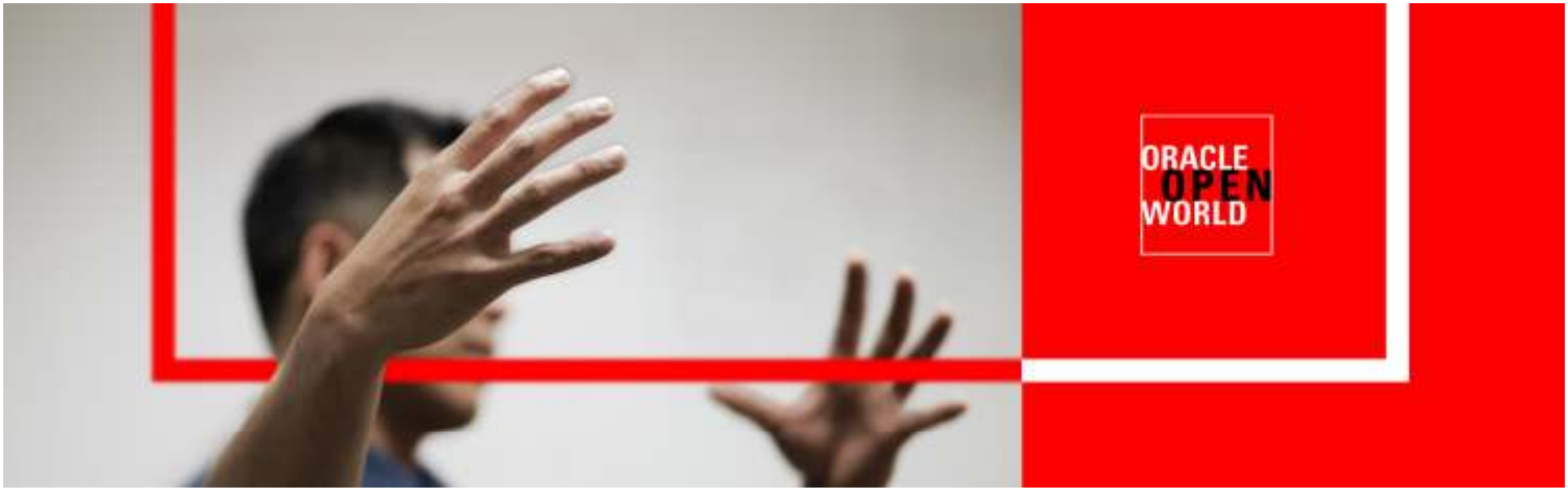

ORACLE®

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.



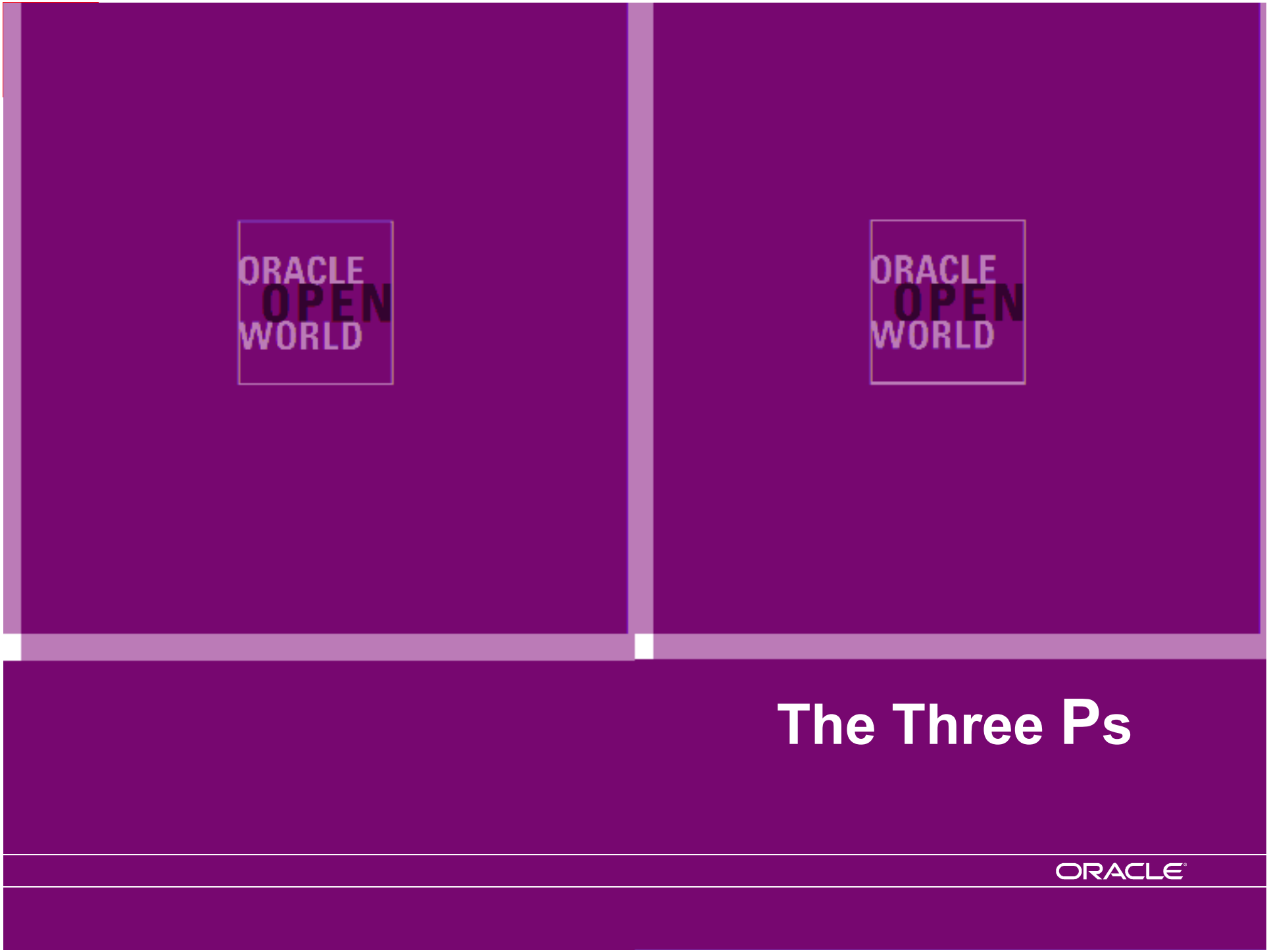
ORACLE

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

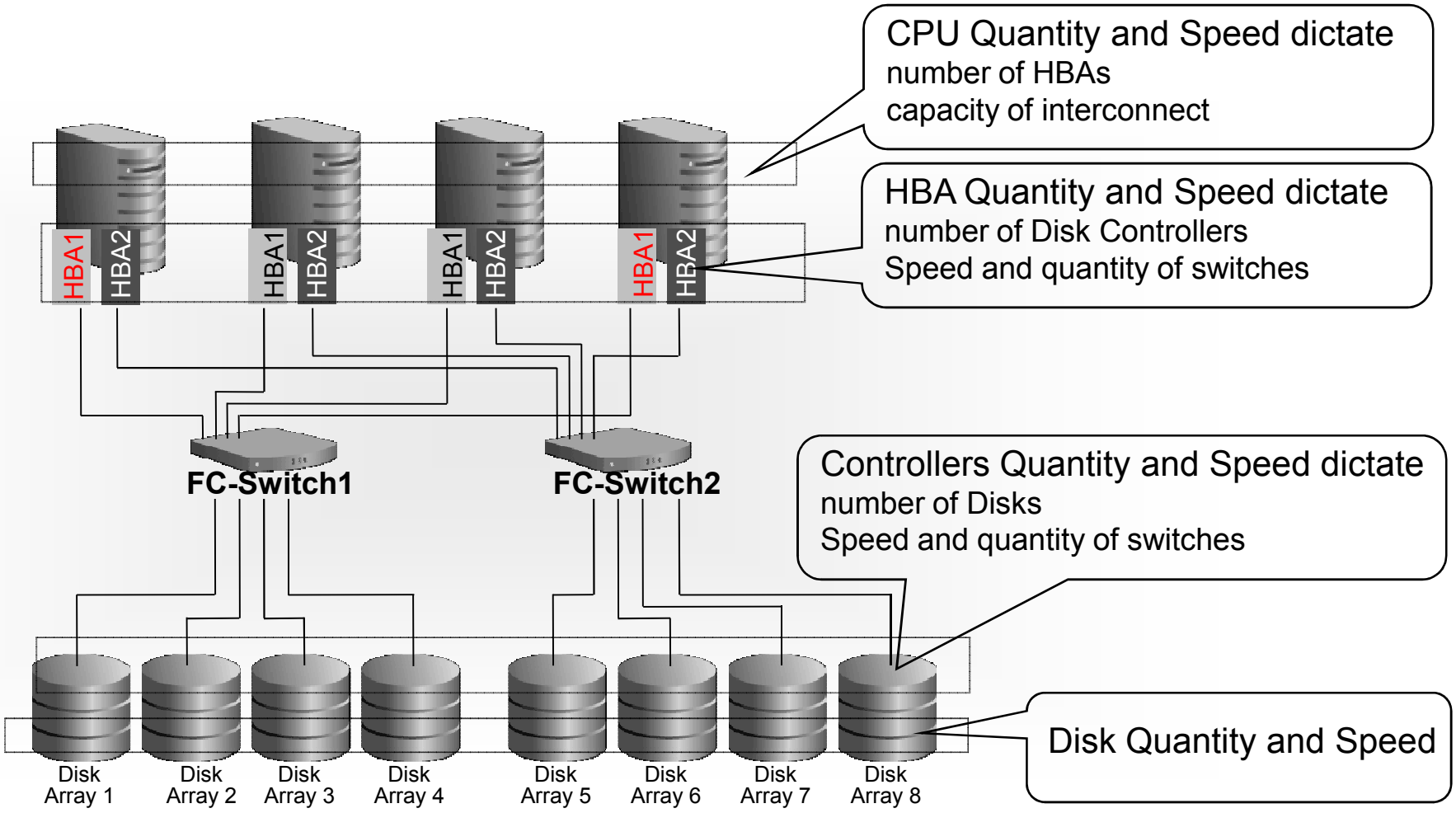

ORACLE
OPEN
WORLD
ORACLE
OPEN
WORLD
The Three Ps
ORACLE

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

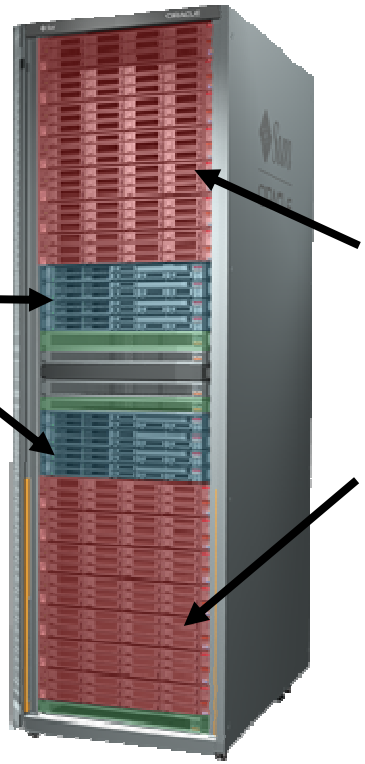
Extreme Performance

RAC Database Server Grid

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

InfiniBand Network

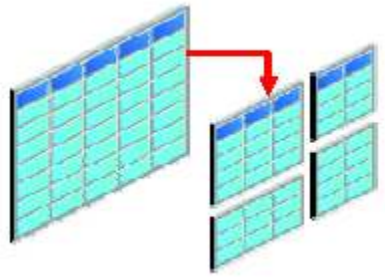
- 3 36-port Infiniband
- 880 Gb/sec aggregate throughput



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

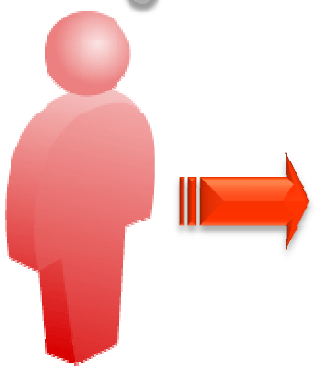
Partitioning



- 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

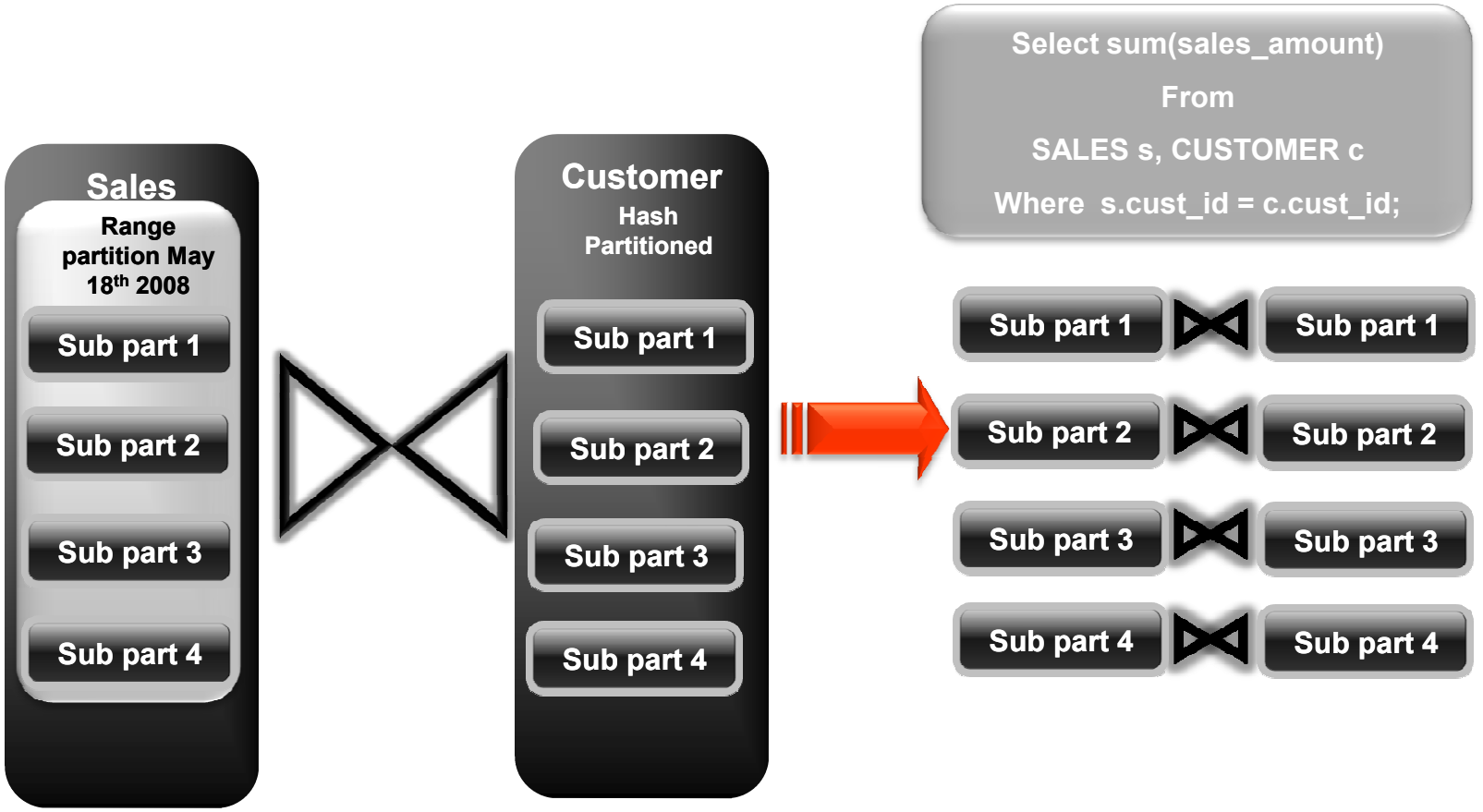
Q: What was the total sales for the weekend of May 20 - 22 2008?



```
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

Partition Wise join



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

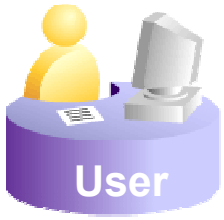
A large join is divided into multiple smaller joins, each joins a pair of partitions in parallel

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

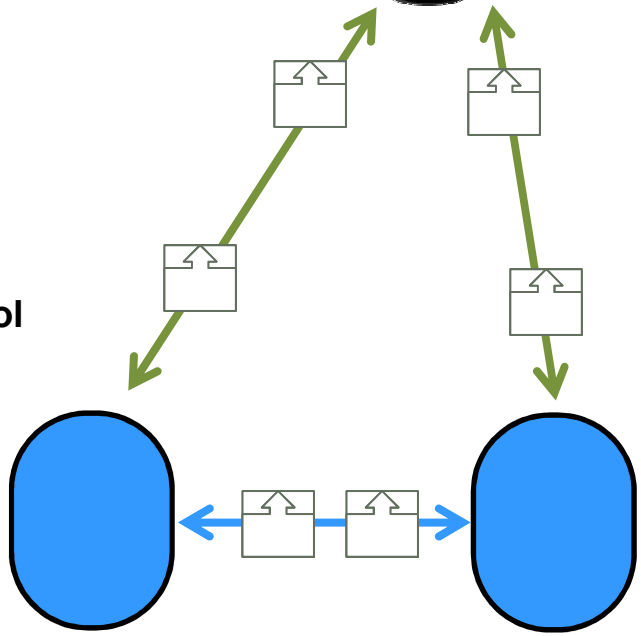


User connects to the database

Background process is spawned

When user issues a parallel SQL statement the background process becomes the Query Coordinator

Parallel servers communicate among themselves & the QC using messages that are passed via memory buffers in the shared pool



Parallel servers - individual sessions that perform work in parallel Allocated from a pool of globally available parallel server processes & assigned to a given operation

Parallel Execution Plan

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

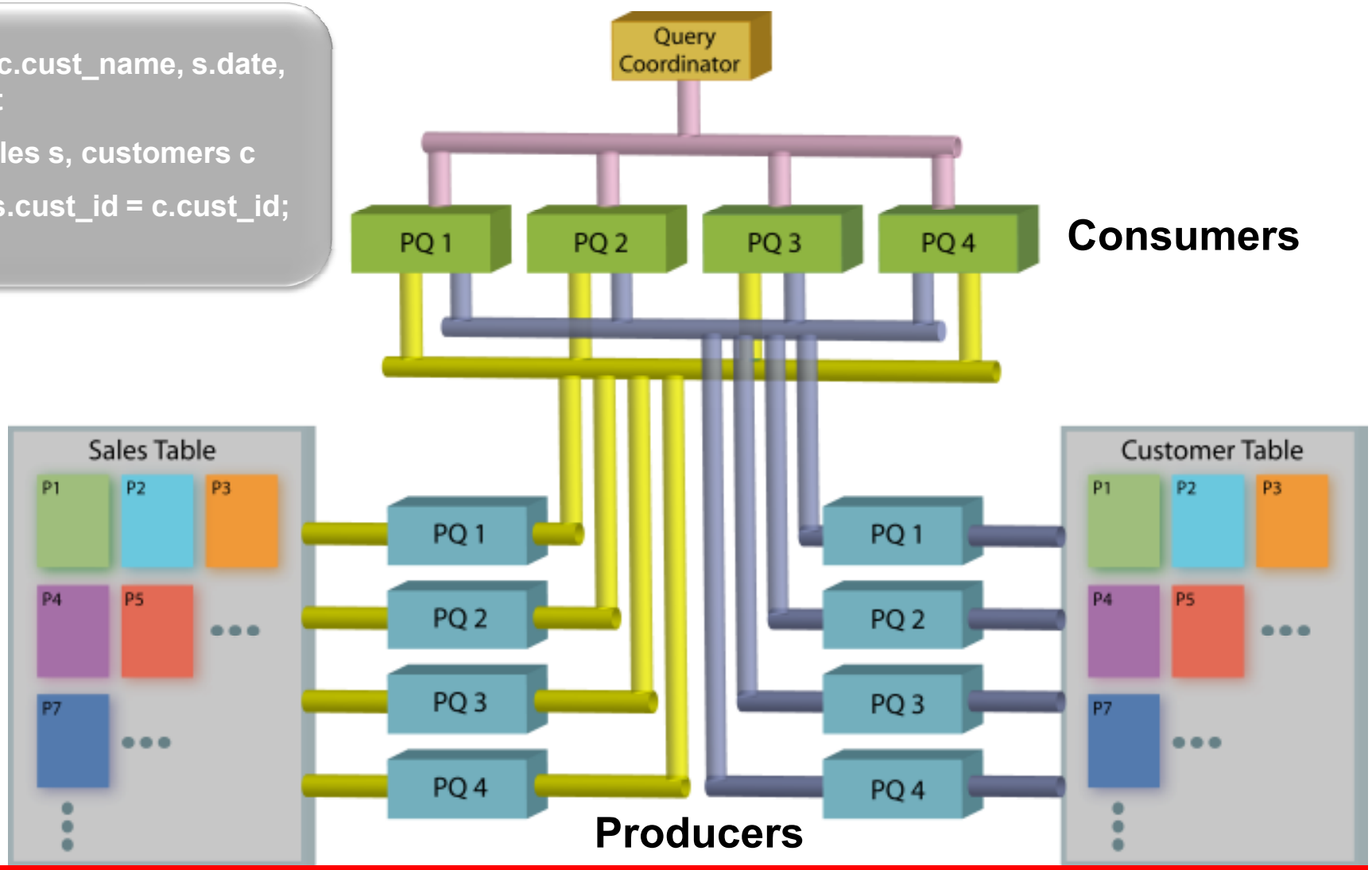
Query Coordinator

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	

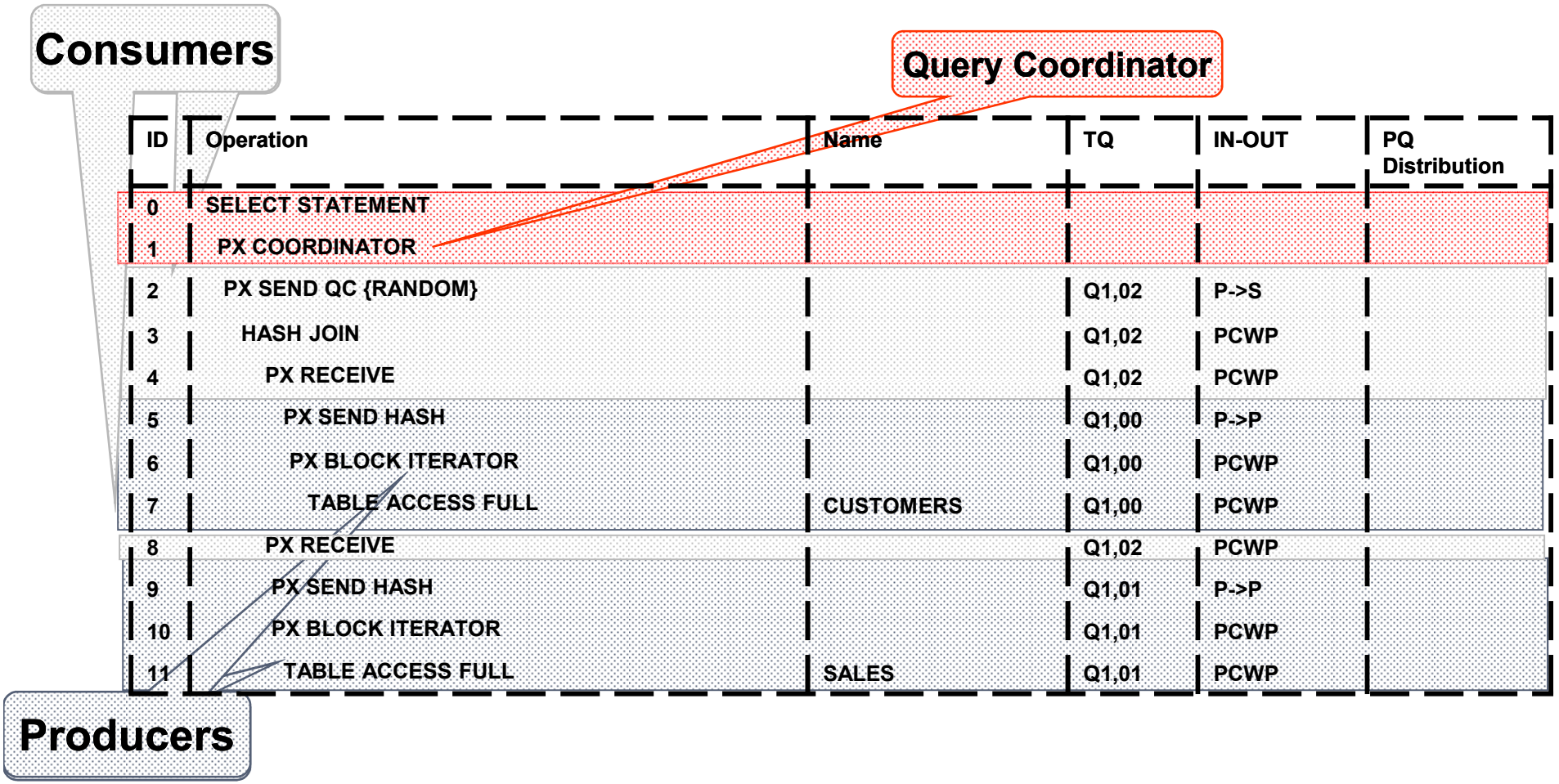
Parallel Servers do majority of the work

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;
```



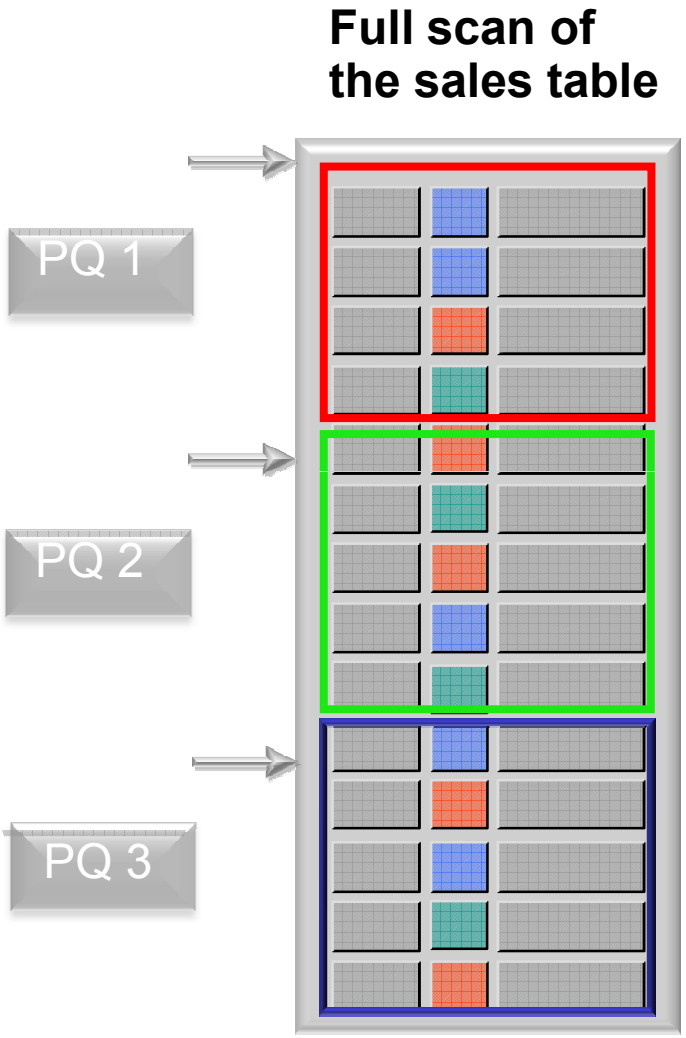
Producers and Consumer in the execution plan



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	

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	PX BLOCK ITERATOR		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	

redicate 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	PX PARTITION RANGE ALL		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	

redicate Information (identified by operation id):

8 - access("CUST_ID">=22300 AND "CUST_ID"<=22810)

Controlling Parallel Execution on RAC

ORACLE DATABASE 11g

1. Use RAC Services

Create two services

Srvctl add service –d database_name

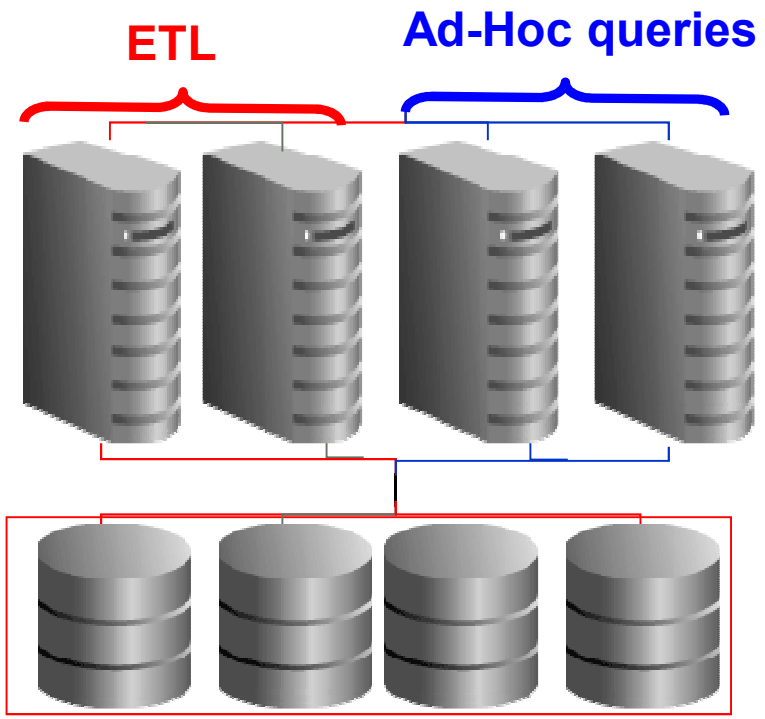
-s ETL

-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

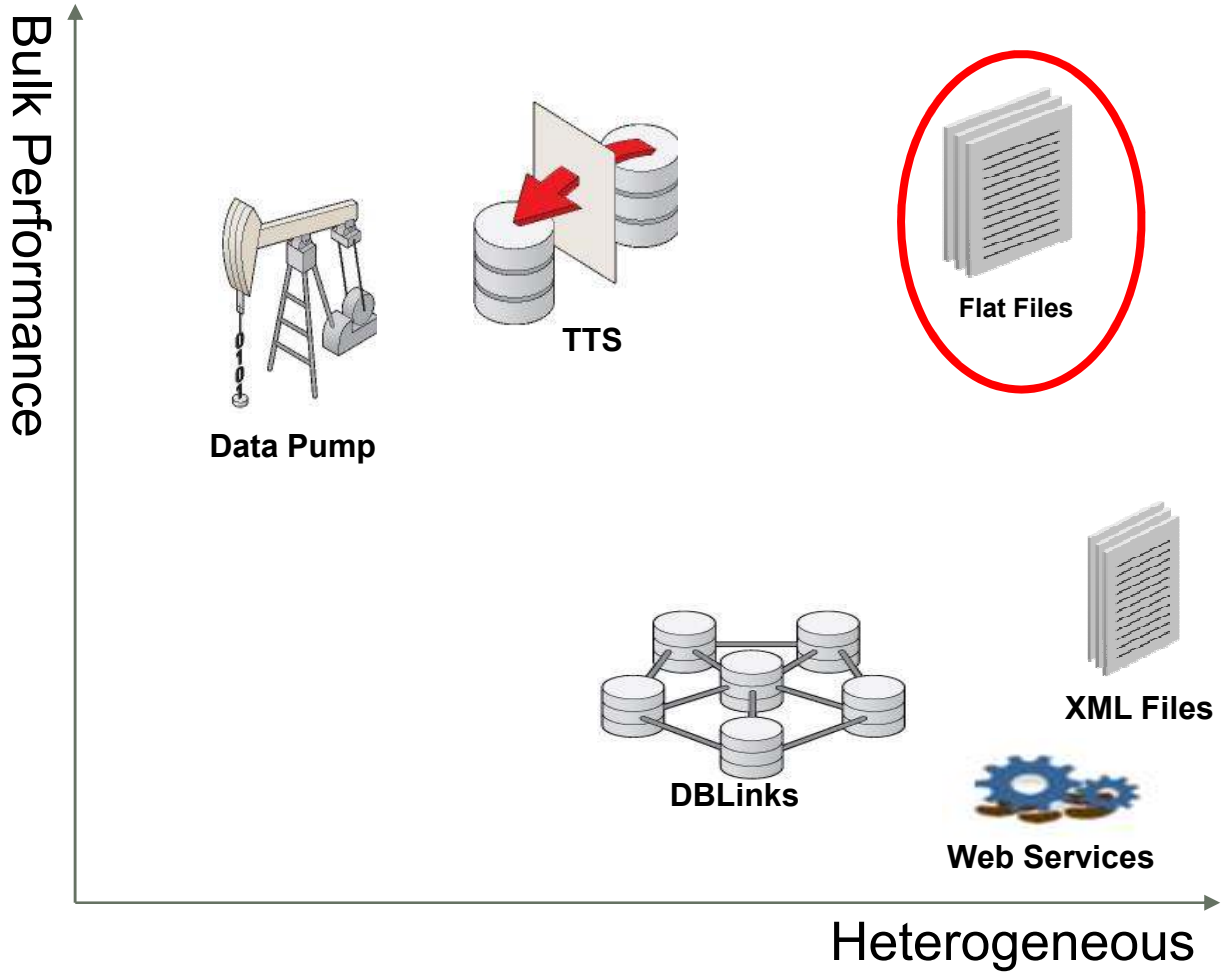
- Parallel execution provides performance boost but requires more resources
- General rules of thumb for determining the appropriate DOP
  - objects smaller than 200 MB should not use any parallelism
  - objects between 200 MB and 5GB should use a DOP of 4
  - objects beyond 5GB use a DOP of 32

Mileage may vary depending on concurrent workload and hardware configuration

Data Loading

ORACLE

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;
```

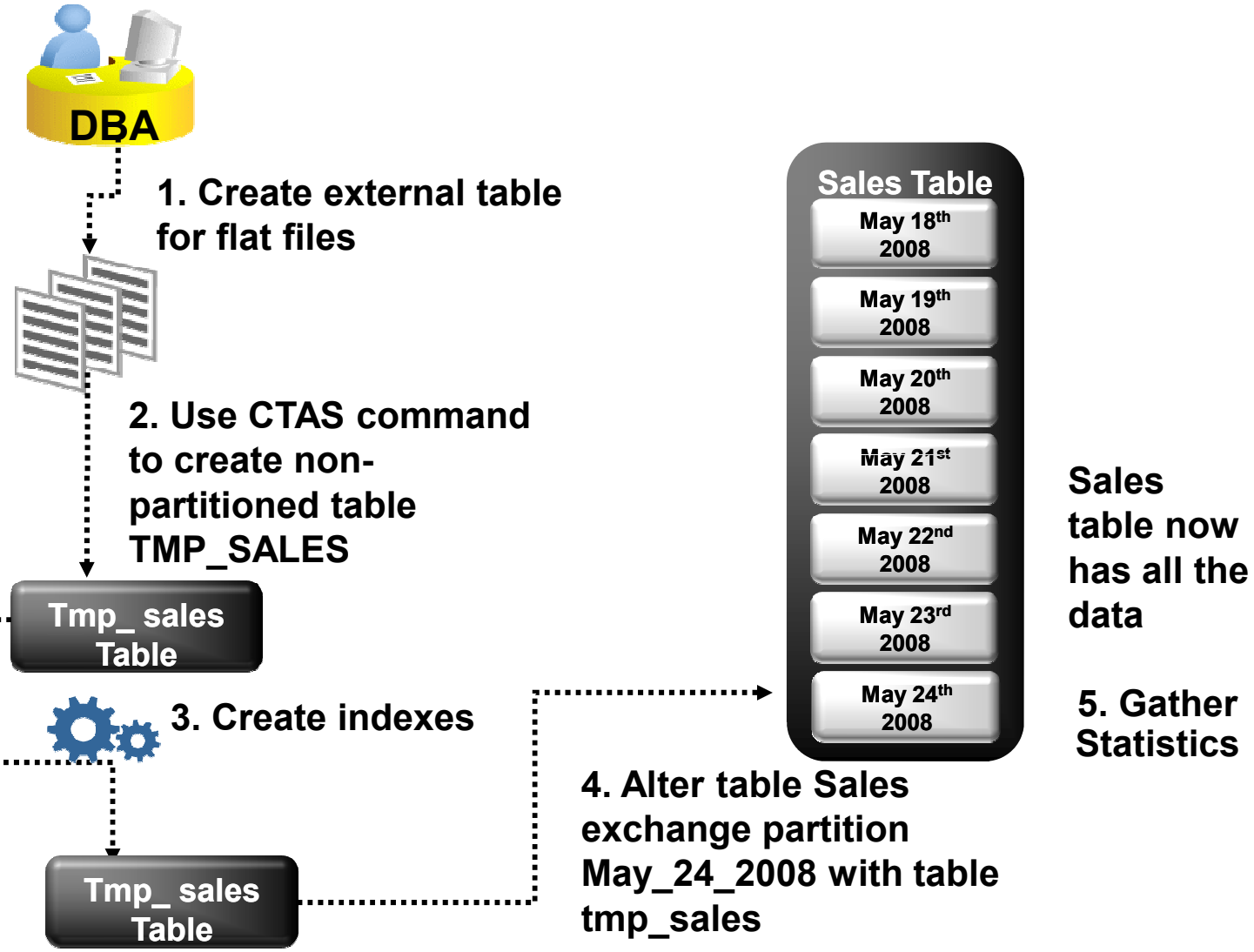
- Only one direct path operation can occur on an object
  - By specifying a specific partition name in the table you can do multiple concurrent direct path loads into a partitioned table

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 DATABASE 11g

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

ORACLE DATABASE 11g

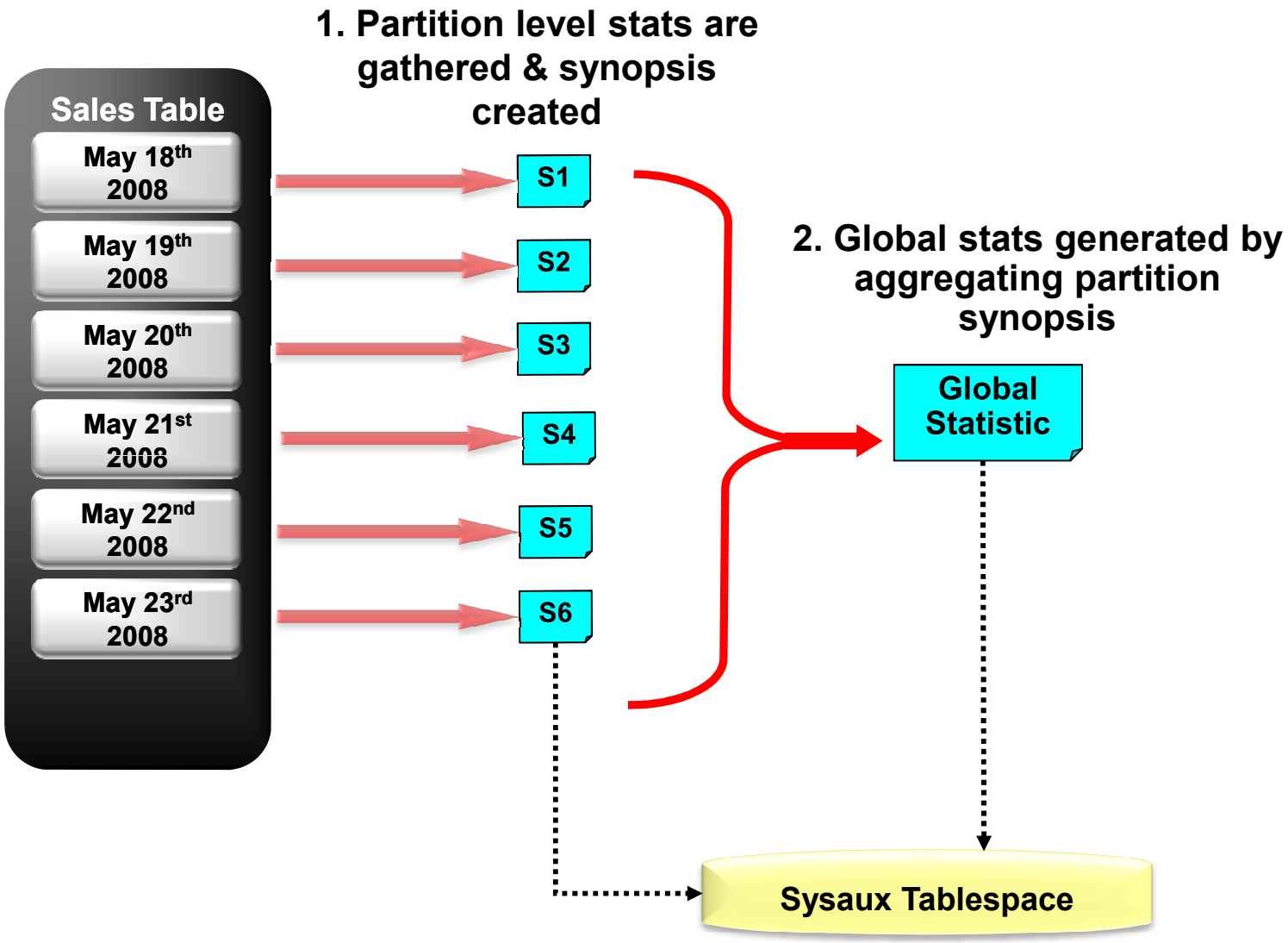
- How do I gather accurate Statistics
  - *" .. Compute statistics gives accurate results but takes too long .."*
  - *" .. Sampling is fast but not always accurate .."*
  - *" .. AUTO SAMPLE SIZE does not always work with data skew .."*

•New groundbreaking implementation for AUTO SAMPLE SIZE
•Faster than sampling
•Accuracy comparable to compute statistics

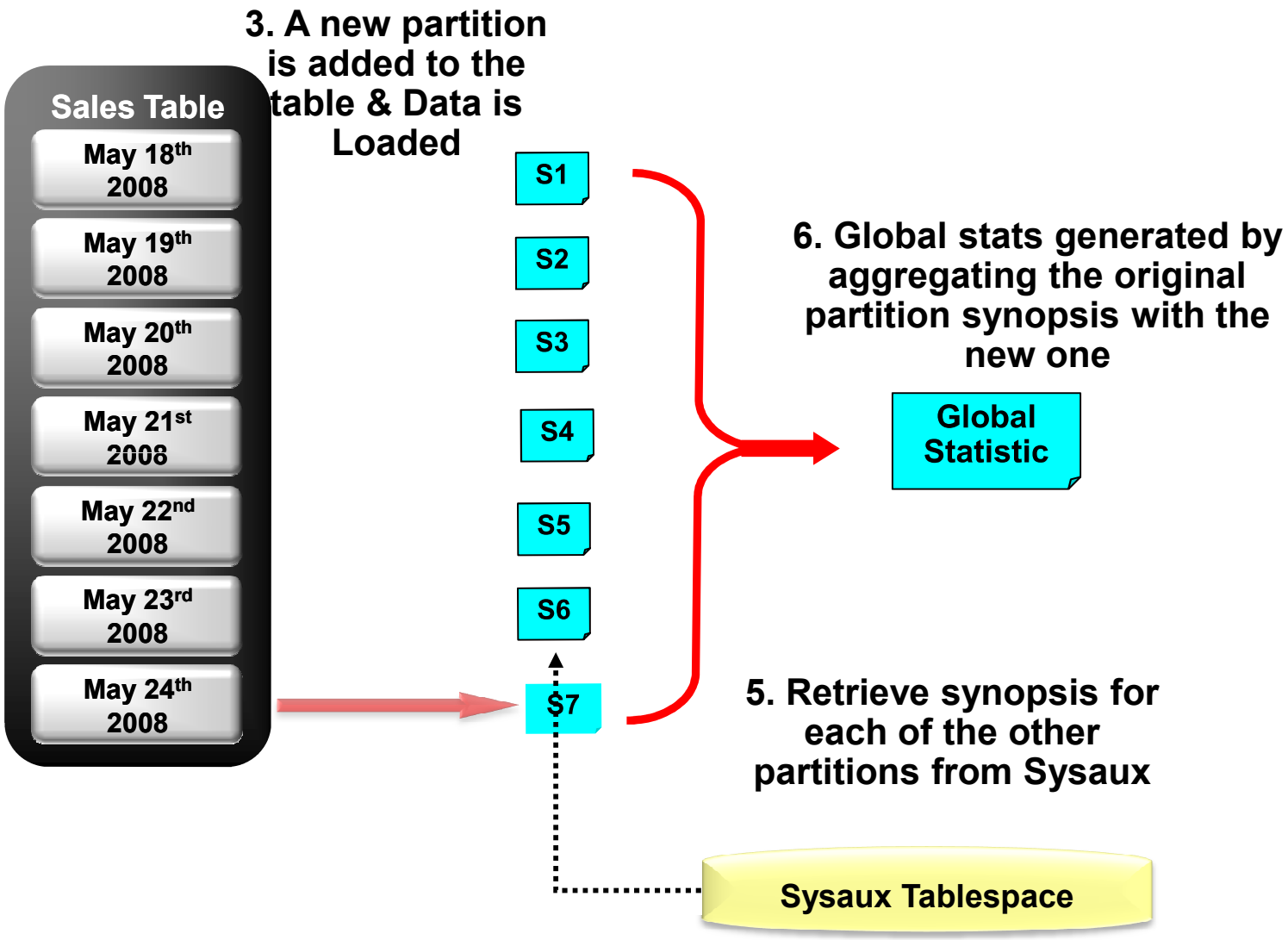
- Gathering statistics on one partition (e.g. after a bulk load) causes a full scan of all partitions to gather global table statistics Extremely time and resource intensive

•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



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');`

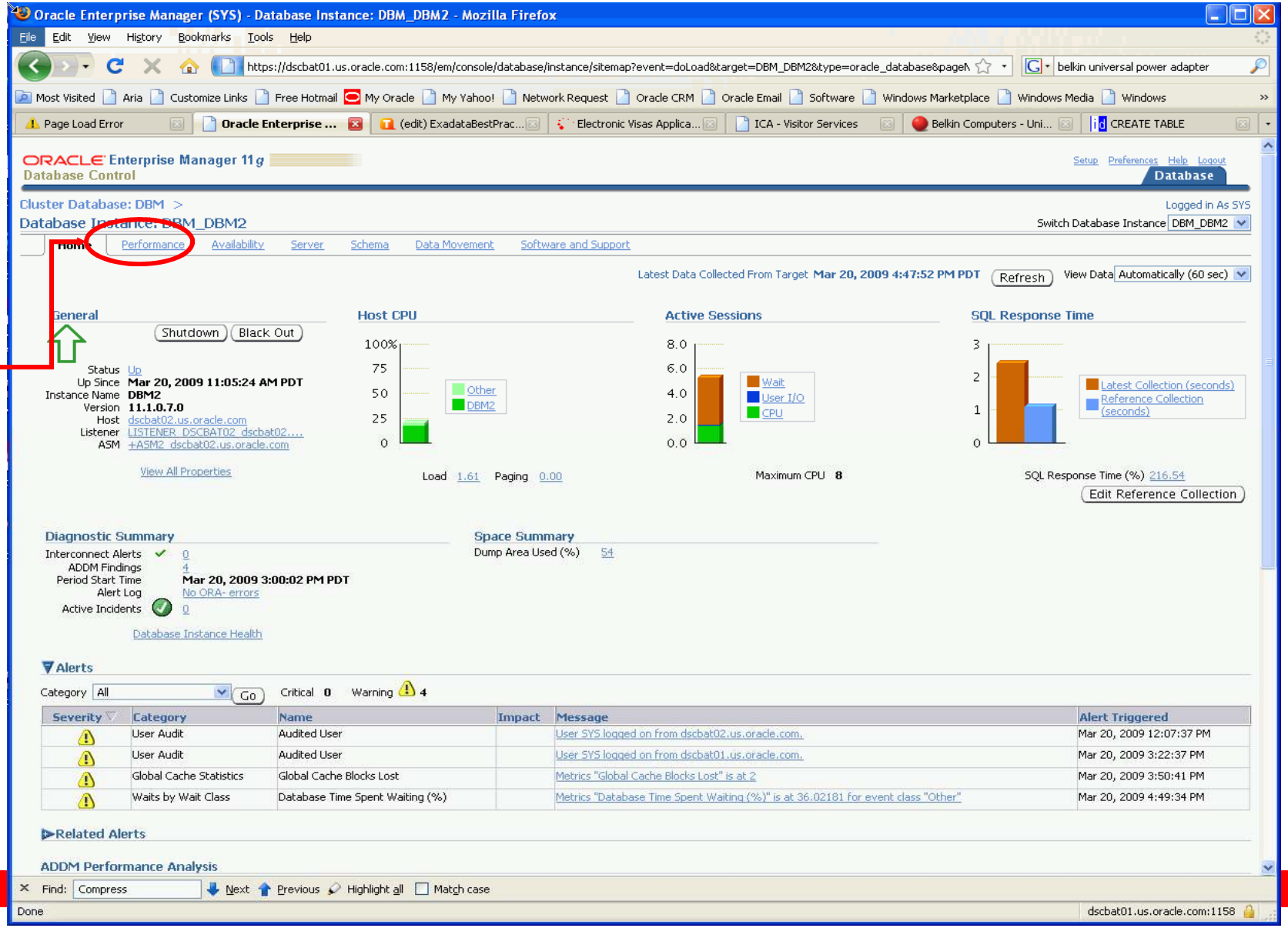
Initialization parameters

Only set what you really need to

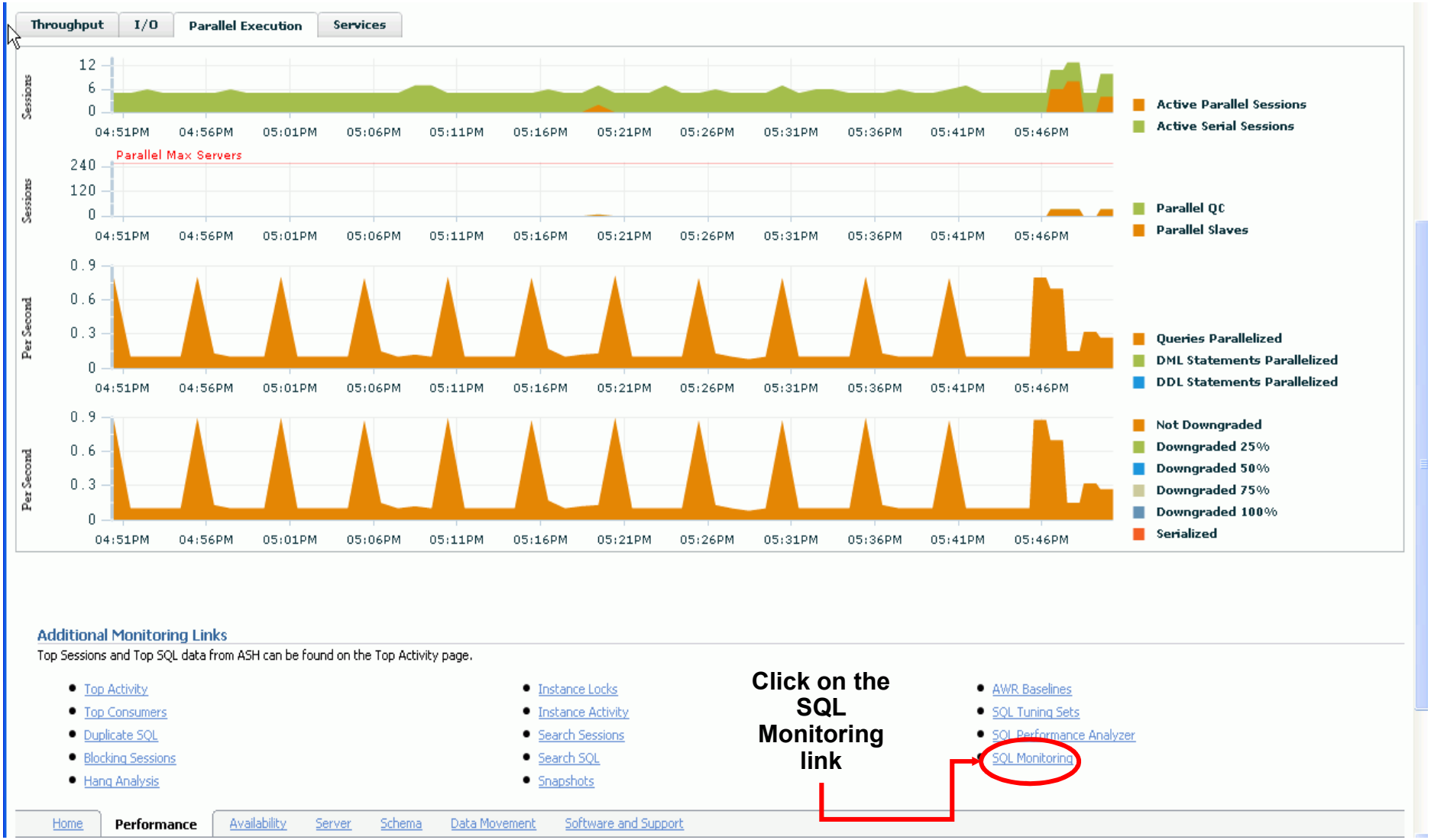
Parameter	Value	Comments
compatible	11.1.0.7.0	Needed for Exadata
db_block_size	8 KB	Larger size may help with compression ratio
db_cache_size	5 GB	Large enough to hold metadata
parallel_adaptive_multi_user	False	Can cause unpredictable response times as it is based on concurrency
parallel_execution_message_size	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

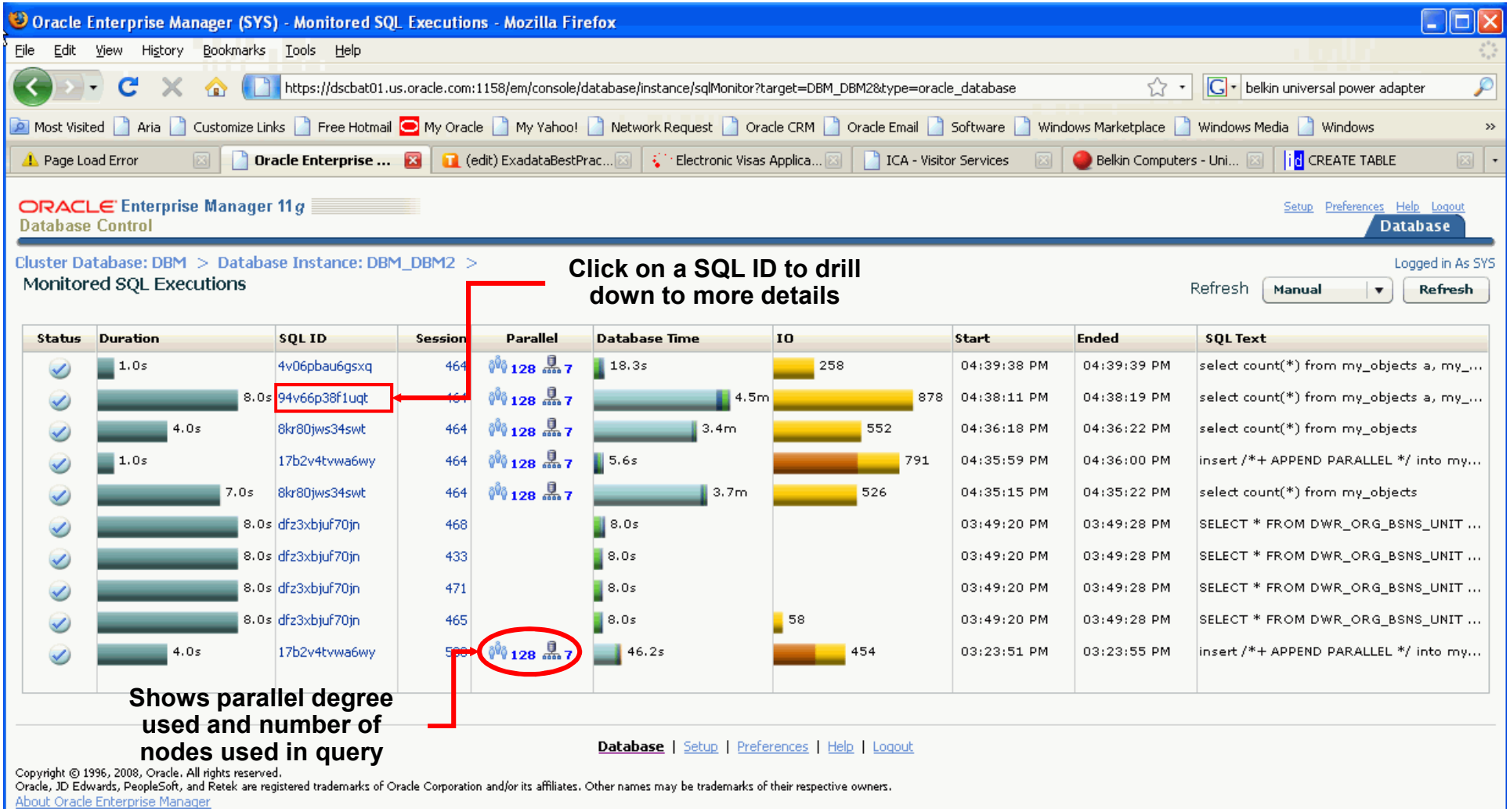
Click on the performance tab



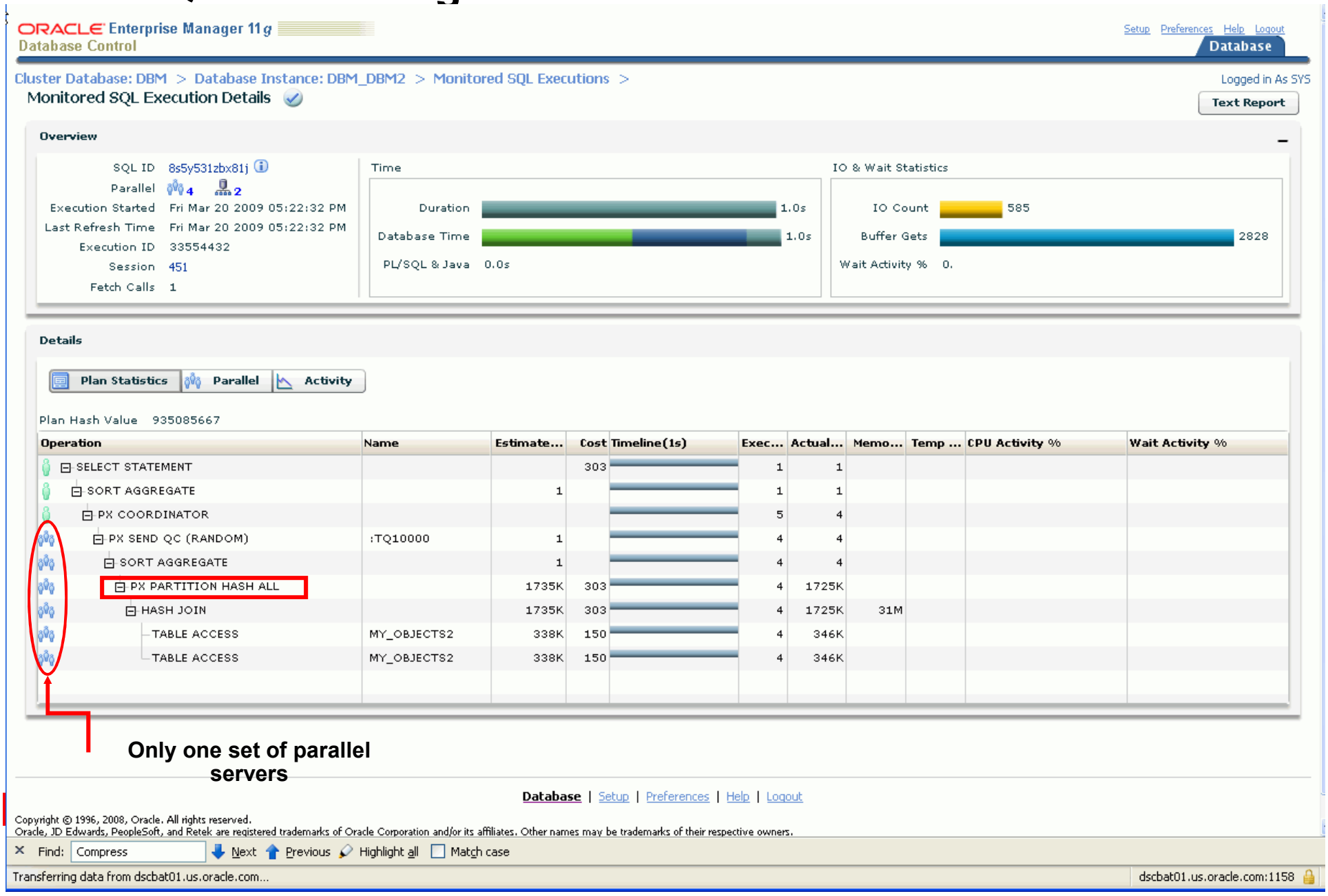
Parallel Execution screens



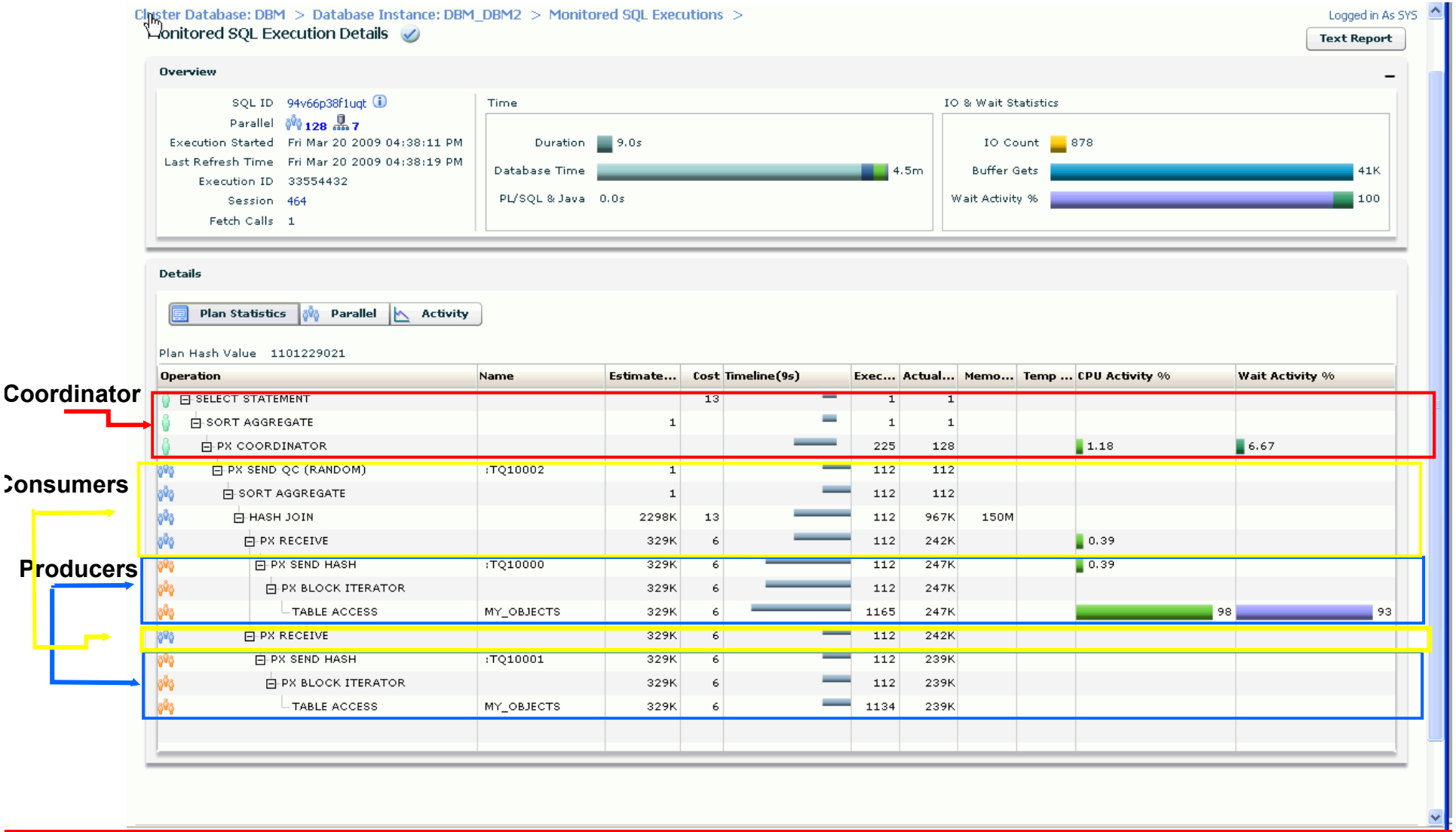
Using EM to monitor Parallel Query



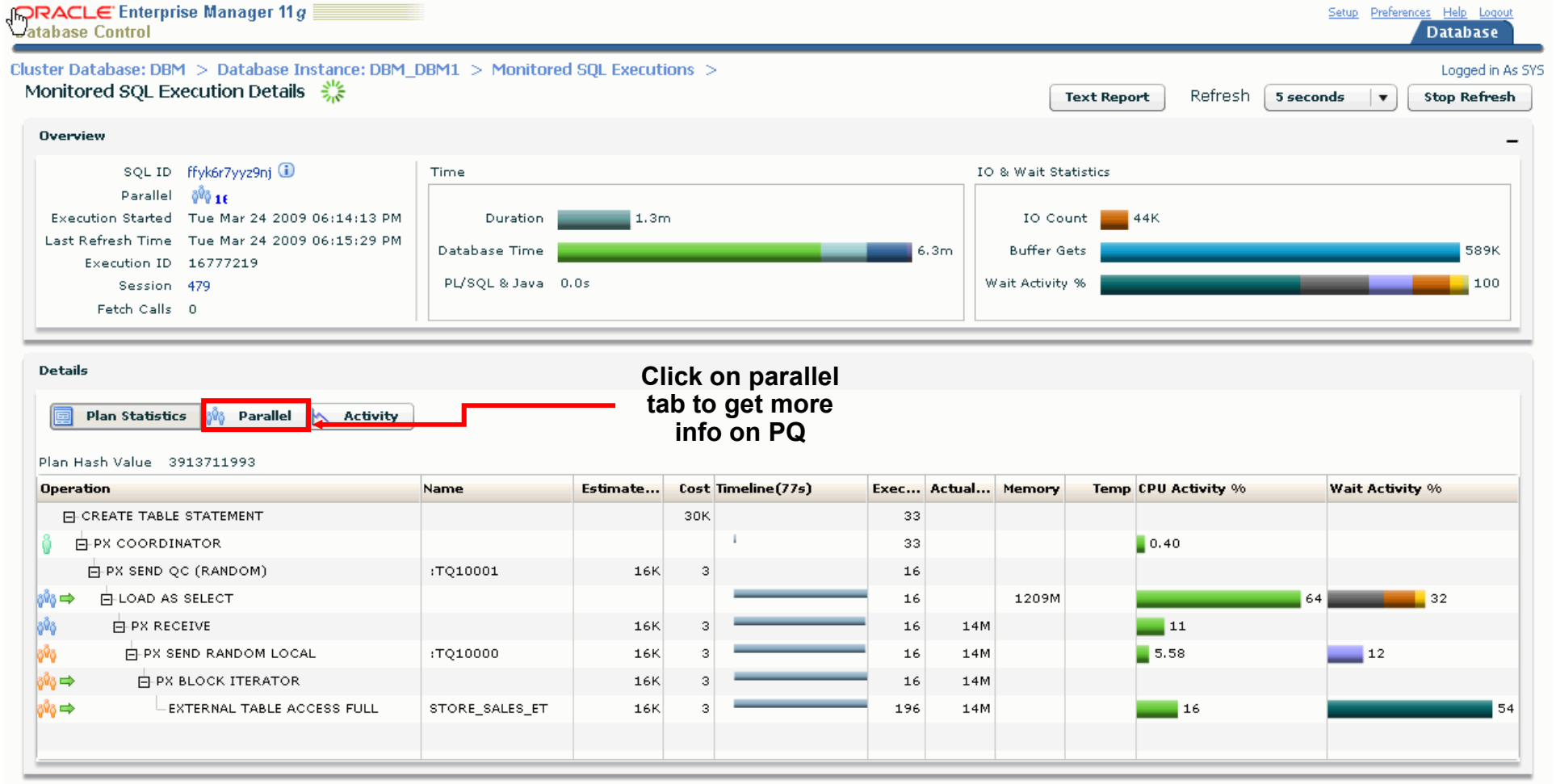
SQL Monitoring Screens - PWJ



Using EM to monitor Parallel Query



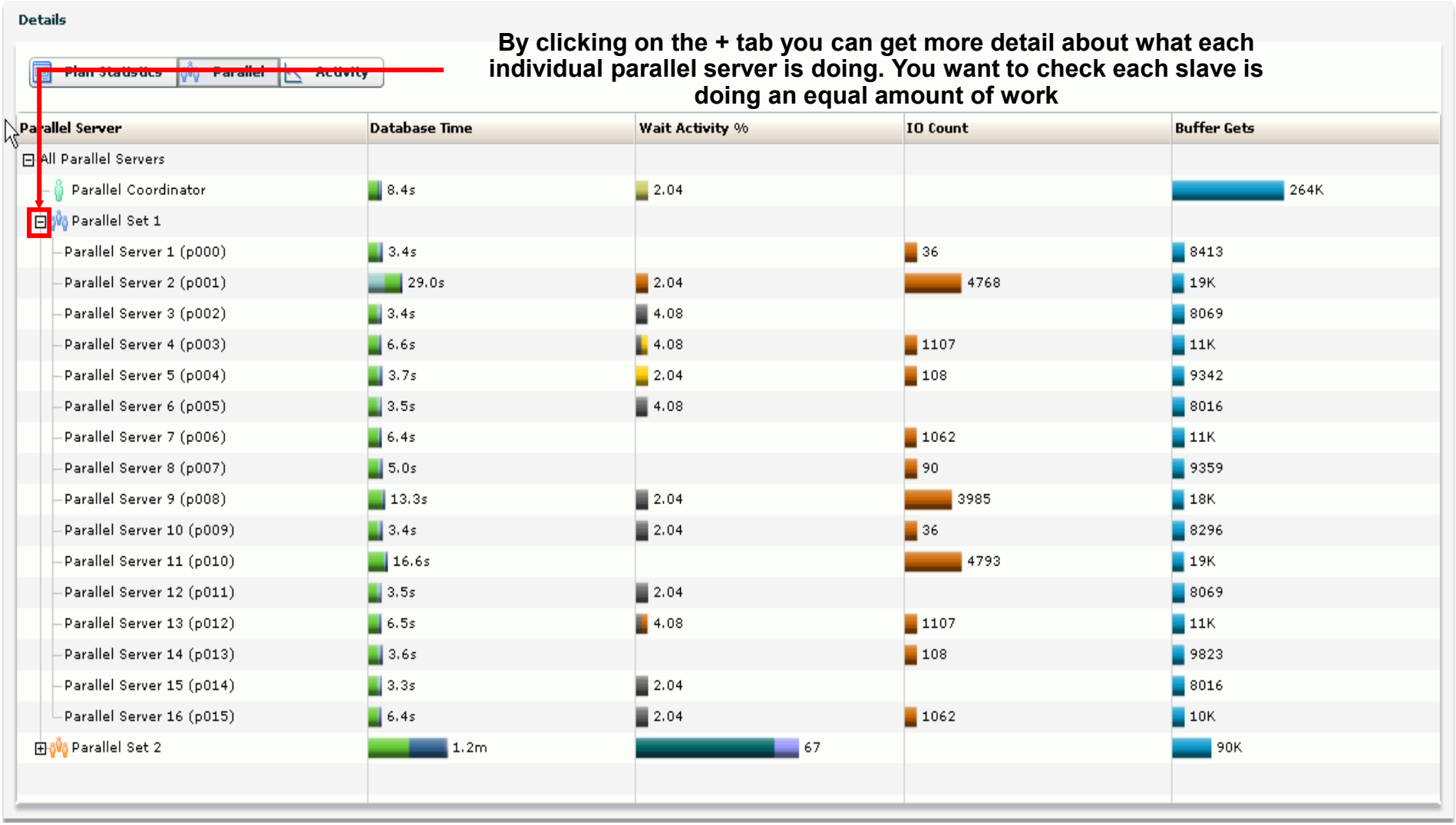
SQL Monitoring screens



Click on parallel tab to get more info on PQ

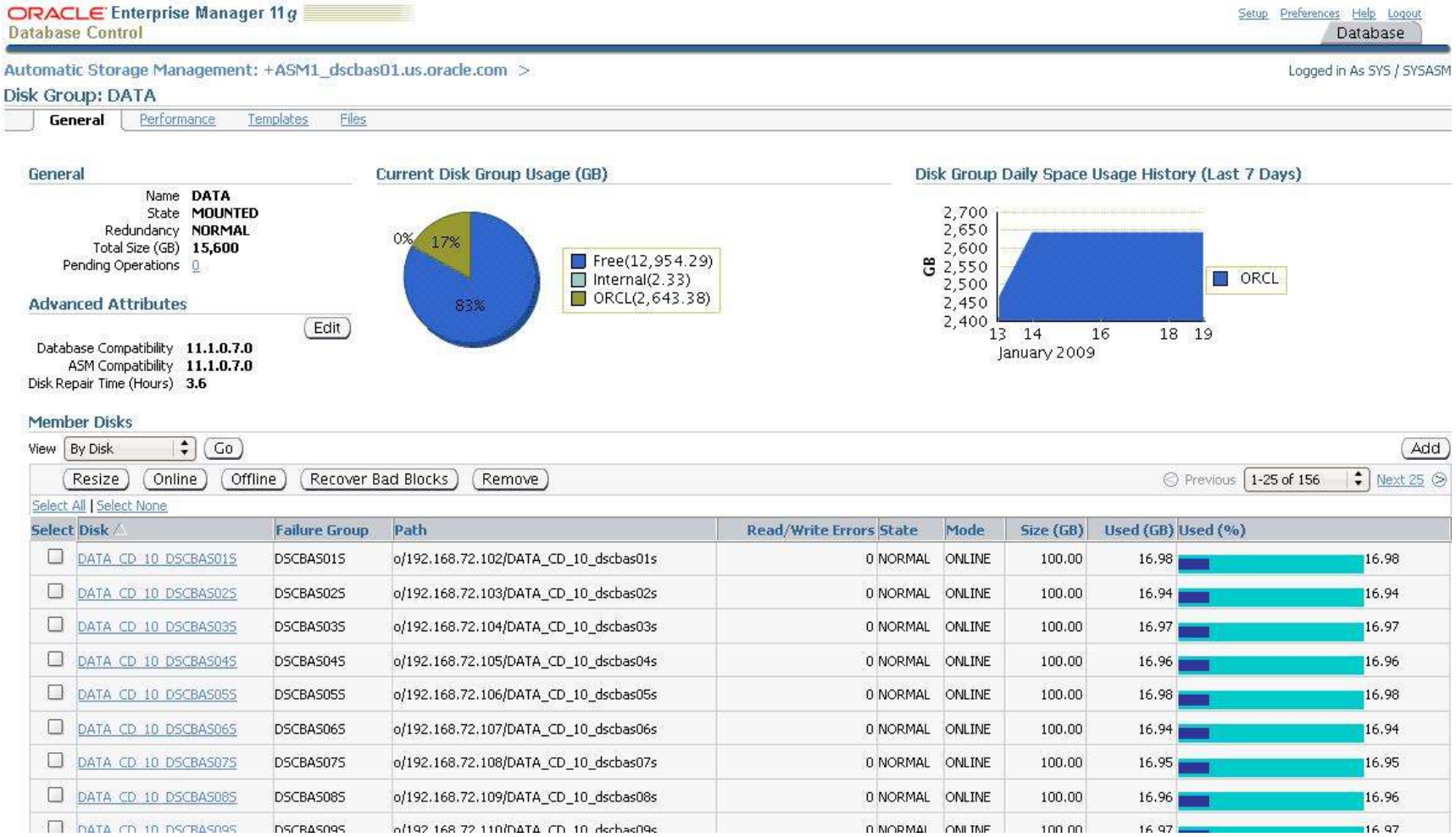
The green arrow indicates which line in the execution plan is currently being worked on

SQL Monitoring Screens



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



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

Exadata Sessions

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


ORACLE®