
ORACLE®

Your. Open. World.

Best Practices for Deploying a Data Warehouse on Oracle Database 11*g*

Maria Colgan – Principal Product Manager

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.

Agenda

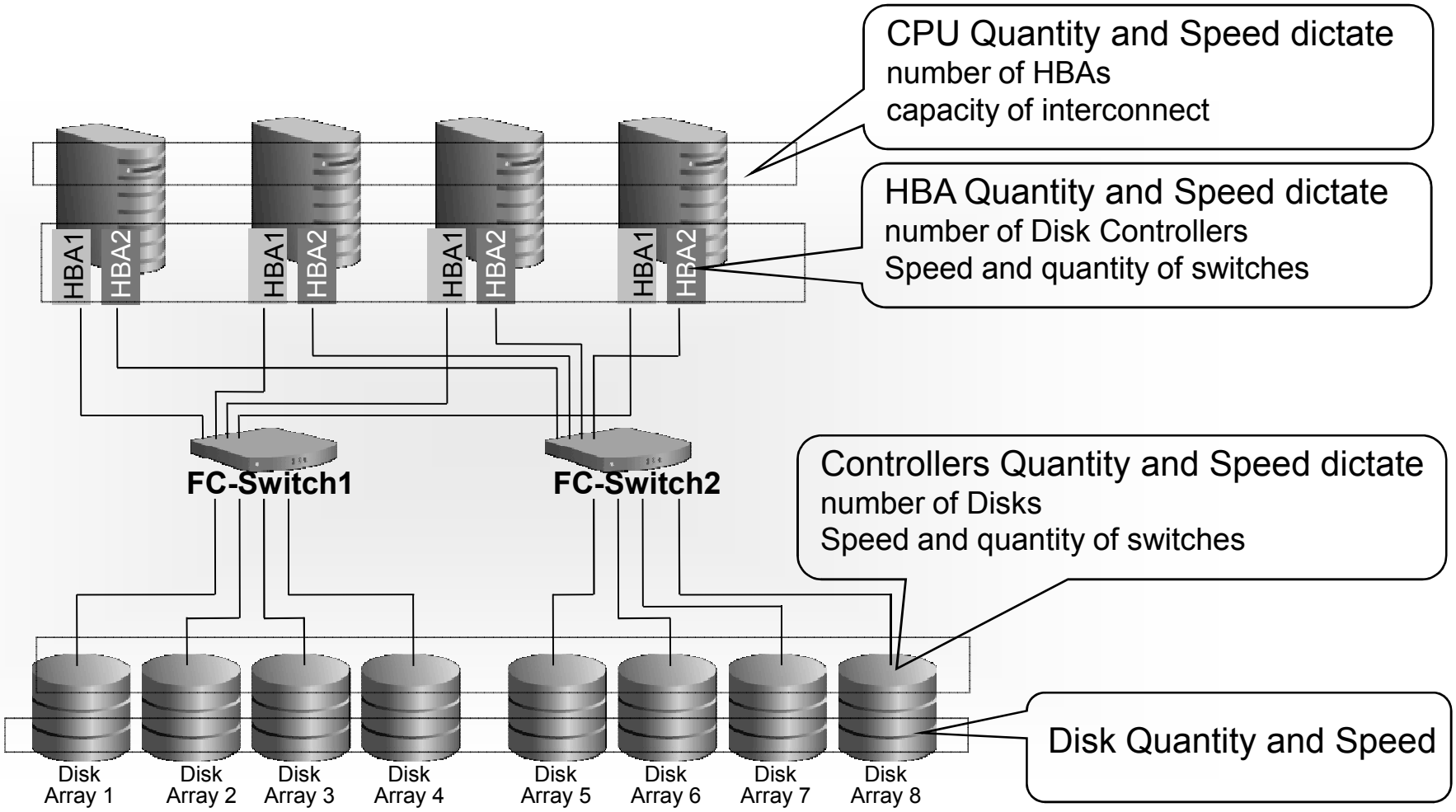
- **Hardware**
  - Building a balanced configuration
  - Disk / ASM configuration
- **Logical model**
  - Third Normal Form
  - Star Schema
- **Physical model**
  - Implementing Logical models
  - Data Loading
- **System management**
  - Parallel Query
  - Statistic management
  - Initialization parameters
  - Workload monitoring

Hardware

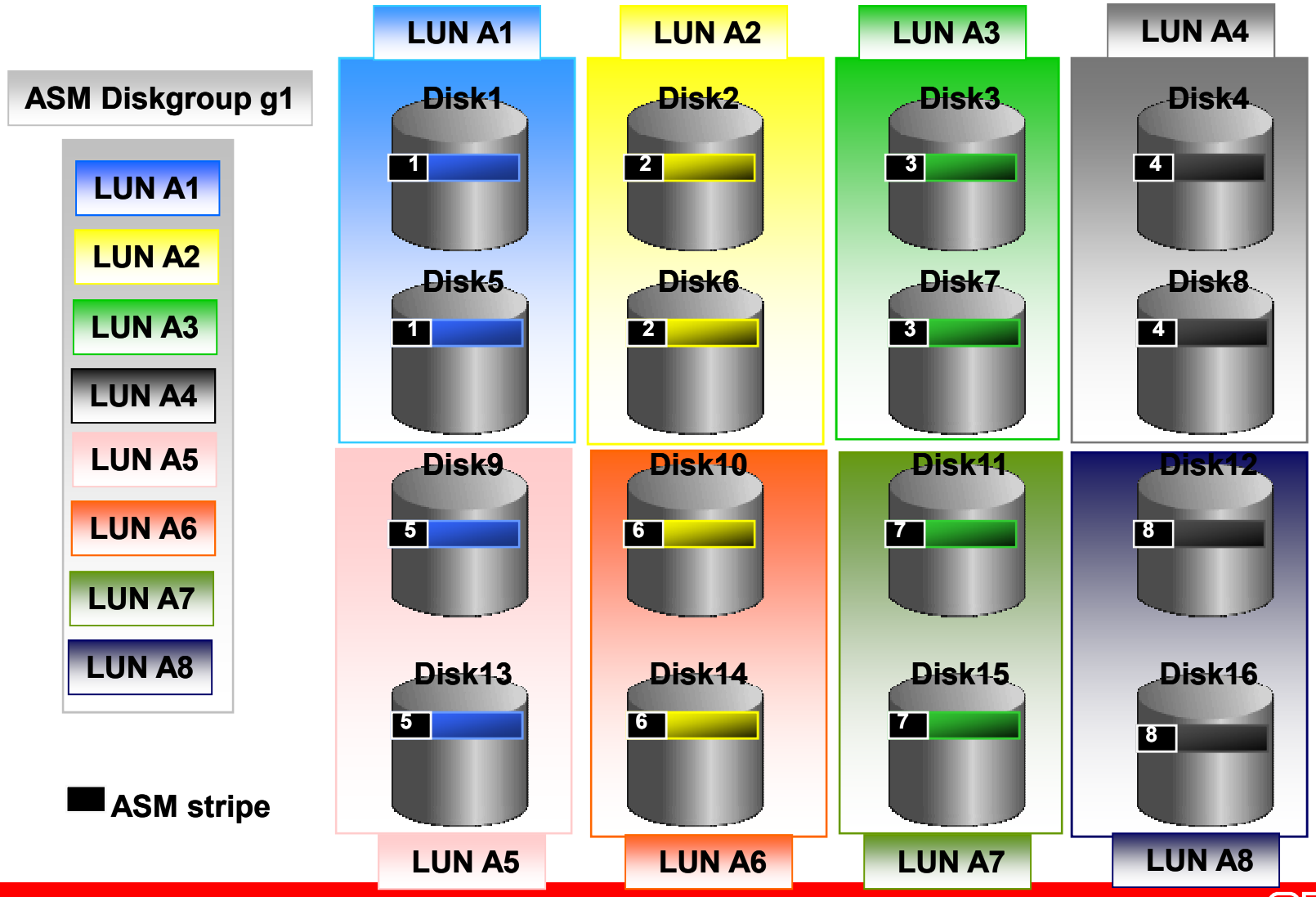
"The weakest link" defines the throughput

Balanced Configuration

"The weakest link" defines the throughput



Disk Layout - S.A.M.E Architecture with RAID 1 and ASM



Data Warehouse hardware configuration best practices

- Build a balance hardware configuration
  - Total throughput = # cores X 100-200 MB (depends on chip set)
  - Total HBA throughput = Total core throughput
    - If total core throughput =1.6GB will need 4 4Gb HBAs
  - Use 1 disk controller per HBA Port (throughput capacity must be equal)
  - Switch must be same capacity as HBA and disk controllers
  - Max of 10 physical disks per controller(Use smaller drives 146 or 300 GB)
- Minimum of 4GB of Memory per core (8GB if using compression)
- Interconnect bandwidth should equal IO bandwidth (Infiniband)
- Use ASM with RAID 1 mirroring for redundancy
  - Create two ASM diskgroups (1 for data, 1 for flash recovery area)
  - Use ATTRIBUTE 'au_size‘ to increase allocation unit

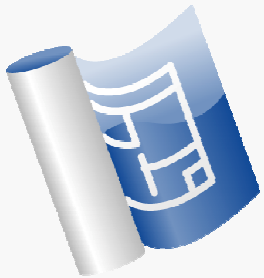
Data Warehousing Hardware Solutions

Custom

- Complete Flexibility
- Any OS, any platform
- Easy fit into a company's IT standards

Reference Configurations

- Documented best-practice configurations for data warehousing



Optimized Warehouse

- Scalable systems pre-installed and pre-configured: ready to run out-of-the-box

HP Oracle Database Machine

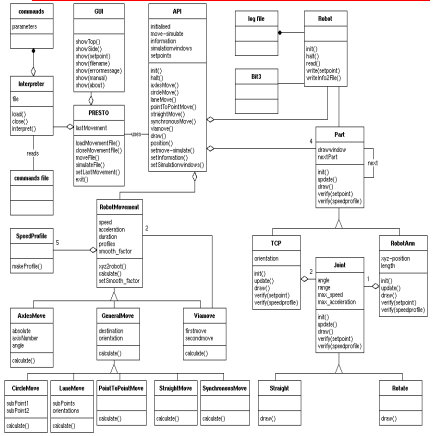
- Highest performance
- Pre-installed and pre-configured
- Sold by Oracle

Pre-configured, Pre-installed, Validated

Complete Flexibility

Logical Models

Visual representation of business concepts and relationships



Why Schema Modeling is important

- **Model according to your business needs**
  - Don't get lost in academia
  - Ignore the physical DB, hardware or end-user tools that will eventually be used
- Two main data warehouse models
  - Third Normal Form (3NF)
    - Minimizes data redundancy through normalization
    - Typically has a large number of tables
  - Star Schema
    - Simplest model
    - One fact table surrounded by multiple dimension tables
- Upon completion model should be
  - Easy to map to fact and dimension tables in physical database
  - Show clearly how information in operational systems will fit in data warehouse

Industry best practices on when to use each model

- 3rd Normal Form
  - Preserve a detailed record of each transaction without any data redundancy
  - Allows for rich encoding of attributes & all relationships between data elements
  - Users typically require a solid understanding of the data in order to navigate
- Star Schema
  - Dimensional approach that simplified the data model to facilitate access
  - Drill paths, hierarchy and query profile are embedded in the data model itself rather than the data
  - Easier for in-experienced users to navigate
- Forrester:
  - 3rd Normal Form is the self-less model (Neutral)
  - Star Schema is the selfish model (Subject oriented)

What does a 3rd Normal Form schema look like?

- Large number of tables due to normalization
- Multiple fact tables
- Lots of large table joins

Optimizing 3rd Normal Form

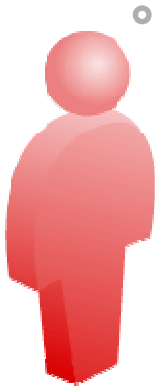
Requires 3 Ps - Power, Partitioning, Parallelism

- larger tables or fact tables should be partitioned
  - Use composite partitioning range-hash
  - Range to facility the data load and data elimination
  - Hash on join column to facility partition wise joins
  - Number of hash partitions should be power of 2 (#CPU X 2) 
- 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

Goal is parallel partition wise joins

Partition Pruning

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

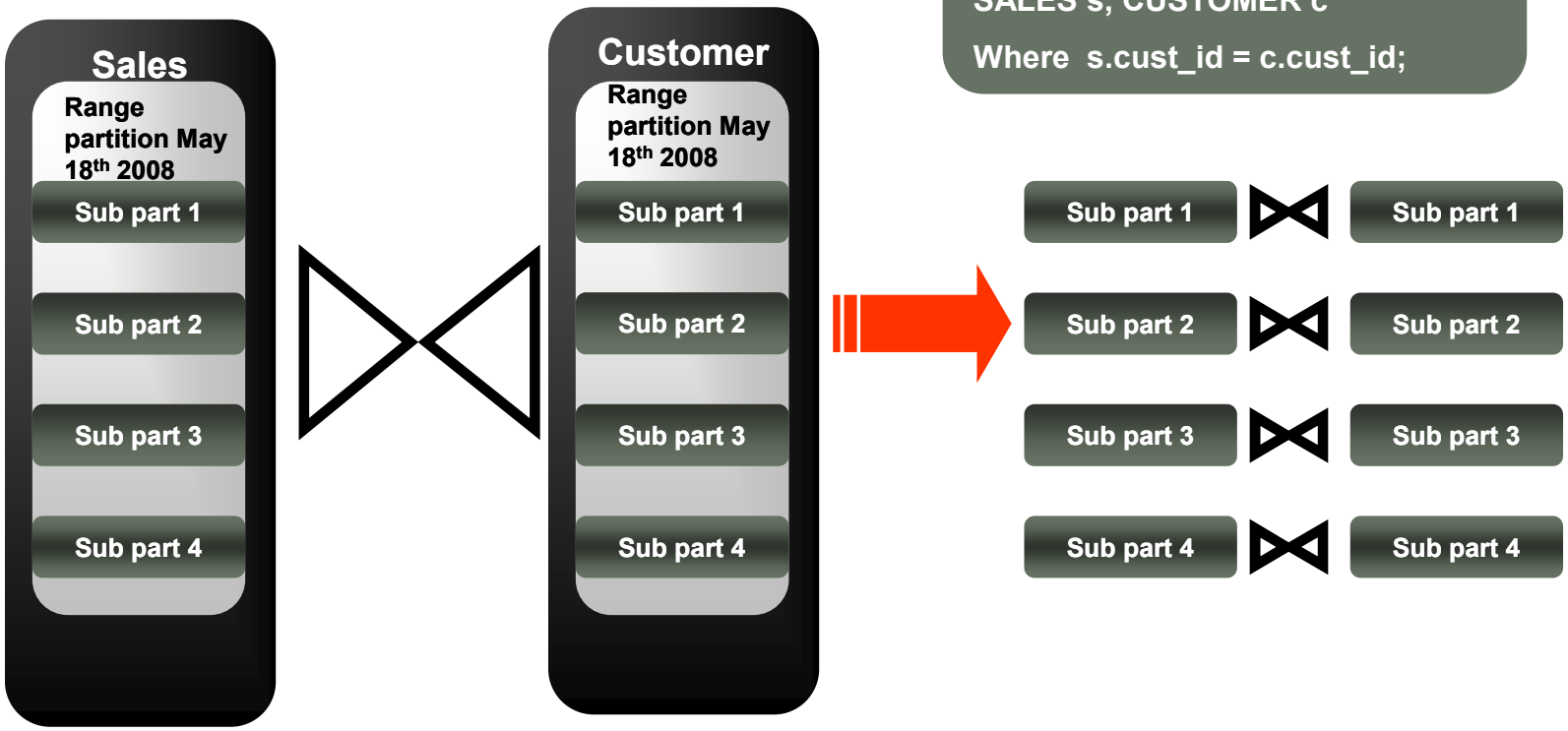


```
Select sum(sales_amount)
From SALES
Where sales_date between
to_date('05/20/2008','MM/DD/YYYY')
And
to_date('05/23/2008','MM/DD/YYYY');
```

Only the 3 relevant partitions are accessed

Partition Wise join

```
Select sum(sales_amount)
From
SALES s, CUSTOMER c
Where  s.cust_id = c.cust_id;
```



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

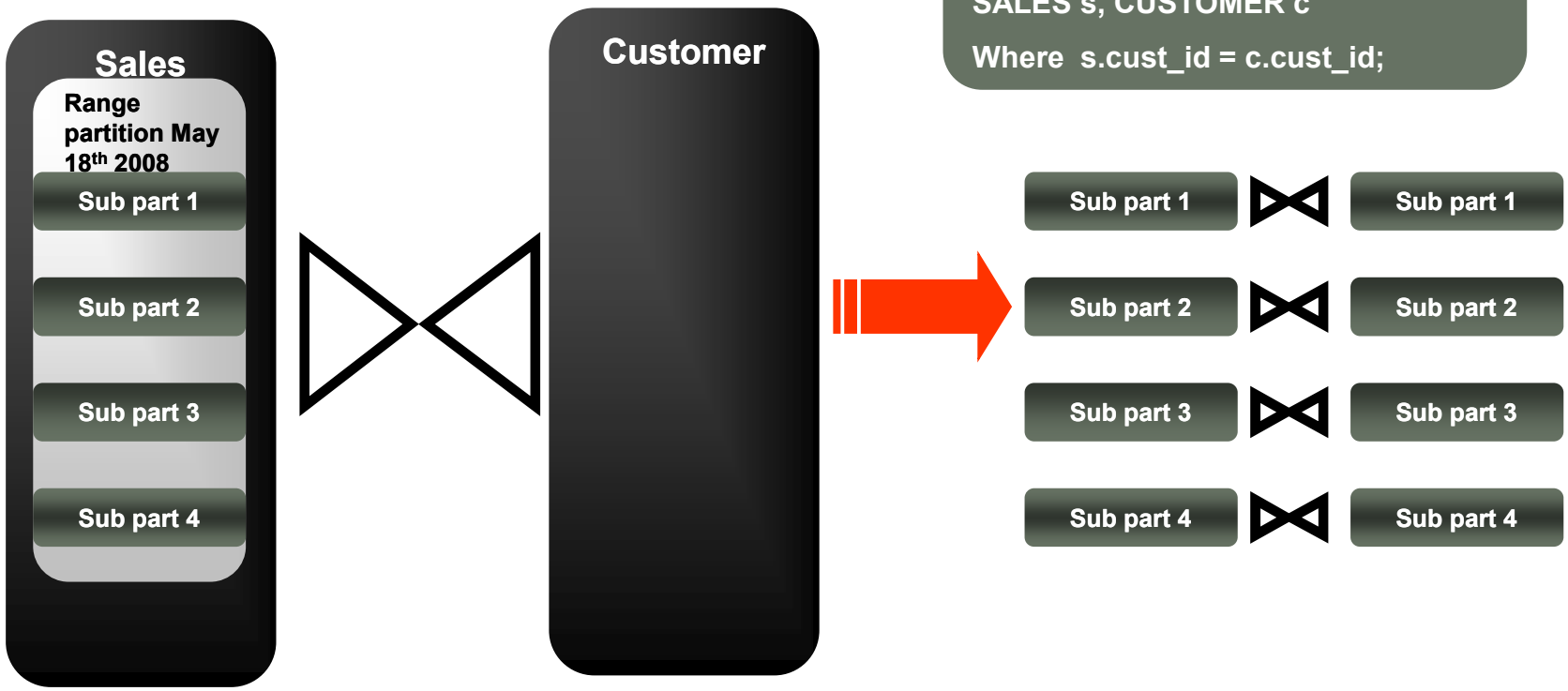
Execution plan for partition-wise join

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	

Partial Partition Wise join

```
Select sum(sales_amount)
From
SALES s, CUSTOMER c
Where  s.cust_id = c.cust_id;
```

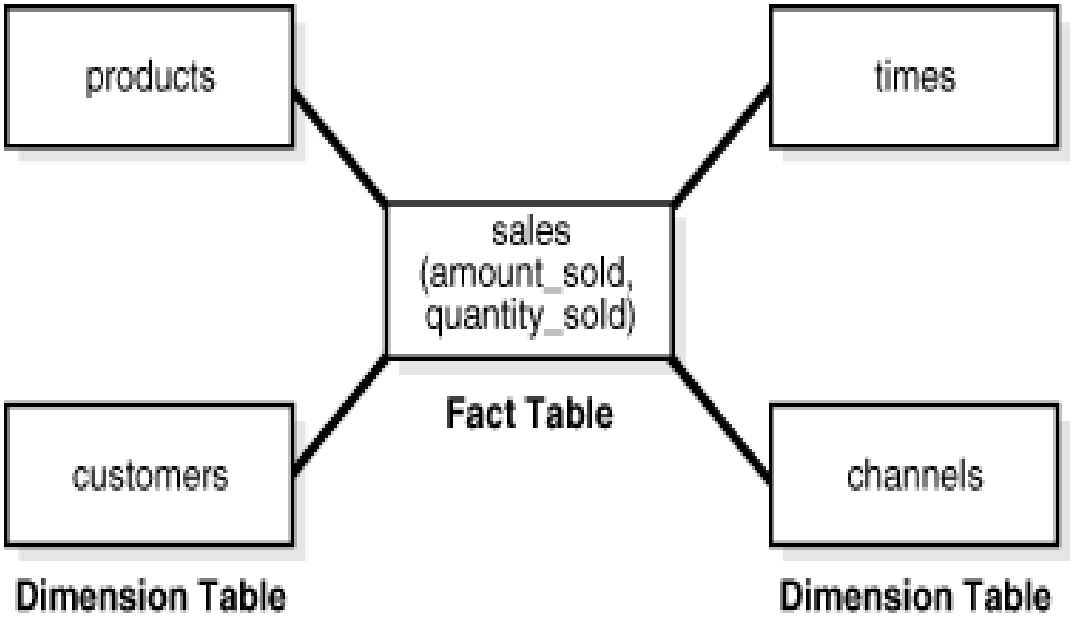


Only the Sales table is hash partitioned on the cust_id column

Rows from customer are dynamically redistributed on the join key cust_id to enable partition-wise join

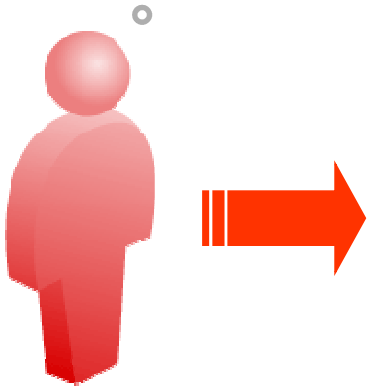
What does a Star Schema look like?

- Called star schema because diagram resembles a star
- The center of the star consists of one or more fact tables
- The points of the star are the dimension tables



Star Query

Q: What was the total number of umbrellas sold in Boston during the month of May 2008 ?



```
Select SUM(quanity_sold) total_umbrellas_sold_in_Boston
From Sales s, Customers c, Products p, Times t
Where s.cust_id   = c.cust_id
And     s.prod_id  = p.prod_id
And     s.time_id   = t.time_id
And     c.cust_city = 'BOSTON'
And     p.product  = 'UMBRELLA'
And     t.month     = 'MAY'
And     t.year        = 2008;
```

Optimizing Star schema

- Create bitmap index on foreign key columns in fact table
- Set STAR_TRANSFORMATION_ENABLED to TRUE

Goal is star transformation

- Powerful optimization technique that rewrites or transform SQL
- Executes the query in two phases
- The first phase retrieves necessary rows (row set) from the fact table
  - Bitmap joins between bitmap indexes on all of the foreign key columns
- The second phase joins this row set to the dimension tables
  - The join back to the dimension tables done using a hash join

Star Transformation in detail

```
Select SUM(quanity_sold)
 From Sales s, Customers c, Products p, Times t
Where s.cust_id   = c.cust_id
And     s.prod_id  = p.prod_id
And     s.time_id   = t.time_id
And     c.cust_city = 'BOSTON'
And     p.product  = 'UMBRELLA'
And     t.month     = 'MAY'
And     t.year        = 2008;
```

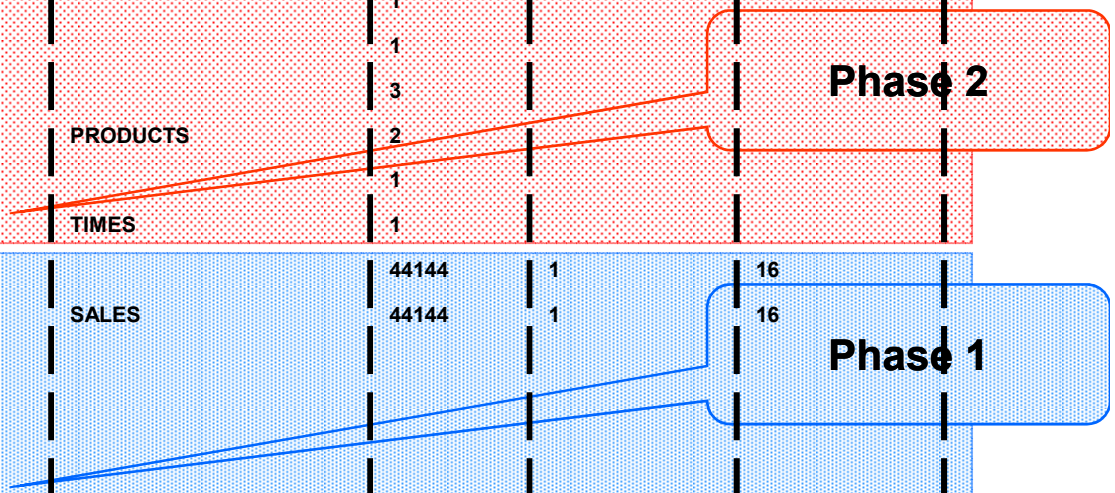
Step 1: Oracle rewrites / transforms the query to retrieve only the necessary rows from the fact table using bitmap indexes on foreign key columns

```
Select SUM(quanity_sold)
From Sales s
Where s.cust_id  IN
(Select c.cust_id From Customers c Where c.cust_city = 'BOSTON')
And     s.prod_id  IN
(Select p.prod_id From Products p where p.product  = 'UMBRELLA')
And     s.time_id   IN
(Select t.time_id From Times t Where t.month ='MAY' And  t.year =2008);
```

Step 2: Oracle joins the rows from fact table to the dimension tables

Execution plan for Star Query

ID	Operation	Name	Rows	Pstart	Pstop
0	SELECT STATEMENT		1		
1	SORT GROUP BY NOSORT		1		
2	HASH JOIN		3		
3	TABLE ACCESS FULL	PRODUCTS	2		
4	HASH JOIN		1		
5	TABLE ACCESS FULL	TIMES	1		
6	PARTITION RANGE SUBQUERY		44144	1	16
7	TABLE ACCESS BY LOCAL INDEX ROWID	SALES	44144	1	16
8	BITMAP CONVERSION TO ROWIDS				
9	BITMAP AND				
10	BITMAP MERGE				
11	BITMAP KEY ITERATION				
12	BUFFER SORT				
13	TABLE ACCESS FULL	TIMES	1		
14	BITMAP INDEX RANGE SCAN	SALES_TIME_BIX		1	16
15	BITMAP MERGE				
16	BITMAP KEY ITERATION				
17	BUFFER SORT				
18	TABLE ACCESS FULL	CUSTOMERS	1		
19	BITMAP INDEX RANGE SCAN	SALES_TIME_BIX		1	16
20	BITMAP MERGE				
21	BITMAP KEY ITERATION				
22	BUFFER SORT				
23	TABLE ACCESS FULL	PRODUCTS	2		
24	BITMAP INDEX RANGE SCAN	SALES_PROD_BIX		1	16

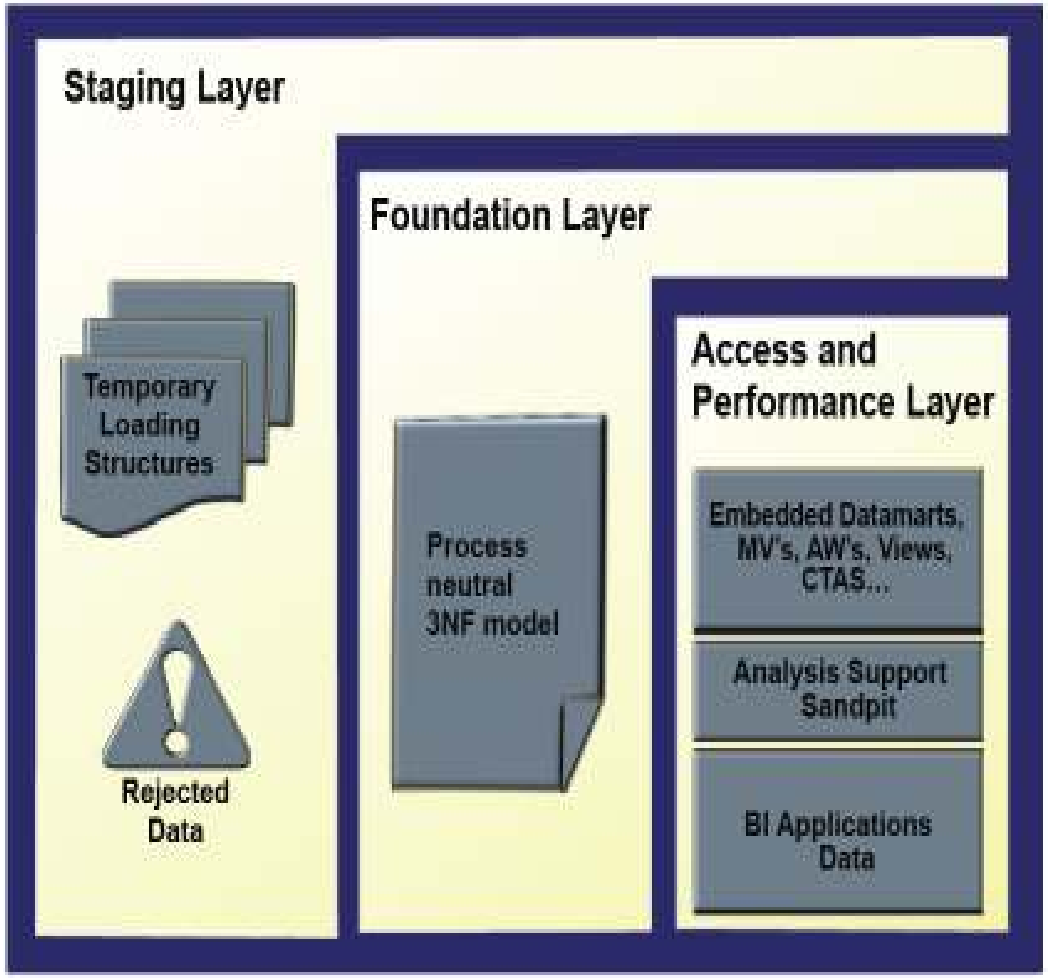


Physical Models

Implementing the logical model
Can both logical models co-exist?

Physical implementation of Logical Model

Blueprint of physical layers in Data warehouse



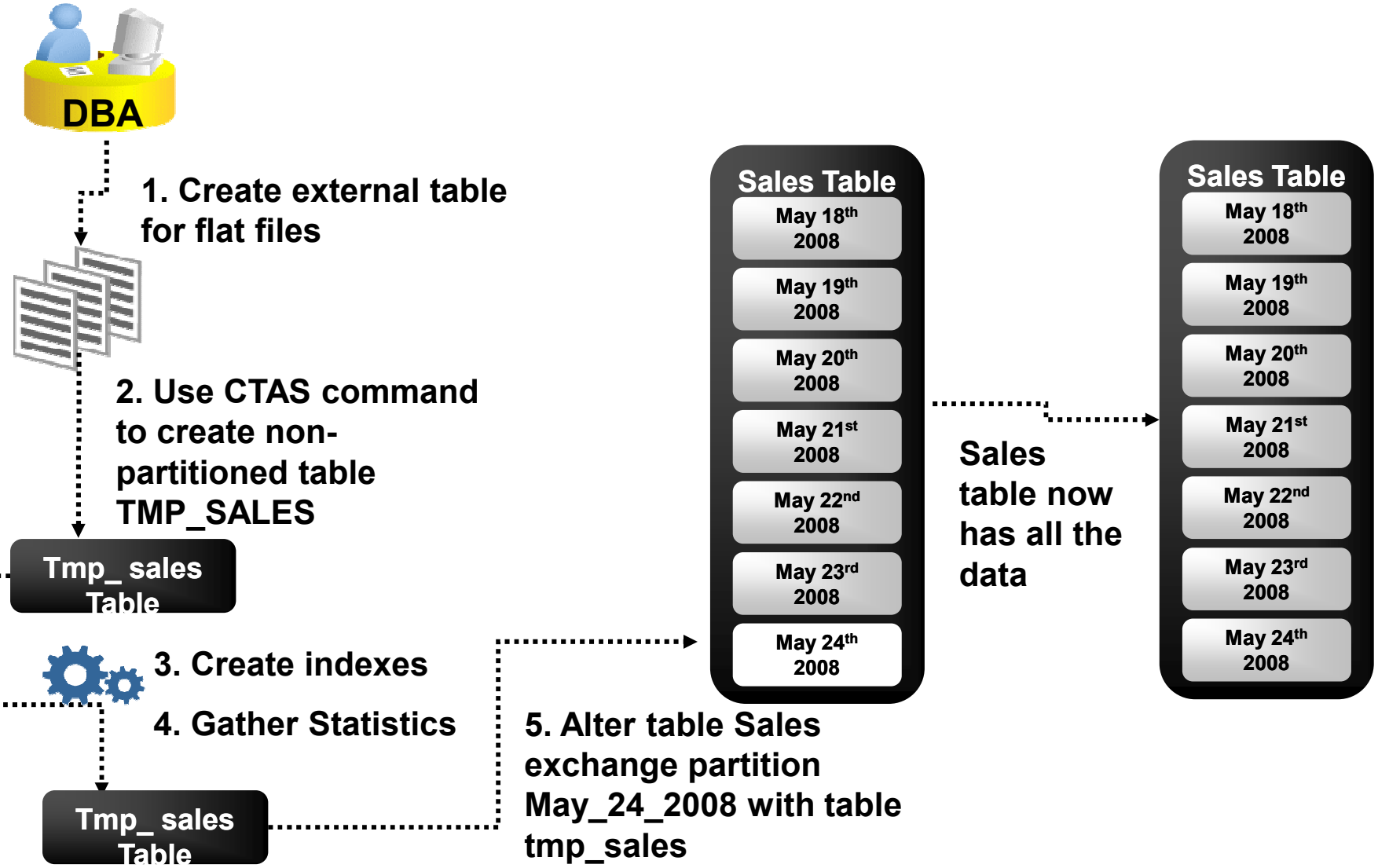
Data Loading

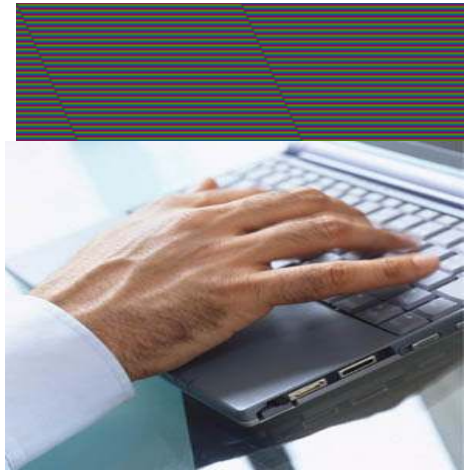
Staging layer and beyond

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 and data cleansing to occur 'on the fly'
  - Avoids space wastage
- **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**

Partition Exchange loading





System management

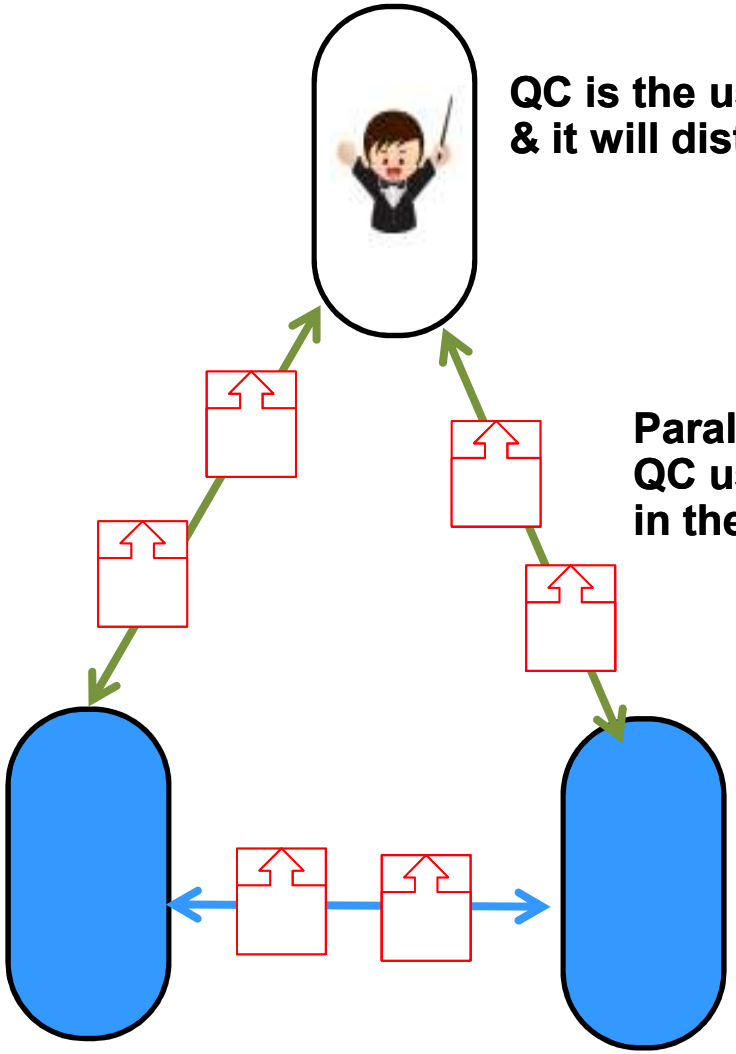
Maximizing Resources
and avoiding “tuning
spiral of death”

System management

Keeping the lights on for your data warehouse

- Parallel Execution
  - Use common sense to apply parallelism only where it will help performance and not hinder it
- Resource Manager
  - If your data warehouse is CPU bound protects critical tasks from interference from non-critical tasks
- Always have accurate Optimizer statistics
  - Use INCREMENTAL statistic maintenance or copy_stats to keep large partitioned fact - table up to date
- Set only the initialization parameters that you need to
  - Avoid tuning spiral of death by not tuning
- Workload Monitoring
  - Take hourly AWR or statspack report
  - Use EM to do real-time system monitoring
    - New Parallel Execution and SQL Monitoring screen in 11g

SQL Parallel Execution



QC is the user session that initiates the parallel SQL statement & it will distribute the work to parallel servers

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 They are allocated from a pool of globally available parallel server processes and assigned to a given operation

Messages Parallel server connection QC connection

SQL 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

Controlling PQ on RAC
Using services

Create two services

Srvctl add service –d database_name

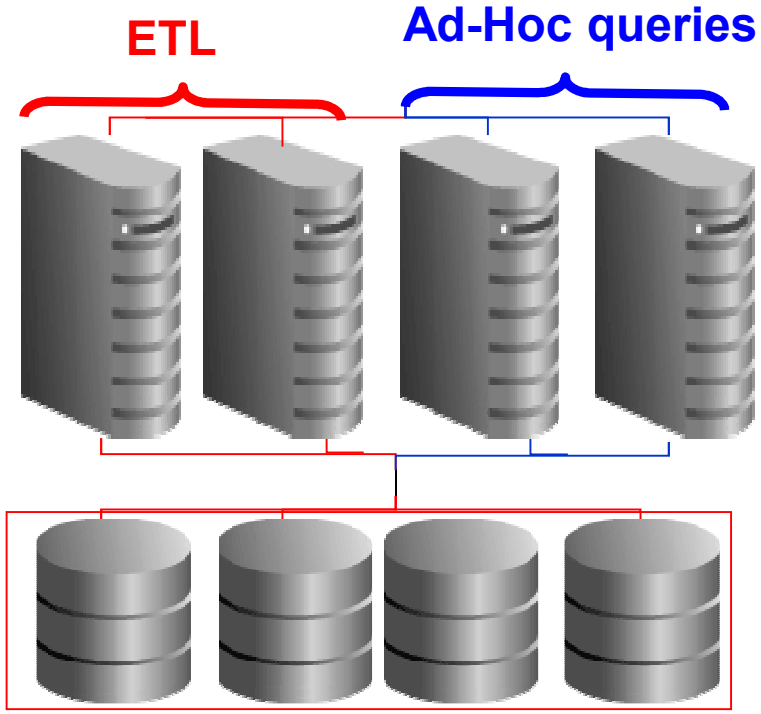
-s ETL

-r sid1, sid2

Srvctl add service –d database_name

-s AHOC

-r sid3, sid4



Note: Prior to 11g use init.ora parameters instance_groups and parallel_instance_group to control PQ on RAC

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

Settings may vary on your system- either in size range or DOP - and highly depend on your target workload & hardware configuration

Efficiency Statistics Management

ORACLE DATABASE 11g

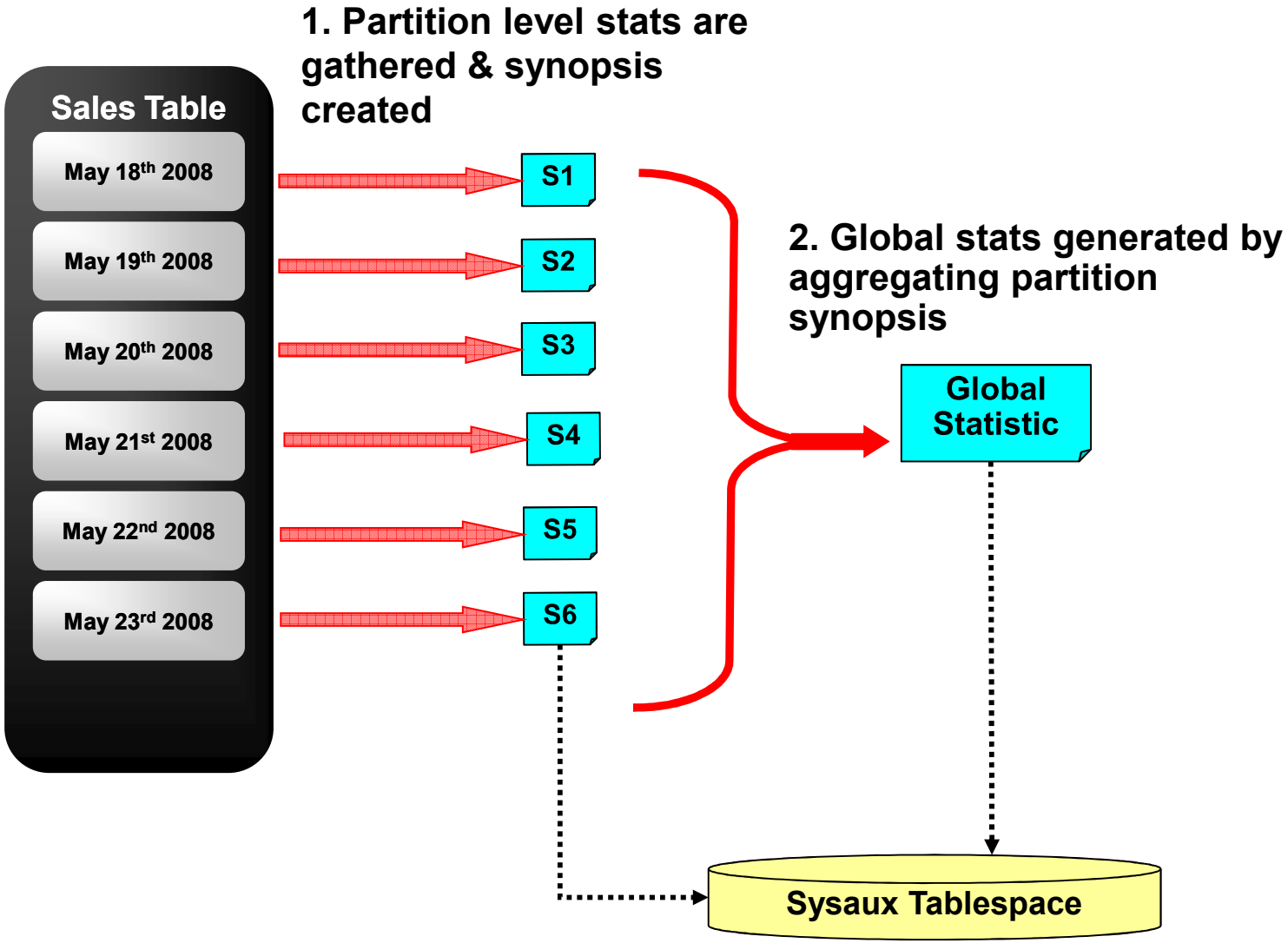
- How do I gather accurate Statistics
  - *" .. Compute statistics gives accurate results but takes too long .."*
  - *" .. Sampling is fast but not always accurate .."*
  - *" .. AUTO SAMPLE SIZE does not always work with data skew .."*

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

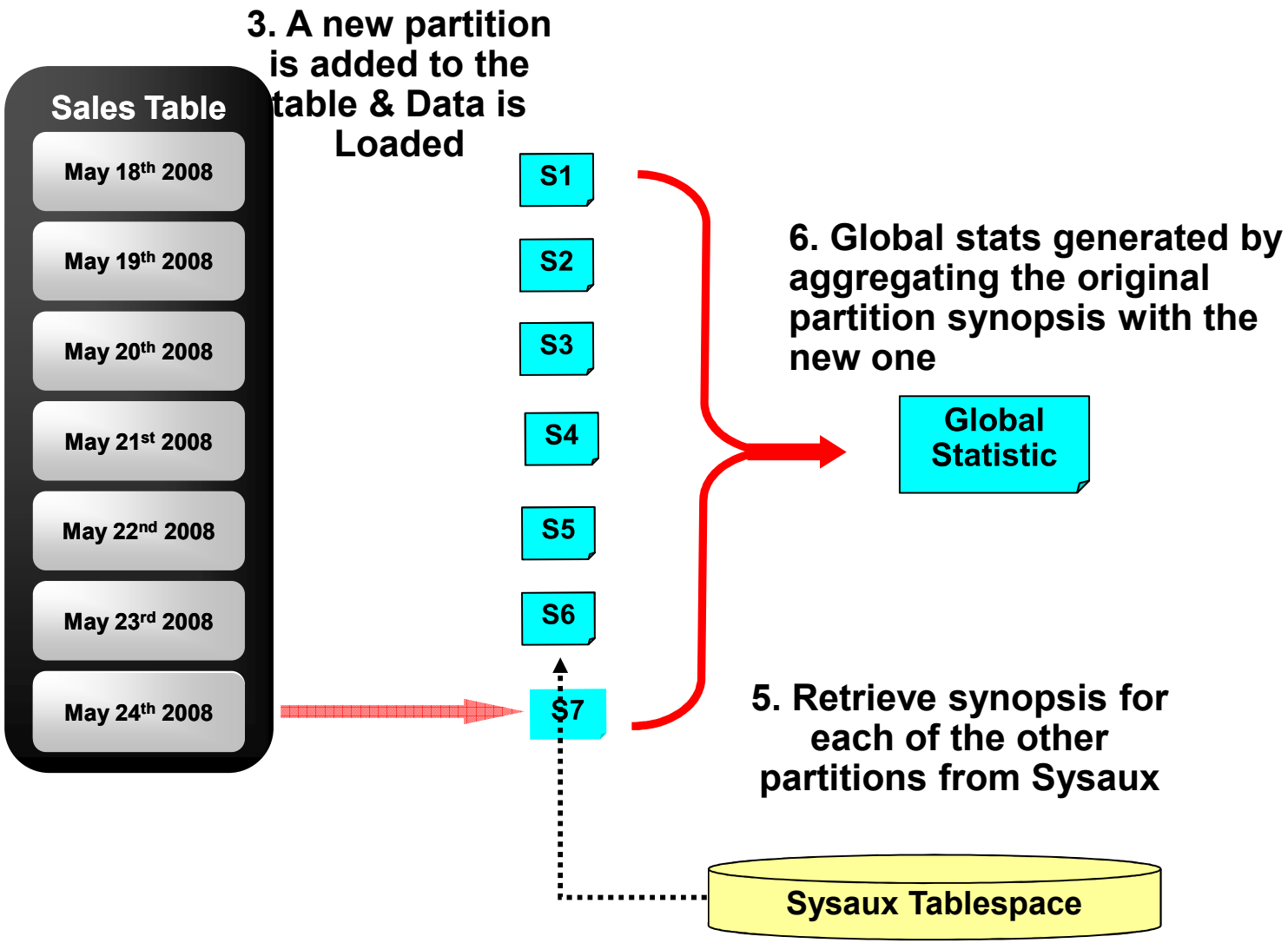
- Gathering statistics on one partition (e.g. after a bulk load) causes a full scan of all partitions to gather global table statistics Extremely time and resource intensive

- 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

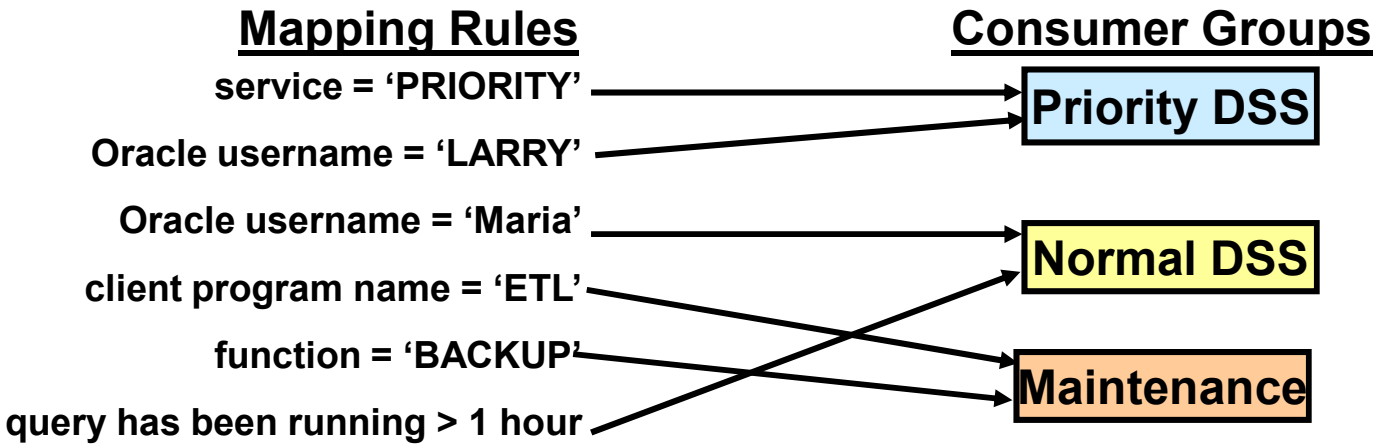


Managing your workload
Oracle Database Resource Manager

- If your data warehouse is CPU bound, consider using CPU Resource Management
  - Protects critical tasks from interference from non-critical tasks
  - Allows CPU to be utilized according to a specific ratio
  - Prevents thrashing and system instability that can occur with excessive CPU loads

Oracle Database Resource Manager

- To use Resource Manager:
  1. Define Consumer Groups for each type of workload
     - Priority DSS consumer group
     - Normal DSS consumer group
     - Maintenance consumer group
  2. Create rules to dynamically map sessions to consumer groups, based on session attributes



  3. Define Resource Plan: policy for managing workloads

Oracle Database Resource Manager

- Resource Manager can manage Degree Of Parallelism (DOP)
  - Specify maximum DOP for each consumer group
    - “OLTP” consumer group: max DOP = 0 (serial queries only)
    - “Low-Priority” consumer group: max DOP = 4
    - “Critical Reports” consumer group: max DOP = unlimited
  - Actual DOP determined by
    - Hints and parallel execution parameters -> proposed DOP
    - MIN(proposed DOP, consumer group’s max DOP) -> actual DOP

Initialization parameters

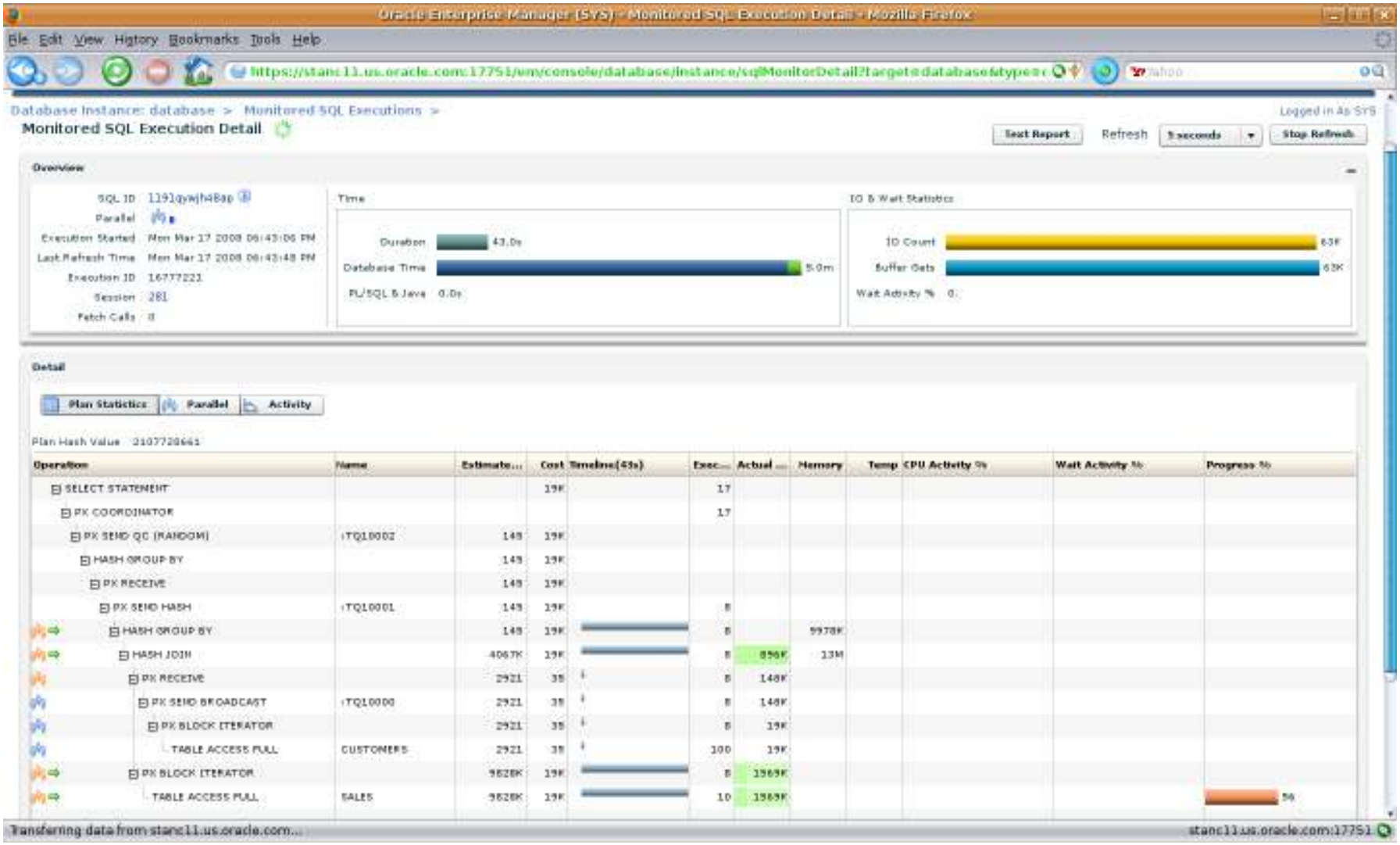
Only set what you really need to

- Db_block_size
  - 8, 16 or 32 (Larger may help with compression ratio)
- Db_file_multiblock_read_count
  - 1024/db_block_size
- Memory_target / SGA_target & PGA_Aggregate_target
  - Shared_pool needs to be size to accommodate PQ message buffers
- Parallel_min_servers
  - DOP X Avg # Concurrent Queries
- Parallel_max_servers
  - DOP X Max # Concurrent Queries
- Parallel_execution_message_size
  - 16K
- Parallel_adaptive_multi_user
  - False
- Star_transformation_enabled
  - True (if have star schema)

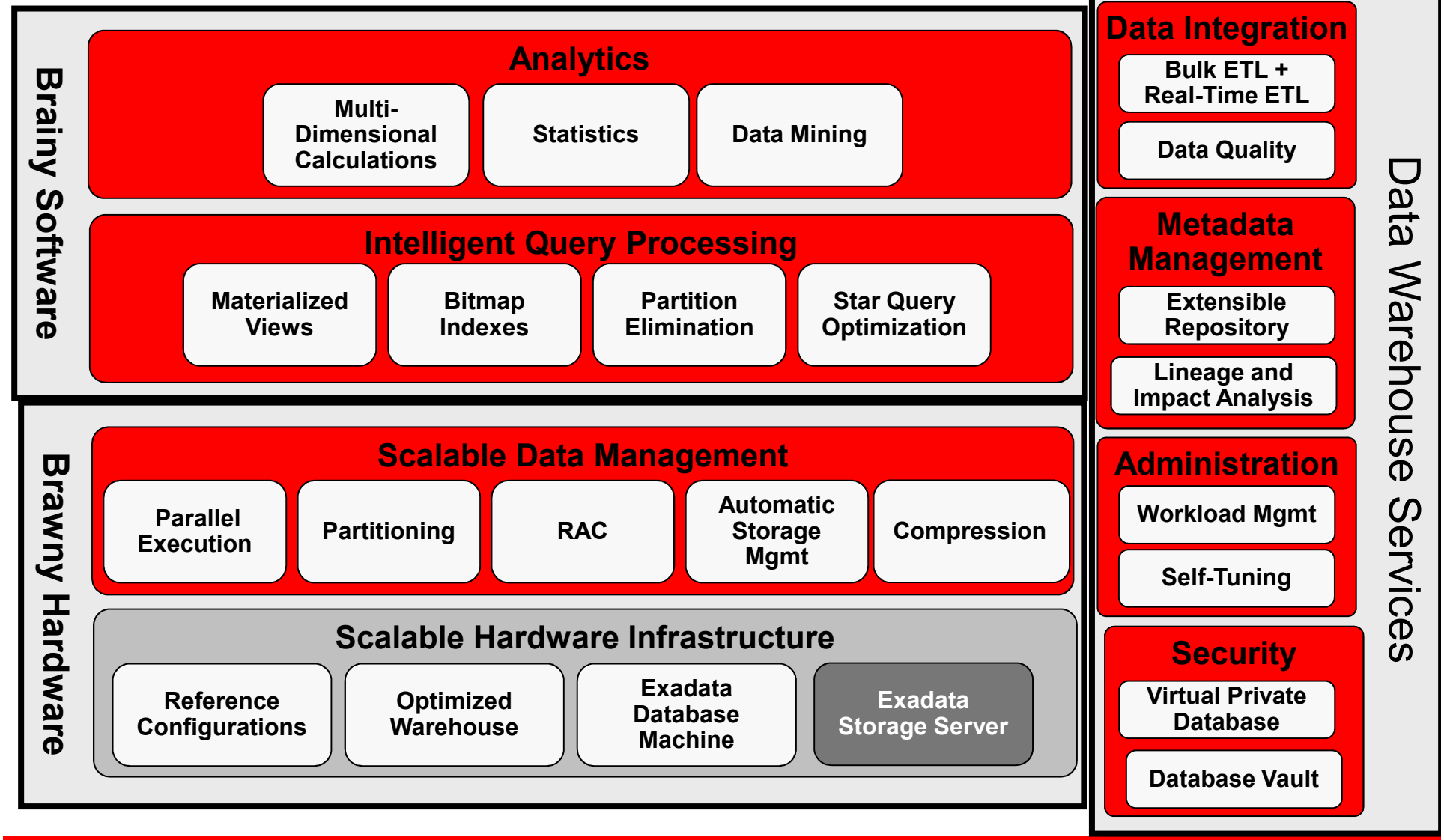
Workload monitoring

- Take hourly AWR or statspack snapshots
  - Establish baseline performance
- Use V$ performance views for command-line real-time monitoring
  - V$_session, (G)V$_PQ, (G)V$_PX, (G)V$PX_PROCESS
  - New view (G)V$SQL_MONITOR
    - Enables real-time monitoring of long-running or parallel SQL statements
- Or use Oracle Enterprise Manager Database Control 11g for
  - New Parallel Execution screens
    - Identifies all parallel execution activity
    - Displays how and where parallel resources are being used
  - SQL Monitoring screens
    - Visually identifies which parts of an execution plan are expensive relative to overall cost of statement
    - Provides information about the parallel server sets & work distribution

SQL Monitoring Screen



Oracle Data Warehouse Platform



For More Information

http://search.oracle.com

Data Warehousing 

OR

http://www.oracle.com/technology/products/bi/db/11g/pdf/twp_dw_best_practies_11g11_2008_09.pdf

Q & A

If you have more questions later, feel free to ask

Campground Demos

Demo	Location
Data Warehousing with Oracle Database 11g – booth L28	Moscone West Exhibit Hall
Oracle Optimized Warehouses - booth L24	Moscone West Exhibit Hall
Oracle Partitioning – booth L34	Moscone West Exhibit Hall
Oracle Data Mining – booth L21	Moscone West Exhibit Hall
Oracle OLAP – booth L27	Moscone West Exhibit Hall

Recommended Sessions

Session Title	Date	Time	Location
Oracle Warehouse Builder Road Map	Tuesday, Sept 23	9:00 a.m.	Moscone South: 102
Oracle Database 11g: Stories from a Data Warehouse Implementation	Tuesday, Sept 23	1:00 p.m.	Moscone South: 304
Get the Best Out of Oracle Partitioning:Things You Always Wanted to Know	Wednesday, Sept. 24	1:00 p.m.	Moscone South: 104
Oracle's New Database Accelerator: Query Processing Revolutionized	Thursday, Sept. 25	9:00 a.m.	Moscone South: 103
Oracle Database 11g for Data Warehousing	Thursday, Sept. 25	10:30 a.m.	Moscone South: 103
Oracle Real Application Clusters and QLogic InfiniBand: Yahoo! Large-Scale Data Warehouse	Thursday, Sept. 25	3:00 p.m.	Moscone South: 306