

ORACLE®



ORACLE®

Best Practices for Extreme Performance with Data Warehousing on Oracle Database

Rekha Balwada
Principal Product Manager

Agenda

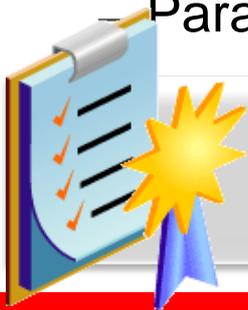
- Parallel Execution
- Workload Management on Data Warehouse
- Oracle Exadata Database Machine



Best Practices for Data Warehousing

3 Ps - Power, Partitioning, Parallelism

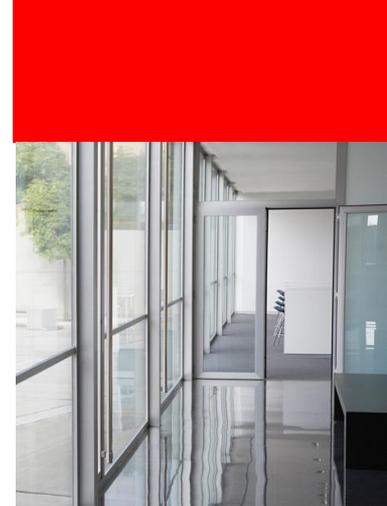
- **Power** – A Balanced Hardware Configuration
 - Weakest link defines the throughput
- **Partition** larger tables or fact tables
 - Facilitates data load, data elimination and join performance
 - Enables easier Information Lifecycle Management
- **Parallel** Execution should be used
 - Instead of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



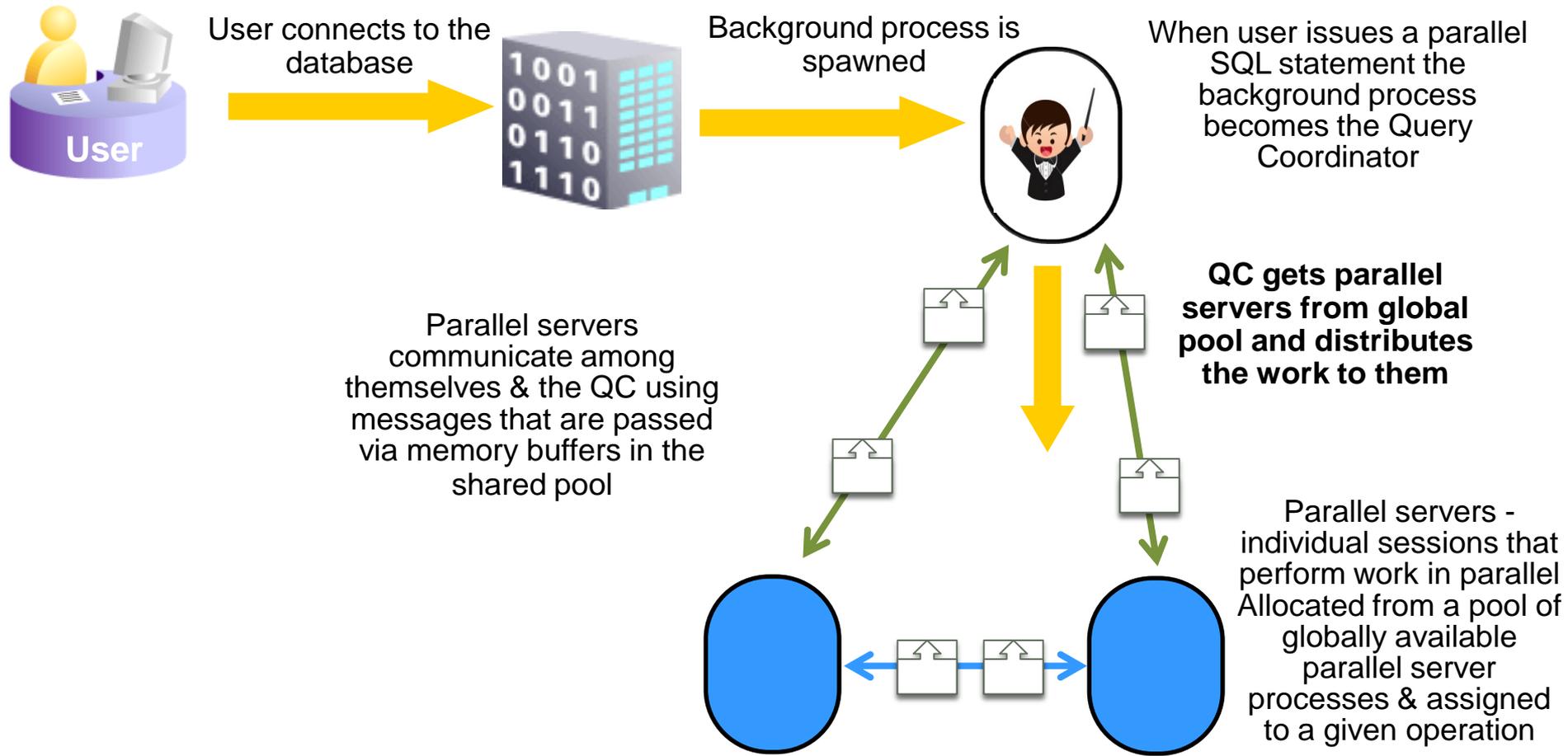
Goal is to minimize the amount of data accessed and use the most efficient joins

Agenda

- **Parallel Execution**
- Workload Management on a Data Warehouse
- Oracle Exadata Database Machine



How Parallel Execution works



Monitoring Parallel Execution

```
SELECT c.cust_last_name, s.time_id, s.amount_sold
FROM sales s, customers c
WHERE s.cust_id = c.cust_id;
```

Query Coordinator

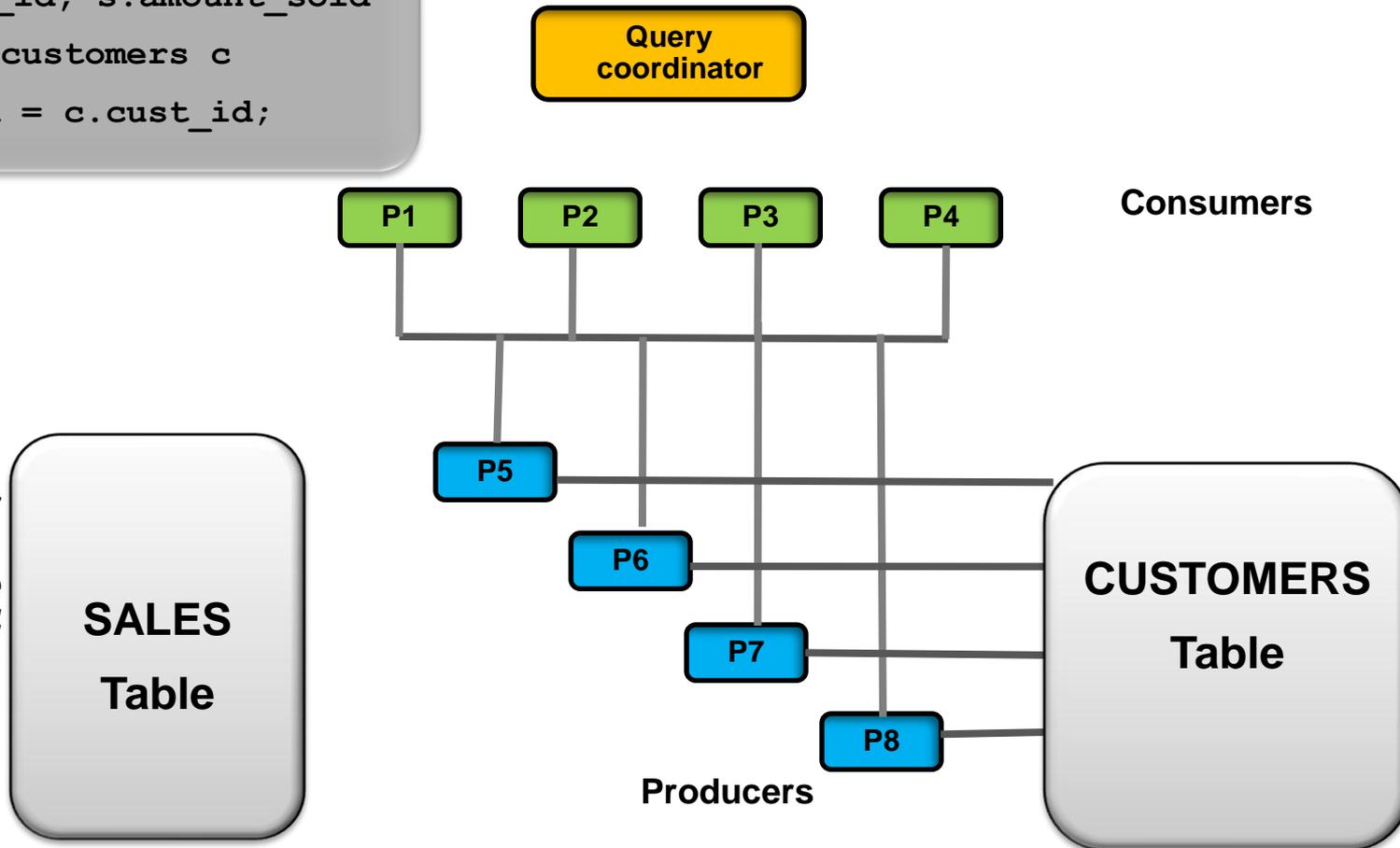
Id	Operation	Name	Parallelism	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				311 (100)	
1	PX COORDINATOR					
2	PX SEND QC (RANDOM)	:TQ10002	1049K	31M	311 (2)	00:00:04
* 3	HASH JOIN BUFFERED		1049K	31M	311 (2)	00:00:04
4	PX RECEIVE		55500	704K	112 (0)	00:00:02
5	PX SEND HASH	:TQ10000	55500	704K	112 (0)	00:00:02
6	PX BLOCK ITERATOR		55500	704K	112 (0)	00:00:02
* 7	TABLE ACCESS FULL	CUSTOMERS	55500	704K	112 (0)	00:00:02
8	PX RECEIVE		1049K	18M	196 (2)	00:00:03
9	PX SEND HASH	:TQ10001	1049K	18M	196 (2)	00:00:03
10	PX BLOCK ITERATOR		1049K	18M	196 (2)	00:00:03
* 11	TABLE ACCESS FULL	SALES	1049K	18M	196 (2)	00:00:03

Parallel Servers
do majority of the work

How Parallel Execution works

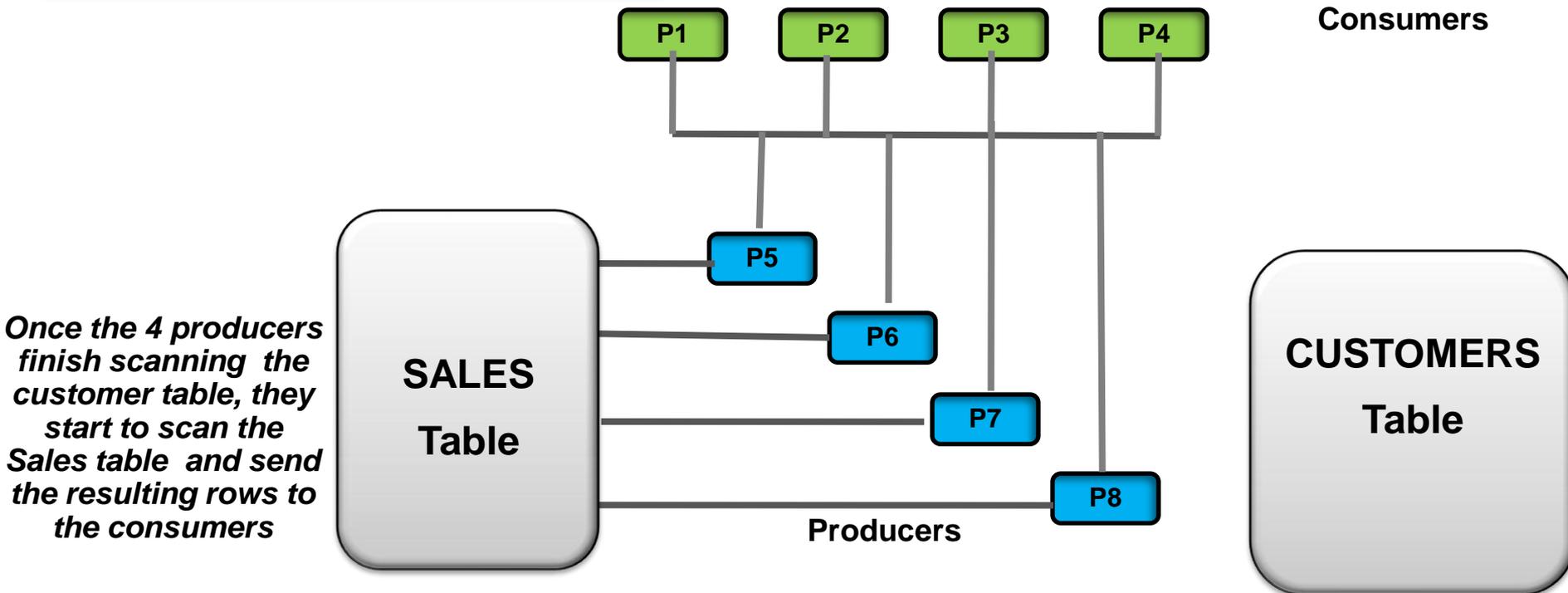
```
SELECT c.cust_last_name,  
       s.time_id, s.amount_sold  
FROM   sales s, customers c  
WHERE  s.cust_id = c.cust_id;
```

Hash join always begins with a scan of the smaller table. In this case that's the customer table. The 4 producers scan the customer table and send the resulting rows to the consumers



How Parallel Execution works

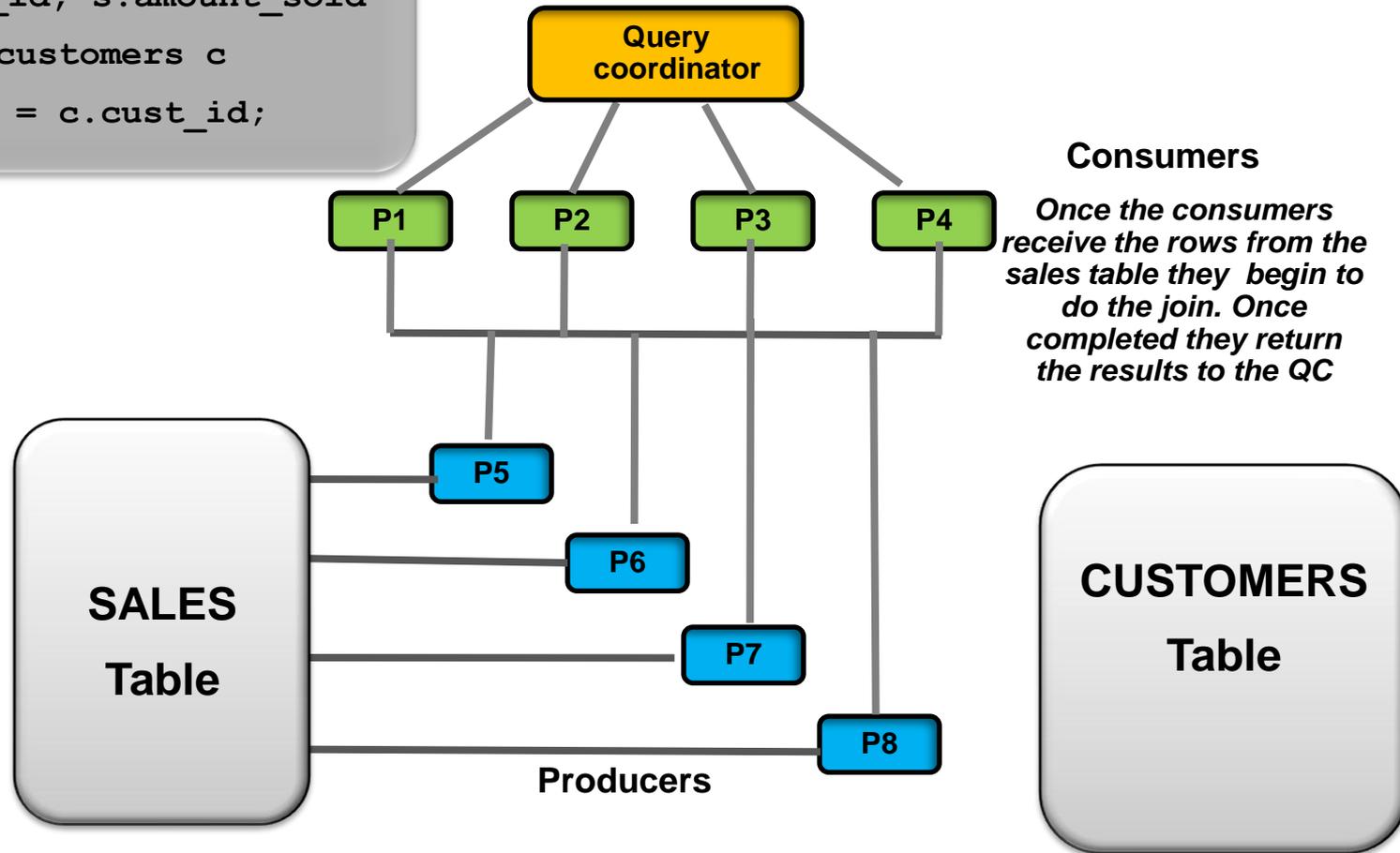
```
SELECT c.cust_last_name,  
       s.time_id, s.amount_sold  
FROM   sales s, customers c  
WHERE  s.cust_id = c.cust_id;
```



Once the 4 producers finish scanning the customer table, they start to scan the Sales table and send the resulting rows to the consumers

How Parallel Execution works

```
SELECT c.cust_last_name,  
       s.time_id, s.amount_sold  
FROM   sales s, customers c  
WHERE  s.cust_id = c.cust_id;
```



Monitoring Parallel Execution

```
SELECT c.cust_last_name, s.time_id, s.amount_sold
FROM sales s, customers c
WHERE s.cust_id = c.cust_id;
```

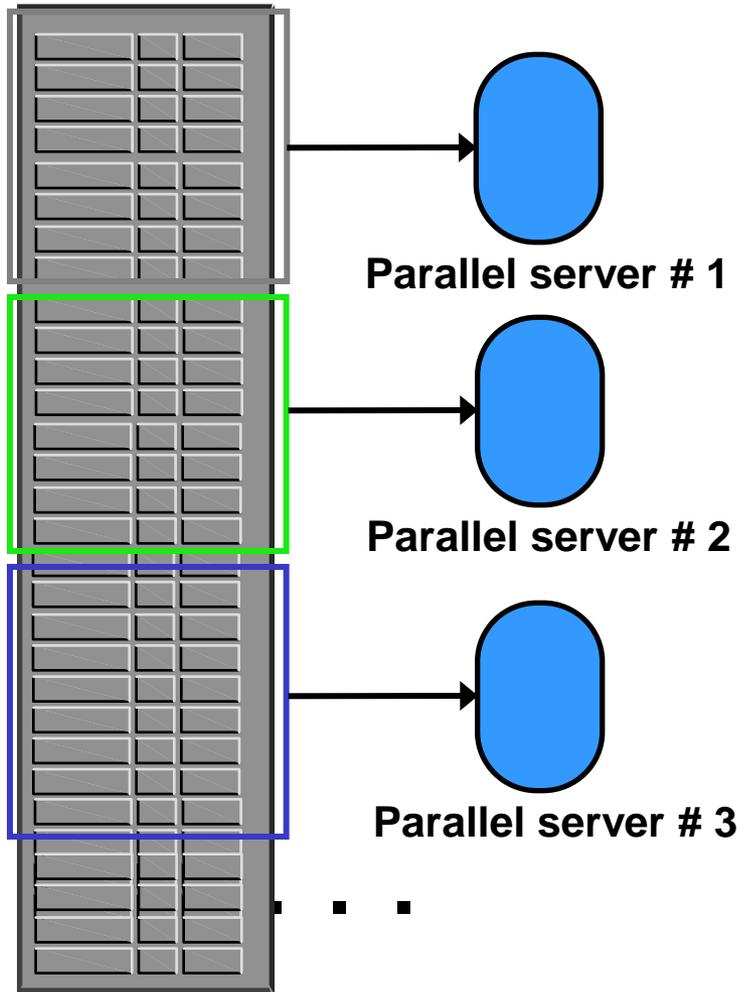
Consumers

Query Coordinator

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				311 (100)	
1	PX COORDINATOR					
2	PX SEND QC (RANDOM)	:TQ10002	1049K	31M	311 (2)	00:00:04
* 3	HASH JOIN BUFFERED		1049K	31M	311 (2)	00:00:04
4	PX RECEIVE		55500	704K	112 (0)	00:00:02
5	PX SEND HASH	:TQ10000	55500	704K	112 (0)	00:00:02
6	PX BLOCK ITERATOR		55500	704K	112 (0)	00:00:02
* 7	TABLE ACCESS FULL	CUSTOMERS	55500	704K	112 (0)	00:00:02
8	PX RECEIVE		1049K	18M	196 (2)	00:00:03
9	PX SEND HASH	:TQ10001	1049K	18M	196 (2)	00:00:03
10	PX BLOCK ITERATOR		1049K	18M	196 (2)	00:00:03
* 11	TABLE ACCESS FULL	SALES	1049K	18M	196 (2)	00:00:03

Producers

Oracle Parallel Query: Scanning a Table



- 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	IIN-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	PCMC	
* 8	TABLE ACCESS FULL	SALES	10M	85M	60 (97)	00:00:01	1	16	Q1,00	PCMP	

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	IIN-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	PCMC	
7	TABLE ACCESS BY LOCAL INDEX ROWID	SALES	26	234	1 (0)	00:00:01	1	16	Q1,00	PCMP	
* 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)

Best Practices for using Parallel Execution

Current Issues

- Difficult to determine ideal DOP for each table without manual tuning
- One DOP does not fit all queries touching an object
- Not enough PX server processes can result in statement running serial
- Too many PX server processes can thrash the system
- Only uses IO resources

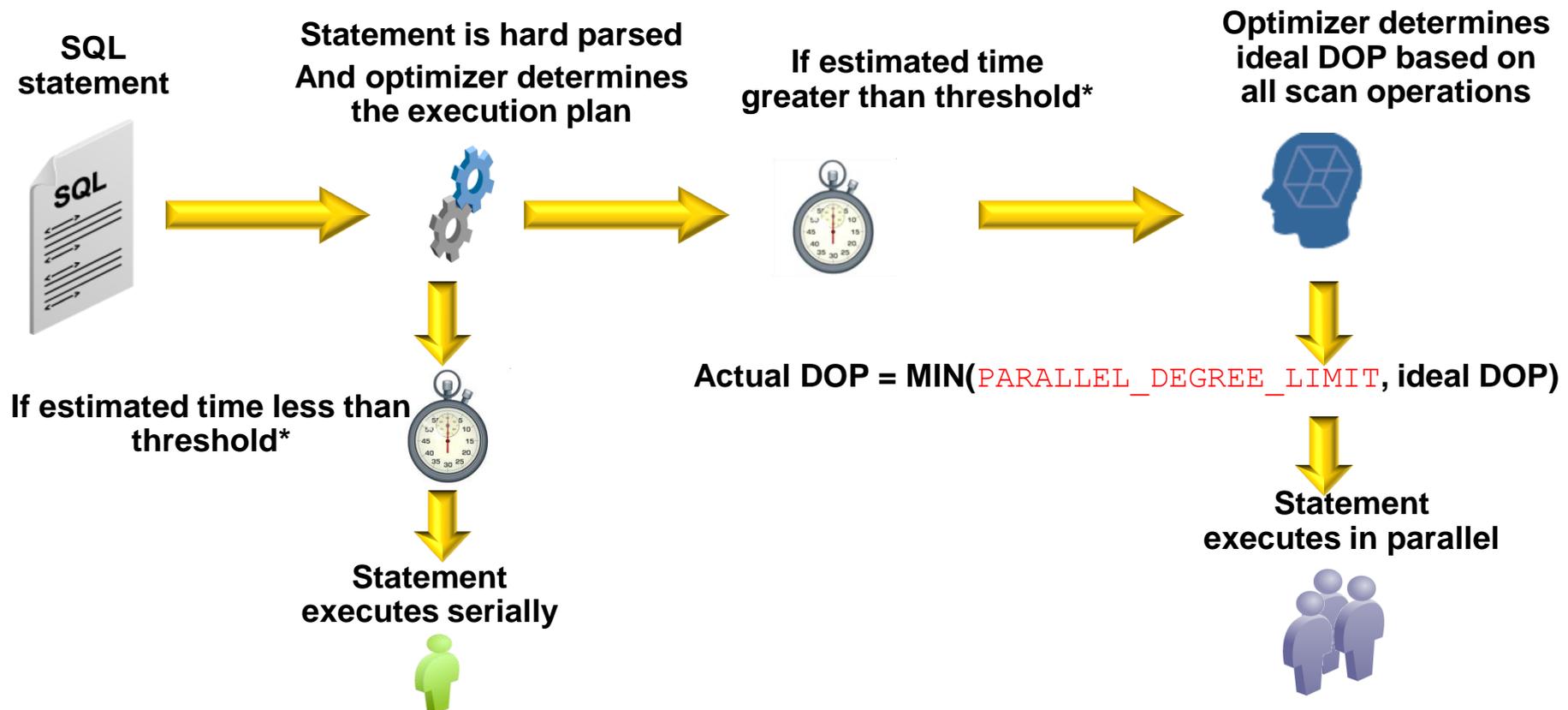
Solution

- Oracle automatically decides if a statement
 - Executes in parallel or not and what DOP it will use
 - Can execute immediately or will be queued
 - Will take advantage of aggregated cluster memory or not

Auto Degree of Parallelism

Enhancement addressing:

- Difficult to determine ideal DOP for each table without manual tuning
- One DOP does not fit all queries touching an object



Controlling Auto DOP (not queuing!)

- Controlled by three init.ora parameters:
 - PARALLEL_DEGREE_POLICY
 - Controls whether or not auto DOP will be used
 - Default is MANUAL which means no Auto DOP
 - Set to AUTO or LIMITED to enable auto DOP
 - PARALLEL_MIN_TIME_THRESHOLD
 - Controls which statements are candidate for parallelism
 - Default is 10 seconds
 - PARALLEL_DEGREE_LIMIT
 - Controls maximum DOP per statement
 - Default setting is the literal value “CPU” meaning DEFAULT DOP

What does this really mean?

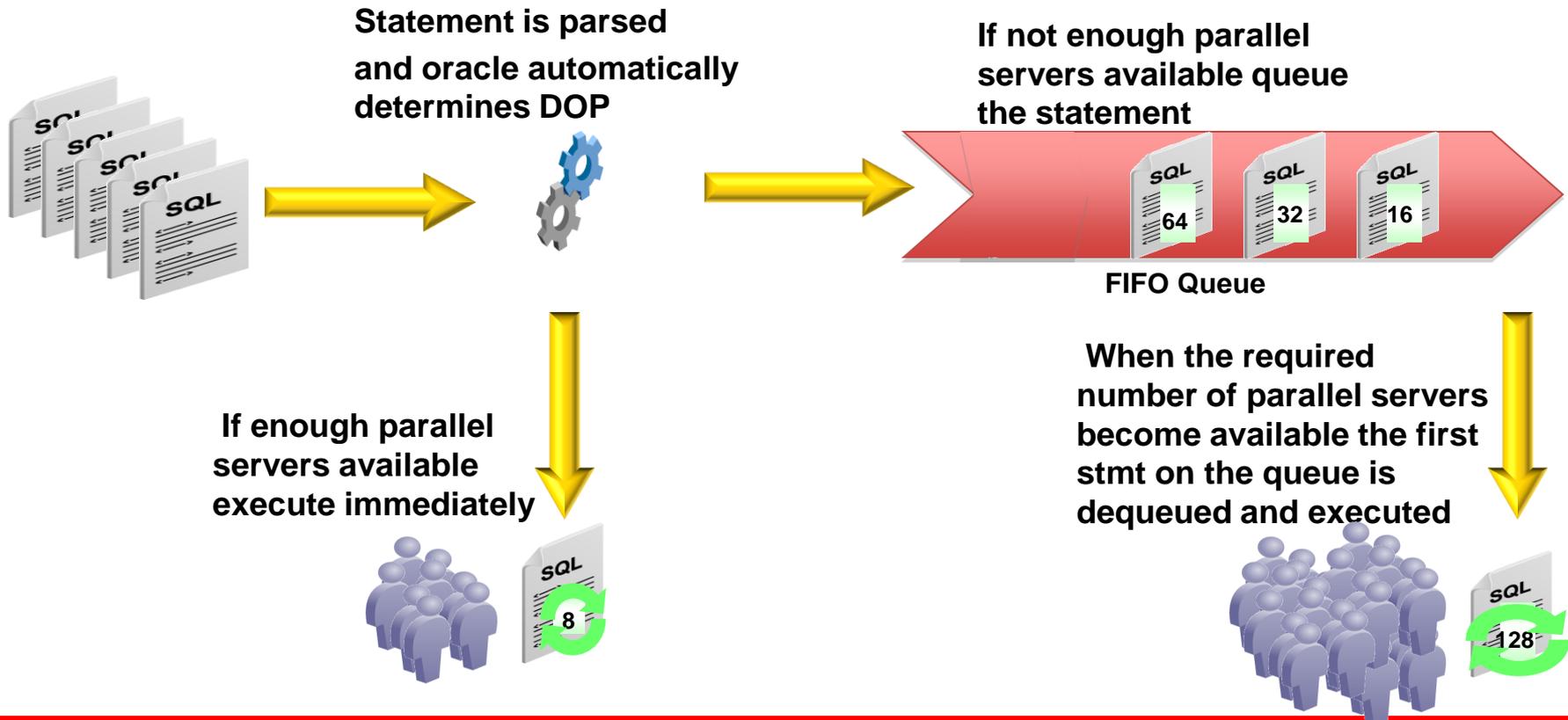
Working with Statement Queuing



Parallel Statement Queuing

Enhancement addressing:

- Not enough PX server processes can result in statement running serial
- Too many PX server processes can thrash the system



ORACLE

NOTE: Parallel_Servers_Target new parameter controls number of active PX processes before statement queuing kicks in

Parallel Statement Queuing

- **Benefits:**
 - Allows for higher DOPs per statement without thrashing the system
 - Allows a set of queries to run at roughly the same aggregate time by allowing the optimal DOP to be used all the time
- **Potential Costs:**
 - Adds delay to your execution time if your statement is queued making elapse times more unpredictable
- **Goal:**
 - Find the optimal queuing point based on desired concurrency

Parallel Statement Queuing

Calculating Minimal Concurrency based on processes

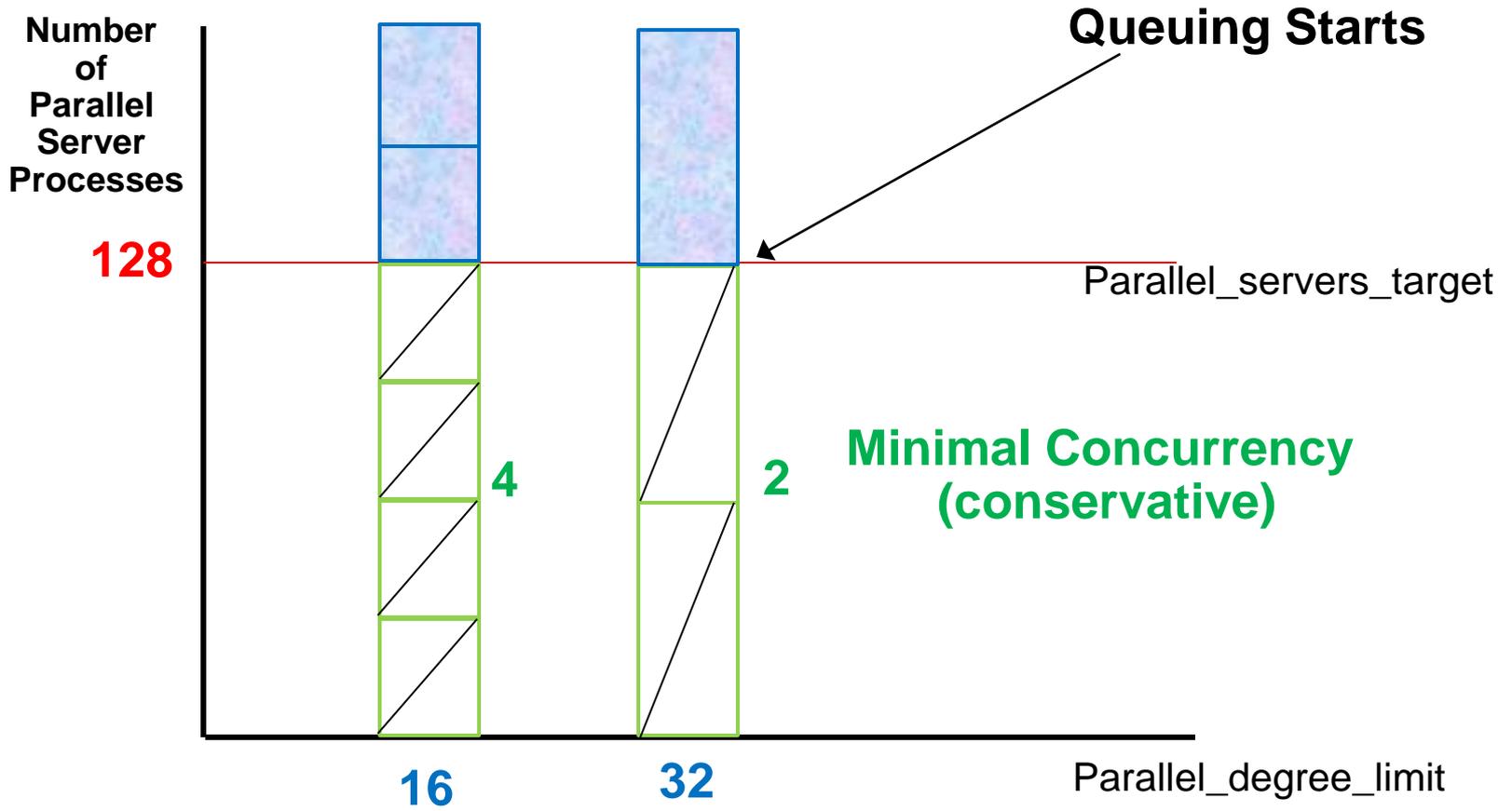
- Minimal concurrency is the minimal number of parallel statements than can run before queuing kicks in:

$$\text{minimal concurrency} = \frac{\text{Parallel_servers_target}}{\text{Parallel_degree_limit}} \times 0.5$$

- The conservative assumption is that you always have producers and consumers (and not PWJ all the time)

Parallel Statement Queuing

Real Minimal Concurrency



Crucial for “all” 11g R2 PX features

Parameter	Default Value	Description
PARALLEL_DEGREE_LIMIT	“CPU”	Max DOP that can be granted with Auto DOP
PARALLEL_DEGREE_POLICY	“MANUAL”	Specifies if Auto DOP, Queuing, & In-memory PE are enabled
PARALLEL_MIN_TIME_THRESHOLD	“AUTO”	Specifies min execution time a statement should have before AUTO DOP will kick in
PARALLEL_SERVERS_TARGET	4*CPU_COUNT* PARALLEL_THREAD S_PER_CPU * ACTIVE_INSTANCES	Specifies # of parallel processes allowed to run parallel stmts before queuing will be use

Enabling the 11g Features

INIT.ORA parameter PARALLEL_DEGREE_POLICY

Three possible modes:

- Manual
 - As before, DBA must manually specify all aspects of parallelism
 - No new features enabled
- Limited
 - Restricted AUTO DOP for queries with tables decorated with default PARALLEL
 - No Statement Queuing, No In-Memory Parallel Execution
- Auto
 - All qualifying statements subject to executing in parallel
 - DOP set on tables are ignored
 - Statements can be queued
 - In-memory PQ available

Parameter Hierarchy

1. Parallel_degree_policy = Manual

- a) None of the parameters have any impact

PX Features:

- NONE

2. Parallel_degree_policy = Limited

- a) Parallel_min_time_threshold = 10s
- b) Parallel_degree_limit = CPU

PX Features:

- Auto DOP

3. Parallel_degree_policy = Auto

- a) Parallel_min_time_threshold = 10s
- b) Parallel_degree_limit = CPU
- c) Parallel_servers_target = Set to Default DOP on Exadata

PX Features:

- Auto DOP
- Queuing
- In-Memory

SQL Monitoring screens

Overview

SQL ID	ffyk6r7yyz9nj	Time	Duration 1.3m	IO & Wait Statistics
Parallel	1f	Database Time	6.3m	IO Count 44K
Execution Started	Tue Mar 24 2009 06:14:13 PM	PL/SQL & Java	0.0s	Buffer Gets 589K
Last Refresh Time	Tue Mar 24 2009 06:15:29 PM			Wait Activity % 100
Execution ID	16777219			
Session	479			
Fetch Calls	0			

Details

Plan Statistics **Parallel** Activity

Plan Hash Value 3913711993

Click on parallel tab to get more info on PQ

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

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

Details

Plan Statistics Parallel Activity

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

Simple Example of Queuing

Queued stmts are indicated by the clock

Cluster Database: DBM >

Monitored SQL Executions

Active in last 2 hours

Status	Duration	Instance ID	SQL ID	User	Parallel	Database Time
	12.0s	2	7cf8uwfb0kmdg	RETAIL		11.6s
	13.0s	2	1phx4cp88a7v9	RETAIL		12.6s
	14.0s	2	4x1n1wbm4ujq3	RETAIL		13.8s
	15.0s	2	f6cfapghhspuc	RETAIL		14.9s
	16.0s	2	870q1dpvahgjk	RETAIL		15.1s
QUEUED	17.0s	2	d2gv1r08z3qsf	RETAIL		16.2s
	18.0s	2	gm0p0236yngxga	RETAIL		16.8s
	19.0s	2	8puusymuf2daq	RETAIL		17.8s
	20.0s	2	527a0nk94kgk1	RETAIL		18.8s
	21.0s	2	4adhvstzk6jnb	RETAIL		18.8s
	22.0s	2	fwm4pzgjqngg	RETAIL	56 8	11.6m
	23.0s	2	fgs5yggzfr38h	RETAIL	64 8	13.3m

8 Statements run before queuing kicks in

Preventing Extreme DOPs

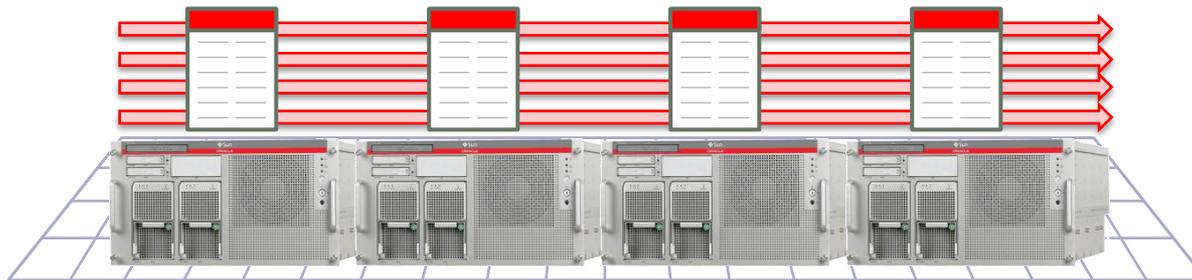
Setting a system wide parameter

- By setting `parallel_degree_limit` you CAP the maximum degree ANY statement can run with on the system
- Default setting is Default DOP which means no statement ever runs at a higher DOP than Default DOP
- Think of this as your safety net for when the magic fails and Auto DOP is reaching extreme levels of DOP

Note: EM will not show a downgrade for capped DOPs!

In-Memory Parallel Execution

Efficient use of memory on clustered servers



In-Memory Parallel Query in Database Tier

- **Compress more data into available memory on cluster**
- **Intelligent algorithm**
 - Places table fragments in memory on different nodes
- **Reduces disk IO and speeds query execution**

Agenda

- Parallel Execution
- Workload Management on a Data Warehouse
- Oracle Exadata Database machine



MIXED WORKLOAD: what does it mean?

Diverse workload running on a Data Warehouse system concurrently.

Examples:

- Continuous data loads while end users are querying data
- OLTP like activities (both queries and trickle data loads) mixed in with more classic ad-hoc data intensive query patterns.

Step 1: Understand the Workload

- Review the customer workload to find out:
 - Who is doing the work?
 - What types of work are done on the system?
 - When are certain types being done?
 - Where are performance problem areas?
 - What are the priorities, and do they change during a time window?
 - Are there priority conflicts?

Step 2: Map the Workload to the System

- Create the resource groups:
 - Map to users
 - Map to estimated execution time
 - Etc
- Create the required Resource Plans:
 - For example: Nighttime vs. daytime, online vs. offline
- Set the overall priorities
 - Which resource group gets most resources
 - Cap max utilizations
- Drill down into parallelism, queuing and session throttles

Why use Resource Manager?

- Manage workloads contending for CPU
- Prevent excessive CPU load, destabilizing server
- Manage parallel query processes, queuing and concurrency
- Prevent runaway queries
- Manage workloads contending for I/O

Resource Manager User Interface



Cluster Database: DBM

- [Home](#)
- [Performance](#)
- [Availability](#)
- Server**
- [Schema](#)
- [Data Movement](#)
- [Software and Support](#)
- [Topology](#)

Storage

- [Control Files](#)
- [Tablespaces](#)
- [Temporary Tablespace Groups](#)
- [Datafiles](#)
- [Rollback Segments](#)
- [Redo Log Groups](#)
- [Archive Logs](#)
- [Migrate to ASM](#)
- [Make Tablespace Locally Managed](#)

Statistics Management

- [Automatic Workload Repository](#)
- [AWR Baselines](#)

Database Configuration

- [Initialization Parameters](#)
- [View Database Feature Usage](#)

Resource Manager

- [Getting Started](#)
- [Consumer Groups](#)
- [Consumer Group Mappings](#)
- [Plans](#)
- [Settings](#)
- [Statistics](#)
- [Parallel Statement Queue](#)

New!

Working example of Workload management using DBRM

Step 1: Understand your workload

Working Example:

- User 1: **RTL** runs long running analytical queries.
- User 2: **RT_CRITICAL** is a “SUPER” business user run short critical queries various times of the day.

GOAL:

Ensure the Critical queries will run in a consistent and timely manner even when the system is loaded with batch analytical queries.

DBRM – RESOURCE GROUP CRITICAL SHORT QUERIES **STEP 1**

ORACLE Enterprise Manager 11g
Database Control

Setup Preferences Help Logout
Cluster Database

Cluster Database: dbm > Consumer Groups >

Logged in As SYS

View Resource Consumer Group: RT_CRITICAL

Actions Create Like Go Edit Return

Consumer Group RT_CRITICAL
Description CRITICAL BUSINESS QUERIES
Scheduling Policy Round Robin

Users permitted to run in this Consumer Group

User	Admin Option
RT_CRITICAL	<input checked="" type="checkbox"/>

Actions Create Like Go Edit Return

TWO CONSUMER GROUPS :

- RT_CRITICAL
- RT_ANALYSIS

Consumer Group Mappings

Cluster Database: dbm >

Logged in As SYS

Consumer Group Mappings

Show SQL Revert Apply

General Priorities

Create rules to enable the resource manager to automatically assign sessions to consumer groups

View

Add Rule for Selected Type

Select	Priority Δ	View	Value	Consumer Group	Remove
<input checked="" type="radio"/>	1	Service Module and Action	No Mappings Specified		
<input type="radio"/>	2	Service and Module	No Mappings Specified		
<input type="radio"/>	3	Module and Action	No Mappings Specified		
<input type="radio"/>	4	Module	No Mappings Specified		
<input type="radio"/>	5	Service	No Mappings Specified		
<input type="radio"/>	6	Oracle User	RTL	 RT_ANALYSIS	
			RT_CRITICAL	 RT_CRITICAL	

- Map RTL USER to consumer group RT_ANALYSIS
- Map RT_CRITICAL to consumer group RT_CRITICAL

Step 2 CREATE RESOURCE PLAN

ORACLE Enterprise Manager 11g Database Control

Cluster Database: dbm > Resource Plans > Edit Resource Plan: ANALYSIS_BATCH_PLAN

Actions: Create Like (dropdown) Go Show SQL Revert Apply

General Parallelism Session Pool Undo Pool Thresholds Idle Time

A Resource Plan contains directives that specify how resources are allocated to Consumer Groups. For each Consumer Group, a directive specifies the amount of CPU resources are allocated. It also specifies limits, such as the maximum degree of parallelism, execution time, and amount of I/O, that each session in the Consumer Group can consume. You can enable a Resource Plan manually or automatically, using Scheduler Windows.

Plan: ANALYSIS_BATCH_PLAN
Description: ANALYSIS AND BATCH QUERIES

Instances: (none) Edit
If any instances are in this list, the Plan is active for those instances.

Resource Allocations
Mode: Percentage Advanced

Group/Subplan	Max Utilization Limit	Percentage
OTHER_GROUPS	<input type="text"/>	<input type="text"/>
RT_ANALYSIS	<input type="text"/>	<input type="text"/>
RT_CRITICAL	<input type="text"/>	<input type="text"/>

General Parallelism Session Pool Undo Pool Thresholds Idle Time

Created two plans:

- ANALYSIS_BATCH_PLAN
- Critical_BATCH_PLAN

Configure Parallel Execution

- Manage parallel execution resources and priorities by consumer groups

The screenshot shows the 'Edit Resource Plan: ANALYSIS_BA' configuration page in Oracle Enterprise Manager 11g. The 'Parallelism' tab is active, displaying a table with the following data:

Group	Max Degree of Parallelism	Max Percentage of Parallel Servers Target	Parallel Queue Timeout
OTHER_GROUPS	UNLIMITED	UNLIMITED	UNLIMITED
RT_ANALYSIS	64	70	UNLIMITED
RT_CRITICAL	UNLIMITED	UNLIMITED	UNLIMITED

Three callout boxes provide context for the settings:

- Limit the max DOP a consumer group can use:** Points to the 'Max Degree of Parallelism' column, specifically the value '64' for the 'RT_ANALYSIS' group.
- Determine when queuing will start per group:** Points to the 'Max Percentage of Parallel Servers Target' column, specifically the value '70' for the 'RT_ANALYSIS' group.
- Specify a queue timeout per group:** Points to the 'Parallel Queue Timeout' column, which is 'UNLIMITED' for all groups.

Allocate resources DAYTIME_CRITICAL_QUERIES_PLAN

ORACLE Enterprise Manager 11g Database Control

Cluster Database: dbm > Resource Plans > View Resource Plan: DAYTIME_CRITICAL_QUERIES_PLAN

Actions: Create Like Go Edit Return

Plan: DAYTIME_CRITICAL_QUERIES_PLAN
 Description: CRITICAL QUERIES
 Instances: (none)
 Automatic Plan Switching Enabled: false
 Is Subplan: false
 Associated Scheduler Window(s):

Resource Allocations

Group/Subplan	Max Utilization Limit	Percentage
OTHER_GROUPS		
RT_ANALYSIS		
RT_CRITICAL		

Directive Values

Group	Max Degree of Parallelism	Max Percentage of Parallel Servers Target	Parallel Queue Timeout	Max Number of Active Sessions	Activation Queue Timeout (sec)	Max Undo Space (KB)	Max Estimated Execution Time (sec)	Max Idle Time (sec)	Max Idle Time if Blocking Another Session (sec)
OTHER_GROUPS	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED
RT_ANALYSIS	16	20	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED
RT_CRITICAL	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED

Statements issued by RTL_ANALYSIS:

DOP is capped at 16

Statements will queue once the number of Max parallel processes $204 = 20/100 * 1024 (1024)$.

Allocate resources

ORACLE Enterprise Manager 11g Database Control

Cluster Database: dbm > Resource Plans > View Resource Plan: ANALYSIS_BATCH_PLAN

Plan: ANALYSIS_BATCH_PLAN
 Description: ANALYSIS AND BATCH QUERIES
 Instances: (none)
 Automatic Plan Switching Enabled: false
 Is Subplan: false
 Associated Scheduler Window(s):

Actions: Create Like [v] Go Edit Return

Resource Allocations

Group/Subplan	Max Utilization Limit	Percentage
OTHER_GROUPS		
RT_ANALYSIS		
RT_CRITICAL		

Directive Values

Group	Max Degree of Parallelism	Max Percentage of Parallel Servers Target	Parallel Queue Timeout	Max Number of Active Sessions	Activation Queue Timeout (sec)	Max Undo Space (KB)	Max Estimated Execution Time (sec)	Max Idle Time (sec)	Max Idle Time if Blocking Another Session (sec)
OTHER_GROUPS	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED
RT_ANALYSIS	64	70	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED
RT_CRITICAL	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED

RTL _ANALYSIS GROUP : MAX DOP capped at 64 and queuing will start once processes reach 70% of Parallel_server_Target

UNLIMITED: MAX DOP is capped by Parallel_Degree_Limit

Example long-running query

No load on the system and no resource mgmt activated

ORACLE Enterprise Manager 11g Database Control

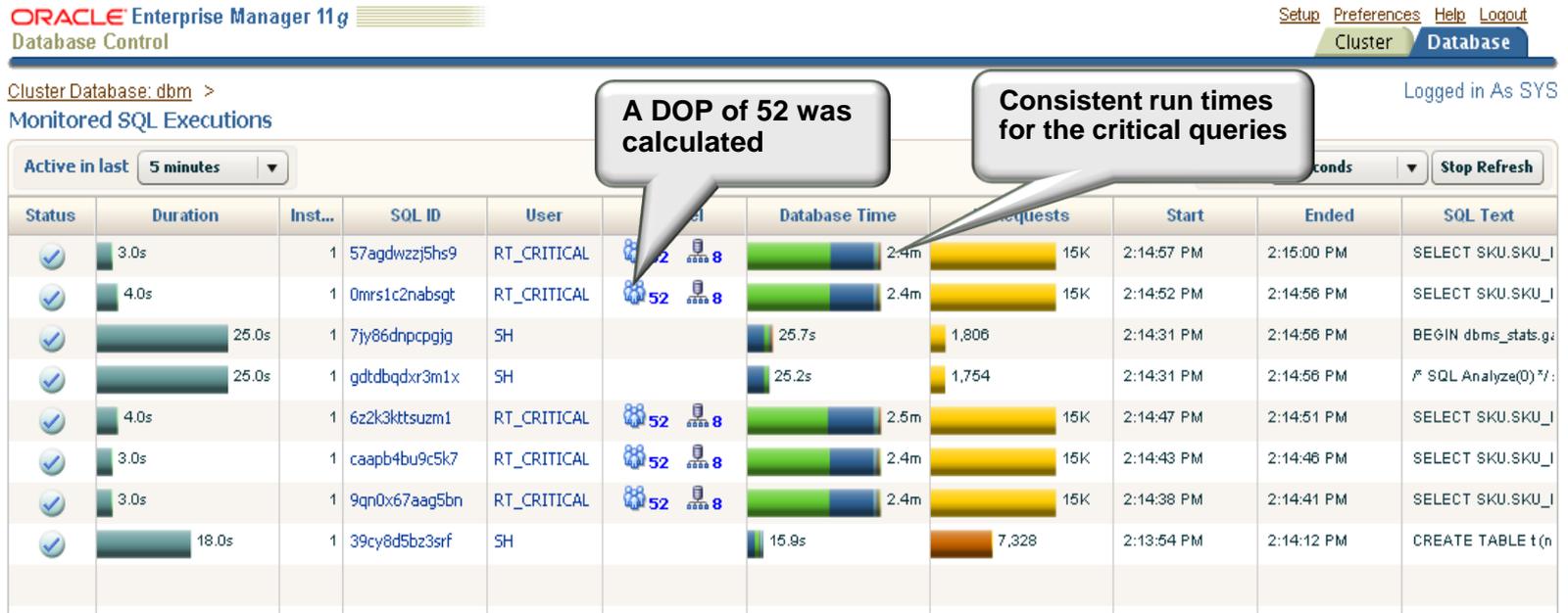
Cluster Database: d1m > Monitored SQL Executions

Active in last 24 hours

Status	Duration	Instance ID	SQL ID	User	Parallel	Database Time	IO Requests	Start	Ended	SQL Text
	25.0s	1	36rybsduxng9n	RTL	128	33.4m	125K	8:14:01 AM		WITH QBUY AS (SELECT P+ P)
	3.2h	8	db58cdjmeu09w	SYS		3.2h		8:00:16 AM		SQL Analyze(1) ?selected ?+ it
	36.0s	1	36rybsduxng9n	RTL	128	40.2m	142K	8:13:07 AM	8:13:44 AM	WITH QBUY AS (SELECT P+ P)
	24.0s	8	a322sqg8b0948	SYS		12.6s	1,714	8:00:04 AM	8:00:28 AM	DECLARE job BINARY_INTEGE
	22.0s	1	a322sqg8b0948	SYS		11.9s	1,714	8:00:04 AM	8:00:28 AM	DECLARE job BINARY_INTEGE
	15.0s	8	5aruc4v6y32f9	SYS		15.0s	9,852	8:00:08 AM	8:00:23 AM	DECLARE job BINARY_INTEGE
	19.0s	3	a322sqg8b0948	SYS		11.8s	1,714	8:00:04 AM	8:00:23 AM	DECLARE job BINARY_INTEGE
	17.0s	5	a322sqg8b0948	SYS		10.0s	1,714	8:00:04 AM	8:00:21 AM	DECLARE job BINARY_INTEGE
	15.0s	2	a322sqg8b0948	SYS		11.5s	1,714	8:00:04 AM	8:00:19 AM	DECLARE job BINARY_INTEGE

- Automatic DOP of 128 chosen by the system
 - Does not ensure requested concurrency

Critical Queries with no load on the system and no plan activated



- NO RESOURCE PLAN ACTIVATED
- NO LOAD ON THE SYSTEM
- DOP OF 52 CALCULATED BY THE SYSTEM

Scenario we want to avoid !!

Critical queries are queued

ORACLE Enterprise Manager 11g
Database Control

Cluster Database: dbm > Logged in As SYS

Monitored SQL Executions

Active in last 5 minutes Refresh 5 seconds Stop Refresh

Status	Duration	Instance ID	SQL ID	User	Parallel	Database Time	IO	Start	Ended	SQL Text
	55.0s	1	0patuy4jv38nj	RTL	64	31.6m	58K	2:24:09 PM		WITH QBUY AS (SE
	42.0s	1	0v4mpxxhj5q56	RTL	64	21.5m	50K	2:24:22 PM		WITH QBUY AS (SE
	27.0s	1	1a6c80qra5340	RTL		26.9s		2:24:38 PM		WITH QBUY AS (SE
	59.0s	1	4c3vty54w2ydx	RTL	64	35	60K	2:24:05 PM		WITH QBUY AS (SE
	38.0s	1	4w1c2sdjh8aas	RTL	64	13.8m	42K	2:24:26 PM		WITH QBUY AS (SE
	27.0s	1	6z2k3ktsuzm1	RT_CRITICAL		26.4s		2:24:38 PM		SELECT SKU.SKU_
	50.0s	1	95udgkar4mwsy	RTL	64	29.9m	55K	2:24:14 PM		WITH QBUY AS (SE
	34.0s	1	9p169bb9f90cj	RTL	64	10.5m	31K	2:24:30 PM		WITH QBUY AS (SE
	9.0s	1	9qn0x67aag5bn	RT_CRITICAL	52	3.6m	15K	2:24:07 PM	2:24:16 PM	SELECT SKU.SKU_
	18.0s	1	caapb4bu9c5k7	RT_CRITICAL	52	6.5m	19K	2:24:18 PM	2:24:36 PM	SELECT SKU.SKU_
	47.0s	1	gnda8a80karmg	RTL	64	27.2m	61K	2:24:17 PM		WITH QBUY AS (SE
	30.0s	1	gyayhqycxpz17	RTL	64	4.3m	15K	2:24:34 PM		WITH QBUY AS (SE
	44.0s	1	gqr3n55fabxa3	RTL	64	20.2m	57K	2:24:01 PM	2:24:45 PM	WITH QBUY AS (SE

When the System is loaded with long running analytical statement
CRITICAL QUERIES are queued and have inconsistent elapsed times

CRITICAL_QUERIES_DAY_PLAN ACTIVATED

ORACLE Enterprise Manager 11g Database Control

Cluster Database: dbm > Monitored SQL Executions

Active in last 5 minutes Refresh 5 seconds Stop Refresh

Status	Duration	Inst...	SQL ID	User	Parallel	Database Time	IO Requests	Start	Ended	SQL Text
	4.0s	1	caapb4bu9c5k7	RT_CRITICAL	52	2.0m	13K	2:38:23 PM		SELECT SKU.SKU_I
	14.7s			RTL		14.7s		2:38:12 PM		WITH QBUY AS (SE
	18.7s			RTL		18.7s		2:38:08 PM		WITH QBUY AS (SE
	20.2s			RTL		20.2s		2:38:07 PM		WITH QBUY AS (SE
	22.8s			RTL		22.8s		2:38:04 PM		WITH QBUY AS (SE
	27.0s			RTL		27.0s		2:38:00 PM		WITH QBUY AS (SE
	31.0s			RTL		31.0s		2:37:56 PM		WITH QBUY AS (SE
	33.5s			RTL	16	33.5s		2:37:52 PM		WITH QBUY AS (SE
	6.4m			RTL	16	6.4m	43K	2:37:48 PM		WITH QBUY AS (SE
	6.8m			RTL	16	6.8m	50K	2:37:44 PM		WITH QBUY AS (SE
	7.7m			RTL	16	7.7m	49K	2:37:40 PM		WITH QBUY AS (SE
	8.2m			RTL	16	8.2m	51K	2:37:36 PM		WITH QBUY AS (SE
	9.1m			RTL	16	9.1m	55K	2:37:32 PM		WITH QBUY AS (SE
	9.6m			RTL	16	9.6m	57K	2:37:28 PM	2:38:26 PM	WITH QBUY AS (SE

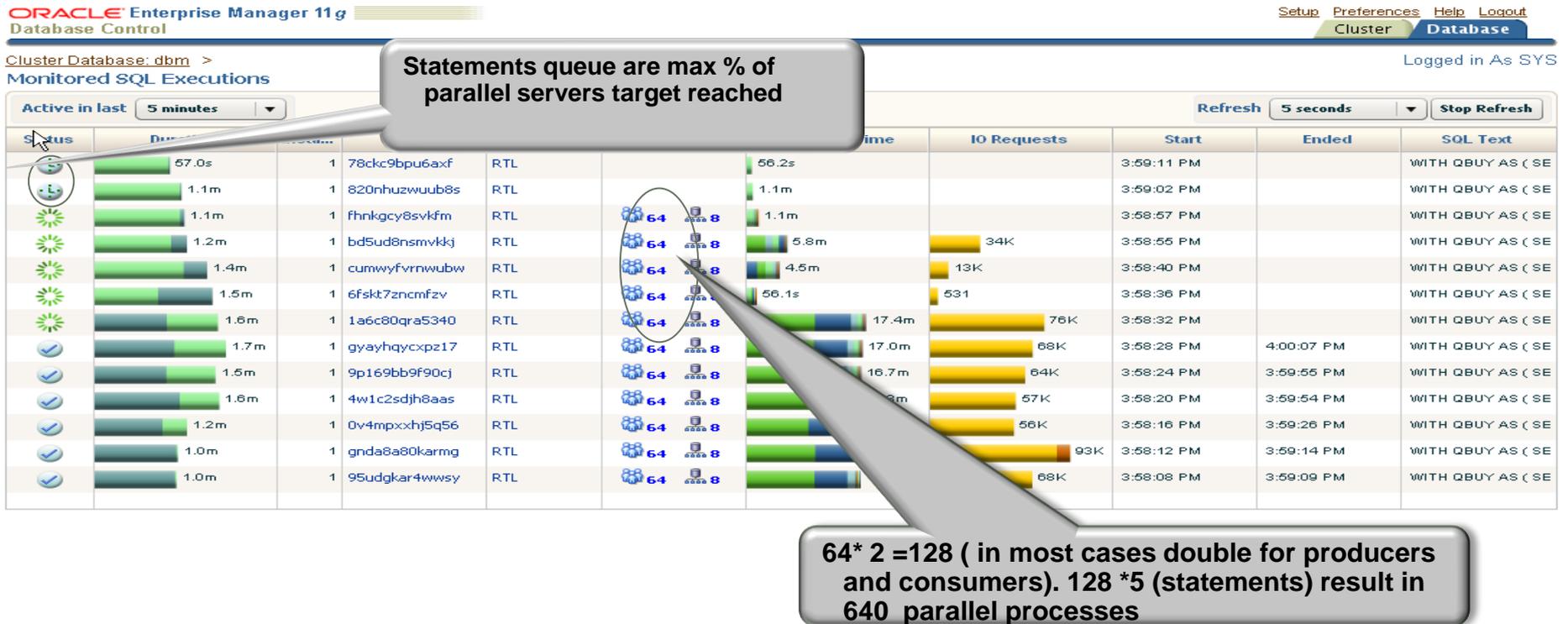
Annotations:

- Critical users statements get priority
- User RTL statements are queued
- only six RTL statements running
- Red arrow indicates DOP downgraded

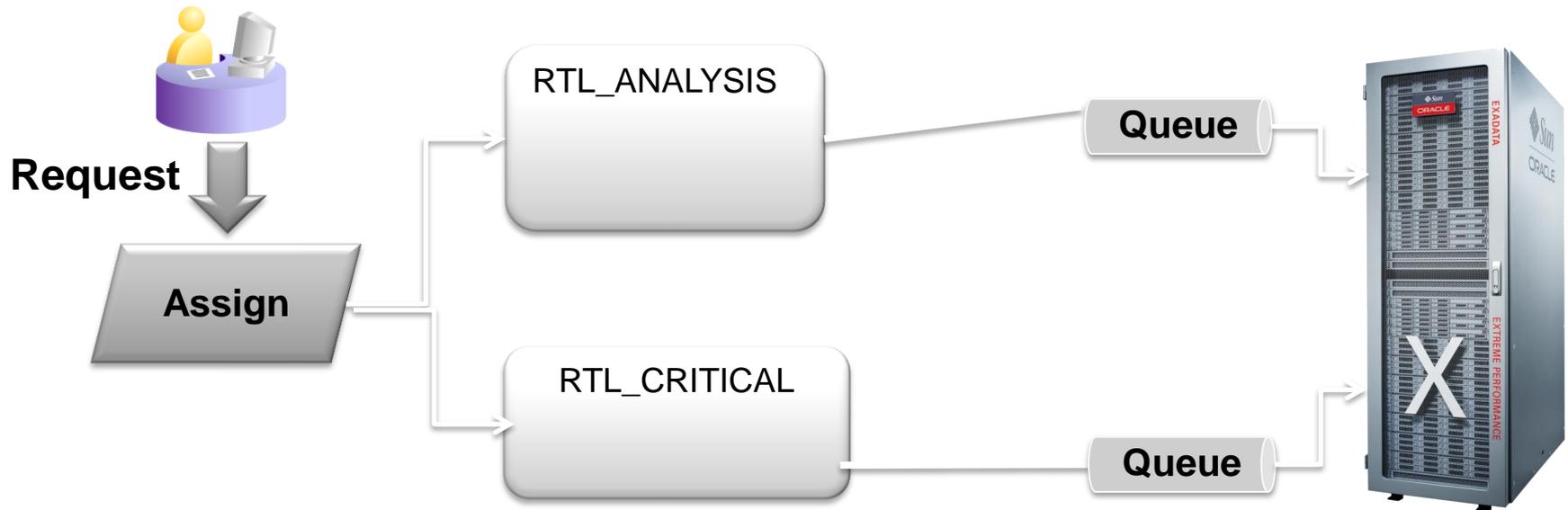
- Why are statements queued when only six statements are running? $6 * 16 = 96$
- RTL _ANLAYSIS group is allowed to up to $20\% \text{ of } 1024 = 204$
- ANSWER: STATEMENT REQUIRES TWO SETS OF PX PROCESS FOR PRODUCERS & CONSUMERS
- $32 * 6 = 192$.. the seventh query would take it over total allowed .. $192 + 32 = 224$

Analysis Batch Plan

- RT_Analaysis group : Max Percentage of parallel servers target = 70% = $70 * 1024 / 100 = 719$

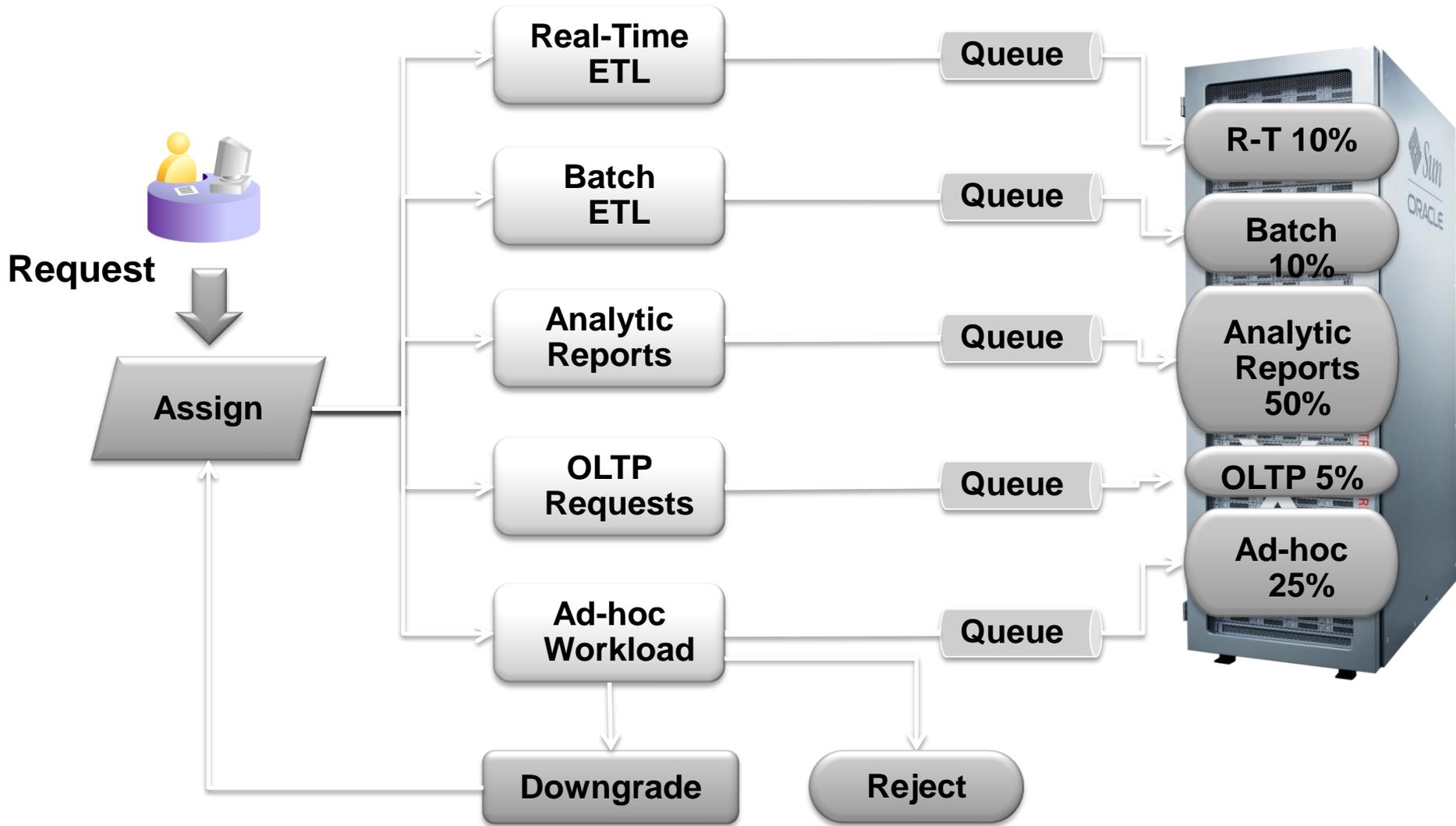


Resource Manager - Statement Queuing



- Queuing is embedded with DBRM
- One queue per consumer group

Workload Management



Step 3: Run and Adjust the Workload

- Run a workload for a period of time and look at the results
- DBRM Adjust:
 - Overall priorities
 - Scheduling of switches in plans
 - Queuing
- System Adjust:
 - How many PX statements
 - PX Queuing levels vs. Utilization levels (should we queue less?)

Agenda

- Parallel Execution
- Workload Management on a Data Warehouse
- Oracle Exadata Database machine



Oracle Exadata Database Machine Family

Oracle Exadata Database Machine X2-8



Oracle Database Server Grid

- 2 8-processor Database Servers
 - 128 CPU Cores
 - 2 TB Memory
 - Oracle Linux or Solaris 11 Express

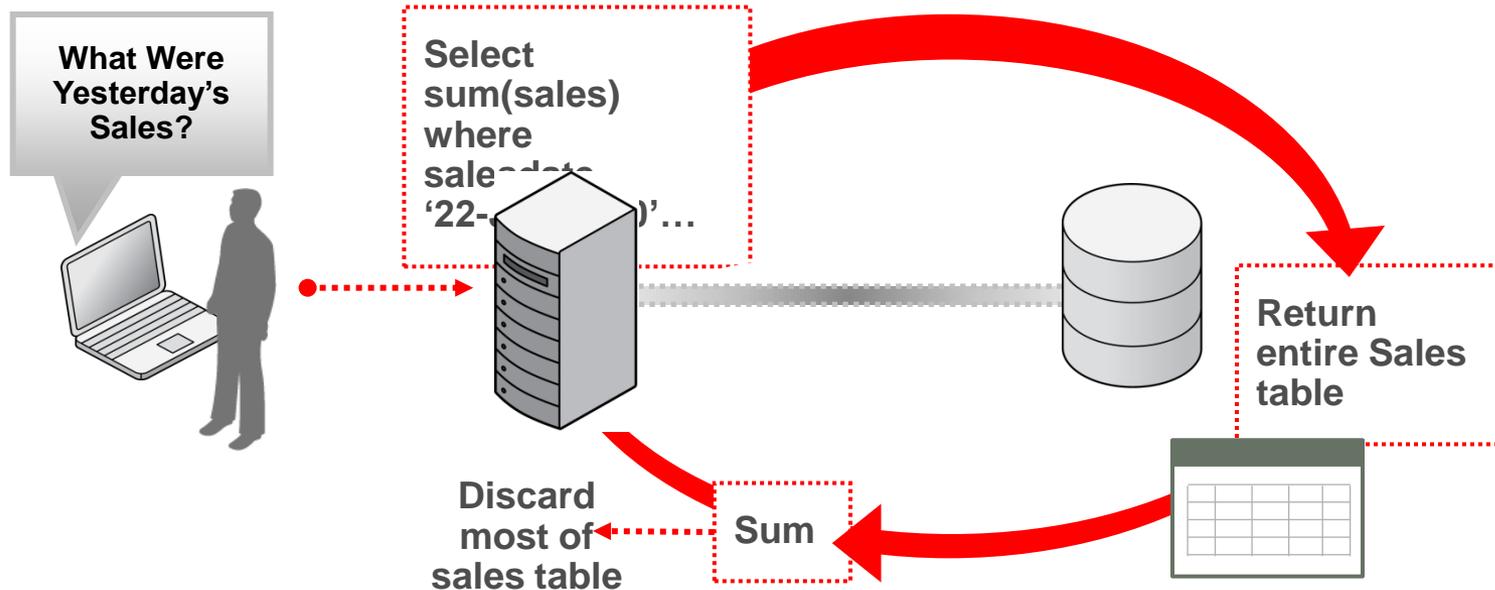
Exadata Storage Server Grid

- 14 Storage Servers
 - 5 TB Smart Flash Cache
 - 336 TB Disk Storage

Unified Server/Storage Network

- 40 Gb/sec Infiniband Links

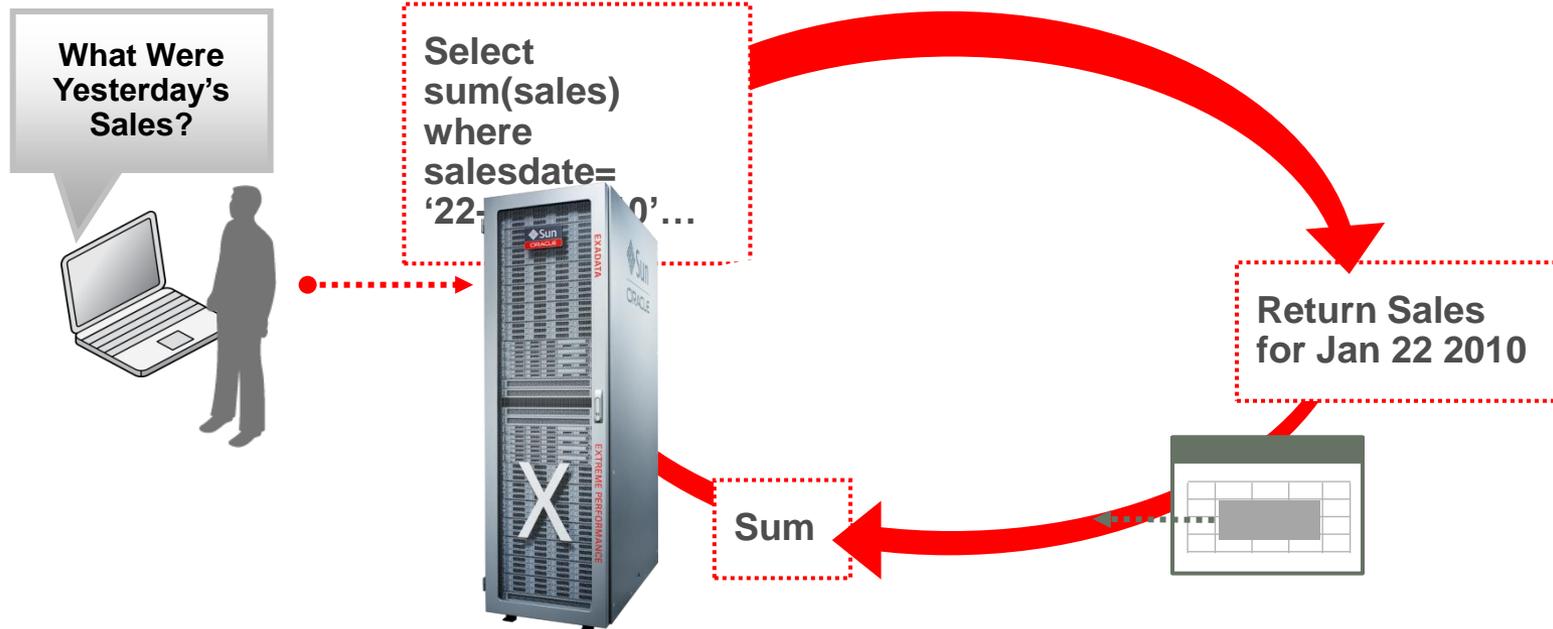
Traditional Query Problem



- Data is pushed to database server for processing
- I/O rates are limited by speed and number of disk drives
- Network bandwidth is strained, limiting performance and concurrency

Exadata Smart Scan

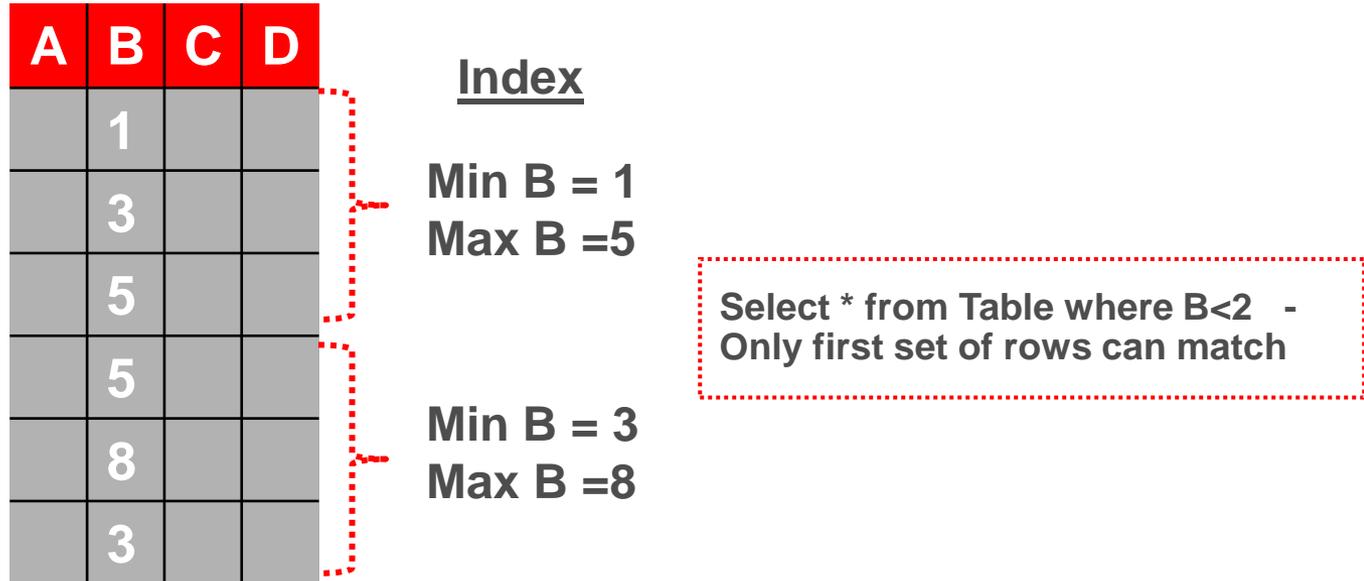
Improve Query Performance by 10x or More



- Off-load data intensive processing to Exadata Storage Server
- Exadata Storage Server only returns relevant rows and columns
- Wide Infiniband connections eliminate network bottlenecks

Exadata Storage Index

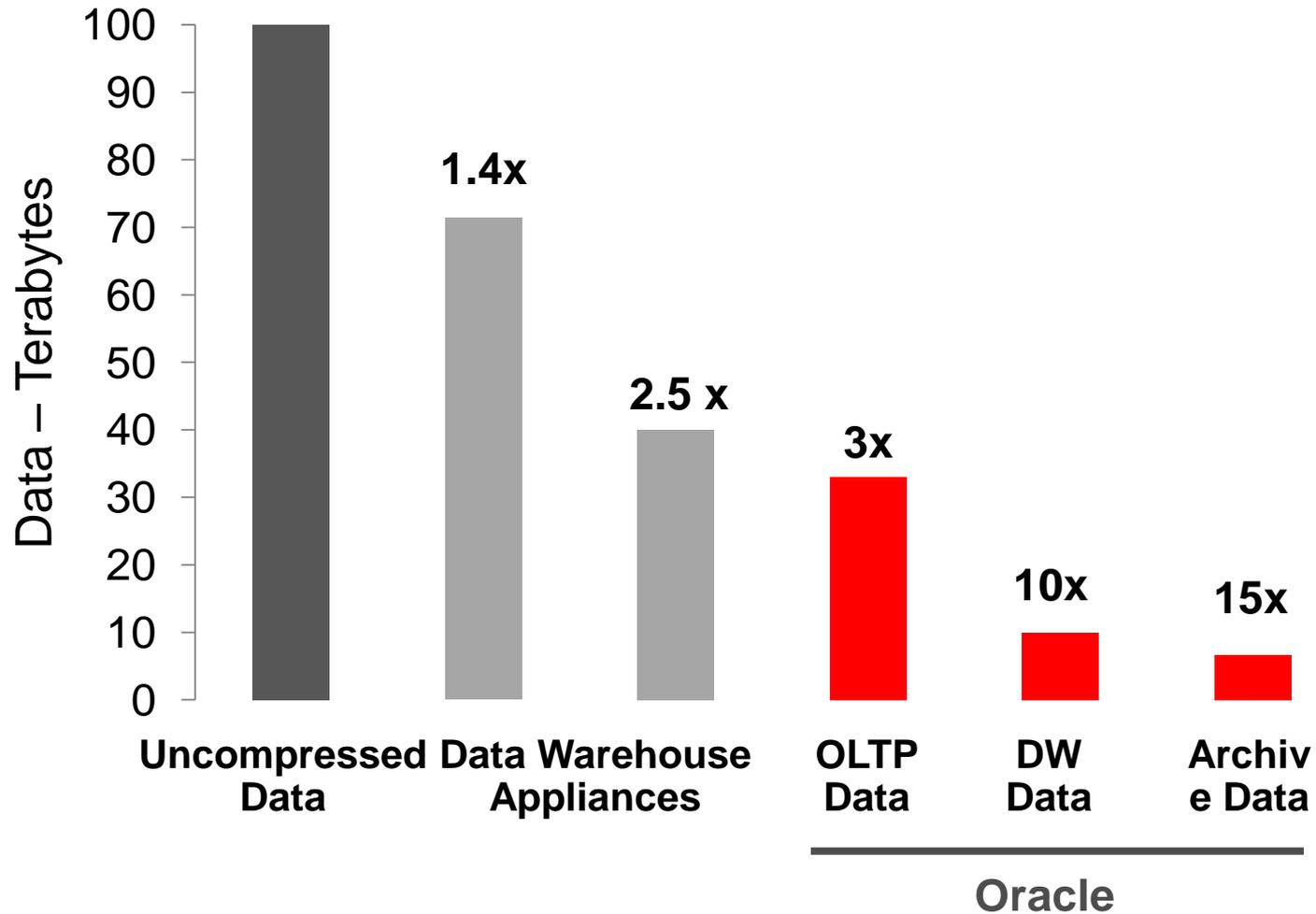
Transparent I/O Elimination with No Overhead



- Maintain summary information about table data in memory
- Eliminate disk I/Os if MIN / MAX never match “where” clause
- Completely automatic and transparent

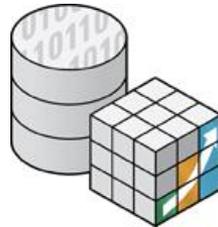
Exadata Hybrid Columnar Compression

Reduce Disk Space Requirements



Built-in Analytics

Secure, Scalable Platform for Advanced Analytics



Oracle OLAP

Analyze and summarize



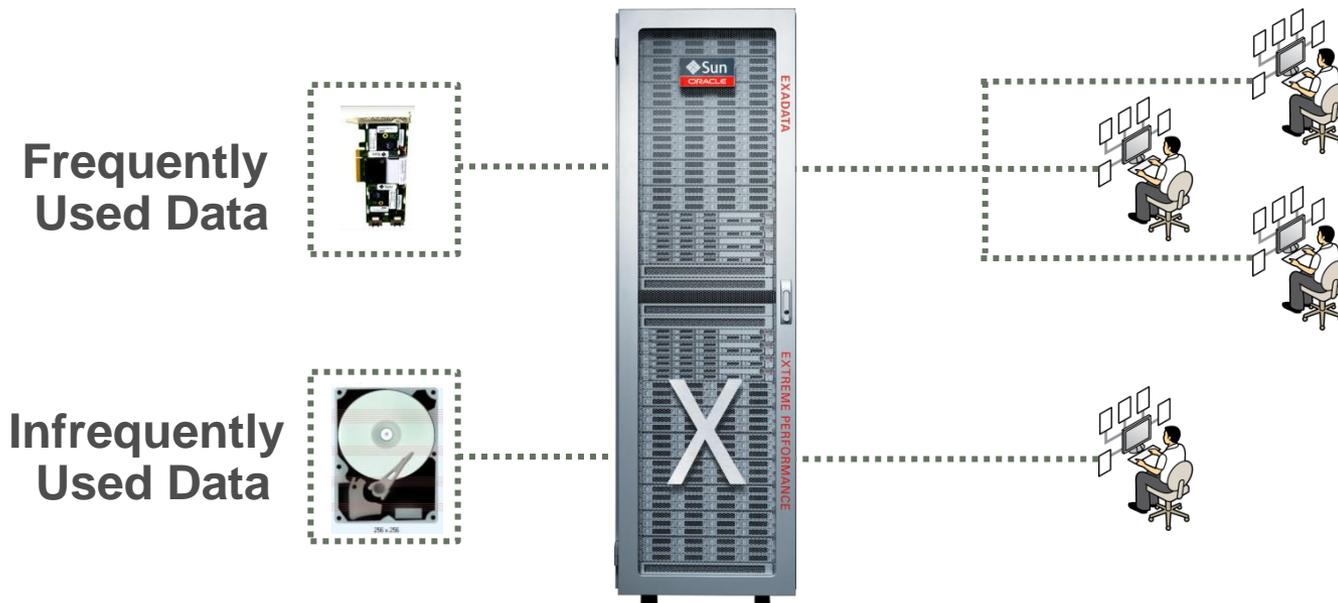
Oracle Data Mining

Uncover and predict

- Complex and predictive analytics embedded into Oracle Database 11g
- Reduce cost of additional hardware, management resources
- Improve performance by eliminating data movement and duplication

Exadata Smart Flash Cache

Extreme Performance for OLTP Applications



- Automatically caches frequently-accessed 'hot' data in flash storage
- Assigns the rest to less expensive disk drives
- Know when to avoid trying to cache data that will never be reused
- Process data at 50GB/sec and up to 1million I/Os per second

Benefits Multiply

Converting Terabytes to Gigabytes



10 TB of User Data

1 TB of User Data

100 GB of User Data

With 10x Compression

With Partition Pruning



20 GB of User Data

5 GB of User Data

Sub second "10 TB" Scan

With Storage Indexes

With Smart Scan

No Indexes



ORACLE

Turkcell Runs 10x Faster on Exadata

Compresses Data Warehouse by 10x

- Replaced high-end SMP Server and 10 Storage Cabinets
- Reduced Data Warehouse from 250TB to 27TB
 - Using OLTP & Hybrid Columnar Compression
 - Ready for future growth where data doubles every year
- Experiencing 10x faster query performance
 - Delivering over 50,000 reports per month
 - Average report runs reduced from 27 to 2.5 mins
 - Up to 400x performance gain on some reports

Summary



- Implement the three Ps of Data Warehousing
 - Power – balanced hardware configuration
 - Make sure the system can deliver your SLA
 - Partitioning – Performance, Manageability, ILM
 - Make sure partition pruning and partition-wise joins occur
 - Parallel – Maximize the number of processes working
 - Make sure the system is not flooded using DOP limits & queuing
- Workload Management on a Data Warehouse
 - Use Database Resource Manager
 - Control maximum DOP each user can have
 - Control when statements should begin queue
 - Control what happens to “run away” queries

Learn More

Parallel Execution

- Oracle University Class
 - Choose between live virtual class or instructor-led
 - http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getCourseDesc?dc=D71882GC10

Read our blogs:

<http://blogs.oracle.com/optimizer>

<http://blogs.oracle.com/datawarehousing>

Best practices papers can be found here:

<http://www.oracle.com/technetwork/database/focus-areas/bi-datawarehousing/index.html>

Q & A

