Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

The Five-Minute Rule

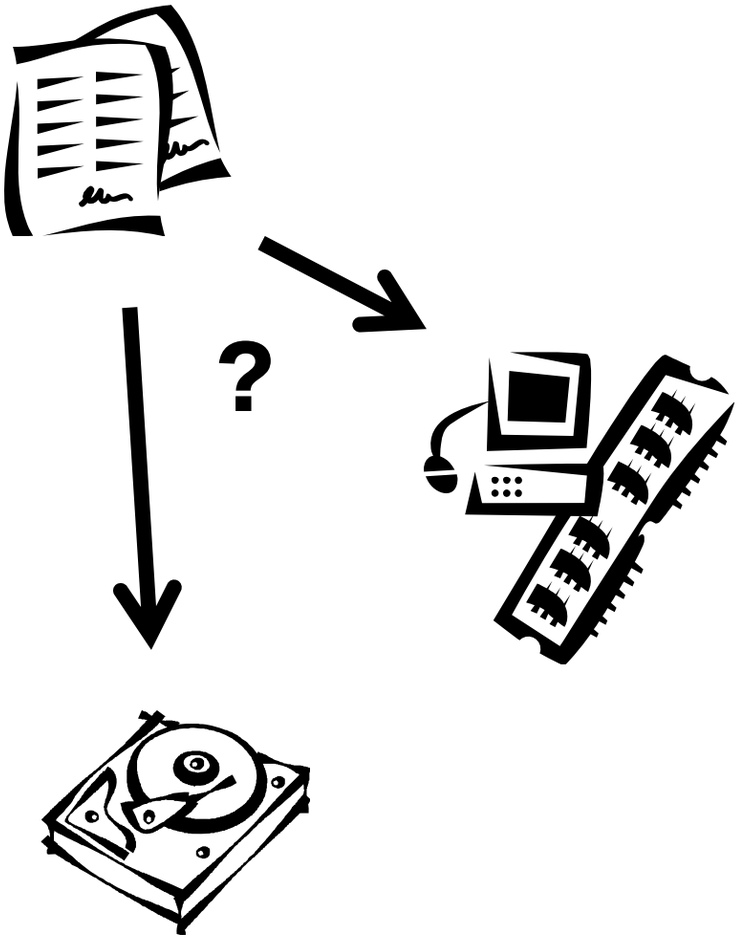
for trading memory for disc accesses

Advanced Topics in Databases

Marcel Taeumel

Agenda

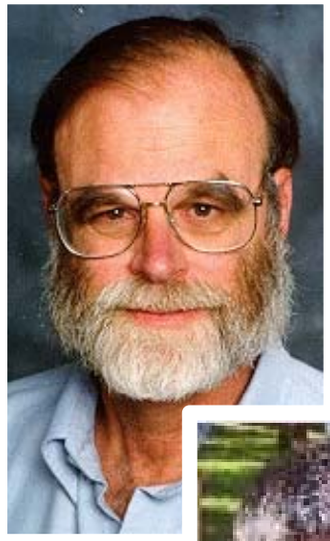
1. **Autoren**
2. **"Five-Minute Rule"**
   - Herleitung der Regel
   - Fallstudie
   - 10 Jahre später – 1997
   - 20 Jahre später – 2007
3. **Weitere Faustregeln**
   - Größe von Indexseiten
   - "One-Minute Sequential Rule"
   - Weitere Themenbereiche



Autoren

Jim Gray

- 1966 B. Sc. in Mathematik und Statistik
- 1969 Promotion in Informatik
- Entwickler bei IBM, Tandem, DEC



Franco Putzolu

- Entwickler bei IBM (u.a. DB2)
- Chef-Designer für NonStop SQL (Tandem)
- Senior Database Architect (Oracle)

Goetz Graefe

- regelbasierte, Top-Down Anfrageoptimierung
- 12 Jahre bei Microsoft tätig (SQL Server)
- seit 2007 bei HP Labs

Erste Gedanken...

1987

Wichtige Größen (1987)

- Kosten pro Festplatte
  - inkl. Betriebskosten (CPU, Controller)

$$\$_{disc} = 30.000 \frac{\$}{disc}$$

- Zugriffsgeschwindigkeit
  - Anzahl Zugriffe pro Sekunde (wahllos)

$$t_{acc} \approx 66\,ms$$
$$\Rightarrow 15\,{}^{access}\!/_{second}$$

- Kosten pro Megabyte Hauptspeicher

$$\$_{RAM} = 5.000 \frac{\$}{MB}$$

- Größe des Datensatzes
- Größe der Datenblöcke
  - maßgebend, falls Datensatz zu groß
  - Fragmentierung

- **Zeit zwischen zwei Zugriffen auf gleiches Datum**

[gray87]

Vorüberlegung

- Kosten für I/O (Festplatte)

$$\$_{I/O} = \frac{PricePerAccessPerSecond}{ReferenceInterval}$$

- Hauptspeicherkosten für Datensatz/-block

$$\$_{RAM,Record} = \$_{RAM} \cdot RecordSize$$

- Kosteneinsparung durch Halten der Daten im Hauptspeicher

$$Savings = \$_{I/O} - \$_{RAM,Record}$$

[gray87]

Herleitung

- Break-Even-Point

$$0 = \$_{I/O} - \$_{RAM,Record}$$

- Beispielrechnung (1KB-Seiten)

$$ReferenceInterval = \frac{PagesPerMBofRAM}{AccessPerSecondPerDisc} \cdot \frac{PricePerDiskDrive}{PricePerMBofRAM}$$

$$ReferenceInterval = \frac{1000}{15} \cdot \frac{30000}{5000} \cdot {}^{sec}/_{acc} = 400\,{}^{sec}/_{acc} \approx 5\,{}^{min}/_{acc}$$

[gray87]

"Five-Minute Rule"

Daten, welche innerhalb von **5 Minuten*** **wiederholt referenziert** werden, sollten im **Hauptspeicher** verbleiben.

* abhängig von der Seitengröße (hier: 1KB)

[gray87]

Fallstudie

- Ein Kunde möchte seine Datenbank vollständig im Hauptspeicher halten. Lohnt sich das?

- Fakten
  - **500.000** Datensätze à **1000** Bytes
  - fast alle Zugriffe auf einzelne Datensätze
  - Spitzenlast: **600** Transaktionen pro Sekunde
  - geforderte Antwortzeit: 1 Sekunde

[gray87]

Fallstudie: "All-In-Main-Memory"

- 36 VLX-Prozessoren à 16 MB Hauptspeicher
- 2 Festplatten (gespiegelt) für:
  - komplette Datenbank
  - Indizes
  - Programme
- Festplatten im Betrieb nicht benötigt
- Kosten: **ca. 11,7 Mio $**

[gray87]

Fallstudie: "Disc-Memory-Tradeoff"

- nur 30.000 der 500.000 Datensätze innerhalb von **5 Minuten** wiederholt referenziert
  - Spitzenlast berücksichtigt
- 96% der Zugriffe für diese 6% der Daten
  - statistische Abschätzungen
- 2 gespiegelte Festplatten für 24 Zugriffe pro Sekunde genügen
  - 4% von 600TPS

- 30 VLX-Prozessoren
- 106 MB Hauptspeicher
- **Kosten:** 8,2 Mio $

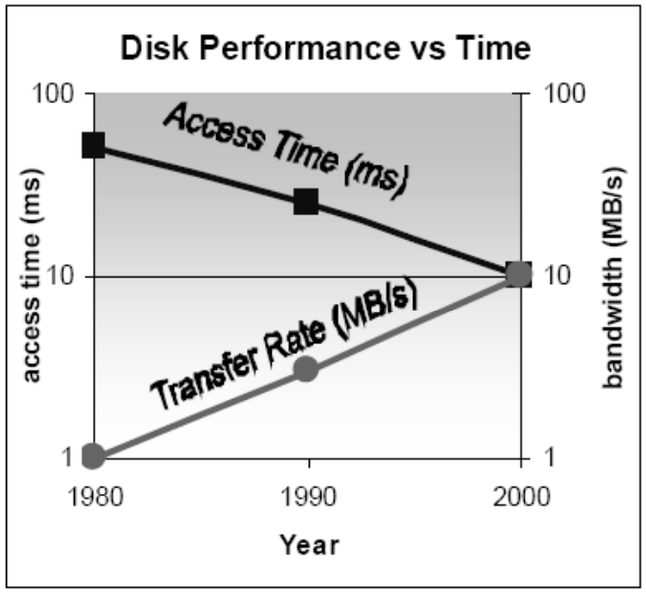
Einsparungen
6 VLX-Prozessoren
470 MB Hauptspeicher
3,5 Mio $ (~30%)

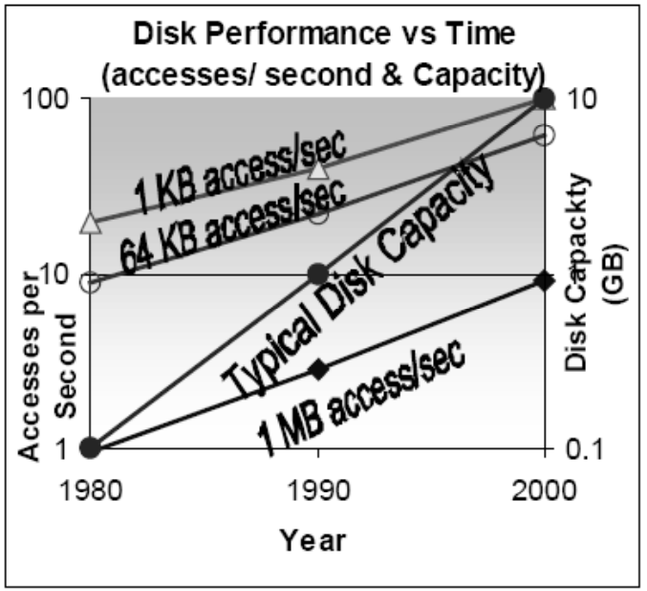
[gray87]

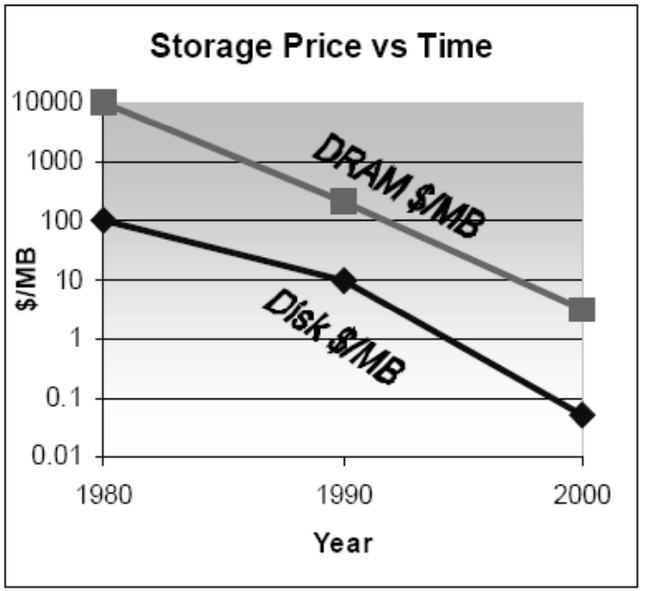
Zehn Jahre später...

1997

Entwicklung







- höhere Bandbreiten und günstigerer Hauptspeicher ermöglichen **größere Seiten**

[gray97]

Technologie und Wirtschaftlichkeit

$$ReferenceInterval = \frac{PagesPerMBofRAM}{AccessPerSecondPerDisc} \cdot \frac{PricePerDiskDrive}{PricePerMBofRAM}$$

$$ReferenceInterval = TechnologyRatio \cdot EconomicRatio$$

10x ↑ 10x ↓

- aktualisierte Beispielrechnung (8KB-Seiten)

$$ReferenceInterval = \frac{128}{64} \cdot \frac{2000}{15} \cdot {}^{sec}\!/_{acc} = 266\,{}^{sec}\!/_{acc} \approx 5\,{}^{min}\!/_{acc}$$

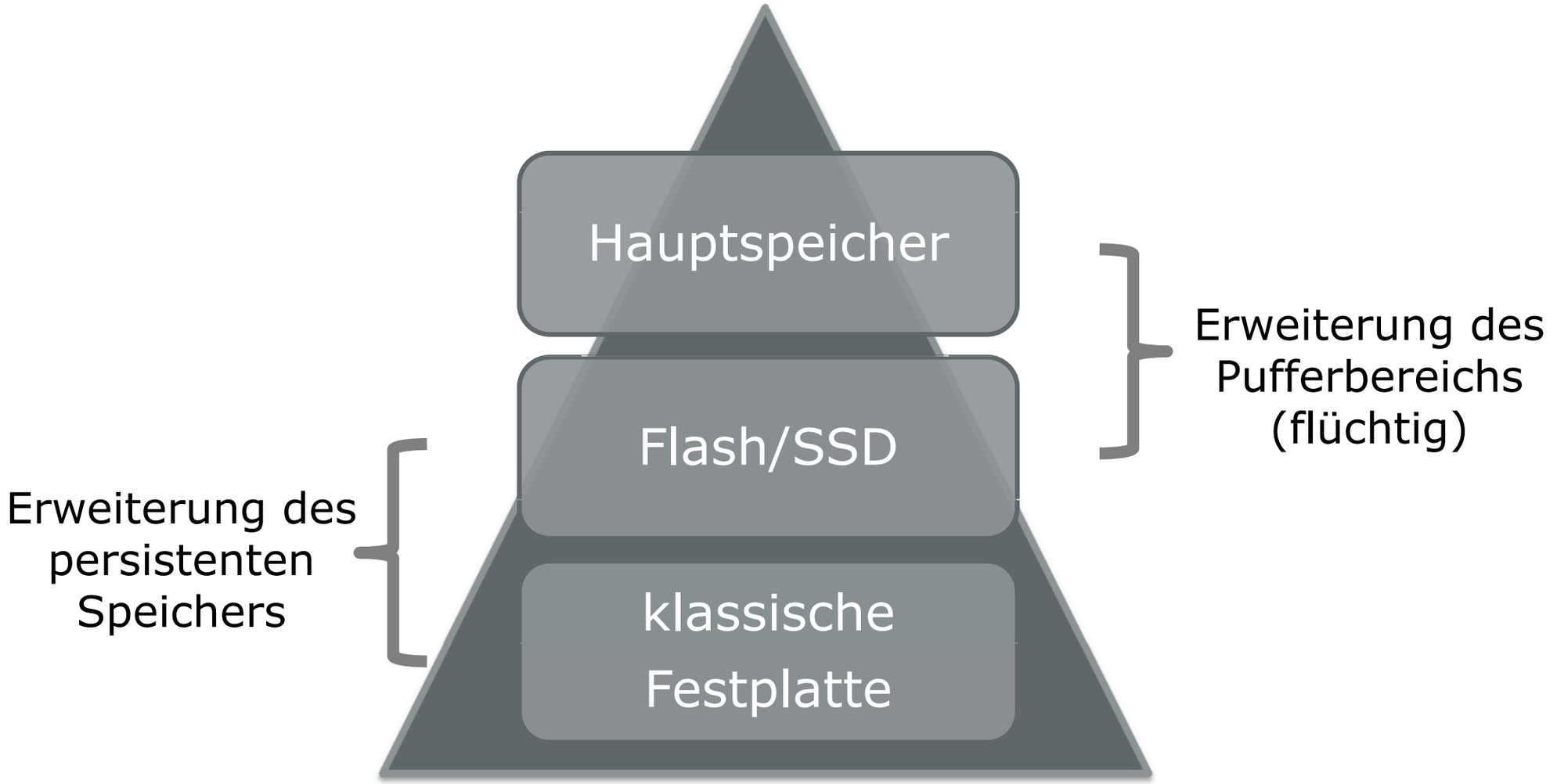
- Vergleich (1987)
  - 1KB-Seite, 400 sec/acc
  - 4KB-Seite, 100 sec/acc

[gray87, gray97]

Nochmal zehn Jahre später...

2007

Neue Ebene der Speicherhierarchie



Neue Werte – Alte Formel

- Festplatte (4KB-Seiten), moderner Hauptspeicher:

$$\frac{256}{83} \cdot \frac{80}{0{,}047} \cdot {}^{sec}\!/_{acc} = 5248\,{}^{sec}\!/_{acc} \approx \underline{\underline{90\,{}^{min}\!/_{acc}}}$$

  - alte Regel nur mit 64KB-Seiten noch gültig (**334** sec/acc)

- SSD (4KB-Seiten), moderner Hauptspeicher:

$$\frac{256}{6300} \cdot \frac{999}{0{,}047} \cdot {}^{sec}\!/_{acc} = 876\,{}^{sec}\!/_{acc} \approx \underline{\underline{15\,{}^{min}\!/_{acc}}}$$

[graefe08]

Zwei neue Regeln

- billigere SSDs bereits verfügbar (ca. 400$)
- 4KB-Seiten zwischen Hauptspeicher und SSD

$$\frac{256}{6300} \cdot \frac{400}{0{,}047} \cdot {}^{sec}\!/_{acc} = 351\,{}^{sec}\!/_{acc} \approx \underline{\underline{6\,{}^{min}\!/_{acc}}}$$

- 256KB-Seiten zwischen SSD und Festplatte

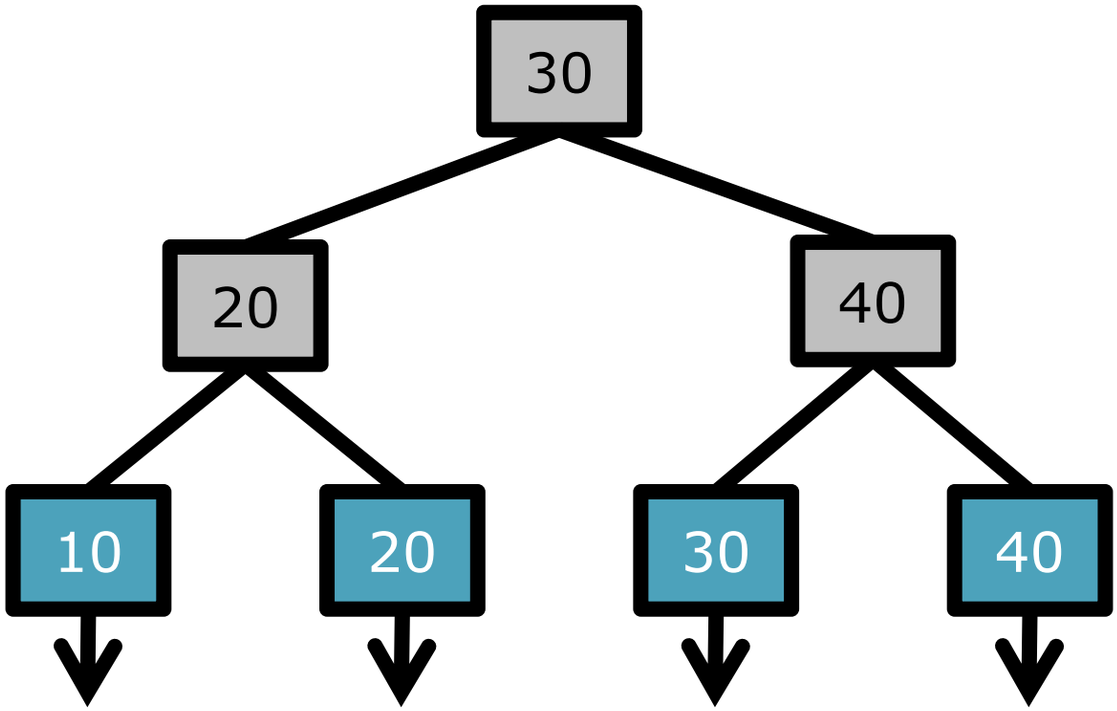
$$\frac{4}{83} \cdot \frac{80}{0{,}012} \cdot {}^{sec}\!/_{acc} = 321\,{}^{sec}\!/_{acc} \approx \underline{\underline{5\,{}^{min}\!/_{acc}}}$$

[graefe08]

Weitere Faustregeln

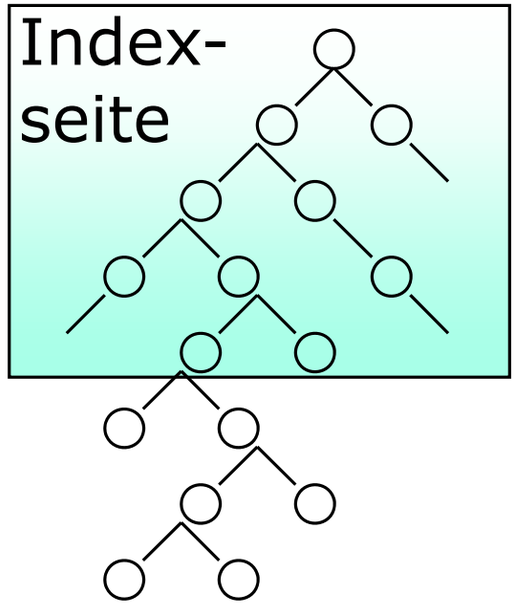
der Speicherverwaltung

Binäre Indexbäume



Größe einer Indexseite

- Bereitstellungs**kosten**
  - Lesen von Festplatte
- **"Fan-Out"**
  - Einträge pro Seite



- **Höhe** des Baumes in Indexseiten

$$IndexHeight \approx \frac{\log_2(N)}{\log_2(EntriesPerPage)}$$

- Kosten? Nutzen? **Optimale Größe?**

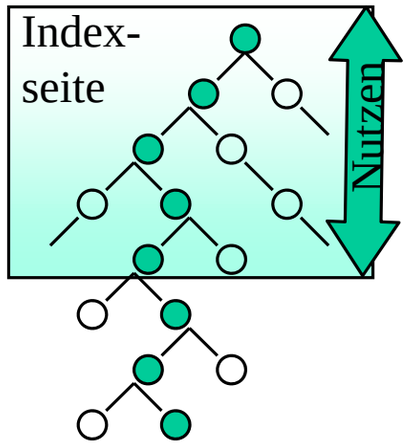
[gray97]

"Index Page Utility"

- Wie viele **Schritte** bringt mich eine **Seite** näher zum **Ziel**?

$$IndexPageUtility = \log_2(EntriesPerPage)$$

  - Anzahl der Ebenen im Baum pro Seite



- Beispiel
  - Größe pro Eintrag: **20 Bytes**
  - Größe einer Seite: **2000 Bytes**
  - Nutzen einer Seite: **6.2**

} ca. 70 Einträge bei 70% Füllstand

- **Zugriffskosten?**

[gray97]

Kosten-Nutzen-Verhältnis

- Zugriffskosten (Lesen der Seite von Festplatte)

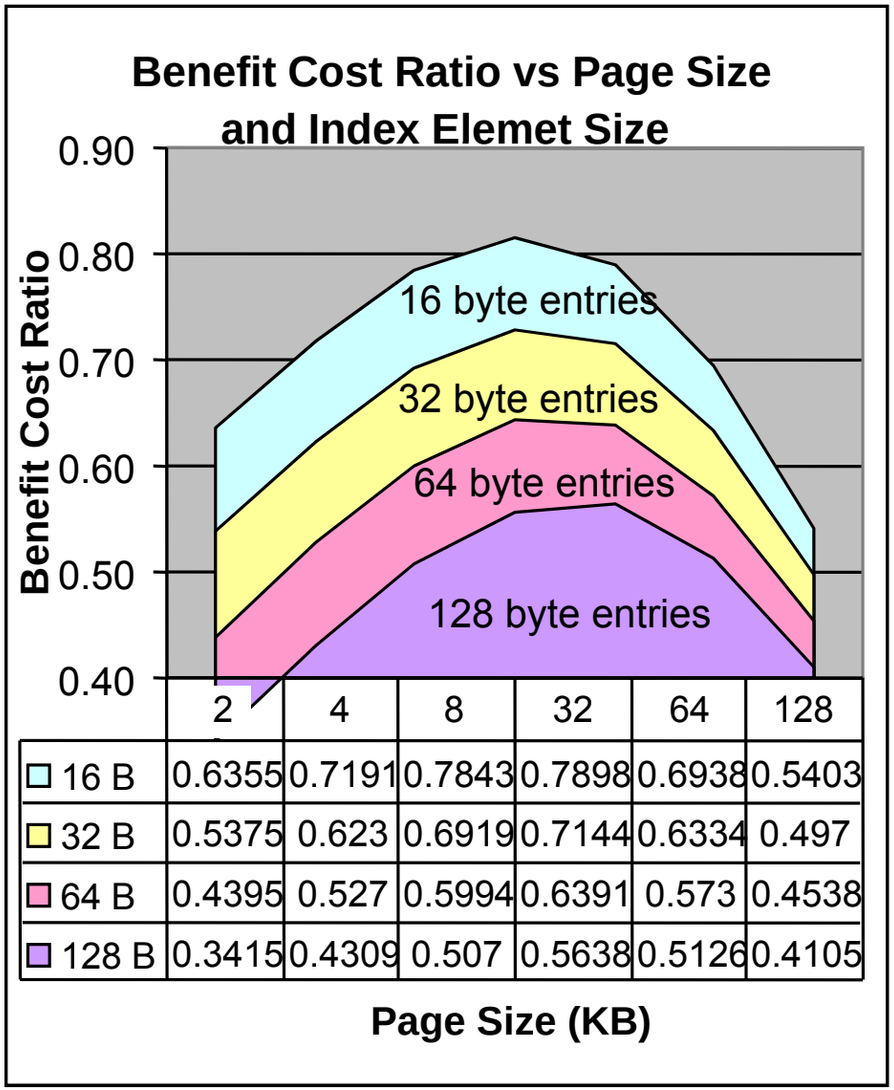
$$IndexPageAccessCost = DiskLatency + \frac{PageSize}{DiskTransferRate}$$

- Indikator für den wirtschaftlichen Wert der Seitengröße

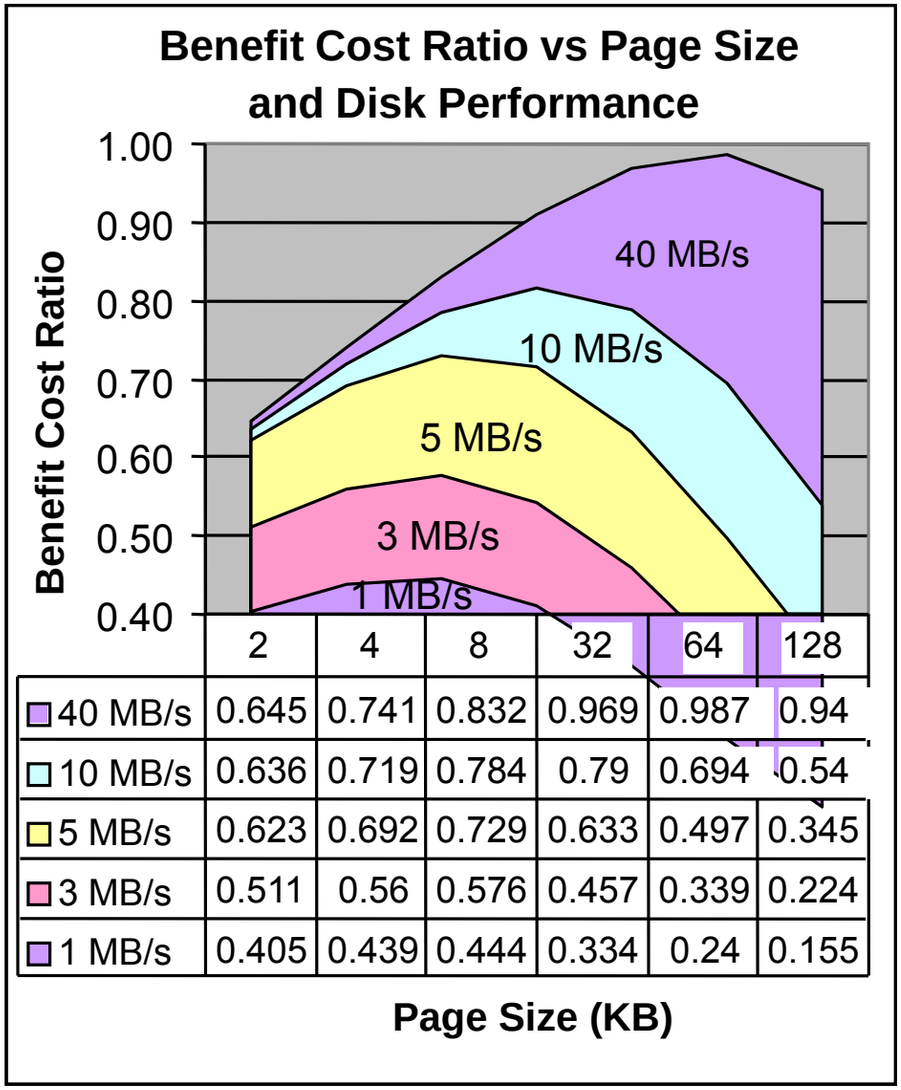
$$BenefitCostRatio = \frac{IndexPageUtility}{IndexPageAccessCost}$$

[gray97]

Optimale Größe einer Indexseite



	2	4	8	32	64	128
16 B	0.6355	0.7191	0.7843	0.7898	0.6938	0.5403
32 B	0.5375	0.623	0.6919	0.7144	0.6334	0.497
64 B	0.4395	0.527	0.5994	0.6391	0.573	0.4538
128 B	0.3415	0.4309	0.507	0.5638	0.5126	0.4105



	2	4	8	32	64	128
40 MB/s	0.645	0.741	0.832	0.969	0.987	0.94
10 MB/s	0.636	0.719	0.784	0.79	0.694	0.54
5 MB/s	0.623	0.692	0.729	0.633	0.497	0.345
3 MB/s	0.511	0.56	0.576	0.457	0.339	0.224
1 MB/s	0.405	0.439	0.444	0.334	0.24	0.155

[gray97]

Aktuelle Werte (2007)

- Eintrag: 20 Bytes, Seite: 2000 Bytes

- Indexseiten auf **Festplatte**
  - Optimale Größe: **256 KB** → Neue 5MR für 256-KB-Seiten
  - Nutzen: **13**
  - Kosten-Nutzen-Verhältnis: **1.01**
- Indexseiten auf **Flashspeicher**
  - Optimale Größe: **2 KB** → Neue 5MR für **4**-KB-Seiten
  - Nutzen: **6**
  - Kosten-Nutzen-Verhältnis: **46.1**

- **SB-Bäume** berücksichtigen zwei Seitengrößen, siehe [oneil92]

[graefe08]

Sequentieller Zugriff

- **höhere Bandbreite** bei sequentiellem Lesen der Daten
  - 10x langsamer bei wahlfreiem Zugriff
- andere Betrachtung für **sequentielle I/O-Operationen**
  - z. B. **Sort**, Cube, Hash-Join, Rollup
- **TPMMS** vs. One-Pass-Sort
  - doppelt so viel I/O
  - viel weniger Hauptspeicher
- **Wann ist es ratsam, einen One-Pass-Sort zu nutzen?**

[gray97]

Neue Regel - Herleitung

- neuer Break-Even-Point (64-KB-Seiten)
  - höhere Transferrate (ca. 5 MB/s)
  - doppelte Intervallgröße für **Lesen und Schreiben**

$$ReferenceInterval = 2 \cdot \frac{PagesPerMBofRAM}{AccessPerSecondPerDisc} \cdot \frac{PricePerDiskDrive}{PricePerMBofRAM}$$

$$ReferenceInterval = 2 \cdot \frac{16}{80} \cdot \frac{2000}{15} \cdot {}^{sec}\!/_{acc} = 2 \cdot 26 \, {}^{sec}\!/_{acc} \approx 1 \, {}^{min}\!/_{acc}$$

- bei hohem Datentransfer sind Zugriffskosten vernachlässigbar
  - nur Transferrate ausschlaggebend

[gray97]

"One-Minute Sequential Rule" (1997)

Sequentielle Operationen, welche Daten innerhalb von **einer Minute wiederholt referenzieren**, sollten diese Daten im **Hauptspeicher** halten.

- Beispiel
  - One-Pass-Sort verarbeitet **5 GB / Minute**
    - variiert je nach Algorithmus
  - **ab 5 GB** Relationsgröße **Two-Pass**-Sort nutzen

[gray97]

Weitere Themenbereiche

- **Energiekosten** berücksichtigen
  - nicht nur Anschaffungskosten
- Hauptspeicher vs. CPU-Zyklen
  - **"10-Byte Rule"** (1987)
- Richtlinien für Datenbewegung in **Speicherhierarchie**
  - Manuell (Admin) vs. Automatisch (LRU)
- Nebeneffekte des Flashspeichers
  - Auswirkungen auf **paralleles Programmieren**
- Skalierbarkeit von **In-Memory-Datenbanken**
- generative **Garbage-Collection**

[graefe08]

Quellen

Stand: 30.12.2008

[gray87] J. Gray, F. Putzolu, "The 5 Minute Rule for Trading Memory for Disc Access and The 10 Byte Rule for Trading Memory for CPU Time", ACM, 1987

[gray97] J. Gray, G. Graefe, "The Five-Minute Rule Then Years Later and Other Computer Storage Rules of Thumb", SIGMOD Record, 1997

[graefe08] G. Graefe, "The Five-minute rule – 20 Years Later and How Flash Memory Changes the Rules", ACM Queue, 2008

[oneil92] P. E. O'Neil, "The SB-tree: An Index-Sequential Structure for High-Performance Sequential Access", Acta Informatica, 1992