




ORACLE®



The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



**ORACLE<sup>®</sup>**

## **Implement Best Practices for Extreme Performance with Oracle Data Warehousing**

Maria Colgan  
Principal Product Manager

# Agenda



- The three **P**s of Data Warehousing
  - Power
  - Partitioning
  - Parallel Execution
- Data Loading
- Workload Management
  - Statistics management
  - Initialization Parameters
  - Workload Monitoring

ORACLE  
**OPEN**  
WORLD

ORACLE  
**OPEN**  
WORLD

## The Three Ps

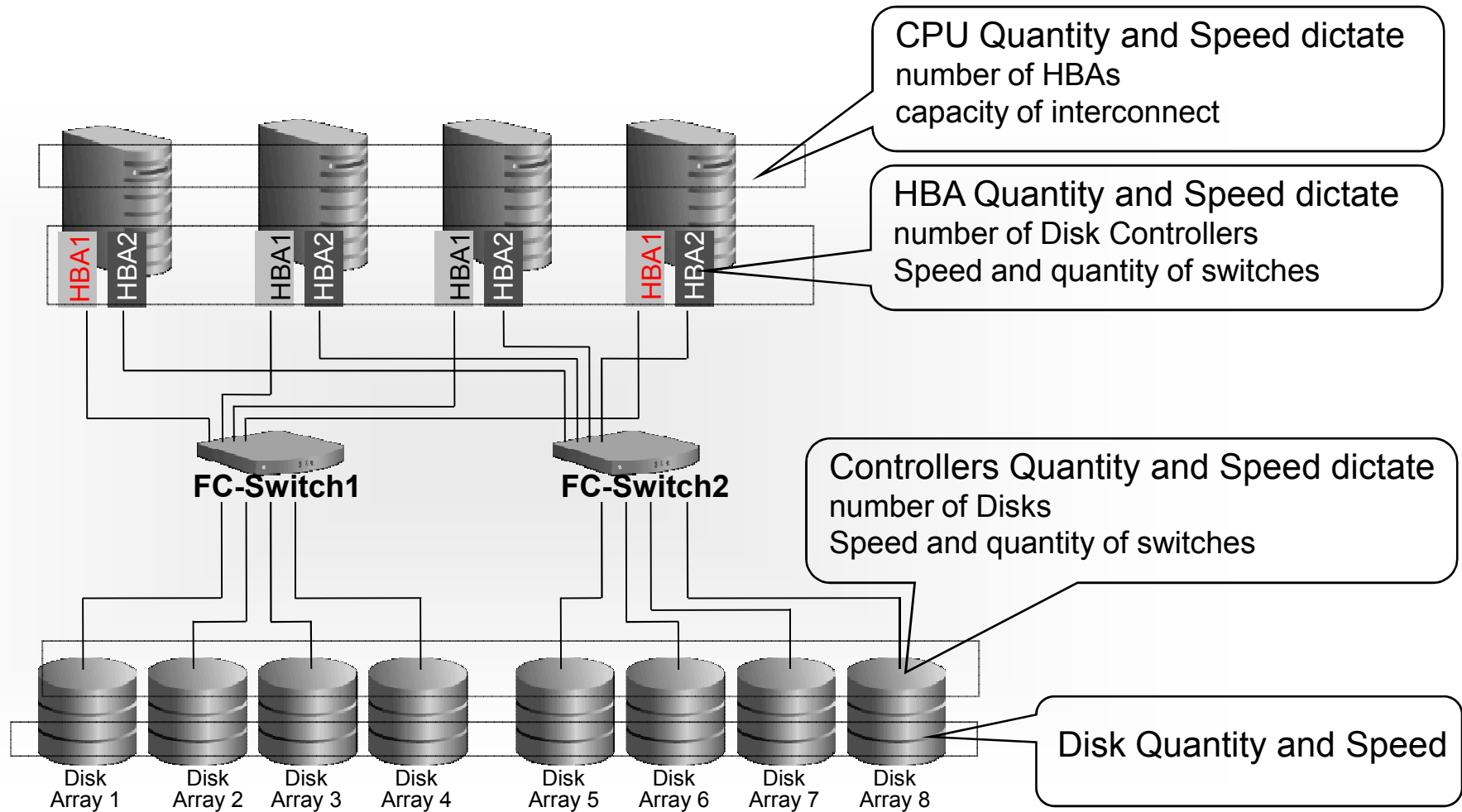


## 3 Ps - Power, Partitioning, Parallelism

- **Balanced Hardware Configuration**
  - Weakest link defines the throughput
- **larger tables or fact tables should be partitioned**
  - Facilitates data load, data elimination and join performance
  - Enables easier Information Lifecycle Management
- **Parallel Execution should be used**
  - Instead of one process doing all the work multiple processes working concurrently on smaller units
  - Parallel degree should be power of 2

# Balanced Configuration

“The weakest link” defines the throughput



# Sun Oracle Database Machine

A Balance Hardware Configuration

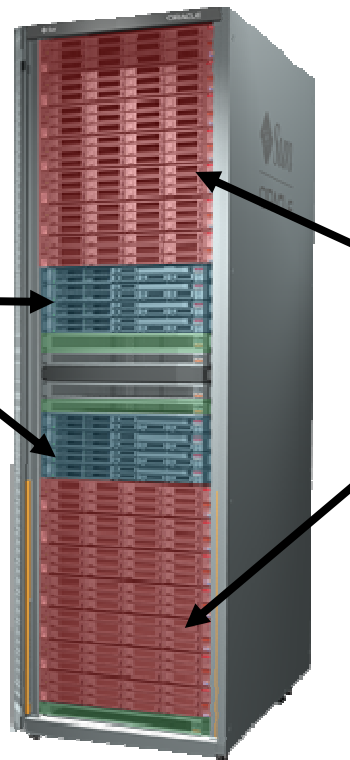
## Extreme Performance

### RAC Database Server Grid ■

- 8 High-performance low-cost compute servers
- 2 Intel quad-core Xeons each

### InfiniBand Network ■

- 3 36-port Infiniband
- 880 Gb/sec aggregate throughput

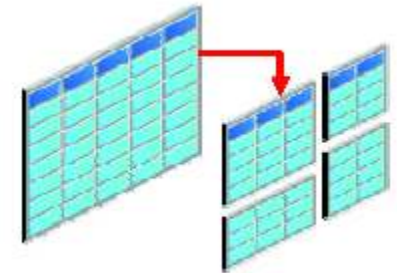


### Exadata Storage Server Grid ■

- 14 High-performance low-cost storage servers
- 100 TB raw SAS disk storage
- 5TB of Flash storage
- 21 GB/sec disk bandwidth
- 50 GB/sec flash bandwidth
- 100GB/sec memory bandwidth



# Partitioning



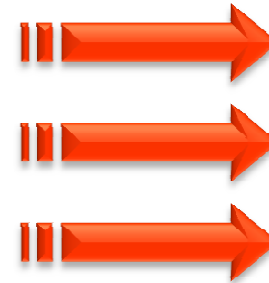
- Range partition large fact tables typically on date column
  - Consider data loading frequency
    - Is an incremental load required?
    - How much data is involved, a day, a week, a month?
  - Partition pruning for queries
    - What range of data do the queries touch - a quarter, a year?
- Subpartition by hash to improve join performance between fact tables and / or dimension tables
  - Pick the common join column
  - If all dimension have different join columns use join column for the largest dimension or most common join in the queries

# Partition Pruning

Q: What was the total sales for the weekend of May 20 - 22 2008?



```
Select sum(sales_amount)
From SALES
Where sales_date between
to_date('05/20/2008','MM/DD/YYYY')
And
to_date('05/23/2008','MM/DD/YYYY');
```



**Only the 3  
relevant  
partitions are  
accessed**

## Sales Table

May 18<sup>th</sup>  
2008

May 19<sup>th</sup>  
2008

May 20<sup>th</sup>  
2008

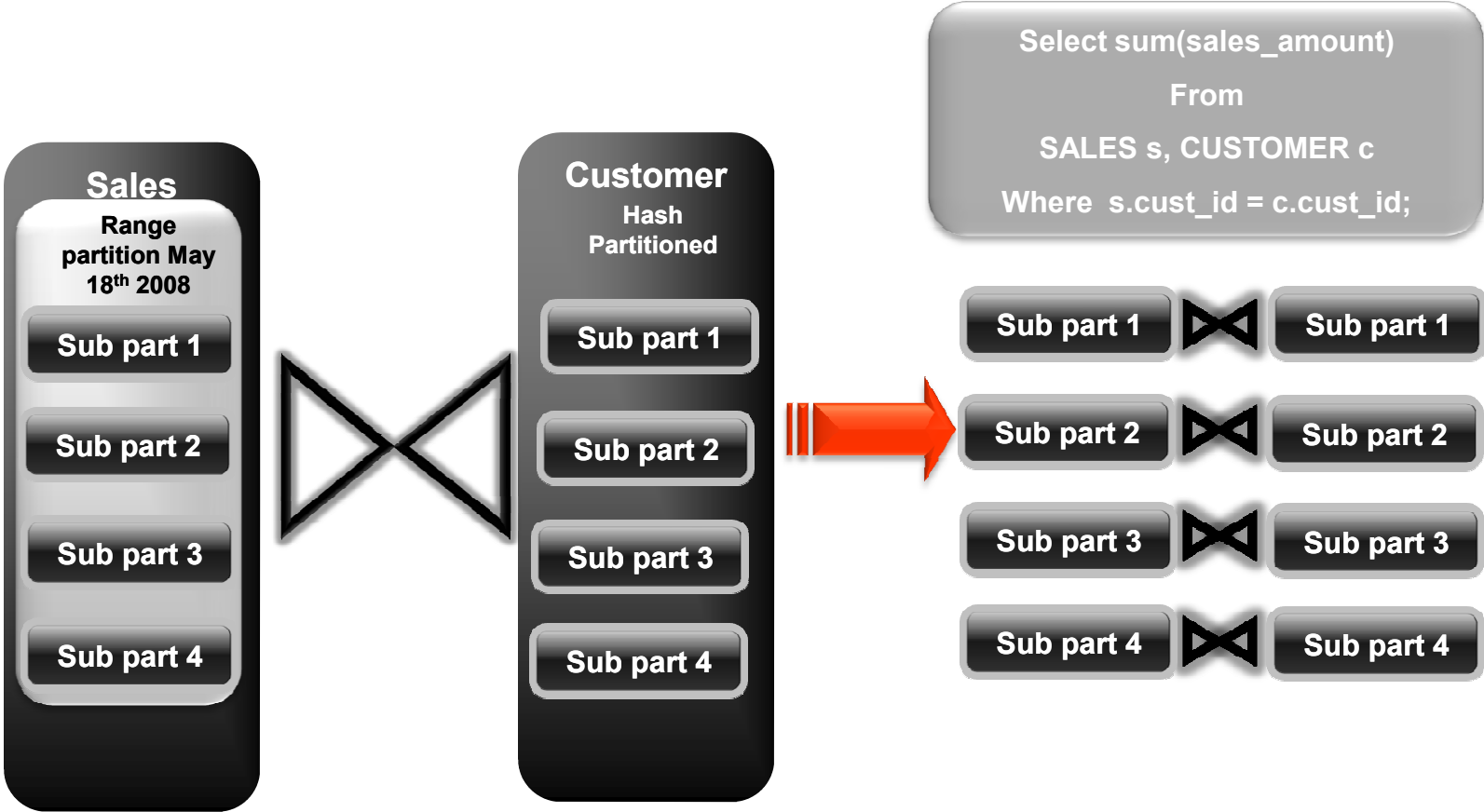
May 21<sup>st</sup>  
2008

May 22<sup>nd</sup>  
2008

May 23<sup>rd</sup>  
2008

May 24<sup>th</sup>  
2008

# Partition Wise join



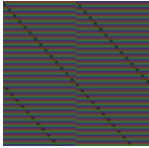
Both tables have the same degree of parallelism and are partitioned the same way on the join column (cust\_id)

A large join is divided into multiple smaller joins, each joins a pair of partitions in parallel

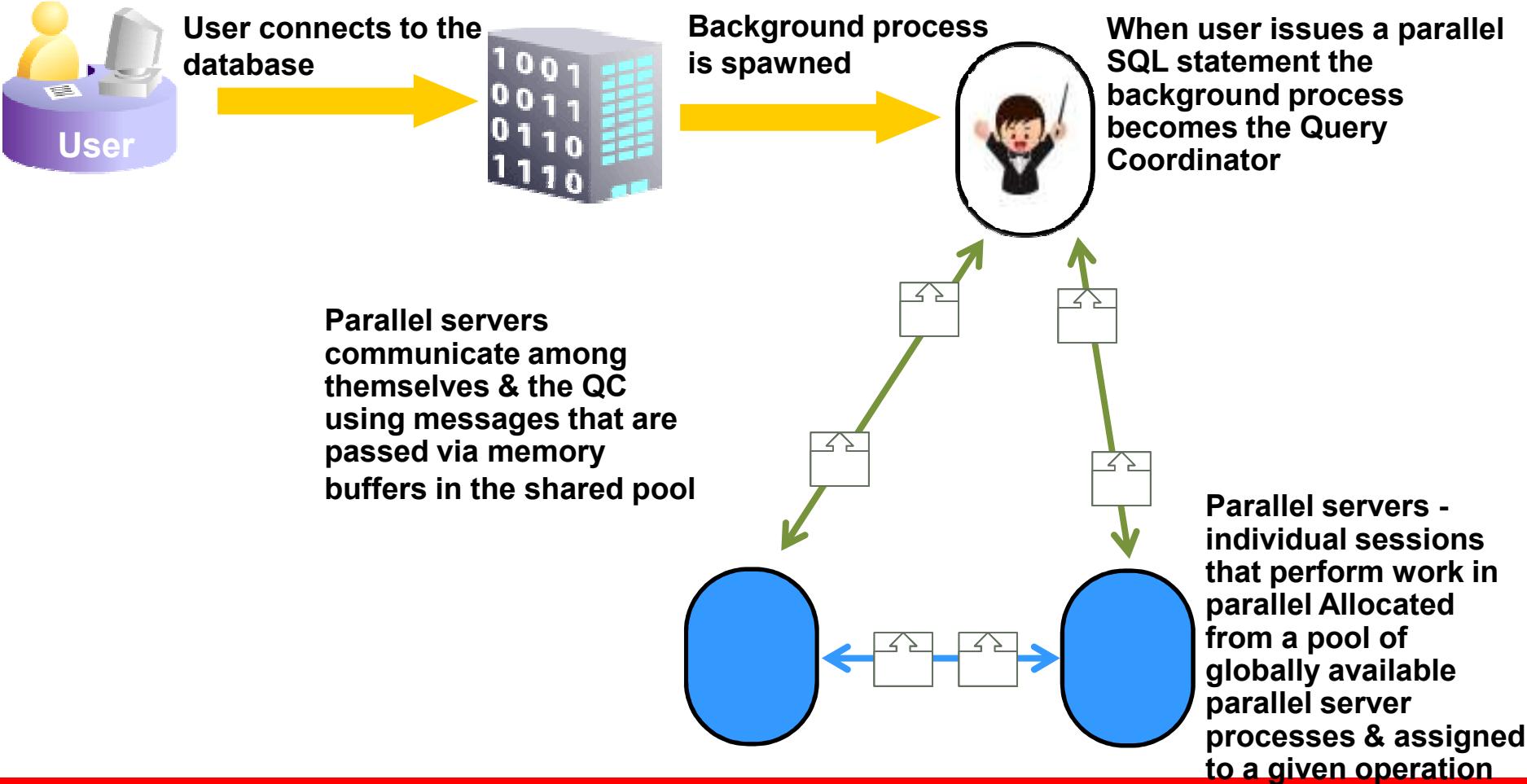
# Execution plan for partition-wise join

Partition Hash All above the join & single PQ set indicate partition-wise join

ID	Operation	Name	Pstart	Pstop	TQ	PQ Distrib
0	SELECT STATEMENT					
1	PX COORDINATOR					
2	PX SEND QC (RANDOM)	:TQ10001			Q1,01	QC (RAND)
3	SORT GROUP BY				Q1,01	
4	PX RECEIVE				Q1,01	
5	PX SEND HASH	:TQ10000			Q1,00	HASH
6	SORT GROUP BY				Q1,00	
7	PX PARTITION HASH ALL		1	128	Q1,00	
8	HASH JOIN				Q1,00	
9	TABLE ACCESS FULL	Customers	1	128	Q1,00	
10	TABLE ACCESS FULL	Sales	1	128	Q1,00	



# How Parallel Execution works



# Parallel Execution Plan

```
SELECT c.cust_name, s.purchase_date, s.amount
FROM sales s, customers c
WHERE s.cust_id = c.cust_id;
```

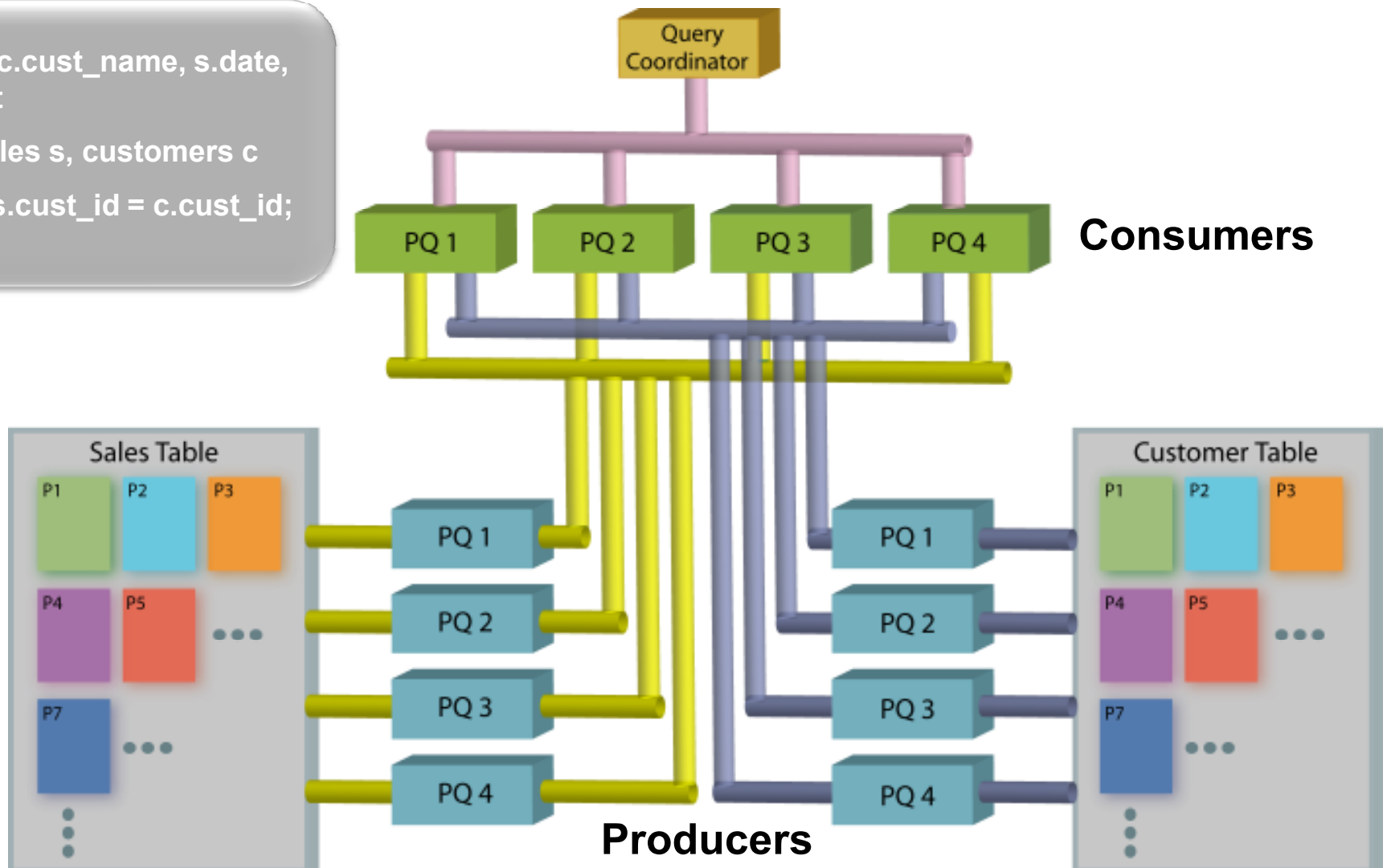
**Query Coordinator**

ID	Operation	Name	TQ	IN-OUT	PQ Distribution
0	SELECT STATEMENT				
1	PX COORDINATOR				
2	PX SEND QC {RANDOM}		Q1,01	P->S	
3	HASH JOIN		Q1,01	PCWP	
4	PX RECEIVE		Q1,01	PCWP	
5	PX SEND BROADCAST		Q1,01	P->P	BROADCAST
6	PX BLOCK ITERATOR		Q1,01	PCWP	
7	TABLE ACCESS FULL	CUSTOMERS	Q1,01	PCWP	
8	PX BLOCK ITERATOR		Q1,01	PCWP	
9	TABLE ACCESS FULL	SALES	Q1,01	PCWP	

**Parallel Servers  
do majority of the work**

# Parallel Execution of a Query

```
SELECT c.cust_name, s.date,
s.amount
FROM sales s, customers c
WHERE s.cust_id = c.cust_id;
```



# Producers and Consumer in the execution plan

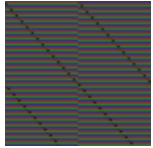
**Consumers**

**Query Coordinator**

ID	Operation	Name	TQ	IN-OUT	PQ Distribution
0	SELECT STATEMENT				
1	PX COORDINATOR				
2	PX SEND QC {RANDOM}		Q1,02	P->S	
3	HASH JOIN		Q1,02	PCWP	
4	PX RECEIVE		Q1,02	PCWP	
5	PX SEND HASH		Q1,00	P->P	
6	PX BLOCK ITERATOR		Q1,00	PCWP	
7	TABLE ACCESS FULL	CUSTOMERS	Q1,00	PCWP	
8	PX RECEIVE		Q1,02	PCWP	
9	PX SEND HASH		Q1,01	P->P	
10	PX BLOCK ITERATOR		Q1,01	PCWP	
11	TABLE ACCESS FULL	SALES	Q1,01	PCWP	

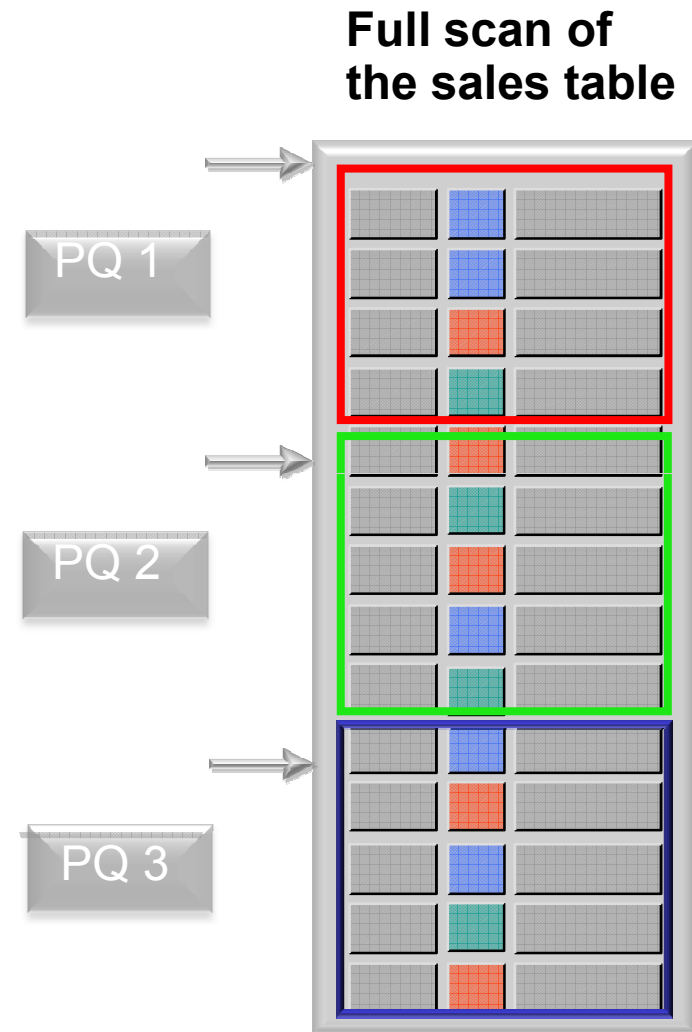
**Producers**





## Parallel Execution of a Scan

- Data is divided into Granules
  - block range or partition
- Each Parallel Server is assigned one or more Granules
- No two Parallel Servers ever contend for the same Granule
- Granules are assigned so that the load is balanced across all Parallel Servers
- **Dynamic Granules chosen by the optimizer**
- Granule decision is visible in execution plan



# Identifying Granules of Parallelism during scans in the plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop	TQ	IN-OUT	PQ	Distrib
0	SELECT STATEMENT		17	153	565 (100)	00:00:07						
1	PX COORDINATOR											
2	PX SEND QC (RANDOM)	:TQ10001	17	153	565 (100)	00:00:07			Q1,01	P->S	QC	(RAND)
3	HASH GROUP BY		17	153	565 (100)	00:00:07			Q1,01	PCWP		
4	PX RECEIVE		17	153	565 (100)	00:00:07			Q1,01	PCWP		
5	PX SEND HASH	:TQ10000	17	153	565 (100)	00:00:07			Q1,00	P->P	HASH	
6	HASH GROUP BY		17	153	565 (100)	00:00:07			Q1,00	PCWP		
7	<b>PX BLOCK ITERATOR</b>		10M	85M	60 (97)	00:00:01	1	16	Q1,00	PCWC		
* 8	TABLE ACCESS FULL	SALES	10M	85M	60 (97)	00:00:01	1	16	Q1,00	PCWP		

predicate Information (identified by operation id):

8 - filter("CUST\_ID"<=22810 AND "CUST\_ID">=22300)

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop	TQ	IN-OUT	PQ	Distrib
0	SELECT STATEMENT		17	153	2 (50)	00:00:01						
1	PX COORDINATOR											
2	PX SEND QC (RANDOM)	:TQ10001	17	153	2 (50)	00:00:01			Q1,01	P->S	QC	(RAND)
3	HASH GROUP BY		17	153	2 (50)	00:00:01			Q1,01	PCWP		
4	PX RECEIVE		26	234	1 (0)	00:00:01			Q1,01	PCWP		
5	PX SEND HASH	:TQ10000	26	234	1 (0)	00:00:01			Q1,00	P->P	HASH	
6	<b>PX PARTITION RANGE ALL</b>		26	234	1 (0)	00:00:01	1	16	Q1,00	PCWC		
7	TABLE ACCESS BY LOCAL INDEX ROWID	SALES	26	234	1 (0)	00:00:01	1	16	Q1,00	PCWP		
* 8	INDEX RANGE SCAN	SALES_CUST	26		0 (0)	00:00:01	1	16	Q1,00	PCWP		

predicate Information (identified by operation id):

8 - access("CUST\_ID">=22300 AND "CUST\_ID"<=22810)

# Controlling Parallel Execution on RAC

## 1. Use RAC Services

Create two services

Srvctl add service -d database\_name

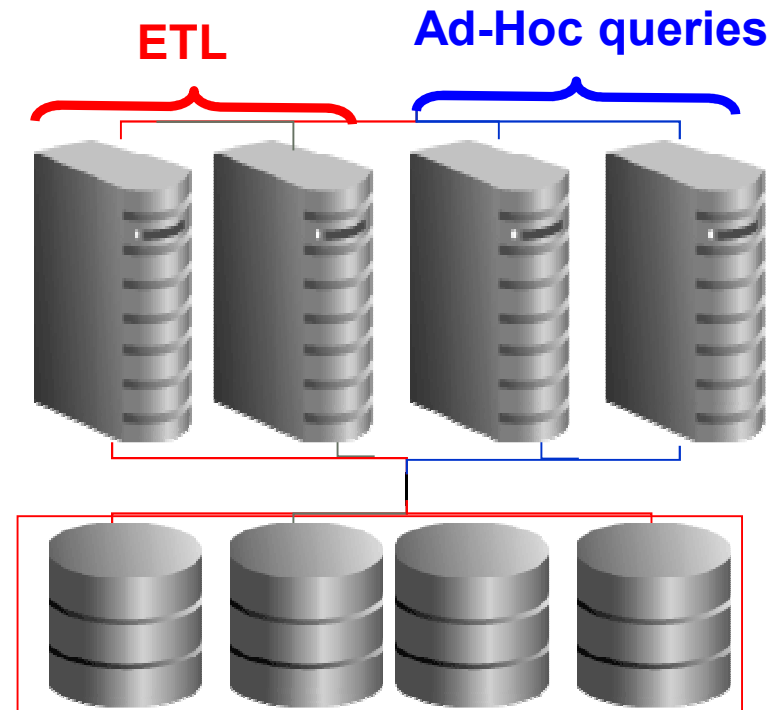
-s ETL

-r sid1, sid2

Srvctl add service -d database\_name

-s AHOC

-r sid3, sid4



2. **PARALLEL\_FORCE\_LOCAL** - New Parameter forces parallel statement to run on just node it was issued on  
Default is **FALSE**



## Use Parallel Execution with common sense

- Parallel execution provides performance boost but requires more resources
- General rules of thumb for determining the appropriate DOP
  - objects smaller than 200 MB should not use any parallelism
  - objects between 200 MB and 5GB should use a DOP of 4
  - objects beyond 5GB use a DOP of 32

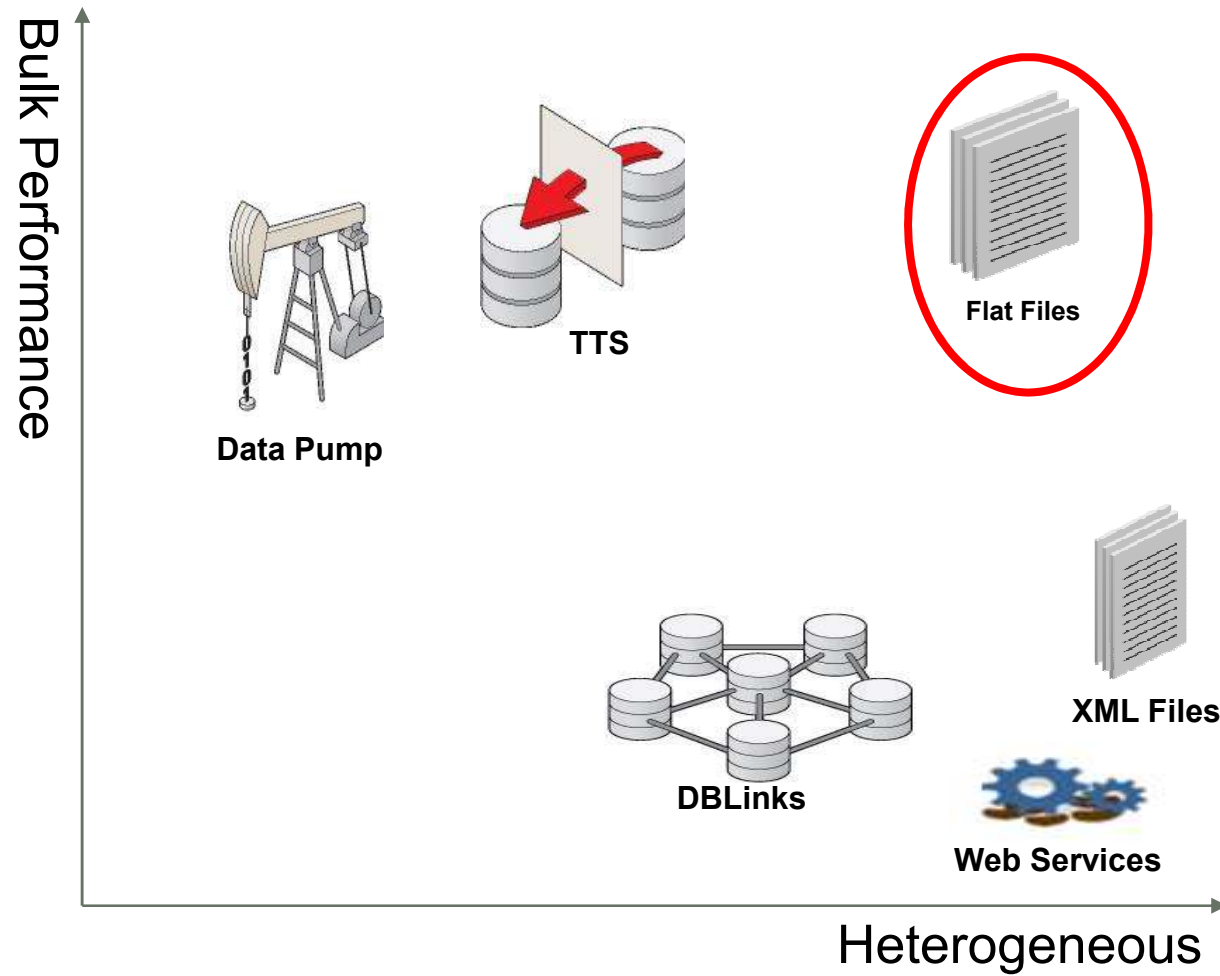
Mileage may vary depending on concurrent workload and hardware configuration

ORACLE  
**OPEN**  
WORLD

ORACLE  
**OPEN**  
WORLD

# Data Loading

# Access Methods





# Data Loading Best Practices

- **External Tables**

- Allows flat file to be accessed via SQL PL/SQL as if it was a table
- Enables complex data transformations & data cleansing to occur “on the fly”
- Avoids space wastage

- **Pre-processing**

- Ability to specify a program that the access driver will execute to read the data
- Specify gunzip to decompress a .gzip file “on the fly” while its being

- **Direct Path in parallel**

- Bypasses buffer cache and writes data directly to disk via multi-block async IO
- Use parallel to speed up load
- Remember to use `Alter session enable parallel DML`

- **Range Partitioning**

- Enables partition exchange loads

- **Data Compression**



## SQL Loader or External Tables

- And the winner is => External Tables
- Why:
  - Full usage of SQL capabilities directly on the data
  - Automatic use of parallel capabilities (just like a table)
  - No need to stage the data again
  - Better allocation of space when storing data
    - High watermark brokering
    - Autoallocate tablespace will trim extents after the load
  - Interesting capabilities like
    - The usage of data pump
    - The usage of pre-processing





## Tips for External Tables

- File locations and size
  - When using multiple files the file size should be similar
  - List largest to smallest in LOCATION clause if not similar in size
- File Formats
  - Use a format allowing position-able and seek-able scans
  - Delimitate clearly and use well known record termination to allow for automatic Granulation
  - Always specify the character set if its different to the database
- Consider compressing data files and uncompressing during loading
- Run all queries before the data load to populate column usage for histogram creation during statistics gathering



## Pre-Processing in an External Table

- New functionality in 11.1.0.7 and 10.2.0.5
- Allows flat files to be processed automatically during load
  - Decompression of large file zipped files
- Pre-processing doesn't support automatic granulation
  - Need to supply multiple data files - # of files will determine DOP
- Need to Grant read, execute privileges directories

```
CREATE TABLE sales_external
(...)
ORGANIZATION EXTERNAL
(
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY data_dir1
  ACCESS PARAMETERS
  (RECORDS DELIMITED BY NEWLINE
   PREPROCESSOR exec_dir: 'gunzip'
   FIELDS TERMINATED BY '|'
  )
  LOCATION (...);
```



## Direct Path Load

- Data is written directly to the database storage using multiple blocks per I/O request using asynchronous writes
- A CTAS command always uses direct path
- An Insert As Select needs an APPEND hint to go direct

```
Insert /*+ APPEND */ into Sales partition(p2)
  Select * From ext_tab_for_sales_data;
```

- Only one direct path operation can occur on an object
  - By specifying a specific partition name in the table you can do multiple concurrent direct path loads into a partitioned table

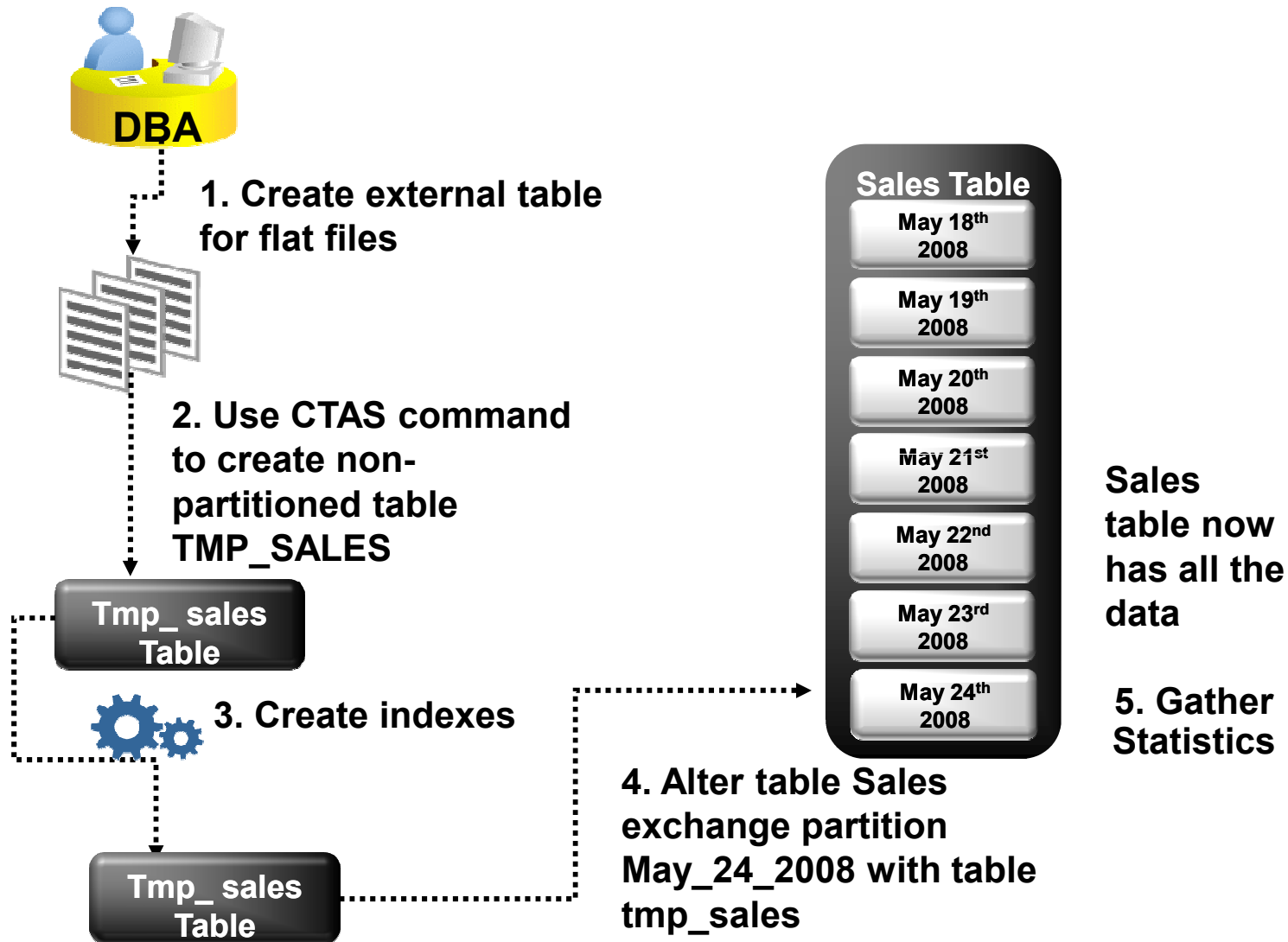


## Parallel Load

- Ensure direct path loads go parallel
  - Specify parallel degree either with hint or on both tables
  - Enable parallelism by issuing alter session command
- CTAS will go parallel automatically when DOP is specified
- IAS will not – it needs parallel DML to be enabled

```
ALTER SESSION ENABLE PARALLEL DML;
```

# Partition Exchange Loading





## Data Compression

- Use if data being loaded will be read / used more than once
- Works by eliminating duplicate values within a database block
- Reduces disk and memory usage, often resulting in better scale-up performance for read-only operations
- Require additional CPU during the initial data load
- But what if workload requires conventional DML access to the data after it has been loaded ?

Use the COMPRESS FOR ALL OPERATIONS

ORACLE<sup>®</sup> 11g<sup>™</sup>  
DATABASE

ORACLE  
**OPEN**  
WORLD

ORACLE  
**OPEN**  
WORLD

# Workload Monitoring

ORACLE®



## Statistics gathering

- You must gather optimizer statistics
  - Using dynamic sampling is not an adequate solution
- Run all queries against empty tables to populate column usage
  - This helps identify which columns automatically get histograms created on them
- Optimizer statistics should be gathered after the data has been loaded but before any indexes are created
  - Oracle will automatically gather statistics for indexes as they are being created





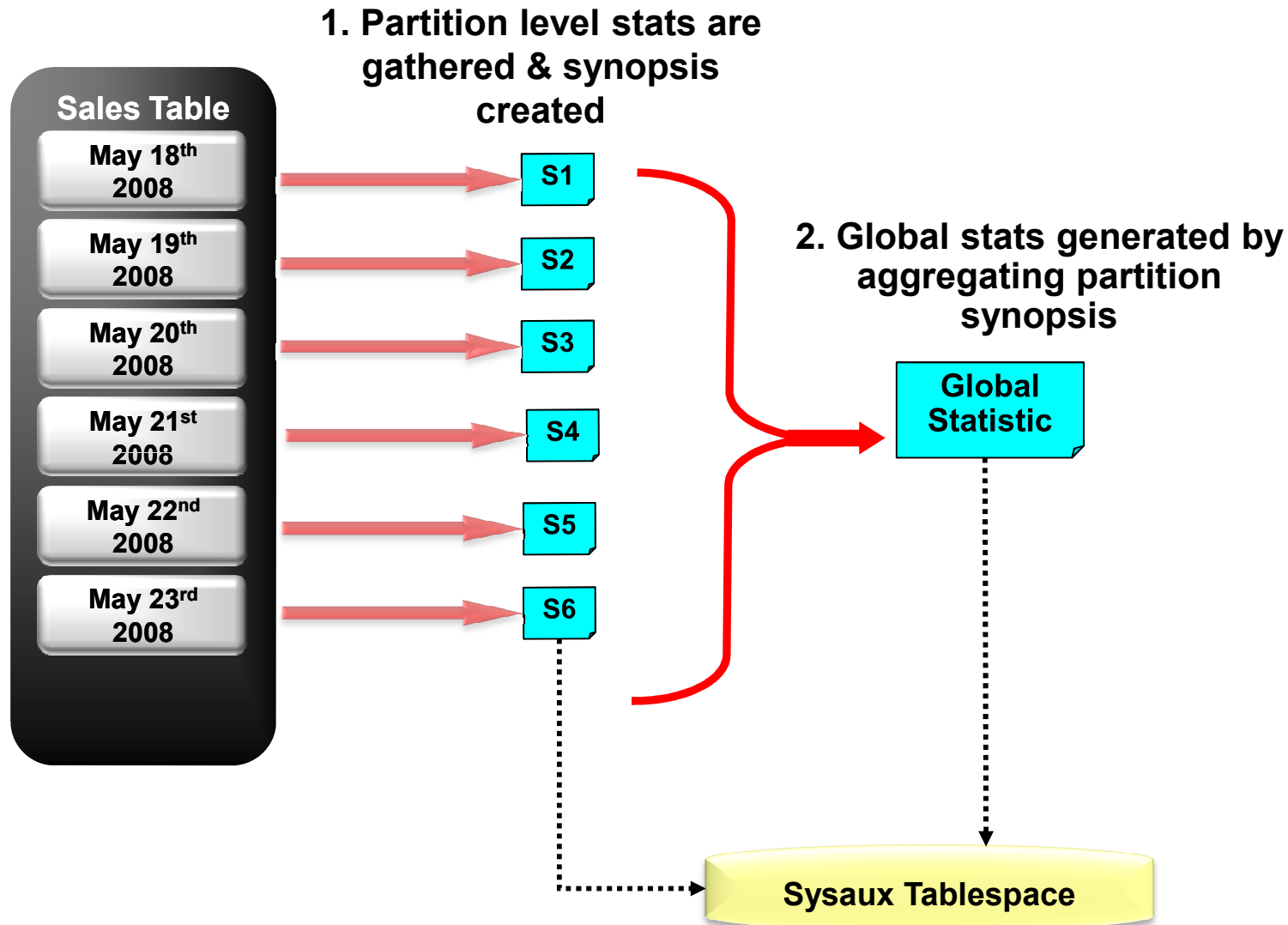
## Statistics Gathering

- By default DBMS\_STATS gathers following stats for each table
  - global (table level)
  - partition level
  - Sub-partition
- Optimizer uses global stats if query touches two or more partitions
- Optimizer uses partition stats if queries do partition elimination and only one partition is necessary to answer the query
  - If queries touch two or more partitions the optimizer will use a combination of global and partition level statistics
- Optimizer uses sub-partition level statistics if your queries do partition elimination and only one sub-partition is necessary to answer query

# Efficiency Statistics Management

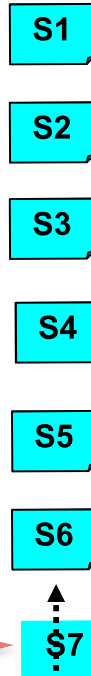
- How do I gather accurate Statistics
  - “ .. Compute statistics gives accurate results but takes too long ..”
  - “ .. Sampling is fast but not always accurate ..”
  - “ .. AUTO SAMPLE SIZE does not always work with data skew ..”
- New groundbreaking implementation for AUTO SAMPLE SIZE
  - Faster than sampling
  - Accuracy comparable to compute statistics
- Gathering statistics on one partition (e.g. after a bulk load) causes a full scan of all partitions to gather global table statistics Extremely time and resource intensive
  - Use incremental statistics
  - Gather statistics for touched partition(s) ONLY
  - Table (global) statistics are built from partition statistics

# Incremental Global Statistics



# Incremental Global Statistics Cont'd

3. A new partition is added to the table & Data is Loaded



6. Global stats generated by aggregating the original partition synopsis with the new one



5. Retrieve synopsis for each of the other partitions from Sysaux





## Step necessary to gather accurate statistics

- Turn on incremental feature for the table

```
EXEC  
  DBMS_STATS.SET_TABLE_PREFS('SH', 'SALES', 'INCREMENTAL', 'TRUE');
```

- After load gather table statistics using GATHER\_TABLE\_STATS command don't need to specify many parameter

```
– EXEC DBMS_STATS.GATHER_TABLE_STATS('SH', 'SALES');
```

- The command will collect statistics for partitions and update the global statistics based on the partition level statistics and synopsis

- Possible to set incremental to true for all tables using

```
– EXEC DBMS_STATS.SET_GLOBAL_PREFS('INCREMENTAL', 'TRUE');
```

## Initialization parameters

Only set what you really need to

Parameter	Value	Comments
compatible	11.1.0.7.0	Needed for Exadata
db_block_size	8 KB	Larger size may help with compression ratio
db_cache_size	5 GB	Large enough to hold metadata
parallel_adaptive_multi_user	False	Can cause unpredictable response times as it is based on concurrency
parallel_execution_message_size	16 KB	Improves parallel server processes communication
parallel_min_servers	64	Avoids query startup costs
parallel_max_servers	128	Prevents systems from being flooded by parallel servers
pga_aggregate_target	18 GB	Tries to keep sorts in memory
shared_pool_size	4 GB	Large enough to for PX communicate and SQL Area

# Using EM to monitor Parallel Query

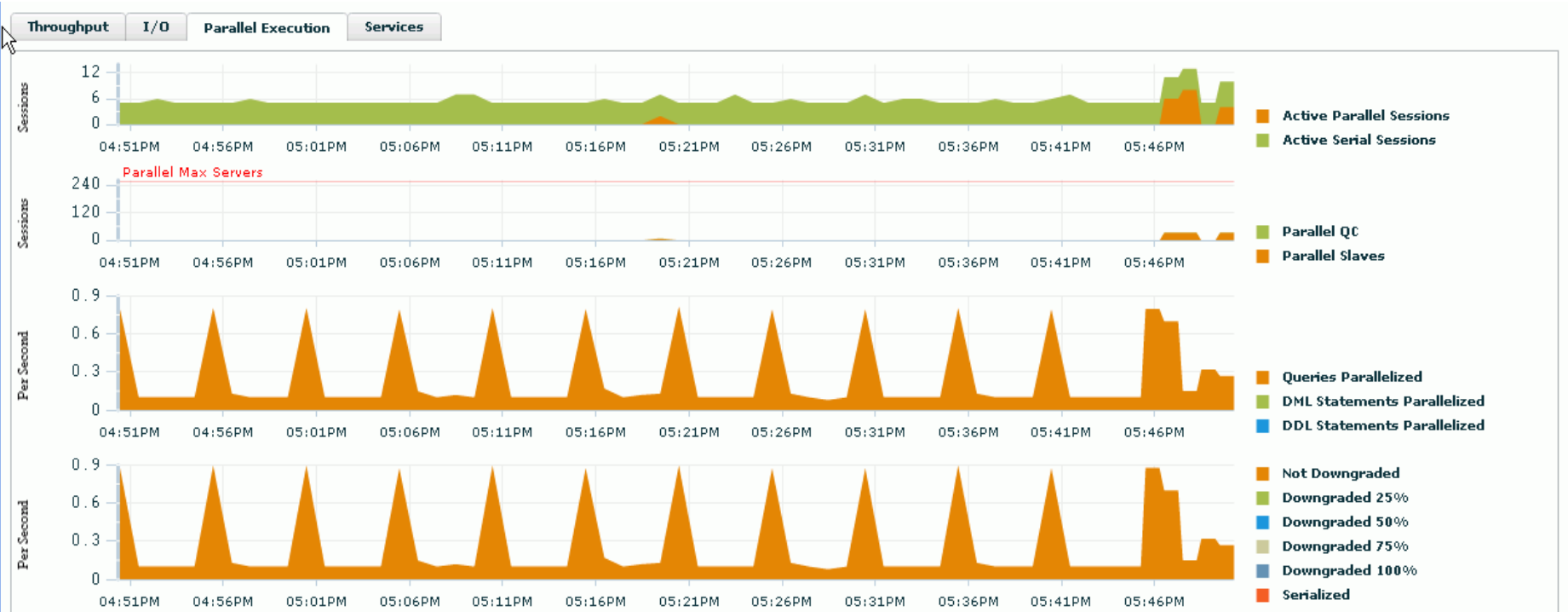
Click on the performance tab

The screenshot displays the Oracle Enterprise Manager 11g Database Control interface for instance DBM\_DBM2. The 'Performance' tab is selected and circled in red. A red arrow points from the text 'Click on the performance tab' to this tab. The interface shows various performance metrics:

- General:** Status is Up, Instance Name is DBM2, Version is 11.1.0.7.0. Includes buttons for Shutdown and Black Out.
- Host CPU:** A bar chart showing CPU usage for 'Other' and 'DBM2'. Load is 1.61 and Paging is 0.00. Maximum CPU is 8.
- Active Sessions:** A bar chart showing session types: Wait, User I/O, and CPU.
- SQL Response Time:** A bar chart comparing 'Latest Collection (seconds)' and 'Reference Collection (seconds)'. SQL Response Time (%) is 216.54.
- Diagnostic Summary:** Interconnect Alerts: 0, ADDM Findings: 4, Period Start Time: Mar 20, 2009 3:00:02 PM PDT, Alert Log: No ORA- errors, Active Incidents: 0.
- Space Summary:** Dump Area Used (%) is 54.
- Alerts:** Category: All, Critical: 0, Warning: 4. A table lists alerts with columns for Severity, Category, Name, Impact, Message, and Alert Triggered.
- Related Alerts:** ADDM Performance Analysis.

The bottom of the interface shows a search bar with 'Compress' entered and navigation buttons (Next, Previous, Highlight all, Match case). The status bar at the bottom right shows 'dscbat01.us.oracle.com:1158'.

# Parallel Execution screens



## Additional Monitoring Links

Top Sessions and Top SQL data from ASH can be found on the Top Activity page.

- [Top Activity](#)
- [Top Consumers](#)
- [Duplicate SQL](#)
- [Blocking Sessions](#)
- [Hang Analysis](#)
- [Instance Locks](#)
- [Instance Activity](#)
- [Search Sessions](#)
- [Search SQL](#)
- [Snapshots](#)
- [AWR Baselines](#)
- [SQL Tuning Sets](#)
- [SQL Performance Analyzer](#)
- [SQL Monitoring](#)

Click on the SQL Monitoring link





# Using EM to monitor Parallel Query

Oracle Enterprise Manager (SYS) - Monitored SQL Executions - Mozilla Firefox

Database Control

Cluster Database: DBM > Database Instance: DBM\_DBM2 > Monitored SQL Executions

Click on a SQL ID to drill down to more details

Shows parallel degree used and number of nodes used in query

Status	Duration	SQL ID	Session	Parallel	Database Time	IO	Start	Ended	SQL Text
✓	1.0s	4v06pbau6gsxq	464	128 7	18.3s	258	04:39:38 PM	04:39:39 PM	select count(*) from my_objects a, my_...
✓	8.0s	94v66p38f1uqt	464	128 7	4.5m	878	04:38:11 PM	04:38:19 PM	select count(*) from my_objects a, my_...
✓	4.0s	8kr80jws34swt	464	128 7	3.4m	552	04:36:18 PM	04:36:22 PM	select count(*) from my_objects
✓	1.0s	17b2v4tywaw6wy	464	128 7	5.6s	791	04:35:59 PM	04:36:00 PM	insert /*+ APPEND PARALLEL */ into my...
✓	7.0s	8kr80jws34swt	464	128 7	3.7m	526	04:35:15 PM	04:35:22 PM	select count(*) from my_objects
✓	8.0s	dfz3xbjuf70jn	468		8.0s		03:49:20 PM	03:49:28 PM	SELECT * FROM DWR_ORG_BSNS_UNIT ...
✓	8.0s	dfz3xbjuf70jn	433		8.0s		03:49:20 PM	03:49:28 PM	SELECT * FROM DWR_ORG_BSNS_UNIT ...
✓	8.0s	dfz3xbjuf70jn	471		8.0s		03:49:20 PM	03:49:28 PM	SELECT * FROM DWR_ORG_BSNS_UNIT ...
✓	8.0s	dfz3xbjuf70jn	465		8.0s	58	03:49:20 PM	03:49:28 PM	SELECT * FROM DWR_ORG_BSNS_UNIT ...
✓	4.0s	17b2v4tywaw6wy	500	128 7	46.2s	454	03:23:51 PM	03:23:55 PM	insert /*+ APPEND PARALLEL */ into my...

Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2008, Oracle. All rights reserved.  
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
[About Oracle Enterprise Manager](#)

# SQL Monitoring Screens - PWJ

Cluster Database: DBM > Database Instance: DBM\_DBM2 > Monitored SQL Executions >

Logged in As SYS

Monitored SQL Execution Details

[Text Report](#)

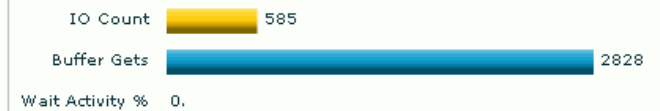
## Overview

SQL ID: 8s5y531zbx81j  
 Parallel: 4  
 Execution Started: Fri Mar 20 2009 05:22:32 PM  
 Last Refresh Time: Fri Mar 20 2009 05:22:32 PM  
 Execution ID: 33554432  
 Session: 451  
 Fetch Calls: 1

## Time



## IO & Wait Statistics



## Details

[Plan Statistics](#) [Parallel](#) [Activity](#)

Plan Hash Value: 935085667

Operation	Name	Estimate...	Cost	Timeline(1s)	Exec...	Actual...	Memo...	Temp ...	CPU Activity %	Wait Activity %
SELECT STATEMENT			303		1	1				
SORT AGGREGATE		1			1	1				
PX COORDINATOR					5	4				
PX SEND QC (RANDOM)	:TQ10000	1			4	4				
SORT AGGREGATE		1			4	4				
PX PARTITION HASH ALL		1735K	303		4	1725K				
HASH JOIN		1735K	303		4	1725K	31M			
TABLE ACCESS	MY_OBJECTS2	338K	150		4	346K				
TABLE ACCESS	MY_OBJECTS2	338K	150		4	346K				

Only one set of parallel servers

# Using EM to monitor Parallel Query

Cluster Database: DBM > Database Instance: DBM\_DBM2 > Monitored SQL Executions > Monitored SQL Execution Details

Logged in As SYS [Text Report](#)

**Overview**

SQL ID: 94v66p38f1uqt

Parallel: 128 7

Execution Started: Fri Mar 20 2009 04:38:11 PM

Last Refresh Time: Fri Mar 20 2009 04:38:19 PM

Execution ID: 33554432

Session: 464

Fetch Calls: 1

**Time**

Duration: 9.0s

Database Time: 4.5m

PL/SQL & Java: 0.0s

**IO & Wait Statistics**

IO Count: 878

Buffer Gets: 41K

Wait Activity %: 100

**Details**

Plan Statistics Parallel Activity

Plan Hash Value: 1101229021

Operation	Name	Estimate...	Cost	Timeline(9s)	Exec...	Actual...	Memo...	Temp ...	CPU Activity %	Wait Activity %
SELECT STATEMENT			13		1	1				
SORT AGGREGATE		1			1	1				
PX COORDINATOR					225	128			1.18	6.67
PX SEND QC (RANDOM)	:TQ10002	1			112	112				
SORT AGGREGATE		1			112	112	150M			
HASH JOIN		2298K	13		112	967K				
PX RECEIVE		329K	6		112	242K			0.39	
PX SEND HASH	:TQ10000	329K	6		112	247K			0.39	
PX BLOCK ITERATOR		329K	6		112	247K				
TABLE ACCESS	MY_OBJECTS	329K	6		1165	247K			98	93
PX RECEIVE		329K	6		112	242K				
PX SEND HASH	:TQ10001	329K	6		112	239K				
PX BLOCK ITERATOR		329K	6		112	239K				
TABLE ACCESS	MY_OBJECTS	329K	6		1134	239K				

Coordinator

Consumers

Producers

# SQL Monitoring screens

## Monitored SQL Execution Details

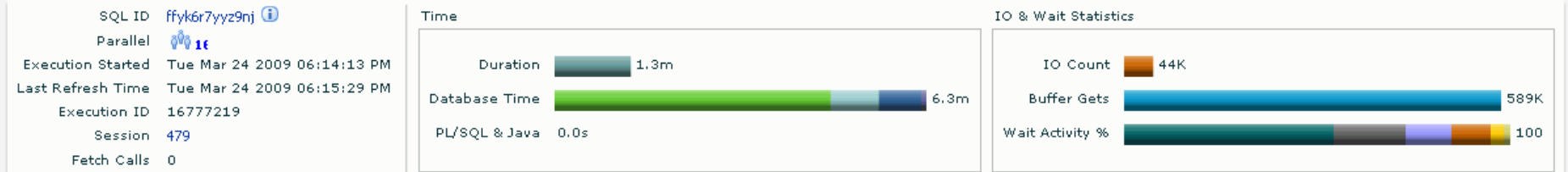
Text Report

Refresh

5 seconds

Stop Refresh

### Overview



### Details

Click on parallel tab to get more info on PQ

Plan Hash Value 3913711993

Operation	Name	Estimate...	Cost	Timeline(77s)	Exec...	Actual...	Memory	Temp	CPU Activity %	Wait Activity %
CREATE TABLE STATEMENT			30K		33					
PX COORDINATOR					33				0.40	
PX SEND QC (RANDOM)	:TQ10001	16K	3		16					
LOAD AS SELECT					16		1209M		64	32
PX RECEIVE		16K	3		16	14M			11	
PX SEND RANDOM LOCAL	:TQ10000	16K	3		16	14M			5.58	12
PX BLOCK ITERATOR		16K	3		16	14M				
EXTERNAL TABLE ACCESS FULL	STORE_SALES_ET	16K	3		196	14M			16	54

➡ The green arrow indicates which line in the execution plan is currently being worked on

# SQL Monitoring Screens

Details

Plan Statistics Parallel Activity

By clicking on the + tab you can get more detail about what each individual parallel server is doing. You want to check each slave is doing an equal amount of work

Parallel Server	Database Time	Wait Activity %	IO Count	Buffer Gets
All Parallel Servers				
Parallel Coordinator	8.4s	2.04		264K
Parallel Set 1				
Parallel Server 1 (p000)	3.4s		36	8413
Parallel Server 2 (p001)	29.0s	2.04	4768	19K
Parallel Server 3 (p002)	3.4s	4.08		8069
Parallel Server 4 (p003)	6.6s	4.08	1107	11K
Parallel Server 5 (p004)	3.7s	2.04	108	9342
Parallel Server 6 (p005)	3.5s	4.08		8016
Parallel Server 7 (p006)	6.4s		1062	11K
Parallel Server 8 (p007)	5.0s		90	9359
Parallel Server 9 (p008)	13.3s	2.04	3985	18K
Parallel Server 10 (p009)	3.4s	2.04	36	8296
Parallel Server 11 (p010)	16.6s		4793	19K
Parallel Server 12 (p011)	3.5s	2.04		8069
Parallel Server 13 (p012)	6.5s	4.08	1107	11K
Parallel Server 14 (p013)	3.6s		108	9823
Parallel Server 15 (p014)	3.3s	2.04		8016
Parallel Server 16 (p015)	6.4s	2.04	1062	10K
Parallel Set 2	1.2m	67		90K

# Disk Configuration with ASM

ORACLE Enterprise Manager 11g Database Control

Automatic Storage Management: +ASM1\_dscbas01.us.oracle.com

Disk Group: DATA

General Performance Templates Files

**General**

Name: **DATA**  
 State: **MOUNTED**  
 Redundancy: **NORMAL**  
 Total Size (GB): **15,600**  
 Pending Operations: 0

**Advanced Attributes** Edit

Database Compatibility: **11.1.0.7.0**  
 ASM Compatibility: **11.1.0.7.0**  
 Disk Repair Time (Hours): **3.6**

**Current Disk Group Usage (GB)**

Category	Usage (GB)	Percentage
Free	12,954.29	83%
Internal	2.33	0%
ORCL	2,643.38	17%

**Disk Group Daily Space Usage History (Last 7 Days)**

Date	Usage (GB)
13	2,400
14	2,500
15	2,600
16	2,650
17	2,650
18	2,650
19	2,650

**Member Disks**

View: By Disk Go Add

Resize Online Offline Recover Bad Blocks Remove Previous 1-25 of 156 Next 25

Select	Disk	Failure Group	Path	Read/Write Errors	State	Mode	Size (GB)	Used (GB)	Used (%)
<input type="checkbox"/>	DATA_CD_10_DSCBAS01S	DSCBAS01S	o/192.168.72.102/DATA_CD_10_dscbas01s	0	NORMAL	ONLINE	100.00	16.98	16.98
<input type="checkbox"/>	DATA_CD_10_DSCBAS02S	DSCBAS02S	o/192.168.72.103/DATA_CD_10_dscbas02s	0	NORMAL	ONLINE	100.00	16.94	16.94
<input type="checkbox"/>	DATA_CD_10_DSCBAS03S	DSCBAS03S	o/192.168.72.104/DATA_CD_10_dscbas03s	0	NORMAL	ONLINE	100.00	16.97	16.97
<input type="checkbox"/>	DATA_CD_10_DSCBAS04S	DSCBAS04S	o/192.168.72.105/DATA_CD_10_dscbas04s	0	NORMAL	ONLINE	100.00	16.96	16.96
<input type="checkbox"/>	DATA_CD_10_DSCBAS05S	DSCBAS05S	o/192.168.72.106/DATA_CD_10_dscbas05s	0	NORMAL	ONLINE	100.00	16.98	16.98
<input type="checkbox"/>	DATA_CD_10_DSCBAS06S	DSCBAS06S	o/192.168.72.107/DATA_CD_10_dscbas06s	0	NORMAL	ONLINE	100.00	16.94	16.94
<input type="checkbox"/>	DATA_CD_10_DSCBAS07S	DSCBAS07S	o/192.168.72.108/DATA_CD_10_dscbas07s	0	NORMAL	ONLINE	100.00	16.95	16.95
<input type="checkbox"/>	DATA_CD_10_DSCBAS08S	DSCBAS08S	o/192.168.72.109/DATA_CD_10_dscbas08s	0	NORMAL	ONLINE	100.00	16.96	16.96
<input type="checkbox"/>	DATA_CD_10_DSCBAS09S	DSCBAS09S	o/192.168.72.110/DATA_CD_10_dscbas09s	0	NORMAL	ONLINE	100.00	16.97	16.97



**For More Information**

search.oracle.com

**Best Practices for Data Warehousing**



**or**

[http://www.oracle.com/technology/products/bi/db/11g/pdf/twp\\_dw\\_best\\_practies\\_11g11\\_2008\\_09.pdf](http://www.oracle.com/technology/products/bi/db/11g/pdf/twp_dw_best_practies_11g11_2008_09.pdf)

# Exadata Sessions

Date	Time	Room	Session Title
<b>Mon</b> 10/12	5:30 PM	Moscone South 307	<b>S311436</b> - Implement Best Practices for Extreme Performance with Oracle Data Warehouses.
<b>Tue</b> 10/13	11:30 AM	Moscone South 307	<b>S311385</b> - Extreme Backup and Recovery on the Oracle Database Machine.
Tue 10/13	1:00 PM	Moscone South 307	<b>S311437</b> - Achieve Extreme Performance with Oracle Exadata and Oracle Database Machine.
Tue 10/13	1:00 PM	Moscone South Room 102	<b>S311358</b> - Oracle's Hybrid Columnar Compression: The Next-Generation Compression Technology
Tue 10/13	2:30 PM	Moscone South 102	<b>S311386</b> - Customer Panel 1: Exadata Storage and Oracle Database Machine Deployments.
Tue 10/13	4:00 PM	Moscone South 102	<b>S311387</b> - Top 10 Lessons Learned Implementing Oracle and Oracle Database Machine.
Tue 10/13	5:30 PM	Moscone South 308	<b>S311420</b> - Extreme Performance with Oracle Database 11g and In-Memory Parallel Execution.
Tue 10/13	5:30 PM	Moscone South Room 104	<b>S311239</b> - The Terabyte Hour with the Real-World Performance Group
Tue 10/13	5:30 PM	Moscone South 252	<b>S310048</b> - Oracle Beehive and Oracle Exadata: The Perfect Match.
<b>Wed</b> 10/14	4:00 PM	Moscone South 102	<b>S311387</b> - Top 10 Lessons Learned Implementing Oracle and Oracle Database Machine.
Wed 10/14	5:00 PM	Moscone South 104	<b>S311383</b> - Next-Generation Oracle Exadata and Oracle Database Machine: The Future Is Now.
<b>Thu</b> 10/15	12:00 PM	Moscone South 307	<b>S311511</b> - Technical Deep Dive: Next-Generation Oracle Exadata Storage Server and Oracle Database Machine





ORACLE®