

Parallele DBMS

Datenbanken Implementierungstechniken – SS2015

Alexander Müller

Fakultät
Informatik, Mathematik und Naturwissenschaften

Gliederung

1. Motivation
2. Arten der Parallelverarbeitung
3. Performance Kennzahlen
4. Architektur
5. Oracle Parallel Query
6. Ausführungspläne

Motivation

- Wachsende Anforderungen an Datenbanksysteme
 - komplexe Anfragen
 - aufwendige Berechnungen
 - Zugriff auf große Datenmengen
-
- Bearbeitungszeit von Transaktionen bzw. Anfragen soll dennoch akzeptabel sein
 - Parallelisierung soll Bearbeitungszeit verkürzen

Arten der Parallelverarbeitung

Inter-Transaktionsparallelität

- parallele Ausführung mehrerer unabhängiger Transaktionen oder DB-Anfragen (Mehrbenutzerbetrieb)

Intra-Transaktionsparallelität

- Parallelverarbeitung innerhalb einer Transaktion (Bearbeitungszeit verkürzen)

Weiter Unterteilbar:

- Query-Parallelität
 - Basisoperatoren wie Selektion, Projektion, Join etc. Parallelisieren
- Operatorparallelität
 - parallele Ausführung der einzelnen Basisoperatoren

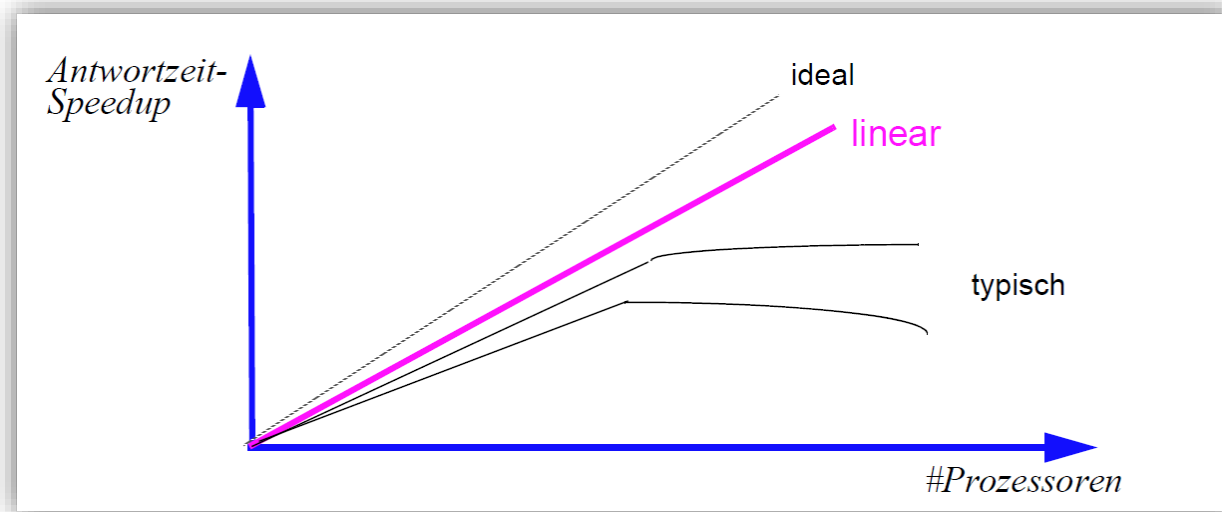
Performance Kennzahl: Speedup

- Bestimmung der Leistungsfähigkeit einer parallelen Verarbeitung im Vergleich zu einer sequentiellen Verarbeitung
- Speedup bestimmt den Einfluss der Prozessoranzahl auf eine **konstante Problemgröße**

$$\text{Speedup}(\mathbf{n}) = \frac{\text{Antwortzeit bei sequentieller Verarbeitung}}{\text{Antwortzeit bei paralleler Verarbeitung auf } \mathbf{n} \text{ Prozessoren}}$$

Performance Kennzahl: Speedup

Idealer, linearer und typischer Antwortzeit-Speedup



- Erhöhung der Prozessorzahl nicht unendlich möglich
→ Kommunikations-Overhead
- Speedup ist auf parallelisierbaren Anteil begrenzt

Performance Kennzahl: Scaleup

- Die Problemgröße (DB-Größe) linear mit der Prozessoranzahl erhöht

Antwortzeit-Scaleup

- Antwortzeit von n Rechnern und **n -facher Datenbankgröße** verglichen mit dem 1-Rechner-Fall
- Optimal: Antwortzeit-Scaleup = 1
- Durch Parallelisierung möglichst gleiche Antwortzeit wie bei einem Rechner

Performance Kennzahl: Scaleup

Durchsatz-Scaleup

- Verhältnis zwischen Transaktionsrate auf **n** gegenüber der Transaktionsrate auf einem Rechner
- auch **n**-fach große Datenbank
- Optimal: lineares Durchsatzwachstum
- idealerweise Scaleup-Wert **n**

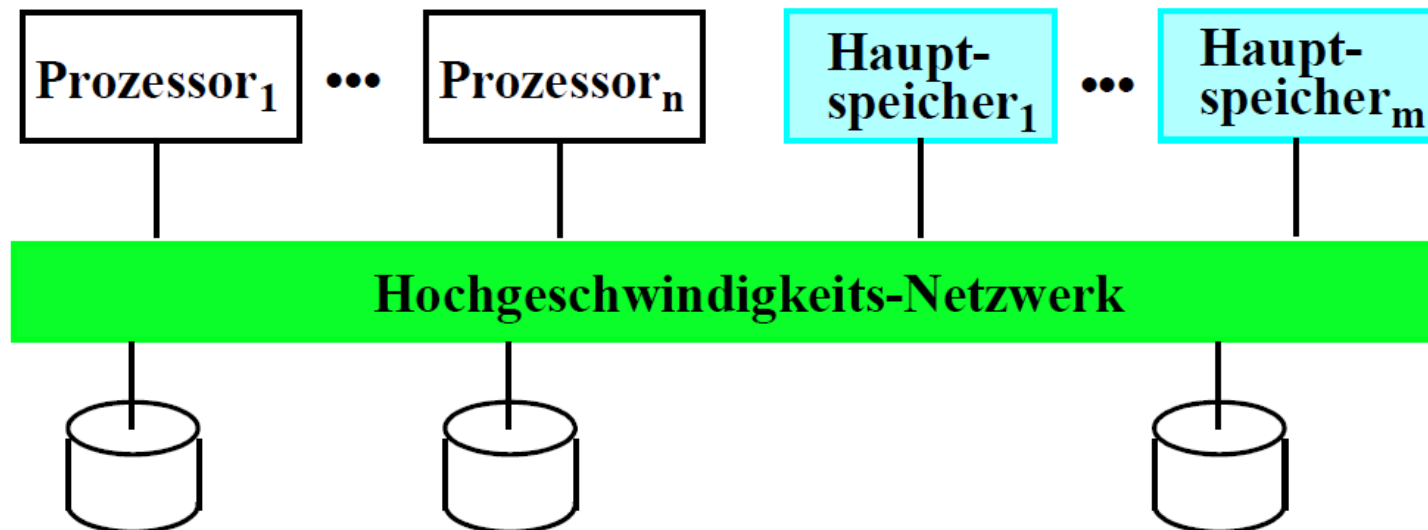
$$\text{Scaleup}(n) = \frac{\text{Durchsatz bei paralleler Verarbeitung}}{\text{Durchsatz bei sequentieller Verarbeitung}}$$

Architektur von Parallelen DBS

- Drei Grundlegende Typen

1. Shared Everything

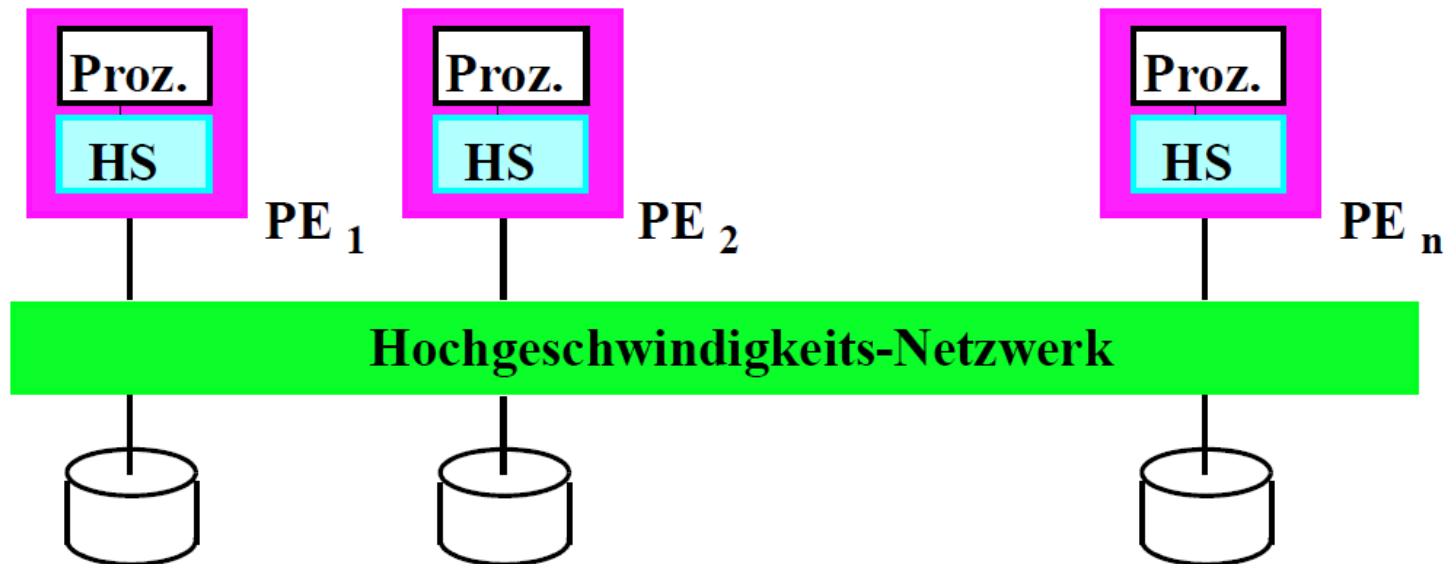
DBMS auf eng gekoppelter Multiprozessor-Rechnerumgebung



Architektur von Parallelen DBS

2. Shared Disk

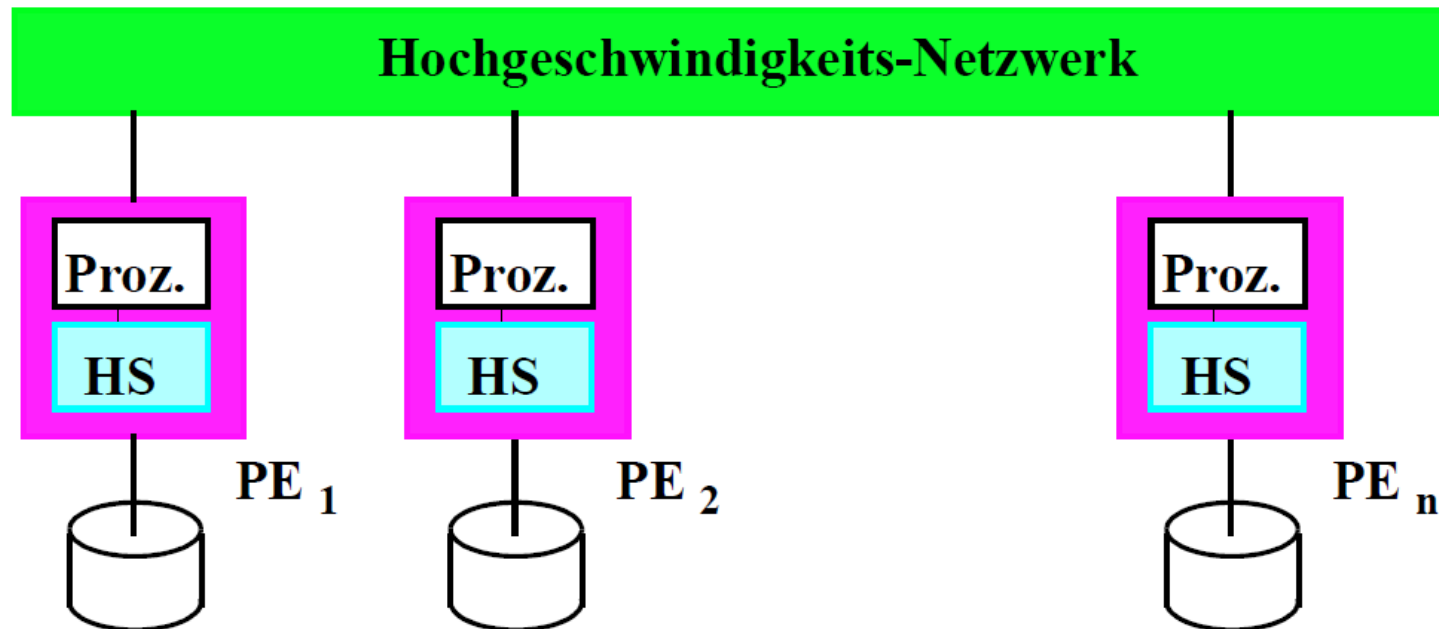
- lose gekoppelte Rechner, jeweils ein DBMS
- gemeinsame Speicherzuordnung



Architektur von Parallelen DBS

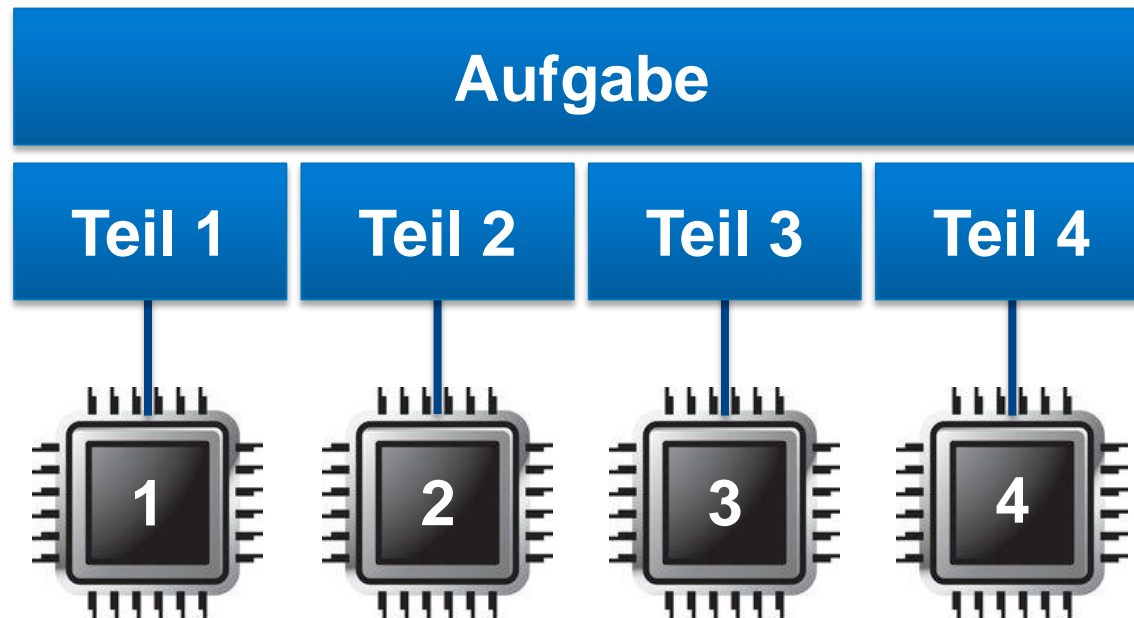
2. Shared Nothing

- lose gekoppelte Rechner, jeweils ein DBMS
- separater externe Speicher



Oracle Parallel Query

- parallele Verarbeitung seit Version 7 unterstützt
- Aufgaben der SQL-Anweisungen werden in kleine Teilaufgaben zerlegt und auf verschiedenen Prozessen ausgeführt



parallele Ausführung eines SQL Statements

Zwei Komponenten:

- Parallel Execution/ Query Coordinator (QC)
 - Daten (Indexe, Tabellen,...) in Granule aufteilen
 - Einzelergebnisse zusammenzuführen
 - übernimmt sequenzielle Anteile, z. B. Summieren aller Zwischenergebnisse

- Parallel Execution (PX) Serverprozesse
 - durch Parameter PARALLEL_MAX_SERVERS limitiert
 - Erhalten Granule von Execution Coordinator
 - Anzahl der Execution Server ist DoP (Degree of parallelism)

Parallelisierbare Operationen

- Table Scans (z.B. Full Table Scan)
- Full Index Scans
- Hash, Nested Loop und Sort Merge Join
- ORDER BY
- SELECT DISTINCT
- UNION und UNION ALL
- Backup/Restore/Recovery
- Ab Version 10g: Datapump Export/Import
- DML
 - UPDATE auf partitionierte Tabelle
 - INSERT...SELECT...
 - DELETE auf partitionierte Tabellen
 - MERGE
- DDL
 - CREATE INDEX
 - CREATE TABLE AS SELECT
 - ALTER TABLE MOVE

Initialisierungsparameter

- **CPU_COUNT**
 - aktuelle CPU-Anzahl –automatisch von Oracle gesetzt
- **PARALLEL_MIN_SERVERS**
 - Minimale Anzahl der parallelen Slave Prozesse pro Instanz
- **PARALLEL_MAX_SERVERS**
 - Maximale Anzahl der parallelen Slave Prozesse pro Instanz

Einstellungen auf Session-Ebene

- Parallel Query auf Session-Ebene einschalten:

```
ALTER SESSION FORCE PARALLEL QUERY;
```

- Vier parallele Prozesse bei Selects zulassen:

```
ALTER SESSION FORCE PARALLEL QUERY PARALLEL 4;
```

- Acht parallele Prozesse bei DML Befehlen zulassen:

```
ALTER SESSION FORCE PARALLEL DML PARALLEL 8;
```


Parallelität auf Tabellenebene

- Parallelität einschalten:

```
CREATE TABLE <tab> ( ... ) PARALLEL;
```

```
ALTER TABLE <tab> PARALLEL;
```

- mit explizitem Parallelisierungsgrad:

```
CREATE TABLE <tab> ( ... ) PARALLEL 4;
```

- Parallelisierung ausschalten:

```
ALTER TABLE <tab> NOPARALLEL;
```

Parallelität durch Hints

- beeinflussen Ausführungsplan
- Parallelisierung für SQL-Statement:

```
SELECT /*+ PARALLEL(b,4) */ * FROM big b.....;
```

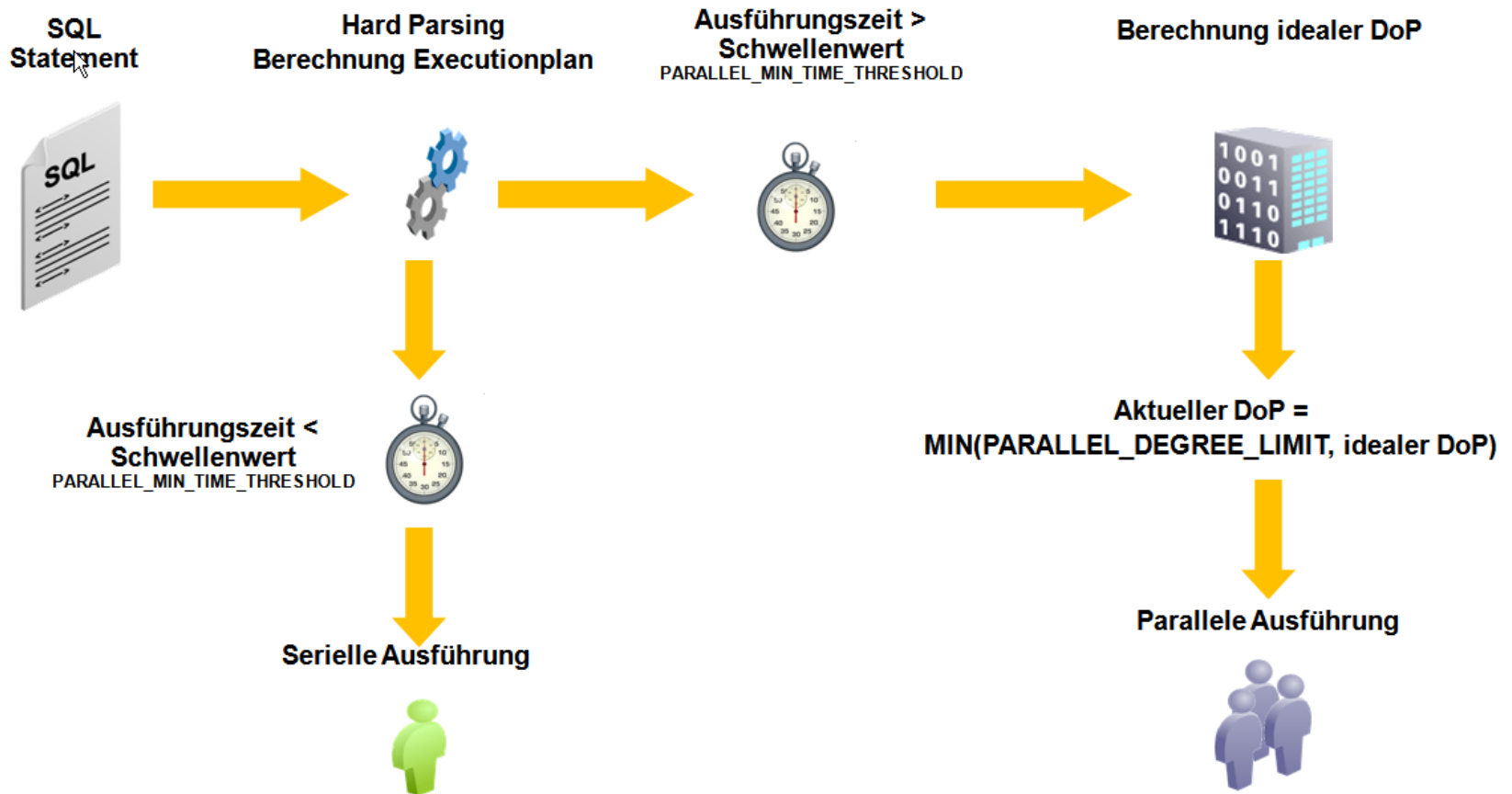
- Parallelisierung für Index-Nutzung:

```
/*+PARALLEL_INDEX(<index>,<degree>) */
```

- Parallelisierung bei Joins:

```
/*+ PQ_DISTRIBUTE(<table> <para1>,<para2>) */
```

Automatic Degree of Parallelism



Ausführungspläne anzeigen

```
explain plan for  
select sum(amount_sold)  
from sales  
where time_id between '1-JAN-01' and '31-DEC-10';
```

- Query wird dabei nicht ausgeführt

```
select * from  
table(dbms_xplan.display());
```

Sequentieller Ausführungsplan

Ausgabe:

```
-----  
|Id|Operation                |Name |Rows|Bytes|Cost |Time      |Pstart|Pstop|  
-----  
| 0|SELECT STATEMENT          |      |  1|  13| 1387|00:00:17|      |      | |
| 1| SORT AGGREGATE           |      |  1|  13|      |      |      |      |  
|*2|  FILTER                  |      |   |   |   |      |      |      |      |  
| 3|  PARTITION RANGE ITERATOR|      |229K|2916K| 1387|00:00:17| KEY  | KEY  |  
|*4|    TABLE ACCESS FULL    |SALES|229K|2916K| 1387|00:00:17| KEY  | KEY  |  
-----
```

Predicate Information (identified by operation id):

- ```

2 - filter(TO_DATE('1-JAN-01')<=TO_DATE('31-DEC-10'))
4 - filter("TIME_ID">='1-JAN-01' AND "TIME_ID"<='31-DEC-10'))
```

# Sequentieller Ausführungsplan

```
SQL> alter table sales parallel 4;
```

```

|Id | Operation | Name | Rows|Bytes| TQ | IN-OUT | PQ Distrib|

0	SELECT STATEMENT		1	13			
1	SORT AGGREGATE		1	13			
*2	PX COORDINATOR						
3	PX SEND QC (RANDOM)	:TQ10000	1	13	Q1,00	P->S	QC (RAND)
4	SORT AGGREGATE		1	13	Q1,00	PCWP	
*5	FILTER				Q1,00	PCWC	
6	PX BLOCK ITERATOR		229K	2916K	Q1,00	PCWC	
*7	TABLE ACCESS FULL	SALES	229K	2916K	Q1,00	PCWP	

```

```
Predicate Information (identified by operation id):

```

- 2 - filter(TO\_DATE('1-JAN-01')<=TO\_DATE('31-DEC-10'))
- 5 - filter(TO\_DATE('1-JAN-01')<=TO\_DATE('31-DEC-10'))
- 7 - filter("TIME\_ID">='1-JAN-01' AND "TIME\_ID"<='31-DEC-10')

- E. Rahm: Mehrrechner-Datenbank-Systeme: Grundlagen der verteilten und parallelen Datenbankverarbeitung, Addison-Wesley-Verlag, 1994.
- F. Schneede: Parallel Query ganz automatisch mit Oracle 11g, [http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tips/parallel\\_query/index.html](http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tips/parallel_query/index.html), ORACLE Deutschland GmbH
- A. Nanda: Automating Parallelism, ORACLE Magazine, Ausgabe 07/08.2010
- M Patzwahl: Präsentation Parallelisierung, [http://www.marcopatzwahl.de/publikationen/doag2008\\_parallel.pdf](http://www.marcopatzwahl.de/publikationen/doag2008_parallel.pdf), MuniQSoft GmbH, 2008.

# Vielen Dank für Ihre Aufmerksamkeit

Parallele DBMS

Alexander Müller

Fakultät  
Informatik, Mathematik und Naturwissenschaften