Resource Manager: Best Practices

Oracle Open World, Session

Sue K. Lee, Director of Development
Akshay Shah, Principal Member of Technical Staff

Oracle Server Technologies

October 1, 2014

ORACLE

Safe Harbor Statement

The preceding 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

- Resource Manager Use Cases
  - Consolidation
  - Mixed Workloads
- Managing Resources
  - CPU
  - Exadata Disk and Flash
  - Standby Databases
  - Runaway Queries

ORACLE

Consolidation and Resource Manager

Why Consolidate?

- Efficient server and storage utilization
  - Each generation of servers and storage is more powerful
  - Typical database workload may not fully utilize hardware
  - Database workloads are often bursty, with long periods of low utilization
  - Lots of test, development, and non-critical databases
- Fewer systems to administer
  - Reduce effort for patching and maintenance

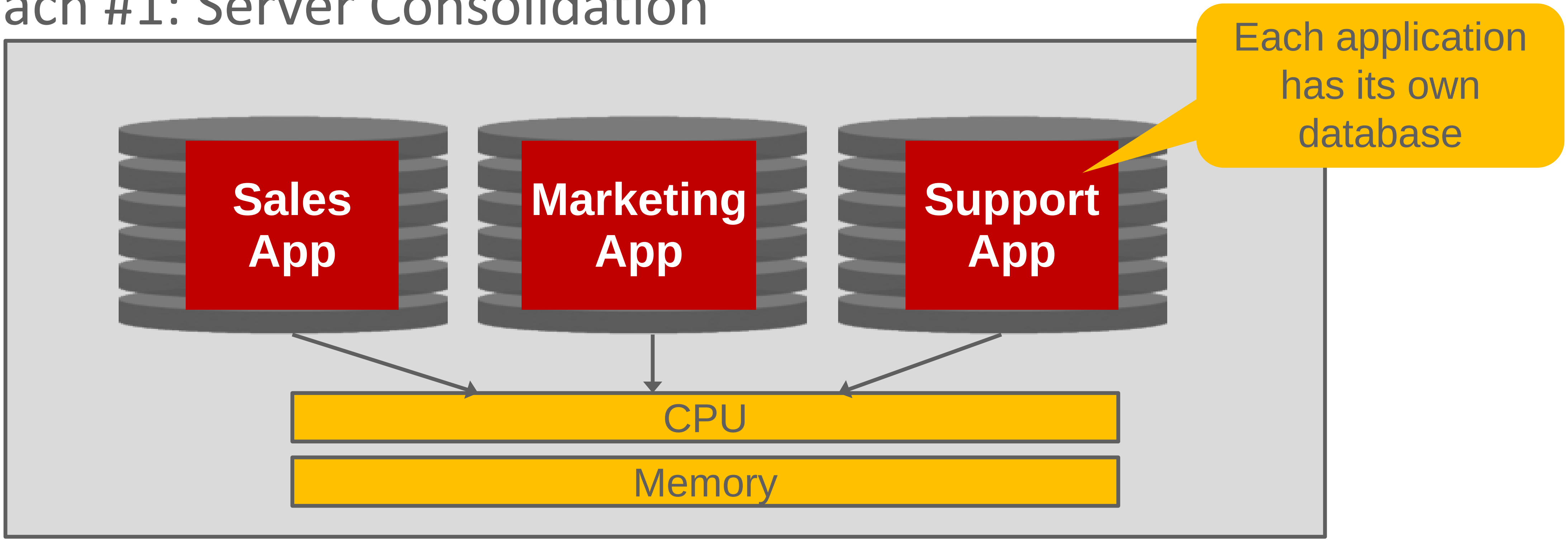
ORACLE

Consolidation Challenges

- Database users apprehensive about consolidation
  - Demand performance guarantees
- Workload surges from one application can affect others
  - Excessive CPU, PGA, or I/O usage
  - Surges can originate from heavy application usage or a single runaway query
- DBAs want to control resource usage
  - Fair access to resources
  - Hosted environments – “get what you pay for”

ORACLE

Approach #1: Server Consolidation



Multiple databases share a server

- ✓ Application isolation
- ✓ Each application is independently maintained and upgraded

But…

- Backgrounds and SGA are not shared – inefficient resource utilization
- Each application is independently maintained and upgraded

ORACLE

Approach #2: Schema Consolidation

Multiple applications share a database

- ✓ Backgrounds and all database resources are shared – efficient resource utilization
- ✓ One database to administer

But…

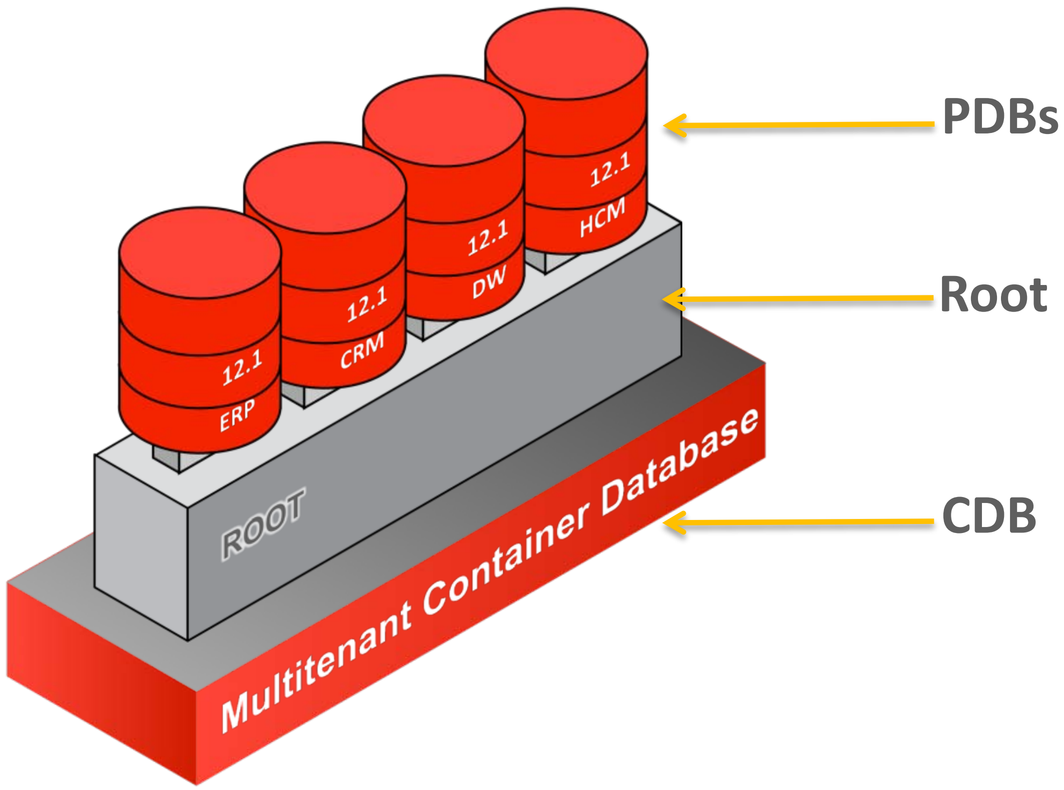
- Object name collisions due to shared dictionary
- May require application-level changes

ORACLE

New in 12c!

Approach #3: Multi-Tenant Container Database

- Container Database (CDB)
  - Container for up to 252 Pluggable Databases
  - Shared backgrounds and SGA
  - One database to administer
- Pluggable Database (PDB)
  - Feels and operates identically to a non-CDB
  - Any database can be converted into a PDB
  - Root is a special PDB used to administer the CDB



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)

ORACLE

Configuring Workloads with Consumer Groups

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

Session		Consumer Group
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

ORACLE

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)

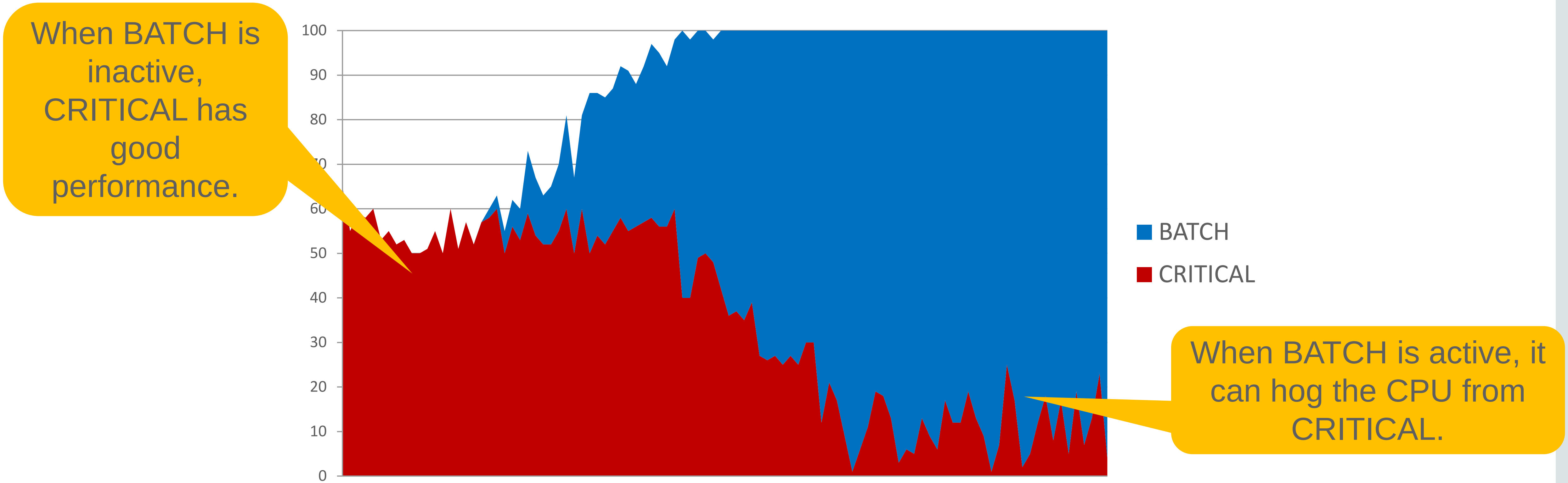
  "inmemory" (in-memory population, defaults to ORA$AUTOTASK) **New in 12.1.0.2**

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

ORACLE

Managing CPU within a Database

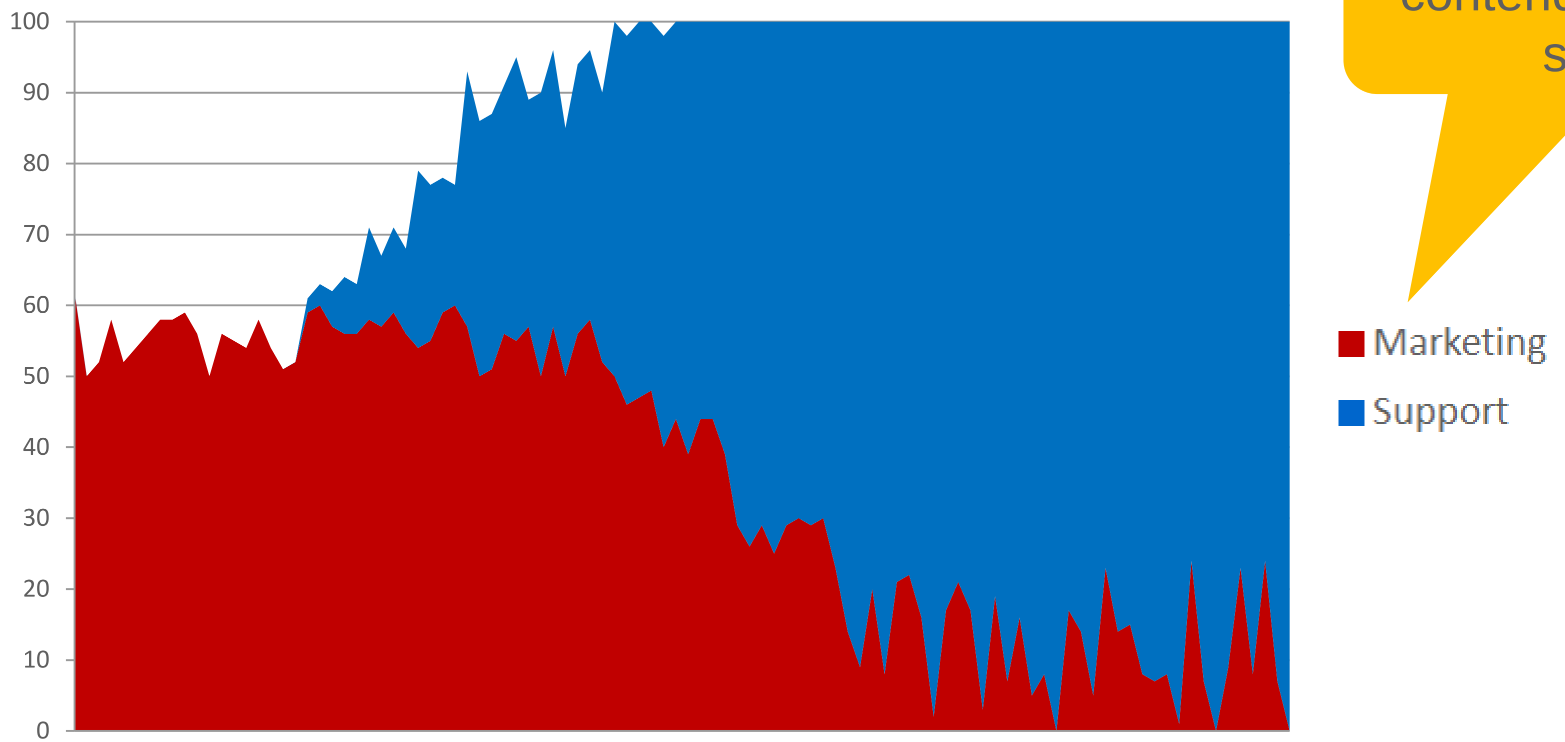
Managing CPU



CPU Utilization by Consumer Group

Without Resource Manager, Consumer Groups with many active sessions, parallel execution, and CPU-intensive work hog the CPU!

Managing CPU



CPU Utilization by Pluggable Database

Pluggable Databases contend for CPU in the same way

Without Resource Manager, Pluggable Databases with
many active sessions, parallel execution, and CPU-intensive work hog the CPU!

Managing CPU

- CPU Resource Manager supports 3 common scenarios
  - Manage workloads within a database - Database Resource Plan
  - Manage PDBs within a CDB - CDB Resource Plan **New in 12c**
  - Manage database instances sharing a server - Instance Caging
- All of these tools can be used together

ORACLE

Managing CPU

Consumer Groups within a Database

A Resource Plan uses “shares” to specify how CPU is distributed between Consumer Groups

Database Resource Plan				
Consumer Group	**Shares***	**Utilization Limit**	**Guaranteed CPU**	**Maximum CPU**
Critical	5		50%	100%
Batch	2		20%	100%
Maintenance	2	90%	20%	90%
Other	1		10%	100%

*In 10g and 11g, use “mgmt_p1” instead of shares.
They work the same way.

ORACLE

Managing CPU

Consumer Groups within a Database

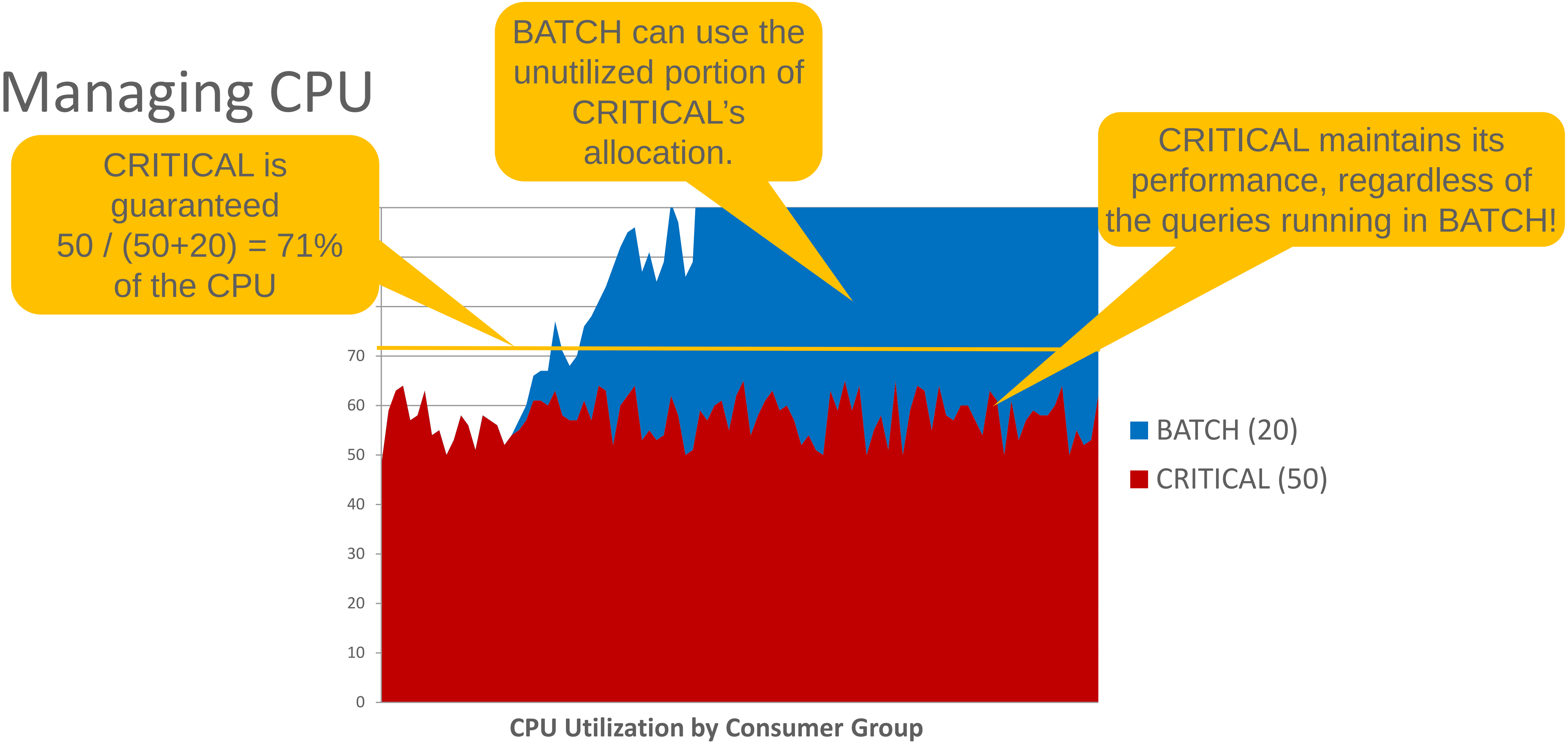
"Critical" is guaranteed 50% of the CPU.
If it doesn't use it, someone else can.

Database Resource Plan				
Consumer Group	**Shares***	**Utilization Limit**	**Guaranteed CPU**	**Maximum CPU**
Critical	5		5/(5+2+2+ 1) = 50%	100%
Batch	2		20%	100%
Maintenance	2	90%	20%	90%
Other	1		10%	100%

*In 10g and 11g, use "mgmt_p1" instead of shares.
They work the same way.

ORACLE

Managing CPU



With Resource Manager, your resource plan controls how the CPU is used!

Managing CPU

Pluggable Databases

A CDB Resource Plan uses “shares” to specify how CPU is distributed between PDBs.

“Sales” is guaranteed 50% of the CPU.
If it doesn’t use it, someone else can.

CDB Resource Plan				
Pluggable Database	**Shares**	**Utilization Limit**	**Guaranteed CPU**	**Maximum CPU**
Sales	2		2/(2+1+1) = 50%	100%
Marketing	1	75%	25%	75%
Support	1	75%	25%	75%

PDBs are managed just like Consumer Groups.

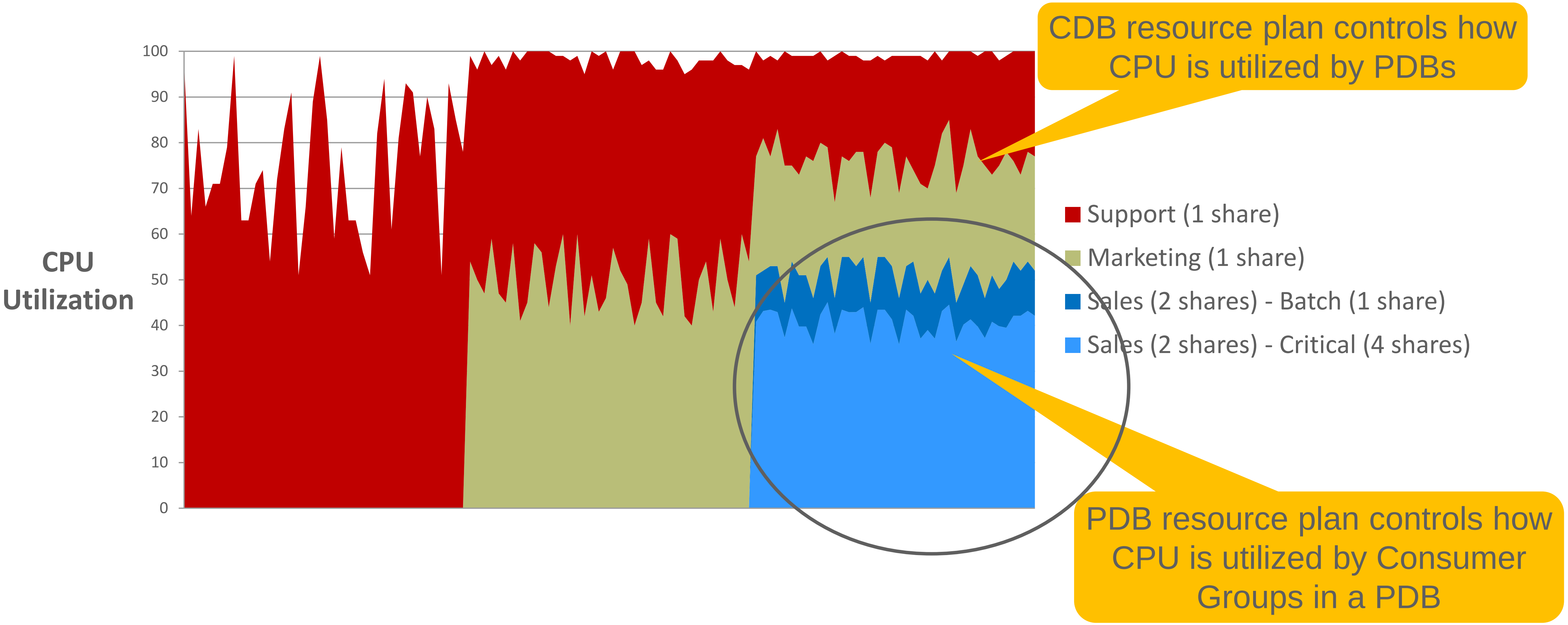
Managing CPU

Pluggable Databases

CDB Resource Plan				
Pluggable Database	**Shares**	**Utilization Limit**	**Guaranteed CPU**	**Maximum CPU**
Sales	2		2/(2+1+1) = 50%	100%
Marketing	default (1)	default (75%)	25%	75%
Support	default (1)	default (75%)	25%	75%
(Default directive)	1	75%		

Configure a "default directive":
the default shares and utilization limit for a PDB

How Do CDB and PDB Resource Plans Work Together?



ORACLE

Managing CPU

Utilization Limits for Pluggable Databases

Utilization Limits are typically used in Cloud Consolidations to enforce “pay for performance”.

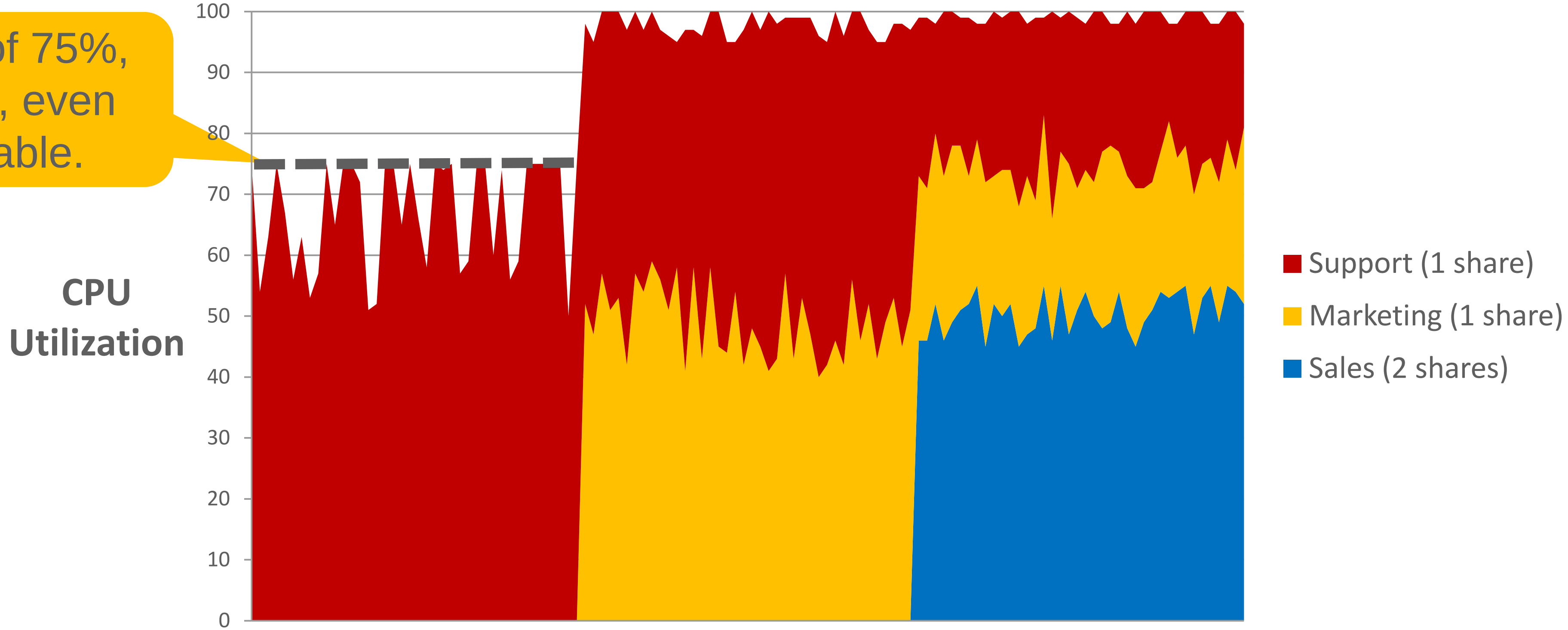
CDB Resource Plan				
Pluggable Database	**Shares**	**Utilization Limit**	**Guaranteed CPU**	**Maximum CPU**
Sales	2		2/(2+1+1) = 50%	100%
Marketing	1	75%	25%	75%
Support	1	75%	25%	75%

ORACLE

Managing CPU

Utilization Limits for Pluggable Databases

With a utilization limit of 75%, SUPPORT is throttled, even though CPU is available.



Utilization Limits provide clients consistent performance.
They also restrict their resource usage, based on what the client paid

ORACLE

Configuring CPU Resource Manager

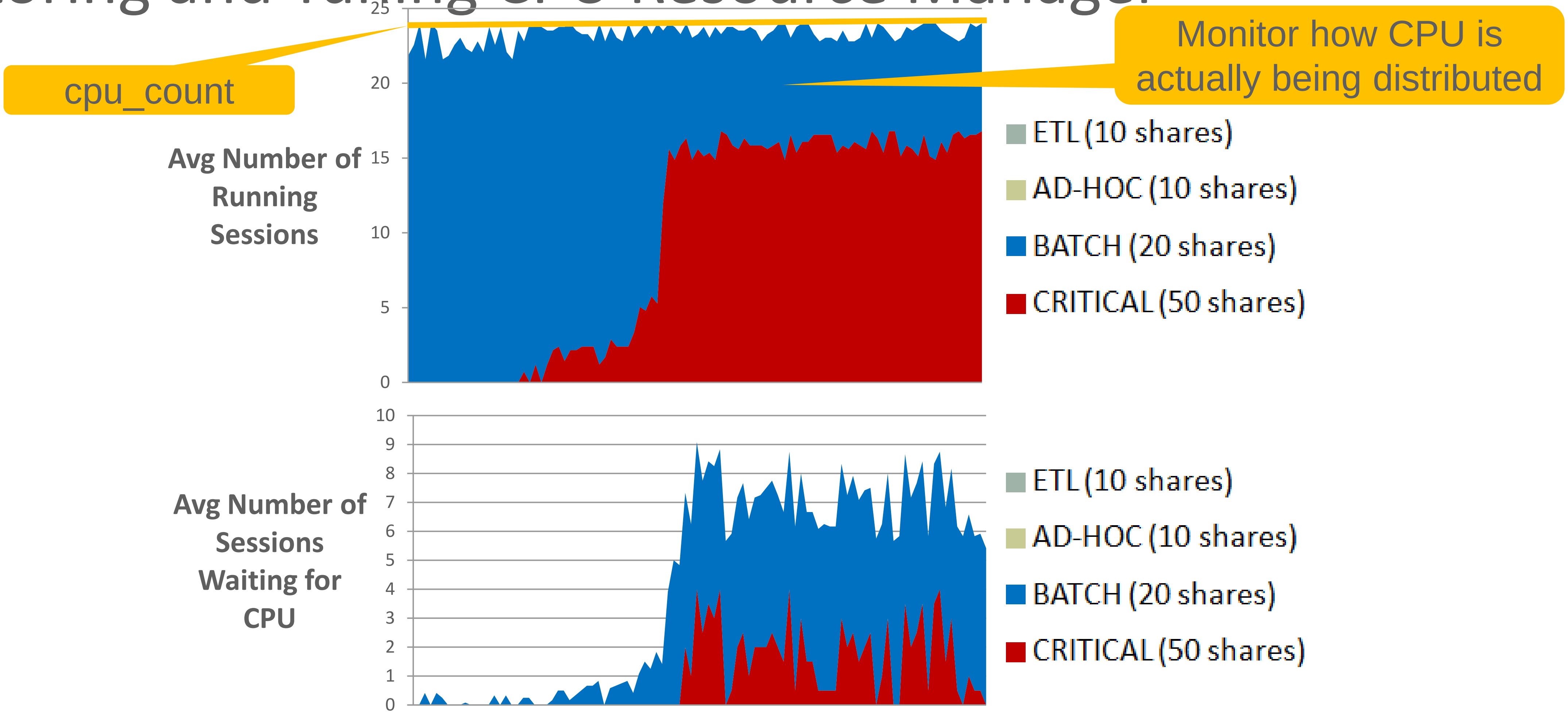
- Resource Manager can be configured with SQL or Enterprise Manager
- dbms_resource_manager
  - PL/SQL package
  - Create consumer group mapping rules
  - Create resource plans
  - Modify resource plans

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
- Any changes to the resource plan are enforced immediately
  - Instance restart NOT required

ORACLE

Monitoring and Tuning CPU Resource Manager



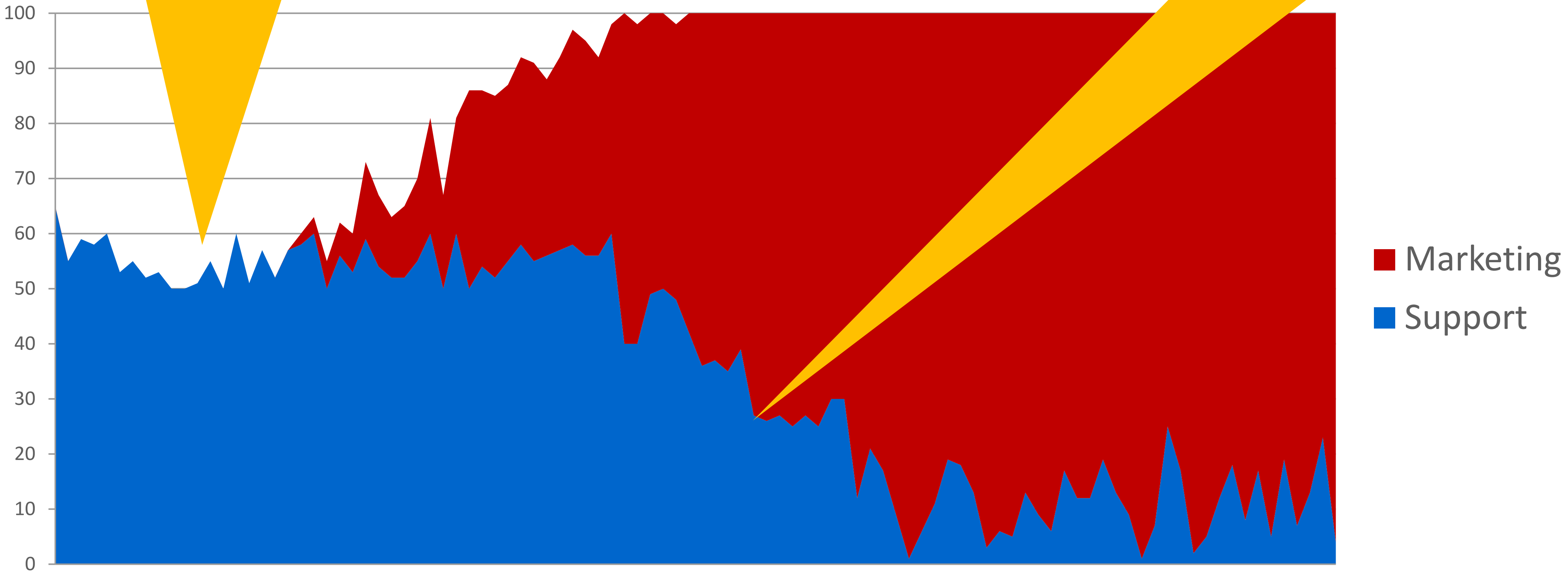
Monitor using v$rsrcmgrmetric_history or Enterprise Manager

Instance Caging

The Need for Instance Caging

Your database starts with enough CPU, resulting in good performance!

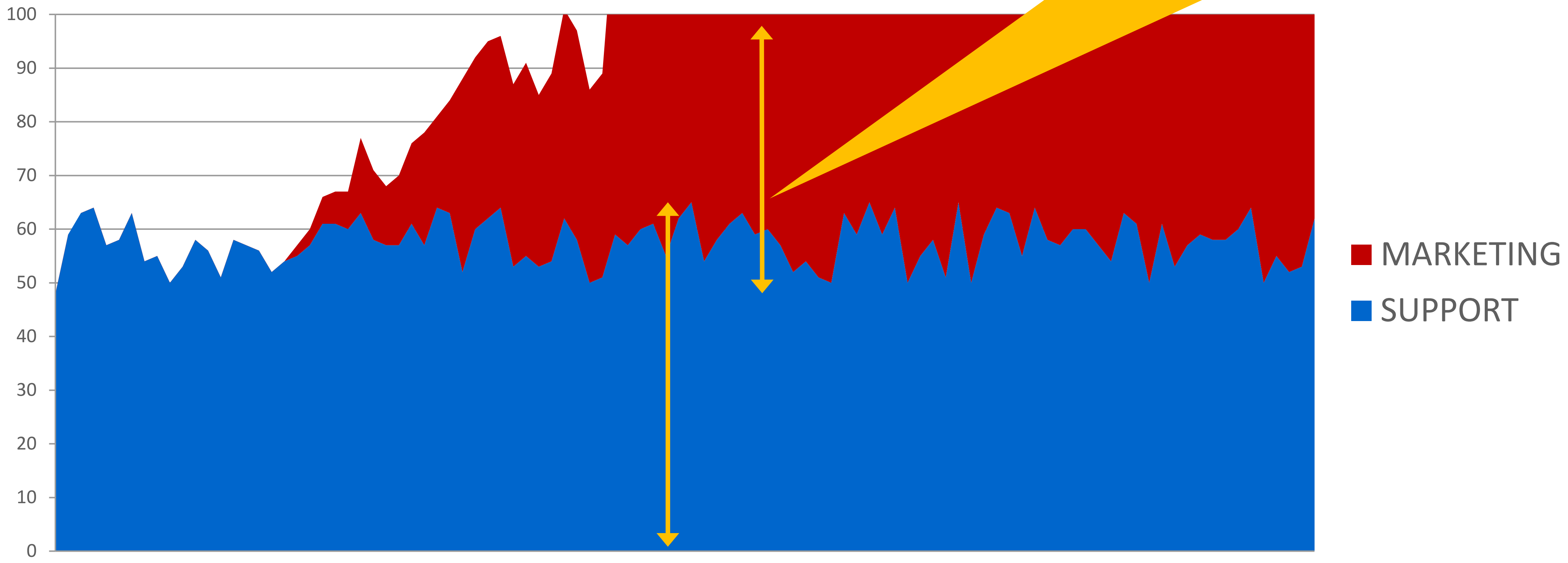
And then another database hogs the CPU, leaving you with not enough CPU and bad performance…



ORACLE

Manage CPU with Instance Caging

“Cage” a database by limiting the amount of CPU it can use.



Step 1: Set “cpu_count” to the max CPU threads the instance can use at any time

```
alter system set cpu_count = 8;
```

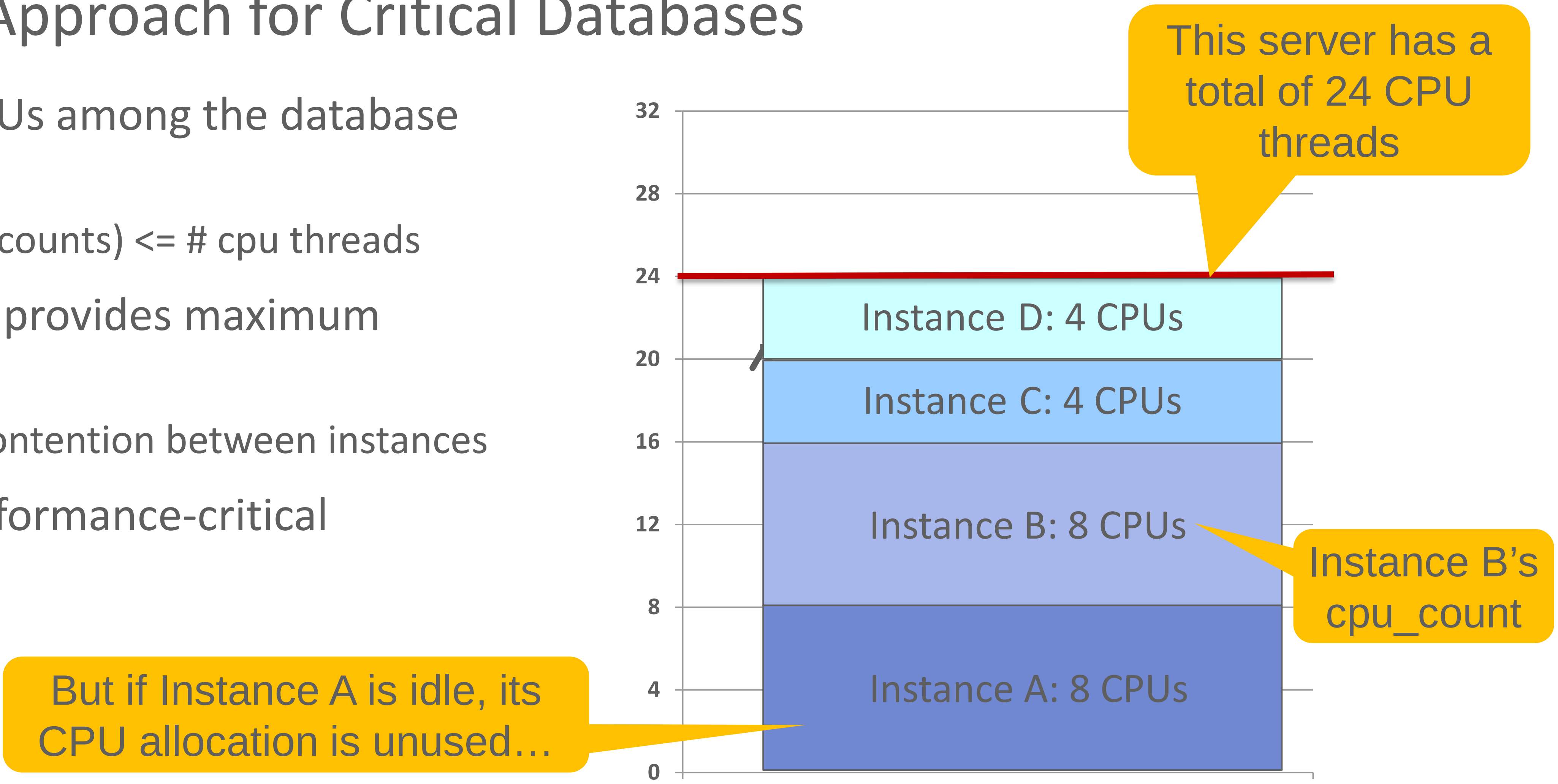
Step 2: Set “resource_manager_plan” to enable CPU Resource Manager

```
alter system set resource_manager_plan = ‘default_plan’;
```

ORACLE

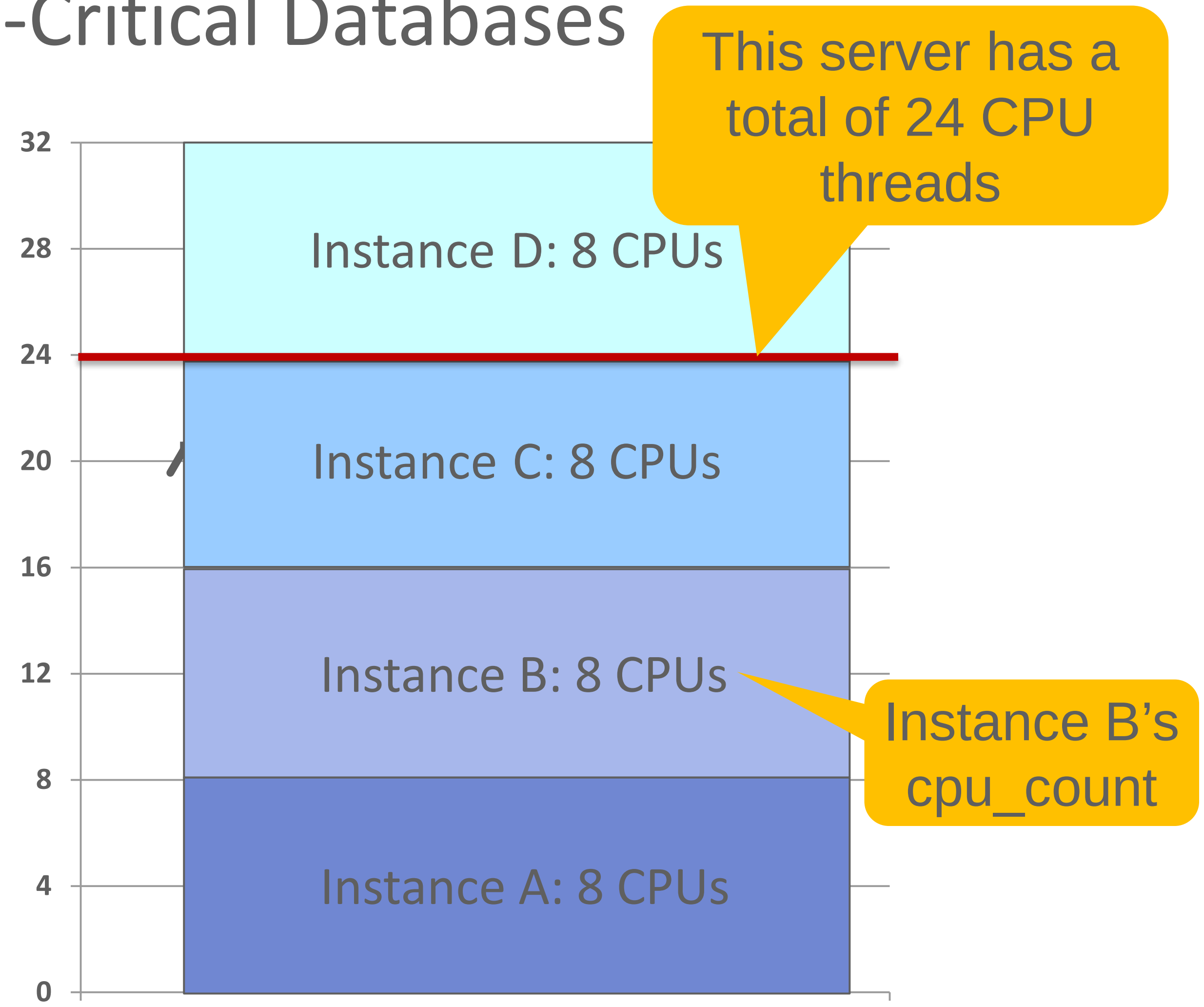
Partition Approach for Critical Databases

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



Over-Subscribe Approach for Non-Critical Databases

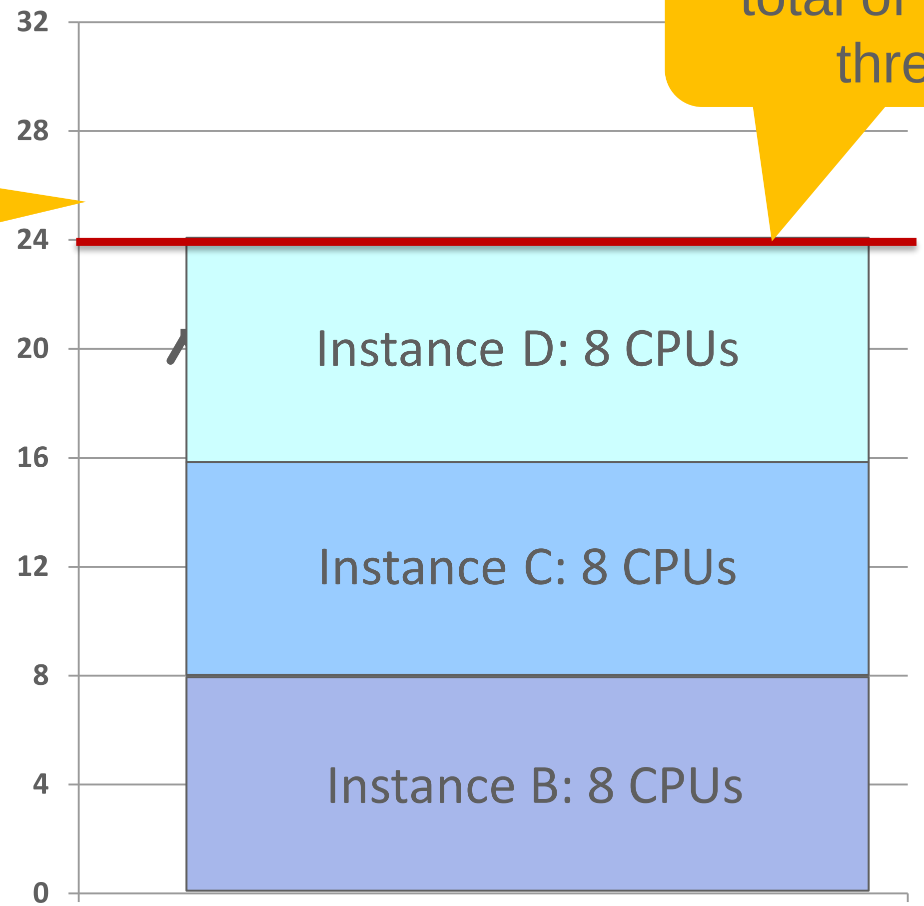
- Over-subscribe the CPUs among the database instances
  – sum(cpu_counts) <= 3 x # cpu threads
- Over-subscribing provides efficient CPU utilization
  – Some contention for CPU if databases are sufficiently loaded
- Best for non-critical databases



Over-Subscribe Approach for Non-Critical Databases

This server has a total of 24 CPU threads

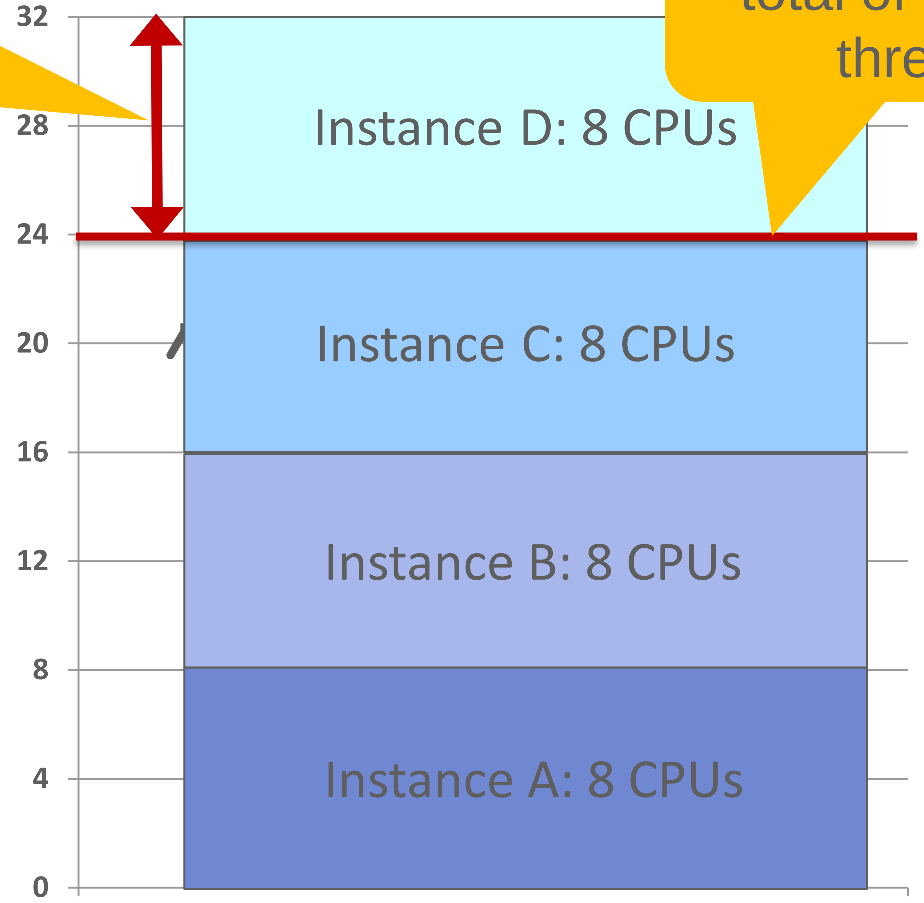
If Instance A is idle,
there is no CPU contention!
The databases can fully utilize the server.



Over-Subscribe Approach for Non-Critical Databases

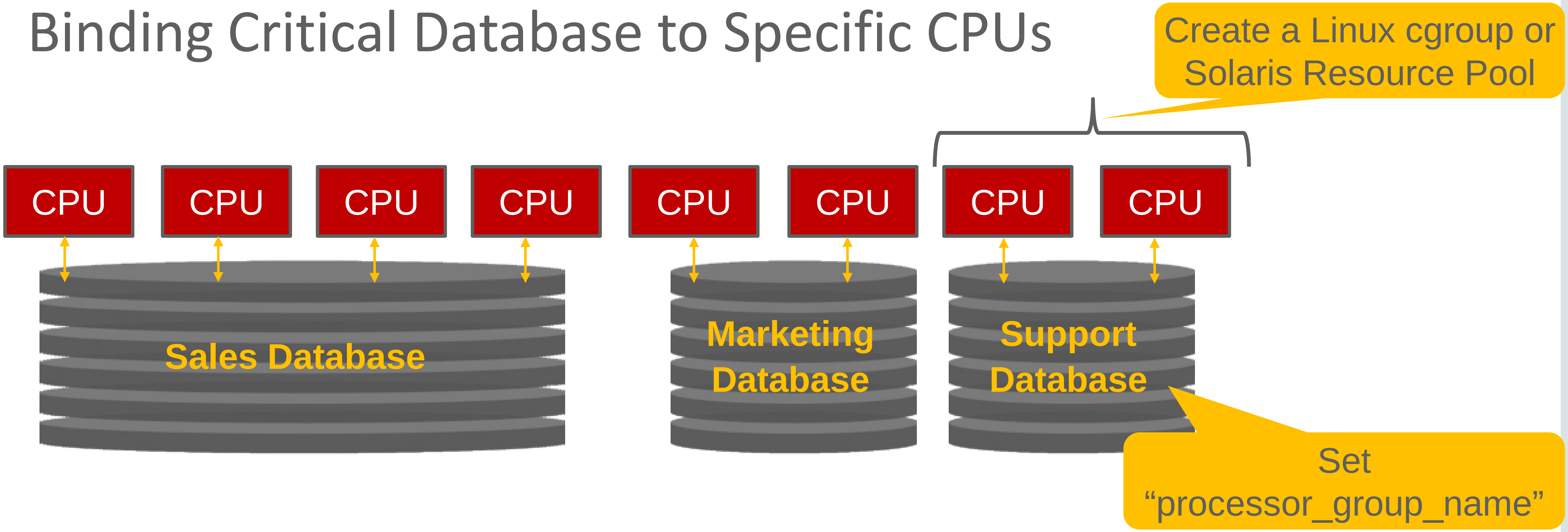
If all databases are active, there is some contention.
This is your maximum excess load.
Without Instance Caging, there is ***no*** limit on your maximum load!

This server has a total of 24 CPU threads



New in 12c!

Binding Critical Database to Specific CPUs

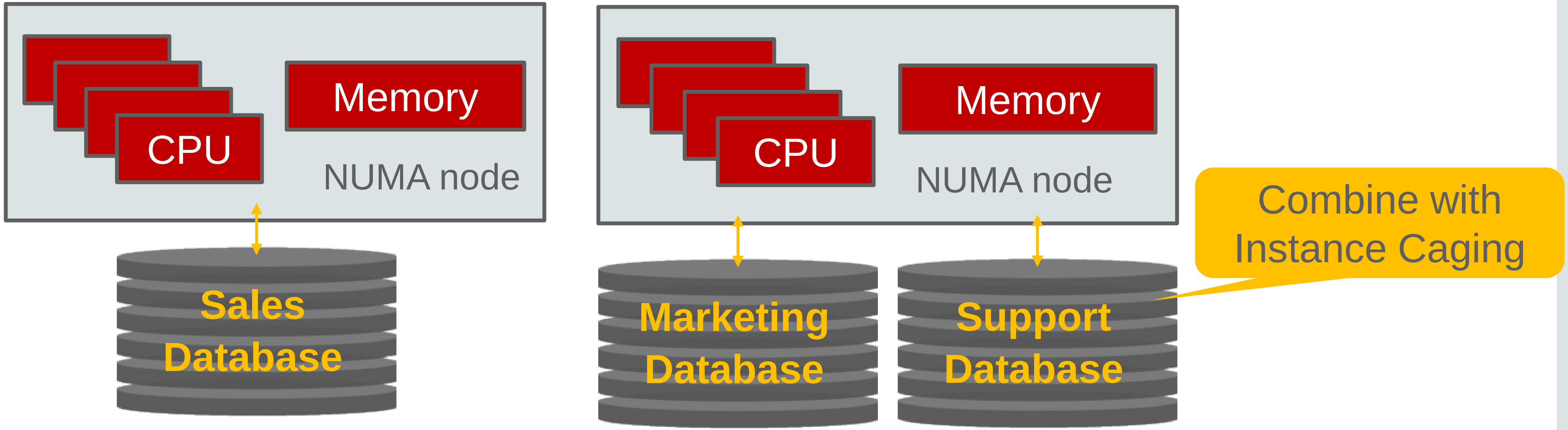


- With CPU binding, databases don't share CPU cores or caches.
- Additional isolation only useful for platinum databases…

See MOS note 1585184.1, 1928328.1 for step-by-step guide and best practices

New in 12c!

Binding Databases to Specific NUMA CPUs



- Bind database instance to specific NUMA nodes (Exadata X4-8)
- Or bind to specific CPU sockets (SPARC T-Series)
- Reduce expensive remote memory accesses for superior performance

See MOS note 1585184.1, 1928328.1 for step-by-step guide and best practices

New in 12c!

Binding Databases to Specific CPUs

To bind a database instance to CPUs or NUMA nodes

1) Specify the CPUs or NUMA nodes by creating a “processor group”
   – Linux cGroups
   – Solaris resource pools

   “pg1” = CPUs 0-3

2) Set the Oracle parameter “processor_group_name” to the name of this processor group
   – All Oracle backgrounds and foregrounds run on these CPUs
   – All SGA and PGA is allocated from these NUMA nodes

   processor_group_name = pg1

See MOS note 1585184.1 for step-by-step guide and best practices

ORACLE

Instance Caging

Best Practices

- Keep the strategy simple
  - Initial cpu_count settings can be a guess
- Monitor actual CPU usage and then tweak
  - cpu_count can be adjusted dynamically – no database bounce needed!
  - If over-subscribing, monitor the server to make sure it's not over-loaded
  - Avoid large changes to cpu_count
- cpu_count corresponds to CPU threads, not cores!
- Beware of over-subscribing on SPARC T-Series processors
- Don't set cpu_count to 1
  - Makes database instance susceptible to hangs or poor performance

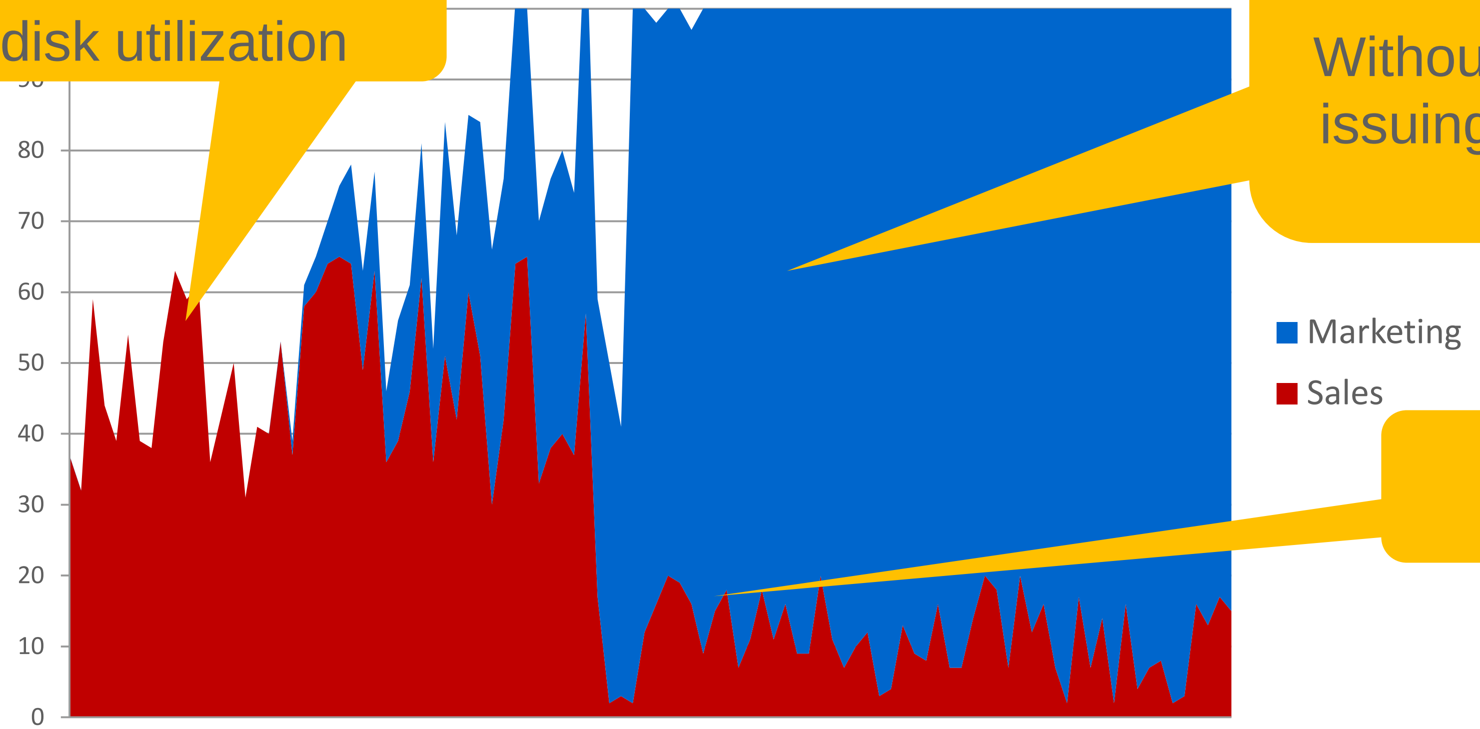
ORACLE

Managing Disk I/O – Exadata

Use Case for IORM: Scan Workloads

Without IORM

SALES starts at 50% disk utilization



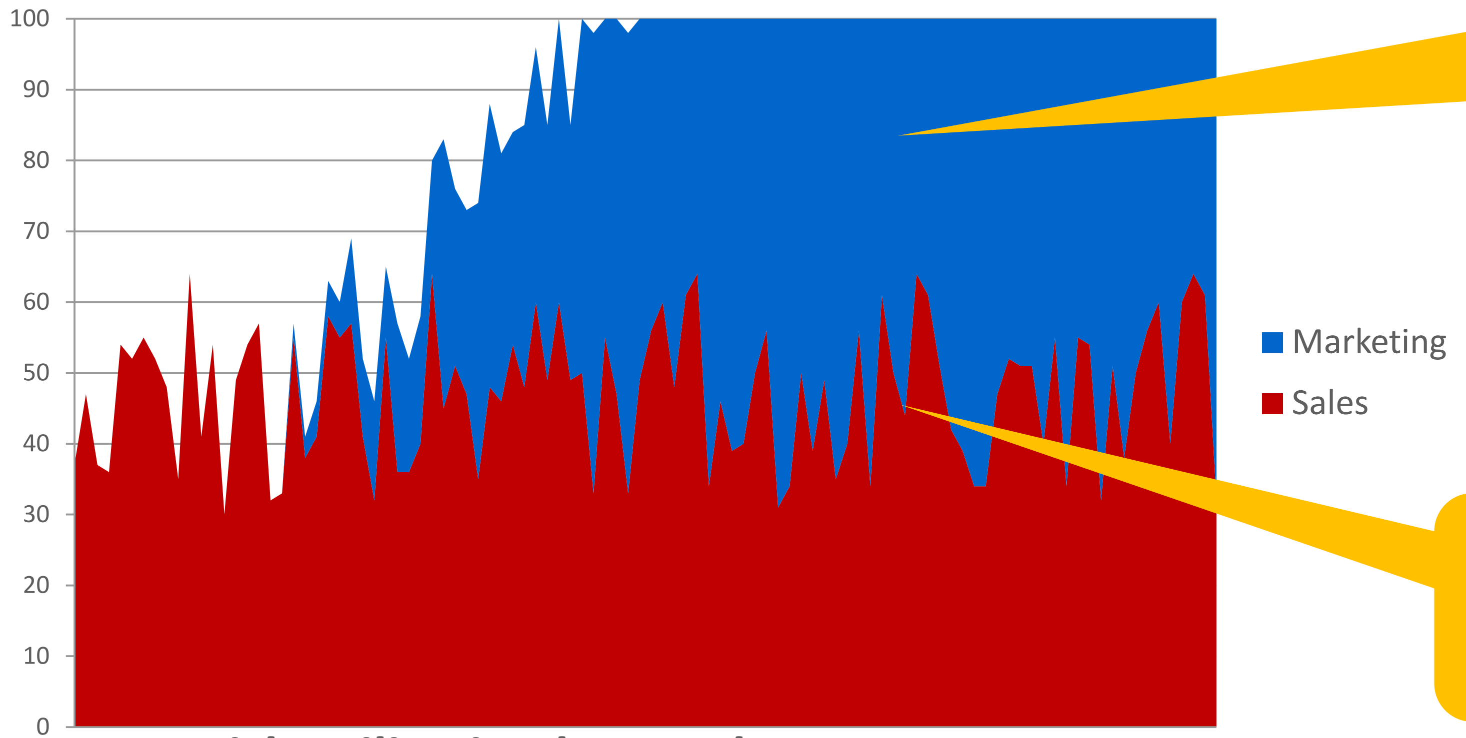
MARKETING database hogs the disk bandwidth.
Without Exadata IORM, databases issuing the most load get the most bandwidth.

SALES database sees a drop in throughput.

Disk Utilization by Database

Use Case for IORM: Scan Workloads

With IORM



Disk Utilization by Database

One database can use excess disk bandwidth without affecting the other!

IORM ensures predictable throughput for each database.

ORACLE

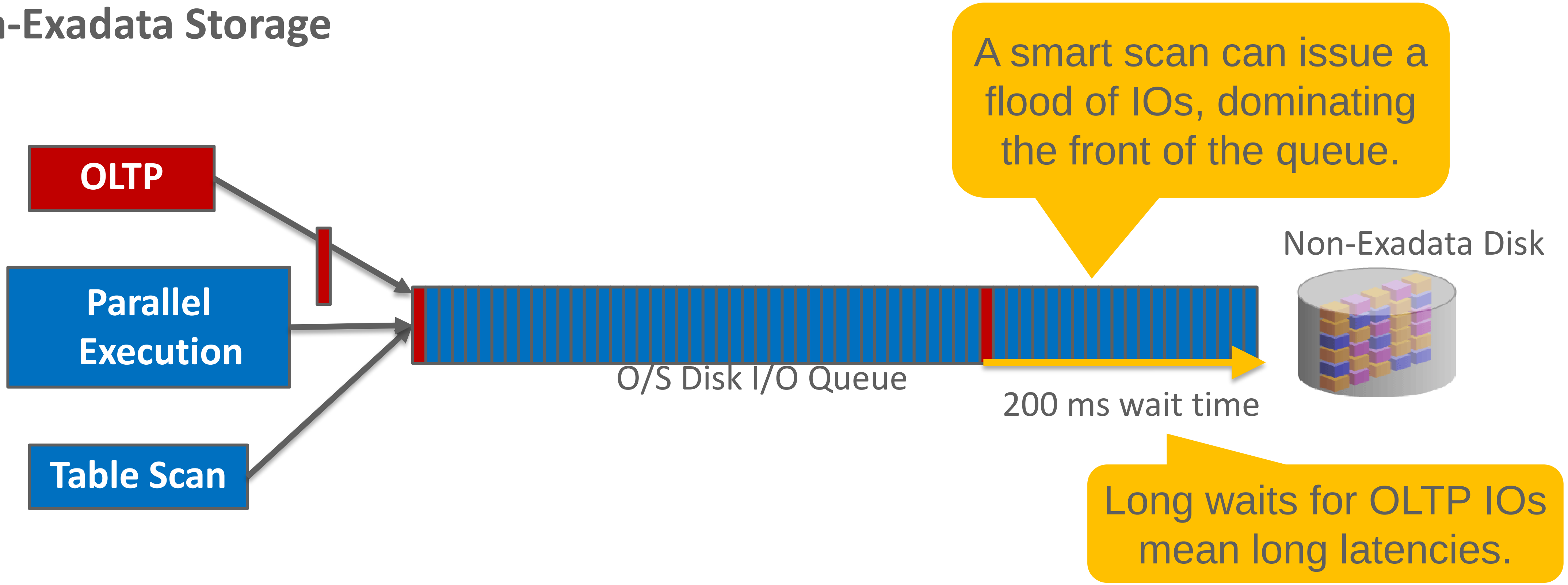
Use Case for IORM: High Disk Latencies for OLTP

		% of Waits							
Event	Total Waits	<1ms	<2ms	<4ms	<8ms	<16ms	<32ms	<=1s	>1s
cell single block physical	213.7	90.1	3.3	.8	.1	.1	2.0	3.6	
log file parallel write	951.5	93.9	2.6	1.3	1.1	1.0	.0	.0	

- On Exadata, most OLTP I/Os are serviced from flash, except
  - Flash cache misses
  - Log writes when flash is busy
- During a smart scan, OLTP disk I/Os can be very slow
- Enable IORM to avoid outliers. Use the “objective” knob to enforce low disk latencies.

Use Case for IORM: High Disk Latencies for OLTP

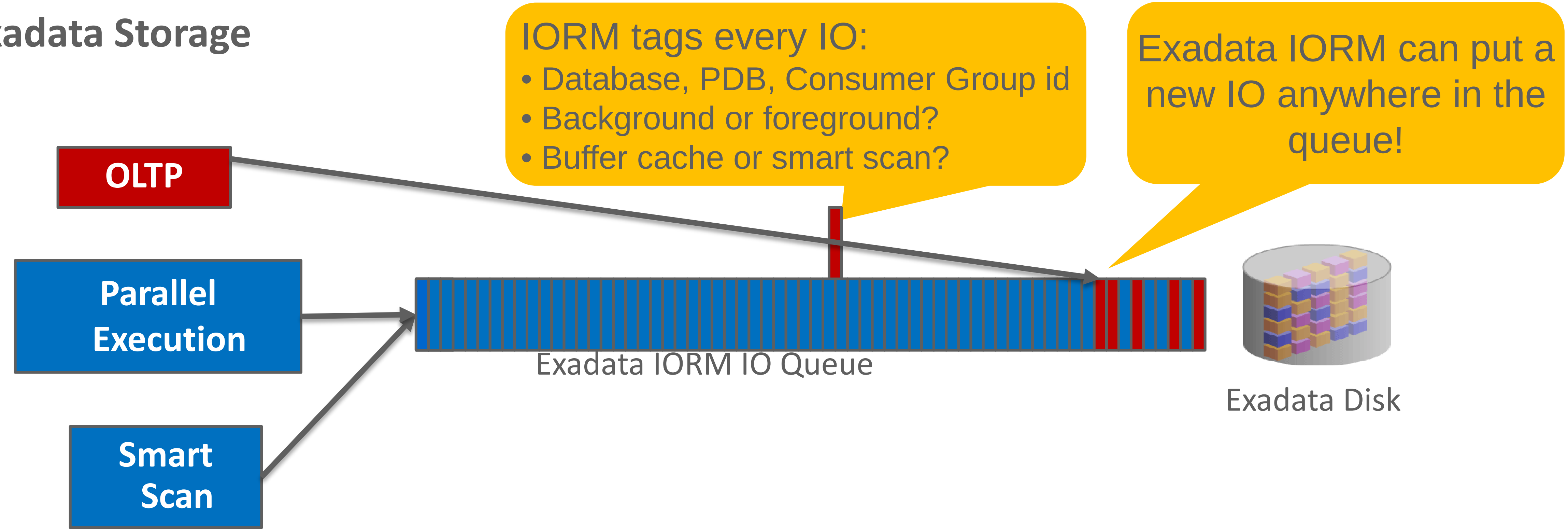
Non-Exadata Storage



I/Os are processed in the order received.
The O/S usually lets small OLTP I/Os cut towards the front of the queue.
But waiting behind the I/Os in front can still take 100s of ms.

Use Case for IORM: High Disk Latencies for OLTP

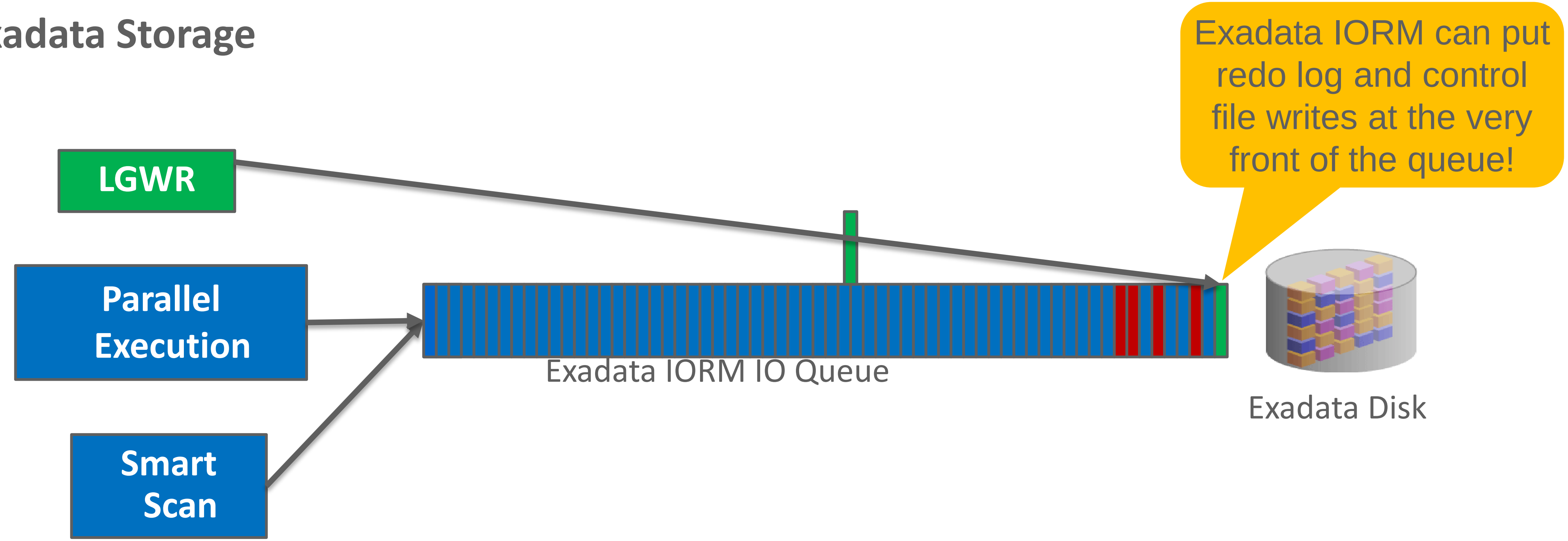
Exadata Storage



Exadata IORM positions the I/O in the queue,
based on the I/O's properties and the Resource Plan

Use Case for IORM: High Disk Latencies for OLTP

Exadata Storage



Background and ASM IOs are prioritized, based on why they were issued.
DBWR IOs for checkpoint: high priority.
DBWR IOs when the buffer cache has plenty of free buffers: low priority.

ORACLE

Exadata IORM

- Exadata I/O Resource Manager
  - Disk I/O scheduler
  - Industry pioneer for disk I/O management
  - For Exadata only!
- Enables multiple workloads to share storage in a predictable way
  - Prioritizes critical workloads over non-critical workloads
  - Allows fair sharing for database consolidation
- Ensures low disk latency when OLTP is actively using disk
- Ensures high disk throughput for throughput-intensive workloads

Exadata IORM

To enable IORM

1) Set IORM objective from “basic” to “auto”, using CellCLI or EM Cloud Control 12c

    CellCLI> alter iormplan objective = auto

    – If workload is latency sensitive, ‘auto’ optimizes for low latency
    – If workload is throughput oriented, ‘auto’ optimizes for maximum throughput

2) Enable a resource plan

ORACLE

Exadata IORM

Types of Resource Plans

Database and CDB resource plans are configured on the database.
They are shared by CPU and I/O Resource Manager.

Consolidation Scenario	CPU Resource Manager	I/O Resource Manager
Consumer Group workloads sharing a database	Database Resource Plan (configured on database)	
PDBs sharing a CDB New in 12c	CDB Resource Plan (configured on database)	
Database Instances sharing a server or storage	Instance Caging (configured on database)	Inter-Database IORM Plan (configured on storage cell)

ORACLE

Exadata IORM

Types of Resource Plans

Inter-Database Plans are only for IORM.
They are configured on the storage cell.

Consolidation Scenario	CPU Resource Manager	I/O Resou nager
Consumer Group workloads sharing a database	Database Resource Plan (configured on database)	
PDBs sharing a CDB New in 12c	CDB Resource Plan (configured on database)	
Database Instances sharing a server or storage	Instance Caging (configured on database)	Inter-Database IORM Plan (configured on storage cell)

ORACLE

IORM for Workloads within a Database

The Database Resource Plan is used to manage both CPU and Disk I/O

Database Resource Plan				
Consumer Group	**Resource Allocation (shares or mgmt_p1)**	**Utilization Limit**	**Guaranteed Disk Utilization**	**Maximum Disk Utilization**
Critical	4		4/(4+1+1+1+1)=50%	100%
Batch	1		12%	100%
AdHoc	1	50%	12%	50%
ETL	1		12%	100%
Other	1		12%	100%

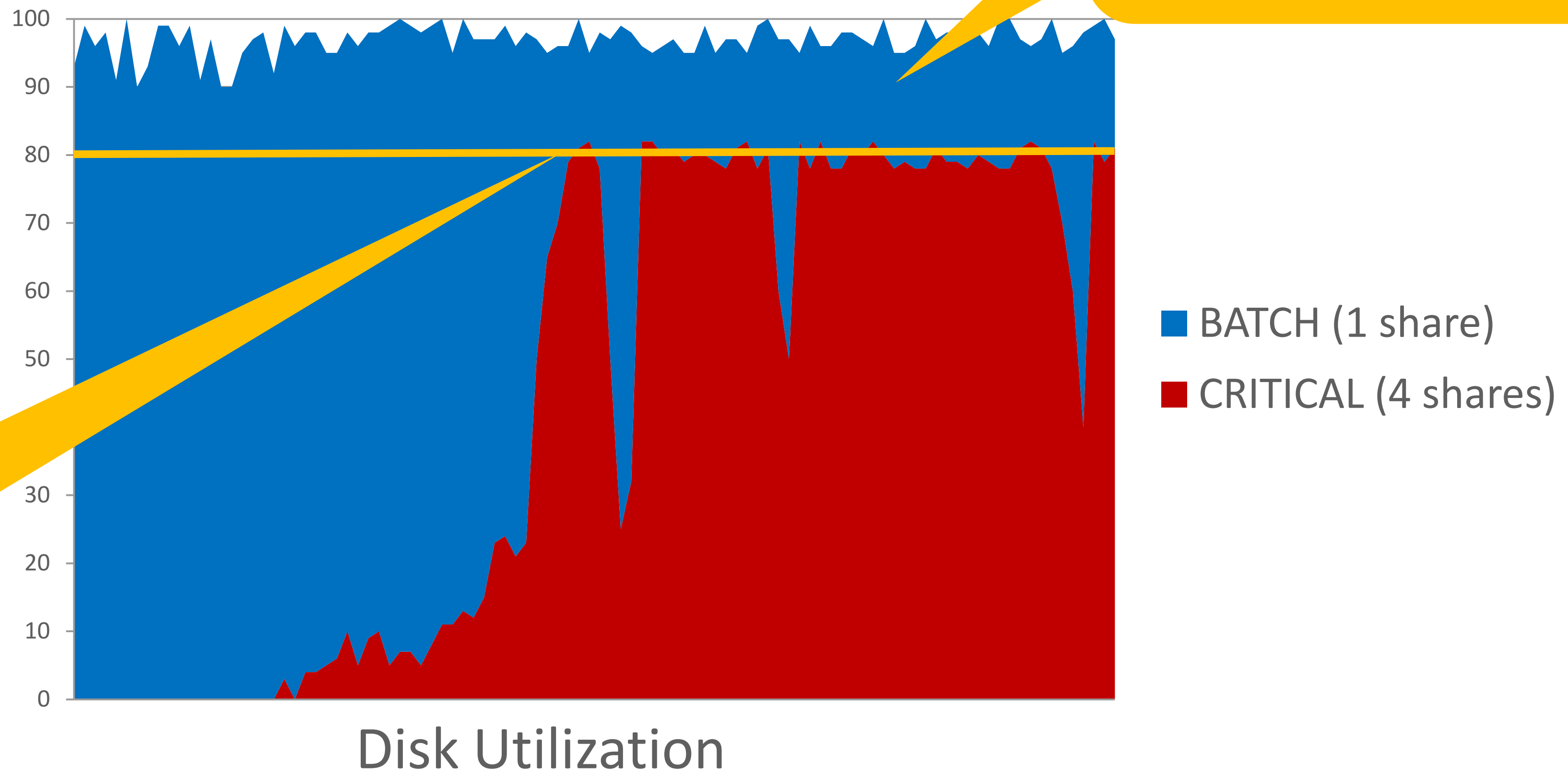
ORACLE

IORM for Workloads within a Database

Configure with Database Resource Plan

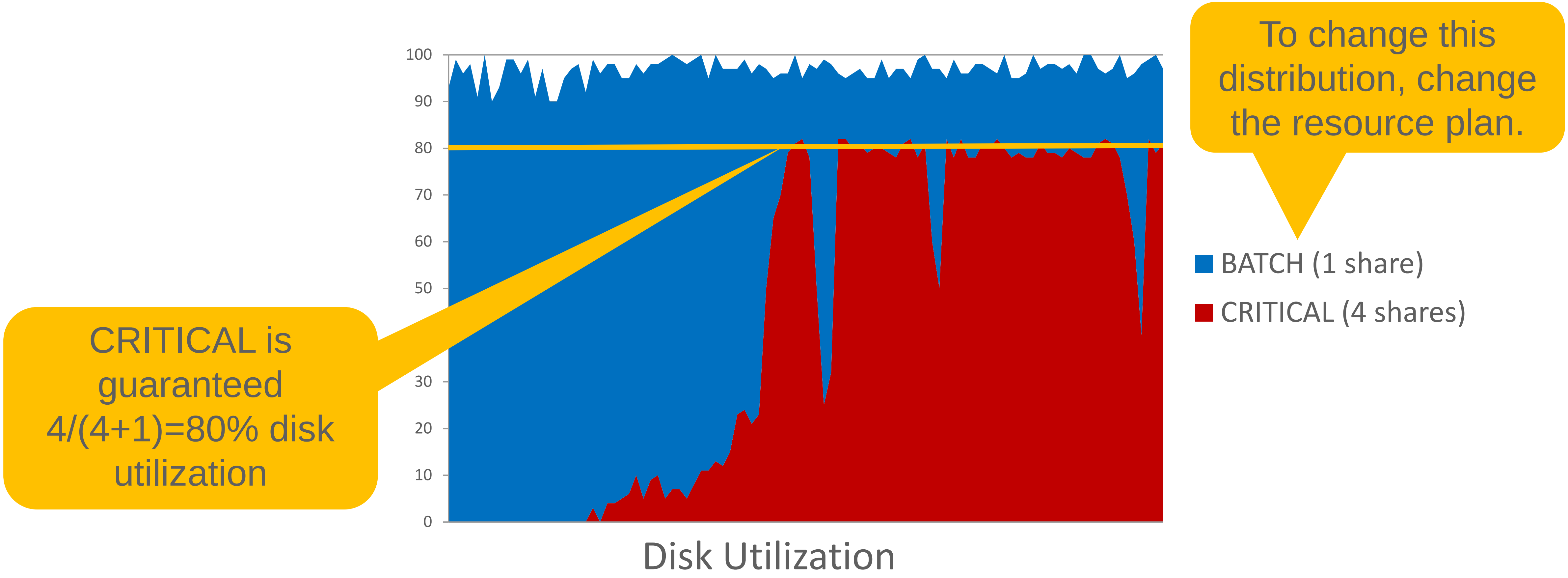
Non-critical batch operations use remaining disk resources

CRITICAL is guaranteed 4/(4+1)=80% disk utilization



IORM for Workloads within a Database

Configure with Database Resource Plan



IORM for Multiple Databases

Inter-Database IORM Plan				
Database	Shares	Utilization Limit	Guaranteed Disk Utilization	Maximum Disk Utilization
Sales	2		2/(2+1+1)=50%	100%
Support	1		25%	100%
Marketing	1	50%	25%	50%
Default	1	25%		

Databases not listed in the plan use the 'default' values

Without an inter-database plan, all databases are scheduled equally

ORACLE

IORM Utilization Limits

IORM can limit the disk utilization of a consumer group, PDB, or database.
Useful for enforcing "pay for performance" in cloud environments.

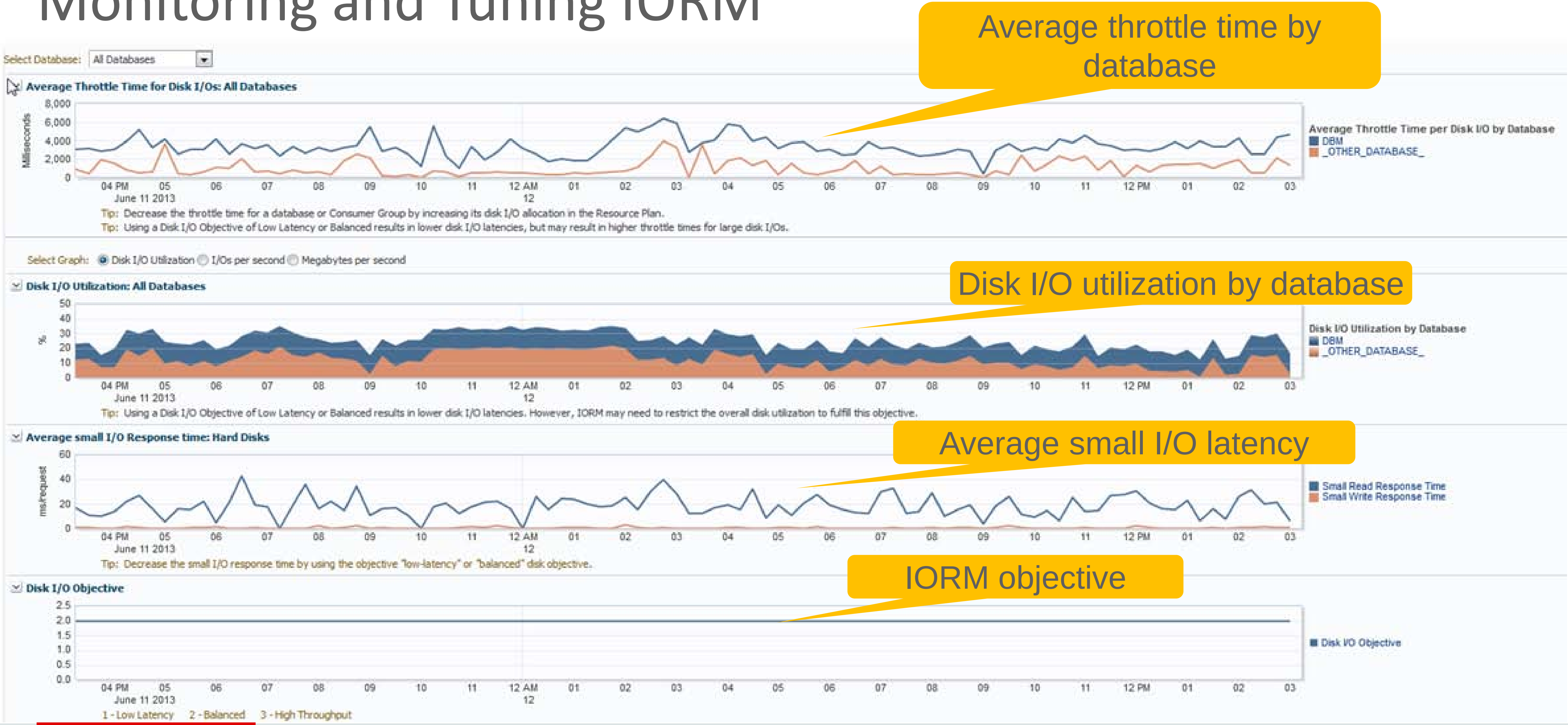
Inter-Database IORM Plan		
Database	**Shares**	**Utilization Limit**
Sales	2	
Marketing	1	50%
Support	1	

CDB Resource Plan		
PDB	**Shares**	**Utilization Limit**
Sales	4	
Marketing	1	50%
Support	1	75%

Database Resource Plan		
Consumer Group	**Shares**	**Utilization Limit**
Critical	4	
Batch	1	
ETL	1	
AdHoc	1	50%

ORACLE

Monitoring and Tuning IORM



Managing Exadata Flash

ORACLE

Exadata Smart Flash Cache

- Exadata Flash provides impressive capacity and bandwidth*
  - 45 TB space
  - 100 GB/s of scan throughput
  - ~2.5 M IOPS
- Use Exadata Flash to
  - Cache frequently-used OLTP data
  - Accelerate scans, using excess flash resources
  - Accelerate log writes

* For a X4-2/8 full rack

ORACLE

Exadata Smart Flash Cache

Why Exadata?

- Other market offerings
  - All-flash storage: expensive
  - Disks with flash to accelerate writes and cache recently-used data: doesn’t support mixed workloads
- Only Exadata guarantees that flash stores frequently-accessed OLTP data
  - Scans do not evict OLTP data from the cache
  - Only Exadata can identify OLTP from scan IOs!
  - All-flash storage guarantees this too, but at many times the price!

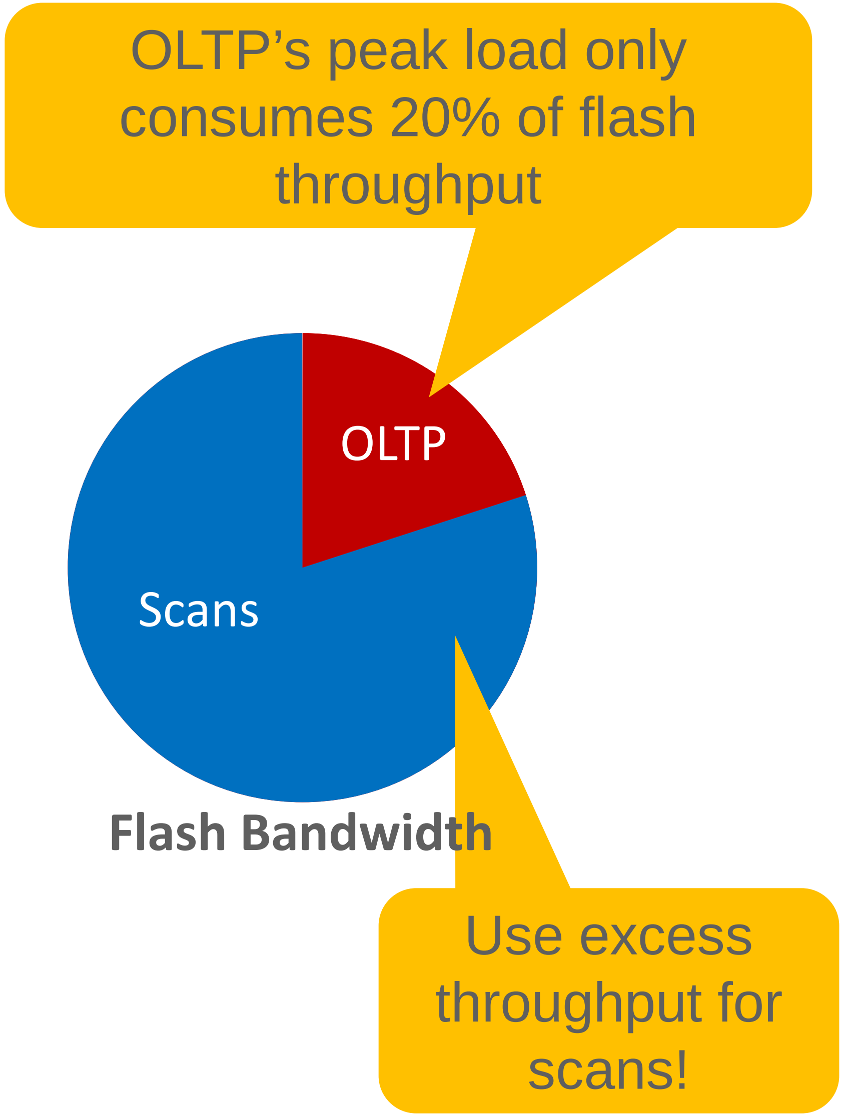
ORACLE

Exadata Smart Flash Cache

Flash Bandwidth

How can you capitalize on 100 GB/s of flash bandwidth?

- Use up to 20% for OLTP workloads
  - Maximum sustainable rate by OLTP
- Use the rest for scans!
  - Pre 11.2.2.3, mark tables with KEEP
  - 11.2.2.3+, scans automatically use flash if there's space
- Scans use both flash and disks for maximum throughput
  - Using flash increases scan throughput by up to 5x!



ORACLE

Exadata Smart Flash Cache

Space Contention

Flash Logs use just 0.02% of total space.
Don't disable!

Disabling flash cache may cause a heavier disk I/O load

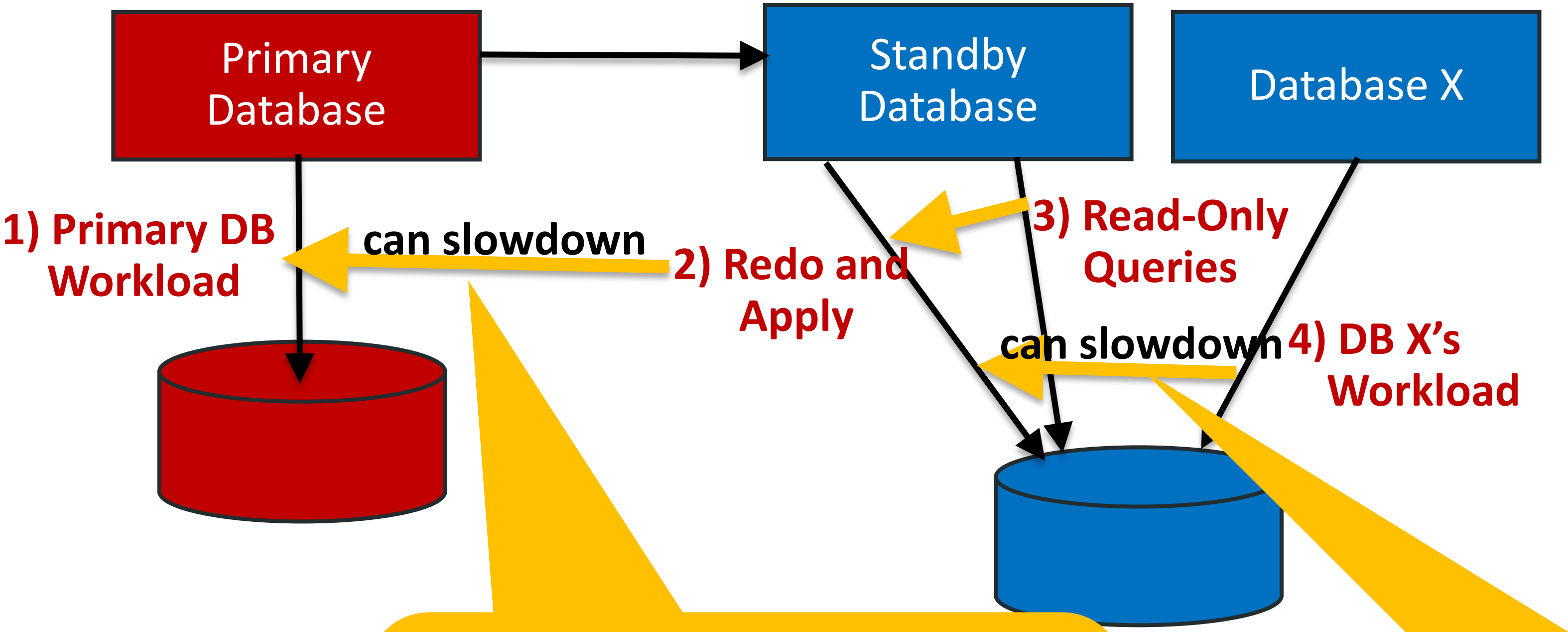
Inter-Database IORM Plan		
	Use Flash Log?	Use Flash Cache?
DB #1	✓	✓
DB #2	✓	✓
DB #3	✓	✓

For most workloads, Exadata already uses flash space optimally.
To customize, use the Inter-Database IORM Plan

ORACLE

Managing Standby Databases

Resource Manager for Standbys



Issue #1: If synchronous redo transport is enabled, the Primary DB will suffer from slow redo writes on the Standby

Issue #2: Heavy read-only queries and other database workloads can slow the Standby's redo and apply

Resource Manager for Standbys

Prioritize Redo and Apply over Read-Only Queries

- Enable CPU and I/O resource management
  - Set a database resource plan on Standby
  - Use the primary's resource plan or DEFAULT_PLAN (best practice - use the same resource plan on primary and standby in case a switch-over occurs!)
  - Enable I/O Resource Manager (alter iormplan set objective = auto)
- On physical standbys, all resource manager objects are replicated on the standby
  - New resource plans cannot be configured on the standby
  - The primary and standby can use different resource plans
  - See MOS note 1930540.1 for known issues for consumer group mappings
- On logical standbys, all resource manager objects must be recreated

ORACLE

Resource Manager for Standbys

Guarantee Resources for the Standby

- Enable Instance Caging for all database instances on the Standby's server
  - CPU_COUNT must be sufficiently high to handle the load in the event of a switch-over
- Enable inter-database IORM for all databases on the Standby's storage cells
  - Create the allocation using standby's DB_UNIQUE_NAME

ORACLE

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!

ORACLE

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**

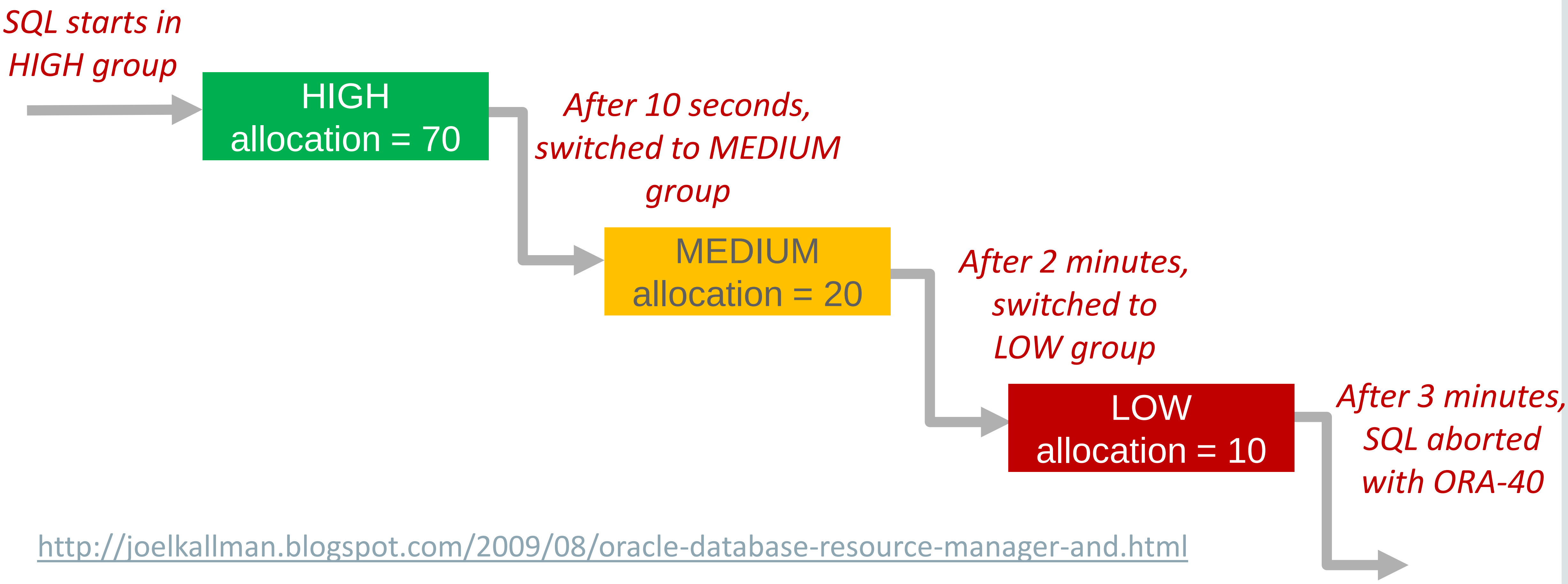
Manage runaway queries:

- ✓ 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



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

For More Information

For More Information

- Master MOS document 1339769.1
  - Recommended bug fixes
  - Step-by-step configuration for common scenarios
  - Monitoring and tuning scripts
- Best Practices for Multi-Tenant Consolidation on Exadata

  http://www.oracle.com/technetwork/database/availability/maa-consolidation-2186395.pdf
- Best Practices for Database Consolidation on Exadata

  http://www.oracle.com/technetwork/database/features/availability/exadata-consolidation-522500.pdf
- Instance Caging white paper

  http://www.oracle.com/technetwork/database/focus-areas/performance/instance-caging-wp-166854.pdf
- Resource Manager white paper

  http://www.oracle.com/technetwork/database/focus-areas/performance/resource-manager-twp-133705.pdf

ORACLE

Hardware and Software

Engineered to Work Together

ORACLE