Erschinen im Tagungsband der 5. GI/NTG Fachtagung "Messung, Modellierung und Bewertung von Rechensystemen und Netzen", Brauschweig, September 1989

Charakteristische Eigenschaften von Transaktionslasten in DB/DC-Systemen

Volker Bohn

Universität Kaiserslautern, Fachbereich Informatik,
Postfach 3049, D-6750 Kaiserslautern

Kurzfassung

Informationen über charakteristische Eigenschaften von Transaktionssystemen (DB/DC-Systemen), über die Zusammensetzung der Transaktionslast, das Referenzierungsverhalten und Lokalitätsverhalten sowie über die Häufigkeit und Intensität von Lastschwankungen spielen eine entscheidende Rolle bei der Bewertung von Architekturvorschlägen zur Realisierung von Hochleistungs-Transaktionssystemen, bei der Entwicklung und Optimierung von leistungskritischen Algorithmen in transaktionsverarbeitenden Systemen und bei der Parametrisierung von Simulationsprogrammen. Wir haben deshalb eine allgemein anwendbare Methodik entwickelt, reale DB/DC-Anwendungen zu untersuchen. Die Vorgehensweise und die in langjährigen umfangreichen Analysen ermittelten Ergebnisse über die Zusammensetzung der Transaktionslast, zeitlichen Änderungen in der Systembelastung und dem Referenzierungs- und Lokalitätsverhalten stellen wir in diesem Aufsatz vor.

Die Untersuchungsergebnisse zeigen, daß in 'typischen' DB/DC-Anwendungen die Transaktionslast heterogen zusammengesetzt ('Länge' der Transaktionen, Verhältnis von Lese- und Schreibzugriffen), sowie die Systembelastung häufigen und intensiven Schwankungen unterworfen ist. Die Verteilung der Zugriffe über die Datenbank ist sehr ungleichmäßig. Es wurde festgestellt, daß das in einer 'Working-Set-Analyse' untersuchte Lokalitätsverhalten besonders für in Mehrechner-Architekturen sehr leistungskritische Algorithmen ('Lastkontrolle') ein teilweise überraschend hohes Optimierungspotential aufweist.

Weil sich die untersuchten realen Anwendungen in den hier betrachteten Aspekten nicht unterscheiden, kann man annehmen, daß die vorgestellten Ergebnisse für ein sehr breites Anwendungsspektrum gültig sind. Die Ergebnisse werden den Eigenschaften des sogenannten Debit/Credit-Benchmarks, der heute oft zur Leistungsbewertung für transaktionsverarbeitende Systeme, insbesondere bei Systemvergleichen, herangezogen wird, gegenübergestellt. Aus dieser Gegenüberstellung folgt, daß der Debit/Credit-Benchmark nur geeignet ist, Transaktionssysteme hinsichtlich eines sehr schmalen Spektrums leistungskritischer Algorithmen zu vergleichen, das durch reale Lasten gestellte Anforderungen nur zu einem geringen Teil abdeckt. Verwendet man demnach diesen Benchmark zur Bewertung von Hochleistungs-Transaktionssystemen oder in der Entwicklung befindlicher Systeme, so werden mit hoher Wahrscheinlichkeit die unrealistischen Anwendungsfälle bewertet bzw. optimiert.

1. Einleitung

Transaktionssysteme gestatten den jeweiligen Anwendungen 'bedeutungsvolle' Verarbeitungsschritte in betrieblichen Abläufen in Form von Transaktionen (TA) durchzuführen und dabei die Vorteile des TA-Konzeptes /HR83/ für die Verarbeitung gemeinsamer betrieblicher Daten auszunutzen. Durch ihren Einsatz wird eine dialogorientierte effiziente Sachbearbeitung direkt am Arbeitsplatz ermöglicht, wobei das System den Benutzer über Bildschirmformulare führt und bei der Auftragsformulierung unterstützt. Heute finden sich Transaktionssysteme typischerweise in Anwendungen, in denen eine große Anzahl von Auskunfts-, Buchungs-, und Datenerfassungsvorgängen anfallen, also beispielsweise als Systeme zur Kontenbuchung, Platzreservierung oder Bestellabwicklung. An derartige Transaktionssysteme (DB/DC-Systeme) werden jetzt neue Forderungen gestellt, die sie bereits in naher Zukunft befriedigen müssen /Gr85/:

- hohe Leistungsfähigkeit der Systeme, z.B. 1000 Transaktionen pro Sekunde mit einer maximalen Antwortzeit von einer Sekunde,
- extrem hohe Verfügbarkeit (annähernd 100%),
- lineares modulares Wachstum, d.h. lineare Durchsatzsteigerung ohne Zunahme der Anwortzeit beim Ergänzen des Systems um einen Verarbeitungsprozessor.

Diese Systemcharakteristika, insbesondere hohe Verfügbarkeit und modulares Wachstum, sind nur durch den Einsatz von Mehrrechner-Architekturen erreichbar.

Das Leistungsverhalten *zentraler* Transaktionssysteme (die auf einem zentralen Verarbeitungsrechner (VAR) zur Verfügung stehen) hängt bereits entscheidend davon ab, wie gut es den verschiedenen Algorithmen gelingt, bestimmte Eigenschaften der Transaktionslast zu Optimierungszwecken zu nutzen. Dabei sind die wichtigsten Eigenschaften

- das referentielle Lokalitätsverhalten der Transaktionen, das im Datenbank-Systempuffer leistungssteigernd ausgenutzt wird und
- der Anteil ändernder und nur lesender Datenzugriffe, der im Zusammenhang mit der Menge und Dauer der von den Transaktionen referenzierten Daten in speziellen Synchronisationsalgorithmen zu Leistungsoptimierungen herangezogen werden kann.

Das Leistungsverhalten von *Mehrrechner*-Transaktionssystemen hängt noch sehr viel mehr von der geschickten Nutzung bestimmter Eigenschaften der Transaktionslast ab. Besonders der Aspekt der Ausnutzung des referentiellen Lokalitätsverhaltens wird dadurch unterstrichen, daß ein hoher Anteil an rechnerübergreifender Kommunikation in Mehrechner-Architekturen zu großen Leistungseinbußen führt, weil hierfür beträchtliche CPU-Ressourcen verschlungen werden und alle Kommunikationszeiten additiv in die Belegungszeiten der Betriebsmittel (insbesondere der Daten und Sperren) sowie in die Antwortzeit der Transaktion eingehen. Diese rechnerübergreifenden Kommunikationsvorgänge sind immer dann notwendig, wenn eine Transaktion auf Daten zugreift, die nicht demjenigen VAR zugeordnet sind, auf dem die TA gestartet wurde (externe Datenzugriffe). In verteilten Datenbanksystemen muß das System, um die Atomizität der Transaktion zu gewährleisten, bei Transaktionsende ein aufwendiges Zwei-Phasen-Commit durchführen, dessen Dauer ebenfalls die Antwortzeit der Transaktion verlängert.

Will man in Mehrrechner-Architekturen vorhandenes Lokalitätsverhalten nutzen, so muß dies geschehen im Rahmen sogenannter Lastkontroll-Maßnahmen durch

- geschickte <u>Verteilung der Daten</u> ('Partitionierung') und daran angepaßte
- <u>Zuordnung der Transaktionslast</u> zu den Verarbeitungsrechnern ('TA-Routing', /LYL88, Ra86/).

Wir wollen hier nicht auf die Vielzahl von Algorithmen zur Verwaltung des DB-Systempuffers, zur Synchronisation und das weite komplexe Gebiet der Lastkontrolle eingehen; wichtig ist uns hier lediglich die Feststellung, daß bei der Entwicklung jedes Transaktionssystems, gerade bei Hochleistungs-Transaktionssystemen, Annahmen über Eigenschaften der Systemlast getroffen werden müssen, die in die Implementierung eingehen und dadurch das Leistungsverhalten des Systems stark beeinflussen. Selbst jede wissenschaftliche Diskussion um geeignete Architekturen zum Erreichen der genannten Forderungen braucht solche Annahmen. Will man außerdem die Eignung bestimmter Architekturtypen durch Simulationen überprüfen, so ergeben sich Probleme bei der Parametrisierung des Simulationsprogramms und seiner Versorgung mit möglichst realistischen Lasten, gerade weil geeignete Informationen fehlen. Auch hier sind solche möglichst realistische Lastannahmen erforderlich.

Wir haben deshalb an verschiedenen realen Systemanwendungen umfangreiche Untersuchungen vorgenommen mit den allgemeinen Zielen,

- eine breitere Argumentationsbasis für die **Bewertung von Architekturvorschlägen** zur Realisierung von Hochleistungs-Transaktionssystemen zu schaffen,
- die Entwicklung und Optimierung von **leistungskritischen Algorithmen und Lastkontrollverfahren** zu unterstützen,
- die realitätsnahe **Parametrisierung von Simulationsprogrammen** zu ermöglichen und

- die **Lasterzeugung für Simulationen**, die sich in /LYL88, MW88, Bo88/ als sehr kritischer Aspekt erwiesen hat, realistisch gestalten zu können.

Speziell haben wir dabei versucht, Antworten auf folgende Fragen zu geben:

- **Zusammensetzung der Transaktionslast:** Ist die Transaktionslast aus verschiedenartigen Transaktionstypen zusammengesetzt oder herrschen in Bezug auf die Länge der Transaktionen und dem Verhältnis von Lese-/Schreibzugriffen eher gleichartige Transaktionstypen vor (Homogenität der Transaktionslast)?
- **Zeitliche Änderung der Systembelastung:** Ist die Systembelastung durch die zu bearbeitende Transaktionslast konstant oder sind starke Schwankungen erkennbar (Variation der Transaktions-Ankunftsraten)?
- **Referenzierungsverhalten der Transaktionslast:** Wie verteilen sich die Datenzugriffe über die Datenbank?
- **Lokalitätsverhalten der einzelnen Transaktionstypen:** Wie ist das 'Working-Set-Verhalten' der einzelnen Transaktionstypen und bei welchen 'Fenstergrößen' überwiegt der Zugriff auf disjunkte, dem einzelnen Transaktionstyp zuzuordnende Datenbereiche?

Aus den Antworten auf diese Fragen lassen sich Schlüsse ziehen über

- die Partitionierbarkeit realer Datenbanken im Hinblick auf Nutzung referentieller Lokalität,
- die Notwendigkeit von dynamischen Lastkontrollmaßnahmen in Mehrrechner-Transaktionssystemen, und außerdem
- das durch Lastkontrollmaßnahmen überhaupt nutzbare Optimierungspotential.

Zusätzlich werden wir unsere Ergebnisse den Eigenschaften des sogenannten Debit/Credit-Benchmarks /An85/, der oft als *die Transaktionslast* schlechthin bezeichnet wird, gegenüberstellen und dessen Eignung zur Bewertung der Leistungsfähigkeit von Hochleistungs-Transaktionssystemen diskutieren. Der Debit/Credit-Benchmark kennt nur einen einzigen Transaktionstyp. Es handelt sich um eine - wie der Name schon andeutet - Kontenbuchungstransaktion. Die Transaktion greift auf einen Konto-Satz, einen Geschäftsstellen-Satz und einen Automaten-Satz (Geldautomat, 'Automatic Teller Machine') zu, ändert jeden Satztyp (es werden Umsätze am Automaten und der Geschäftsstelle erfaßt) und vermerkt die Buchung zu Revisionszwecken. Sie besteht somit aus 9 einfachen datenmodell-unabhängigen Datenbank-Befehlen: 3 lesende und 4 schreibende Zugriffe sowie Beginn und Ende der Transaktion. Als Leistungskennzahl dient der mit dieser Transaktion maximal erreichbare Durchsatz in Transaktionen pro Sekunde (TPS), wobei 95% der ausgeführten Transaktionen eine Antwortzeit von maximal einer Sekunde haben dürfen. Mit dem Maß 'TPS' kann die Leistungsfähigkeit eines Transaktionssystems mit einer einzigen Zahl ('golden number') ausgedrückt werden, und es wird heute oft als (wichtiges) Leistungsmaß für transaktionsverarbeitende Systeme insbesondere bei Leistungsvergleichen (und Kaufverhandlungen) herangezogen.

Im folgenden Kapitel werden wir zunächst unser Vorgehen beschreiben. Anschließend stellen wir (Kapitel 3) die Ergebnisse über die Zusammensetzung der Transaktionslast, die zeitlichen Änderungen der Systembelastung und das Lokalitätsverhalten vor, um die gestellten Fragen der Reihe nach zu beantworten. Dabei können wegen des umgfangreichen Zahlenmaterials nur ausgewählte, repräsentative Daten vorgestellt werden. Im Kapitel 4 gehen wir auf die sich aus unseren Ergebnissen ergebenden Folgerungen ein.

2. Die Vorgehensweise

Wir haben über Jahre hinweg verschiedenste Analysemethoden für DB/DC-Systeme entwickelt, wobei ihr Einsatz bei realen DB/DC-Anwendungen zu ständiger Verfeinerung führte. Durch das Sammeln von sehr detaillierter Protokollinformation, die die Systemlast und systeminterne Abläufe recht genau beschreibt, ist es uns heute möglich, die im folgenden Text vorgestellten Auswertungen vorzunehmen und charakteristische Eigenschaften von Transaktionslasten, die sich auf einen Beobachtungszeitraum von deutlich mehr als fünf Jahren erstrecken, zu untersuchen. Der Aufbau der Protokolldaten (später als 'Referenzstrings' bezeichnet) und die darin enthaltenen Informationen sind in der hier vorgestellten Form nicht nur für statistische Auswertungen interessant; vielmehr dienen sie uns auch als realitätsnahe Lasten zum Treiben von Simulationsprogrammen. Wir halten unsere Vorgehensweise für eine allgemein einsetzbare Methodik zur Untersuchung DB-basierter Anwendungen. Ihre Vorteile liegen vor allem in der detaillierten Analyse von Datenbank- und Lastcharakteristika. Weiterhin kann man dadurch, daß die Protokolldaten das Aufgezeichnete *konservieren*, auch rückblickend Auswertungen vornehmen und die Werte mit aktuelleren vergleichen; dadurch werden Fehleinschätzungen bei der Systementwicklung und später im Systembetrieb vemieden.

Wir beschäftigten uns mit DB/DC-Anwendungen, in denen die ausführbaren Transaktionen als vordefinierte Programme vorliegen und keine oder wenige Ad-Hoc-Anfragen mittels Anfragesprachen vorgenommen werden. Jedes vordefinierte Programm beschreibt einen Transaktionstyp, der vom (parametrischen) Benutzer aktiviert werden kann. Alle von uns untersuchten Anwendungen sind Anwendungen des Codasyl-Datenbanksystems UDS auf zentralen Verarbeitungsrechnern (keine Mehrrechner-Anwendungen) /UDSa, UDSb/.

Weil die umfangreichen Untersuchungen den Betrieb realer DB/DC-Anwendungen stark behindern würden und zudem sichergestellt werden muß, daß sich verschiedene Auswertungen auf denselben zeitlichen Ausschnitt der Systemlast der betrachteten Anwendungen beziehen, wurden diese Auswertungen nicht im laufenden Betrieb durchgeführt. Stattdessen wurden durch die Aktivierung verschiedener Meßpunkte im System anwendungsbezogene Protokolle erstellt, die eine Fülle von Detailinformationen beinhalten. Aus diesen, im allgemeinen aus mehreren Dateien bestehenden Protokollinformationen wurde mit Hilfe eines Filterprogramms für jede der betrachteten Anwendungen - insgesamt drei - ein sogenannter Seitenreferenzstring erzeugt (Bild 2.1). Die vorgestellten Auswertungen wurden dann anhand dieser Referenzstrings durchgeführt.

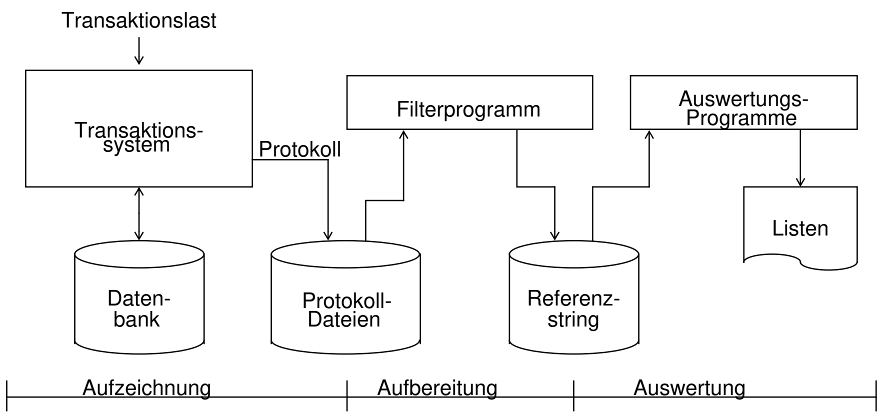

Bild 2.1: Erstellung und Auswertung der Referenzstrings

Jeder Seitenreferenzstring enthält - unter anderem - folgende Informationen:

- Zeitpunkt des Beginns jeder Transaktion, den Namen des Transaktionstyps sowie eine eindeutige Kennzeichnung der Transaktion (TA-Nummer),

- Zeitpunkt und eindeutige Identifikation jeder durch die Transaktion (TA-Nummer) referenzierten Datenbankseite, die Art des Zugriffs (lesend, schreibend) und den
- Zeitpunkt des Transaktionsendes jeder während der Protokollphase ausgeführten Transaktion.

Damit ist jeder Seitenreferenzstring in Bezug auf den chronologischen Ablauf, den 'Transaktionsmix' und das Referenzierungsverhalten der Transaktionen ein genaues Abbild der protokollierten Systemlast und geeignet, umfangreiche Untersuchungen anwendungsbezogen durchzuführen.

Wegen der begrenzten Seitenzahl können wir hier nur drei verschiedene, von uns untersuchte Anwendungen präsentieren. Weil das Datenmaterial sehr umfangreich ist, sind wir weiterhin gezwungen, uns auf die Darstellung einer repräsentativen Auswahl zu beschränken. Aufgrund unserer Erfahrungen gelten unsere Beobachtungen jedoch für 'typische' DB/DC-Anwendungen.

Tabelle 2.1 gibt einen Überblick zu den verwendeten Referenzstrings, denen wir die Namen DOA, DOD und KD gegeben haben; man erkennt neben der Zahl der ausgeführten Transaktionen und der Zahl der Referenzen insbesondere die 'Länge' der Transaktionen (in Anzahl ausgeführter Referenzen) und die unterschiedlichen Anteile von Änderungs-Transaktionen sowie ändernden Zugriffen auf Datenbankseiten.

	DOA	DOD	KD
#Transaktionen	17478	674	2014
Anteil Änderungs-TA (%)	29.5%	46.5%	13.4%
#Transaktionstypen	12	13	2
#referenzierte Areas	9	9	12
#Seitenreferenzen	1024207	40751	57959
Anteil ändernde Referenzen (%)	1.9%	3.7%	9.8%
#Referenzen pro Transaktion	58.6	60.5	28.8
Protokollierungdauer (hh:mm:ss)	2:47:42	0:20:56	0:29:55

Tabelle 2.1: Einige wichtige Merkmale der Referenzstrings 'DOA', 'DOD' und 'KD'

Erschinen im Tagungsband der 5. GI/NTG Fachtagung "Messung, Modellierung und Bewertung von Rechensystemen und Netzen", Brauschweig, September 1989

3. Ergebnisse und Analysen

3.1 Zusammensetzung der Transaktionslast

Wir wollen hier die Frage beantworten, ob die Transaktionslast im allgemeinen aus verschiedenartigen Transaktionstypen zusammengesetzt ist oder eher gleichartige Transaktionstypen vorherrschen. Zur Platzersparnis diskutieren wir diese Frage nur am repräsentativen Beispiel DOD.

TA-TYP	A3	A4	A5	A7	A10	A11	A12	A13	A14	Summe
ABSTART1	13333 801				914 -		4969 112			19216 112
LS20	2707 44	130 8	1 1	1616 62	619 -	102 -	7 -	167 39	76 60	5425 214
LSDISPO	1567 41	8 4		2 -	1765 -		32 -	299 47	137 110	3811 202
ABSTART2	1392 24		5 5	118 18	150 -	18 -	1006 192	918 213		3608 452
ABSTART3	3046 -				260 -					3306 -
ADHOCST	1444 -	207 84		521 12	727 -			263 88	45 36	3207 220
ACBATCH	34 -				51 -			1132 6		1217 6
ABSTART4	34 -	15 9							211 164	414 192
UEBERFUE								363 -		363 -
LSSTART	16 -	49 4		14 3	12 -			21 5	5 4	106 16
IQS					18 -				52 -	52 -
GDDIALOG										18 -
ZAROOT	1 -	7 4								8 4
Summe	23729 128	417 113	6 6	2271 95	4505 -	120 -	4016 304	3162 398	526 374	40751 1418

Tabelle 3.1: Referenzmatrix für den Referenzstring DOD

Tabelle 3.1 zeigt die Referenzmatrix für den String DOD. Für jeden Transaktionstyp wird für seine Aktivierungen nach Tabelle 3.2 angegeben, wieviele Seiten pro Datenbereich ('Area') referenziert wurden (Zeilen); die obere Zahl gibt die gesamte Zahl der Zugriffe an, die untere die Summe der darin enthaltenen ändernden Zugriffe. Dabei wird unterschieden, auf welche 'Area' (in Codasyl-Systemen oft auch 'Realm' genannt) diese Zugriffe entfallen (Spalten); außerdem sind die Spalten- und Zeilensummen angegeben. In Tabelle 3.2 ist die Anzahl der während der Aufzeichnung ausgeführten Transaktionen (alphabetisch sortiert) für jeden Transaktionstyp dargestellt.

TA-TYP	#Aktivierungen
ABSTART1	105
ABSTART2	88
ABSTART3	39
ABSTART4	126
ACBATCH	1
ADHOCST	65
GDDIALOG	3
IQS	2
LS20	105
LSDISPO	128
LSSTART	9
UEBERFUE	1
ZAROOTB	2
Summe	674

Tabelle 3.2: Transaktionsaktivierungen für DOD

Die Analyse der Tabellen führt zu folgenden Ergebnissen:

- Es gibt wenige dominante Transaktionstypen, die die deutliche Mehrzahl der ausgeführten Referenzen verursachen (ABSTART1).
- Die Transaktionstypen unterscheiden sich erheblich in der Zahl der ausgeführten Referenzen und dem Anteil ändernder Zugriffe (vergleiche LS20 und ABSTART3).
- Die Zahl der Transaktions-Aktivierungen (siehe Tab. 3.2) pro Transaktionstyp schwankt beträchtlich.
- Der am häufigsten ausgeführte Transaktionstyp (LSDISPO) ist nicht derjenige mit den meisten Referenzen; vielmehr sind große Unterschiede der durchschnittlich pro Transaktion ausgeführten Referenzen zwischen den Transaktionstypen offensichtlich (LSDISPO: 29.8; UEBERFUE: 363). Die Transaktionstypen unterscheiden sich demnach in der 'Länge der Transaktion'.

Die Zusammensetzung der Transaktionslast ist somit im Hinblick auf die Zahl und Art der Referenzen der Transaktionstypen und die unterschiedliche Zahl von Ausführungen **sehr heterogen**.

3.2 Zeitliche Änderungen in der Systembelastung

Um die Frage zu beantworten, ob die Belastung des Transaktionssystems eher konstant ist oder starken Schwankungen unterliegt, haben wir die Zeitpunkte des Transaktionsbeginns jeder Transaktion ausgewertet und Histogramme erstellt, die die Transaktionsankunftsraten bildlich darstellen; dabei war auch eine Unterscheidung nach den einzelnen Transaktionstypen möglich.

Bild 3.1 zeigt die Transaktionsankunftsraten für den Referenzstring DOA ohne Unterscheidung nach Transaktionstypen. An der Waagerechten ist die Zeit in Sekunden angegeben; für jede im Intervall neu gestartete ('ankommende') Transaktion wurde in der Senkrechten ein Stern plaziert. In Bild 3.2 ist dargestellt, wie Aktivierungen des Transaktionstyps mit dem Namen 'AR10' zur Gesamtlast beitragen.

Bild 3.1: Ankunftsratenhistogramm für den Referenzstring DOA

Bild 3.2: Ankunftsratenhistogramm des TA-Typs AR10 im Referenzstring DOA

Bei der Auswertung dieser und der nicht dargestellten Histogramme stellten wir fest, daß nur etwa 3/4 aller Transaktionstypen zu jedem Zeitpunkt (genauer: in jedem Zeitintervall, das uns als kleinste Einheit auf der Zeitachse diente) der der Aufzeichnungsdauer überhaupt aktiv waren, d.h. mit Transaktionsaktivierungen zur Gesamtlast beigetragen hatten. Es gibt jedoch sowohl Transaktionstypen, die während der gesamten Aufzeichnungsdauer aktiv waren und andere, die im Extremfall nur zu etwa 2% der gesamten Zeitspanne 'punktuell' auftraten. Zusammengefaßt gilt:

- Die Transaktionsankunftsraten weisen eine starke Varianz auf; dies gilt sowohl für die gesamte Last als auch für jeden einzelnen Transaktionstyp.
- Längst nicht alle Transaktionstypen sind während des Systembetriebes 'aktiv', d.h. tragen zur Gesamtlast bei.

Die Systemlast ist also nicht nur - wie in Kapitel 3.1 festgestellt - heterogen zusammengesetzt, sondern weist zusätzlich starke Schwankungen auf, die so extrem sind, daß sie häufig einer 'Sprungfunktion' gleichkommen. Dadurch, daß nicht alle Transaktionstypen zu jedem Zeitpunkt aktiv sind, ändert sich außerdem die Art der Systembelastung ('aktueller Transaktionsmix') laufend. Die Systembelastung ist also **inhomogen und nicht konstant**.

3.3 Referenzierungsverhalten

In der Einleitung haben wir festgestellt, daß das Leistungsverhalten von Mehrrechner-Transaktionssystemen ganz besonders von der Partitionierung des Datenbestandes (also der Möglichkeit der Zuordnung solcher Partitionen zu den einzelnen Rechnern) und des daran angepaßten Transaktionsroutings abhängt. Diese Probleme sind jedoch nur dann einfach lösbar, wenn es eine Art Referenzmatrix, die das Referenzverhalten von Transaktionstypen und Datenbereichen beschreibt, gibt, wie wir sie in Bild 3.3 dargestellt haben. Bild 3.3 zeigt eine 'ideale Last' insofern, daß sich sehr leicht eine Partitionierung und ein daran angepaßtes Routing finden läßt. Die schraffierten Bereiche geben an, in welche Datenbereiche die Referenzen der einzelnen Transaktionstypen fallen (z.B. TA-Typ T1 und T2 referenzieren Area A1 und A2). Geht man davon aus, daß die Gesamtzahl der Referenzen einigermaßen gleichmäßig über die vier ’grauen’ Bereiche aufgeteilt sind, so ist eine Daten- und Lastverteilung, die sogar vollkommen ohne externe Referenzen auskommt, für eine Mehrrechner-Konfiguration mit 4 VAR einfach anzugeben: VAR1 bearbeitet T1 und T2 und erhält Area A1 und A2; VAR2 bearbeitet T3, T4 und T5 und erhält A3 bis A5, und so fort..

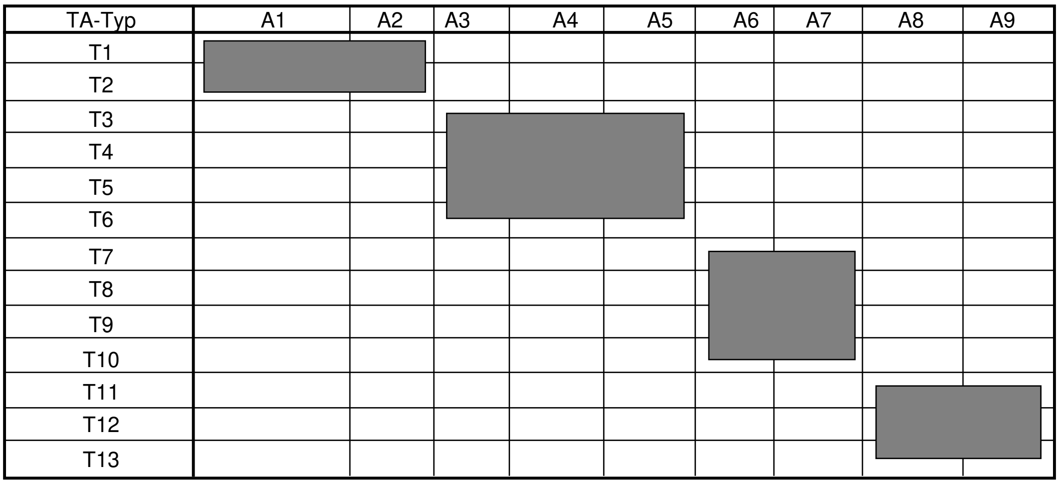

Bild 3.3: Referenzmatrix einer für 4 Verarbeitungsrechner ideal partitionierbaren Last

Bild 3.3 ähnelt aber überhaupt nicht den in Tabelle 3.1 dargestellten Realitäten:

- Es existieren dominante Referenzierungsbereiche, d.h., es gibt Datenbereiche, die die Majorität der Seitenreferenzen auf sich vereinen (Area 3 in Tab. 3.1).
- Es existieren im allgemeinen mehrere Datenobjekte mit extremer Referenzierungs- und Konflikthäufigkeit (’Hot Spots’, ’High Traffic Objects’); dies wird erst durch eine hier nicht gezeigte Detailanalyse der Daten deutlich; es handelt sich um ein wohlbekanntes Phänomen, für dessen Entschärfung bereits Vorschläge existieren /Hä88/.

Bis auf seltene Sonderfälle kann man sagen, daß es derartige ideal partitionierbare Lasten nicht gibt - tatsächlich gilt dies für alle von uns untersuchten Anwendungen.

Bei unseren bisherigen Betrachtungen über das Referenzierungsverhalten wurden zwei sehr wichtige Aspekte völlig außer Acht gelassen, die wir jetzt berücksichtigen müssen.

- **Die Zeit:**
  Wir haben Referenzmatrizen angegeben, die sich auf die gesamte Protokollierungsdauer der Referenzstrings beziehen. Die Aussage, daß aufgrund dieser Werte nicht ’ideal partitioniert’ werden kann (im Sinne der Minimerung externer Datenzugriffe), bedeutet deshalb auch nur, daß dies für die untersuchte Zeitspanne nicht möglich ist. Es könnte durchaus sein, daß kürzere Zeitspannen das Gewünschte liefern. Die Annahme hierbei ist, daß durch das Senken der Beobachtungsdauer die innerhalb des betrachteten Zeitraums aktiven Transaktionstypen disjunkte - und nicht zusammenhängende - Datenbereiche gleichmäßiger referenzieren.
- **Das Granulat:**
  Die angegebene Matrix hat als Datenobjekte ganze Areas und Transaktionstypen als ’Bemaßungseinheiten’. Diese wurden deshalb gewählt, weil sie in einem allerersten Ansatz in unserer Umgebung das am einfachsten nutzbare Maß waren. Wir haben gesehen, daß wir damit viel zu grobe Maße eingeführt haben; kleinere Datengranulate erhöhen - ähnlich wie kürzere Zeitspannen - die Wahrscheinlichkeit, eine ’ideale’ Partitionierung finden zu können. Deshalb betrachten wir bei den im fol-

genden dargestellten Analysen Seitenmengen ('Working-Sets') einzelner Transaktionstypen anstatt Areas.

Will man Algorithmen entwickeln, die durch geschickte Partitionierung und sinnvolles Transaktionsrouting die Zahl der externen Datenzugriffe in Mehrrechner-Architekturen minimieren, so haben wir bisher lediglich festgestellt, daß die Area als Datengranulat für die Partitionierung der Datenbank eine viel zu grobe Einheit darstellt und die Zeitspanne, die wir berücksichtigt haben, viel zu grob ist.

Doch wie klein muß die Zeitspanne sein, damit Lastkontrollverfahren sinnvolle Ansatzpunkte haben, und wie groß muß sie sein, daß diese Verfahren nicht durch laufend neue Berechnungen einen zu großen Overhead erzeugen? Erst wenn diese Fragen beantwortbar sind, kann man damit beginnen, Algorithmen zu implementieren und in Simulationen zu validieren. Deshalb haben wir die in 3.4 vorgestellten laufzeitintensiven Untersuchungen über das 'Working-Set-Verhalten' der Transaktionstypen angestrengt.

3.4 Lokalitätsverhalten der Transaktionstypen

Lastkontrollverfahren, die durch Zuordnung von (Teil-)Lasten zu den Verarbeitungsrechnern eines Mehrrechner-Transaktionssystems die Zahl der externen Datenreferenzen minimieren, können folgende Informationen besonders effektiv nutzen:

- Transaktionstypen referenzieren denselben Datenbereich. Transaktionen dieser Typen sind gemeinsam demjenigen Verarbeitungsrechner zuzuordnen, der den betreffenden Datenbestand verwaltet (Zusammenfassen von Transaktionstypen zu sog. 'Workload-Groups', betrifft die Zahl der Zeilen der Referenzmatrix).
- Transaktionstypen referenzieren disjunkte Datenbereiche. Transaktionen dieser Typen können relativ frei zusammen mit den betreffenden Datenbereichen auf die Rechner des Systems aufgeteilt werden.

Ob Transaktionstypen zu bestimmten Zeitintervallen disjunkte oder überlappende Datenbereiche referenzieren, hängt von der Beobachtungsdauer ab. Wir haben deshalb eine Working-Set-Analyse mit den 'Fenstergrößen' 60, 120, 180, 300, 600 und 900 Sekunden durchgeführt, wobei wir die Referenzierungszeitpunkte aus den Referenzstrings verwendet haben. Die Tatsache, daß es nicht notwendig war, Untersuchungen mit Fenstergrößen kleiner als 60 Sekunden durchzuführen, nimmt das wichtigste Ergebnis dieser Untersuchungen in gewisser Weise vorweg; wir haben 'schon' bei einer recht großen Beobachtungsdauer überraschend brauchbare Ergebnisse erhalten.

Unser Interesse war es, festzustellen, bei welchen Fenstergrößen überhaupt ein von Lastkontroll-Algorithmen nutzbares Optimierungspotential vorhanden ist. Dazu haben wir untersucht, ob die von den verschiedenen Transaktionstypen referenzierten Seitenmengen disjunkt, gleich bzw. überlappend waren - in Abhängigkeit von der jeweiligen Fenstergröße. Konkret haben wir uns dafür interessiert, wie oft Situationen wie 'TA-Typen referenzieren im aktuellen Fenster überlappende Seitenbereiche' (analog: gleiche bzw. disjunkte Bereiche) vorkommen. Hierzu haben wir aus Sicht jedes einzelnen Transaktionstyps gezählt, in wievielen Fenstern diese Eigenschaften erfüllt waren und den Anteil f an der Gesamtzahl der betrachteten Fenster aufgelistet ('fensterbezogene Auswertung'). Weil damit die Zahl der tatsächlich ausgeführten Referenzen unberücksichtigt blieb, wurde eine Gewichtung nach Referenzen eingeführt: Wir haben die Zahl r der in Fenstern mit disjunkten, gleichen oder überlappenden Datenbereichen vorkommenden Referenzen gezählt und diese durch die Gesamtzahl der derart betrachteten Referenzen dividiert ('referenzenbezogene Auswertung'). An diese verbale und unscharfe Beschreibung der Werte f und r wollen wir eine präzise Beschreibung anschließen:

$$f_x := \left(\frac{\sum_T (\text{\#Fenster mit Bedingung } F_T^x)}{\text{\#Fenster mit Bedingung } F} \right) * 100$$

$$r := \frac{\sum_T (\text{\#Referenzen in Fenstern mit Bedingung } F_T)}{\text{\#Referenzen in Fenstern mit Bedingung } F}$$

wobei gilt:

T,S ∈ {Transaktionstypen}, sowie x ∈ {disjunkt, gleich, überlappend}

F_T^x gilt genau dann, wenn ein Transaktionstyp S≠T existiert, der im gleichen Zeitfenster eine Seitenmenge referenziert, die zur referenzierten Seitenmenge von T die Eigenschaft x hat.

F gilt genau dann, wenn ein Transaktionstyp S≠T existiert, der im gleichen Zeitfenster eine Seitenmenge referenziert, die zur referenzierten Seitenmenge von T disjunkt oder gleich oder überlappend ist.

Die von uns ermittelten Ergebnisse sind in den Tabellen 3.3 (f) und 3.4 (r) dargestellt. Dabei werden für jeden Referenztrings drei Spalten von Zahlen angegeben, die sich von links nach rechts auf disjunkte, gleiche bzw. überlappende Seitenmengen beziehen. Zusätzlich enthält Tabelle 3.5 die durchschnittliche Größe der Working-Sets. Fehlende Zahlen am Tabellenende resultieren aus der Aufzeichnungsdauer der Referenzstrings, die für große Zeitfenster teilweise unzureichend war; dies jedoch ist kein Mangel, wenn man sich die Motivation zu diesen Untersuchungen in Erinnerung ruft.

Fenstergröße	DOA			DOD			KD		
	disjunkt	gleich	überl.	disjunkt	gleich	überl.	disjunkt	gleich	überl.
60s	60.47	20.93	16.28	52.38	33.33	9.52	46.67	0	53.33
120s	57.14	16.67	21.43	50.00	25.00	15.00	24.14	0	75.86
180s	56.10	14.63	26.83	52.63	21.05	15.79	10.71	0	89.29
300s	53.85	12.82	30.77	52.94	17.65	23.53	3.85	0	96.15
600s	50.00	8.82	38.42	50.00	8.33	33.33	0	0	100
900s	44.83	6.90	44.83					0	100

Tabelle 3.3: Durchschnittlicher Anteil (%) der Fenster mit in Bezug auf Referenzen der verschiedenen Transaktionstypen disjunkten, gleichen bzw. überlappenden Working-Sets (Werte 'f')

Fenstergröße	DOA			DOD			KD		
	disjunkt	gleich	überl.	disjunkt	gleich	überl.	disjunkt	gleich	überl.
60s	0.571	0	0.492	0.668	0	0.332	0.449	0	0.551
120s	0.503	0	0.497	0.613	0	0.387	0.231	0	0.769
180s	0.461	0	0.593	0.581	0	0.419	0.098	0	0.902
300s	0.424	0	0.576	0.524	0	0.476	0.690	0	0.310
600s	0.381	0	0.619	0.430	0	0.566	0	0	1
900s	0.345	0	0.655				0	0	1

Tabelle 3.4: Nach ihrer Gesamtzahl normierter Anteil von Referenzen 'in' Fenstern mit in Bezug auf Referenzen der verschiedenen Transaktionstypen disjunkten, gleichen bzw. überlappenden Working-Sets (Werte 'r')

Fenstergröße	DOA	DOD	KD
60s	84.77	40.67	249.60
120s	160.51	75.06	460.43
180s	223.12	101.39	637.41
300s	339.90	148.96	964.08
600s	608.48	252.94	1707.26
900s	841.86		2456.48

Tabelle 3.5: Durchschnitliche Größe des Working-Sets (Anzahl verschiedener referenzierter Seiten).

Die Analyse der Werte liefert neben erwarteten Resultaten, die für die Korrektheit unserer Auswertungen sprechen, auch Überraschendes:

- Mit wachsender Fenstergröße wächst die Größe des Working-Sets der referenzierten Seiten teilweise beachtlich (bei KD beispielsweise insgesamt etwa um den Faktor 10).
- Die Summe der Werte f, die sich auf disjunkte und gleiche Seitenmengen beziehen und damit Algorithmen einen großen Zuordnungsspielraum ermöglichen (siehe oben), ist zwar mit wachsender Fenstergröße sinkend, aber dennoch überraschend groß. Dies gilt vor allem für die Anwendungen DOA und DOD mit einer recht großen Zahl von Transaktionstypen. Selbst bei KD, wo die Werte für f und r über disjunkte Working-Sets mit wachsender Fenstergröße stark abfallen (0 bei 900 s), sind bei 60s-Fenstern recht hohe Werte (f=56.67%, r=0.449) festgestellt worden.
- Die Werte r betonen mit zunehmender Fenstergröße die überlappenden Datenbereiche, doch bei DOD und DOA liegen bei einer Fenstergröße von 5 Minuten noch 40% der Referenzen in disjunkten Referenzierungsbereichen; eine analoge Aussage ist bei KD allerdings nur bis zu 1 Minute möglich.
- Es gibt zwar Fenster, in denen die referenzierten Seitenmengen gleich sind, dies liegt jedoch ausschließlich daran, daß es sich hierbei um leere Mengen ohne Referenzen handelt (vergleiche r und f in der jeweils mittleren Spalte der Tabellen 3.3 und 3.4).

Das Wichtige und Überraschende an diesen Resultaten ist der unerwartet hohe Anteil von Referenzen in disjunkten Datenbereichen von teilweise mehr als 50% und die Tatsache, daß diese Ergebnisse im Bereich von 1 bis 5 Minuten Fenstergröße liegen.

Welche Folgerungen aus diesen und den zuvor dargestellten Resultaten abzuleiten sind, diskutieren wir in Kapitel 4.

4. Zusammenfassung und Folgerungen

Zunächst fassen wir unsere Erkenntnisse kurz zusammen, um die in der Einleitung formulierten Fragen zu beantworten:

- Die Zusammensetzung der Transaktionslast ist **heterogen**.
- Die Änderungen der Systembelastung sind **sehr häufig und intensiv**.
- Die Datenbank wird, nimmt man Beobachtungsspannen von mehr als 10 Minuten an, **sehr ungleichmäßig referenziert**.
- Das Lokalitätsverhalten der Transaktionstypen bietet **Optimierungspotential** bei Zeitfenstern im Bereich von einigen Minuten.

Einfache Partitionierbarkeit realer Lasten im Hinblick auf Nutzung referentieller Lokalität ist - wie wir gesehen haben - im allgemeinen nicht gegeben. Um bei Mehrrechner-Architekturen eine hohe Leistungsfähigkeit zu erzielen, ist der Einsatz dynamischer Lastkontrollmechanismen erforderlich, die auf den aktuellen Systemzustand und die aktuelle Systemlast reagieren. Daß statische Verfahren nicht ausreichen, wurde z.B. in /Bo88/ nachgewiesen. Diese Tatsache wurde durch die hier vorgestellten Ergebnisse erklärt und untermauert: die Verfahren müssen schnell und wirksam auf Änderungen im Lokalitätsverhalten der Transaktionslast eingehen, denn Lastschwankungen sind häufig und heftig. Wie genau dynamische Lastkontrollverfahren Änderungen beobachten müssen, d.h., in welchen - aus Effizienzgründen möglichst langen - Zeitspannen Optimierungspotential vorhanden ist, wurde in Kapitel 3.4 beantwortet: wir haben gezeigt, daß bei Fenstergrößen von wenigen Minuten überraschend gute Einsatzmöglichkeiten für derartige Algorithmen vorhanden sind.

Diese Aussagen gelten für alle von uns untersuchten Anwendungen - nicht nur für die hier vorgestellten. Die Aufzeichnung der Referenzstrings verteilt sich auf einen Zeitraum von mehr als fünf Jahren. Da sie sich in den hier untersuchten Aspekten nicht unterscheiden, kann man annehmen, daß sie für ein sehr breites Anwendungsspektrum gültig sind. Es ist deshalb sinnvoll, sie mit den Eigenschaften des Debit/Credit-Benchmarks /An85/, der zur Bewertung von Hochleistungs-Transaktionsystemen herangezogen wird, zu vergleichen. Wir können hier nicht näher auf Implementierungsdetails eingehen und wollen neben den bereits in der Einleitung vorgestellten Fakten einfach die wesentlichen Eigenschaften des Benchmarks auflisten:

- Die Transaktionen sind sehr kurz und einfach.
- Die Datenbank ist aufgrund der vorgegebenen 'Organisationsstruktur' einfach auf beliebige Granulate hin partitionierbar, denn beispielsweise werden ganze Bereiche von Kontonummern Geschäftsstellen fest zugeordnet.
- Innerhalb der Partitionen sind die Referenzen relativ gleichmäßig verteilt.
- Durch geschickte Implementierung werden Synchronisationskonflikte vollkommen vermieden.
- Hot-Spots können entschärft werden.

Diese Eigenschaften stehen im krassen Widerspruch zu unseren Ergebnissen, denn die Lastzusammensetzung des Benchmarks ist homogen: er kennt nur einen einzigen extrem einfachen Transaktionstyp, und Lastschwankungen kommen in der von uns festgestellten Form nicht vor. Das Referenzierungsverhalten ist bekannt und eignet sich hervorragend, den Datenbestand gleichmäßig unter den Verarbeitungsrechnern mit etwa gleichen Lastanteilen aufzuteilen. Deshalb stellt er eine ideal partitionierbare und damit **realitätsfremde Anwendung** dar. Reale Lasten zeichnen sich durch eine Vielzahl sehr unterschiedlicher Transaktionstypen aus, die mit verschiedener Häufigkeit die Datenbank 'ungleichmäßig' referenzieren. Hinzu kommen häufige starke Lastschwankungen. Hieraus resultierende Probleme bei der Nutzung des (trotzdem vorhandenen - siehe Kapitel 3.4) Lokalitätsverhaltens der Transaktionslast zur Minimierung externer Datenzugriffe treten im Debit/Credit-Benchmark nicht auf. Zukünftige Hochleistungs-Transaktionssysteme *brauchen* jedoch Komponenten, die auf die Beherrschung heterogener Lasten abgestimmt sind; derartige Probleme stellen sich jedoch mit Benchmarks wie Debit/Credit gar nicht.

Das Maß TPS ist demnach nur geeignet, Transaktionssysteme hinsichtlich eines sehr schmalen Spektrums leistungskritischer Algorithmen zu vergleichen, das die eigentlichen, durch reale Lasten gestellten Anforderungen nur zu einem geringen Teil abdeckt. Zieht man demnach diesen Benchmark oder ähnliche 'Monokulturen' zur Bewertung von Hochleistungs-Transaktionssystemen oder in der Entwicklung befindlicher Systeme heran, so werden mit hoher Wahrscheinlichkeit die unrealistischen Anwendungsfälle bewertet bzw. optimiert.

Ein Architekturvorschlag für ein Hochleistungs-System, der versucht, durch heterogene Lasten entstehende Probleme gezielt anzugehen, ist in /HR86, RS84, Ra88/ als 'DB-Sharing'-Architektur vorgestellt worden. Dadurch, daß in dieser Architektur alle Verarbeitungsrechner prinzipiell auf die gesamte Datenbank zugreifen, spezielle Synchronisations- und Lastkontrollverfahren auf sich in unseren Ergebnissen widerspiegelnde Lastcharakteristika eingehen und sich sogar das in Kapitel 3.4 vorgestellte Optimierungspotential auch bei chronologischen Veränderungen im Bereich von einigen Minuten erschließen, ist DB-Sharing vermutlich der vielversprechendste Ansatz, die Forderungen nach hoher Leistungsfähigkeit, extremer Verfügbarkeit und modularem Wachstum zu erfüllen.

Literaturverzeichnis:

An85 Anon et. al.: A Measure of Transaction Processing Power, Datamation, February 1985, pp. 112-118.

Bo88 Bohn, V.: Untersuchung des Leistungsverhaltens von DB-Distribution-Systemen durch diskrete Ereignis-Simulation, ZRI-Bericht 1/88, Universität Kaiserslautern, Fachbereich Informatik, 1988.

Gr85 Gray, J.N., Good, B., Gawlick, D., Homan P., Sammer, H.: One Thousand Transactions per Second, Proc. IEEE Spring Compcon, San Francisco, Feb. 1985, pp. 96-101.

Hä88 Härder, T.: Handling Hot Spot Data in DB-Sharing Systems, Information Systems, Vol. 13, No. 2, 1988, pp. 155-156.

HR83 Härder, T., Reuter, A.: Principles of Transaction-Oriented Database-Recovery, ACM Computing Surveys, Vol. 15, No. 4, 1983, pp. 287-317.

HR86 Härder, T., Rahm, E.: Mehrrechnerdatenbanksysteme für Transaktionssysteme hoher Leistungsfähigkeit, Informationstechnik, Jhrg. 28, Heft 4, 1986.

LYL88 Lee, Y.-H., Yu, P.S., Leff, A.: Robust Transaction Routing in Distributed Database Systems, Proc. Int. Symp. on Databases in Parallel and Distributed Computing, Austin, Texas, Dec. 1988, pp. 210-219.

MW88 Meyer-Wegener, K.: Transaktionssysteme, Teubner-Verlag, Stuttgart, 1988.

Ra86 Rahm, E.: Algorithmen zur effizienten Lastkontrolle in Mehrrechner-Datenbanksystemen, Angewandte Informatik4/86, S. 161-169.

Ra88 Rahm, E.: Synchronisation in Mehrrechner-Datenbanksystemen - Konzepte Realisierungsmöglichkeiten und quqntitative Bewertung, Dissertation, Universität Kaiserslautern, Fachbereich Informatik, 1988.

RS84 Reuter, A., Shoens, K.: Synchronization in a Data Sharing Environment, Technischer Bericht (vorläufige Fassung), IBM Almaden Research Lab., San Jose, 1984.

UDSa Aufbauen und Umstrukturieren, Benutzerhandbuch UDS V5.0C, Siemens AG, Best.-nr. U931-J-Z55-6, 1988.

UDSb Anwendungen programmieren, Benutzerhandbuch UDS V5.0A, Siemens AG, Best.-nr. U930-J-Z55-3, 1988.

Dieser Bericht wurde durch die Siemens-AG (Abteilung D-ST-DB-1) gefördert.