

Zusammenfassung

Das Zugriffszeitverhalten von Relationalen Datenbanksystemen wird unter Einbeziehung realer Speichermedien (Magnetplatten) in einer deterministischen "worst case"-Analyse untersucht. Dabei wird die Debatte "navigierenden versus deskriptive Datenmanipulationssprachen", die in der Fachwelt beträchtliche Meinungsverschiedenheiten auslöste, auf der Zugriffsebene unter dem Stichwort "satzweiser und paralleler Zugriff" fortgesetzt. Es stellt sich heraus, daß die Einbettung der Zugriffslogik in das Anwendungsprogramm beim Aufsuchen von Mengen von Datensätzen große Leistungsverluste mit sich bringt. Im Vergleich dazu kann bei Delegation der Zugriffsverantwortlichkeit an das Datenverwaltungssystem mit maximalen Verbesserungen um den Faktor 7 gerechnet werden. Die strikte Trennung von deskriptiver Sprache und Zugriffsaspekten erhöht die Dispositionsfreiheit des Datenverwaltungssystems und erlaubt zu jedem Zeitpunkt, optimale Zugriffspfade auszuwählen.

1. Einleitung

Das Relationenmodell nach E. F. Codd (Co70) ist gekennzeichnet durch explizite Relationendarstellung, die sich auf der Schnittstelle zum Benutzer (Ebene der logischen Datenstrukturen) auf die Aggregation von Attributnamen beschränkt und keinerlei Information über ihre physische Realisierung beinhaltet, und durch deskriptive Datenbankteilsprachen (Co71, Ch74). Die Vollständigkeit des Relationenmodells auf der logischen Ebene läßt eine von bestimmten Speicherungsstrukturen und Zugriffspfaden unabhängige Implementierung zu, d. h., es brauchen keine zusätzlichen Informationen zur Ergänzung des logischen Datenmodells durch Verfolgung von Zugriffspfaden abgeleitet werden. Durch diese Charakteristika bietet es Gewähr für einen hohen Grad an Datenunabhängigkeit.

Hierarchische Datenmodelle und Netzwerkmodelle (IBM1, CODA) dagegen erfüllen die wichtige und weitreichende Forderung nach Vollständigkeit des Datenmodells auf logischer Ebene nicht. Bestimmte Beziehungen zwischen den darzustellenden Entities (Objekten) werden durch die Wahl geeigneter Speicherungsstrukturen oder Implementierungstechniken repräsentiert (z. B. Pointer, Wie-

derholungsgruppe, physische Nachbarschaft). Folge davon ist die Notwendigkeit prozeduraler Datenmanipulationssprachen, die dem Anwender unnötiges Detailwissen über vorhandene Zugriffspfade aufbürdet. Beim Navigieren durch die Datenstrukturen (Ba73) muß er zu jeder Zeit wissen, woher er kommt und wo er sich befindet. Er bestimmt den Weg durch die Datenstrukturen und wählt satzweise die gesuchten Datensätze aus ("one record at a time"-Logik). Diese starke logische Bindung der Datenstruktur an das Benutzerprogramm hat schwerwiegende Folgen für die Datenunabhängigkeit und die Konsistenz der Daten.

Es soll aufgezeigt werden, welche Zugriffsmöglichkeiten in Relationalen Datenbanksystemen vorhanden sind und in welcher Weise das Auswahlvermögen deskriptiver Sprachen, das auch als "multi-record at a time"-Logik bezeichnet wird, durch das Datenbanksystem optimal zu nutzen ist. Durch Vergleich der Zugriffszeiten beim Zugriff auf Mengen von Datensätzen werden mögliche Leistungsverbesserungen gegenüber dem satzweisen Zugriff navigierender Sprachen diskutiert.

2. Datenmodell und Speicherungsstruktur

Wir gehen vom Relationenmodell nach E. F. Codd aus, durch das sich die logischen Datenstrukturen einer Datenbank beschreiben lassen. Danach ist eine n -stellige normalisierte Relation $R(A_1, A_2, \dots, A_n)$ als Untermenge des Cartesischen Produktes über ihre Definitionsbereiche A_i definiert:

$$R(A_1, A_2, \dots, A_n) \subseteq A_1 \times A_2 \times \dots \times A_n$$

Die Mengen A_1, A_2, \dots, A_n müssen nicht notwendigerweise disjunkt sein. Die Namen der Mengen sind Attributnamen und deren Elemente Attribut-Werte. Eine n -stellige Relation heißt normalisiert, wenn alle Bereiche der Relation einfach sind, d. h., wenn keiner ihrer Bereiche wiederum eine Relation ist. Eine n -stellige Relation läßt sich als eine Menge von n -Tupeln, für die Eindeutigkeit gefordert wird, darstellen. Ein Attribut oder eine minimale Gruppe von Attributen, durch die Eindeutigkeit der Tupeln gewährleistet wird, heißt Primärschlüssel der Relation. Die physische Ausprägung einer Relation in der Ebene der Speicherungsstrukturen wird als Datei, die eines Tupels als Satz fester Länge bezeichnet.

Wir nehmen an, daß N_{REC} Datensätze der Datei in aufeinander folgenden Zylindern einer Magnetplatte gleichverteilt gespeichert sind und daß Zugriffe zu allen Datensätzen bei den Transaktionen mit gleicher Häufigkeit erfolgen. Wedekind nennt dies die innere und äußere Bedingung der Datenorganisation (We73). Beim wahlfreien Zugriff führen diese beiden Annahmen auf die für die Zugriffsbewegung ungünstigsten Bedingungen.

3. Zugriffspfadmodell und zugehörige Implementierung

Für unser Datenmodell lassen sich zwei Klassen von Zugriffspfaden unterscheiden. Wird ein Primärschlüssel als Suchargument benutzt, so führt der dazugehörige Zugriffspfad auf höchstens einen Datensatz. Dagegen qualifiziert ein Nicht-Schlüsselattribut (Sekundärschlüssel) n Datensätze. Den entsprechenden Zugriffspfad für das direkte Aufsuchen solcher Mengen von Datensätzen nennen wir auch sekundären Index.

Ein Zugriffspfad für den Primärschlüssel resultiert vor allem aus der Notwendigkeit der wirksamen Unterstützung von Stapelanwendungen. Die drei Verarbeitungsprimitive

- wahlfreie Verarbeitung eines bestimmten Satzes,
- fortlaufende Verarbeitung aller Sätze nach einer vorgegebenen Ordnung,
- Änderungsdienst

sind bei seiner Implementierung zu berücksichtigen. Deshalb wird der Zugriffspfad des Primärschlüssels oft durch eine bestimmte Speicherungsstruktur repräsentiert. In großen Datenbanksystemen, in denen die Datensätze auf externen Speichermedien mit halbdirektem Zugriff gespeichert sind, kommen als Speicherungsstrukturen, die den Primärschlüsselzugriff unterstützen, nur gestreute Speicherungsstrukturen und Mehrwegbäume in Frage (Hä75). Zur Unterstützung der Bearbeitung von ad hoc-Anfragen (Queries) werden sekundäre Indizes herangezogen. Ein Index für das i -te Attribut von R ist eine Abbildung von Elementen in A_i auf solche Tupeln in R , die dieses Element als i -ten Attributwert besitzen, d. h. eine Abbildung

$$I_i : A_i \rightarrow 2^R.$$

Für das Attribut A_i der Relation R seien j_i Attributwerte mit den Häufigkeiten N_1, N_2, \dots, N_{j_i} vorhanden, wobei

$$\sum_{k=1}^{j_i} N_k = N_{\text{REC}}$$

gilt. Zur Beschreibung der Verteilungsfunktion für die Häufigkeiten der einzelnen Attributwerte des Attributes A_i ziehen wir den Auflösungsfaktor nach C. P. Wang (Wa73)

$$Rf_i = \frac{N_{\text{REC}}}{j_i} \quad (3.1)$$

als Erwartungswert heran. Diese Annahme unterstellt eine Gleichverteilung der Attributwerte.

Die Vorteile deskriptiver Datenbanksprachen für relationale Datenbanksysteme, die Datensätze aufgrund ihres Inhaltes und nicht aufgrund ihrer relativen Position innerhalb einer Speicherungsstruktur qualifizieren und somit ein hohes Auswahlvermögen besitzen, sollten durch geeignete Implementierungstechniken für die Zugriffspfade gewahrt werden. Durch Trennung von Zugriffspfaden und Primärdaten ist die Teilmengenbildung der durch eine Frage qualifizierten Sätze allein in den Sekundärdaten möglich. Das System kann dann den günstigsten Weg auswählen, die gesuchten Sätze durch Zugriffe zu den Primärdaten bereitzustellen. Bei Vermischung von Primär- und Sekundärdaten hat das System nur geringe Möglichkeiten, die Auswertung von Fragen zu optimieren, da die Zugriffsfolge durch die Speicherungsstruktur vorgeschrieben wird.

Wenn ein Bereich einer Relation invertiert ist, so nennen wir das zugehörige Attribut auch Deskriptor. Wir nehmen an, daß die Invertierung durch Indextabellen- oder Bitlistentechnik erfolgt (Hä74) und somit alle Zugriffspfadinformationen unabhängig von den Datensätzen der Relation gespeichert sind. Die Zielverweise eines Deskriptorwertes zu den Datensätzen stehen in einer Zielliste variabler Länge, die in einem oder mehreren physischen Sätzen auf einem externen Speichermedium untergebracht ist. Der Zugriff zur Zielliste ist durch eine dreistufige Hierarchie, wie sie schematisch in Bild 1 dargestellt ist, organisiert.

Unabhängig voneinander können alle Hierarchiestufen durch Tabellen, B-Bäume oder B^* -Bäume implementiert und für sich optimiert werden. Das Mitführen der Anzahl der jeweils zugehörigen untergeordneten Elemente auf allen Hierarchiestufen durch Summenfelder gestattet die Durchführung von Kontrollen und die Beantwortung bestimmter statistischer Fragen, ohne auf die Datensätze der Relation zugreifen zu müssen.

Die Tabellen mit den Relationen- und Deskriptoreintragungen können wegen ihrer begrenzten Anzahl im allgemeinen kernspeicherresident gehalten werden. Die Deskriptorwerteintragungen dagegen sind im allgemeinen wegen ihres beträchtlichen Umfanges auf ein externes Speichermedium ausgelagert. Zur Abschätzung der Anzahl der Zugriffe auf den Deskriptorwertindex nehmen wir an, daß dieser als B-Baum organisiert ist. Die Höhe h des B-Baumes für einen Deskriptorwertindex, die wir als Maß für den Zugriffsaufwand zur Lokalisierung einer Zielliste heranziehen, wird im wesentlichen von der physischen Blocklänge B und der Länge eines Deskriptorwert-Eintrages D bestimmt. Die Blocklänge

ist in modernen Betriebssystemen als Größe einer Speicherseite mit 1,2 oder 4 K Bytes vorgegeben. D dagegen liegt in der Größenordnung von B/100. Folglich erhalten wir zur Beschreibung der Anzahl der notwendigen Zugriffe zum Deskriptorwertindex des i-ten Deskriptors folgende Funktion:

$$h_i = \begin{cases} 1 & \text{für } j_i \leq \frac{B}{D} \approx 10^2 \\ 2 & \text{für } \frac{B}{D} < j_i \leq \left(\frac{B}{D}\right)^2 \approx 10^4 \\ 3 & \text{für } \left(\frac{B}{D}\right)^2 < j_i \leq \left(\frac{B}{D}\right)^3 \approx 10^6 \end{cases} \quad (3.2)$$

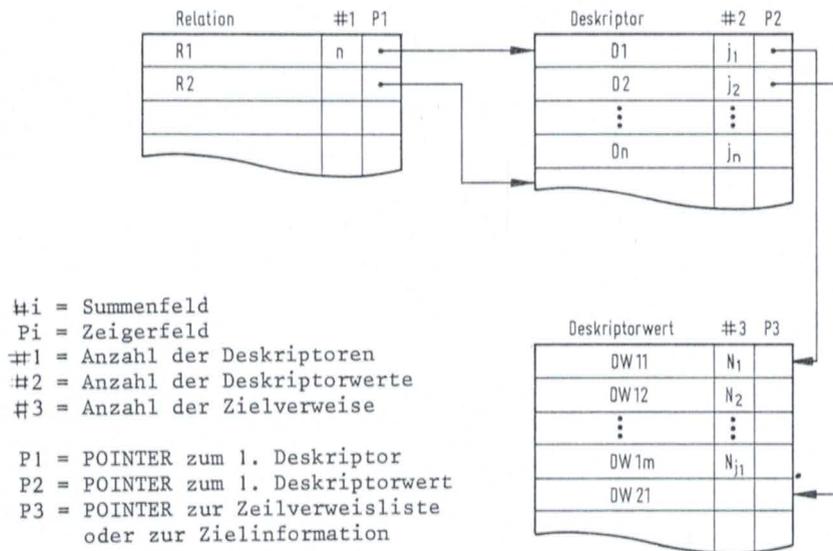


Bild 1: Schematischer Aufbau des Katalogs mit Sekundärschlüssel

Nach Beziehung (3.1) hat eine Zielliste für einen Deskriptorwert des i-ten Deskriptors Rf_i Zielverweise, die bei Indextabellen als logische Zeiger, Primärschlüssel oder physische Satzadressen ausgeprägt sein können. Im Falle von komprimierten Bitlisten kann die Markierung nach logischen oder physischen Kriterien erfolgen. Der Speicherplatzbedarf für einen Zielverweis (Markierung) S_M ist bei Indextabellen konstant, bei komprimierten Bitlisten eine Funktion von j_i (Hä74). Eine Zielliste ist folglich in

$$N_{B_i} = \left\lceil \frac{N_{REC} \cdot S_M}{j_i \cdot B} \right\rceil \quad (3.3)$$

physischen Blöcken abgespeichert.

Die Wirksamkeit von Suchoperationen hängt wesentlich von der Anzahl der Zugriffe auf externe Speichermedien ab, die zur Verfolgung und Auswertung von Zugriffspfaden nötig sind. Sind die einzelnen Attribute, die zur Qualifikation in einer Anfrage herangezogen werden, invertiert, so läßt sich eine Zielliste mit den Primärschlüsseln oder Satzadressen der qualifizierten Datensätze ableiten (Hä75). Dazu sind gemäß dem in Bild 1 angegebenen Katalogschema die Ziellisten für die einzelnen Attributwerte aufzusuchen, in den Kernspeicher zu übertragen und entsprechend den verwendeten Verknüpfungsoperatoren zu mischen. Nach der daraus resultierenden Zielliste $Z(L)$ erfolgt der Zugriff zu den Datensätzen. Fehlen für Attribute sekundäre Indizes, so ist es in bestimmten Fällen möglich, eine Zielliste $O(L)$, die eine Obermenge zu den gesuchten Tupeln umfaßt, zu bilden. Die so gefundenen Datensätze müssen nach ihrer Bereitstellung im Kernspeicher gemäß der Fragebedingung überprüft werden. Läßt sich die Menge der Datensätze durch vorhandene, den Prädikaten der Frage entsprechende Zugriffspfade nicht einschränken, so muß der gesamte Datenbestand fortlaufend durchsucht werden.

4. Modelle für die wahlfreie Verarbeitung

Der Aufwand für die wahlfreie Verarbeitung einer Transaktion hängt wesentlich davon ab, wie viele Datensätze (N_{qual}) bereitzustellen sind. Zu seiner Beschreibung führen wir die Trefferrate einer Transaktion durch

$$TR = \frac{N_{\text{qual}}}{N_{\text{REC}}} \cdot 100 \% \quad (4.1)$$

ein. Falls ein Prädikat einer Anfrage nur einen Attributwert umfaßt, entspricht N_{qual} dem Auflösungsfaktor R_f . Unter Berücksichtigung der Beziehung (3.1) erhalten wir als Erwartungswert der Trefferrate einer solchen Transaktion

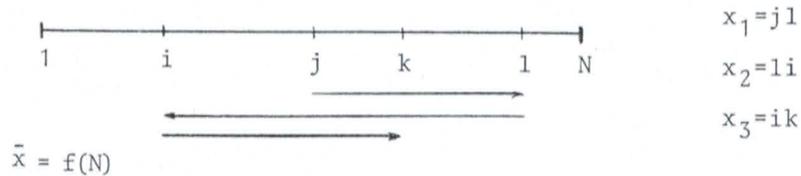
$$TR_i = \frac{100}{J_i} \%$$

Der Gesamtaufwand für den wahlfreien Zugriff zu den qualifizierten Datensätzen einer Transaktion wächst mit der Trefferrate, während er bei der fortlaufenden Verarbeitung konstant bleibt. Daraus ergibt sich, daß die wahlfreie Verarbeitung ab einer bestimmten Trefferrate (Grenztrefferrate TR_0) nicht mehr vorteilhaft ist. Es ist auch dann, wenn ein Zugriffspfad existieren sollte, fortlaufende Verarbeitung vorzuziehen.

4.1 Wahlfreie Verarbeitung nach ungeordneten und sortierten Listen

Wir unterscheiden beim wahlfreien Zugriff auf die qualifizierten Datensätze zwei Fälle:

- 1) Die Zugriffsbewegungen x_i werden nicht koordiniert. Es erfolgt ein gleichverteilter Zugriff in zufälliger Reihenfolge. Die mittlere Zugriffsbewegung \bar{x} ist eine Funktion der Anzahl der zusammenhängend belegten Zylinder N.

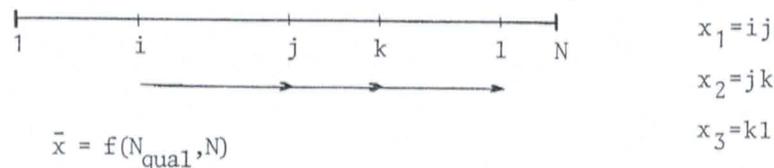


Daraus resultiert eine mittlere Zugriffszeit \bar{t}_u die unabhängig von der Trefferrate für jeden Datensatz gleich ist. Wir bezeichnen dies eine wahlfreie Verarbeitung nach ungeordneten Listen.

Als mittlere Zugriffszeit für eine solche Transaktion erhalten wir

$$\bar{t}_{TR} = N_{qual} \cdot \bar{t}_u \tag{4.2}$$

- 2) Da vor der wahlfreien Verarbeitung einer Transaktion dem Datenverwaltungssystem alle Satzadressen bekannt sind, kann durch Sortierung die Zugriffsfolge zum externen Speichermedium so festgelegt werden, daß die Zugriffsbewegungen nur in einer Richtung erfolgen und so minimal werden. Die mittlere Zugriffsbewegung ist eine Funktion von N und N_{qual} .



Die mittlere Zugriffszeit \bar{t}_s vermindert sich mit steigender Trefferrate. Für diese wahlfreie Verarbeitung nach sortierten Listen ergibt sich

$$\bar{t}_{TR} = N_{qual} \cdot \bar{t}_s(N_{qual}) \tag{4.3}$$

als gesamte mittlere Zugriffszeit.

4.2 Satzweiser und paralleler Zugriff

Das Ablaufdiagramm in Bild 2 zeigt die satzweise Bereitstellung aller durch eine Anfrage qualifizierten Sätze. Kennzeichnend für diese Verarbeitungsart ist die Notwendigkeit, den Zugriff zum Folgesatz erst dann beginnen zu können, wenn die Übertragung des gesuchten Satzes abgeschlossen ist. Man bezeichnet diese Eigenschaft eines Datenverwaltungssystems auch als "one record at a time"-Logik. Bei der satzweisen Verarbeitung können keine Strategien zur Verringerung der gesamten Zugriffszeit vom Datenverwaltungssystem angewendet werden.

Bild 2:
Schematische Darstellung
der satzweisen Verarbeitung

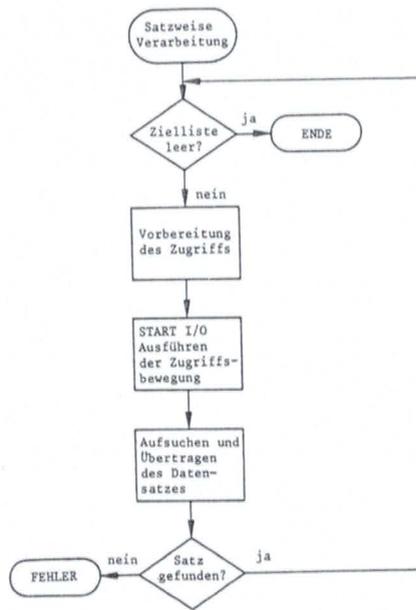
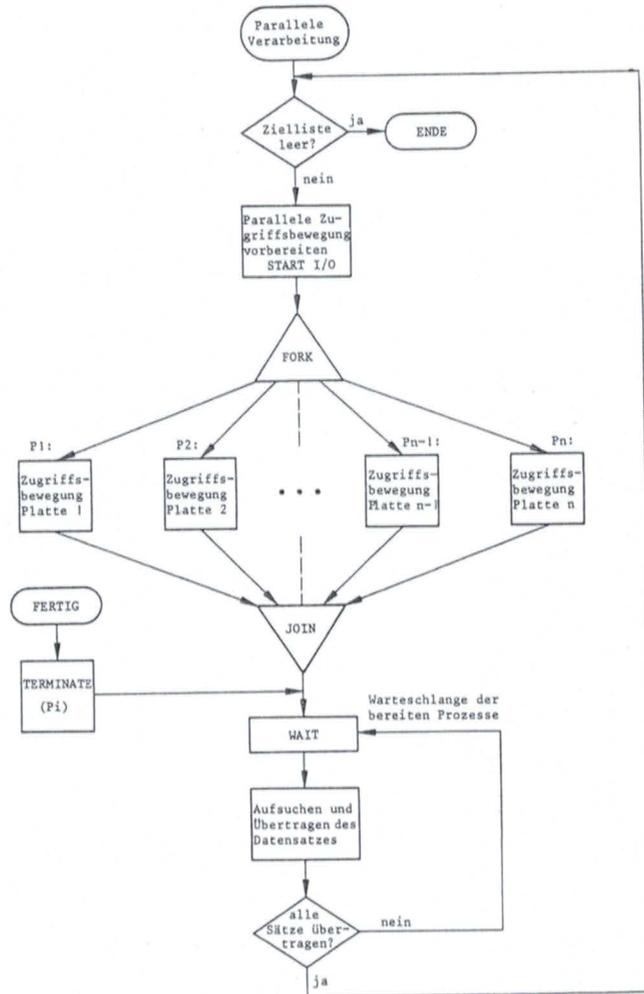


Bild 3:
Schematische Darstellung
der Verarbeitung bei
parallelem Zugriff



Die Art der Vorbereitung des Zugriffs, die Folge der Zugriffsbewegung und das Aufsuchen und Übertragen des gesuchten Datensatzes werden in Bild 2 nicht näher spezifiziert, da sie abhängig von der Datenorganisationsform sind.

Bei großen Dateien, die sich über mehrere Magnetplatten erstrecken, läßt sich die Komponente der Zugriffsbewegungszeit reduzieren. Sind einem Datenverwaltungssystem die Suchanforderungen für mehrere Sätze bekannt, so können Zugriffsbewegungen zu verschiedenen Plattenstapeln parallel initiiert und abgewickelt werden. Voraussetzung dafür ist, daß durch eine geeignete Datenbanksprache, beispielsweise durch eine Sprache nach dem Prädikatenkalkül erster Ordnung, die zu suchenden Sätze aufgrund ihres Inhalts qualifiziert werden. Entsprechend wird dies als 'multi-record at a time'-Logik bezeichnet.

Die einzelnen Zeitkomponenten des Zugriffs - Zugriffsbewegungs-, Umdrehungswarte- und Datenübertragungszeit - können durch Überlappung reduziert werden, wenn die entsprechenden Betriebsmittel vorhanden sind. Es wird davon ausgegangen, daß wegen der Verfügbarkeit eines Kanals die Datenübertragung strikt sequentiell erfolgt. Bei den Magnetplattengeräten IBM 2311 und IBM 2314 können nur die Bewegungen der Zugriffskämme überlappt werden, da während der gesamten Umdrehungswartezeit der Kanal von einem Magnetplattengerät belegt ist.

Bei der IBM 3330 können jeweils Zugriffsbewegung und Teile der Umdrehungswartezeit überlappt werden, da wegen der Positionsbestimmungseinrichtung (rotational position sensing) der Kanal erst kurz vor der Datenübertragung beansprucht wird. Sollten bei einer Hardwarekonfiguration mehrere Kanäle vorhanden sein, so lassen sich die zu erwartenden mittleren Zugriffszeiten noch weiter reduzieren.

In unserem Ansatz berücksichtigen wir wegen ihres relativ hohen Zeitanteils nur die Überlappungen der Zugriffsbewegungen untereinander. Deshalb können wir unsere Ergebnisse als Abschätzungen der oberen Grenze für die Zugriffszeit heranziehen. Die untere Grenze für die Zugriffszeit wird dann erreicht, wenn bei hinreichend großen Datenbeständen die einzelnen Zugriffsbewegungen völlig durch Aufsuch- und Übertragungszeiten auf anderen Magnetplatten überlappt werden. Die Anzahl der Puffer, die das Anwendungsprogramm vorzusehen hat, bleibt von dieser Verarbeitungsform unberührt, da das Datenverwaltungssystem durch dynamische Speicherplatzbelegung dafür sorgen muß, daß die Systempuffer alle qualifizierten Sätze aufnehmen können. Das logische Ablaufdiagramm in Bild 3 soll nur das prinzipielle Schema der parallelen Verarbeitung zeigen. Wir benutzen in diesem Diagramm die Begriffe FORK, JOIN und TERMINATE, die in (An65) für parallele Prozesse eingeführt wurden. Das FORK-Statement initiiert dynamisch n parallele Prozesse P_i ($i=1,2,\dots,n$),

die die Zugriffsbewegungen auf den n Magnetplatten durchführen und die Zugriffskämme auf die angesprochenen Zylinder positionieren. Das JOIN-Statement deaktiviert zwar parallele Prozesse, wenn es in deren Verarbeitungsfolge auftritt, arbeitet aber als AND-Funktion, d. h., das Nachfolger-Statement von JOIN wird erst erreicht, wenn alle parallelen Prozesse, die durch ein FORK-Statement initiiert werden, beendet sind. Um nun zu erreichen, daß mit dem Aufsuchen und Übertragen des ersten Datensatzes begonnen werden kann, sobald irgendein Prozeß P_i mit der Positionierung fertig ist, wird die Möglichkeit vorgesehen, daß jeder Prozeß durch ein TERMINATE-Statement unabhängig von den anderen Prozessen beendet werden kann. Dies geschieht dadurch, daß jeder Prozeß nach erfolgreicher Beendigung einen Ausgang (GOTO-Statement) zur Marke FERTIG hat. Durch das WAIT-Statement soll gekennzeichnet werden, daß eine Warteschlange der E/A-Anforderungen für die Geräte, bei denen die Zugriffsbewegung abgeschlossen ist, gebildet wird. Da nur ein Kanal zur Verfügung steht, ist diese Warteschlange sequentiell abzuarbeiten.

Bei der parallelen Verarbeitung werden zwei oder mehr Prozesse zur selben Zeit initialisiert und arbeiten völlig unabhängig zeitparallel. Es interessiert vor allem die Frage, wann der erste Prozeß beendet ist. In unserem Fall bedeutet dies, daß der Erwartungswert für die Überquerung einer minimalen Anzahl von Zylindern als Funktion von n Magnetplatten zu ermitteln ist. Daraus resultiert eine minimale mittlere Zugriffsbewegungszeit. Nach Beendigung dieser Zugriffsbewegung kann sofort mit dem Aufsuchen und Übertragen des ersten von n Datensätzen begonnen werden. Zur Prüfung der Frage, ob im Verlauf dieser sequentiellen Phase der Verarbeitung eventuell Wartezeiten in Kauf zu nehmen sind, berechnen wir auch den Erwartungswert für die maximale Zugriffsbewegung als Funktion der n Magnetplatten. Solche Wartezeiten treten dann ein, wenn Prozesse ihre Zugriffsbewegung noch nicht beendet haben und der sequentielle Prozeß, der das Aufsuchen und Übertragen der Daten durchführt, eine leere Warteschlange (WAIT-Statement) vorfindet.

Zur Lösung dieser Aufgaben ziehen wir die "parallele Kombination" von Zufallsveränderlichen heran, die in (IBM2) ausführlich dargestellt ist. Die Zufallsveränderlichen X_1, X_2, \dots, X_n und ihre Verteilungsfunktionen $F_1(x), F_2(x), \dots, F_n(x)$ seien statistisch unabhängig voneinander. Dann sind die Verteilungsfunktionen für die Minima und Maxima aller n Zufallsgrößen

$$C_{\min}^{(n)}(x) = 1 - [1 - F_1(x)] \cdot [1 - F_2(x)] \cdot \dots \cdot [1 - F_n(x)] \quad (4.4)$$

$$C_{\max}^{(n)}(x) = F_1(x) \cdot F_2(x) \cdot \dots \cdot F_n(x) \quad (4.5)$$

Sind alle Verteilungsfunktionen gleich, was bei Geräten eines Typs, Gleichverteilung der Datensätze und gleichen Zugriffshäufigkeiten zu allen Daten

der Fall ist, so reduzieren sich die Gleichungen (4.4) und (4.5) zu

$$C_{\min}^{(n)}(x) = 1 - [1 - F(x)]^n \quad (4.6)$$

$$C_{\max}^{(n)}(x) = [F(x)]^n. \quad (4.7)$$

Mit Hilfe der Beziehungen (4.6) und (4.7) sollen nun die Erwartungswerte $\bar{x}_{\min}^{(n)}$ und $\bar{x}_{\max}^{(n)}$ bei wahlfreier Verarbeitung nach ungeordneten Listen ermittelt werden. Die Dichtefunktion $f(x)$ der Verteilungsfunktion $F(x)$ ergibt sich nach (We73) zu

$$f(x) = \begin{cases} \frac{1}{N} & \text{für } x = 0 \\ \frac{2(N-x)}{N^2} & \text{für } 1 \leq x \leq N-1 \end{cases} \quad (4.8)$$

Durch Summieren der Dichtefunktion über den gesamten Bereich von x erhalten wir

$$F(x) = \frac{1}{N} + \frac{2}{N^2} \cdot \sum_{\xi=1}^x (N-\xi) = \frac{1}{N^2} [N + 2 \cdot N \cdot x - x^2 - x].$$

Damit gehen die Gleichungen (4.6) und (4.7) über in

$$C_{\min}^{(n)}(x) = 1 - \frac{(N-x)^n \cdot (N-x-1)^n}{N^{2n}} \quad (4.9)$$

$$C_{\max}^{(n)}(x) = \frac{1}{N^{2n}} \cdot (N+2 \cdot N \cdot x - x^2 - x)^n. \quad (4.10)$$

Für die Berechnung des Mittelwertes $\bar{x}_{\min}^{(n)}$ und $\bar{x}_{\max}^{(n)}$ benötigen wir die Dichtefunktionen der gemeinsamen Verteilungsfunktionen (4.9) und (4.10):

$$P_{\text{omin}}^{(n)} = 1 - \frac{(N-1)^n}{N^n}; \quad P_{\text{omax}}^{(n)} = \frac{1}{N^n}$$

$$P_{\text{xmin}}^{(n)} = C_{\min}^{(n)}(x) - C_{\min}^{(n)}(x-1) = \frac{1}{N^{2n}} \cdot (N-x)^n \cdot [(N-x+1)^n - (N-x-1)^n]$$

$$P_{\text{xmax}}^{(n)} = C_{\max}^{(n)}(x) - C_{\max}^{(n)}(x-1) = \frac{1}{N^{2n}} \cdot [(N+2Nx-x^2-x)^n - (2Nx-N-x^2+x)^n].$$

Somit ergibt sich

$$\begin{aligned} \bar{x}_{\min}^{(n)} &= \sum_{x=1}^{N-1} x \cdot P_{\text{xmin}}^{(n)} = \frac{1}{N^{2n}} \cdot \sum_{x=1}^{N-1} x \cdot [(N-x)^n \cdot \{(N-x+1)^n - (N-x-1)^n\}] \\ &= \frac{1}{N^{2n}} \sum_{k=1}^{N-1} [k \cdot (k+1)]^n \end{aligned} \quad (4.11)$$

$$\bar{x}_{\max}^{(n)} = \sum_{x=1}^{N-1} x \cdot P_{x\max}^{(n)} = (N-1) - \frac{1}{N^2 n} \cdot \sum_{x=0}^{N-2} [N(2x+1) - x(x+1)]^n \quad (4.12)$$

Als Sonderfälle resultieren daraus für $n = 1$

$$\bar{x}_{\min}^{(1)} = \bar{x}_{\max}^{(1)} = \frac{N^2 - 1}{3N} = \frac{N}{3} \cdot \left(1 - \frac{1}{N^2}\right)$$

und für $n = 2$

$$\bar{x}_{\min}^{(2)} = \frac{N}{5} \cdot \left(1 - \frac{5}{3N^2} + \frac{2}{3N^4}\right), \quad \bar{x}_{\max}^{(2)} = \frac{7}{15} N \cdot \left(1 - \frac{5}{7N^2} - \frac{2}{7N^4}\right).$$

Dichte- und Verteilungsfunktionen für die wahlfreie Verarbeitung nach sortierten Listen sind in (Hä75) ausführlich hergeleitet. Wir fassen hier die Ergebnisse zusammen. Wenn auf einer Magnetplatte m Zugriffe erfolgen, erhalten wir als Dichtefunktion bei Gleichverteilung der Zugriffe über N Zylinder

$$g(m, x) = \frac{\binom{N+m-x-2}{m-1}}{\binom{N+m-1}{m}} \quad \text{für } 0 \leq x \leq N-1$$

und als dazugehörige Verteilungsfunktion

$$G(m, x) = 1 - \frac{\binom{N+m-x-2}{m}}{\binom{N+m-1}{m}} \quad \text{für } 0 \leq x \leq N-1.$$

Mit Hilfe von Gleichung (4.5) und (4.7) ergeben sich die gemeinsamen Verteilungsfunktionen der Minima und Maxima von x zu

$$C_{\min}^{(n)}(m, x) = 1 - \left[\prod_{k=1}^m \frac{(N+m-x-k-1)}{(N+m-k)} \right]^n \quad (4.13)$$

$$C_{\max}^{(n)}(m, x) = \left[1 - \prod_{k=1}^m \frac{(N+m-x-k-1)}{(N+m-k)} \right]^n \quad (4.14)$$

Zur Berechnung der Erwartungswerte $\bar{x}_{\min}^{(n)}(m)$ und $\bar{x}_{\max}^{(n)}(m)$ müssen wir von der gemeinsamen Dichtefunktion der n parallelen Zufallsveränderlichen ausgehen, die wir durch Differentiation der Gleichungen (4.13) und (4.14) nach x erhalten. Die Auswertung der Ableitungen, die durch Differenzenquotienten angenähert werden, und die Mittelwertbildung sind nur numerisch durchzuführen:

$$\bar{x}_{\min}^{(n)}(m) = \sum_{x=0}^{N-1} \frac{\Delta C_{\min}^{(n)}(m, x)}{\Delta x} \cdot x = \sum_{x=0}^{N-2} \left[\prod_{k=1}^m \frac{(N+m-x-k-1)}{(N+m-k)} \right]^n \quad (4.15)$$

$$\bar{x}_{\max}^{(n)}(m) = \sum_{x=0}^{N-1} \frac{\Delta C_{\max}^{(n)}(m,x)}{\Delta x} \cdot x = N-1 - \sum_{x=0}^{N-2} \left[1 - \prod_{k=1}^m \frac{(N+m-x-k-1)}{(N+m-k)} \right]^n \quad (4.16)$$

Als Sonderfall erhält man daraus für $n = 1$

$$\bar{x}_{\min}^{(1)}(m) = \bar{x}_{\max}^{(1)}(m) = \frac{N-1}{m+1} .$$

4.3 Fortlaufende Verarbeitung

Falls für eine Anfrage kein Zugriffspfad vorhanden ist oder die Zielliste so viele Einträge aufweist, daß die Grenztrefferrate TR_0 überschritten wird, ist der gesamte Datenbestand fortlaufend zu lesen und zu überprüfen. Diese fortlaufende Verarbeitung kann in logischer Reihenfolge entsprechend dem Zugriffspfad für den Primärschlüssel oder in physischer Reihenfolge der Datensätze erfolgen. Da die Reihenfolge, in der die Datensätze gelesen werden, in normalisierten Relationen keine Rolle spielt, wird man aus Kostengründen eine physisch fortlaufende Verarbeitung wählen. Zur Berechnung der Verarbeitungszeit für den gesamten Datenbestand nehmen wir an, daß die Datensätze vom externen Speichermedium blockweise unter Ausnutzung der maximalen Kanalgeschwindigkeit zu übertragen sind. Durch Anwendung einer Wechselpuffertechnik lassen sich die internen Rechenzeiten für die Auswahl der qualifizierten Datensätze überlappen. Es wird dabei unterstellt, daß Geschwindigkeit und Verfügbarkeit der Zentraleinheit die fortlaufende Verarbeitung nicht hemmt. Gegebenenfalls läßt sich das physisch fortlaufende Suchen des gesamten Datenbestandes an periphere Suchrechner delegieren (Ka75). Die gesamte Verarbeitungszeit hängt somit linear von N_{REC} ab:

$$\bar{t}_{ges} = K_{seq} \cdot N_{REC} \quad (4.17)$$

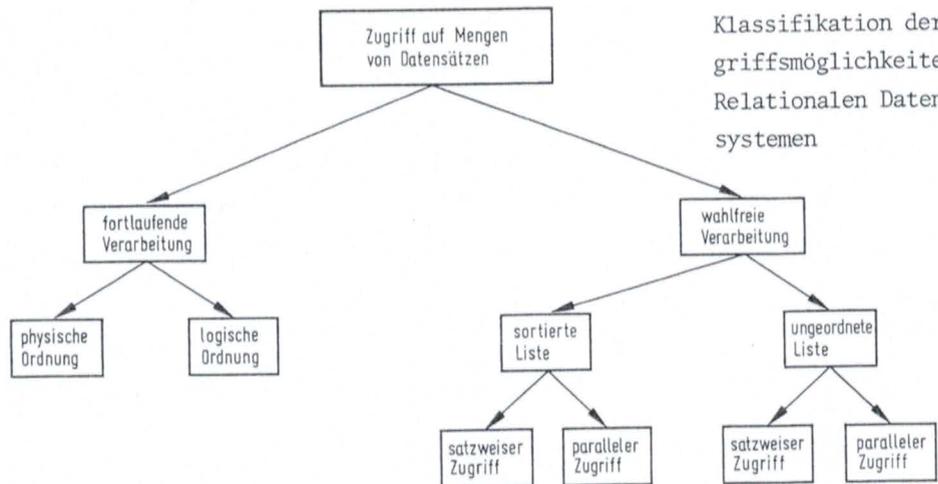
Die Größe von K_{seq} ist im Anhang abgeleitet. Unter Berücksichtigung der Beziehungen (4.2) bzw. (4.3) läßt sich die Grenztrefferrate TR_0 bestimmen, für die folgende Bedingung gelten muß:

$$\bar{t}_{TR_0} = \bar{t}_{ges} \quad (4.18)$$

4.4 Zugriffsmöglichkeiten in Relationalen Datenbanksystemen

Zusammenfassend soll hervorgehoben werden, daß eine deskriptive Datenbanksprache bei der Behandlung einer Anfrage beträchtliche Vorteile bietet und die Freiheitsgrade des Datenverwaltungssystems bei einer geeigneten Implemen-

tierungstechnik für Zugriffspfade erhöht. Das folgende Klassifikationsschema für den Zugriff auf Mengen von Datensätzen verdeutlicht diesen Zusammenhang:



Während im Falle einer deskriptiven Sprache alle dargestellten Möglichkeiten offenstehen, ist mit einer navigierenden Sprache bei wahlfreier Verarbeitung im allgemeinen nur satzweiser Zugriff nach einer ungeordneten Liste möglich, und zwar unabhängig davon, ob die Zugriffspfade eingebettet in die Primärdaten sind oder nicht. Bei hierarchischen Datenstrukturen muß im allgemeinen die fortlaufende Verarbeitung nach der logischen Ordnung erfolgen, da implizite Beziehungen in den Speicherungsstrukturen nur zu erkennen sind, wenn der richtige Zugriffspfad verfolgt wird. Da man bei normalisierten Relationen ohne derartige Hilfsmittel auf der Ebene der Speicherungsstrukturen auskommt, ist die schnellere physisch fortlaufende Verarbeitung anzuwenden.

5. Zugriffszeiten für die gestreute Speicherungsstruktur mit direkter Adressierung

Für die im Abschnitt 4 beschriebenen Zugriffsmodelle werden die erzielbaren Zugriffszeiten beim Zugriff auf Mengen von Datensätzen ohne Berücksichtigung von Warteschlangenzusammenhängen berechnet. Wir beschränken uns dabei exemplarisch auf den besonders einfachen Fall der gestreuten Speicherungsstruktur mit direkter Adressierung. Aus den Primärschlüsseln der Zielliste wird mit Hilfe eines Hash-Algorithmus eine eindeutige Satzadresse gewonnen, auf die ohne sequentielle Suche in der Spur zugegriffen werden kann. Es wird unter

allen möglichen Datenorganisationsformen das günstigste Zugriffszeitverhalten bei wahlfreier Verarbeitung erzielt. Da das Problem der Überlaufsätze nicht auftritt, wird jeder Satz in einem einzigen Zugriffszyklus - Zugriffsbewegung, Aufsuchen und Übertragen - erreicht. Ein Zugriff läßt sich durch das einfache Ablaufdiagramm in Bild 5 beschreiben. Falls in der Zielliste, nach der die Datensätze bereitzustellen sind, schon physische Satzadressen vorliegen, ist das Zugriffszeitverhalten gleich dem eben diskutierten Fall, da die Rechenzeit für die Schlüsseltransformation zu vernachlässigen ist.

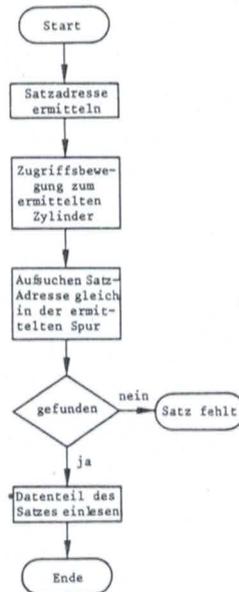
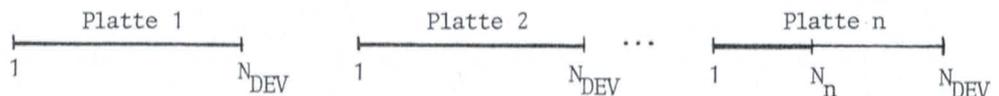


Bild 5: Logischer Ablauf bei Verarbeitung einer gestreuten Speicherstruktur mit direkter Adressierung (Bn73, S. 39)

Folgende Zuordnung einer Datei auf n Magnetplatten wählen wir für unsere Zeitbetrachtungen:



Die verwendeten Bezeichnungen sind im Anhang eingeführt und definiert. Die N_{ZYL} belegten Zylinder der Datei werden lückenlos auf n Magnetplatten ange-

ordnet. Die Anzahl der auf der n-ten Magnetplatte belegten Zylinder bestimmt sich durch

$$N_n = N_{ZYL} - (n-1) \cdot N_{DEV},$$

wobei sich N_{ZYL} aus der Anzahl der Datensätze und den Charakteristika der Speicherzuordnungsstruktur herleiten läßt:

$$N_{ZYL} = \frac{1}{\beta} \cdot \frac{N_{REC}}{b \cdot N_B \cdot S_{ZYL}} \quad (5.1)$$

5.1 Zugriffe nach ungeordneten Listen

Unter Berücksichtigung von Gleichung (4.11), der im Anhang abgeleiteten Näherungsgleichung (A.3) für die Zugriffsbewegung auf einer Magnetplatte läßt sich die mittlere Zugriffszeit für einen Datensatz, der nach einer ungeordneten Liste im satzweisen Zugriff bereitgestellt wird, nach Gleichung (A.2)

$$\bar{t} = \frac{1}{2} t_{rev} + \frac{1}{u} \cdot b \cdot S_{\ell} + (n-1) \cdot \frac{N_{DEV}}{N_{ZYL}} \cdot (t_o + s_o \cdot \bar{x}_{min}^{(1)}(N_{DEV})) + \frac{N_n}{N_{ZYL}} \cdot (t_o + s_o \cdot \bar{x}_{min}^{(1)}(N_n)) \quad (5.2)$$

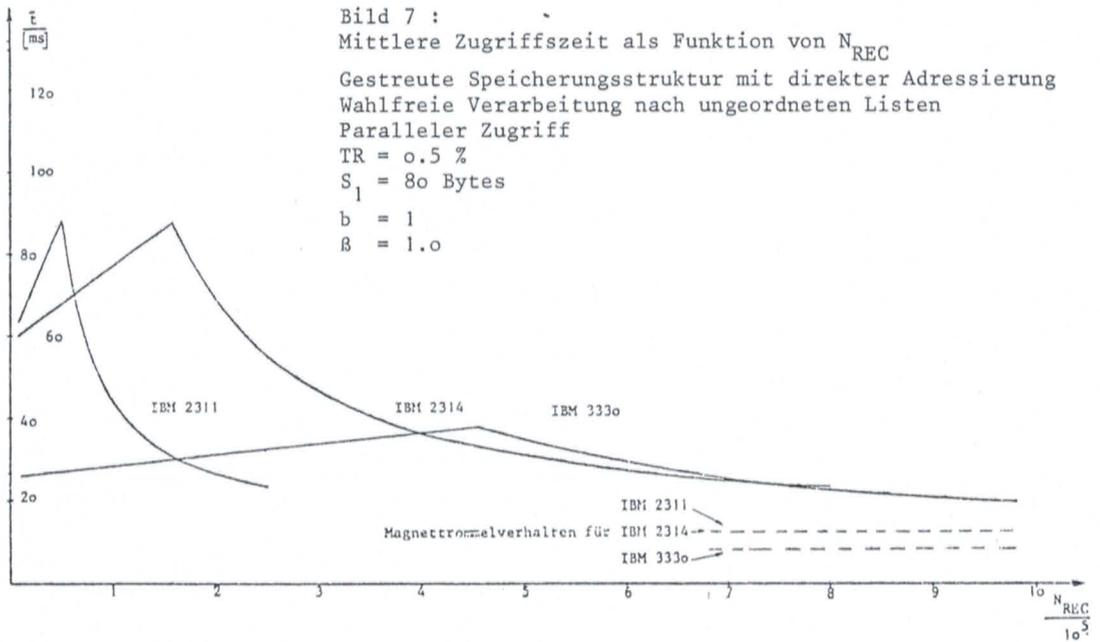
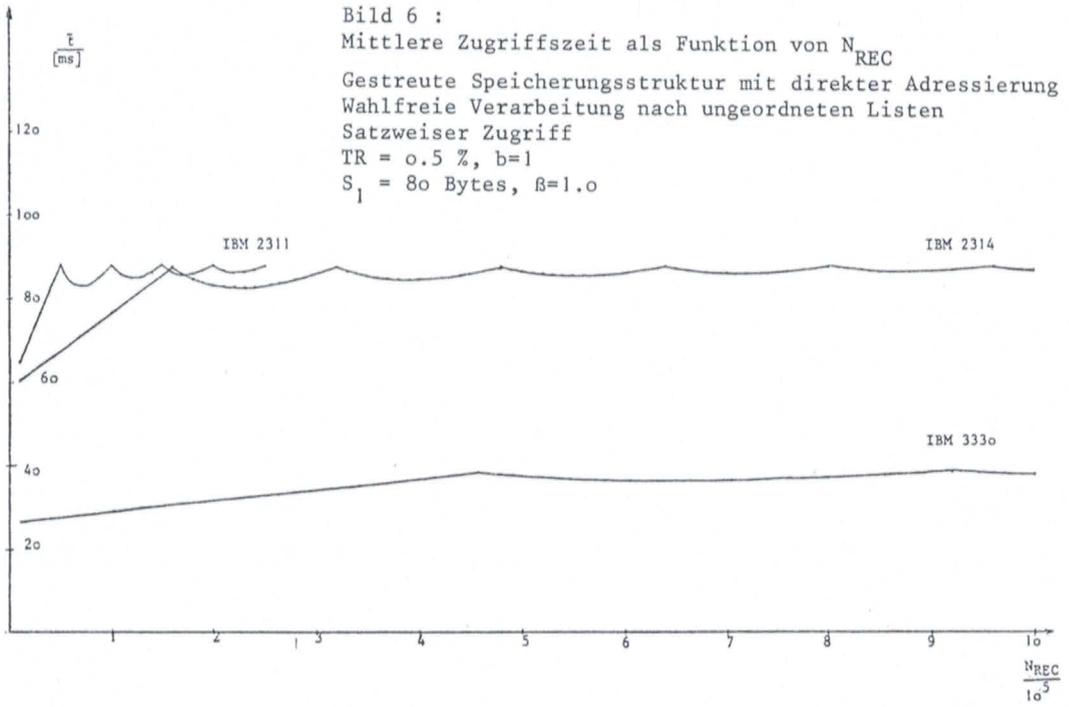
allgemein beschreiben.

Falls paralleler Zugriff angewendet werden kann ($N_{ZYL} > N_{DEV}$), müssen wegen der Gleichverteilung der Zugriffe als äußerer Bedingung und der gewählten Zugriffsstrategie nur die Zugriffsbewegungen zu einer voll belegten Magnetplatte in den Ansatz einbezogen werden. Wir erhalten folglich

$$\bar{t} = \frac{1}{2} \cdot t_{rev} + \frac{1}{u} \cdot b \cdot S_{\ell} + \frac{N_{DEV}}{N_{ZYL}} \cdot (t_A + s \cdot \bar{x}_{min}^{(n)}(N_{DEV})) \quad (5.3)$$

als mittlere Zugriffszeit für einen Datensatz, wenn nach einer Zielliste $N_{qual} > n$ Sätze aufzusuchen sind.

In den Bildern 6 und 7 sind die Beziehungen (5.2) und (5.3) für gegebene Hardware-Konstanten und Parameter der verwendeten Speicherzuordnungsstruktur als Funktion von N_{REC} ausgewertet. Der Einfluß eines Belegungsfaktors $\beta < 1$ ist dadurch abzuschätzen, daß die Kennlinien konstant gehalten und die Abszissenwerte mit β multipliziert werden. Dies führt auf eine Spreizung der Maßstabskala. Die Parameter b und N_B sind über Gleichung (A.1) miteinander verknüpft. Die Wahl eines Blockungsfaktors $b > 1$ führt auf eine bessere Ausnutzung des Speicherplatzes, da die Anzahl der notwendigen Hardwareklüfte reduziert wird. Dies bedeutet eine Verdichtung der Skala auf der Abszissenachse. Dadurch, daß größere physische Sätze zu übertragen sind, wächst gleichzeitig die Komponente der Übertragungszeit linear mit dem Blockungsfaktor.



Wie in Bild 7 veranschaulicht ist, kann durch parallelen Zugriff nach ungeordneten Listen in großen Datenbeständen annähernd "Magnettrommelverhalten" erzielt werden, d. h., durch die Zugriffsstrategie wird die Komponente der Zugriffsbewegungszeit im Mittel weitgehend eliminiert.

5.2 Zugriffe nach sortierter Liste

Bei der Verarbeitung nach einer sortierten Liste wird der Zugriffskamm nur in einer Richtung bewegt. Die mittlere Zugriffsbewegungszeit ist abhängig von der Anzahl der gesamten Zugriffe, also eine Funktion der Trefferrate. Der größte Anteil der Zugriffsbewegungszeit an der gesamten Zugriffszeit wird dann erreicht, wenn in jedem Zylinder mindestens ein Satz aufzusuchen ist ($N_{\text{qual}} \geq N_{\text{ZYL}}$). In diesem Fall kann $N_{\text{ZYL}}/N_{\text{qual}}$ als Wahrscheinlichkeit aufgefaßt werden, daß bei einem Zugriff neu zum nächsten Zylinder zu positionieren ist. Ist $N_{\text{qual}} < N_{\text{ZYL}}$, so wird bei jedem Zugriff eine Zugriffsbewegung erforderlich.

Für die Berechnung der mittleren Zugriffszeit beim Zugriff nach sortierten Listen ist eine Fallunterscheidung durchzuführen. Für die Anzahl der Treffer auf einer Magnetplatte setzen wir näherungsweise $m = \lceil N_{\text{qual}}/n \rceil$ oder unter Berücksichtigung von Gleichung (4.1) $m = \left\lceil \frac{\text{TR} \cdot N_{\text{REC}}}{100 \cdot n} \right\rceil$

Für satzweisen Zugriff erhalten wir

$$\bar{t}(\text{TR}) = \frac{1}{2} \cdot t_{\text{rev}} + \frac{1}{u} \cdot b \cdot S_{\ell} + \frac{N_{\text{DEV}}}{m} \cdot t_{z \text{ min}} \quad \text{für } m \geq N_{\text{DEV}} \quad (5.4)$$

$$\bar{t}(\text{TR}) = \frac{1}{2} \cdot \bar{t}_{\text{rev}} + \frac{1}{u} \cdot b \cdot S_{\ell} + t_o + s_o \cdot \bar{x}_{\text{min}}^{(1)}(m) \quad \text{für } m < N_{\text{DEV}}$$

Sobald $N_{\text{ZYL}} > N_{\text{DEV}}$ ist, läßt sich wiederum eine parallele Zugriffsstrategie anwenden. Wegen der Überlappung der Zugriffsbewegungen ist in jedem parallelen Zugriffszyklus entsprechend Bild 3 für alle gefundenen Datensätze nur eine gewichtete Zugriffsbewegung $\bar{x}_{\text{min}}^{(n)}(m)$ zu berücksichtigen. Beim zylinderweisen Fortschreiten der Zugriffsarme ist $\bar{x}_{\text{min}}^{(n)}(m)$ wiederum die Wahrscheinlichkeit, daß neu zu positionieren ist. Wir fassen beide Fälle zusammen:

$$\bar{t}(\text{TR}) = \frac{1}{2} t_{\text{rev}} + \frac{1}{u} \cdot b \cdot S_{\ell} + \frac{m \cdot 100}{\text{TR} \cdot N_{\text{REC}}} \cdot \bar{x}_{\text{min}}^{(n)}(m) \cdot t_{z \text{ min}} \quad \text{für } 0 \leq \bar{x}_{\text{min}}^{(n)}(m) \leq 1$$

$$\bar{t}(\text{TR}) = \frac{1}{2} t_{\text{rev}} + \frac{1}{u} \cdot b \cdot S_{\ell} + \frac{N_{\text{DEV}}}{N_{\text{ZYL}}} \cdot (t_o + s_o \cdot \bar{x}_{\text{min}}^{(n)}(m)) \quad \text{für } \bar{x}_{\text{min}}^{(n)}(m) > 1 \quad (5.5)$$

In Bild 8 und 9 sind die Beziehungen (5.4) und (5.5) für vorgegebene Parameter als Funktion der Trefferrate TR ausgewertet. Beim satzweisen Zugriff ist die mittlere Zugriffszeit unabhängig von der Dateigröße, während N_{REC} beim parallelen Zugriff den Grad der Parallelität und damit das Maß der Über-

Bild 8 :
 Mittlere Zugriffszeit als Funktion von TR
 Gestreute Speicherungsstruktur mit direkter Adressierung
 Wahlfreie Verarbeitung nach sortierten Listen
 Satzweiser Zugriff
 $s_1 = 80$ Bytes, $b = 1$
 $\beta = 1.0$

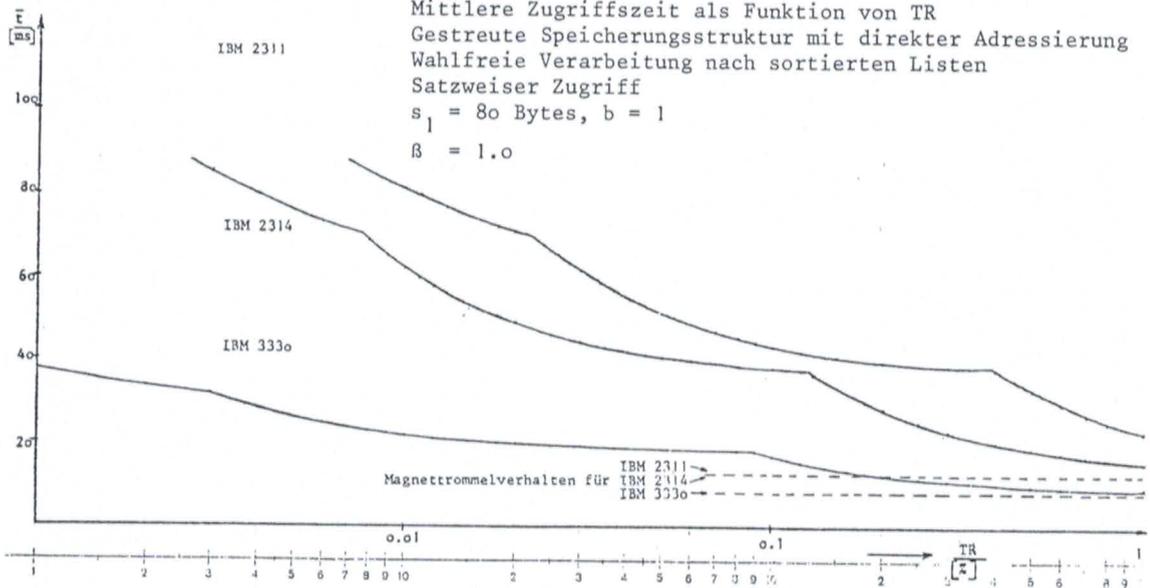
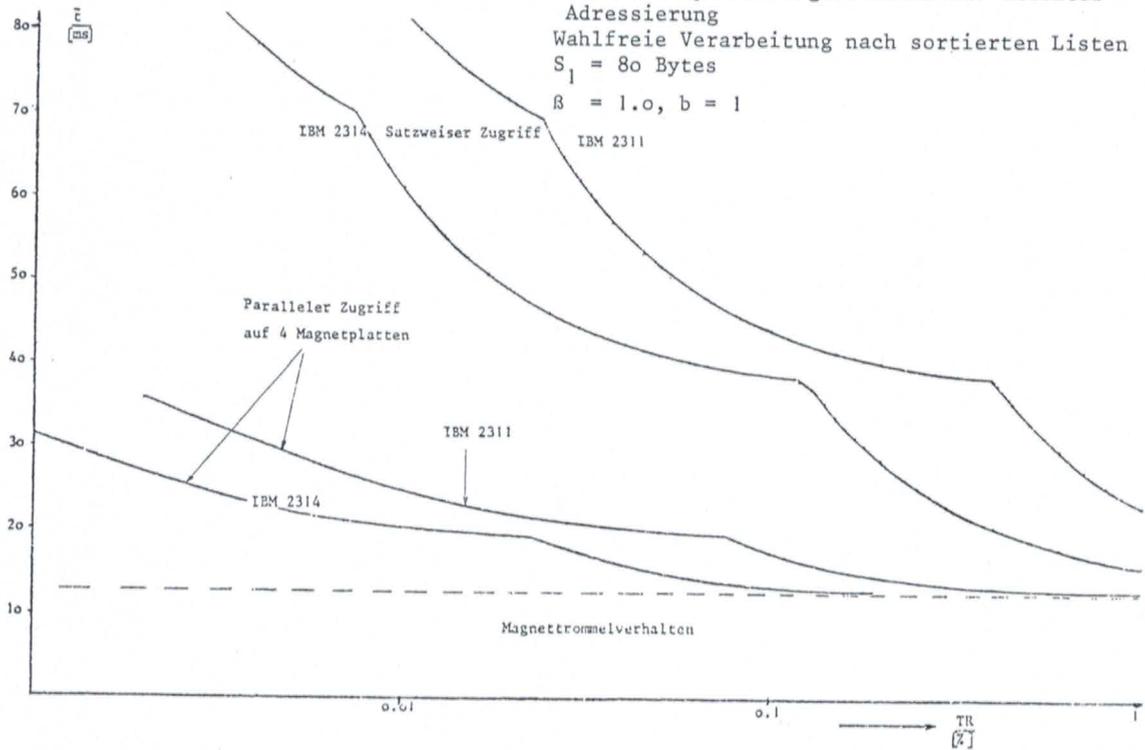


Bild 9 :
 Mittlere Zugriffszeit als Funktion von TR
 Gestreute Speicherungsstruktur mit direkter Adressierung
 Wahlfreie Verarbeitung nach sortierten Listen
 $s_1 = 80$ Bytes
 $\beta = 1.0$, $b = 1$



lappung bestimmt. In beiden Fällen ist schon bei relativ kleinen Treffer-
raten "Magnettrommelverhalten" zu erzielen. Durch Vergleich der durch geeig-
nete Strategien verbesserten Zugriffszeiten mit den in Bild 6 aufgezeichneten
Zugriffszeiten, die den satzweisen Zugriff nach ungeordneten Listen oder den
navigierenden Suchprozeß, bei dem das Datenbanksystem durch die ständige In-
teraktion mit dem Anwendungsprogramm die einzelnen Zugriffe nicht koordinieren
kann, charakterisieren, läßt sich feststellen, daß diese Steigerung der Leistungs-
fähigkeit durch das hohe Auswahlvermögen deskriptiver Sprachen und durch die
Separierung der Zugriffsinformation von den Datensätzen zustande gekommen ist.
Eine Einbettung der Zugriffspfade in die Datensätze durch Adreßketten verhin-
dert diese Optimierungsmöglichkeit ebenso wie der Gebrauch einer prozeduralen
Datenmanipulationssprache.

Anhang

A. Herleitung der Verarbeitungs- und Zugriffszeiten

Zur Herleitung des Speicherplatzbedarfs und der verschiedenen Verarbeitungs-
und Zugriffszeiten sind eine Reihe von Parametern der Speicherungs- und
Speicherzuordnungsstruktur und Gerätecharakteristika einzuführen.

Folgende Größen werden benötigt:

\bar{t}	= mittlere Zugriffszeit zu einem Datensatz
\bar{t}_z	= mittlere Zugriffsbewegungszeit
\bar{t}_r	= mittlere Undrehungswartezeit
\bar{t}_{tr}	= mittlere Datenübertragungszeit
t_o	= Anfangswert der Näherungsgeraden der Zugriffsbewegung
s_o	= Steigung der Näherungsgeraden der Zugriffsbewegung
x	= Anzahl der zu überquerenden Zylinderabstände
N_{ZYL}	= Anzahl der belegten Zylinder der Datei
N_B	= Anzahl der gespeicherten Sätze pro Spur (physische Sätze)
b	= Blockungsfaktor
S_λ	= Länge eines logischen Satzes
$b \cdot S_\lambda$	= durch den Kanal zu übertragende Daten
β	= Belegungsfaktor = Verhältnis zwischen genutztem und insgesamt verfügbarem Speicherplatz
KL	= Schlüssellänge eines separat gespeicherten Schlüssels

Die technischen Charakteristika der verwendeten Magnetplatteneinheiten sind in der folgenden Tabelle zusammengefaßt:

Merkmal	Magnetplattentyp		
	IBM 2311	IBM 2314	IBM 3330
t_{zmin} = Zugriffsbewegungszeit (Min.)	25 ms	25 ms	10 ms
t_{zav} = " (Mittelwert)	75 ms	75 ms	30 ms
t_{zmax} = " (Maximum)	135 ms	135 ms	55 ms
t_{rev} = Umdrehungszeit	25 ms	25 ms	16.7 ms
C = verfügbare Spurkapazität	3625 Bytes	7294 Bytes	13030 Bytes
S_{ZYL} = Anzahl der Spuren pro Zylinder	10	20	19
N_{DEV} = Anzahl der Zylinder	200 (+3)	200 (+3)	404 (+7)
u = Übertragungsgeschwindigkeit	156 $\frac{\text{Bytes}}{\text{ms}}$	312 $\frac{\text{Bytes}}{\text{ms}}$	806 $\frac{\text{Bytes}}{\text{ms}}$
K_D = Kluft zwischen Datensätzen	61 Bytes	101 Bytes	135 Bytes
K_S = Kluft zwischen Schlüssel und Datensatz	20 Bytes	45 Bytes	56 Bytes
K_V = Faktor für die variable Kluftlänge	0.05	0.04	0.04
t_A = Anfangswert der Näherungsgeraden der Zugriffsbewegung für $x \gg 1$	45 ms	45 ms	17.5ms
S = Steigung der Näherungsgeraden der Zugriffsbewegung für $x \gg 1$	0.45	0.45	0.094
S_n = Steigung der Näherungsgeraden der Zugriffsbewegung für $x \leq N_{DEV}/10$	1.6	1.6	0.325

a) Als Anzahl der physischen Sätze pro Spur erhält man nach (Hä75)

$$N_B = 1 + (C - K_S - KL - b \cdot S_\ell) / (b \cdot S_\ell + K_D + K_S + KL + K_V \cdot (b \cdot S_\ell + KL)) \quad \text{mit } S_\ell \ll C \quad (\text{A.1})$$

Falls kein separat gespeicherter Schlüssel vorhanden ist, wird $KL=0$ und gleichzeitig $K_S=0$.

b) Bei der Bestimmung der mittleren Zugriffszeit \bar{t} zu einem externen Speichermedium mit halbdirektem Zugriff sind drei ihrer Komponenten - Zugriffsbewegungszeit \bar{t}_z , Umdrehungswartezeit \bar{t}_r und Datenübertragungszeit \bar{t}_{tr} - zu berücksichtigen:

$$\bar{t} = \bar{t}_z + \bar{t}_r + \bar{t}_{tr} \quad (\text{A.2})$$

Umdrehungswartezeit und Datenübertragungszeit für einen physischen Datensatz ergeben sich zu:

$$\bar{t}_r = \frac{1}{2} \cdot t_{\text{rev}}$$

$$\bar{t}_{\text{tr}} = \frac{1}{u} \cdot b \cdot S_{\text{Z}}$$

Die Zugriffsbewegungszeit kann man für Entwurfsüberlegungen durch Näherungsgeraden mit hinreichender Genauigkeit beschreiben (Hä75). Als allgemeine Näherungsgerade wird

$$t_z = t_0 + s_0 \cdot x \quad (\text{A.3})$$

eingeführt. Durch den speziellen Verlauf der Zugriffsscharakteristik ist jeweils für alle Magnetplattentypen eine Unterscheidung zwischen Nahbereich und Fernbereich von x zu machen. Gleichung (A.3) geht über in

$$t_z = t_{z \text{ min}} + s_n \cdot x \quad \text{für } 1 < x \leq N_{\text{DEV}}/10$$

und

$$t_z = t_A + s \cdot x \quad \text{für } x > N_{\text{DEV}}/10.$$

- c) Bei der physisch fortlaufenden Verarbeitung ist der Zugriffsarm einmal über alle Spuren und Zylinder der Datei zu führen. Wenn mit voller Kanalgeschwindigkeit übertragen werden kann, läßt sich die benötigte gesamte Lesezeit durch

$$\bar{t}_{\text{ges}} = N_{\text{ZYL}} \cdot S_{\text{ZYL}} \cdot t_{\text{rev}} + N_{\text{ZYL}} \cdot t_{z \text{ min}}$$

in guter Näherung ausdrücken. Wegen $t_{z \text{ min}} \leq t_{\text{rev}}$ und unter Berücksichtigung von Gleichung (A.2) folgt somit

$$t_{\text{ges}} = \frac{S_{\text{ZYL}} + 1}{\beta \cdot b \cdot N_B \cdot S_{\text{ZYL}}} \cdot t_{\text{rev}} \cdot N_{\text{REC}}$$

Wir fassen die Parameter der Speicherzuordnungsstruktur und die Gerätecharakteristika zu einer sequentiellen Verarbeitungsconstanten

$$K_{\text{seq}} = \frac{S_{\text{ZYL}} + 1}{\beta \cdot b \cdot N_B \cdot S_{\text{ZYL}}} \cdot t_{\text{rev}} \quad (\text{A.4})$$

zusammen.

LITERATURVERZEICHNIS

- (An 65) Anderson, J.P.: Programming Structures for Parallel Processing, in: Comm. of the ACM, Vol. 8, No.12, Dec. 1965, S. 786 - 788.
- (Ba 73) Bachmann, C.W.: The Programmer as Navigator, in: Comm. of the ACM (1973 Turing Award Lecture), Vol. 16, No. 11, Nov. 73, S. 653 - 658.
- (Bn 73) Bastian, R: Zeitberechnungen peripherer Speicher, S. Toeche-Mittler-Verlag, Darmstadt 1973.
- (Ch 74) Chamberlin, D.D., Boyce, R.F.: SEQUEL : A Structured English/Query Language, in : Proc. of 1974 ACM SIGFIDET Workshop on Data Description, Access and Control, Ann Arbor, Michigan, April 1974, S. 249 - 264.
- (CODA) CODASYL DATA BASE TASK GROUP (DBTG) REPORT, April 1971, erhältlich bei IFIP Administrative Data Processing Group, 40 Paulus Potterstraat, Amsterdam.
- (Co 70) Codd, E.F.: A Relational Model of Data for Large Shared Data Banks, in: Comm. of the ACM, Vol.13, No6, June 1970, S. 377 - 387.
- (Co 71) Codd, E.F.: A Data Base Sublanguage Founded on the Relational Calculus, in : 1971 ACM SIGFIDET Workshop on Data Description, Access and Control, San Diego, Calif. , Nov. 1971, S. 35 - 68.
- (Hä 74) Härder, T.: Implementierung von Zugriffspfaden durch Bitlisten, Forschungsbereich DV 74-2, FG Datenverwaltungssysteme, TH Darmstadt, 1974.
- (Hä 75a) Härder, T.: Auswahl von Zugriffspfadstrukturen in Datenbanksystemen, in: Informatik Anwendungen, Beiträge zum Kolloquium des ZAI der TH Darmstadt, 3.- 4 Feb. 1975, S. 35 - 61.
- (Hä 75) Härder, T.: Das Zugriffszeitverhalten von Relationalen Datenbanksystemen, Dissertation im Fachbereich Informatik der Technischen Hochschule Darmstadt, Juni 1975.
- (IBM 1) IBM Information Management System, "IMS/360 Specifications, IBM Manual GH20-4106.
- (IBM 2) IBM-Broschüre: IBM Data Processing Techniques; Analysis of some Queuing Models in Real-Time Systems, Best.-Nr.: F20-0007-1.
- (Ka 75) Karlowsky, I, Leilich, H.-D., Stiege, G.: Ein Suchrechnerkonzept für Datenbankanwendungen, in : Elektronische Rechenanlagen 17 (1975), H. 3, S. 108-118.
- (Wa 73) Wang, C.P.: Parametrization of Information System Application, IBM Research Report, RJ 1199, (#19228), San Jose, April 11, 1973.
- (We 73) Wedekind, H.: Systemanalyse, Carl Hanser Verlag, München 1973.

Sonderdruck aus
Applied Computer Science 2:
Hoßfeld, Praxis der Realisierung
von Informationssystemen
Carl Hanser Verlag München Wien
1976