Prof. Dr. Th. Härder
Universität Kaiserslautern
Fachbereich Informatik
Zi. 36/330, Tel.: 0631-205-4030
E-Mail: haerder@informatik.uni-kl.de
http://wwwdbis.informatik.uni-kl.de

Verteilte und Parallele Datenbanksysteme

Sommersemester 2002

Universität Kaiserslautern
Fachbereich Informatik
Postfach 3049
67653 Kaiserslautern

Vorlesung:

Ort: 42 - 110

Zeit: Mi., 10.00 - 11.30 Uhr
Beginn: Mi., 17. 4. 2002

Vorläufiges Inhaltsverzeichnis (1)

Vorläufiges Inhaltsverzeichnis (2)

5. Verteilte Anfrageverarbeitung

- Phasen der verteilten Anfragebearbeitung
- Übersetzung und Optimierung
- verteilte Ausführung - Algorithmen
- Parallelisierung einfacher Operatoren
- Parallele Join-Algorithmen

6. Verteilte Transaktionsverwaltung

- Commit-Protokolle
- Synchronisation
- Deadlock-Behandlung
- Logging und Recovery

7. Replizierte Datenbanken

- Motivation und Einführung
- Aktualisierungsstrategien bei strenger Konsistenz
- Schwächere Formen der Datenreplikation
- Datenreplikation in Parallelen DBS

8. Shared-Disk-DBS (DB-Sharing)

- Architektur und Probleme
- Synchronisation
- Behandlung von Pufferinvalidierungen
- Einsatz einer nahen Rechnerkopplung

9. . . .

LITERATURLISTE

Lehrbücher:

Rahm, E.: *Mehrrechner-Datenbanksysteme: Grundlagen der verteilten und parallelen Datenbankverarbeitung*. Addison-Wesley, 1994

Dadam, P.: *Verteilte Datenbanken und Client/Server-Systeme,* Springer-Verlag, 1996

Özsu, M.T.; Valduriez, P.: *Principles of Distributed Database Systems*, 2nd Edition, Prentice Hall, 1999

Gray, J., Reuter, A.: *Transaction Processing - Concepts and Techniques*, Morgan Kaufmann Publishers Inc., San Mateo, CA., 1993.

Bell, D., Grimson, J.: *Distributed Database Systems*, Addison-Wesley, 1992

Rahm, E.: *Hochleistungs-Transaktionssysteme,* Vieweg-Verlag, 1993

Aufsätze:

DeWitt, D., Gray, J.: *Parallel Database Systems: The Future of High Performance Database Systems*, in: CACM 35:6, June 1992, pp.85-98

Reuter, A.: *Grenzen der Parallelität*, in: Informationstechnik IT 34:1, 1992, S. 62-74

Allgemein:

Härder, T., Rahm, E.: Datenbanksysteme — Konzepte und Techniken der Implementierung, 2. Auflage, Springer-Verlag, 2001 (Es sind Hörerscheine erhältlich)

Weikum, G., Vossen, G.: *Transactional Information Systems - Theory, Algorithms, and Practice of Concurrency Control and Recovery*, Morgan Kaufmann Publ., 2002

1. Einführung

- **Technologievorhersage**
- **Allgemeine Definition von Mehrrechner-DBS**

  Einsatz mehrerer Rechner/DBVS zur koordinierten Verarbeitung von Datenbankoperationen

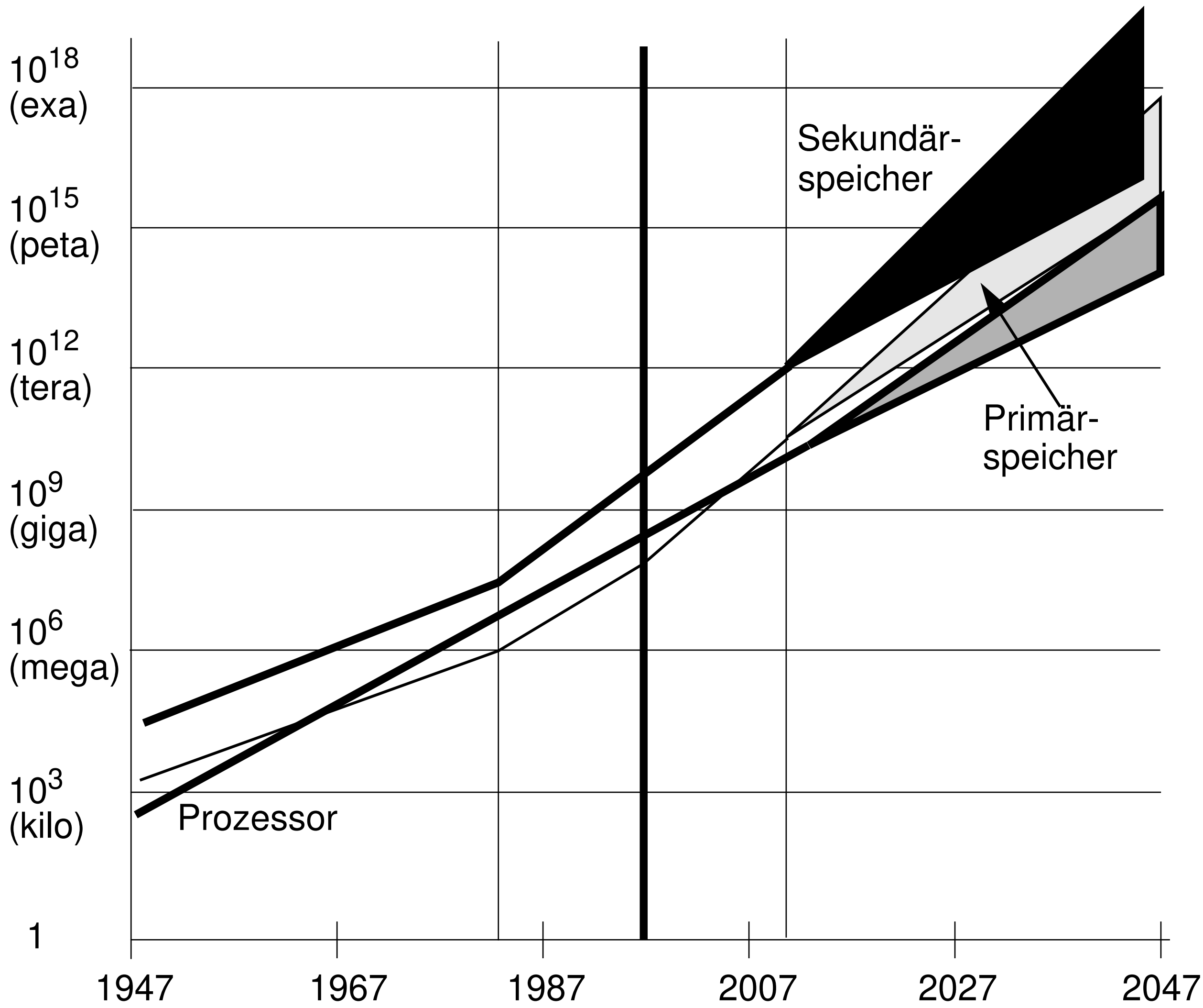
- **Unterscheidung: verteilte und parallele DBS**
- **Anforderungen: Hohe Leistung**
  - Beispiele
  - TPC-C-Benchmark
- **Anforderungen: Hohe Verfügbarkeit**
  - Analyse der Ausfallursachen
  - Fehlertoleranz
- **Anforderungen: Erweiterbarkeit**
- **Arten der Parallelität**
  - Inter-/Intra-Transaktionsparalleität
  - Inter-/Intra-Query-Parallelität
  - Inter-/Intra-Operator-Parallelität
  - Daten- vs. Pipeline-Parallelität
  - E/A-Parallelität
- **Leistungsmaße für Parallelverarbeitung**
  - Speedup und Scaleup
  - Grenzen der Skalierbarkeit

Was sind die technologischen Triebkräfte?

■ Moore's Gesetz

Gordon Moore (Mitgründer von Intel) sagte 1965 voraus, daß die Transistordichte von Halbleiter-Chips sich grob alle 18 Monate verdoppeln würde:

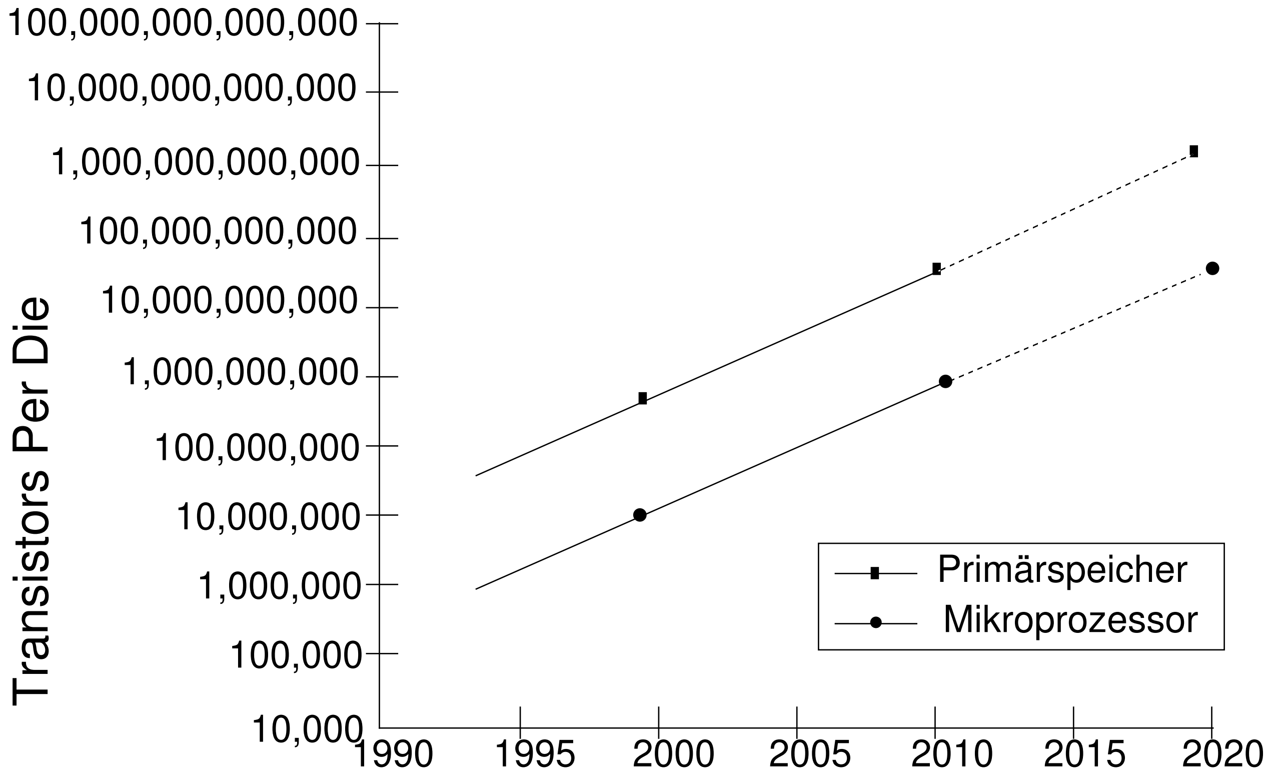
$$\text{CircuitsPerChip (year)} = 2^{(\text{year}-1975)/1.5} * K$$



Evolution der Verarbeitungsgeschwindigkeit von Rechnern in Instruktionen pro Sekunde und Primär-/Sekundärspeichergröße in Bytes von 1947 bis zur Gegenwart, mit einer „überraschungsfreien" Projektion bis 2047. Jede Teilung repräsentiert 3 Größenordnungen und passiert grob in 15-Jahres-Schritten.

Was erwartet uns in nächster Zukunft?

- **Voraussage beim Mikroprozessor: Wachstum bis 2020**
- **Technologie:**
  - heute: 0.1 - 0.2 micron
  - 2020: 0.04 micron
- **„Lineare“ Extrapolation**



- **Prozessor**
  - 2000: 48 M, 0.20-0.25 micron
  - 2010: 1 B, 0.08 micron, 36 mm^2
  - 2020: 20 B, 0.04 micron, 60 mm^2
- **Primärspeicher**
  - 2001: 0.25 Gbit (32 Mbyte)
  - 2010: 64 Gbit (8 GByte)
  - <2020: 1 Tbit (128 Gbyte)

Technologievorhersage

- **Alle HW-Bausteine eines Rechners sind Waren (Massenprodukte)**

- **HW-Komponenten***
  - Technologievorhersage

Jahr	1-Chip- CPU: Mips	1-Chip- DRAM: Mb	Platte 1 GB	Band GB	LAN Mbps	WAN Mbps
1990	10	4	8"	0,3	10 (Ethernet)	0.064 (ISDN)
1995	100	16+	3"	10	150 (ATM)	1
2000	1000	64+	1"	100	1000 (Gigabit-Ethernet)	155 (ATM)
2005	10 (GHz)	1024+	0.5"	400+	10000 (10 Giga-bit-Ethernet	2400 (ATM)

  - Kostenvorhersage für das Jahr 2005

	CPU	DRAM (1024 MB)	Platte 100 GB	Band- roboter	LAN	WAN
Kosten pro Einheit	250$	25$	100$	1000$	50$	200$
für 1000 $	4 CPUs	40 GB	10 x100 G (Array)	2 TB Robot	20 x LAN	5 x WAN

- **Daraus resultierten 4G-Maschinen im Jahr 2000**
  (Commodity PC zu 1000 $)
  - 1 **G**ips CPU-Geschwindigkeit, 0.1 **GB** Hauptspeichergröße
  - 1 **G**bps Netzbandbreite, 10 **GB** Platte,
  - 5G-Maschinen haben noch einen Display von 1 **G**-Pixel
    (3000 x 3000 x 24)

  ➥ **Analogie**: 5M-Maschinen in 1985!

- **Aussehen: Smoking-Hairy Golf Balls**

---

* Gray, J. : Super-Servers: Commodity Computer Clusters Pose a Software Challenge, in: Tagungsband BTW'95, Informatik aktuell, Springer, 1995, S. 30-47.(in 2002 hochgerechnet)

Technologievorhersage (2)

■ **Dramatische Entwicklungen bei Externspeichern und Netzen**

■ **Disk-Farms**

- Zusammenbau aus preiswerten 1''-Platten
- (10 x 10)-Matrix erlaubt die Speicherung von 1 TB
- sehr hohe Leistung und sehr hohe Zuverlässigkeit

➥ Parallelisierung des mengenorientierten Zugriffs

■ **Tape-Farms**

- Zusammenbau aus preiswerten Bandrobotern (für jeweils 100 Bänder)
- Mit 400 GB Daten pro Band lassen sich so 40 TB Daten **„nearline"** speichern
- Mehrfache Transportwege erweitern Bandbreite bei parallelen Transfers

➥ automatische Organisation und Suche:

Speicherung und Bereitstellung der Daten, Archivierung und Wiederauffinden über lange Zeiträume

■ **Netzwerke**

- Sie werden viel schneller. Glasfaser-basierte Kommunikation erlaubt Gb-Datenraten (bleibt aber teuer), im LAN ist 1 Gbps sogar auf Kupferbasis möglich
- Gb-WANs existieren bereits für strategische Strecken zwischen Ballungszentren, aus Kostengründen sind Mb-WANs noch die Regel
- aber: Netzwerkleistung und Kosten ändern sich dramatisch

➥ Netzwerke sind Schlüsselkomponenten für Super-Server.

Sie erlauben schnellen und unmittelbaren weltweiten Zugriff auf Daten und Bilder

Technologievorhersage (3)

- **Server (Mainframes) sind 4T-Maschinen**

~ 1 000 Prozessoren	~ 1 Tops
~ 100,000 DRAMs (@256 Mb+)	= 2.0 TB
~ 10,000 Platten (@10 GB)	= 100 TB
~ 10,000 Netzschnittstellen (@1Gbps)	= 10 Tb

➥ **Aufgabengebiete**: Datenhaltung, Kommunikation, Multimedia-Dienste

- **Welche Architekturform für 4T-Maschinen?**
  - Cluster: Clients benötigen beim Zugriff keine Information, wo sich die Server und die Daten befinden
  - Cluster sind skalierbar und fehlertolerant
  - Sie sind aus „Commodity"-Komponenten aufgebaut
  - Sie halten die "Performance-Rekorde" für alle TPC-*-Benchmarks, weil sie so gut skalierbar sind

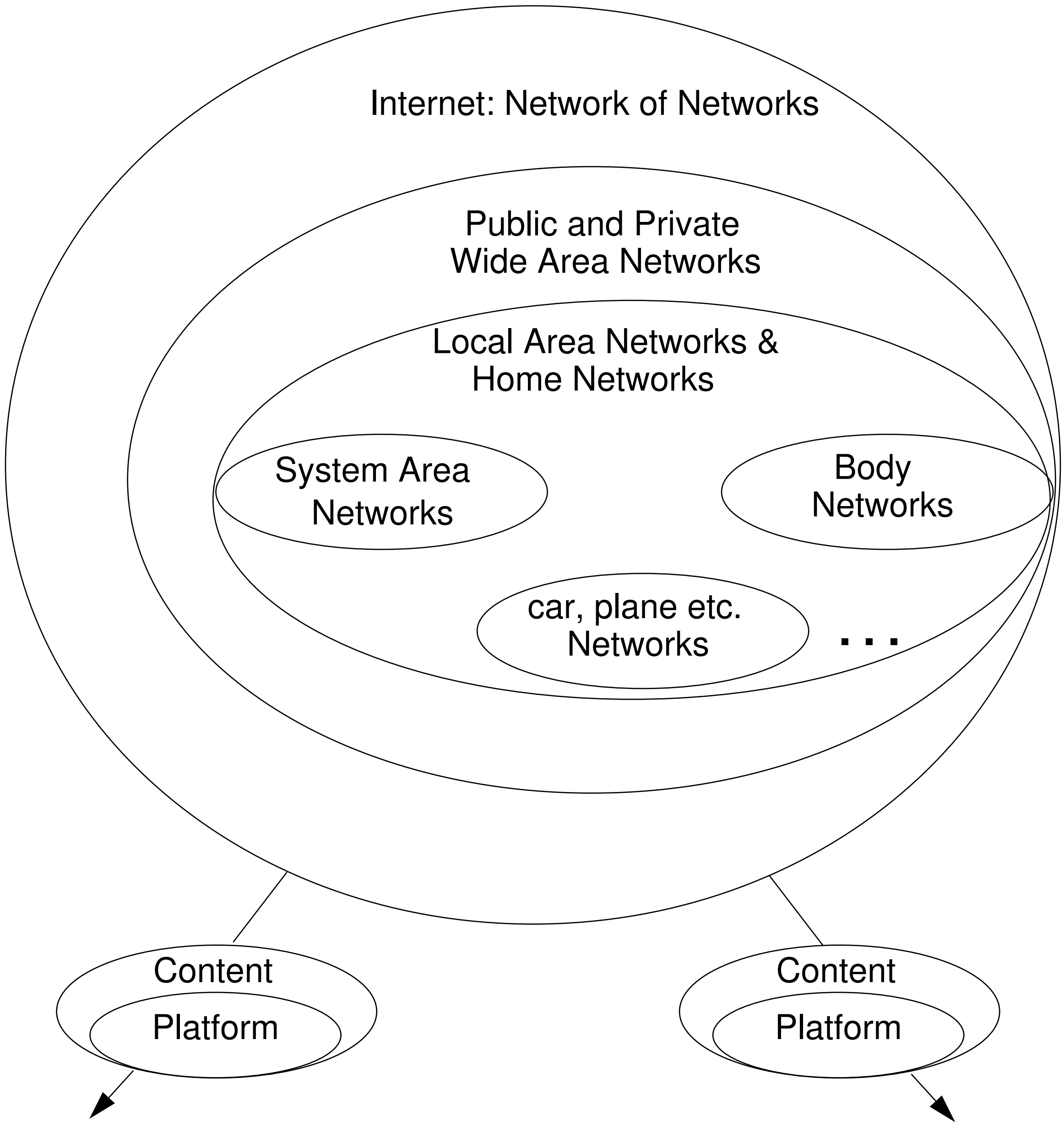
- **Leistungszahlen von Google (2002)**
  - Suchmaschine besteht aus Cluster von > 10.000 PCs
  - Suchvorgänge
    - durchschnittlich 150 Mio/Tag
    - Spitzenlast > 2000/sec
  - Index über
    - > 2 Mrd Dokumente, > 300 Mio Bilder
    - > 700 Mio Usenet-Nachrichten

➥ **Die große Herausforderung sind die SW-Strukturen:**

Wie programmiert man 4T-Maschinen ?
Wie erzielt man Mengenorientierung und Parallelität ?

Was erwartet uns in nächster Zukunft? (2)

■ Allgegenwärtige Rechner in einer Hierarchie von Netzwerken

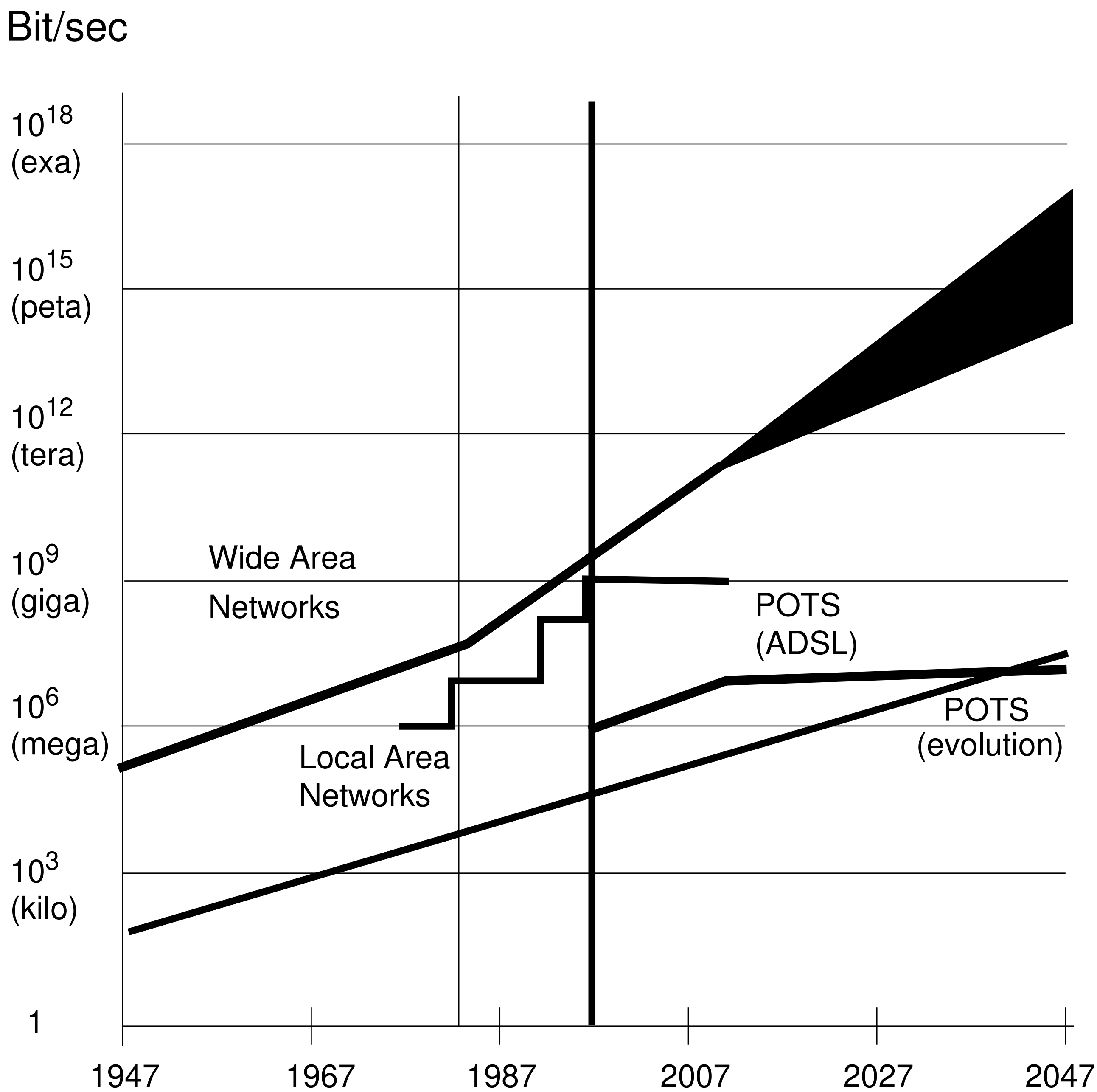


Digitale Schnittstellen bei Menschen und dem Rest der physischen Welt

Der Cyberspace besteht aus einer Hierarchie von Netzwerken, die Rechnerplattformen verbindet. Diese verarbeiten, speichern und machen Schnittstellen verfügbar zur Cyberspace-Benutzerumgebung in der physischen Welt

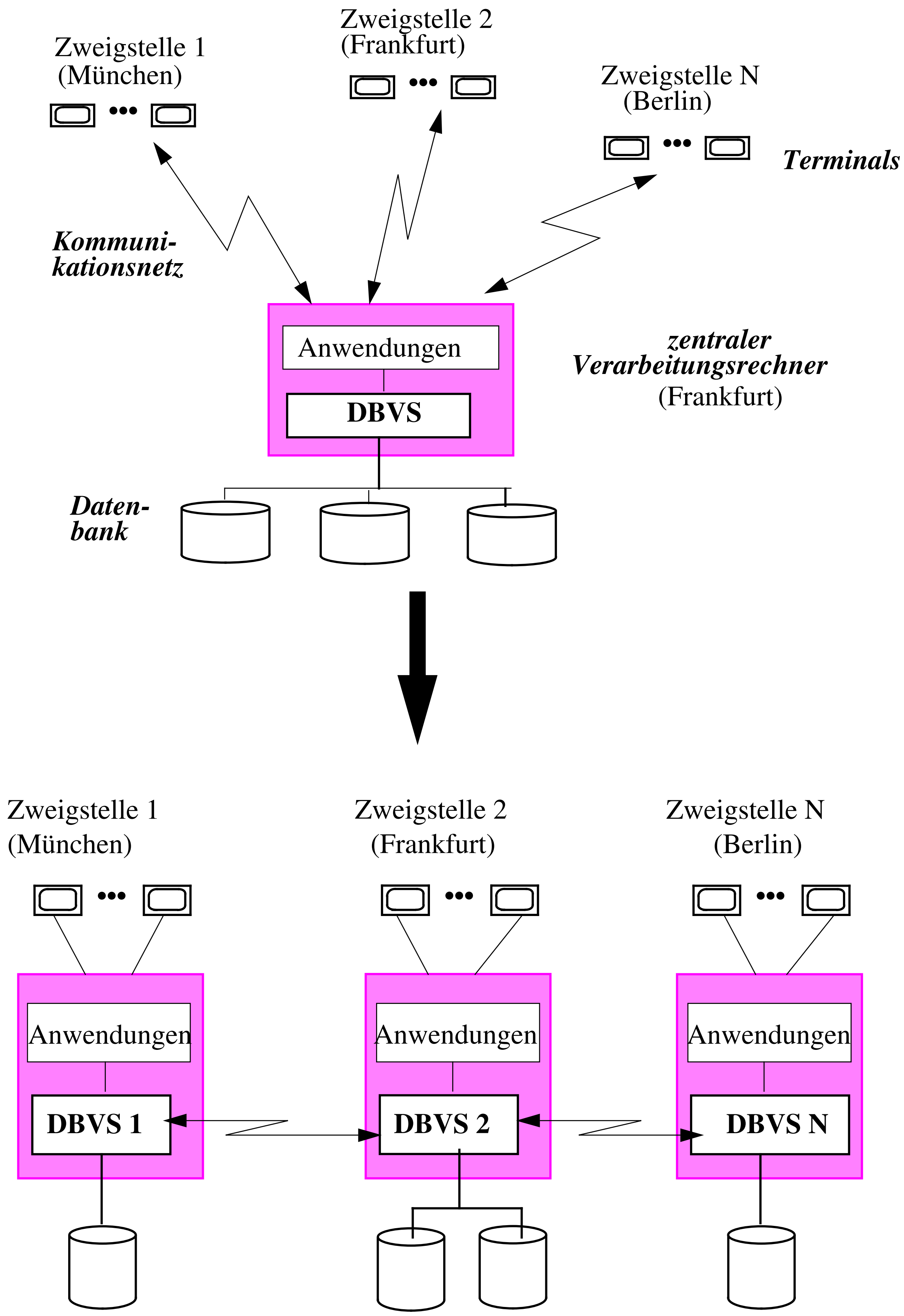
Was erwartet uns in nächster Zukunft? (3)

- **The revolution yet to happen in the network area**



Netze mit fester Infrastruktur:
Evolution von WAN-, LAN- und „plain old telephone service" (POTS)- Bandbreiten in Bits pro Sekunde von 1947 bis zur Gegenwart, und eine Projektion bis 2047.
Infrastrukturlose Netze:
Sogenannte Ad-hoc-Netze erlauben die spontane Vernetzung von mobilen Geräten (Teilnehmern) unterstützen kontextbezogene Anwendungen.
Internet der 4. Generation:
Es umgibt jeden von uns mit einer Vielzahl von Rechnern (ambient internet, μ-chips: RFID-Chips (Radio Frequency Identification)).

Von zentralisierten zu verteilten DBS

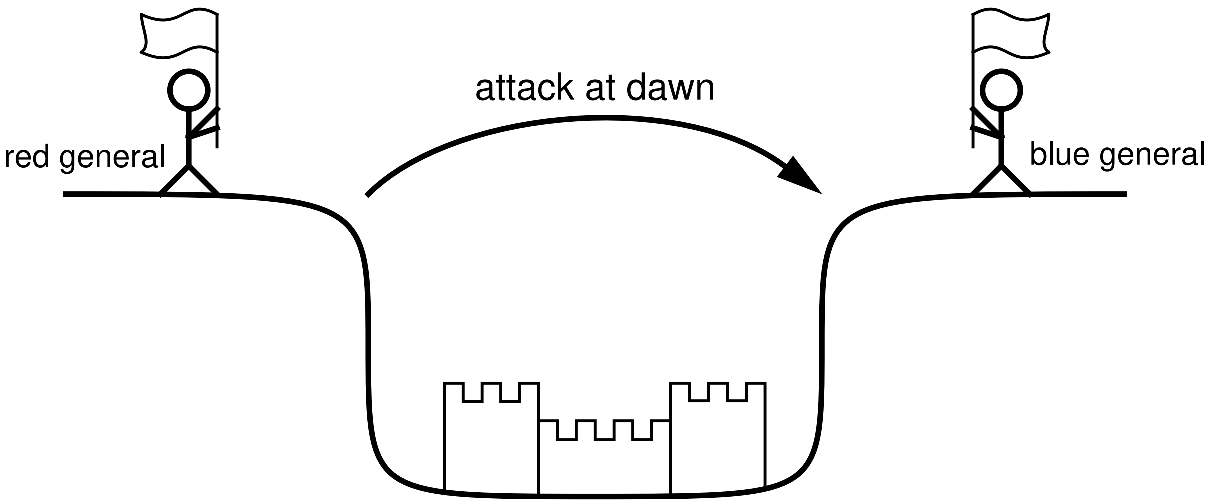


Verteilung und Parallelität

■ Verteiltes System

Es besteht aus autonomen Subsystemen, die oft (weit) entfernt voneinander angeordnet sind, aber koordiniert zusammenarbeiten, um eine gemeinsame Aufgabe zu erfüllen

■ Analogie: The „Coordinated Attack“ Problem



Generals-Paradoxon

➥ Das für verteilte Systeme charakteristische Kernproblem ist der **Mangel an globalem (zentralisiertem) Wissen**

■ Paralleles System

Es besteht aus einer Vielzahl gleichartiger Subsysteme (Komponenten), die lokal zueinander angeordnet sind und nur einen geringen Grad an Autonomie aufweisen.
Charakteristisch ist eine **enge und hochgradig parallele Bearbeitung eines Benutzerauftrags** (Intra-Transaktionsparallelität)

■ Unterscheide

- Parallele Verarbeitung ---> ~ 10^4 - 10^5 Rechner
- Verteilte Verarbeitung ---> Unsicherheit bei globalen Entscheidungen

Verteilung und Parallelität (2)

- **Typische Leistungsmerkmale bei sequentieller Verarbeitung:**
  - Platte 5 MB/s
  - Suchen (Relationen-Scan) 1 MB/s
  - Sortieren 0.1 MB/s
  - Verbund (Join) ?

  ➥ Bearbeitungszeit für eine 1 TB Datenbank ?

- **Intra-Transaktionsparallelität:**
  Einsatz von Parallelität innerhalb von Transaktionen
  - Operationen auf großen Relationen: Scan, Join-Berechnung, Sortierung, Indexgenerierung usw.
  - Volltextsuche in Literaturdatenbanken
  - Multimedia-Anwendungen
  - komplexe Logikprogramme, • • •

  ➥ **kurze Antwortzeiten für daten- und/oder berechnungsintensive DB-Anfragen**

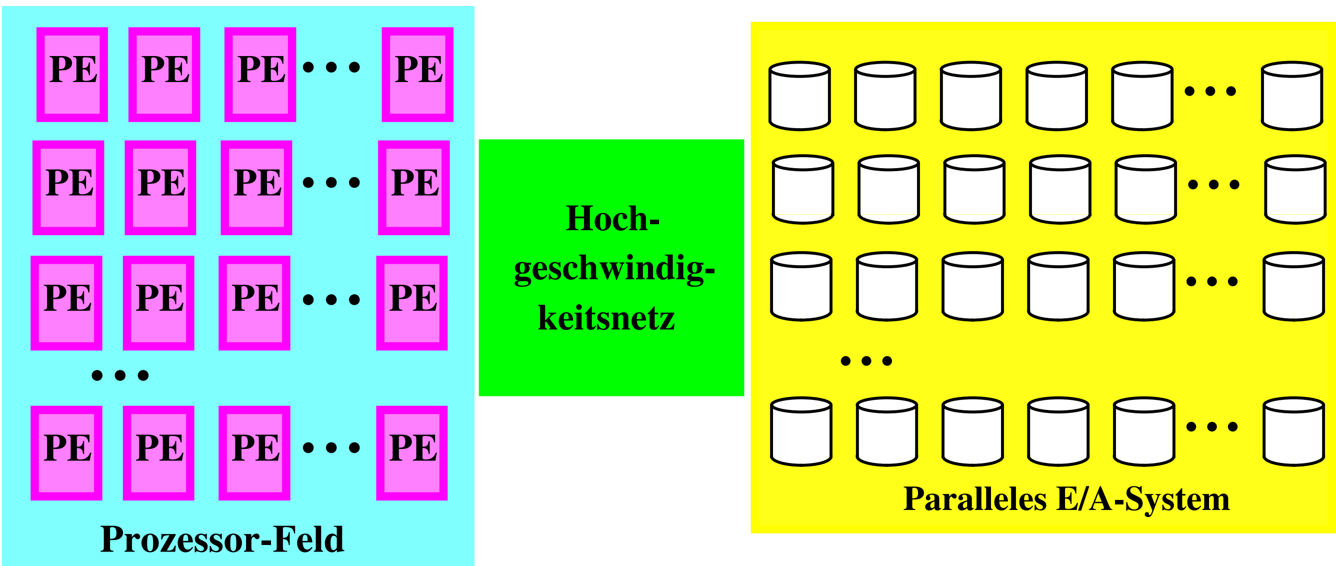
- **Inter-Transaktionsparallelität:**
  - hohe Transaktionsraten für OLTP
  - lineares Durchsatzwachstum

Parallele Datenbanksysteme

- **Voraussetzung für parallele Verarbeitung:**

  Zerlegen eines großen Problems in kleinere Teilaufgaben, die parallel gelöst werden

- **Architektur**
  - Parallelrechner mit hoher Anzahl von Mikroprozessoren
  - lokale Verteilung (Cluster)
  - skalierbares Hochgeschwindigkeitsnetzwerk
  - E/A-Parallelität



- **Spezieller Typ von Mehrrechner-DBS mit den Hauptzielen:**
  - hohe Leistung
  - Verfügbarkeit
  - Skalierbarkeit und
  - Kosteneffektivität

Anforderungen an Mehrrechner-Datenbanksysteme

■ **Ziel:**

Einsatz mehrerer Rechner/DBVS zur koordinierten Verarbeitung von Datenbankoperationen

■ **Anforderungen:**

- Hohe Leistung
  (hohe Transaktionsraten bei kurzen Antwortzeiten)

- Hohe Verfügbarkeit / Fehlertransparenz

- Modulare Erweiterungsfähigkeit
  (vertikales und horizontales Wachstum)

- Verteiltransparenz für DB-Benutzer
  (für Anwendungsprogramme bzw. Endbenutzer)

- Koordinierter Zugriff auf heterogene Datenbanken

- Unterstützung geographisch verteilter Datenbanken
  (Wahrung einer hohen Knotenautonomie)

- Hohe Kosteneffektivität
  (Nutzung leistungsfähiger Mikroprozessoren, Workstations usw.)

- Einfache Handhabbarkeit / Administrierbarkeit

Anforderungen: Hohe Leistung

- **Hohe Transaktionsraten (Durchsatz)**

  >> 1000 TPS (vom Typ 'Kontenbuchung')

- **Kurze Antwortzeiten**
  - Akzeptanz für Dialogbetrieb
  - trotz höheren Durchsatzes / Kommunikationsverzögerungen
  - Parallelisierung komplexer Anfragen

- **Ständig steigende Leistungsanforderungen:**
  - wachsende Zahl von Benutzern/Terminals
  - Einführung neuer Anwendungen / Transaktionstypen
  - ständiges Wachstum der Datenbanken
  - Bearbeitung komplexerer Vorgänge und Integritätsbedingungen
  - Benutzung höherer Programmiersprachen
  - komfortablere Benutzerschnittstellen
  - zunehmend auch Mehrschrittdialoge

➥ Einsatz von Mehrrechner-DBS erforderlich

Beispiele für hohe Leistungs-
anforderungen

1. Bankanwendungen/ Reservierungssysteme

   Kontenbuchungen oder Platzreservierungen sollen mit einem Durchsatz

   - von mehreren 1000 TPS und
   - einer Antwortzeit < 2 sec

   bearbeitet werden.

2. Telefonvermittlung

   Pro Telefongespräch ist ein Benutzerprofil aus der DB zu lesen sowie ein Abrechnungssatz zu schreiben.

   In Zeiten hohen Verkehrsaufkommens ist

   - mit > 15.000 solcher Transaktionen pro Sekunde zu rechnen
   - die Antwortzeit sollte < 0.2 sec sein.

3. Entscheidungsunterstützung/ Data Warehousing

   Auf einer 5 TB großen DB sollen komplexe Ad-hoc-Anfragen ablaufen, die im worst-case ein vollständiges Durchlesen der DB erfordern.

   Das DBVS soll

   - einen Durchsatz von 5 TPS und
   - eine Antwortzeit < 30 sec erreichen.

4. E-Commerce / Digitale Bibliotheken / Geo-Informationssysteme, ...

TPC-Benchmarks

- **Herstellergremium zur Standardisierung von DB-Benchmarks**

  Gründung 1988

- **Erste Benchmarks basierend auf Kontenbuchung (“Debit-Credit”): TPC-A (1989) und TPC-B (1990)**

- **Besondere Merkmale**
  - Leistung eines Gesamt-Systems wird bewertet
  - Bewertung der Kosteneffektivität (Kosten / Leistung)
  - skalierbare Konfigurationen
  - verbindliche Richtlinien zur Durchführung und Dokumentation (Auditing; Full Disclosure Reports)
  - Ausschluß von “Benchmark Specials” innerhalb von DBVS usw.

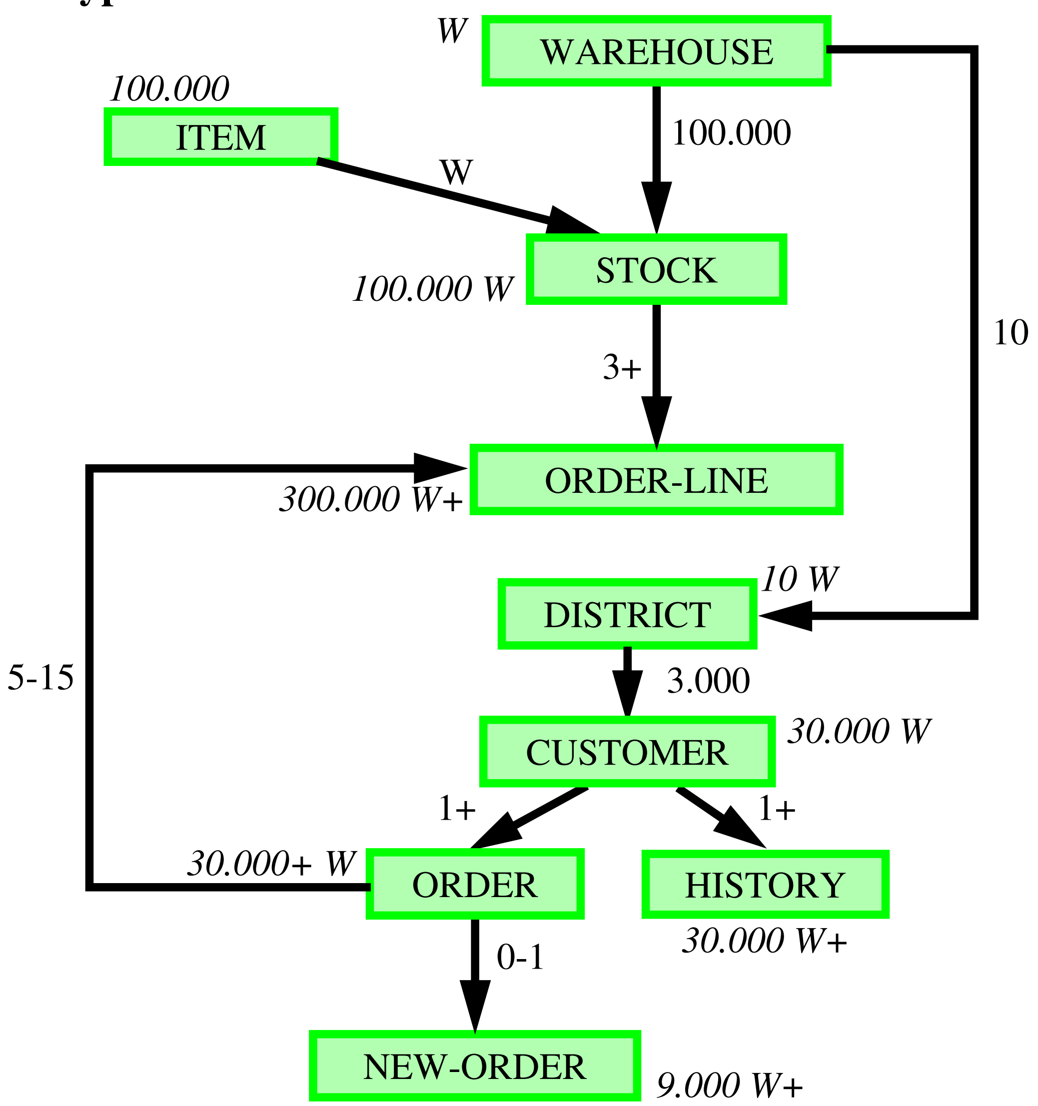
- **Aktuelle Benchmarks für**
  - OLTP (TPC-C)
  - Decision Support (TPC-H/R)
  - Web-Transaktionen (TPC-W)

TPC-C-Benchmark

- **Verabschiedung: August 1992**

- **Anwendung: Bestellverwaltung im Großhandel (order entry)**
  - Betrieb umfaßt W Warenhäuser, pro Warenhaus 10 Distrikte, pro Distrikt 3000 Kunden
  - 100.000 Artikel; pro Warenhaus wird Anzahl vorhandener Artikel geführt
  - 1% aller Bestellungen müssen von nicht-lokalem Warenhaus angefordert werden

- **9 Satztypen**



über 500.000 Sätze (50 MB) pro Warenhaus

TPC-C (2)

■ Haupttransaktionstyp NEW-ORDER

BEGIN WORK { Beginn der Transaktion }

SELECT ... FROM **CUSTOMER**
WHERE c_w_id = *:w_no* AND c_d_id = *:d_no* AND c_id = *:cust_no*

SELECT ... FROM **WAREHOUSE** WHERE w_id = *:w_no*

SELECT ... FROM **DISTRICT** (* -> *next_o_id* *)
WHERE d_w_id = *:w_no* AND d_id = *:d_no*

UPDATE DISTRICT SET d_next_o_id := *:next_o_id* + 1
WHERE d_w_id = *:w_no* AND d_id = *:d_no*

INSERT INTO NEW_ORDER ...

INSERT INTO ORDERS ...

pro Artikel (im Mittel 10) werden folgende Anweisungen ausgeführt:

SELECT ... FROM **ITEM** WHERE ...

SELECT ... FROM **STOCK** WHERE ...

UPDATE STOCK ...

INSERT INTO ORDER-LINE ...

COMMIT WORK { Ende der Transaktion }

- im Mittel 48 SQL-Anweisungen
  (BOT, 23 SELECT, 11 UPDATE,
  12 INSERT, EOT)
- 1% der Transaktionen sollen zurückgesetzt werden

TPC-C (3)

- **5 Transaktionstypen:**
  - *New-Order:* Artikelbestellung (Read-Write)
  - *Payment:* Bezahlung einer Bestellung (Read-Write)
  - *Order-Status*: Status der letzten Bestellung eines Kunden ausgeben (Read-Only)
  - *Delivery:* Verarbeitung von 10 Bestellungen (Read-Write)
  - *Stock-Level:* Anzahl von verkauften Artikeln bestimmen, deren Bestand unter bestimmtem Grenzwert liegt (Read-Only)
- **Durchsatzangabe für New-Order-Transaktionen in tpm-C**
  (Transaktionen pro Minute)
- **Festlegung des Transaktions-Mixes**
  - New-Order-Anteil variabel, jedoch höchstens 45 %
  - Payment-Transaktionen müssen mindestens 43 % der Last ausmachen
  - Order-Status, Delivery und Stock-Level je mindestens 4 %
- **Pro Transaktionstyp festgelegte Antwortzeitrestriktion**
  - 90% unter 5s bzw. 20 s für Stock-Level
  - mittlere Denkzeiten und Eingabezeiten
- **Kosteneffektivität ($/tpm-C)**
  unter Berücksichtigung aller Systemkosten für 5 Jahre (ab V5: 3 Jahre)

Entwicklung der TPC-C-Ergebnisse

■ **Entwicklung der Ergebnisse***

TPC-C by Performance

	tpmC	$/tpmC	Bemerkung
Sep. 1992	54	188 562	
Jan. 1993	269	3000	
Nov. 1995	11 456	286	Oracle 8xAlpha 350
Sep. 1997	39 469	95	Sybase 16xHP PA200
Sep. 1999	135 461	97	Oracle 4x24 Sparc 400
Mai 2001	688 220	22.5	MS SQL-Server (IBM)
Sep. 2001	709 220	15	MS SQL-Server (Compaq)

TPC-C by Price/Performance

	tpmC	$/tpmC	Bemerkung
Mai 2001	15 533	4.67	MS SQL-Server (IBM)
Dez. 2001	11 314	4.38	MS SQL-Server (Dell)
März 2002	17 078	3.99	MS SQL-Server (Compaq)
März 2002	11 537	3.68	MS SQL-Server (Dell)

■ **Jim Gray, 1999:**
in 2002 wird erreicht: 1MtpmC @ 10 $/tpmC => 0.01 $/tpd

■ **Welche Transaktionslasten müssen verarbeitet werden ???**

- 6 Milliarden Menschen, 12 Stunden/Tag, 10 s Denkzeit
- jedoch eher Faktor 100 weniger
  (tatsächliche Benutzer/Arbeitszeiten/Denkzeiten)

■ **Welche EDV-Kosten sind damit verbunden (0.01 $/tpd) ?**

* www.tpc.org

Anforderungen: Hohe Verfügbarkeit

■ **Ziel: Continuous Operation**

- zumindest Tolerierung von Einfachfehlern
- System darf nicht (vollständig) ausfallen, da Wiederanlauf zu aufwendig (Netzwerk)
- dynamische Reorganisation, Erweiterbarkeit usw. aller Datenstrukturen
- Installation neuer SW-Versionen usw. im laufenden Betrieb

■ **Voraussetzungen:**

- redundante Systemkomponenten (Fehlertoleranz)
  - HW- und SW-Komponenten
  - Daten (Log, Spiegelplatten, replizierte DB)
- automatische Fehlererkennung und -behandlung
- Umkonfigurierbarkeit im laufenden Betrieb

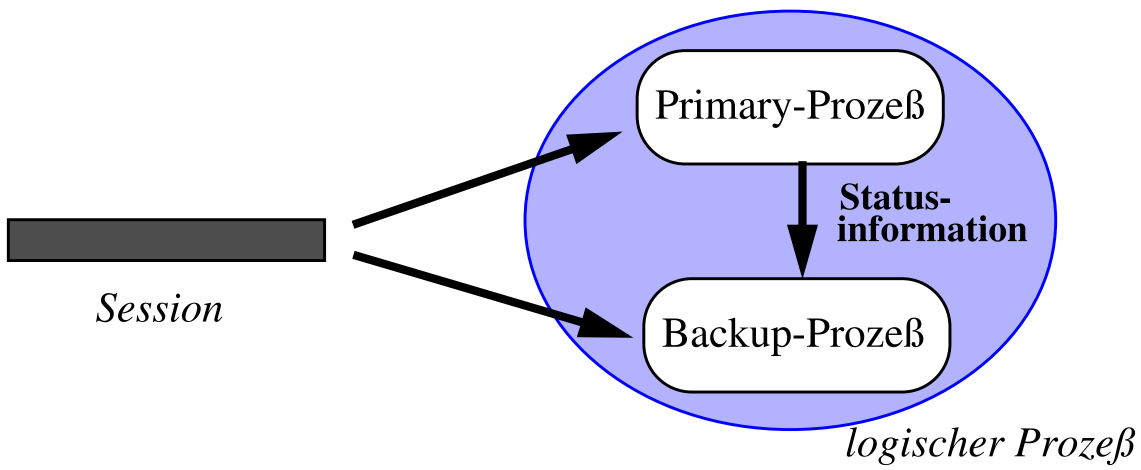
■ **MTBF (meantime between failures):**

konventionelle Systeme:	> 10	Tage
fehlertolerante Systeme:	> 10	Jahre

Anforderungen: Hohe Verfügbarkeit (2)

- **Verfügbarkeit =** MTBF / (MTBF + MTTR)
  - MTTR (meantime to repair)
  - Warum darf das Gesamtsystem nicht ausfallen?

- **Hilfreiche Konzepte für Verfügbarkeit**
  - Prozeß-Paare und Transaktionskonzept zur Fehlermaskierung
  - Aktiver Primary- und passiver Backup-Prozeß
  - Aktualisierung des Backups durch periodische Checkpoint-Nachrichten (häufiges Checkpointing, um Forward-Recovery zu ermöglichen)



Analyse von Ausfallursachen

- **Aufgliederung der Ausfallursachen*:**

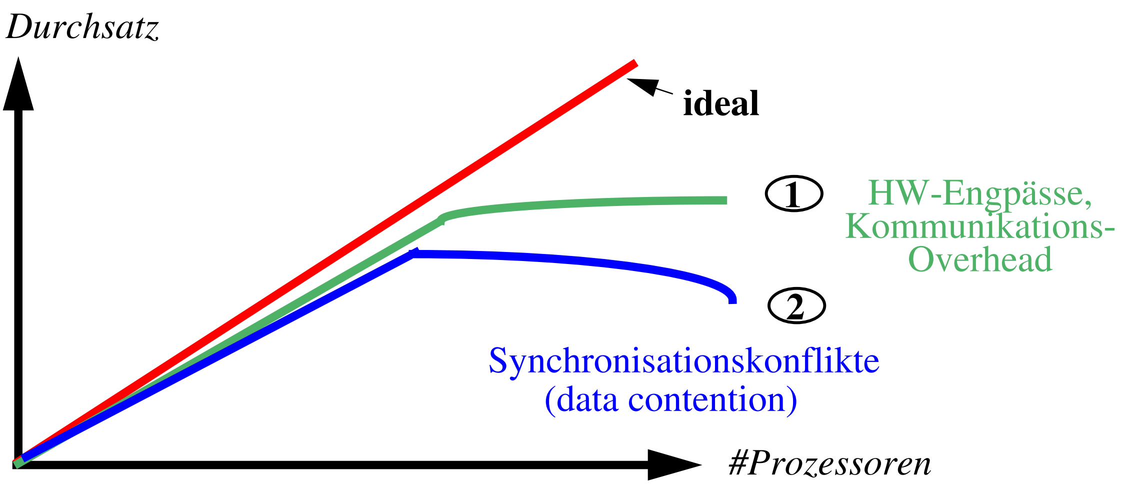
Ursache	1985	1989
Software	33 %	62 %
Hardware	29 %	7 %
Maintenance	19 %	5 %
Operations	9 %	15 %
Environment	6 %	6 %
System MTBF	8 Jahre	21 Jahre

- **"Under-reporting" bei Operating, Anwendungs-SW**
- **Software-Fehler dominierende Ausfallursache**
  - stark wachsender Umfang an Anwendungs- und System-SW
  - gestiegene SW-Komplexität
- **Starke Verbesserungen bei Hardware und Maintenance**
  - zunehmende Integrationsdichte (VLSI), kompaktere Platten, Kommunikation über Glasfaserkabel
  - HW-Redundanz maskiert Mehrzahl aller Ausfälle
  - MTBF für Platten verbesserte sich von 8 K auf >100 K Stunden!
- **Stärkere Automatisierung beim Operating erforderlich**

* *J. Gray: A Census of Tandem System Availability Between 1985 and 1990, in: IEEE Trans. on Reliability 39 (4), 1990, 409-418. Neuere Untersuchungen sind nicht verfügbar. Die prinzipiellen Aussagen sind immer noch gültig.*

Anforderungen: Erweiterbarkeit

- **Ziel: modulares (horizontales) Wachstum**
  - zusätzliche Rechner, Platten usw.
  - lineare Durchsatzsteigerung (bei kurzen Antwortzeiten)
  - komplexe Anfragen: Antwortzeitverkürzung durch Parallelisierung proportional zur Rechneranzahl



- **TPC-Benchmark-Lasten stellen Idealfall hinsichtlich Skalierbarkeit dar**
  - DB wächst proportional mit Rechneranzahl (Durchsatz)
  - ideale Partitionierung von DB und Last möglich (➥ minimaler Kommunikationsaufwand)

- **„Reale" Lasten**
  - begrenzte Partitionierbarkeit (mit #Rechner wachsende Nachrichtenhäufigkeit pro Transaktion)
  - schwierige Lastbalancierung
  - potentiell wachsendes Ausmaß an Sperrkonflikten

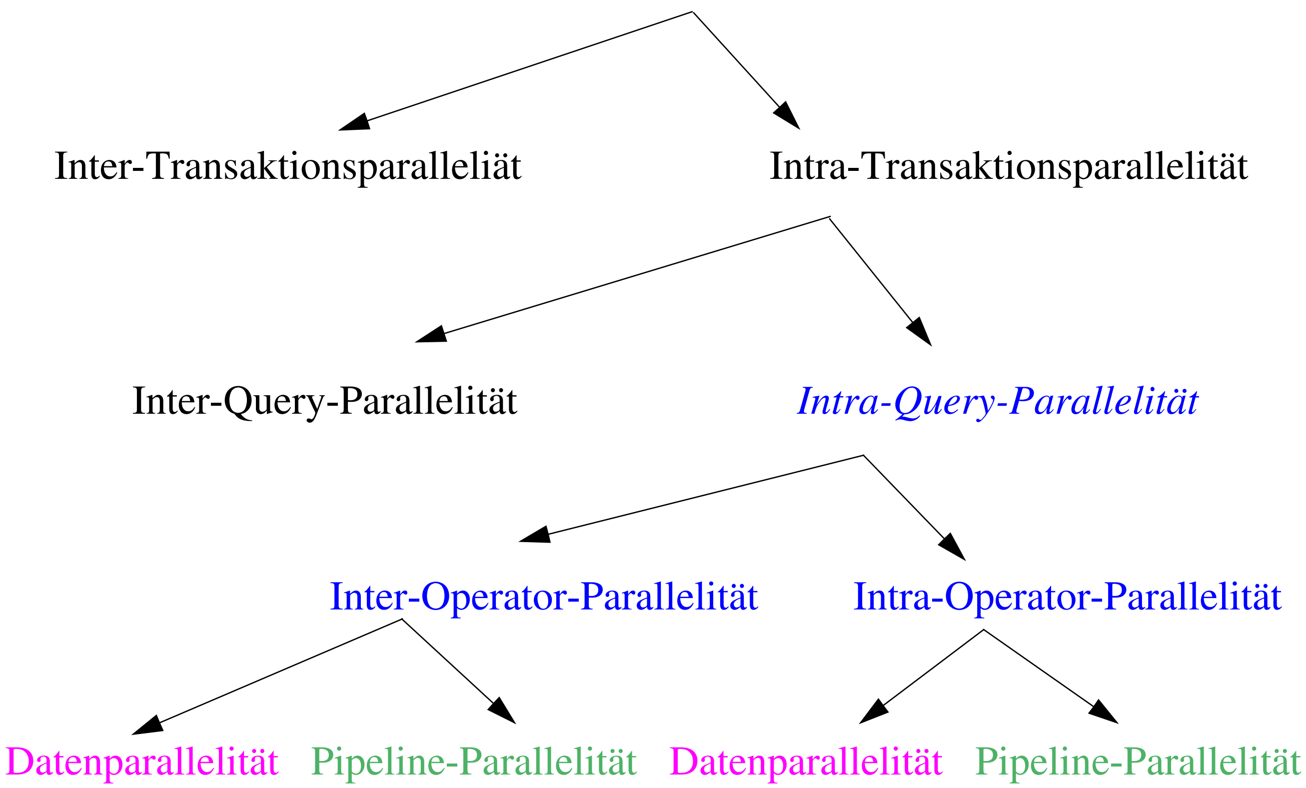
Relative Referenzmatrix (DOA-Last)

ca. 17 500 Transaktionen, 1 Million Seitenreferenzen auf ca. 66 000 verschiedene Seiten

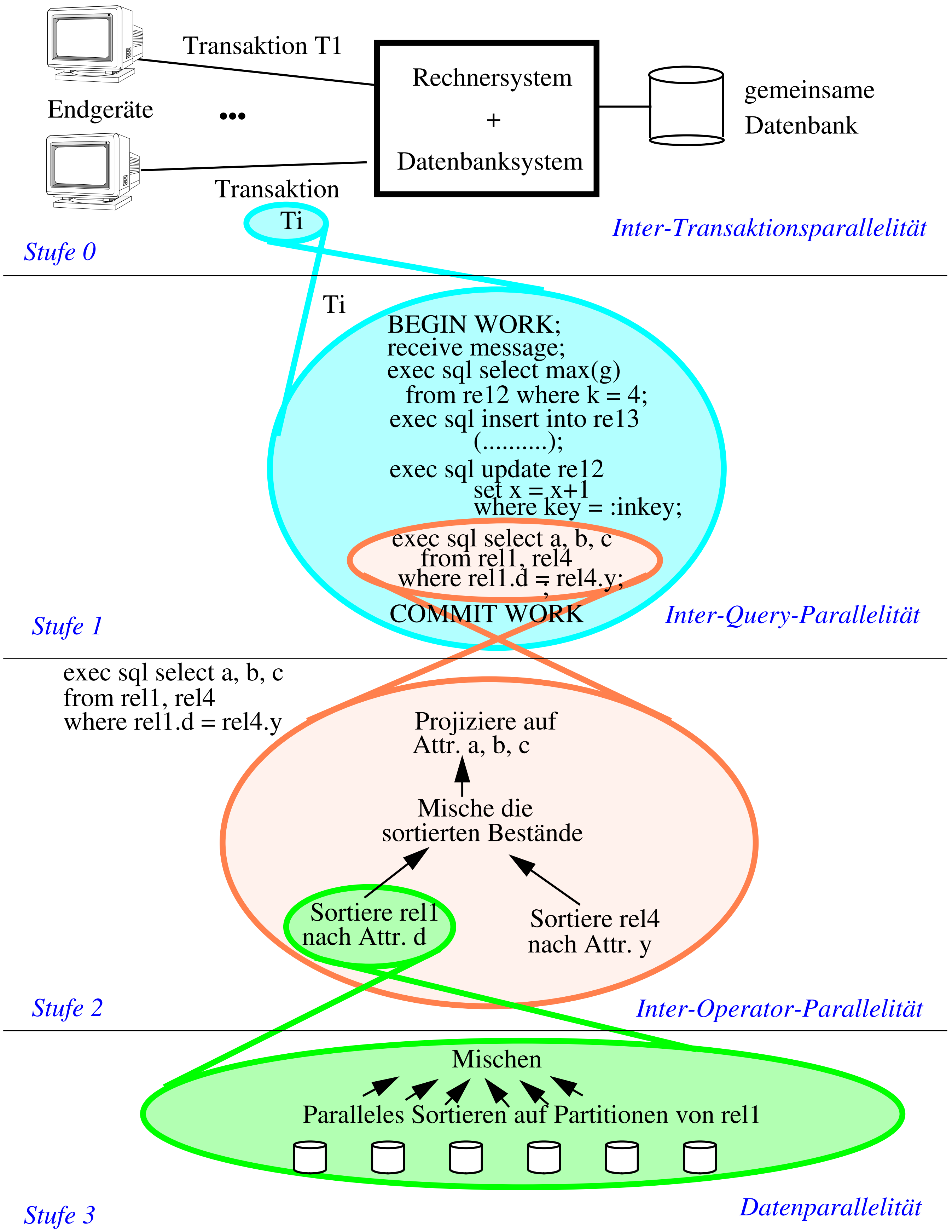
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	Total
TT1	9.1	3.5	3.3		5.0	0.9	0.4	0.1				0.0		22.3
TT2	7.5	6.9	0.4	2.6	0.0	0.5	0.8	1.0	0.3	0.2	0.0			20.3
TT3	6.4	1.3	2.8	0.0	2.6	0.2	0.7	0.1	1.1	0.4		0.0	0.0	15.6
TT4	0.0	3.4	0.3	6.8			0.6	0.4			0.0			11.6
TT5	3.1	4.1	0.4		0.0		0.5	0.0						8.2
TT6	2.4	2.5	0.6		0.7		0.9	0.3						7.4
TT7	1.3		2.6			2.3	0.1							6.2
TT8	0.3	2.3	0.2		0.0		0.1							2.9
TT9	0.0	1.4	0.0					1.1						2.6
TT10	0.3	0.1	0.3			1.0	0.1					0.0		1.8
TT11		0.9						0.2						1.1
TT12		0.1												0.1
Total	30.3	26.6	11.0	9.4	8.3	4.9	4.1	3.3	1.4	0.6	0.0	0.0	0.0	100.0
partition size (%)	31.3	6.3	8.3	17.8	1.0	20.8	2.6	7.3	2.6	1.3	0.8	0.0	0.0	100.0
% referenced	11.1	16.6	8.0	2.5	18.1	1.5	9.5	4.4	5.2	2.7	0.2	13.5	5.0	6.9

Arten der Parallelität

- **Unterscheidungsmerkmale**
  - Granularität der parallelisierten Verarbeitungsschritte (Transaktion, Query, Operator)
  - Datenparallelität vs. Funktionsparallelität (Pipeline-Parallelität)
  - Verarbeitungs- vs. E/A-Parallelität
- **Klassifikation**



Gleichzeitiger Einsatz mehrerer Parallelisierungsarten



Inter-Query-Parallelität

- **Parallele Bearbeitung unabhängiger DB-Operationen (Queries)** eines Transaktionsprogrammes
  - Programmierer muß Parallelisierung spezifizieren

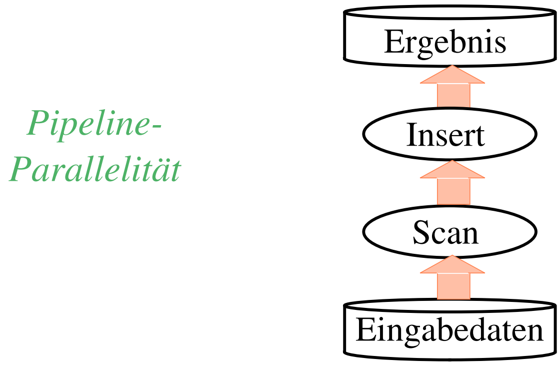
```
BEGIN WORK
SELECT * FROM A ... WHERE ...
UPDATE A ...
SELECT * FROM B WHERE ...
COMMIT WORK
```

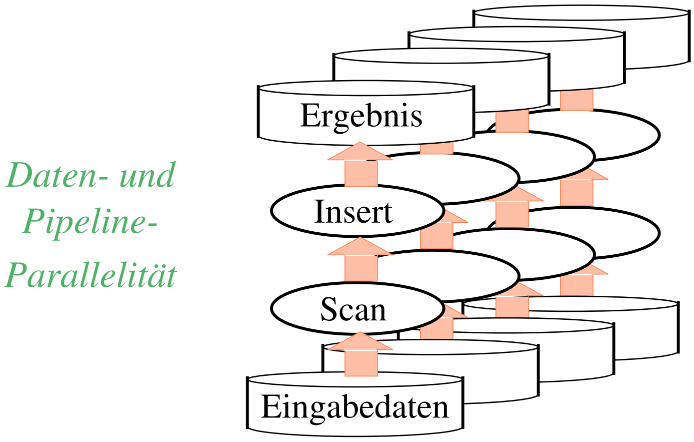
```
BEGIN WORK

PARBEGIN                          PARBEGIN
SELECT * FROM A ...               SELECT * FROM B
    WHERE ...                         WHERE ...
UPDATE A ...                      PAREND
PAREND

COMMIT WORK
```

  - nur begrenzter Parallelitätsgrad möglich

Pipeline- vs. Daten-Parallelität

■ **Pipeline-Parallelität**





- Datenfluß-Prinzip zum Datenaustausch zwischen Operatoren / Teiloperatoren
- frühzeitige Weitergabe von Tupeln bei Zwischenergebnissen
- Einsatz vor allem bei Inter-Operator-Parallelität
- Pipeline-Unterbrechung bei Operatoren, die vollständige Eingabe zur Ergebnisberechnung verlangen: Sortierung, Duplikat-eliminierung, Gruppierung (GROUP BY), Aggregatfunktionen usw.
- Pipelines oft sehr kurz ($\leq$ 10 Operatoren)

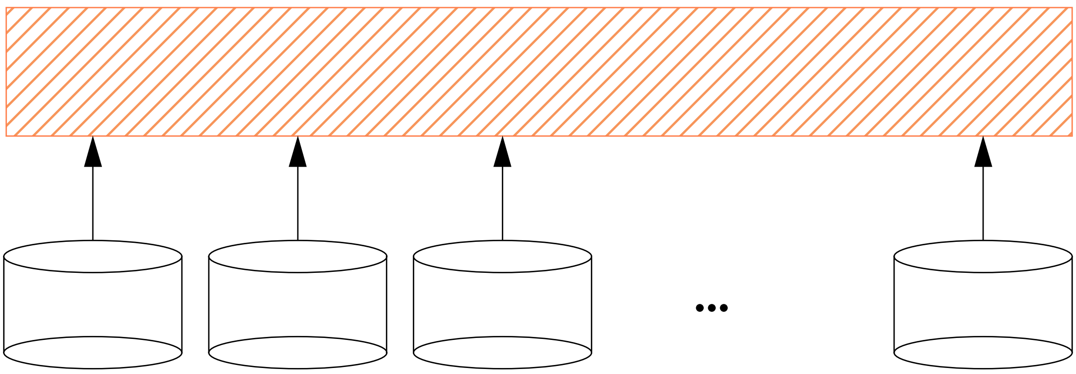
■ **Datenparallelität**

- basiert auf breiter (horizontaler) Datenverteilung (Declustering)
- parallele Bearbeitung von Teiloperationen auf disjunkten Datenmengen
- Parallelitätsgrad kann mit Datenumfang gesteigert werden

E/A-Parallelität

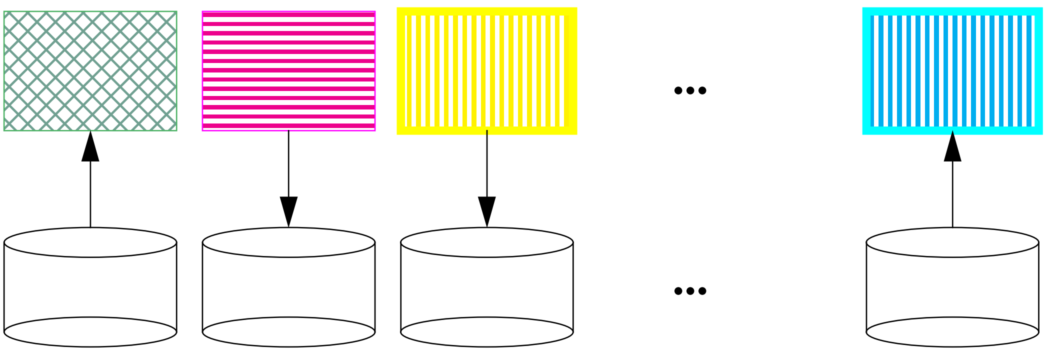
- **Voraussetzung: Declustering von Dateien über mehrere Platten**

- **Intra-E/A-Parallelität (Zugriffsparallelität)**



Parallele Ausführung eines E/A-Auftrags (➥ Datenparallelität)

- **Inter-E/A-Parallelität (Auftragsparallelität)**



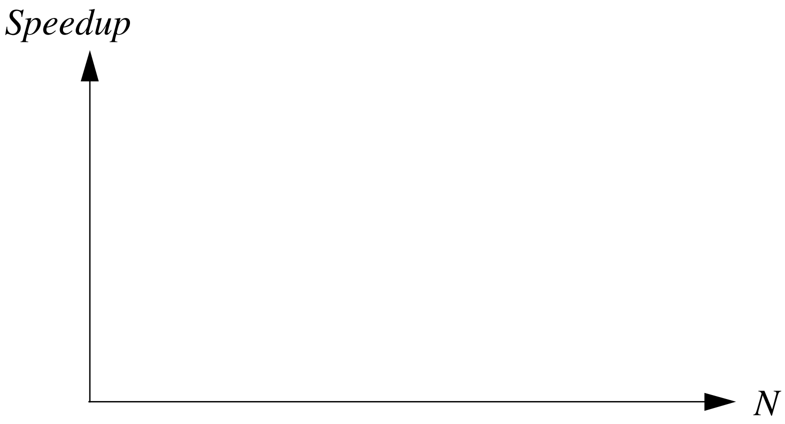
Parallele Ausführung unabhängiger E/A-Aufträge verschiedener Platten
(➥ Inter-Transaktionspararallelität, Inter-Query-Parallelität)

Leistungsmaße für Parallelverarbeitung: Speedup

- **Vorgabe: konstante Datenbankgröße**

- **Antwortzeit-Speedup** (batch speedup) mißt Antwortzeitverbesserung für komplexe Operationen durch Parallelverarbeitung

$$\text{Antwortzeit-Speedup (N)} = \frac{\text{Antwortzeit bei 1 Rechner}}{\text{Antwortzeit bei N Rechnern}}$$



- **Ziel:**
  Lineare Antwortzeitverkürzung mit wachsender Rechneranzahl durch Einsatz von Intra-Transaktionsparallelität

- **Amdahls Gesetz**
  - Speedup ist begrenzt durch nicht-optimierte (sequentielle) Komponenten der Antwortzeit

$$\text{Antwortzeit-Speedup} = \frac{1}{(1 - F_{opt}) + \frac{F_{opt}}{S_{opt}}}$$

F_{opt} = Anteil der optimierten Antwortzeitkomponente ($0 \le F_{opt} \le 1$)
S_{opt} = Speedup für optimierten Antwortzeitanteil

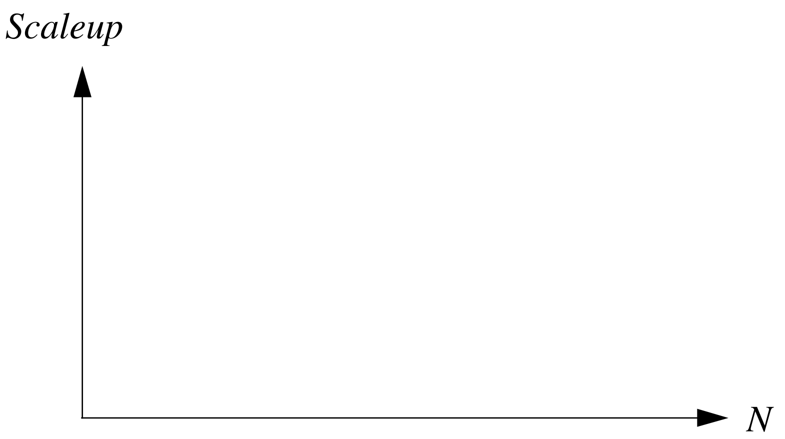
  - Beispiel: 5% sequentieller Anteil ->

Leistungsmaße für Parallelverarbeitung: Scaleup

- **Vorgabe: Datenbankgröße wächst linear mit der Rechneranzahl**

- **Durchsatz-Scaleup** (OLTP scaleup) mißt relatives Durchsatzwachstum (bei gegebener Antwortzeitrestriktion)

$$\text{Durchsatz-Scaleup (N)} = \frac{\text{Transaktionsrate bei N Rechnern}}{\text{Transaktionsrate bei 1 Rechner}}$$



➥ Ziel: lineares Wachstum durch Nutzung von Inter-Transaktionsparallelität

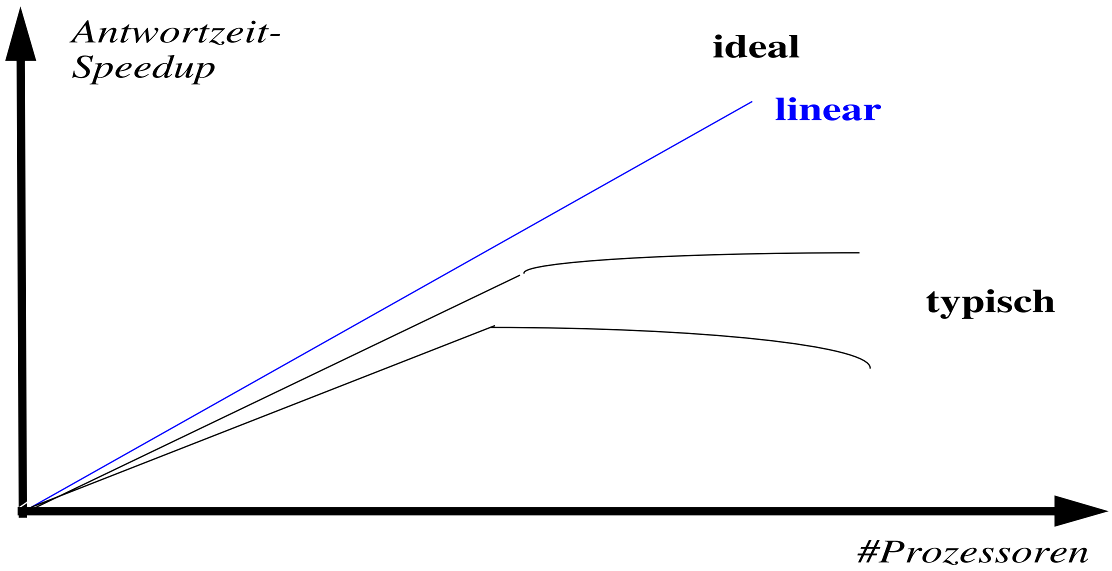
- **Antwortzeit-Scaleup** (batch scaleup) mißt Antwortzeitveränderung für komplexe Operationen auf unterschiedlichen Datenmengen

$$\text{Antwortzeit-Scaleup (N)} = \frac{\text{Antwortzeit bei N Rechnern}}{\text{Antwortzeit bei 1 Rechner}}$$

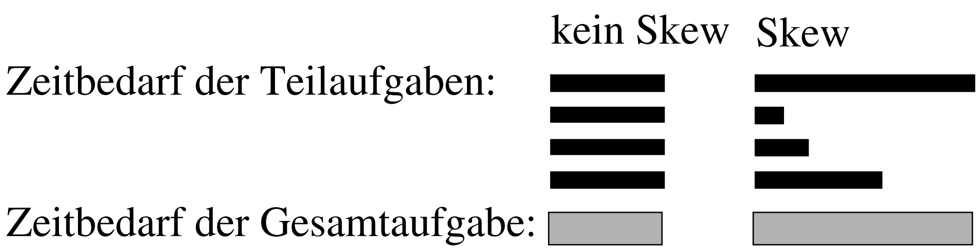
➥ Ziel: gleichbleibende Antwortzeit trotz wachsender DB durch Intra-Transaktionsparallelität

Grenzen der Skalierbarkeit

■ **Charakterisierung der Antwortzeitverbesserung**
Die reale Nutzung von Parallelität ist i. allg. bei weitem nicht "ideal"!



- maximale inhärente Parallelität ist begrenzt
- Startup- und Terminierungs-Overhead
- Interferenzen bei physischen und logischen Ressourcen
- **Varianz (Skew)** in den Ausführungszeiten der Teilaufgaben



➥ **Amdahls Gesetz** begrenzt Antwortzeit-Speedup!

Zusammenfassung

- **Wesentliche Klassen von MRDBS: Verteilte DBS und Parallele DBS**

- **Eigenschaften von MRDBS**
  - Hohe Leistungsfähigkeit (hohe Transaktionsraten, Parallelsierung komplexer Anfragen)
  - Unterstützung sehr großer DB
  - Hohe Verfügbarkeit und Fehlertoleranz in allen Komponenten
  - Modulare Erweiterungsfähigkeit, u. a.

- **Hochleistungs-DB-Server**
  - Verwendung kosteneffektiver, in Massenproduktion hergestellter Standard-Komponenten
  - skalierbare Architektur mit lokaler Verteilung
  - Unterstützung von Standards und Interoperabilität

- **TPC-Benchmarks zur Leistungsmessung von DBS/Transaktionssystemen**

- **Unterstützung unterschiedlicher Arten von Intra-Transaktionsparallelität**

- **Speedup und Scaleup-Metriken zur Parallelverarbeitung**