Inforsacom LOGICALIS

Business and technology working as one

Parallelisierung

Grundlagen und Nutzung

Stefan Seck
Solution Engineer
Inforsacom Logicalis GmbH

Düsseldorf, 30.05.2017

ORACLE Platinum Partner
Cloud Select
EMEA

Inforsacom Logicalis

Über

185 Mio. $

Umsatz im Fiskaljahr

Ca.

250

Mitarbeiter in Deutschland

Niederlassungen in

Berlin, Düsseldorf, Frankfurt, Köln, München, Stuttgart

Über

930

Kunden aus dem privaten und öffentlichen Bereich

Inforsacom LOGICALIS
Business and technology working as one

ituma

Inforsacom LOGICALIS
Business and technology working as one

Ein international einheitliches Solutions-, Services- und Cloud-Angebot

Rechenzentren

Nordamerika
Cincinnati (US)
Phoenix (US)
Edison (US)
Dayton (US)

Lateinamerika
São Paulo (Brasilien)

Europa
Slough (UK)
Bracknell (UK)
Guernsey (Kanalinseln)

Asien-Pazifik-Raum
Shanghai (China)
Cyberjaya (Malaysia)
Sydney (Australien)

Managed Services Center

Nordamerika
Cincinnati (US)
Detroit (US)

Lateinamerika
São Paulo (Brasilien)
Buenos Aires (Argentinien)

Europa
Slough (UK)
Jersey (Kanalinseln)
Cardiff (UK)
München (Deutschland)
Köln (Deutschland)
Rijswijk (Niederlande)
Madrid (Spanien)
Barcelona (Spanien)

Asien-Pazifik-Raum
Shanghai (China)
Cyberjaya (Malaysia)
Kuala Lumpur (Malaysia)
Singapur
Taipei (Taiwan)

Nordamerika

Niederlassungen
USA (über 20)

Lateinamerika

Niederlassungen
Brasilien, Argentinien, Bolivien, Chile, Kolumbien, Ecuador, Mexiko, Peru, Paraguay und Uruguay

Europa

Niederlassungen
Großbritannien, Irland, Niederlande, Spanien, Kanalinseln und Deutschland

Asien-Pazifischer-Raum

Niederlassungen
Australien, Singapur, Taiwan, Malaysia, Indonesien, China und Hongkong

Inforsacom LOGICALIS
Business and technology working as one

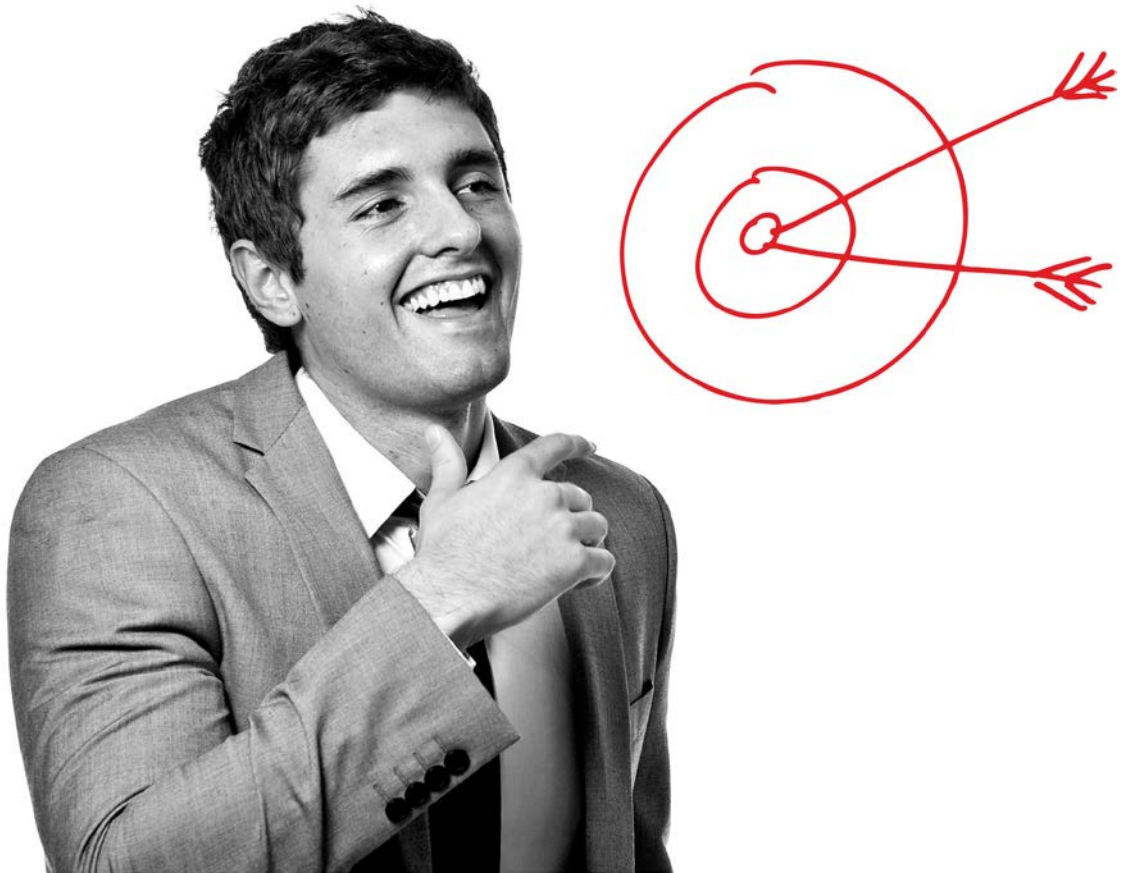
Agenda

- **Einführung**
- **Überblick parallelisierte Ausführungen**
- **Einstellungen und Parameter**
- **Zusammenfassung**

Inforsacom LOGICALIS

Business and technology working as one

Einführung



Einführung parallelisierte Ausführungen

- Eingeführt mit Oracle v7
- Ziel: Reduzierung der Ausführungsdauer großer Abfragen
  - Nutzung zusätzlicher bereitgestellter Prozesse
  - Arbeitet am Buffer Cache vorbei, direkt von Disk (Direct Path Read)
- Funktionalität der Oracle Enterprise Edition

Einführung parallelisierte Ausführungen

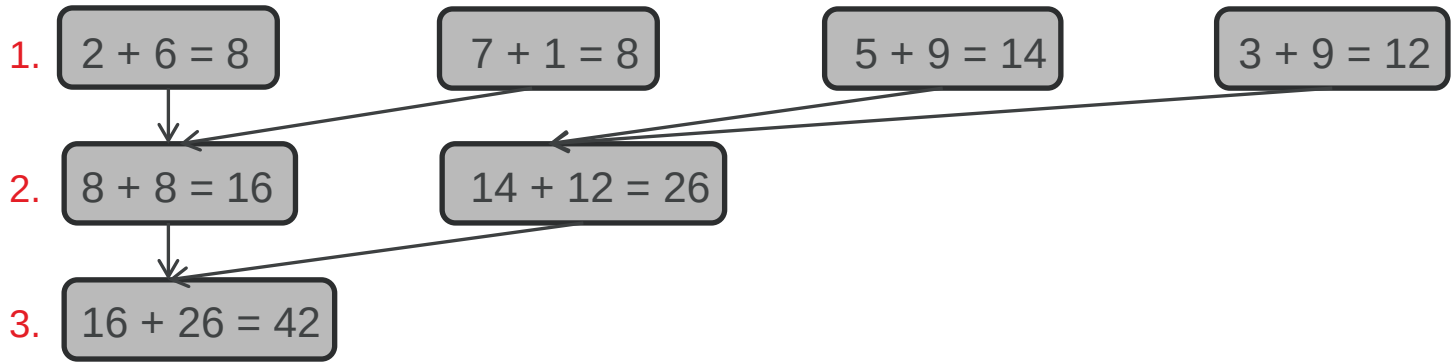
- Große Datenmengen in kürzerer Zeit verarbeiten
- Parallelisierte Ausführungen können alle vorhandenen Resourcen besser nutzen
  - CPU
  - I/O Channel
  - Storage Einheiten
  - Mehrere Nodes / Cluster
- DWH und Batchläufe in OLTP Applikationen können profitieren

Beispiel: Einfache Rechenaufgabe

- 2 + 6 + 7 + 1 + 5 + 9 + 3 + 9
- Seriell:
  1. 2 + 6 = 8
  2. 8 + 7 = 15
  3. 15 + 1 = 16
  4. 16 + 5 = 21
  5. 21 + 9 = 30
  6. 30 + 3 = 33
  7. 33 + 9 = 42

Beispiel: Einfache Rechenaufgabe

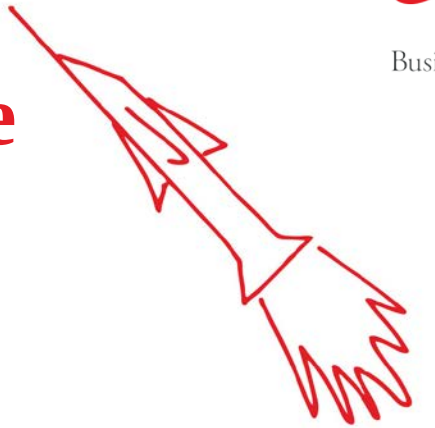
- 2 + 6 + 7 + 1 + 5 + 9 + 3 + 9
- Parallel:



Inforsacom LOGICALIS

Business and technology working as one

Überblick parallelisierte Ausführungen



Parallelisierte Ausführungen

- Serielle Ausführung

```
Plan hash value: 3549450340

-------------------------------------------------------------------------------------------------------
| Id  | Operation            | Name      | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |           |   918K|    27M|   942   (2)| 00:00:01 |       |       |
|*  1 |  HASH JOIN           |           |   918K|    27M|   942   (2)| 00:00:01 |       |       |
|   2 |   TABLE ACCESS FULL  | CUSTOMERS | 55500 |   704K|   423   (1)| 00:00:01 |       |       |
|   3 |   PARTITION RANGE ALL|           |   918K|    15M|   517   (2)| 00:00:01 |     1 |    28 |
|   4 |    TABLE ACCESS FULL | SALES     |   918K|    15M|   517   (2)| 00:00:01 |     1 |    28 |
-------------------------------------------------------------------------------------------------------

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

   1 - access("S"."CUST_ID"="C"."CUST_ID")
```

Parallelisierte Ausführungen

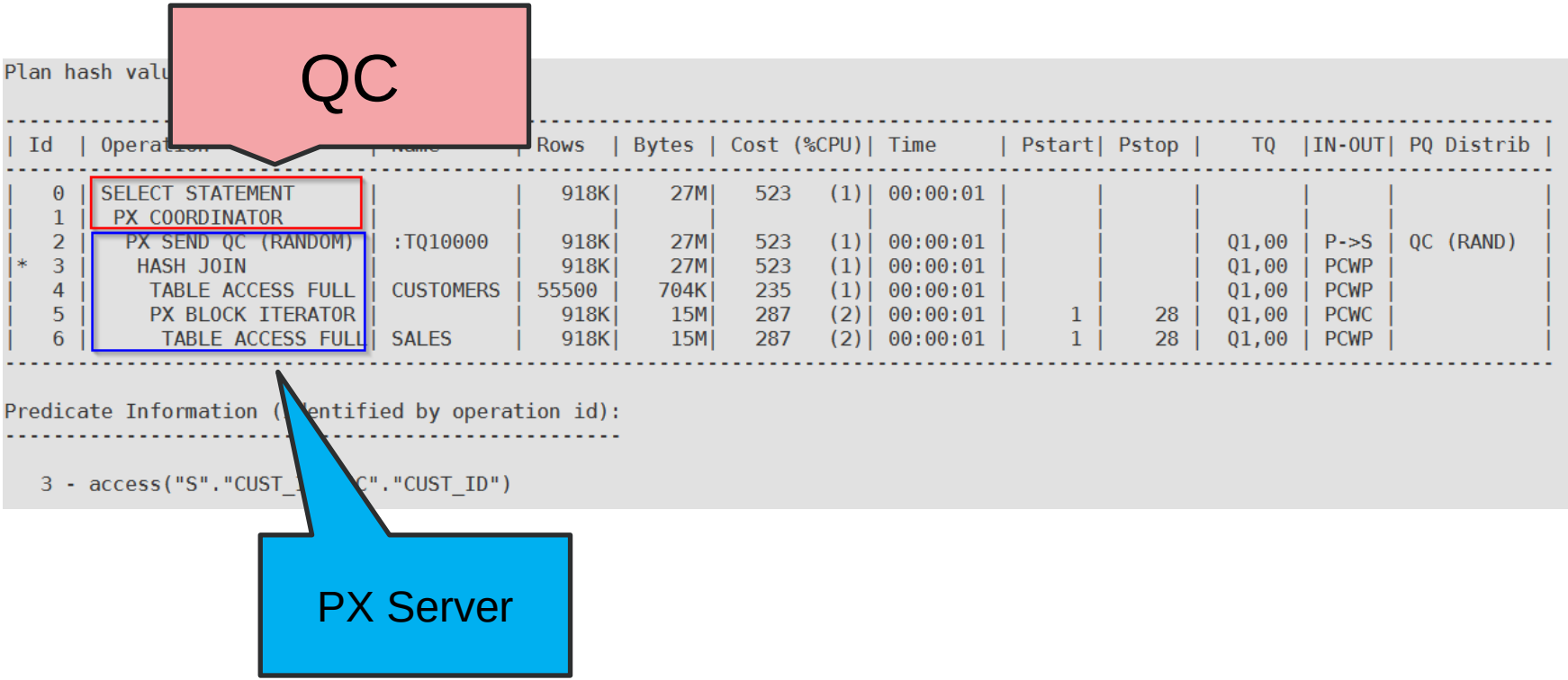
- Parallele Ausführung

```
Plan hash value: 2224329192

-----------------------------------------------------------------------------------------------------------------------------------
| Id  | Operation             | Name      | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |    TQ  |IN-OUT| PQ Distrib |
-----------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |           |   918K|    27M|   523   (1)| 00:00:01 |       |       |        |      |            |
|   1 |  PX COORDINATOR       |           |       |       |            |          |       |       |        |      |            |
|   2 |   PX SEND QC (RANDOM) | :TQ10000  |   918K|    27M|   523   (1)| 00:00:01 |       |       |  Q1,00 | P->S | QC (RAND)  |
|*  3 |    HASH JOIN          |           |   918K|    27M|   523   (1)| 00:00:01 |       |       |  Q1,00 | PCWP |            |
|   4 |     TABLE ACCESS FULL | CUSTOMERS | 55500 |   704K|   235   (1)| 00:00:01 |       |       |  Q1,00 | PCWP |            |
|   5 |     PX BLOCK ITERATOR |           |   918K|    15M|   287   (2)| 00:00:01 |     1 |    28 |  Q1,00 | PCWC |            |
|   6 |      TABLE ACCESS FULL| SALES     |   918K|    15M|   287   (2)| 00:00:01 |     1 |    28 |  Q1,00 | PCWP |            |
-----------------------------------------------------------------------------------------------------------------------------------

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

   3 - access("S"."CUST_ID"="C"."CUST_ID")
```

QC und PX Server

- QC – Query Coordinator
  - Koordiniert die parallele Ausführung
  - Verteilt die Arbeit an die PX Server
  - Führt alle Schritte durch, die nicht parallelisierbar sind
  - Empfängt die Einzelergebnisse der PX Server
- PX Server
  - Aus einem Pool von Prozessen heraus genutzt
  - Arbeiten Teilbereiche ab
  - Kommunizieren untereinander und mit dem QC

QC und PX Server



QC und PX Server

- SQL Monitoring des EM liefert das auch in bunt

Vorgang	Name	Geschätzt...	Kost...	Timeline(1354s)	A
SELECT STATEMENT					
PX COORDINATOR					
PX SEND QC (RANDOM)	:TQ10001	919K	2807		
HASH JOIN		919K	2807		
PX RECEIVE		56K	294		
PX SEND BROADCAST	:TQ10000	56K	294		
PX BLOCK ITERATOR		56K	294		
TABLE ACCESS FULL	CUSTOMERS	56K	294		
PX BLOCK ITERATOR		919K	1914		
TABLE ACCESS FULL	SALES	919K	1914		

Quelle: www.oracle.com

- Achtung: Oracle Tuning Pack zu lizensieren!

Producer / Consumer Model

- Parallele Prozesse arbeiten in Sets
  - Ein Set liest die Zeilen (Producer)
    - Sammeln und Filtern
    - Ergebnis an Consumer schicken
  - Ein Set empfängt und verarbeitet die Zeilen (Consumer)
    - Empfangen von Zeilen der Producer
    - Weitere Verarbeitung (Join-Operationen)
- Einfache bzw. partitionsweise Abfragen werden nicht in Sets verarbeitet

Producer / Consumer Model

- Einfache Abfrage

```
Global Stats
=============================================================
| Elapsed | Queuing |   Cpu    |  Other   | Fetch | Buffer |
| Time(s) | Time(s) | Time(s)  | Waits(s) | Calls |  Gets  |
=============================================================
|    1.67 |    0.00 |     0.98 |     0.69 | 61258 |   5124 |
=============================================================

Parallel Execution Details (DOP=2 , Servers Allocated=2)
=================================================================================================
|      Name      | Type   | Server# | Elapsed | Queuing |   Cpu    |  Other   | Buffer | Wait Events |
|                |        |         | Time(s) | Time(s) | Time(s)  | Waits(s) |  Gets  | (sample #)  |
=================================================================================================
| PX Coordinator | QC     |         |    1.06 |    0.00 |     0.39 |     0.67 |    192 |             |
| p000           | Set 1  |       1 |    0.31 |         |     0.29 |     0.01 |   2467 |             |
| p001           | Set 1  |       2 |    0.30 |         |     0.30 |          |   2465 |             |
=================================================================================================

SQL Plan Monitoring Details (Plan Hash Value=2224329192)
=========================================================================================================================================
| Id |        Operation         |   Name    |  Rows   | Cost |   Time    | Start  | Execs |   Rows   | Mem  | Activity | Activity Detail |
|    |                          |           | (Estim) |      | Active(s) | Active |       | (Actual) | (Max) |   (%)    |   (# samples)   |
=========================================================================================================================================
|  0 | SELECT STATEMENT         |           |         |      |        22 |     +0 |     3 |     919K |       |          |                 |
|  1 |   PX COORDINATOR         |           |         |      |        22 |     +0 |     3 |     919K |       |          |                 |
|  2 |    PX SEND QC (RANDOM)   | :TQ10000  |    919K |  523 |        20 |     +2 |     2 |     919K |       |          |                 |
|  3 |     HASH JOIN            |           |    919K |  523 |        21 |     +1 |     2 |     919K |    9M |   100.00 | Cpu (1)         |
|  4 |      TABLE ACCESS FULL   | CUSTOMERS |  55500 |  235 |         1 |     +2 |     2 |     111K |       |          |                 |
|  5 |      PX BLOCK ITERATOR   |           |    919K |  287 |        20 |     +2 |     2 |     919K |       |          |                 |
|  6 |       TABLE ACCESS FULL  | SALES     |    919K |  287 |        20 |     +2 |    33 |     919K |       |          |                 |
=========================================================================================================================================
```

Producer / Consumer Model

- Komplexere Abfrage

Parallel Execution Details (DOP=2 , Servers Allocated=4)

Name	Type	Server#	Elapsed Time(s)	Queuing Time(s)	Cpu Time(s)	Other Waits(s)	Buffer Gets	Wait Events (sample #)
PX Coordinator	QC		0.01	0.01	0.01		32	
p000	Set 1	1	0.01		0.01		1260	
p001	Set 1	2	0.01		0.01	0.00	1297	
p002	Set 2	1	0.02		0.02		211	
p003	Set 2	2	0.01		0.01		156	

SQL Plan Monitoring Details (Plan Hash Value=645391525)

Id	Operation	Name	Rows (Estim)	Cost	Time Active(s)	Start Active	Execs	Rows (Actual)
0	SELECT STATEMENT				1	+0	5	45
1	PX COORDINATOR				1	+0	5	45
2	PX SEND QC (RANDOM)	:TQ10002	145	257	1	+0	2	45
3	HASH GROUP BY		145	257	1	+0	2	45
4	PX RECEIVE		145	257	1	+0	2	90
5	PX SEND HASH	:TQ10001	145	257	1	+0	2	90
6	HASH GROUP BY		145	257	1	+0	2	90
7	HASH JOIN		1526	255	1	+0	2	16613
8	JOIN FILTER CREATE	:BF0000	2921	237	1	+0	2	37040
9	PX RECEIVE		2921	237	1	+0	2	37040
10	PX SEND BROADCAST	:TQ10000	2921	237	1	+0	2	37040
11	HASH JOIN		2921	237	1	+0	2	18520
12	JOIN FILTER CREATE	:BF0001	1	2	1	+0	2	2
13	TABLE ACCESS FULL	COUNTRIES	1	2	1	+0	2	2
14	JOIN FILTER USE	:BF0001	55500	235	1	+0	2	18520
15	PX BLOCK ITERATOR		55500	235	1	+0	2	18520
16	TABLE ACCESS FULL	CUSTOMERS	55500	235	1	+0	51	18520
17	JOIN FILTER USE	:BF0000	28995	19	1	+0	2	21982
18	PX BLOCK ITERATOR		28995	19	1	+0	2	21982
19	TABLE ACCESS FULL	SALES	28995	19	1	+0	33	21982

Inforsacom LOGICALIS
Business and technology working as one

Granules

- Kleinste Einheit, mit der Daten verarbeitet werden
- block-based Granules
  - Entsprechen einem oder mehreren physikalischen Blöcken einer Tabelle
  - Die Größe und Anzahl wird zur Laufzeit berechnet
    - Optimaler und ausbalancierter Zugriff
  - PX BLOCK ITERATOR
- partition-based Granules
  - Entsprechen einer Partition
  - Eine Partition wird von einem PX Server verarbeitet
  - PX PARTITION RANGE ALL oder PX PARTITION RANGE

Data Redistribution

- Komplexere Abfrage

Parallel Execution Details (DOP=2 , Servers Allocated=4)

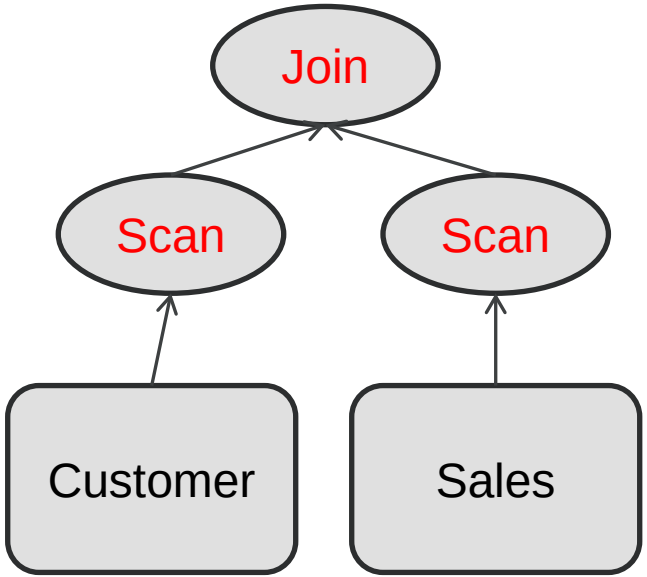
Name	Type	Server#	Elapsed Time(s)	Queuing Time(s)	Cpu Time(s)	Other Waits(s)	Buffer Gets	Wait Events (sample #)
PX Coordinator	QC		0.01	0.01	0.01		32	
p000	Set 1	1	0.01		0.01		1260	
p001	Set 1	2	0.01		0.01	0.00	1297	
p002	Set 2	1	0.02		0.02		211	
p003	Set 2	2	0.01		0.01		156	

SQL Plan Monitoring Details (Plan Hash Value=645391525)

Id	Operation	Name	Rows (Estim)	Cost	Time Active(s)	Start Active	Execs	Rows (Actual)
0	SELECT STATEMENT				1	+0	5	45
1	PX COORDINATOR				1	+0	5	45
2	PX SEND QC (RANDOM)	:TQ10002	145	257	1	+0	2	45
3	HASH GROUP BY		145	257	1	+0	2	45
4	PX RECEIVE		145	257	1	+0	2	90
5	PX SEND HASH	:TQ10001	145	257	1	+0	2	90
6	HASH GROUP BY		145	257	1	+0	2	90
7	HASH JOIN		1526	255	1	+0	2	16613
8	JOIN FILTER CREATE	:BF0000	2921	237	1	+0	2	37040
9	PX RECEIVE		2921	237	1	+0	2	37040
10	PX SEND BROADCAST	:TQ10000	2921	237	1	+0	2	37040
11	HASH JOIN		2921	237	1	+0	2	18520
12	JOIN FILTER CREATE	:BF0001	1	2	1	+0	2	2
13	TABLE ACCESS FULL	COUNTRIES	1	2	1	+0	2	2
14	JOIN FILTER USE	:BF0001	55500	235	1	+0	2	18520
15	PX BLOCK ITERATOR		55500	235	1	+0	2	18520
16	TABLE ACCESS FULL	CUSTOMERS	55500	235	1	+0	51	18520
17	JOIN FILTER USE	:BF0000	28995	19	1	+0	2	21982
18	PX BLOCK ITERATOR		28995	19	1	+0	2	21982
19	TABLE ACCESS FULL	SALES	28995	19	1	+0	33	21982

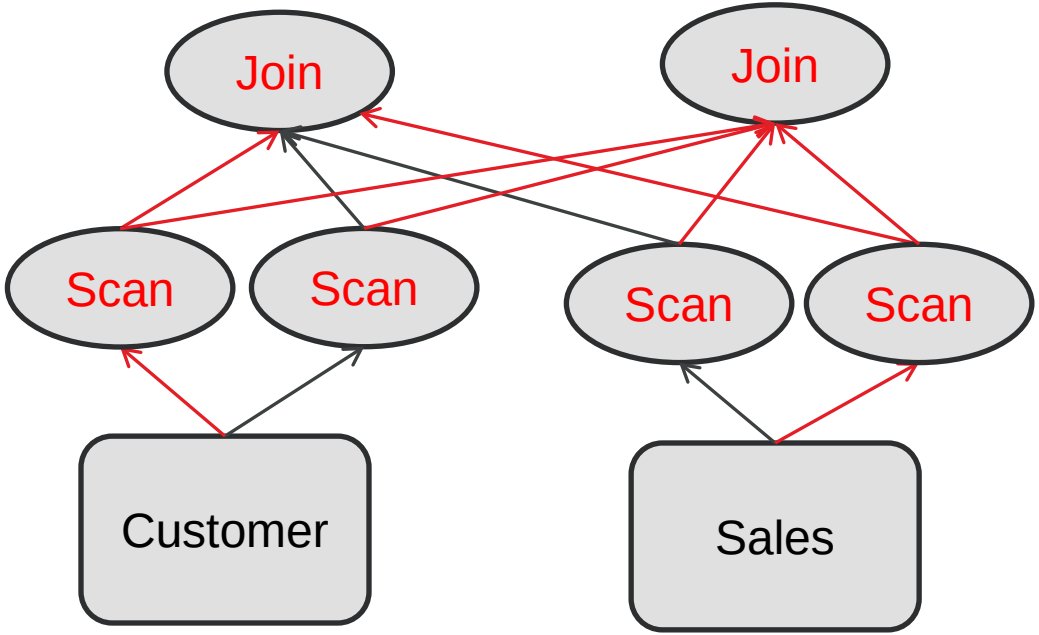
Data Redistribution

- Serielle Join Abfragen



Data Redistribution

- Parallele Join Abfragen



InMemory Parallele Ausführungen

- Oracle nutzt dabei keine Direct Path Reads
- Abhängig von Größe des Objekts und des Buffer Caches, werden die benötigten Blöcke in den Buffer Cache geladen
- Im RAC wird dabei cache-fusion abgeschaltet.
  - Quasi ein InMemory Shared-Nothing

Einstellungen und Parameter

Inforsacom LOGICALIS

Business and technology working as one

Einstellungen für parallelisierte Abfragen

- Für Queries und DDL per default aktiviert
- Für DML muss es noch aktiviert werden
  - ALTER SESSION ENABLE PARALLEL DML;
  - ALTER SYSTEM ENABLE PARALLEL DML;
- Es geht im Grunde um 3 Parameter
  - parallel_degree_policy
  - parallel_max_server
  - parallel_server_target

parallel_degree_policy

- PARALLEL_DEGREE_POLICY = *Auto / Limited / Manual*
- *AUTO*
  - Oracle aktiviert Auto DOP, statement queuing und in-memory parallel execution
  - Oracle berechnet selbst, ob und wie parallelisiert werden soll
  - Zuvor muss DBMS_RESOURCE_MANAGER.CALIBRATE_IO() gelaufen sein.
    - Achtung: Nicht bei Exadata ausführen.
- Limited
  - Kein statement queuing und in-memory parallel execution
  - Parallelisierte Abfragen nur bei Objekten mit parallel Degree

Weitere relevante Parameter

- PARALLEL_MAX_SERVERS = *PARALLEL_THREADS_PER_CPU * CPU_COUNT * concurrent_parallel_users * 5*
  - Concurrent_parallel_users is derived from memory settings.
    Use a value of 1 if automatic memory management is disabled.
    Use a value of 2 if PGA automatic memory management is enabled.
    Use a value of 4 if Global memory management or SGA memory target are used with PGA automatic memory management.
- PARALLEL_SERVERS_TARGET = *PARALLEL_THREADS_PER_CPU * CPU_COUNT * concurrent_parallel_users * 2*
- PARALLEL_MIN_TIME_THRESHOLD = *AUTO*
  - *Nur bei parallel_degree_policy = Auto /Limited*
- PARALLEL_FORCE_LOCAL

Parallel Statement Queueing

- Hints
  - /*+ NO_STMT_QUEUING */
  - /*+ STMT_QUEUING */
- Wait Event: resmgr:pq queued
- Status abfragen

```
SELECT  s.sql_id, s.sql_text
FROM v$SQL_MONITOR m, v$SQL s
WHERE status='QUEUED' AND m.sql_id = s.sql_id;
```

Database Resource Manager

- Parallelität lässt sich durch parallel_degree_limit einstellen
  - Nachteil: es gilt systemweit
- Alternativ lässt sich der DBRM nutzen, um verschiedene Gruppen aufzubauen
  - exec dbms_resource_manager.create_plan(plan => 'LIMIT_DOP', comment => 'Limit Degree of Parallelism');
  - exec dbms_resource_manager.create_plan_directive(plan=> 'LIMIT_DOP', -
  - group_or_subplan =>'OTHER_GROUPS' , -
  - comment => 'limits the parallelism', parallel_degree_limit_p1=> 4);

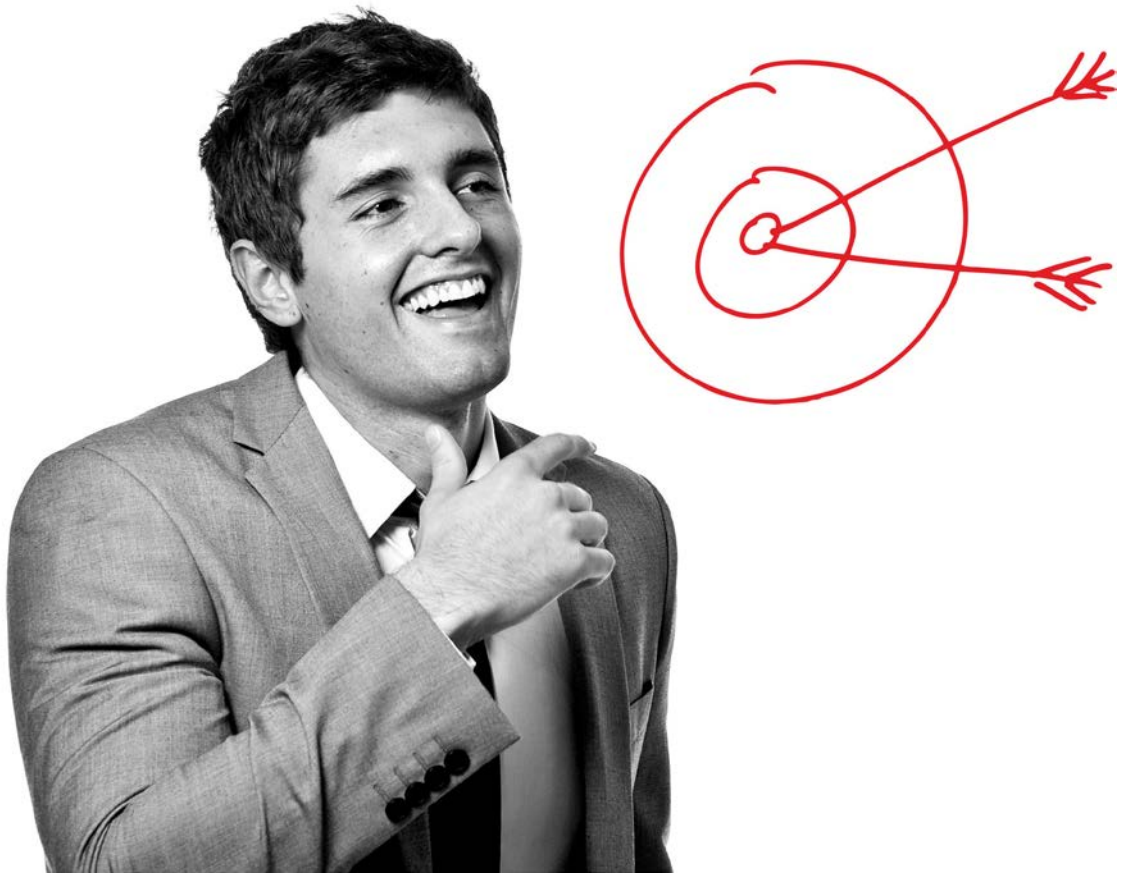
Monitoring paralleler Ausführungen

- Enterprise Manager
  - SQL Monitoring
- Sqlplus o.ä.
  - V$SQL_MONITOR
  - V$$PX_SESSION
  - V$PX_SESSTAT
  - V$PX_PROCESS
  - V$PX_PROCESS_SYSSTAT
  - V$PQ_TQSTAT

Inforsacom LOGICALIS

Business and technology working as one

Fazit



Fazit

- Parallelisierte Ausführungen können die Dauer von Statements dramatisch reduzieren
- Anzahl der PX Server ist entscheidend
  - Zu wenig ist nicht gut
  - Zu viel ist auch nicht gut
- Kommunikations zwischen den PX Server erfordert Overhead
- Kennen sie ihre Datenbank und die Workload
- Ziel: Alle Resourcen gleichmäßig ausnutzen

Links

- Using Parallel Execution (Doc ID 203238.1)
- Setup, Monitor, And Tune Parallelism In The Database (Doc ID 1549214.1)
- How to Limit the Parallel Degree for PX Queries with Resource Manager (Doc ID 1321099.1)
- Database VLDB and Partitioning Guide
  - http://docs.oracle.com/database/122/VLDBG/toc.htm

Inforsacom LOGICALIS
Business and technology working as one

Vielen Dank

Stefan Seck
Solution Engineer
Inforsacom Logicalis GmbH

stefan.seck@logicalis.de
Tel: +49 211 415592-30
Mobil: +49 151 52643576