Methodik zur Optimierung in Datenbanken

Anja Rommel, 14-INM

03.07.2015

Gliederung

1. Einleitung

- Was versteht man denn unter der Optimierung einer Datenbank ?

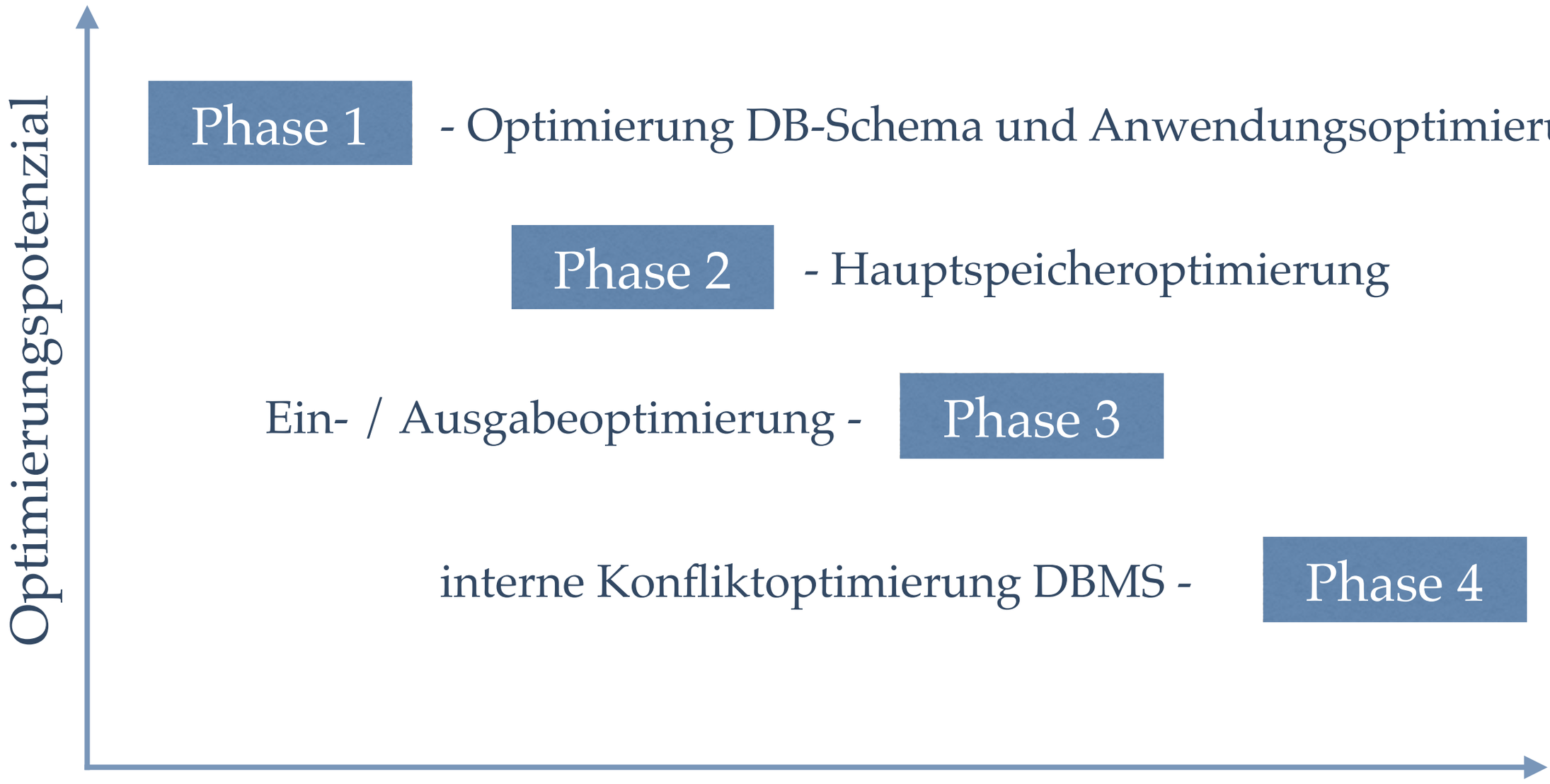
- nutzen der Möglichkeiten des Gesamtsystems (DBMS, DB, Anwendung, Betriebssystem, Netzwerk)

- gemessen an den Anforderungen der Anwendung

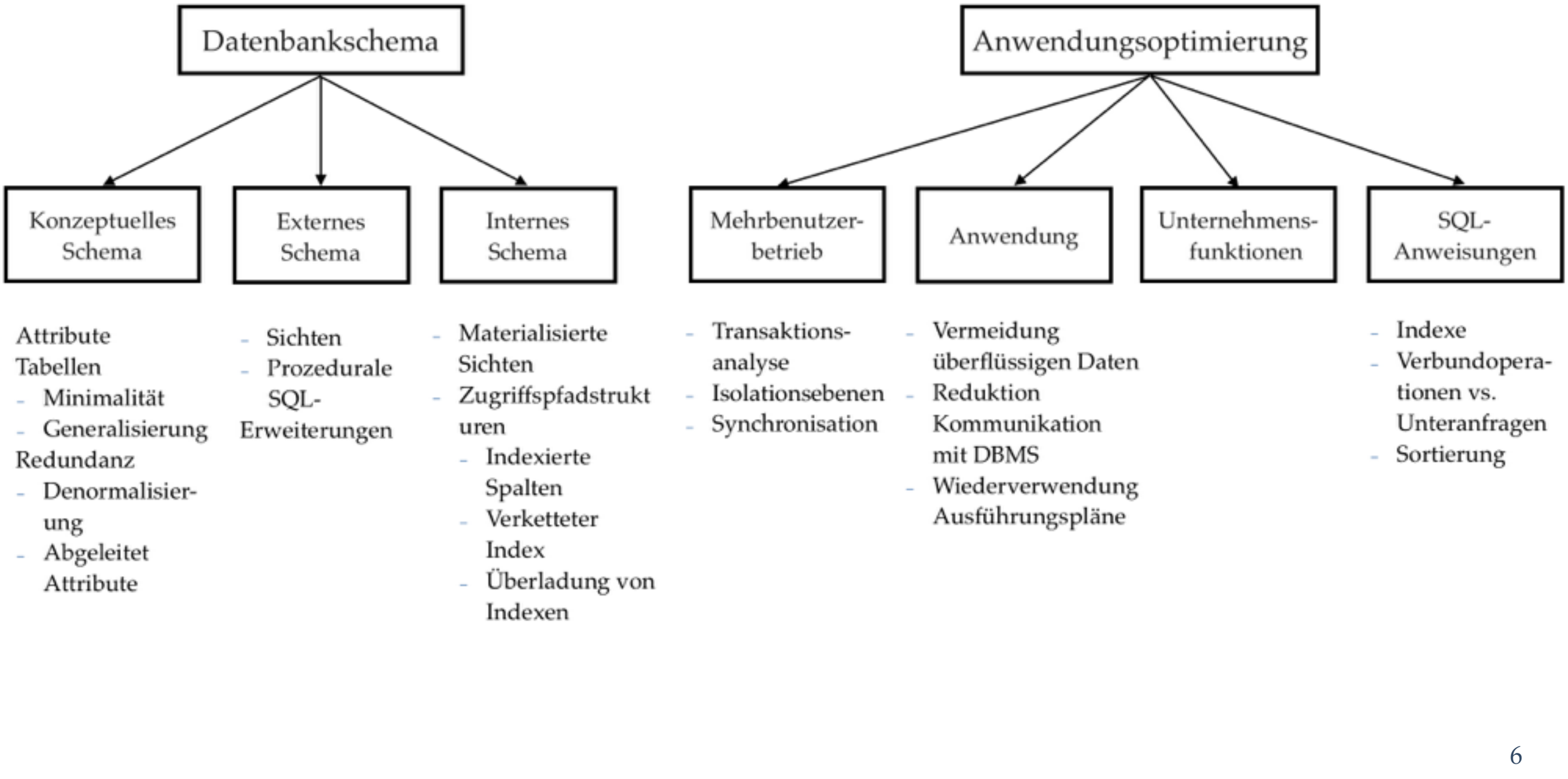
2. Motivation und Ziele

- keine zusätzliche Hardware
- kostengünstiger Wartung
- schnellere Antwortzeiten —> höherer Datendurchsatz

3. Phasen der Optimierung



3.1 Phase 1: Optimierung des DB-Schemas und Anwendungsoptimierung



Optimierung DB-Schema

- Ziel: Speichereffizienz und Zugriffseffizienz optimieren

- Optimierungsmöglichkeiten auf allen 3 Ebenen
  - konzeptuelles Schema
  - internen Schema
  - externen Schema

konzeptuelles Schema

- Attribute
  - verzicht auf Datentypen variabler Länge (VARCHAR)
- Tabellen
  - Minimalität (Anzahl der Tabellen)
  - Generalisierung/Spezialisierung
    - Konflikt zwischen Konsistenz und Ausführungszeit
- Redundanz
  - Denormalisierung (Laufzeit vs. referenzielle Integrität)
  - Abgeleitete Attribute (Vermeidung berechneter Werte als Attribute)

externes Schema

- Sichten
  - lohnen sich nur wenn nicht eine einzelne Zeile oder Spalte benötigt werden
- Prozedurale SQL-Erweiterungen
  - entlasten Anwendung und Netzwerk
  - Beurteilung Hardware des DB-Rechners

internes Schema

- Materialisierte Sichten
  - erzeugen Redundanzen (Reports, OLAP, Spiegelung von Tabellen)
- Zugriffspfadstrukturen
  - indexierte Spalten
  - verketteter Index
  - Überladung von Indexen

Anwendungsoptimierung

- Zusammenfassen von Maßnahmen, welche eine anwendungsspezifische Anpassung der beteiligten Systemkomponenten an die konkreten Anforderungen der Anwendungsumgebung ermöglichen

Unternehmungsfunktionen und Mehrbenutzerbetrieb

- Optimierung von Unternehmensfunktionen
  - Optimierung der Arbeitsabläufe
- Optimierung im Mehrbenutzerbetrieb
  - Transaktionanalyse (kurze Transaktionen)
  - Ebenen der Isolation
  - Synchronisation

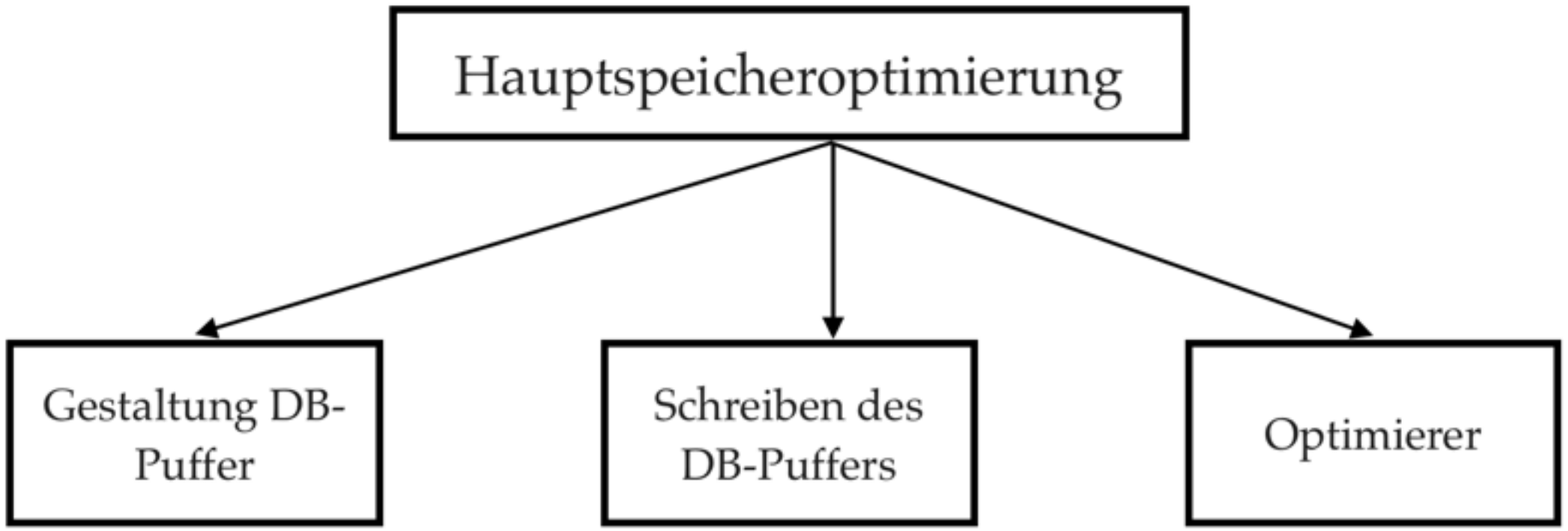
Optimierung der Anwendung

- Vermeidung überflüssiger Daten
  - nur benötigte Spalten abfragen
- Reduktion der Kommunikation mit dem DBMS
- Wiederverwendung von Ausführungsplänen
  - SELECT * FROM Mitarbeiter WHERE Name = ‚Lehmann‘;
  - SELECT * FROM Mitarbeiter WHERE Name = ‚Lehmann‘;
  - select * from Mitarbeiter where Name = ‚Lehmann‘;
  - -> unterschiedliche Ausführungspläne

SQL-Anweisungen

- Nutzung von Indexen
  - keine Funktionen auf Indexierte Spalten anwenden
- Verbundoperation vs. Unteranfragen
  - wenn gleiche Ergebnismengen -> Verbundoperationen
- Unteranfragen
  - Reihenfolge nach Selektivität der Anfragen
- Sortierung von Datensätze

3.2 Phase 2: Hauptspeicheroptimierung



- Komponenten des DB-Puffers
- Größe des DB-Puffers
- Blockfüllgrad

- Sicherungspunkt-Intervalle
- Protokolldatei

- Statistiken
- Planhinweise

Gestaltung DB-Puffers

- Eine beliebige Vergrößerung des Hauptspeichers muss nicht zwangsläufig zu weiteren Leistungssteigerungen führen.

- Komponenten:
  - **Datenpuffer:** Daten der Tabellen und Indexe
  - **SQL-/Prozedurpuffer:** optimierte SQL-Anweisungen
  - **Protokollpuffer:** Log-Einträge

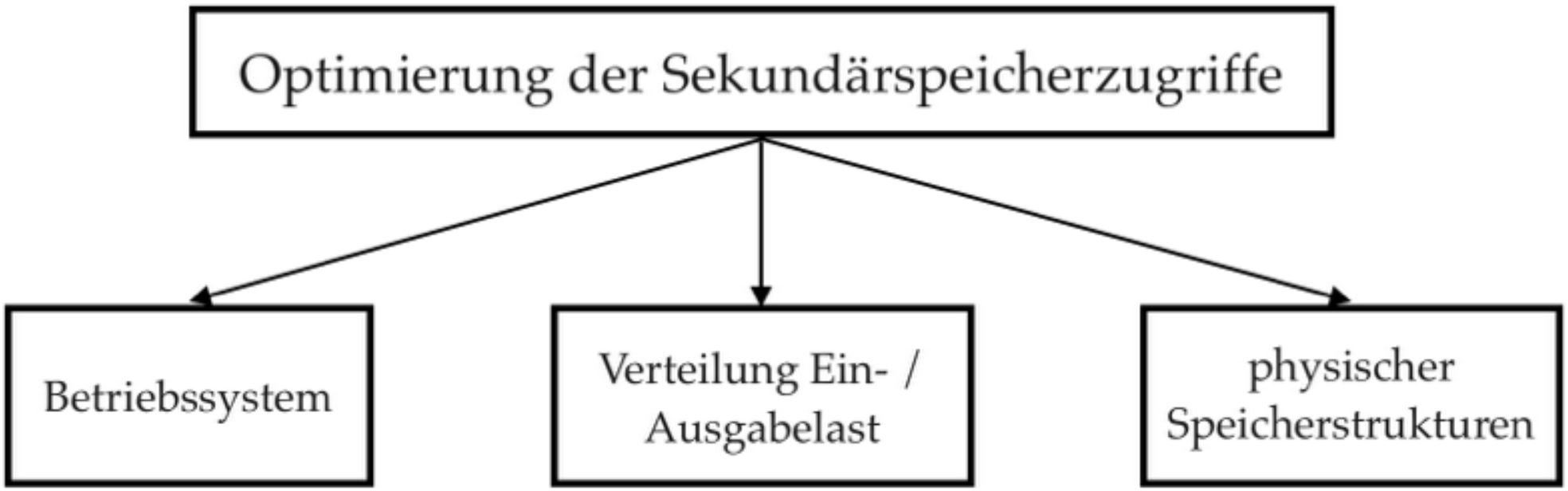
Schreiben DB-Puffer

- erfolgt spätestens wenn DB-Puffer voll
- Sicherungspunkt-Intervalle
  - nicht zu häufig und nicht zu selten
- Protokolldatei
  - bei Wechsel auslösen eines Sicherungspunktes

Optimierer

- legt Ausführungsplan fest
  - erst günstige Operationen, dann teure
- **regelbasierte Optimierer**
  - verwendet „in der Regel“ günstigere Operationen zuerst
- **kostenbasierte Optimierer**
  - verwendet tatsächliche günstigere Operationen zuerst
- Statistiken
- Planhinweise (Wahl Optimierer, Nutzung von Indexen, Reihenfolge Verbundoperation, Ausführung Verbundoperationen)

3.3 Phase 3: Eingabe- / Ausgabeoptimierung



Betriebssystem

- Übergabe an BS
  - DBMS -> Hauptspeicherpuffer BS -> Protokoll Dateisystempuffer -> Festplatte
- Raw-Device
  - direkter Zugriff auf Festplattenpartition

Verteilung Ein-/Ausgabelast

- Verteilung Daten- und Indexdateien
- Verteilung der Daten
  - Partitionierung -> Parallelisierung
- Verteilung der Protokolldatei
  - andere Platten als Datendateien
- RAID-Level (Kombination 0 und 1)

Optimierung physischer Speicherstrukturen

- Blockgröße
- Clustertechniken
  - ermöglichen die gebündelte Speicherung zusammenhängender Datensätze möglichst in einem Block
- Kompressions-Techniken
- Reorganisation
  - zusammenhängende Speicherung der Daten wiederherstellen

3.4 Phase 4: interne Konfliktoptimierung des DBMS

- Mechanismen zur Lösung von Zugriffskonflikten im Bereich der internen logischen Struktur
- Ursache: Vielzahl Prozesse nutzen interne Betriebsmittel
- Optimierung: Abhängig vom eingesetzten DBMS

Quellen

- Harm Knolle: Optimierung von Datenbanken und Leistungsbewertung, in Taschenbuch Datenbanken, Hanser, 2007

- https://static.aminer.org/pdf/PDF/000/238/087/entity_relationship_schema_optimierung.pdf

- Database Tuning: Principles, Experiments, and Troublshooting Techniques von Dennis Shasha http://bigbe.su/lectures/2014/06.1.pdf