

Kap. 1: Einführung IDBS

- Aufbau (Architektur) von DBS
 - 3-Schichtenmodell
- Datenbankadministration
- Tuning von DBS

Merkmale von Datenbanksystemen

- mächtige Datenmodelle (v.a. relationale DBS...) und Anfragesprachen (SQL)
- hohe Datenunabhängigkeit
- effiziente Verwaltung großer Datenmengen
- Transaktionskonzept (ACID)
 - Mehrbenutzerfähigkeit
 - automatische Fehlerbehandlung
 - Integritätsicherung

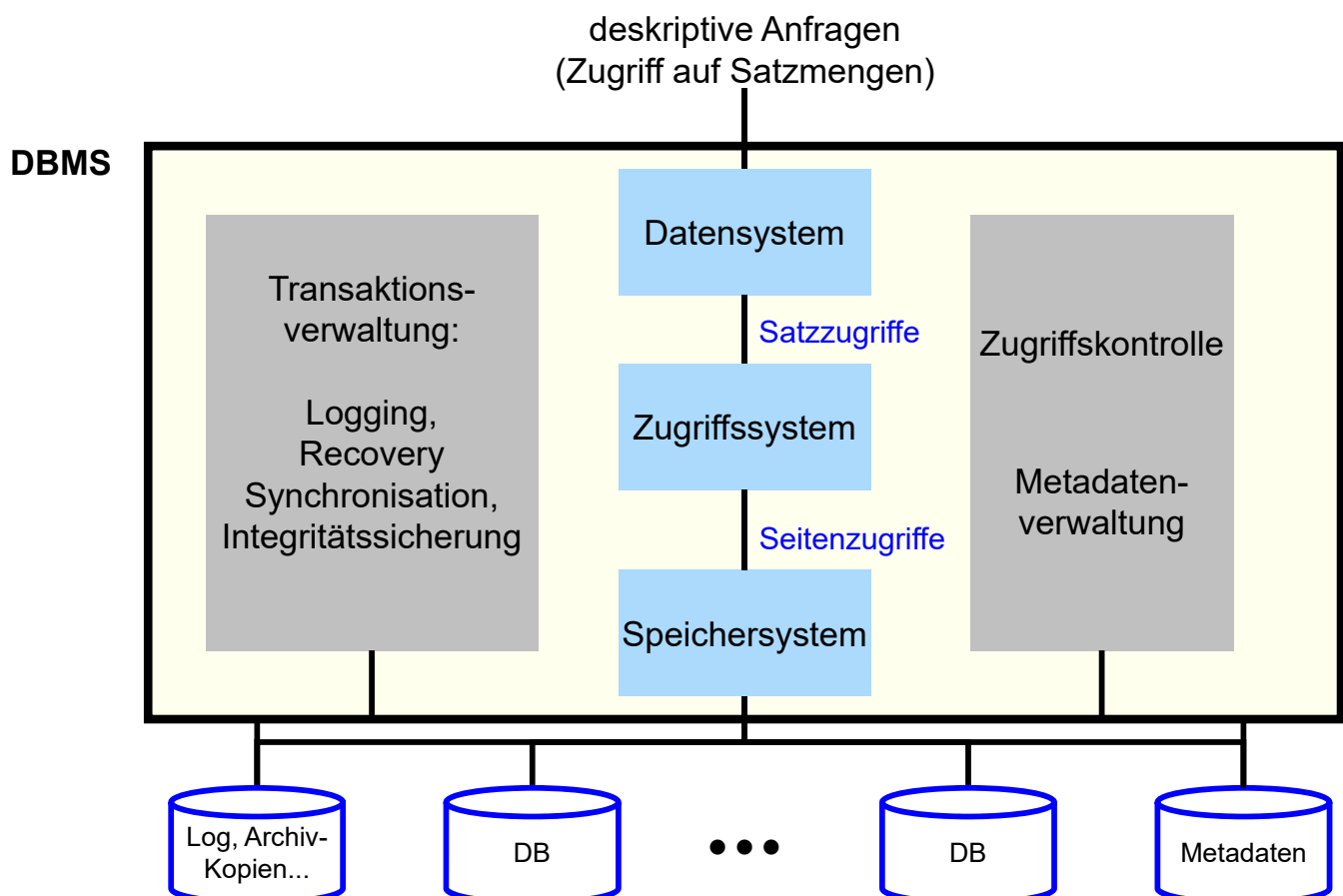
Einsatzformen von Datenbanksystemen

- Anwendungen im Unternehmensbereich:
 - Online-Shops, Produktionsplanung und -steuerung, Personalverwaltung, Datenerfassung ...
- OLTP (kurze Transaktionen) und OLAP (Analyse)
- Verwaltung von wissenschaftlichen Daten, Dokumenten ...

■ Architekturausprägungen

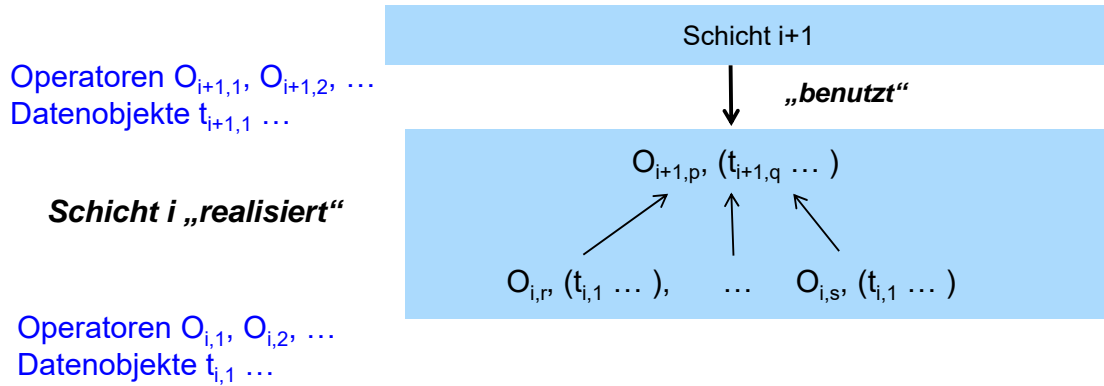
- zentraler Server, z.B. mit Web-Anbindung über Applikations-Server
- Parallele DBS im lokalen Cluster
- heterogene verteilte DBS (z.B. Mediator-gekoppelt)

Grobaufbau eines DBS



Schichtenmodelle

- Ziel: Architektur eines datenunabhängigen DBS
- Bildung von konzeptionellen Schichten zur hierarchischen Strukturierung
- jede Hierarchieebene kann als virtuelle Maschine aufgefasst werden

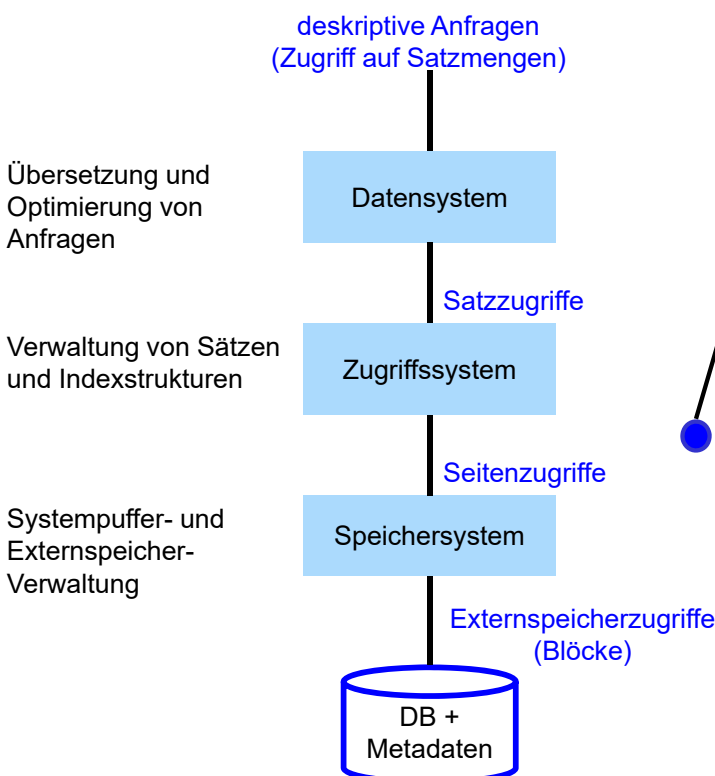


Vorteile hierarchischer Schichtenbildung

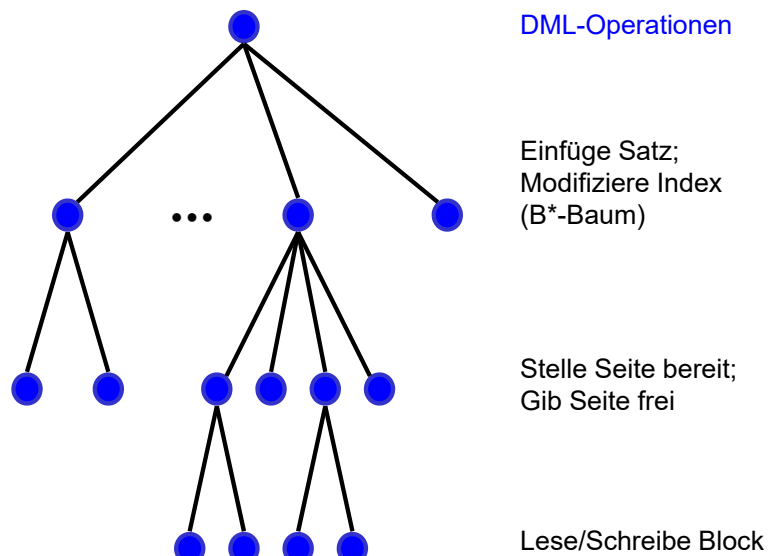
- höhere Ebenen (Systemkomponenten) werden einfacher, weil sie tiefere Ebenen (Systemkomponenten) benutzen können
- tiefere Ebenen können getestet werden, bevor die höheren Ebenen lauffähig sind
- Änderungen in Ebenen werden gekapselt

Grobaufbau eines DBS (2)

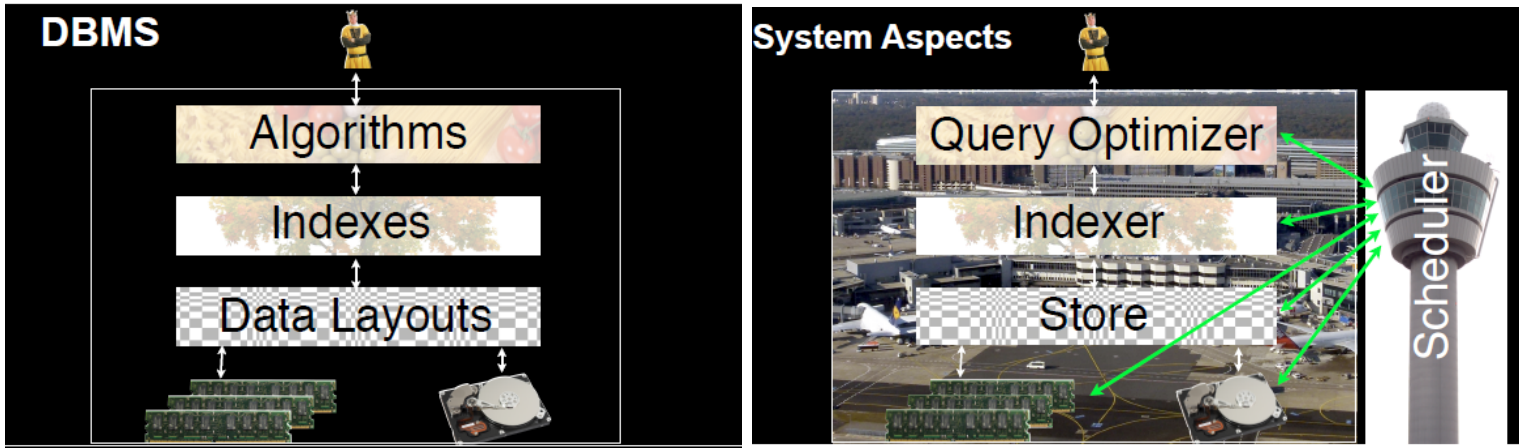
Schichtenmodell



Dynamischer Kontrollfluss einer DB-Operation

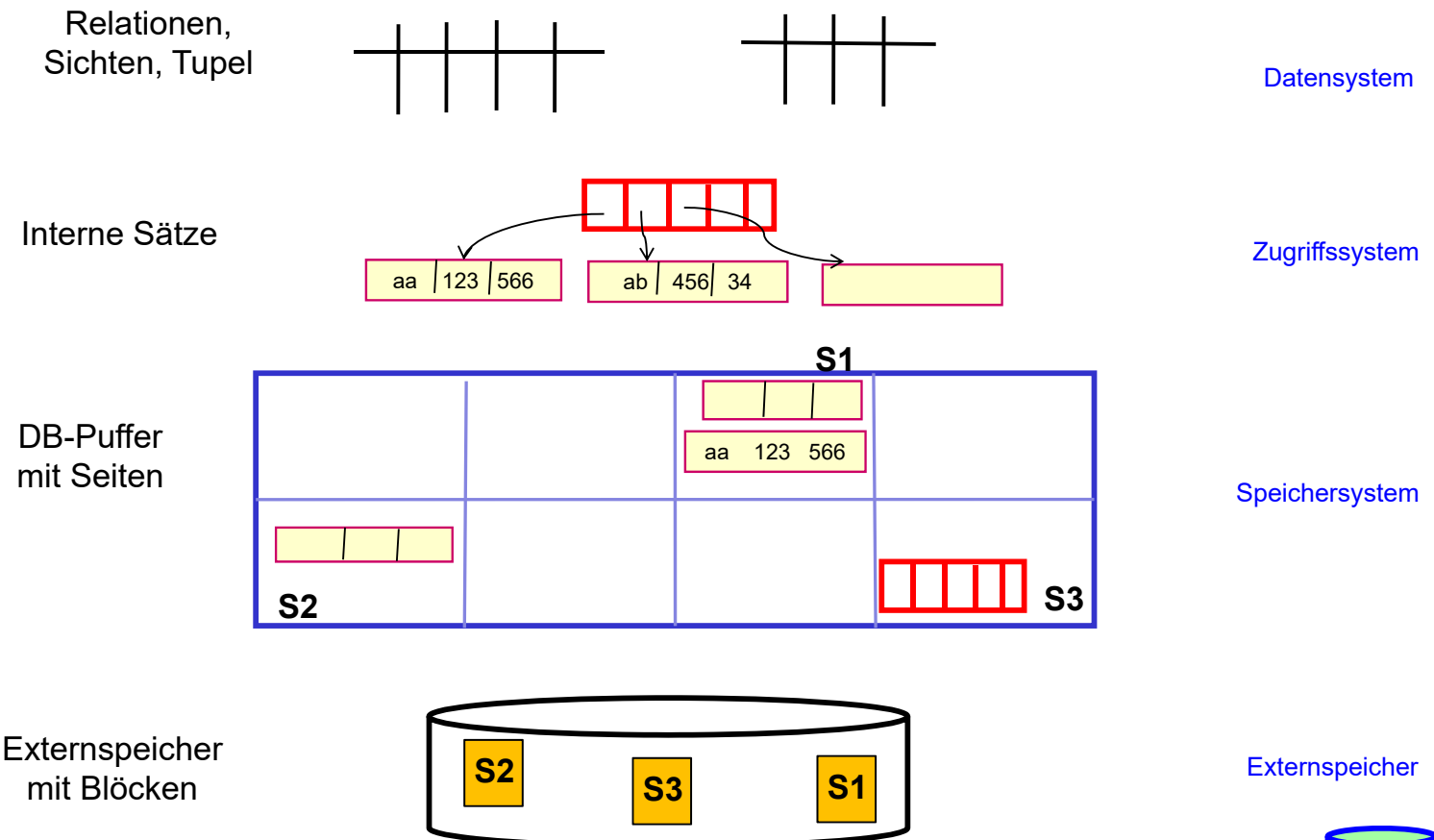


Alternative Architektursicht*



* Dittrich: Patterns in Data Management, 2016

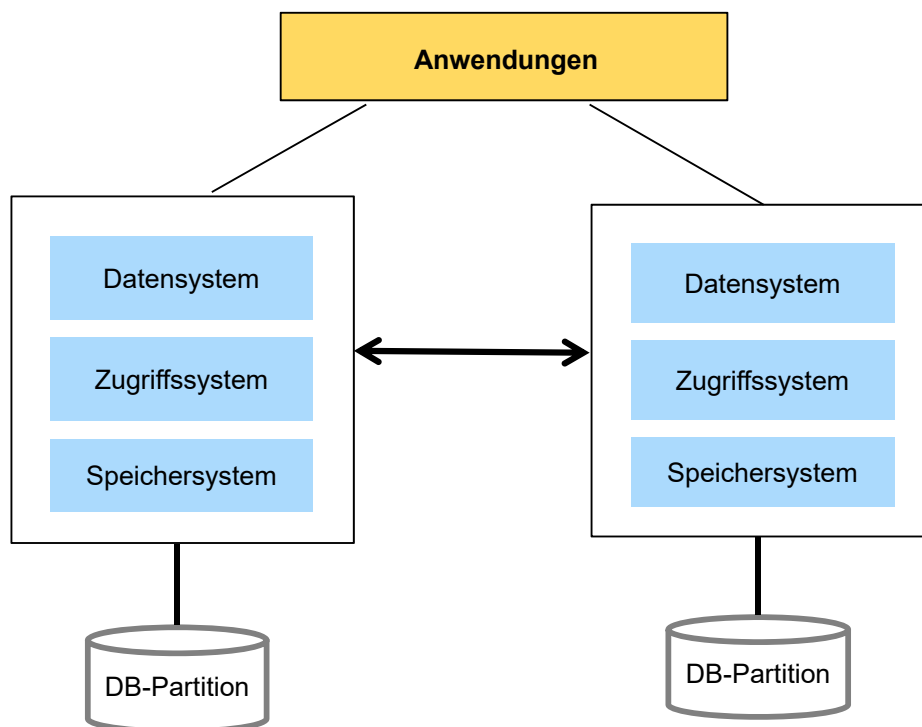
Beispiel für Daten-Abbildung



Datenunabhängigkeit im Überblick

Ebene	Was wird verborgen?
Datensystem	Indexstrukturen; interne Satzdarstellung
Zugriffssystem	Pufferverwaltung
Speichersystem	Dateiabbildung, technische Eigenschaften der externen Speichermedien

Architektur von Mehrrechner-DBS



- Knoten in Mehrrechner-DBS haben ähnliche Funktionen wie ein zentrales DBS
- unterschiedliche Möglichkeiten der Kooperation zwischen DBS

Aktuelle Entwicklungen

■ In-Memory-Datenbanksysteme

- Server-Systeme mit Hauptspeicher im TB-Bereich: Datenbanken können komplett im Hauptspeicher gehalten werden
- hohe Kosten
- Speichersystem entfällt
- geänderte Verfahren im Zugriffssystem / Datensystem
- zugeschnittene Verfahren zur Transaktionsverwaltung (Synchronisation, Logging/Recovery)
- Optimierung von Cache-Zugriffen erforderlich

■ Einsatz von Hardware-Beschleunigern

- Nutzung von Ko-Prozessoren (GPUs / FPGAs) zur Beschleunigung aufwändiger Operationen

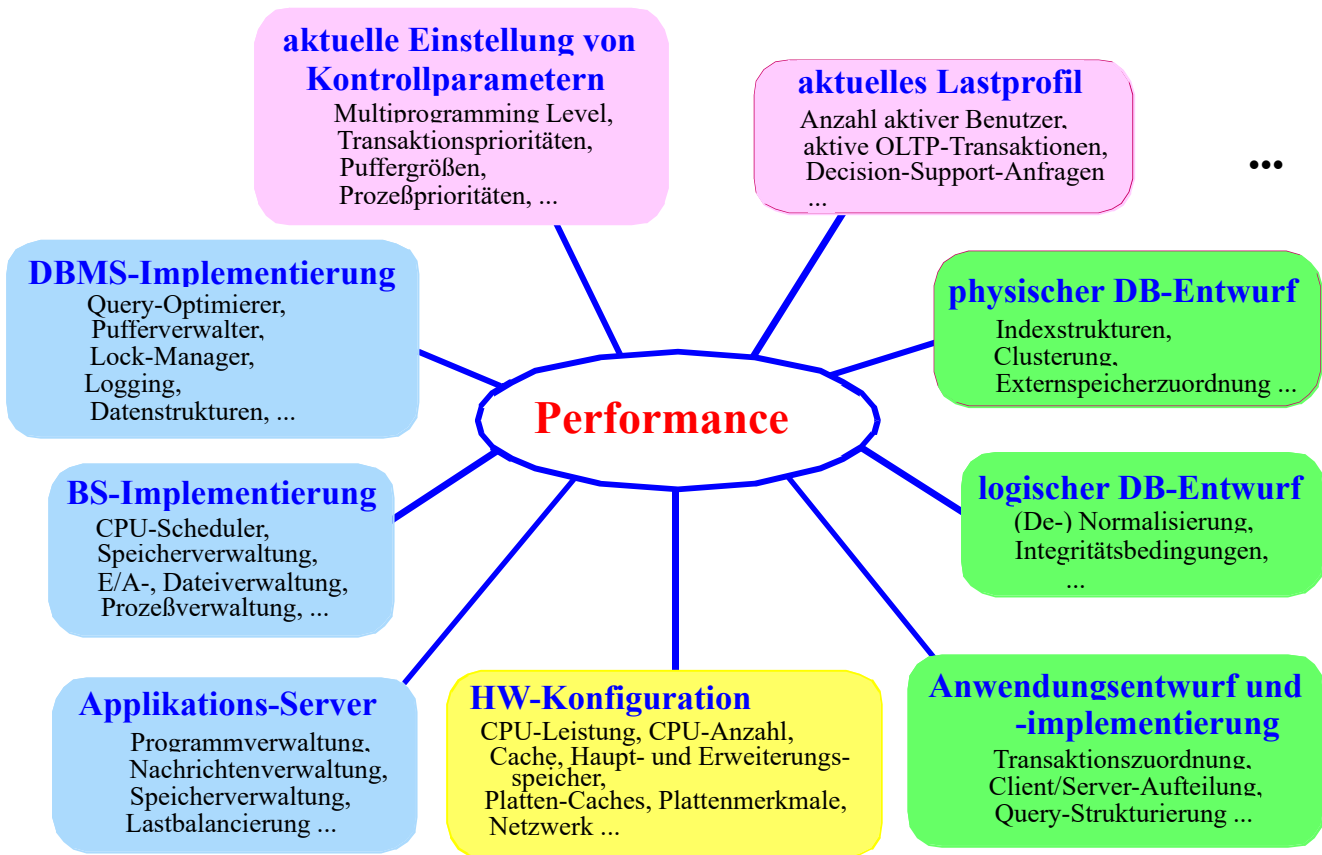
Datenbankadministrator (DBA)

■ Aufgaben

- Festlegung des internen (physischen) DB-Schemas
- ggf. Entwurf des konzeptionellen DB-Schemas mit Festlegung von Integritätsbedingungen
- Kontrolle über externe Schemata
- Vergabe / Entzug von Zugriffsberechtigungen
- Strategie für Datensicherung und Recovery festlegen
- Installation neuer DBMS-Releases
- Überwachung des Systembetriebs
- **Performance-Tuning** (höherer Durchsatz, bessere Antwortzeiten) ...

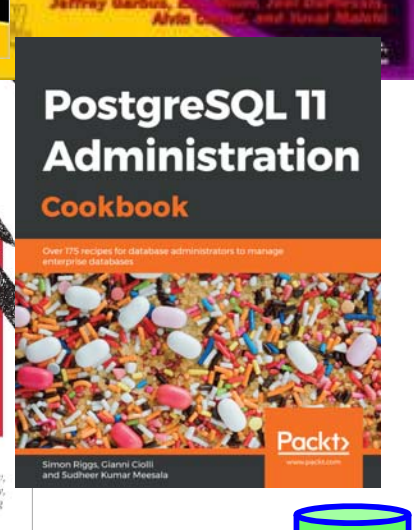
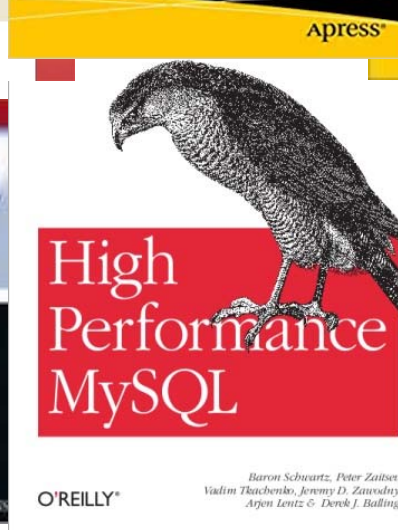
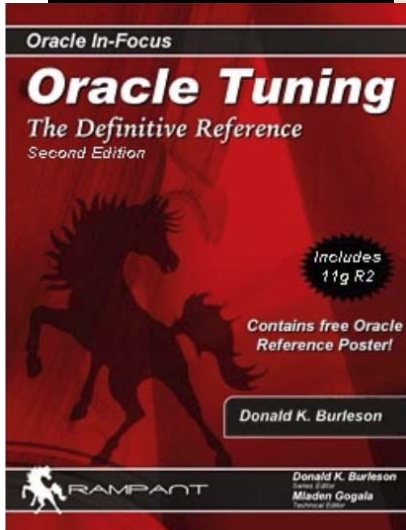
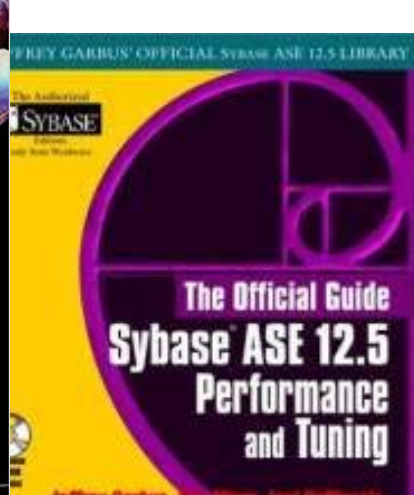
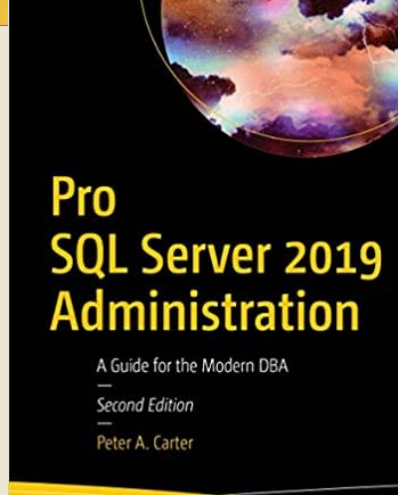
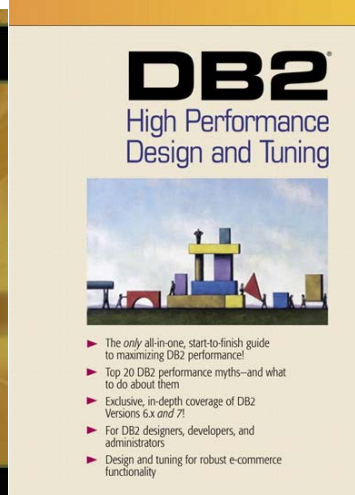
■ Unterstützung durch Dienstprogramme, z.B. für

- Laden der DB
- Erstellen von Archivkopien
- DB-Reorganisation
- Erfassung und Analyse von Messwerten



DBS-Kontrollparameter (Bsp. Sybase)

Function	Default	Function	Default	Function	Default
max online engines	1	deadlock checking period	500	default fill factor percent	0
# user connections	25	deadlock retries	5	i/o accounting flush interval	1000
# remote connections	20	page lock spinlock ratio	100	max network packet size	512
total memory	-	table lock spinlock ratio	20	remote server pre-read packets	3
user log cache size	2 KB	partition spinlock ratio	10	stack size	-
procedure cache percent	20	user log cache spinlock ratio	20	time slice	100
number of sort buffers	0	number of open databases	12	event buffers per engine	100
sort page count	0	number of open objects	500	freelock transfer block size	30
housekeeper free write percent	1	number of index trips	0	max engine freelocks	10
recovery interval	5 min.	i/o polling process count	10	page utilization percent	95
additional network memory	0	disk i/o structures	256	partition groups	1024
audit queue size	100	max async i/os per engine	2 Mrd.		
number of locks	5000	max async i/os per server	2 Mrd.		
lock promotion HWM	200	# pre-allocated extents	2		
lock promotion LWM	200	number of devices	10		
lock promotion PCT	100	# extent i/o buffers	0		



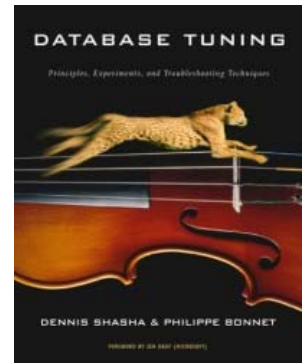
Tuning-Probleme

- in großen DBS-Installationen noch weitgehend manuelles Tuning, u.a. durch DBA
- komplizierte Systemverwaltung durch Vielzahl von internen Parametern und Abhängigkeiten, u.a. zu Betriebssystem
- komplexe und sich ändernde Transaktions- und Anfragelasten
- Verschärfung der Situation in verteilten Systemen / Big Data Infrastrukturen
 - stark zunehmende Anzahl von Kontrollparametern
 - Festlegung der Verteilung von Daten, Programmen und Lasteinheiten
 - komplexere Lastsituationen
 - ggf. heterogene Systemstruktur



Tuning-Leitlinien*

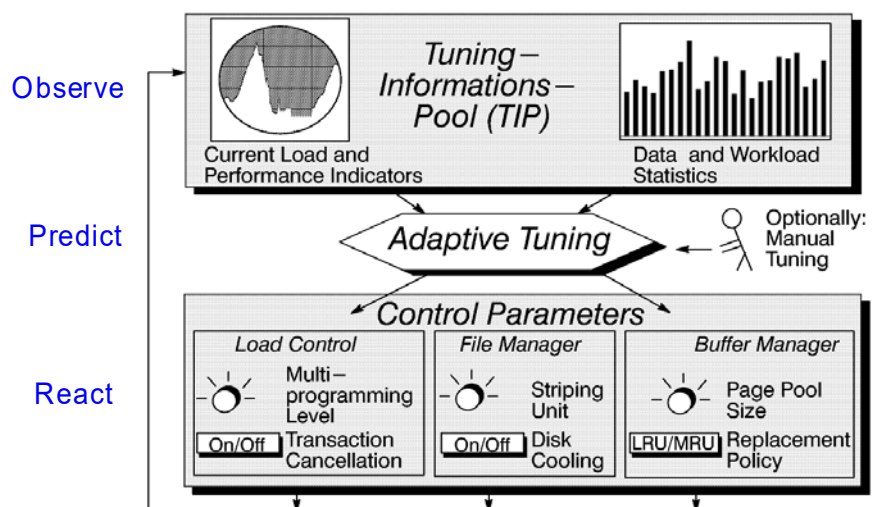
- Think globally, fix locally (does it matter?)
 - slow query. Tune it? (Check: Is query frequent enough?)
 - disk is saturated. Buy new disk ? (Check: scans instead of index usage? log and data mixed?)
- Partitioning breaks bottlenecks (spatial and temporal)
 - dividing work across several resources to avoid saturation
 - e.g. data partitioning; multiple data structures; splitting of transactions;
- Start-up costs are high; running costs are low
 - disk access (number of IOs more significant than data volume)
 - client/server communication (number of messages more critical than message size)
 - high cost of query parsing (save compiled query)
- Be prepared for trade-offs
 - indexes and inserts
 - Shasha, D./Bonnet, P.: *Database Tuning*. Morgan Kaufmann, 2003
 Online: <http://www.sciencedirect.com/science/book/9781558607538>



Automatisches Tuning?

- zunehmende Unterstützung durch „Advisors“ für Indexierungsvorschläge, materialisierte Sichten etc. (Microsoft SQL-Server, DB2, Oracle ...)
 - erfordert Spezifikation zu optimierender Anfragen / Transaktionen für gegebene DB
- weitergehende Zielvorstellung: Self-Tuning / Configuration

- Online-Monitoring
- automatische Erkennung von Leistungsproblemen
- automatische Bestimmung der Engpässe
- automatische Linderung der Probleme durch geeignete Anpassung von Kontrollparametern



- Probleme:
 - Stabilität, hohe Dynamik und große wechselseitige Abhängigkeiten
 - Effizienz vs. Effektivität



Zusammenfassung

- Schichtenmodell ist allgemeines Erklärungsmodell für die Realisierung von DBS
 - unterschiedliche Schichtenbildungen möglich (z.B. 3-Schichten-Ansatz)
 - Schichtenbildung unterstützt Datenunabhängigkeit
- Implementierungskonzepte zentralisierter DBS finden sich auch in Mehrrechner-DBS
 - jeder Knoten verfügt über alle Funktionen eines zentralisierten DBS
- DBA: verantwortlich für sicheren Betrieb und effektive Nutzung großer Datenbanken
- DB-Tuning (Performance-Optimierung)
 - hohe Komplexität aufgrund zahlreicher Parameter, wechselseitigen Abhängigkeiten und hoher Dynamik
 - zunehmende Tool-Unterstützung / Automatisierung für Teilaufgaben