Database Consolidation: Resource Management Best Practices

OOW Session CON6581

October 1–5, 2017
SAN FRANCISCO, CA

Sue K. Lee
sue.k.lee@oracle.com
Director of Development
Oracle

Marty Stogsdill
Senior Manager, Oracle Architecture and Engineered Systems
martinstogsdill@discover.com
Discover Financial Services

Safe Harbor Statement

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.

Program Agenda

1 Consolidation Concepts

2 Resource Controls

3 Managing a Cloud

4 Use Case: Discover

ORACLE

Program Agenda

1 Consolidation Concepts

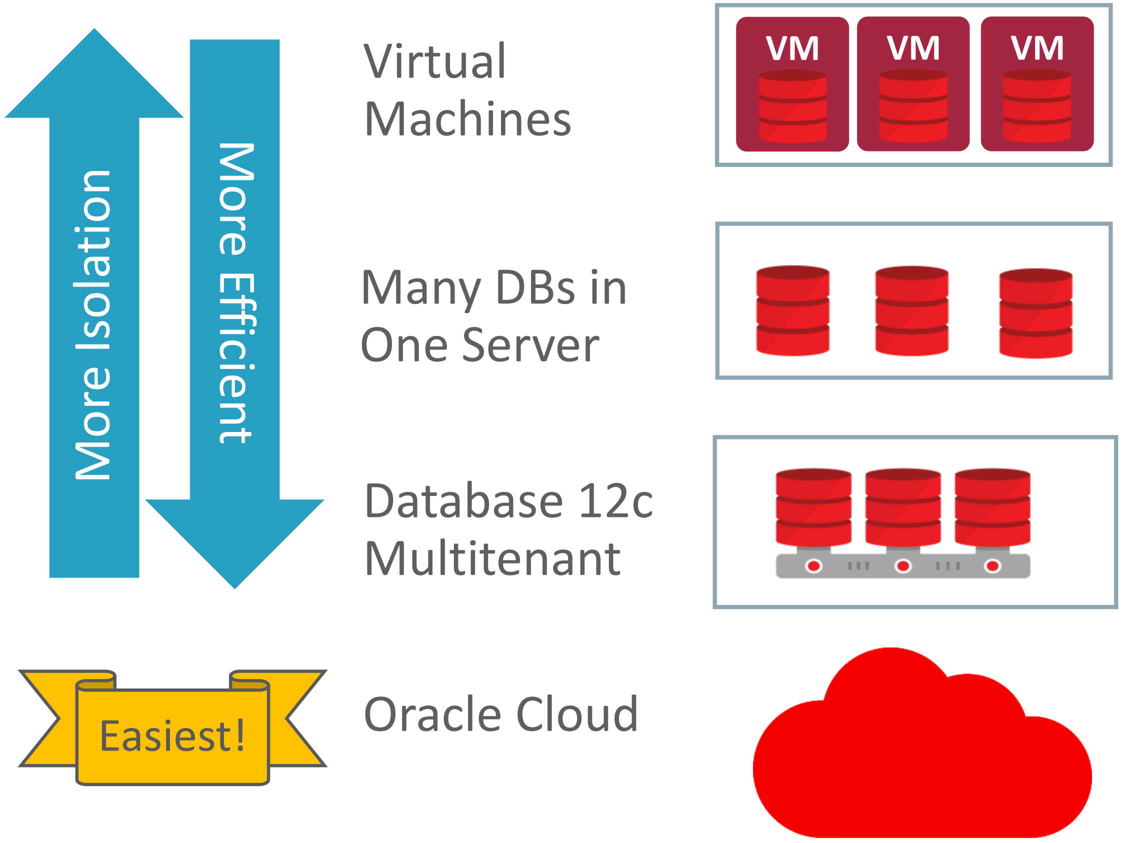
2 Resource Controls

3 Managing a Cloud

4 Use Case: Discover

Why Consolidate?

- More bang for the buck!
  - Most companies have lots of low-load, test, or development databases
  - Fully utilize powerful servers and storage
- Which style of consolidation?
  - Many options
  - No “right” approach!



Virtual Machines

Many DBs in One Server

Database 12c Multitenant

Oracle Cloud

Managing Resources

- Anticipate contention for resources!
  - CPU, memory, I/O, flash space, storage space…
- DBA is the arbiter of resources who determines:

What sort of performance guarantees are required?

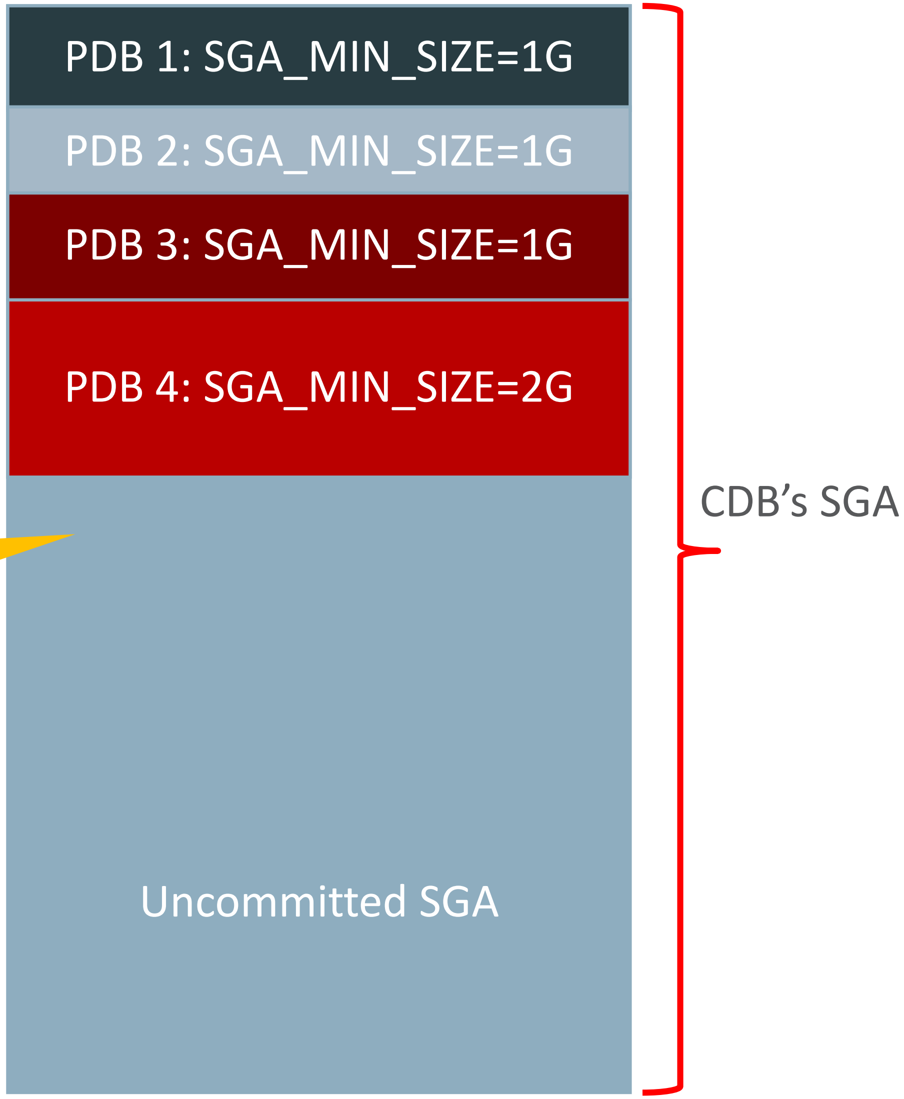
How densely to consolidate?

Limit access to "extra" resources?

How much SGA is each PDB guaranteed?

How to Guarantee Resources?

- Resource guarantees == minimum performance
- Best way to protect against "noisy neighbor"
- Method 1: Parameters!



PDB 4 is guaranteed 2G of SGA. But it can use more!

Best Practices!

Don't let the minimums consume all the space!

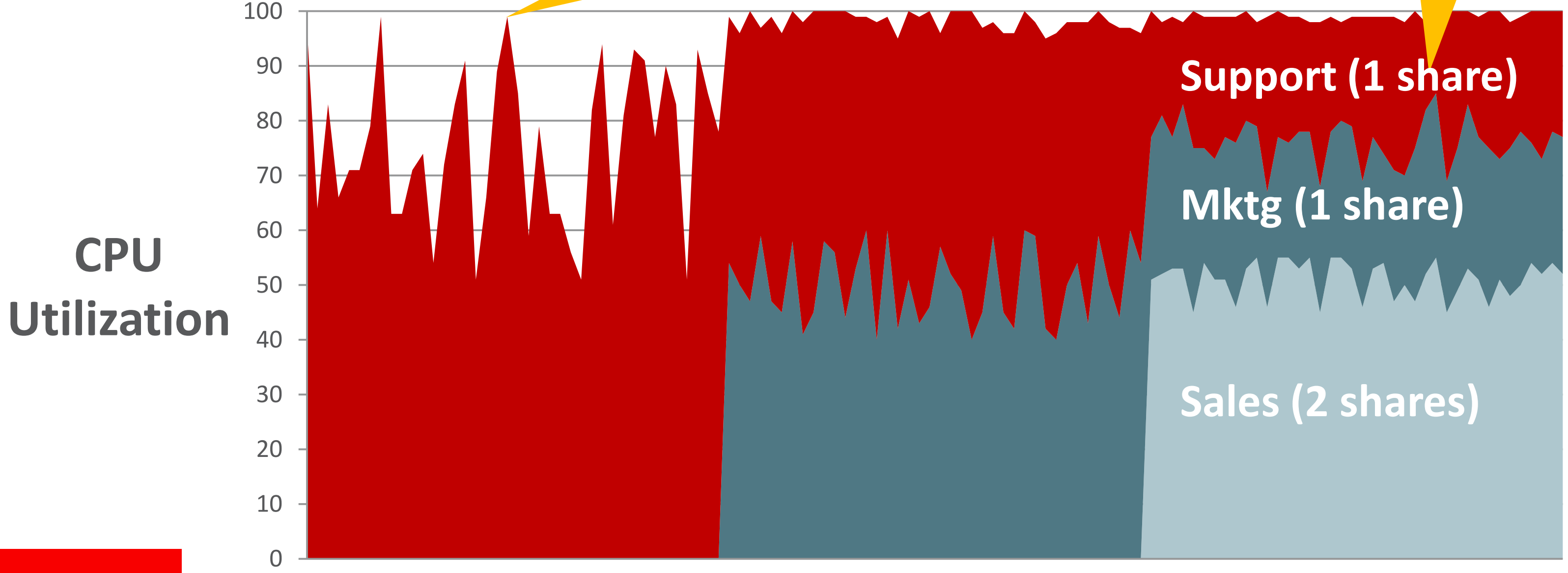
Leave room for databases to get more than the minimum!

How to Guarantee Resources?

- Method 2: Shares!
  - Guarantees a percentage of the total
  - Indicates a priority

Support PDB has 1 out of 4 shares. It is guaranteed 25% of the CPU.

But when it is the only active PDB, it can use all the CPU!



How to Limit Resources?

- Some resources can be constrained with a limit
  - CPU_COUNT limits CPU
  - SGA_TARGET limits SGA
  - PGA_AGGREGATE_LIMIT limits PGA

Use with caution! Use limits if:

(1) No other way to control the resource :-(

(2) Pay for performance! You don't want to give away resources for free!

Program Agenda

1. Consolidation Concepts
2. Resource Controls
3. Managing a Cloud
4. Use Case: Discover

Managing CPU

For PDBs or Consumer Groups (Workloads)

Shares specify the PDB's or Consumer Group's priority

"Sales" is guaranteed 50% of the CPU.
If there is unused CPU, Sales can use more!

Resource Plan				
PDB or Consumer Group	Shares	Utilization Limit	Guaranteed CPU	Maximum CPU
Sales	2		2/(2+1+1) = 50%	100%
Marketing	1		25%	100%
Support	1	75%	25%	75%

A PDB's CPU can also be limited with CPU_COUNT **New in 12.2**

For examples and tuning, see MOS notes 1358709.1, 1338988.1

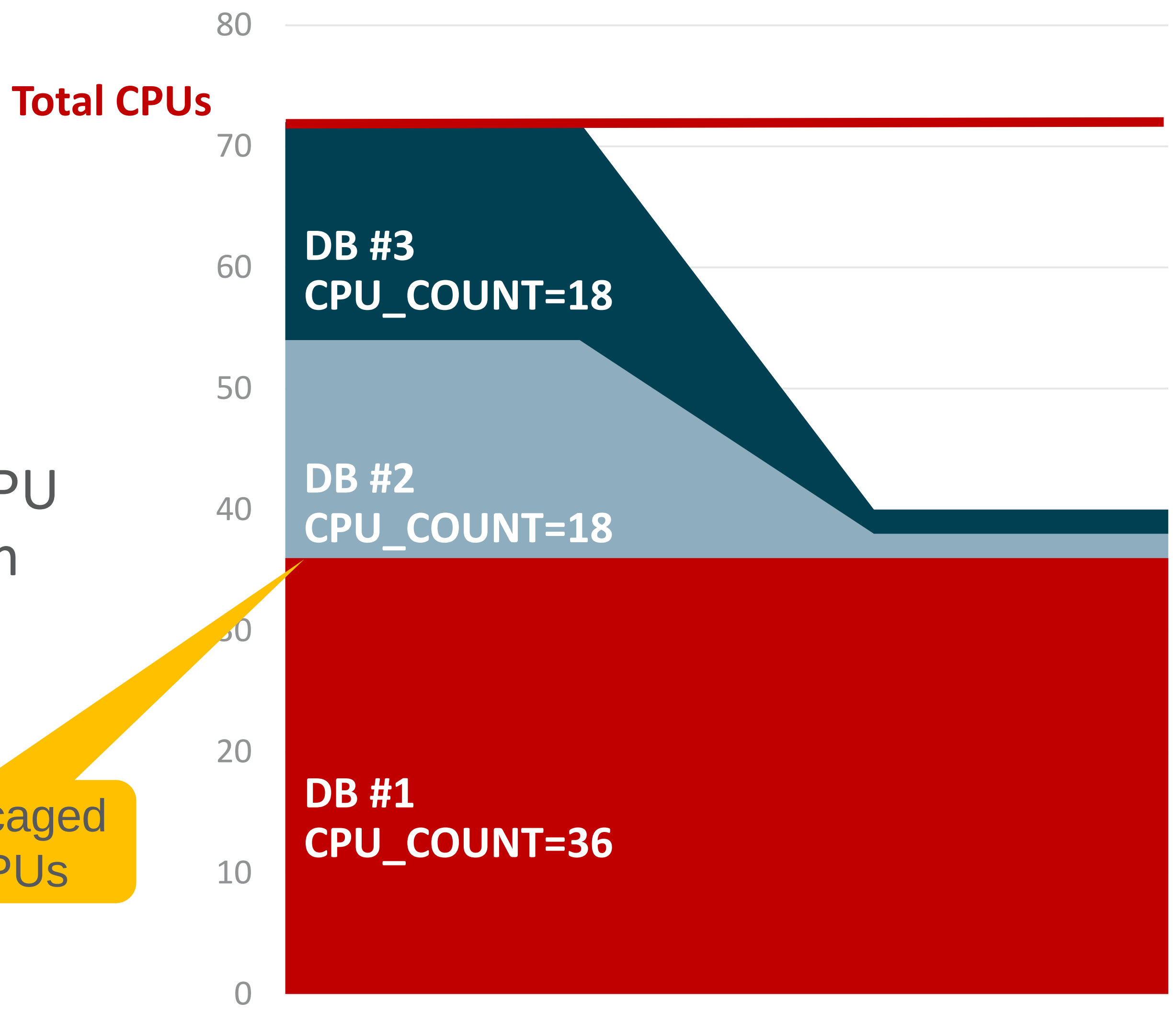
Managing CPU

For Database Instances

Instance Caging manages CPU between database instances

Just 2 steps:

1) Set CPU_COUNT to the maximum CPU threads (not cores!) the instance can use
2) Set RESOURCE_MANAGER_PLAN (e.g. to default_plan)



DB #1 is caged to 36 CPUs

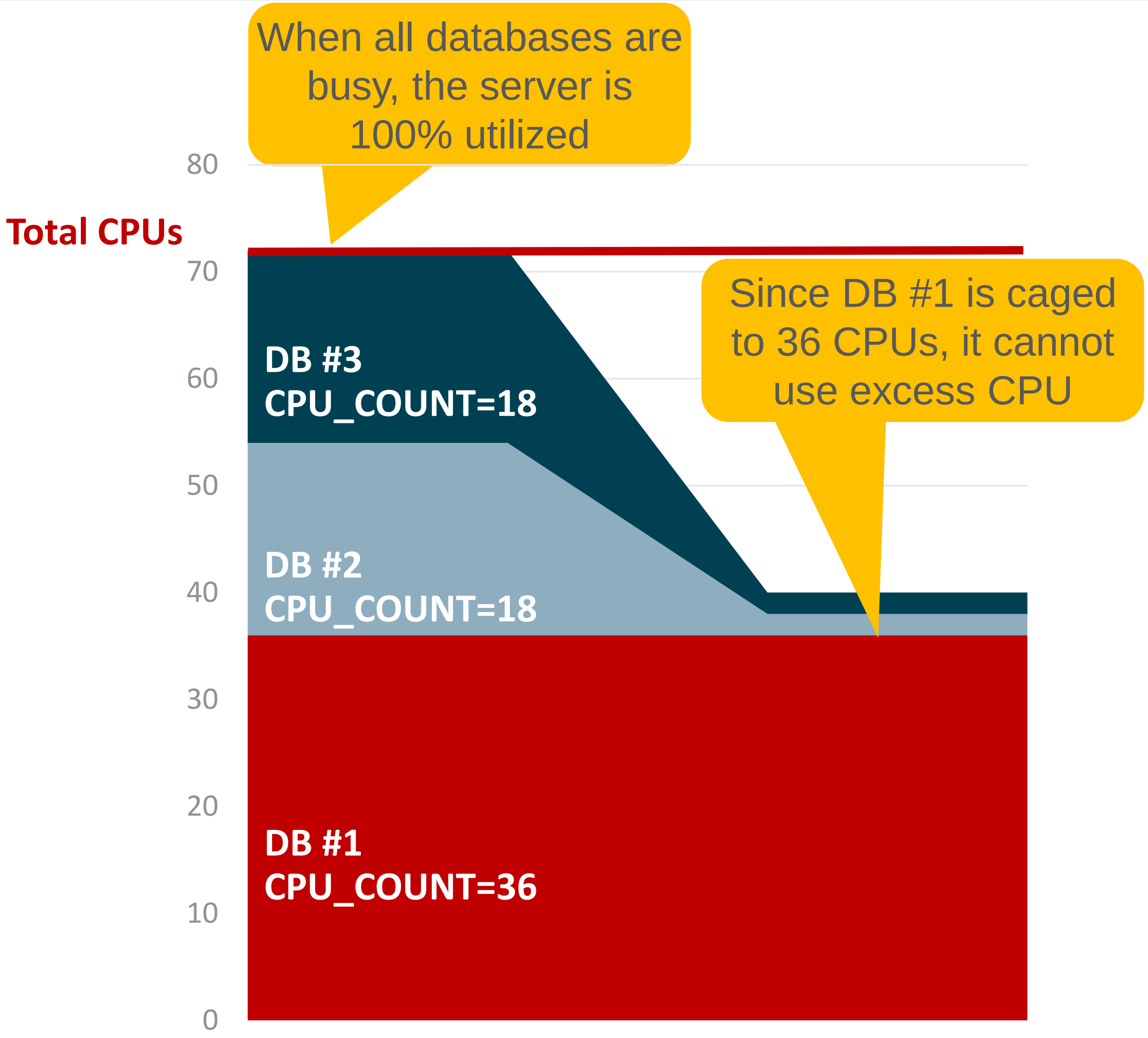
For examples and tuning, see MOS note 1362445.1

Managing CPU

Instance Caging

Approach #1: Partition CPUs

- sum(cpu_counts) <= #cpu threads
- Partitioning provides maximum isolation
- No CPU contention between instances!
- Best for performance-critical databases



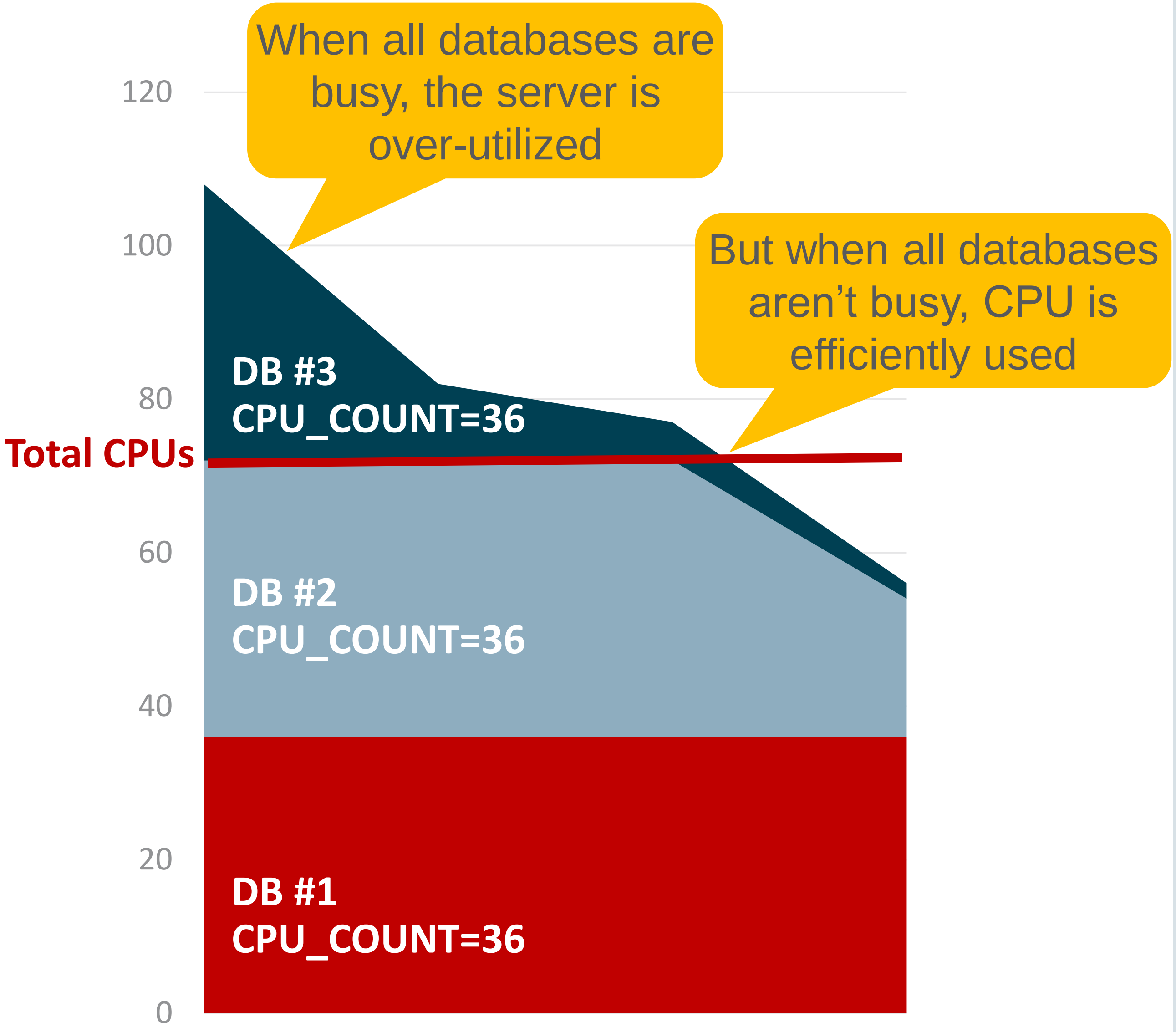
For examples and tuning, see MOS note 1362445.1

Managing CPU

Instance Caging

Option #2: Over-Subscribe CPUs

- sum(cpu_counts) <= 3 x #cpu threads
- Over-subscribing provides efficient CPU utilization
- Some contention for CPU if databases are sufficiently loaded
- Best for non-critical databases



For examples and tuning, see MOS note 1362445.1

Exadata Storage

- Exadata provides extreme I/O performance via flash and hard disks
- Use I/O Resource Manager to tune consolidations for
  - Databases (including VM-based databases)
  - PDBs
  - Consumer Groups (workloads)

Exadata Storage

Managing Flash Space

Key to great OLTP performance? High flash cache hit rate!

Inter-Database IORM Plan			
Database	Flash Cache Min	Flash Cache Limit	Flash Cache Size
DB-1	500 MB		
DB-2		2 GB	
CDB-1			10 GB

ORACLE

Exadata Storage

Managing Flash Space

Minimum guarantees space for a database.
Useful for critical databases that are sometimes inactive.

Soft **Limits** are only applied when the Flash Cache is full.

Size limits and reserves space for a database.
Use cautiously!

Inter-Database IORM Plan

Database	Flash Cache Min	Flash Cache Limit	Flash Cache Size
DB-1	500 MB		
DB-2		2 GB	
CDB-1			10 GB

Exadata Storage

Managing Flash Space

To control flash space for databases, configure an IORM plan on the storage cells.

To control flash space for PDBs, configure a database resource plan

Inter-Database IORM Plan		
Database	Flash Cache Min	Flash Cache Limit
DB-1	500 MB	
DB-2		2 GB
CDB-1	1 GB	10 GB

CDB Resource Plan		
PDB	Memory Min	Memory Limit
Sales	15%	75%
Marketing	30%	
Support		50%

For CDB plans, apply percentages against the CDB's min and limit

Exadata Storage

Managing Flash and Disk I/Os

CPU, flash I/Os, and disk I/Os are managed with the same plan!

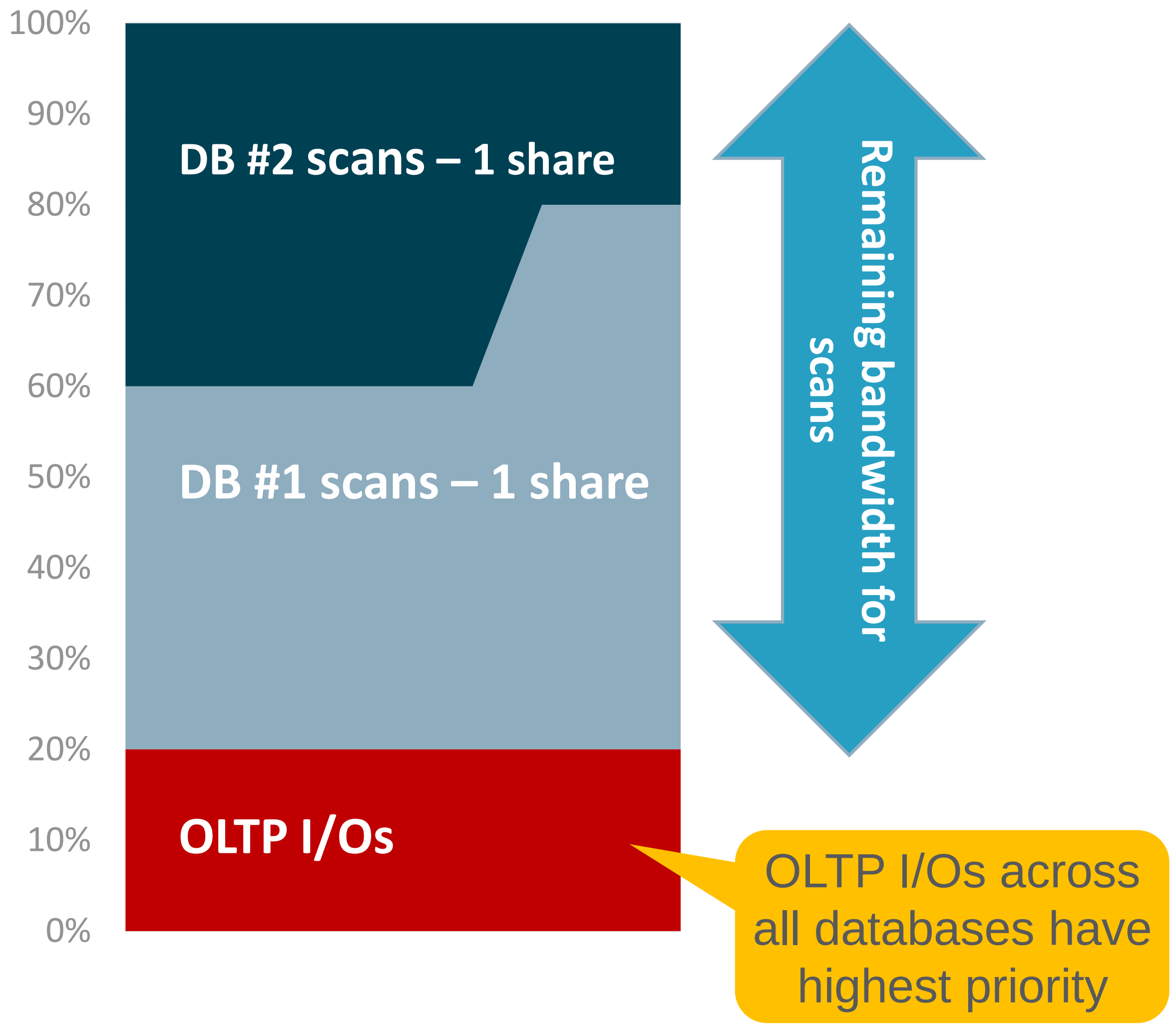
(Only exception is Instance Caging)

	Type of Plan	Where to Configure?	Shares	Limits
Databases	Inter-Database IORM plan for I/Os	Configured on Exadata cell. Default shares based on CPU_COUNT **New in 18.1**	✓	✓
PDBs in a CDB	CDB Resource Plan	Configured on Database. Default shares based on CPU_COUNT **New in 18.1**	✓	✓
Consumer Groups in a Database	Database Resource Plan	Configured on Database	✓	✓

Exadata Storage

Managing Flash I/Os

- Exadata has extremely high flash bandwidth
- Flash I/Os managed via shares and utilization limits New in 18.1
- OLTP I/Os always have top priority, regardless of database
  - Plenty of bandwidth to accommodate all OLTP I/Os!



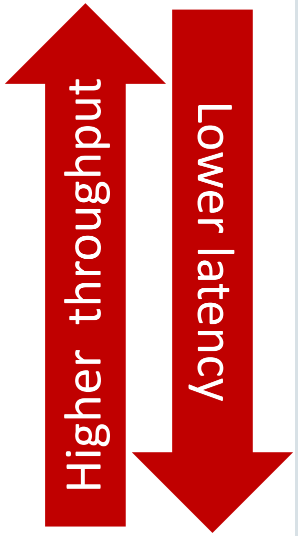
OLTP I/Os across all databases have highest priority

Exadata Storage

Managing Disk I/Os

- Two goals for managing disk I/Os
  - Low latency: important for OLTP
  - High throughput: important for scans
- High throughput results in high latency! You cannot have both goals!
- IORM "objective"
  - Choose what is more important: latency or throughput

Objective	Peak Disk Utilization for Scans	OLTP Read Latencies
Basic (IORM is off)	100%	< 2 sec
High Throughput	100%	< 1 sec
Balanced	90%	< 160 ms
Low Latency	40%	< 30 ms
Auto	varies	varies



Higher throughput

Lower latency

Use "auto" objective:
IORM chooses for you, based on your current workload.

New in 12.2!

Non-Exadata I/Os

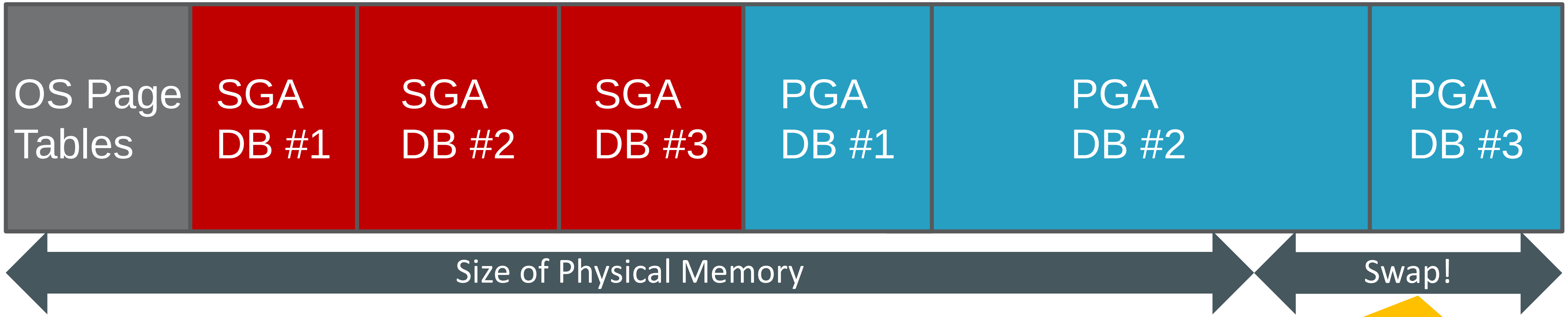
PDB Rate Limits

- Rate limit parameters prevent PDBs from swamping the storage system
  - MAX_IOPS: maximum I/O requests per second
  - MAX_MBPS: maximum megabytes per second of I/O
- For non-Exadata storage only!
  - On Exadata, use IORM - much more powerful!
- What about background I/Os?
  - DBWR I/Os are counted, but not throttled
  - LGWR and Root I/Os aren’t counted, nor throttled

Monitor the “resmgr:io rate limit” wait event and v$rsrcmgrmetric_history

Managing Physical Memory

Preventing Swapping

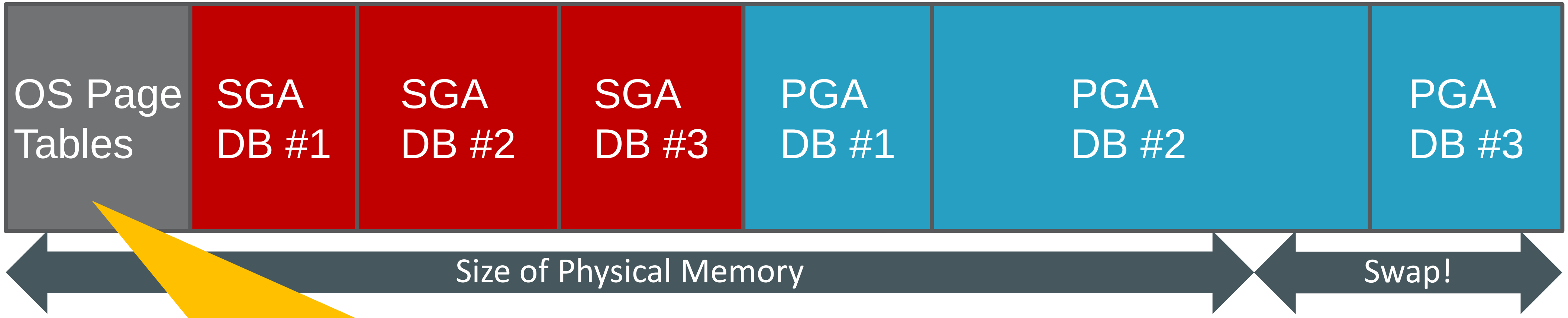


All memory allocations should fit within the physical memory!
If not, the server will page,
leading to performance problems and RAC evictions.

If you see *“WARNING: Heavy swapping observed on system”* in the alert log, take action!

Managing Physical Memory

Linux Huge Pages

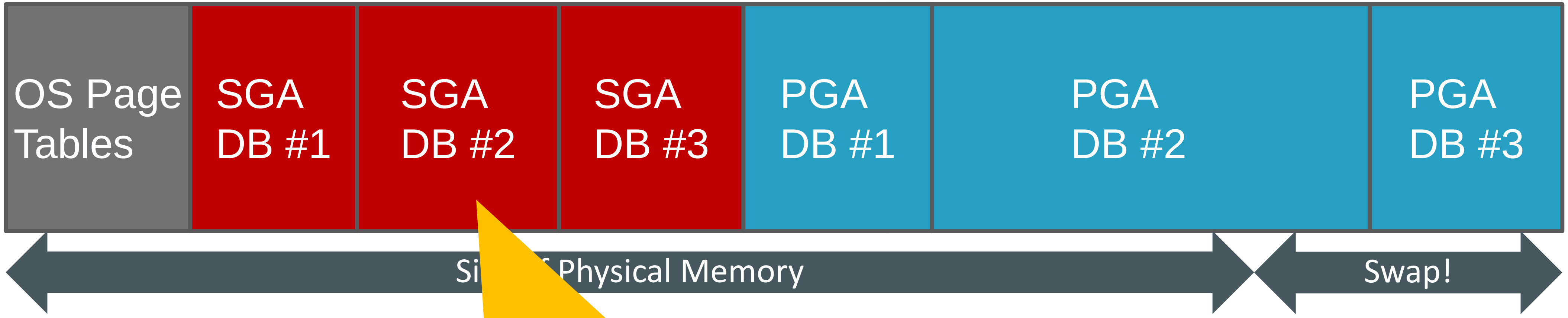


With huge pages,
shrink OS page tables from 8 GB to 16 MB!
And boost performance!

Use Huge Pages!
See MOS notes #361468.1 and #401749.1.

Managing Physical Memory

SGA

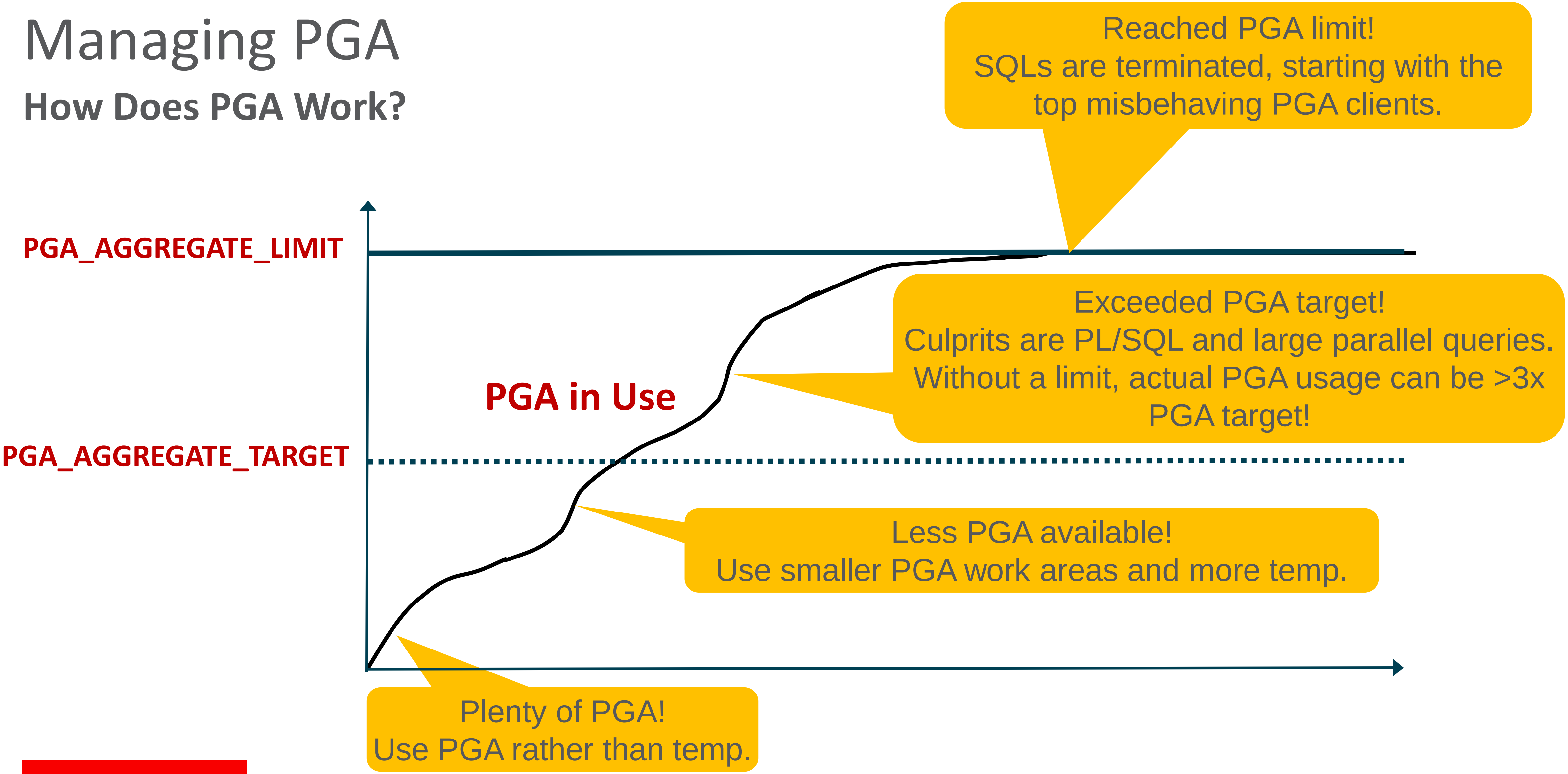


Configure SGA with SGA_TARGET and USE_LARGE_PAGES

Use MEMORY_TARGET for small SGAs.
Huge pages and Memory RM are not supported with MEMORY_TARGET!

Managing PGA

How Does PGA Work?



Reached PGA limit!
SQLs are terminated, starting with the top misbehaving PGA clients.

Exceeded PGA target!
Culprits are PL/SQL and large parallel queries. Without a limit, actual PGA usage can be >3x PGA target!

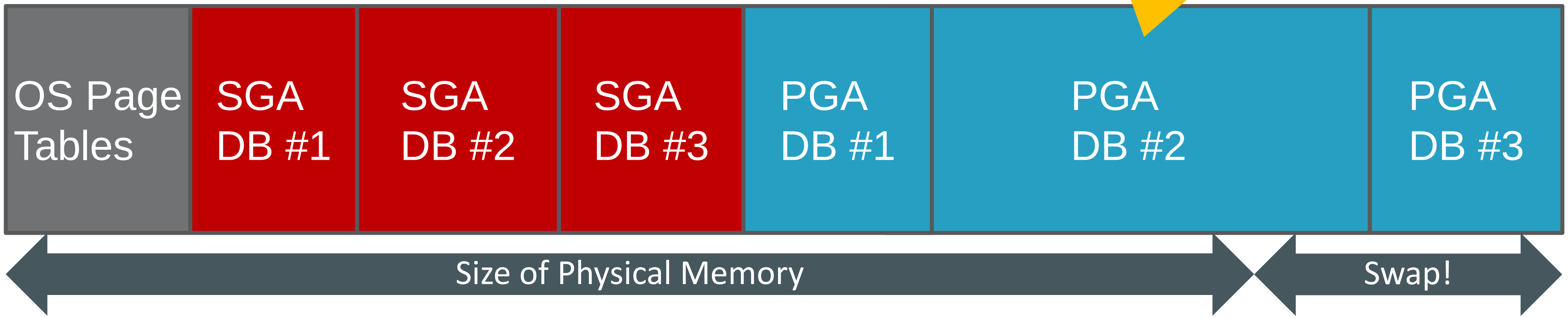
Less PGA available!
Use smaller PGA work areas and more temp.

Plenty of PGA!
Use PGA rather than temp.

Managing Physical Memory

PGA

Configure PGA with PGA_AGGREGATE_LIMIT.



- Monitor v$pgastat
  - “maximum PGA allocated” – historical max
  - “total PGA allocated” – current usage

Verify that PGA_AGGREGATE_LIMIT >= PGA_AGGREGATE_TARGET x 2!

New in 12.2!

Managing Memory in a CDB

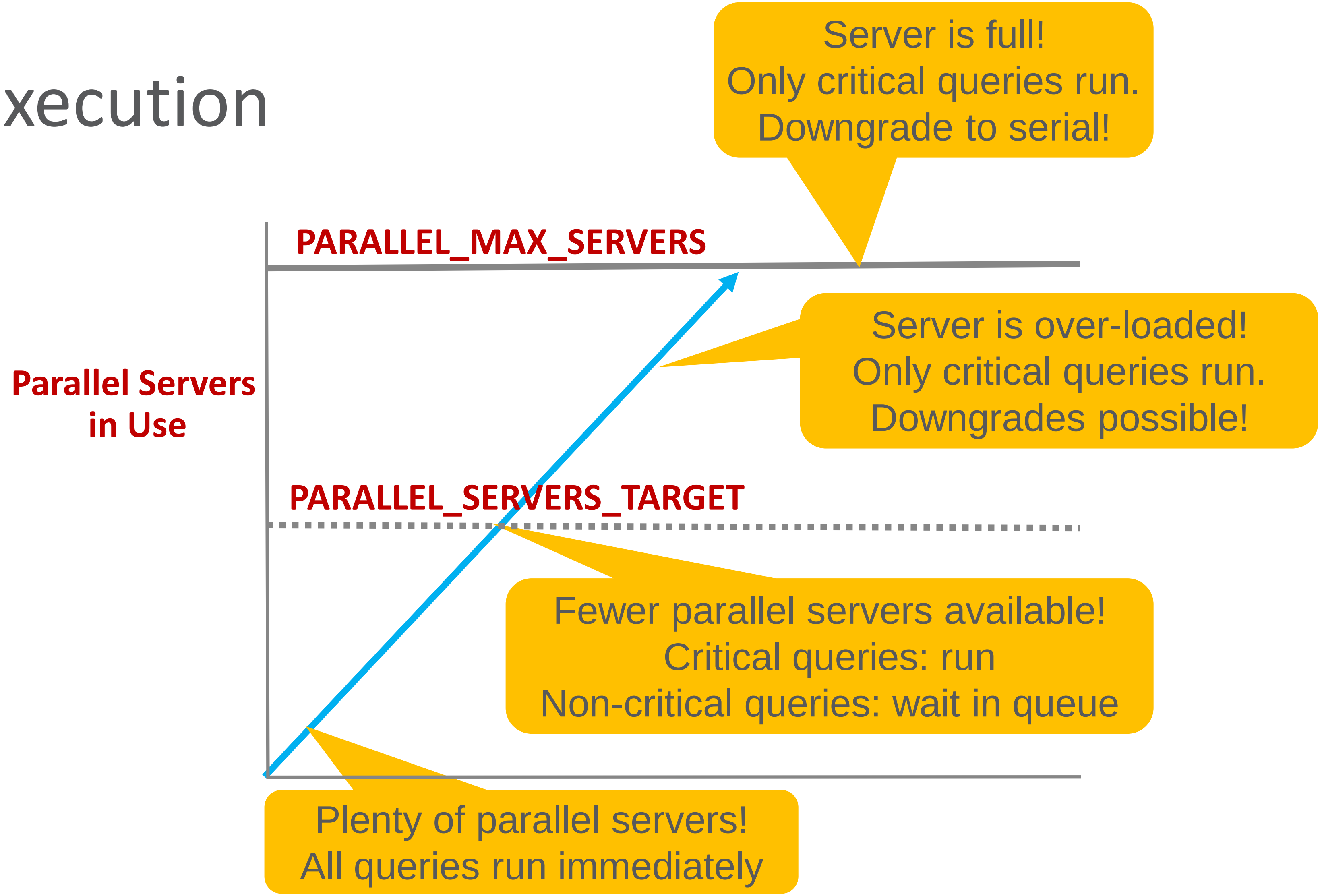
PDB Parameter	Description	When to Set?
PGA_AGGREGATE_LIMIT	**Maximum PGA size**	**Set for unruly PDBs.** **Verify that the limit >= 2x target.** **Should set on Cloud!**
PGA_AGGREGATE_TARGET	**Target PGA size**	**Set for unruly PDBs.** **Should set on Cloud!**
SGA_MIN_SIZE	Minimum SGA size	Useful for small PDBs or critical PDBs
DB_CACHE_SIZE	Minimum buffer cache size	Set sparingly – if PDB's shared pool is shrinking its buffer cache
SHARED_POOL_SIZE	Minimum shared pool size	Set sparingly - if PDB's buffer cache is shrinking its shared pool
SGA_TARGET	**Maximum SGA size**	**Set for unruly PDBs.** **Should set on Cloud!**

ORACLE

Managing Parallel Execution

Parallel Statement Queuing

- Use queuing to avoid parallel execution downgrades
- Queue non-critical queries when server is busy
- Two types of queries
  - Critical: not queued
  - Non-critical: “queue”able



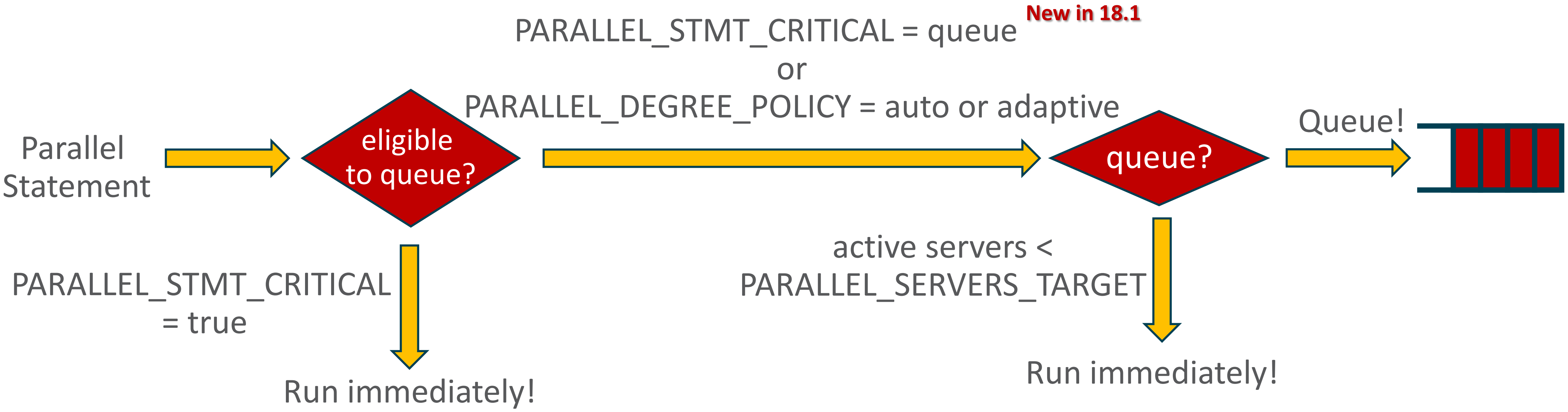
Server is full!
Only critical queries run.
Downgrade to serial!

Server is over-loaded!
Only critical queries run.
Downgrades possible!

Fewer parallel servers available!
Critical queries: run
Non-critical queries: wait in queue

Plenty of parallel servers!
All queries run immediately

Managing Parallel Execution



New in 18.1

Parallel Statement

eligible to queue?

PARALLEL_STMT_CRITICAL = queue
or
PARALLEL_DEGREE_POLICY = auto or adaptive

queue?

Queue!

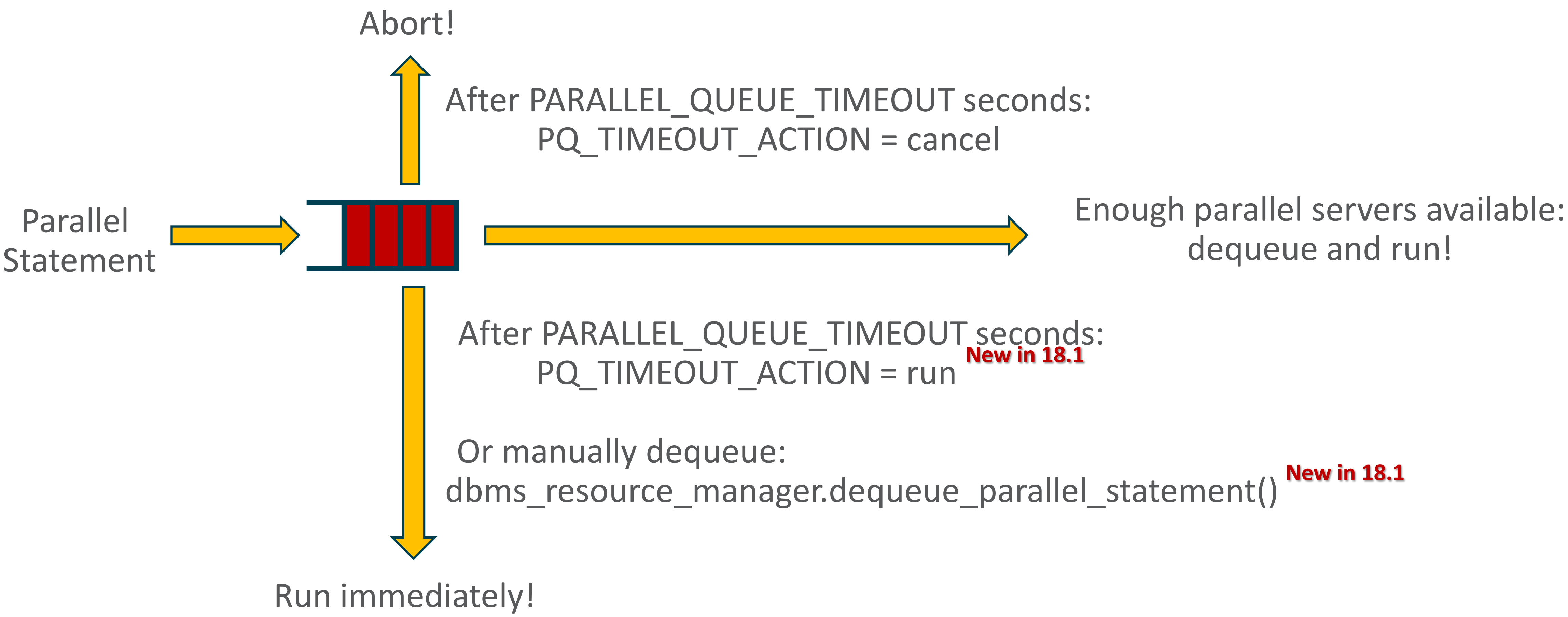
PARALLEL_STMT_CRITICAL = true

Run immediately!

active servers < PARALLEL_SERVERS_TARGET

Run immediately!

Managing Parallel Execution



Abort!

After PARALLEL_QUEUE_TIMEOUT seconds:
PQ_TIMEOUT_ACTION = cancel

Parallel Statement

Enough parallel servers available:
dequeue and run!

After PARALLEL_QUEUE_TIMEOUT seconds:
PQ_TIMEOUT_ACTION = run

New in 18.1

Or manually dequeue:
dbms_resource_manager.dequeue_parallel_statement()

New in 18.1

Run immediately!

Parallel Statement Queuing

Multi-Tenant

Configure parallel statement queuing at CDB level, PDB level, or both!

	Set in PDB?	Default Value
Max DOP		PDB's CPU_COUNT
PARALLEL_SERVERS_TARGET	Yes! **New in 18.1**	Calculated from PDB's CPU_COUNT
PARALLEL_MAX_SERVERS	Yes! **New in 12.2**	Calculated from PDB's CPU_COUNT

ORACLE

Program Agenda

1 Consolidation Concepts

2 Resource Controls

3 Managing a Cloud

4 Use Case: Discover

ORACLE

Managing a Cloud

- Use Resource Manager in all types of deployments
  - Private Cloud, Cloud at Customer, Public Cloud
- In Public Cloud, Resource Manager natively configured to (a) define levels of service and (b) provide isolation between tenants
  - Exadata Express Cloud Service
  - Autonomous Database Warehouse Cloud **Just Announced!**
- Driving principles
  - Pick a simple way of defining a few PDB profiles
  - Know your workload and pick a partitioned or over-subscription strategy
  - Configure and lockdown resource plans and parameters

New in 12.2!

Simplified CDB Management

Use case: Exadata Express Cloud Service

1) Establish a small number of PDB Profiles
2) Create a CDB Resource Plan, based on these PDB Profiles

CDB Resource Plan	
Performance Profile	Shares
X20	1
X250	1
X500	2
X1000	4

3) Lockdown each profile's parameters.
The PDB DBA cannot change these values!

Lockdown Profile for X1000	
Parameter	Value
PDB_LOCKDOWN	X1000
DB_PERFORMANCE_PROFILE	X1000
CPU_COUNT	8
SGA_TARGET	15G
PGA_AGGREGATE_LIMIT	15G
SESSIONS	
MAX_IDLE_TIME	

ORACLE

Automatic Management

Reduce CPU and Exadata I/O Impact

Simple ways to use Resource Management

Task	Goal	Action
RMAN backup or image copy	Deprioritize CPU and Exadata I/Os	Automatically mapped to the BATCH consumer group. Just enable DEFAULT_PLAN.
Data Pump data load	Deprioritize CPU and Exadata I/Os (Can also prioritize!)	Automatically mapped to the ETL consumer group. Just enable ETL_PLAN or DSS_PLAN.
Automated maintenance tasks: • Optimizer Stats Collection • Segment Advisor • SQL Tuning Advisor • SQL Plan Management	Deprioritize CPU and Exadata I/Os	Automatically mapped to the ORA$AUTOTASK consumer group. Automatically managed, using Maintenance Windows.

Automatic Management

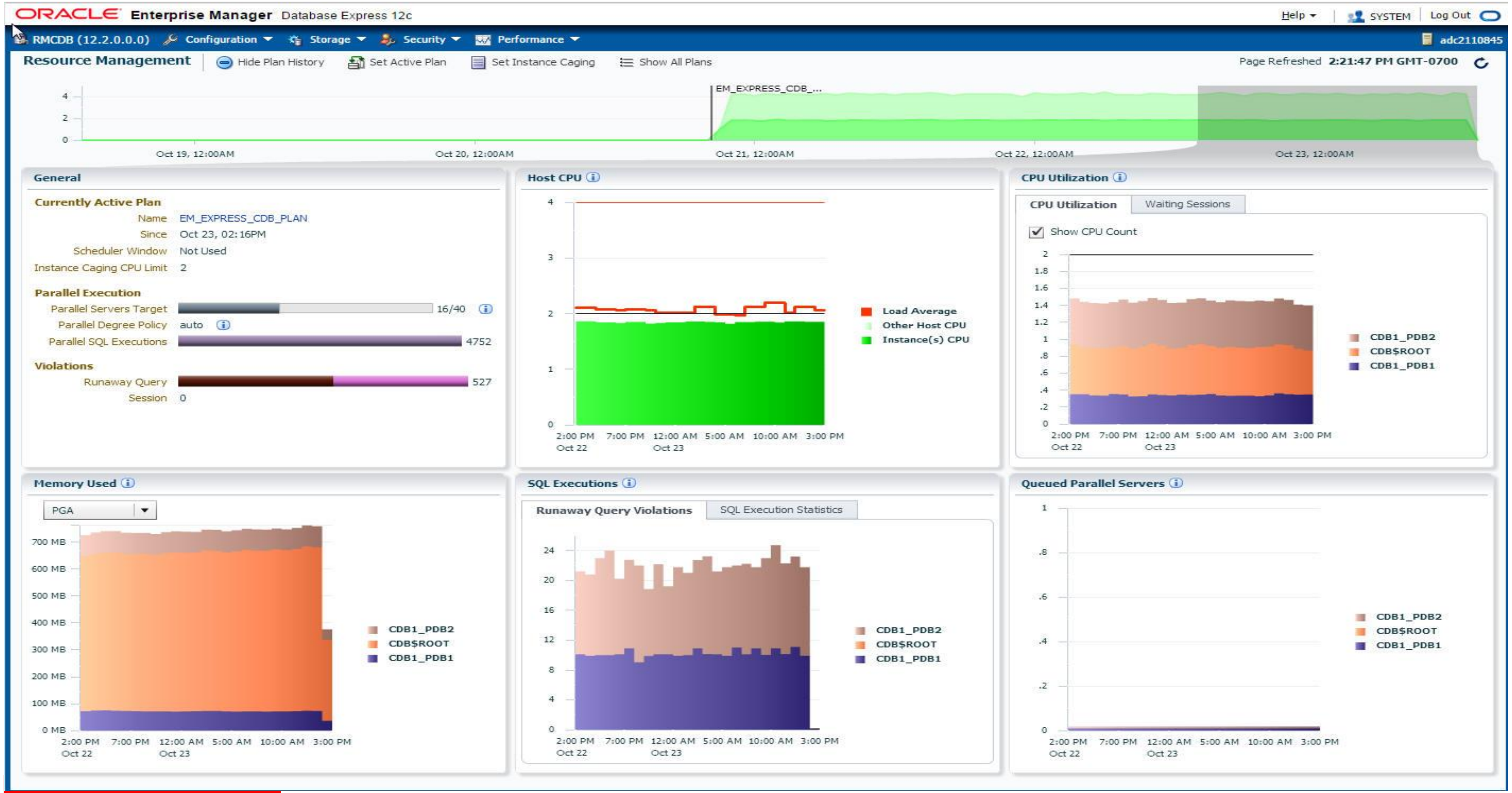
InMemory Operations

Simple ways to use Resource Management

Task	Goal	Action
Population of InMemory data New in 12.2	Reduce CPU impact.	Automatically categorized into the ora$autotask consumer group. Deprioritize by enabling DEFAULT_PLAN.
Dynamic scans of InMemory data New in 12.2	Accelerate InMemory scans, using leftover CPU and light-weight threads	Just enable CPU Resource Manager. Requires >= 24 CPU threads.
InMemory Fast Start writes New in 18.1	Reduce impact of Fast Start writes on Exadata	Just enable IORM by setting Objective = “auto”.
Prioritize smart scans over InMemory population New in 12.2.1.1	Reduce CPU impact on Exadata storage cells	No action required!

Enterprise Manager Express UI

Configure and Monitor PDB Resource Manager



General

Currently Active Plan

Name	EM_EXPRESS_CDB_PLAN
Since	Oct 23, 02:16PM
Scheduler Window	Not Used
Instance Caging CPU Limit	2

Parallel Execution

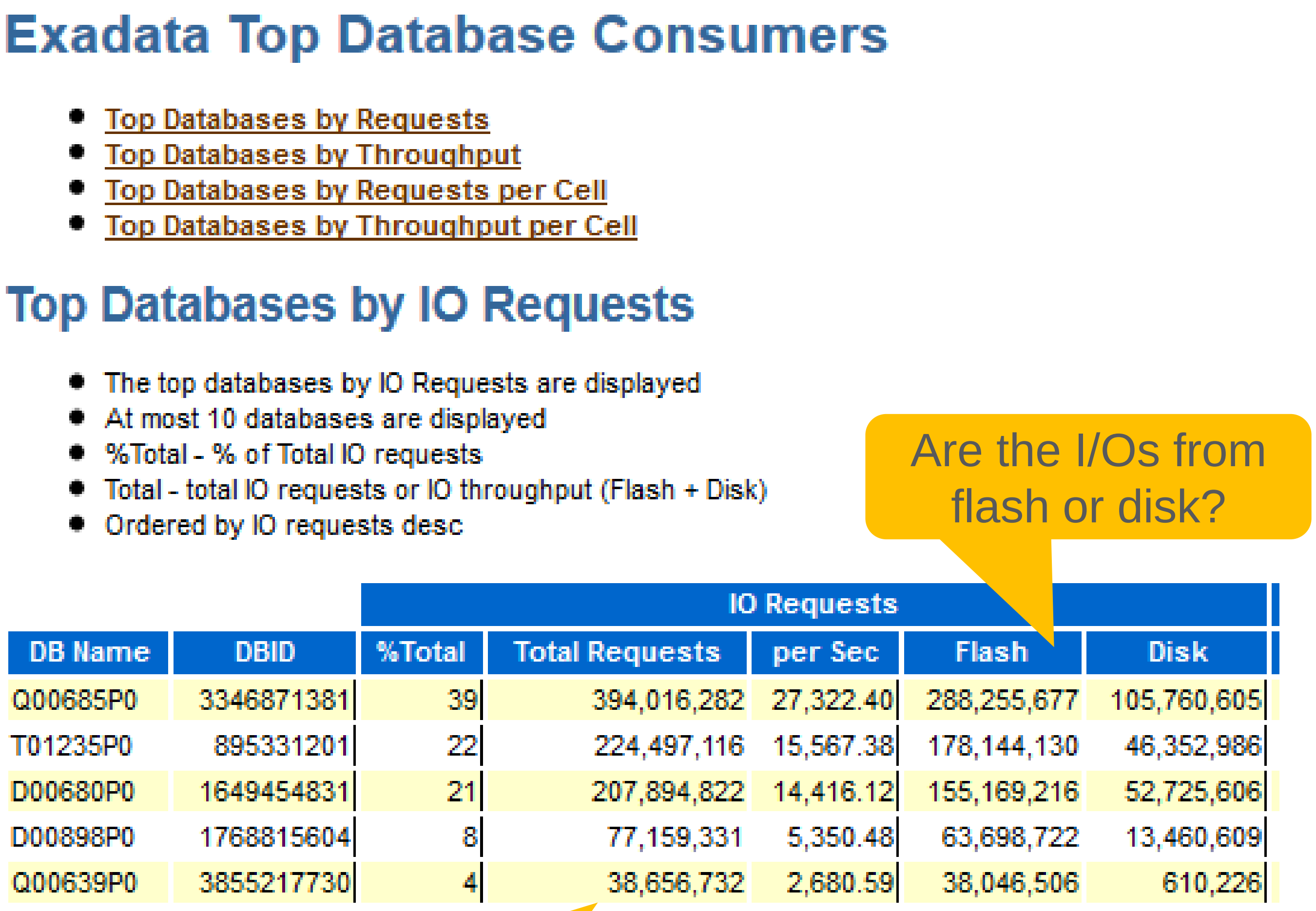
Parallel Servers Target	16/40
Parallel Degree Policy	auto
Parallel SQL Executions	4752

Violations

Runaway Query	527
Session	0

Exadata Storage

- Use the Exadata sections of AWR report to configure and tune IORM
- Or, use Enterprise Manager Cloud Control UI!



Exadata Top Database Consumers

- Top Databases by Requests
- Top Databases by Throughput
- Top Databases by Requests per Cell
- Top Databases by Throughput per Cell

Top Databases by IO Requests

- The top databases by IO Requests are displayed
- At most 10 databases are displayed
- %Total - % of Total IO requests
- Total - total IO requests or IO throughput (Flash + Disk)
- Ordered by IO requests desc

		IO Requests				
DB Name	DBID	%Total	Total Requests	per Sec	Flash	Disk
Q00685P0	3346871381	39	394,016,282	27,322.40	288,255,677	105,760,605
T01235P0	895331201	22	224,497,116	15,567.38	178,144,130	46,352,986
D00680P0	1649454831	21	207,894,822	14,416.12	155,169,216	52,725,606
D00898P0	1768815604	8	77,159,331	5,350.48	63,698,722	13,460,609
Q00639P0	3855217730	4	38,656,732	2,680.59	38,046,506	610,226

Are the I/Os from flash or disk?

How much I/O is each database generating?

Program Agenda

1 Consolidation Concepts

2 Resource Controls

3 Managing a Cloud

4 Use Case: Discover

ORACLE

Integrated Cloud

Applications & Platform Services

For questions, go to Oracle DemoGrounds, Booth SOA 149

ORACLE®

And a few extra slides for content we didn’t have time to cover…

Mixed Workloads and Resource Manager

Mixed Workloads and Resource Management

- Every database runs multiple workloads with different priorities
  - OLTP database
    - OLTP
    - Real-time reports
    - Maintenance (backup, stats gathering, etc.)
  - Data warehouse
    - Critical, tuned reports
    - Batch jobs
    - ETL
    - Ad-hoc reports
- These workloads compete for resources
- Use Resource Manager to allocate resources to workloads

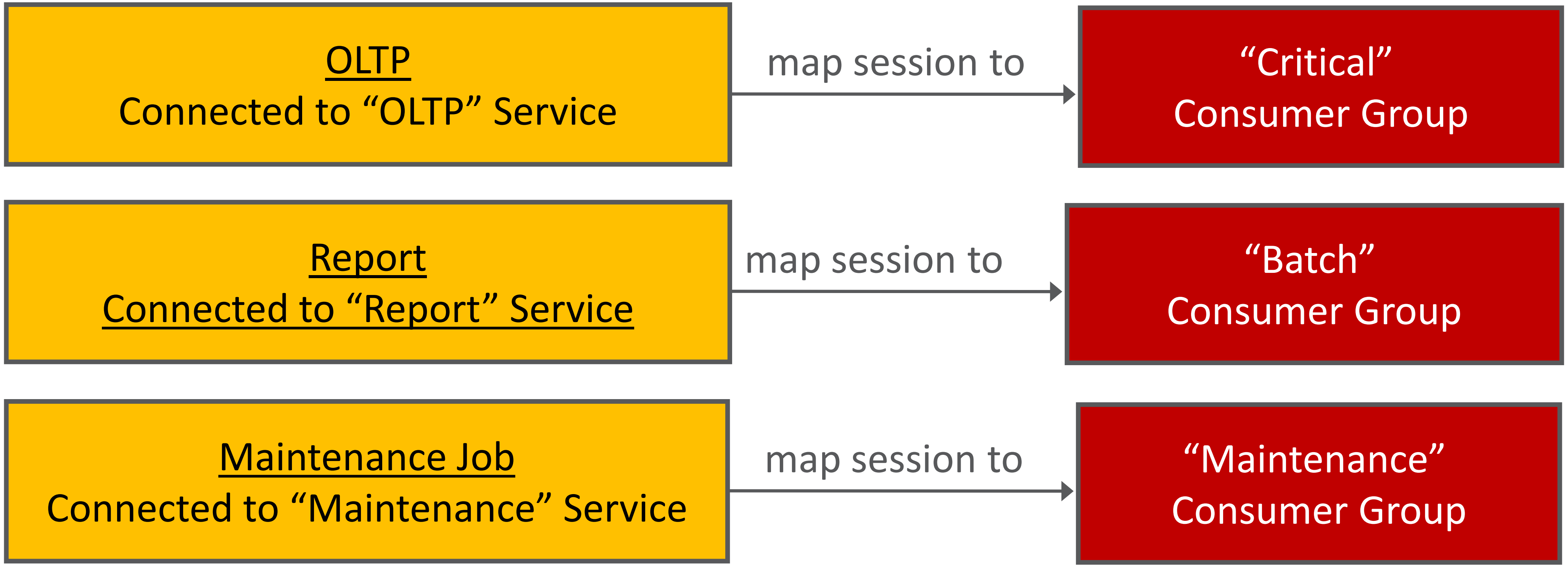
ORACLE

Configuring Workloads with Consumer Groups

- First step
  - Group database sessions that comprise a workload into a Consumer Group
- Sample consumer groups
  - Critical
  - Batch
  - Maintenance
  - Other (default)

Configuring Workloads with Consumer Groups

If you use services, create a Consumer Group for each service



OLTP
Connected to “OLTP” Service

map session to

“Critical”
Consumer Group

Report
Connected to “Report” Service

map session to

“Batch”
Consumer Group

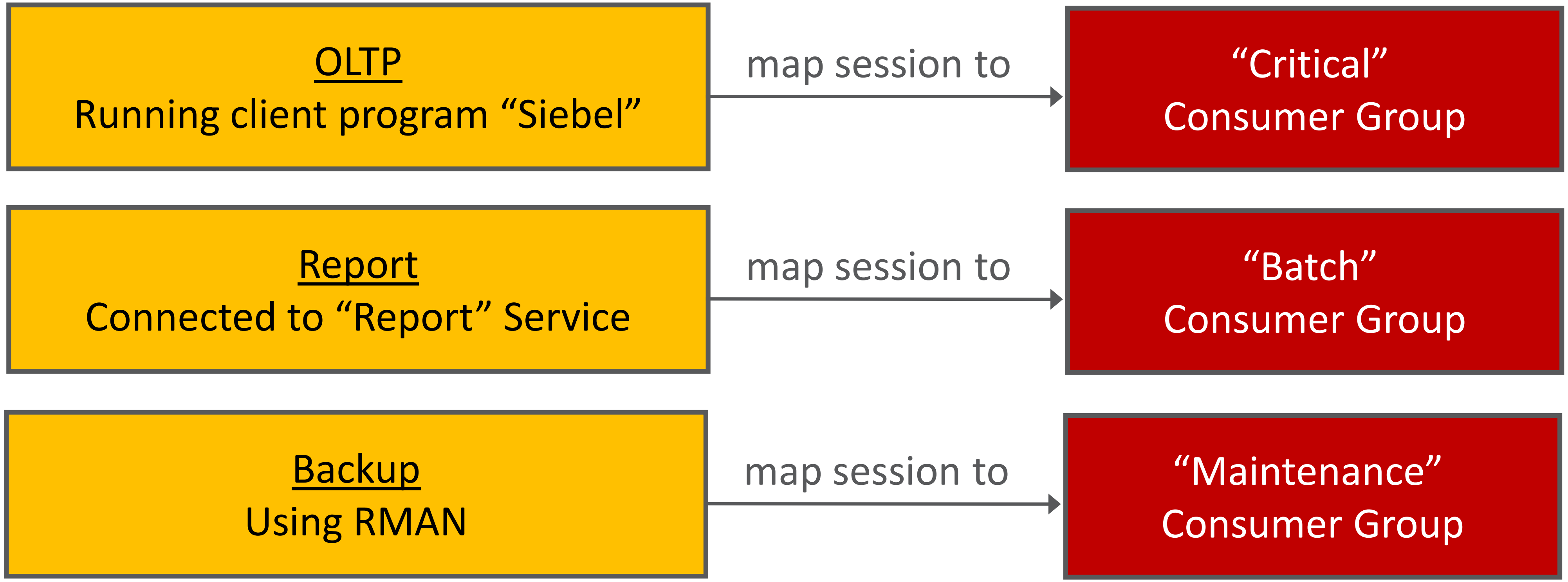
Maintenance Job
Connected to “Maintenance” Service

map session to

“Maintenance”
Consumer Group

Configuring Workloads with Consumer Groups

Or use other session attributes…



OLTP
Running client program “Siebel”
map session to
“Critical”
Consumer Group

Report
Connected to “Report” Service
map session to
“Batch”
Consumer Group

Backup
Using RMAN
map session to
“Maintenance”
Consumer Group

Configuring Workloads with Consumer Groups

Use any of these session attributes to map sessions to Consumer Groups:

Session Attributes:

- Oracle user name
- Client O/S user name
- Client program name
- Client machine name
- Client id
- Service name
- Module name
- Action name

- **Function being performed**

  "backup" (RMAN backup, defaults to BATCH_GROUP)

  "copy" (RMAN image copy, defaults to BATCH_GROUP)

  "dataload" (datapump, defaults to ETL_GROUP)

  "inmemory" (in-memory population, defaults to ORA$AUTOTASK)

  New in 12.1.0.2

```
dbms_resource_manager.set_consumer_group_mapping(
attribute => 'ORACLE_FUNCTION', value => 'INMEMORY', consumer_group => 'BATCH_GROUP');
```

Managing CPU within a Database

Configuring CPU Resource Manager

Sample SQL Script

```
begin
  dbms_resource_manager.create_pending_area;
  dbms_resource_manager.create_plan(plan => ‘myplan’);
  dbms_resource_manager.create_plan_directive(plan =>‘myplan’,
      group_or_subplan => ‘critical_group’, shares => 5);
dbms_resource_manager.create_plan_directive(plan =>‘myplan’,
      group_or_subplan => ‘other_groups’, shares => 1);
  dbms_resource_manager.submit_pending_area;
end;
```

Mandatory 1st step for creating or modifying a Resource Plan

Create a new Resource Plan

Set resource allocations and other directives for each Consumer Group

Mandatory final step. Verifies the Resource Plan. If this passes, you can enable the Resource Plan.

Configuring CPU Resource Manager

Sample SQL Script

```
begin
  dbms_resource_manager.create_pending_area;
  dbms_resource_manager.create_cdb_plan(plan => ‘myplan’);
  dbms_resource_manager.create_cdb_plan_directive(plan =>‘myplan’,
      pluggable_database => ‘sales’, shares => 2);
  dbms_resource_manager.update_cdb_default_directive(plan =>‘myplan’,
      new_shares => 1);
  dbms_resource_manager.submit_pending_area;
end;
```

Mandatory 1st step for creating or modifying a CDB Resource Plan

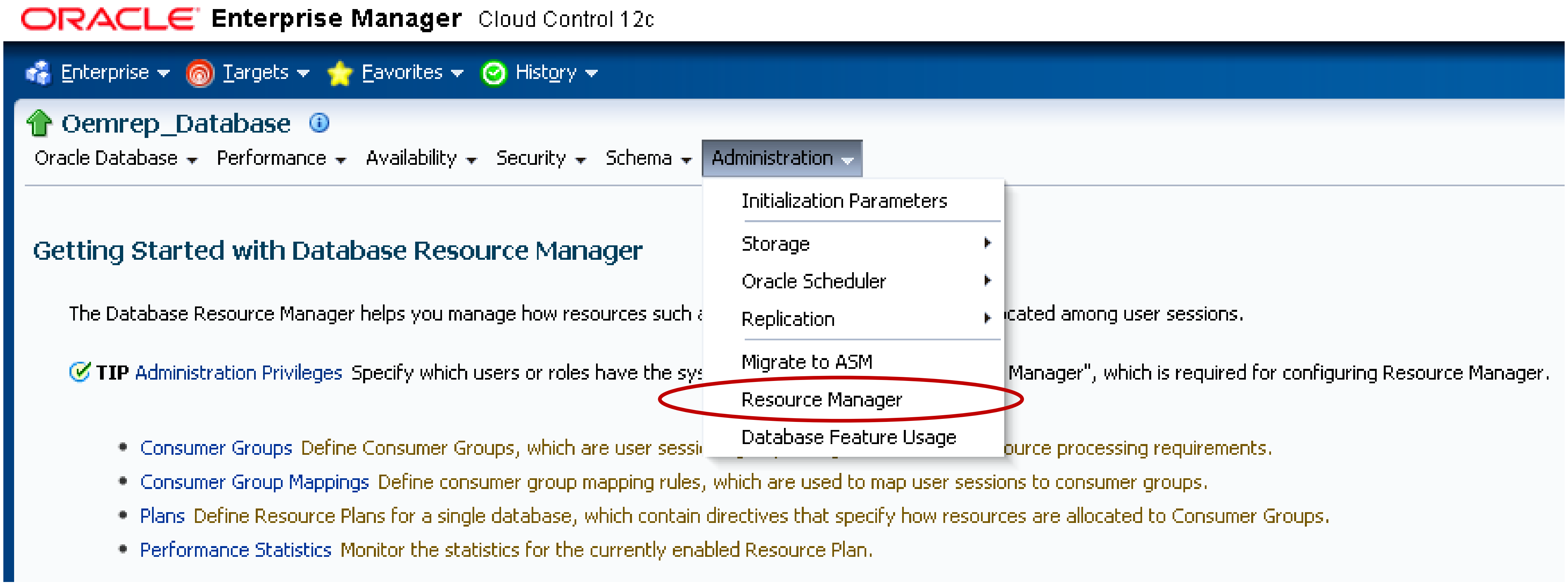
Create a new CDB Resource Plan

Set resource allocations and other directives for each PDB

Mandatory final step. Verifies the CDB Resource Plan. If this passes, you can enable the CDB Resource Plan.

Configuring CPU Resource Manager

Enterprise Manager UI



Getting Started with Database Resource Manager

The Database Resource Manager helps you manage how resources such a ... cated among user sessions.

TIP Administration Privileges Specify which users or roles have the sys ... Manager", which is required for configuring Resource Manager.

- Consumer Groups Define Consumer Groups, which are user sessi ... ource processing requirements.
- Consumer Group Mappings Define consumer group mapping rules, which are used to map user sessions to consumer groups.
- Plans Define Resource Plans for a single database, which contain directives that specify how resources are allocated to Consumer Groups.
- Performance Statistics Monitor the statistics for the currently enabled Resource Plan.

Monitoring and Tuning CPU Resource Manager

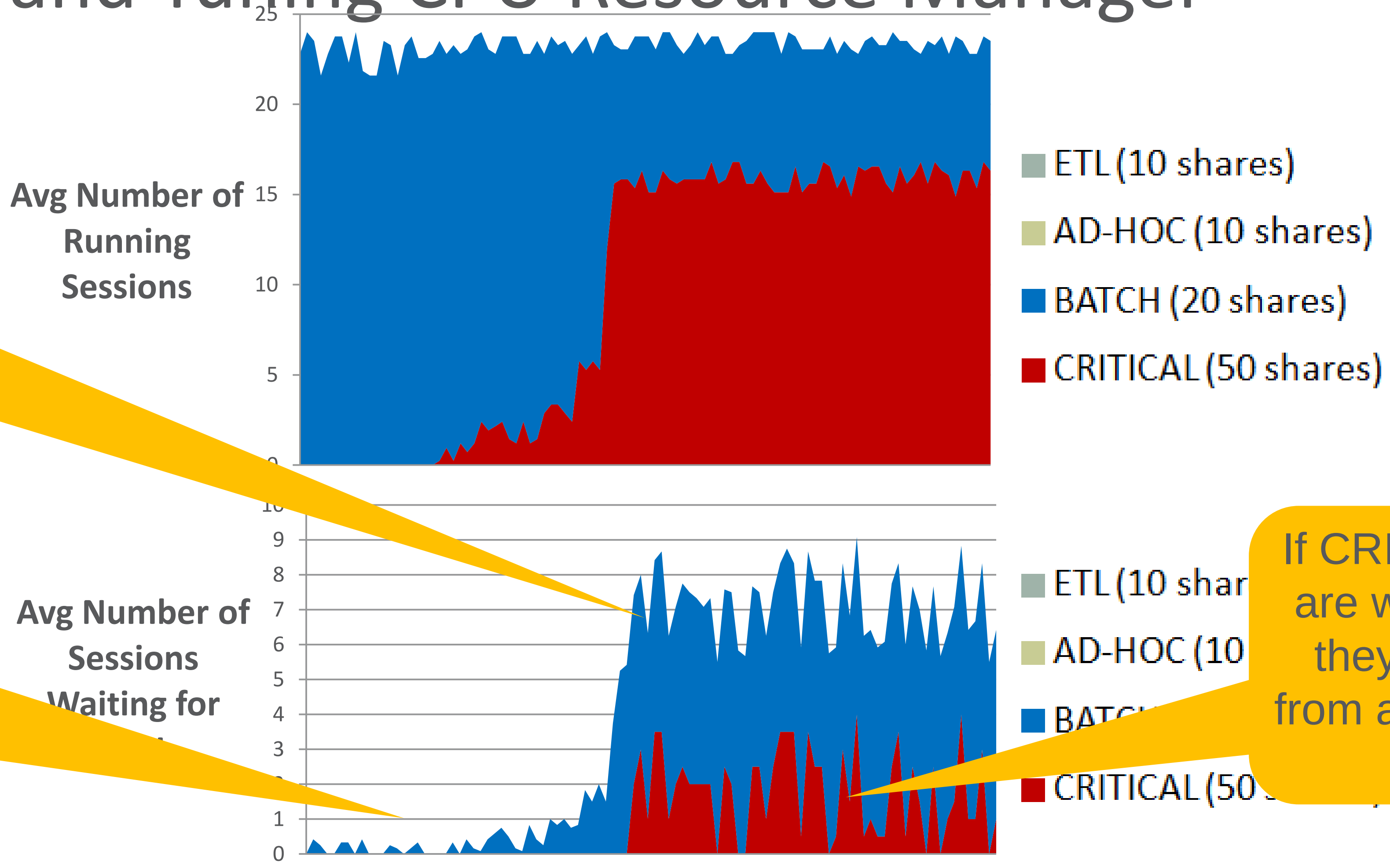
- Configuring a resource plan is an iterative process
  - Create a resource plan
  - Monitor application performance and Resource Manager metrics
  - Adjust resource allocations and re-monitor
- Changes to the resource plan are enforced immediately
  - Instance restart NOT required

Monitoring and Tuning CPU Resource Manager

Monitoring Basics

- v$rsrc_plan
  - Current resource plan
- v$rsrc_session_info
  - Current and historical statistics for each session
- v$rsrc_consumer_group
  - Current and historical statistics for each consumer group
- v$rsrcmgrmetric_history, v$rsrcpdbmetric_history
  - Minute-by-minute metrics for each consumer group or PDB

Monitoring and Tuning CPU Resource Manager



If *any* sessions are waiting for CPU, the database instance would benefit from a larger server.

If sessions aren't waiting for CPU, Resource Manager isn't actively managing

If CRITICAL sessions are waiting for CPU, they would benefit from a larger resource allocation.

Monitor using v$rsrcmgrmetric_history, v$rsrcpdbmetric_history or Enterprise Manager

Managing Runaway Queries

Managing Runaway Queries

- Runaway queries can be caused by
  - Badly written SQL
  - Bad execution plans
- Severely impact performance of well-behaved queries
- Very hard to completely eradicate!

Managing Runaway Queries

Configure by Consumer Group

Define runaway query thresholds:

- ✓ Estimated execution time
- ✓ Elapsed time **New in 12c**
- ✓ Amount of CPU time used
- ✓ Number of I/Os issued
- ✓ Bytes of I/O issued
- ✓ Number of logical I/Os issued **New in 12c**

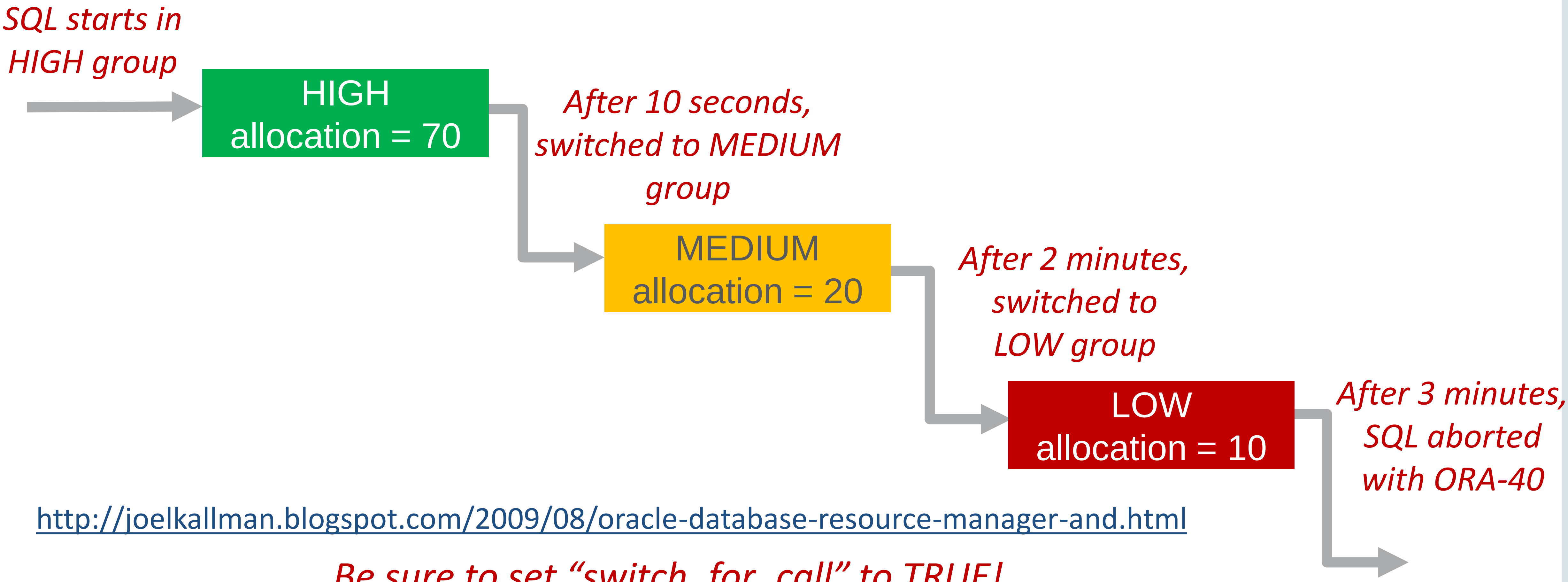
Manage runaway queries:

- ✓ Switch to a lower-priority consumer group
- ✓ Abort call
- ✓ Kill session
- ✓ Log to SQL Monitor **New in 12c**

ORACLE

Managing Runaway Queries

Workload Management in a Cloud Database



SQL starts in HIGH group

HIGH
allocation = 70

After 10 seconds, switched to MEDIUM group

MEDIUM
allocation = 20

After 2 minutes, switched to LOW group

LOW
allocation = 10

After 3 minutes, SQL aborted with ORA-40

http://joelkallman.blogspot.com/2009/08/oracle-database-resource-manager-and.html

Be sure to set “switch_for_call” to TRUE!

Monitoring Runaway Queries with v$sql_monitor

New in 12c

Column	Description
RM_CONSUMER_GROUP	Current consumer group name
RM_LAST_ACTION	Action that was taken (if any): SWITCH TO <consumer group name> CANCEL_SQL KILL_SESSION LOG_ONLY New in 12c
RM_LAST_ACTION_REASON	The reason why the action above was taken: SWITCH_CPU_TIME ⍰SWITCH_IO_REQS ⍰SWITCH_IO_MBS ⍰SWITCH_ELAPSED_TIME New in 12c ⍰SWITCH_IO_LOGICAL New in 12c
RM_LAST_ACTION_TIME	The time at which this action was taken

ORACLE