollständig übertragen werden.

Das Netzwerk ist in einem solchen Fall daher der limitierende Faktor für das Antwortzeitverhalten.

Umgekehrt könnte aus diesen Ergebnissen der Schluß gezogen werden, daß ein Datenmodell mit vielen "kleinen" Indexklassen grundsätzlich performanter sein müßte als das hier untersuchte Datenmodell. Ein solches optimiertes Datenmodell wäre jedoch auf einen performanten Folder Manager bei der Verknüpfung von Informationen aus verschiedenen Indexklassen angewiesen.



## Fazit



- 
- Der limitierende Faktor bei den Antwortzeiten sind die großen übertragenen Datenvolumen und die intensive Netzkommunikation
  - Performance des DL-Client-Rechners hat größere Auswirkung auf die Gesamtperformance als vermutet
  - Skalierungsverhalten der IBM DB2 Digital Library unter großer Last besser als erwartet



## Fix IR40907



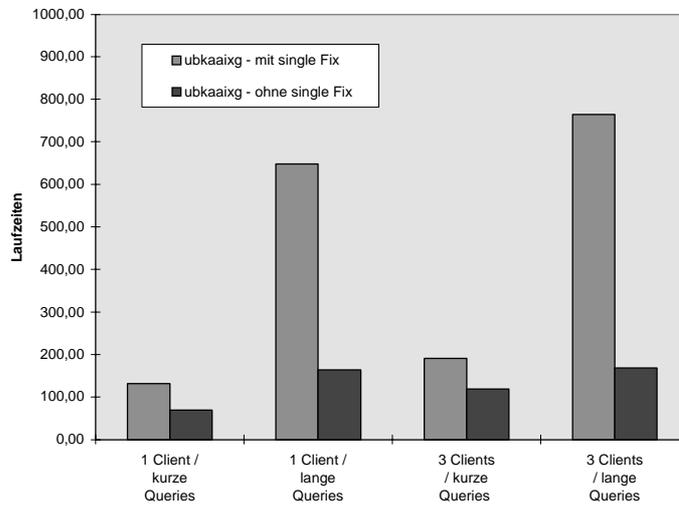
- 
- Installation: Fixpack 8 + Austausch einzelner Dateien  
Unter AIX:

- dksdk.jar
  - libdkmm.a
  - libkdSDL.so

- Tests zeigen: Fix verbessert nur die Funktion auf der DL-Client-Seite, Wirkung bei Library Server unklar
- Alle Dateien sind auch bei einer reinen Client-Installation vorhanden
- Bis zu 5000 Ergebnis-Objekte (DDO's) nun offenbar möglich



## Auswirkungen des Fix IR 40907





## weitere Messungen



- “combined queries”
  - Folder Manager (!)
- Textminer - Queries
- Queries über die JDBC Datenbankschnittstelle