

Kapitel 8

*Benchmarks*¹

Inhalt

- ❑ Allgemeines
- ❑ OLTP-Benchmarks: TPC-A/B/C,
- ❑ Decision-Support-Benchmarks: TPC-D/H/R
- ❑ Web-Benchmark: TPC-W
- ❑ Zusammenfassung

1. Benchmark: Vergleichspunkt, Bezugswert, Maßstab

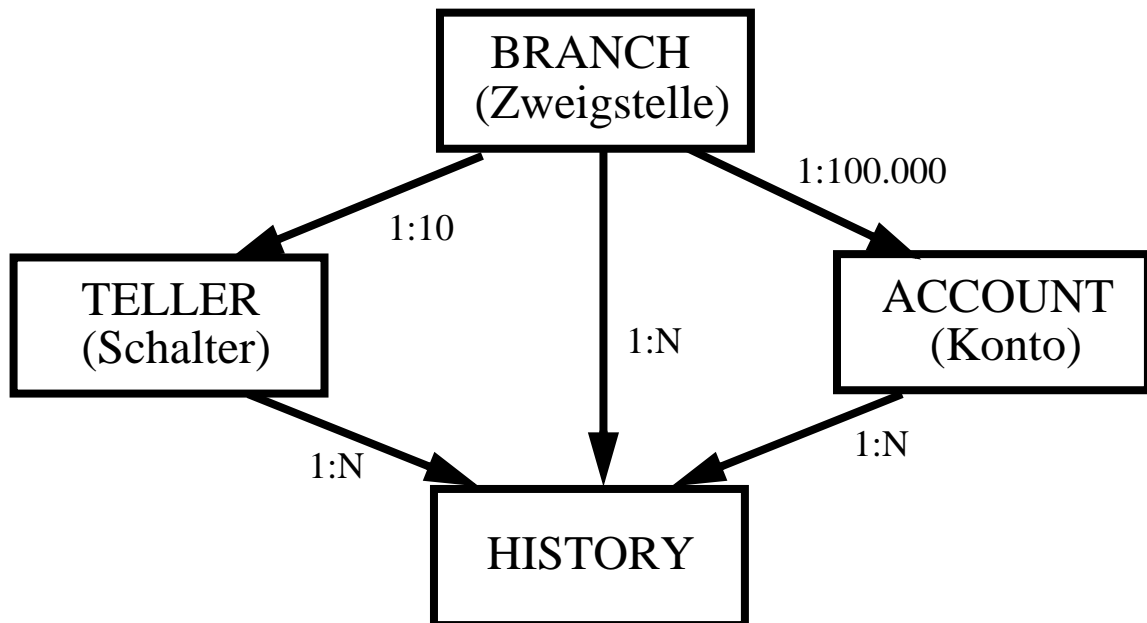
Allgemeines

□ Standardisierung von Arbeitslasten

- ⇒ bei interaktiven TA ist die **Antwortzeit** eine kritische Größe; Stapel-TA können dagegen Stunden oder Tage laufen und enorme Ressourcen verbrauchen (CPU, Speicher)
- ⇒ Leistungsbestimmung oder -abschätzung eines TP-Systems ist sehr schwierig
 - Komplexität des Systems
 - Leistungsverhalten bei verschiedenartigen Lasten kann sehr verschieden sein!
- ⇒ Im Laufe der Zeit wurden unterstützende Maßnahmen getroffen
 - es wurden verschiedene Nutzungsmuster erkannt
 - **Standard-Benchmarks** wurden definiert, um die TA-Last jedes Problemtyps zu repräsentieren
- ⇒ es wurden eine Reihe von Benchmarks TPC-A/B/C/D/H/R/W durch ein Konsortium TPC (Transaction Processing Council) aus ca 40 Organisationen (DEC, Tandem, IBM, ...) festgelegt

TPC-A/B (1)

- ❑ Standard-Transaktion: Kontenbuchung („Debit-Credit“)¹
- ❑ erste OLTP-Benchmarks
 - ⇒ TPC-A: 1989
 - ⇒ TPC-B: 1990
- ❑ Satztypen:



1. Gray (ed.): *The Benchmark Handbook for Database and Transaction Processing Systems*. 2nd edition, Morgan Kaufmann, 1993, <http://www.benchmarkresources.com/handbook/index.html>

TPC-A/B (2)

❑ Mengengerüst abhängig vom Durchsatzziel (in Transaktionen pro Sekunde / tps)

⇒ pro tps:

- 1 BRANCH-Satz
- 10 TELLER-Sätze
- 100.000 ACCOUNT-Sätze
- HISTORY für 90 Tage ($90 \cdot 8 \cdot 60 \cdot 60 = 2.592.000$ Sätze)

⇒ daneben 10 Terminals pro tps

❑ Debit-Credit: Transaktionsprogramm

Read message from Terminal (*acctno, branchno, tellerno, delta*);

BEGIN_WORK { Beginn der Transaktion }

UPDATE ACCOUNT

SET balance = balance + *:delta*
WHERE acct_no = *:acctno*

SELECT balance INTO :abalance

FROM ACCOUNT
WHERE acct_no = *:acctno*;

UPDATE TELLER

SET balance = balance + *:delta*
WHERE teller_no = *:tellerno*

UPDATE BRANCH

SET balance = balance + *:delta*
WHERE branch_no = *:branchno*

INSERT INTO HISTORY (Tid, Bid, Aid, delta, time)

VALUES (*:tellerno, :branchno, :acctno, :delta, CURRENT*);

COMMIT_WORK ; { Ende der Transaktion }

Write message to Terminal (*abalance, ...*);

TPC-A/B (3)

- ❑ Benchmark-Forderungen:
 - ⇒ 15 % der Transaktionen betreffen Konto einer anderen Zweigstelle
 - ⇒ 90 % der TA sollen Antwortzeit von höchstens 2 Sekunden haben

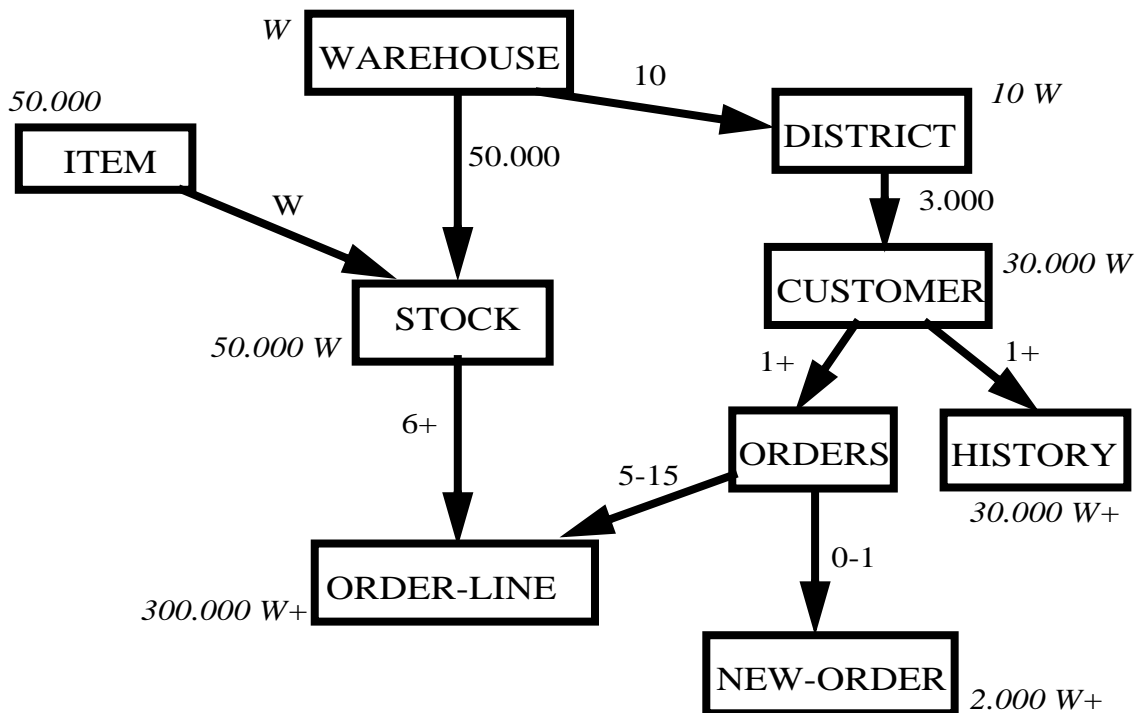
- ❑ Bestimmung der Kosten pro tps:
HW-, SW- und Wartungskosten für 5 Jahre

- ❑ TPC-A:
 - ⇒ Unterscheidung zwischen geographisch und lokal verteilten Systemen

- ❑ TPC-B:
 - ⇒ keine Berücksichtigung der DC-Komponente
 - ⇒ History-Sätze nur für 30 Tage

TPC-C (1)

- ❑ aktueller OLTP-Benchmark, Verabschiedung: 1993
- ❑ realistischere Transaktionslast aus mehreren Transaktionstypen unterschiedlicher Komplexität und Änderungshäufigkeit
- ❑ Anwendung: Bestellverwaltung im Großhandel
 - Betrieb umfaßt W Warenhäuser, pro Warenhaus 10 Distrikte, pro Distrikt 3000 Kunden
 - 50.000 Artikel; pro Warenhaus wird Anzahl vorhandener Artikel geführt
 - 1% aller Bestellungen müssen von nicht-lokalem Warenhaus angefordert werden
 - wenigstens ca. 500.000 Sätze (50 MB) pro Warenhaus
- ❑ 9 Satztypen



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)

❑ Durchsatzangabe für New-Order-Transaktionen in tpm-C (Transaktionen pro Minute)

TPC-C (3)

❑ 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 Artikel bestimmen, deren Bestand unter bestimmtem Grenzwert liegt (Read-Only)

❑ Festlegung des Transaktions-Mixes

- ⇒ Payment-TA müssen mindestens 43 % der Last ausmachen
- ⇒ Order-Status, Delivery und Stock-Level je mindestens 4 %
- ⇒ New-Order-Anteil variabel

❑ Antwortzeitrestriktionen

- ⇒ 90 % unter 5 Sek.
(New-Order, Payment, Order-Status, Delivery)
- ⇒ Stock-Level: 90% unter 20 Sek.

TPC-D (1)

❑ Decision Support

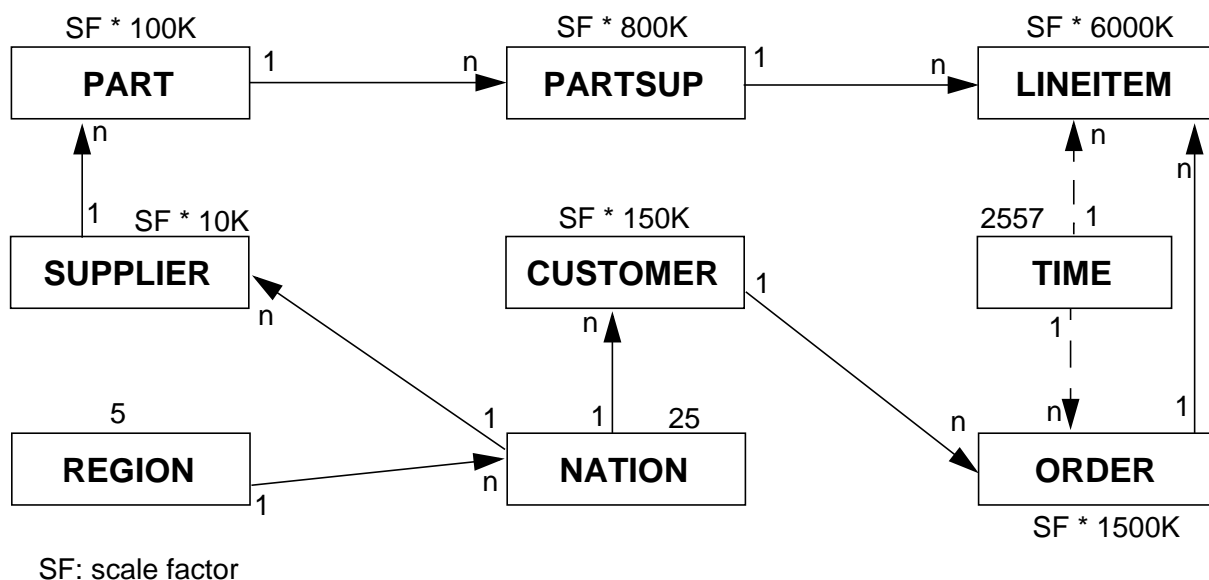
- ⇒ business critical questions
- ⇒ große Datenmengen
- ⇒ komplexe Anfragen

❑ erster Decision-Support-Benchmark: TPC-D (1995)

❑ Maßeinheiten:

- ⇒ query processing power at selected size (QppD@Size)
- ⇒ throughput at selected size (QthD@Size) in multi-user environment
- ⇒ price/performance: $\$/\text{QphD@Size}$;
query-per-hour rating derived from QppD and Qth

❑ Schema:



TPC-D (1)

□ Anfragen:

⇒ insgesamt 17 Anfragetypen

⇒ Beispiel: Q9

The Product Type Profit Measure Query finds, for each nation and each year, the profit for all parts ordered in that year which contain a specified substring in their names and which were filled by a supplier in that nation. The profit is defined as the sum of

$$[(L_EXTENDEDPRICE * (1 - L_DISCOUNT)) - (PS_SUPPLYCOST * L_QUANTITY)]$$

for all lineitems describing parts in the specified line. The query lists the nations in ascending alphabetical order and, for each nation, the year and profit in descending order by year (most recent first).

□ Anfrage Q2:

The Minimum Cost Supplier Query finds, in a given region, for each part of a certain type and size, the supplier who can supply it at minimum cost. If several suppliers in that region offer the desired part type and size at the same (minimum) cost, the query lists the parts from suppliers with the 100 highest account balances. For each supplier, the query lists the supplier's account balance, name and nation, the part's number and manufacturer, the supplier's address, phone number and comment information.

TPC-D (3)

❑ SQL-Umsetzung von Q2:

Note: return only the first 100 selected rows

```
SELECT S_ACCTBAL, S_NAME, N_NAME, P_PARTKEY, P_MFGR,  
        S_ADDRESS, S_PHONE, S_COMMENT  
FROM   PART, SUPPLIER, PARTSUPP, NATION, REGION  
WHERE  P_PARTKEY = PS_PARTKEY  
        AND S_SUPPKEY = PS_SUPPKEY  
        AND P_SIZE = [size]  
        AND P_TYPE LIKE '%[type]'  
        AND S_NATIONKEY = N_NATIONKEY  
        AND N_REGIONKEY = R_REGIONKEY  
        AND R_NAME = '[region]'  
        AND PS_SUPPLYCOST =  
        (SELECT MIN(PS_SUPPLYCOST)  
         FROM   PARTSUPP, SUPPLIER, NATION, REGION  
         WHERE  P_PARTKEY = PS_PARTKEY  
                 AND S_SUPPKEY = PS_SUPPKEY  
                 AND S_NATIONKEY = N_NATIONKEY  
                 AND N_REGIONKEY = R_REGIONKEY  
                 AND R_NAME = '[region]')  
ORDER BY S_ACCTBAL DESC, N_NAME, S_NAME, P_PARTKEY;
```

❑ weitere (aktuelle) Decision Support Benchmarks:

- ⇒ TPC-H: *ad-hoc, decision support benchmark*
- ⇒ TPC-R: *business reporting, decision support benchmark*

TPC-W

□ *transactional web e-commerce benchmark*

⇒ anhand White-Paper:

Wayne D. Smith (Intel):

TPC-W: Benchmarking An Ecommerce Solution

Zusammenfassung

- zunehmende Vielfalt an (TPC-)Benchmarks
 - ⇒ zentral: Standardisierung von Arbeitslasten für Transaktionssysteme
 - ⇒ einfach strukturierte Lasten: TPC-A, TPC-B
 - ⇒ komplex strukturierte Lasten: TPC-C
 - ⇒ Decision Support: TPC-D, TPC-H, TPC-R
 - ⇒ Web-basierte DB-Anwendungen: TPC-W
 - ⇒ weitere Application-Server-Benchmarks