ORACLE

Resource Manager Overview

Sue Lee, Director of Development

Agenda

- Resource Manager Use Cases
  - Consolidation
  - Mixed Workloads
- Managing Resources
  - CPU
  - Disk I/O
  - Runaway Queries
  - Parallel Execution
  - Idle Time
- Other Considerations
  - Memory
  - Sessions

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.

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

Consolidation Challenges

- Database users apprehensive about consolidation
  - Users want 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 runaway query
- DBAs want to control resource usage
  - Hosted environments – “get what you pay for”
  - Limit resource utilization for consistent performance

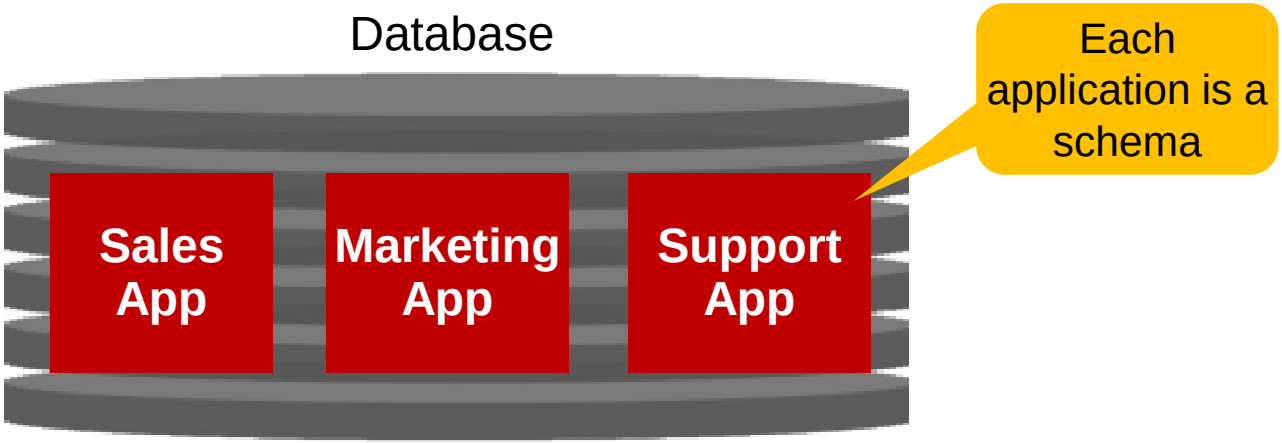
Consolidation Methodologies

- Schema Consolidation
  - Multiple applications share a database
- Server Consolidation
  - Multiple databases share a server
- Multi-Tenancy **New in 12c**
  - Multiple Pluggable Databases share a Container

No “right” approach!

Each approach has its pros and cons!

Schema Consolidation



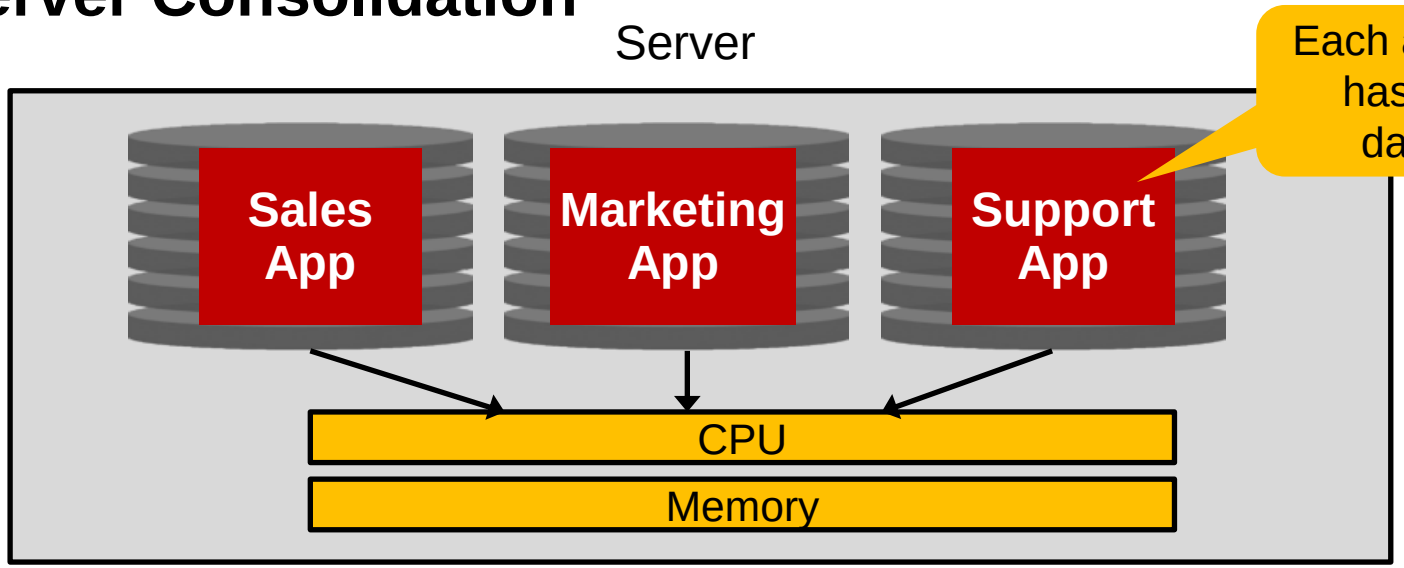
Multiple applications share a database

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

But…

- Object name collisions due to shared dictionary
- Often requires application-level changes!

Server Consolidation



Multiple databases share a server

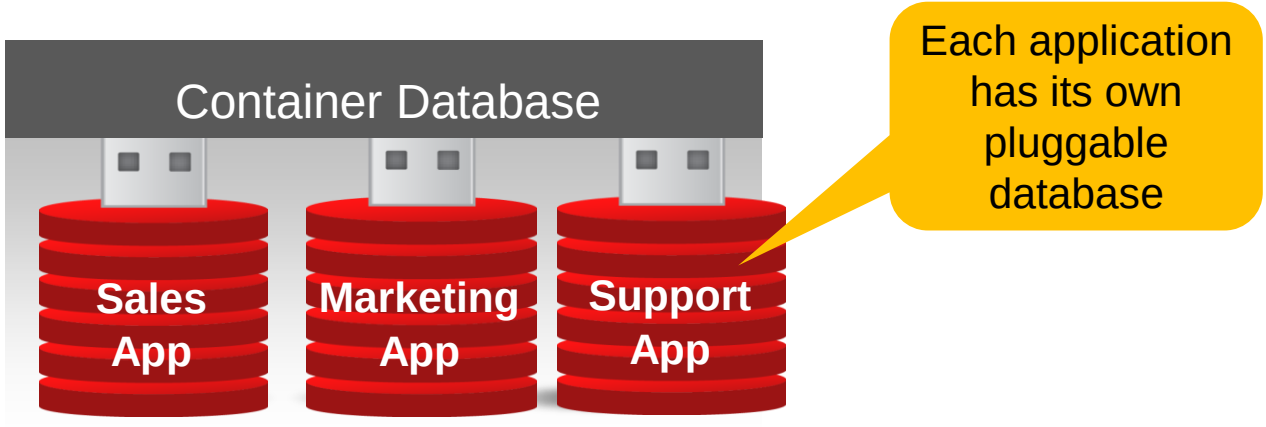
- ✓ Application isolation
- ✓ Each application is independently maintained and upgraded

But…

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

Pluggable Databases

New in 12c!



Container Database

- ✓ Shared backgrounds and SGA
- ✓ One database to administer

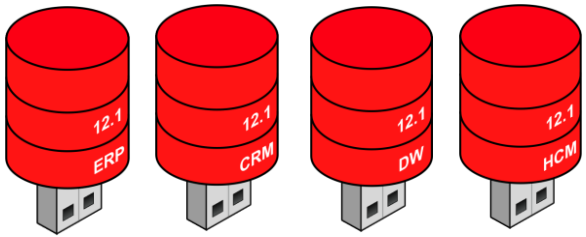
Pluggable Database

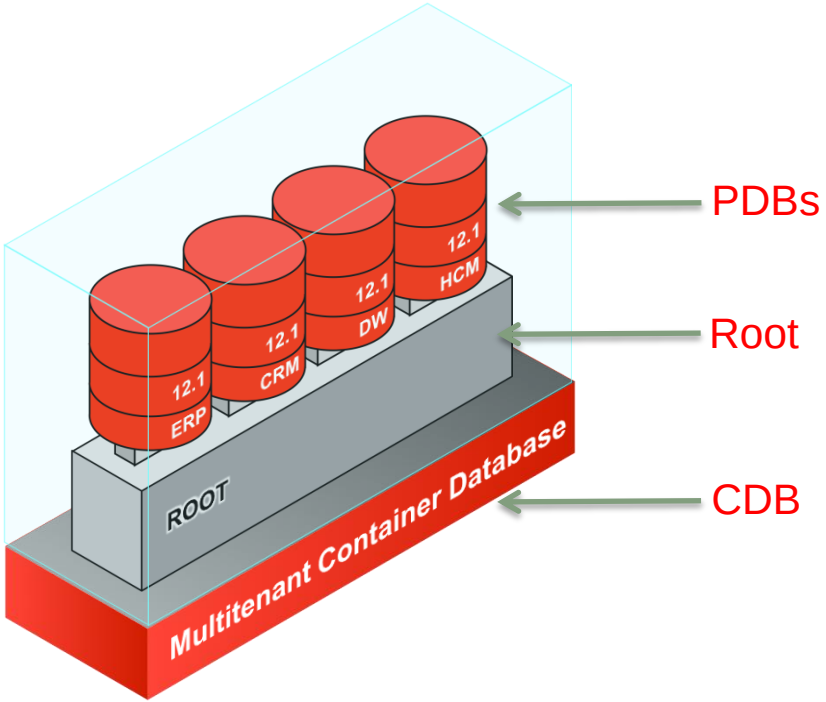
- ✓ Each application has its own name space
- ✓ Easy to plug and unplug from the Container Database
- ✓ No application changes required

Multitenant Architecture

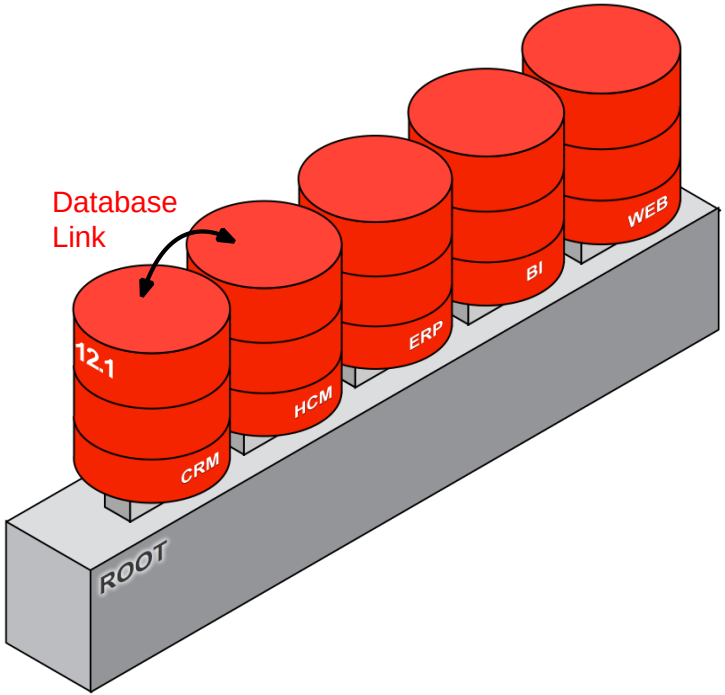
Components of a Multitenant Container Database (CDB)

Pluggable Databases (PDBs)





Multitenant Architecture



- Multitenant architecture can currently support up to 252 PDBs
- A PDB feels and operates identically to a non-CDB
- You cannot tell, from the viewpoint of a connected client, if you're using a PDB or a non-CDB

Using Resource Manager for Consolidation

- Control resource contention and cap resource usage using Resource Manager
- Resource Manager supports all types of consolidations
  - Schema consolidation
  - Server consolidation
  - Pluggable Databases
- Resource Manager manages
  - CPU
  - Exadata disk I/O
  - Parallel execution
  - Runaway queries
  - And more…

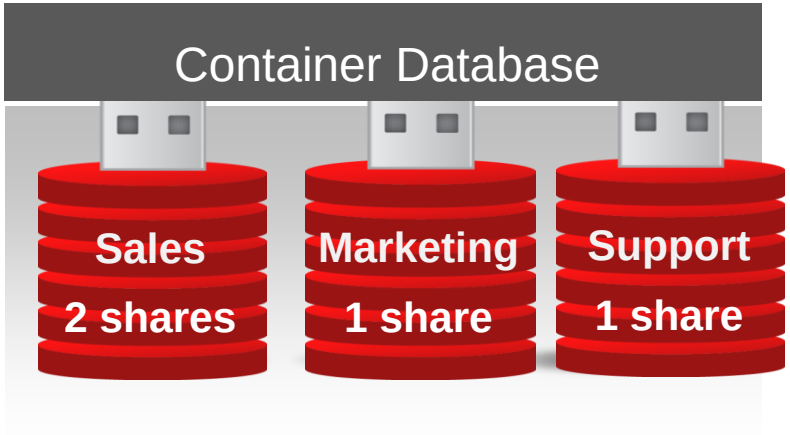
MANAGING CPU

Managing CPU

- Multiple tools for managing CPU
  - Database Resource Plan: for workloads within a database
  - CDB Resource Plan: for PDBs within a CDB
  - Instance Caging: for database instances within a server
- All of these tools work in a very similar way
- All of these tools can be used together or separately

Managing CPU

A CDB Resource Plan uses "shares" to specify how CPU is distributed between PDBs



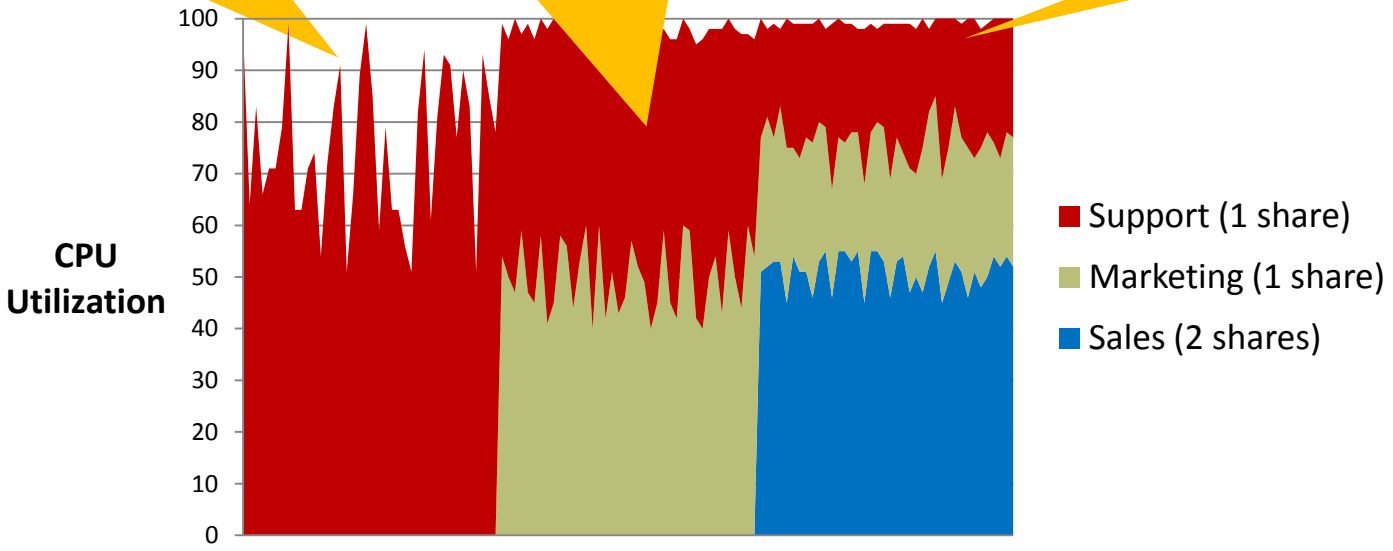
CDB Resource Plan			
Pluggable Database	**Shares**	**Guaranteed CPU**	**Maximum CPU**
Sales	2	2/4 = 50%	100%
Marketing	1	1/4 = 25%	100%
Support	1	1/4 = 25%	100%

Managing CPU

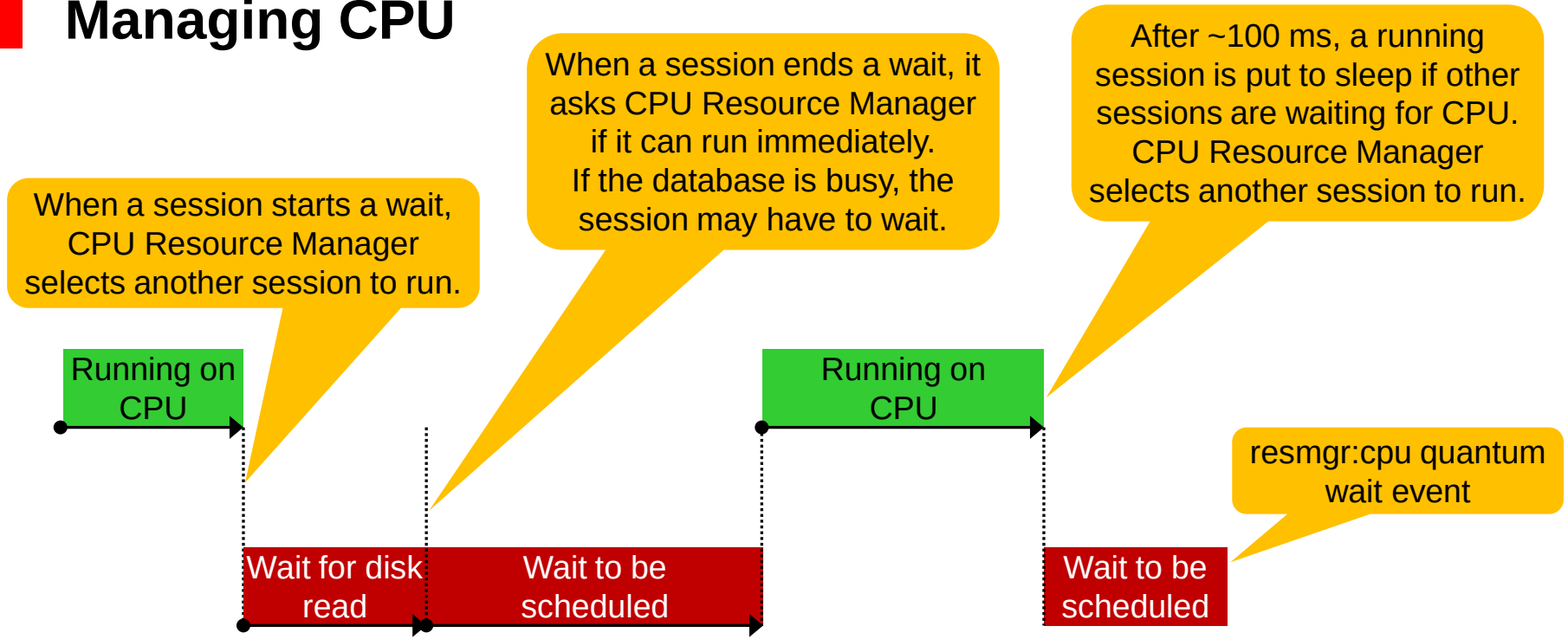
When SUPPORT is the only active workload, it gets as much CPU as it needs, even with just 1 share.

When SUPPORT and MARKETING are both active, they get the same amount of CPU since they have an equal number of shares.

When all workloads are active, they share the CPU based on their ratio of shares.



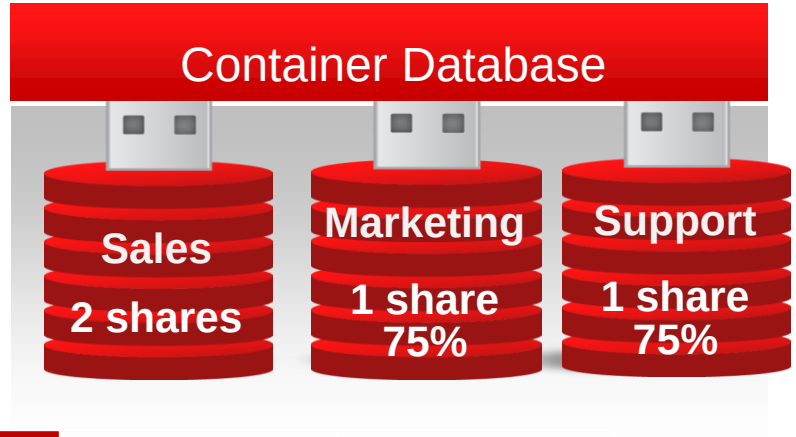
Managing CPU



Life of a session with CPU Resource Manager.
CPU Resource Manager uses fine-grained scheduling, just like an O/S.

Managing CPU

A CDB Resource Plan uses "utilization limits" to enforce a hard limit on the CPU usage for a PDB.

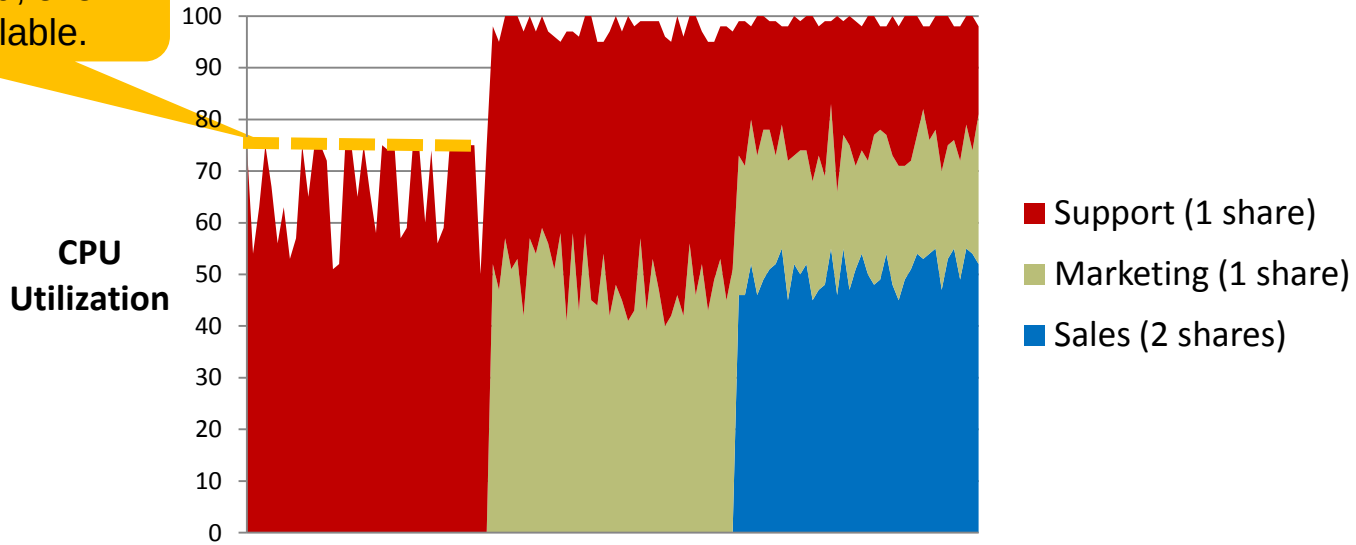


CDB Resource Plan

Pluggable Database	Shares	Utilization Limit	Guaranteed CPU	Maximum CPU
Sales	2		2/4 = 50%	100%
Marketing	1	75%	1/4 = 25%	75%
Support	1	75%	1/4 = 25%	75%

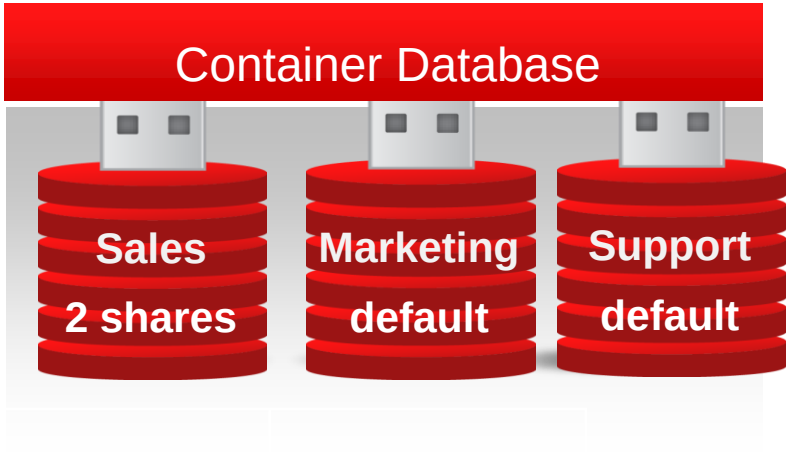
Managing CPU

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

Managing CPU



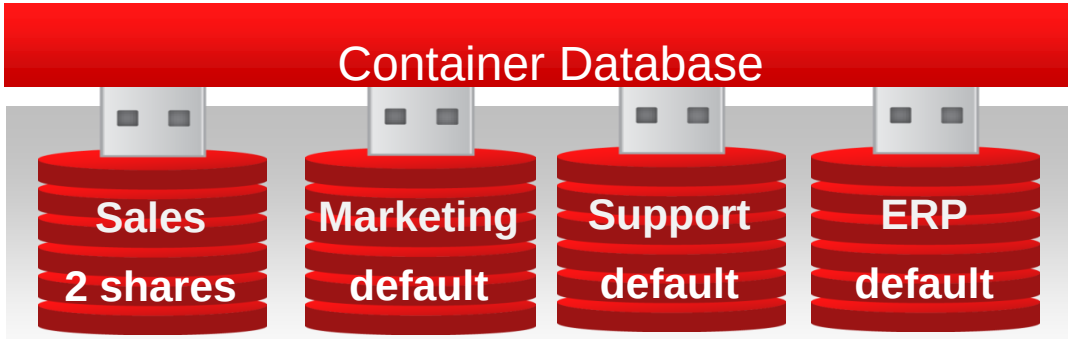
Configure a “default directive”: the default shares and utilization limit for PDBs.

CDB Resource Plan

Pluggable Database	Shares	Utilization Limit	Guaranteed CPU	Maximum CPU
(Default directive)	**1**	**75%**		
Sales	2		2/4 = 50%	100%
Marketing	**default (1)**	**default (75%)**	1/4 = 25%	75%
Support	**default (1)**	**default (75%)**	1/4 = 25%	75%

Managing CPU

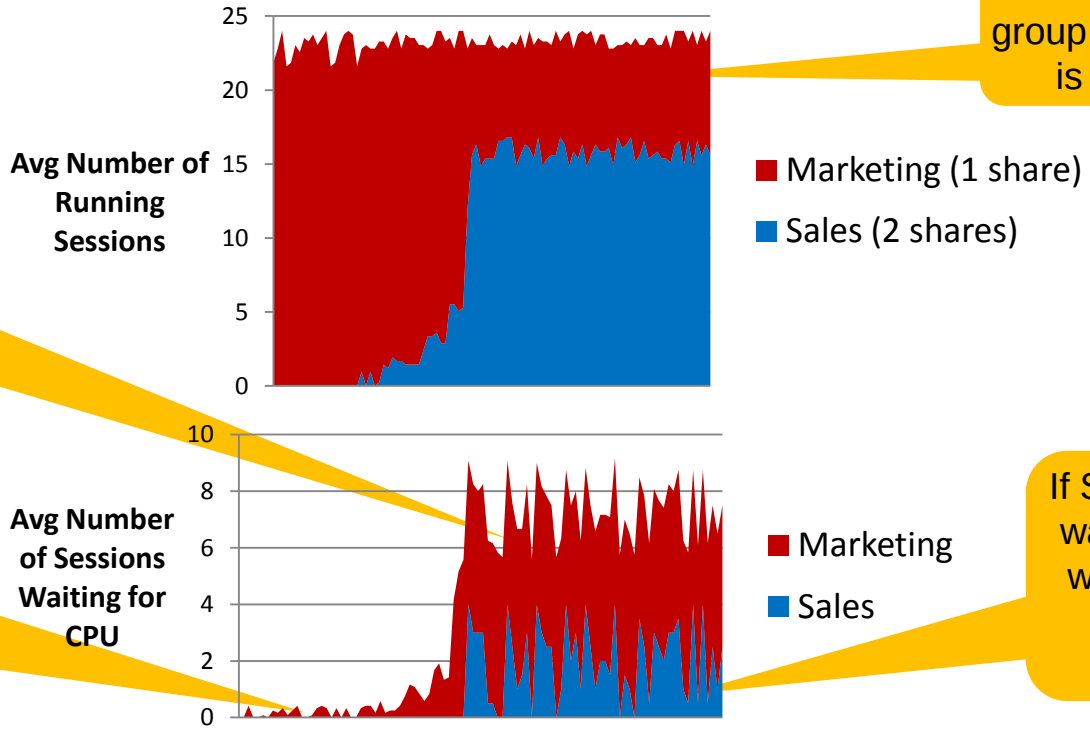
With a default directive, you don't need to modify the resource plan when a PDB is added or removed!



CDB Resource Plan				
Pluggable Database	**Shares**	**Utilization Limit**	**Guaranteed CPU**	**Maximum CPU**
(Default directive)	1	75%		
Sales	2		2/5 = 40%	100%
Marketing	default (1)	default (75%)	1/5 = 20%	75%
Support	default (1)	default (75%)	1/5 = 20%	75%
ERP	**default (1)**	**default (75%)**	**1/5 = 20%**	**75%**

Monitoring and Tuning CPU Resource Manager

CPU usage by consumer group shows how the CPU is being distributed



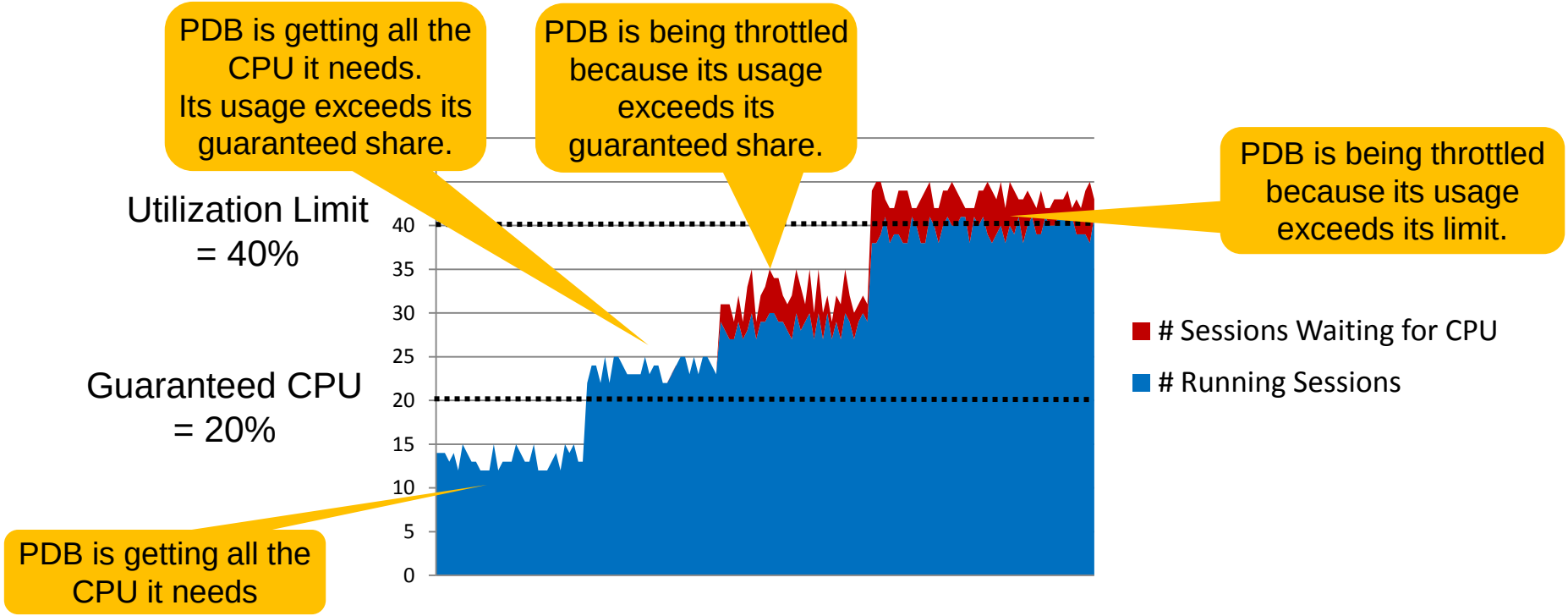
If *any* sessions are waiting for CPU, the database instance would benefit from a larger server.

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

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

Monitor using v$rsrcmgrmetric_history or Enterprise Manager

Monitoring and Tuning a PDB



See how the PDB is affected by its resource plan settings.
Monitor using v$rsrcmgrmetric_history.

Monitoring and Tuning CPU Resource Manager

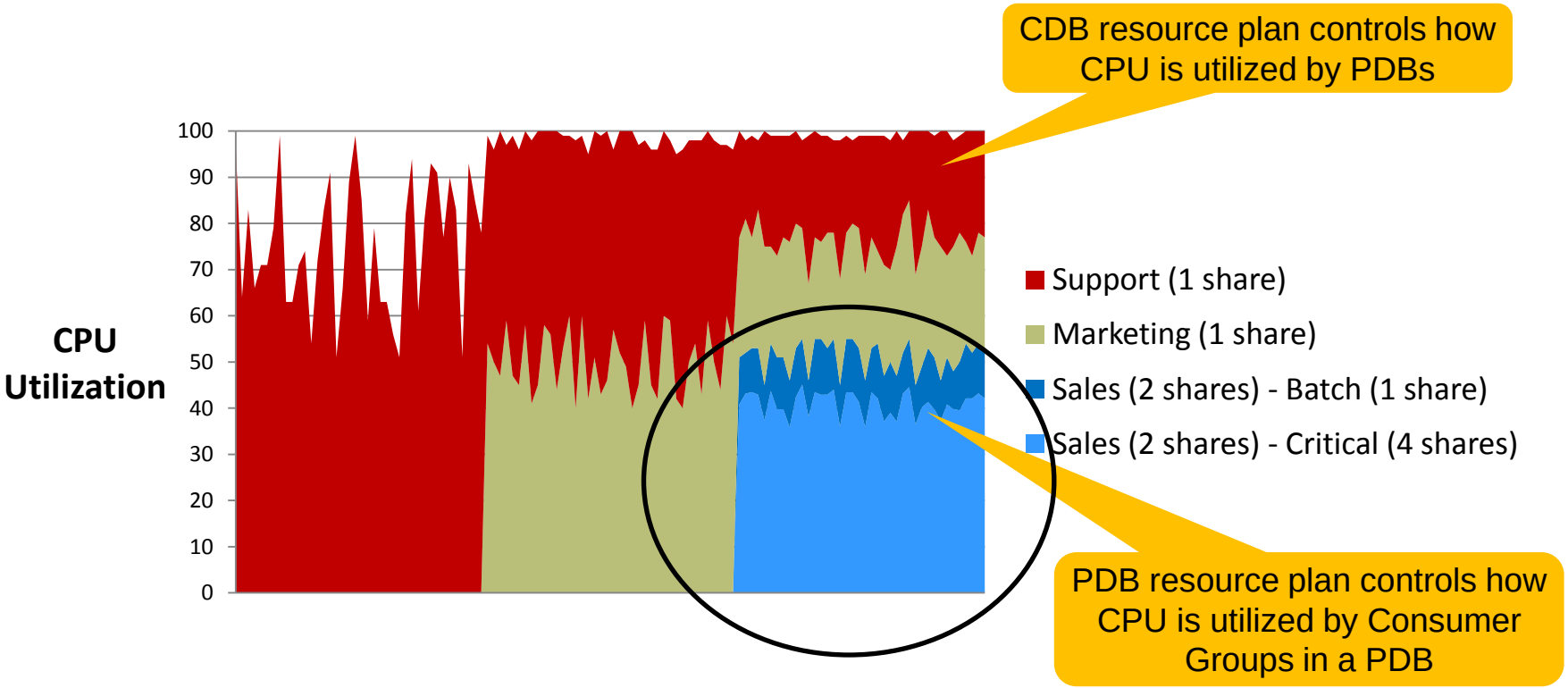
- Configuring a resource plan is an iterative process
  - Create a resource plan
  - Monitor application performance and Resource Manager metrics
  - Adjust resource allocations and re-monitor
- Is it bad for sessions to wait for CPU?
  - Equivalent to processes waiting on the O/S run queue
  - Lots of waiting sessions means server is overloaded!
  - If performance is unsatisfactory, increase resource allocation for that Consumer Group
  - With Resource Manager, critical background processes and O/S are not starved

Managing Workloads in a Database or PDB

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

Database Resource Plans manage Consumer Groups using the same concepts!

How Do CDB and PDB Resource Plans Work Together?



Migrating to PDBs

PDB Resource Plan Restrictions

Non-PDB Resource Plan	PDB Resource Plan
Multi-level resource plans	Single-level resource plans
Up to 32 consumer groups	Up to 8 consumer groups
Subplans	No subplans

Best practice: keep all resource plans simple!

- Most users misconfigure multi-level plans
- PDB is typically not used to consolidate multiple workloads
- Less need for many consumer groups and subplans

Migrating to PDBs

- Automatic conversion for 11g Resource Plans during plug-in
  - Plans with multiple levels and subplans are flattened to a single level
  - Plans with >8 consumer groups use the top 8 consumer groups
  - Original plan is saved and available for viewing

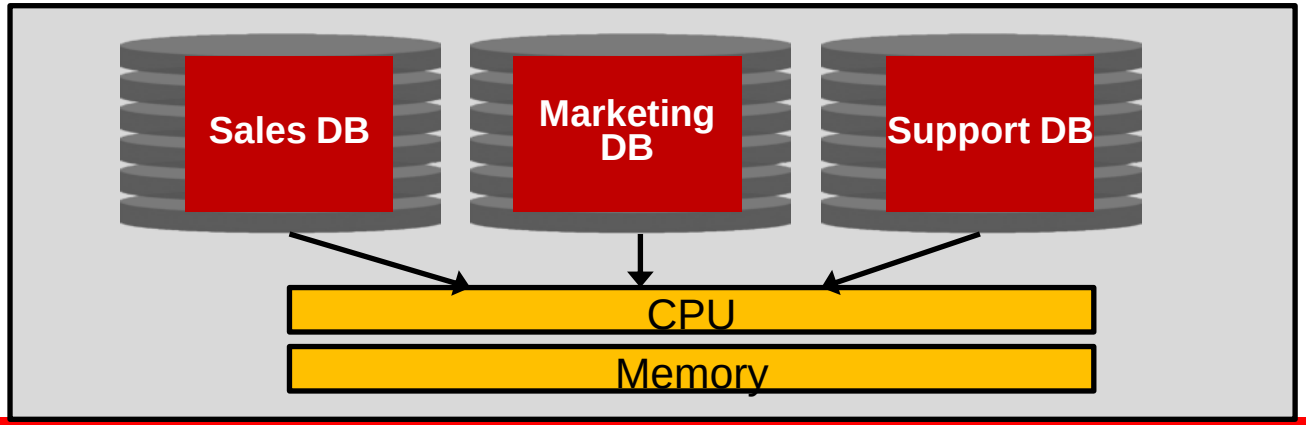
11g Database Resource Plan			
Consumer Group	Level 1 Allocation	Level 2 Allocation	Level 3 Allocation
SYS	50		
GOLD		50	
SILVER			80
BRONZE			20

is equivalent to

PDB Resource Plan	
Consumer Group	Shares
SYS	50
GOLD	25
SILVER	20
BRONZE	5

Managing CPU with Instance Caging

- Problem: CPU contention between database instances sharing a server
  - Heavy workload from one database instance affects other instances' performance
  - Runaway workload from one database instance can destabilize server
- Solution: Instance Caging
  - Limit or "cage" the amount of CPU that a database instance can use at any time



Manage CPU with Instance Caging

Configure Instance Caging with just 2 steps

1. Set “cpu_count” parameter to the maximum number of CPUs the instance can use at any time

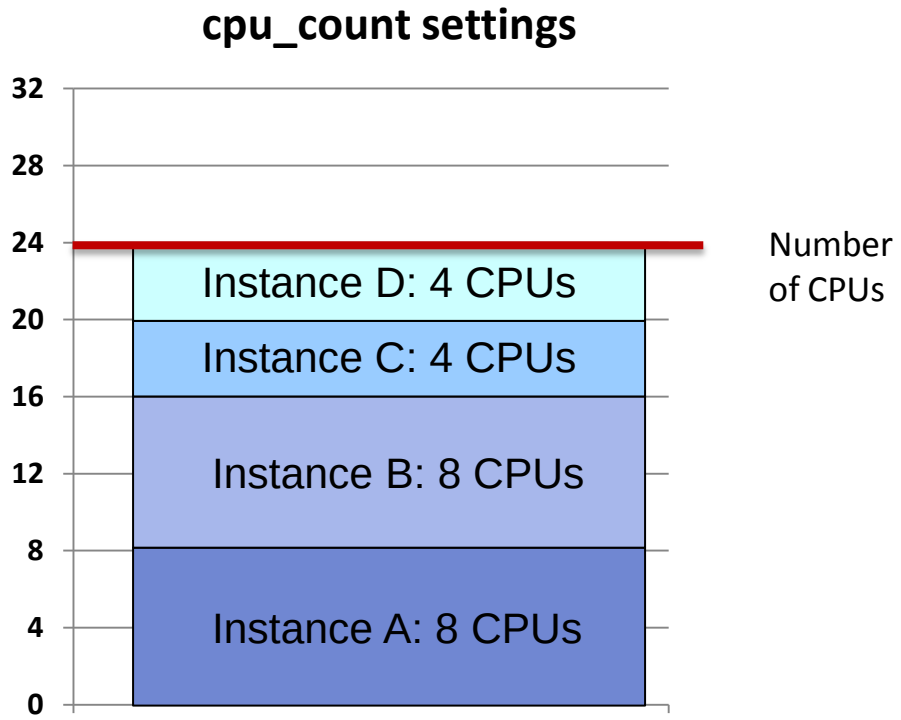
   ```
   alter system set cpu_count = 4;
   ```

2. Set “resource_manager_plan” parameter to enable CPU Resource Manager

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

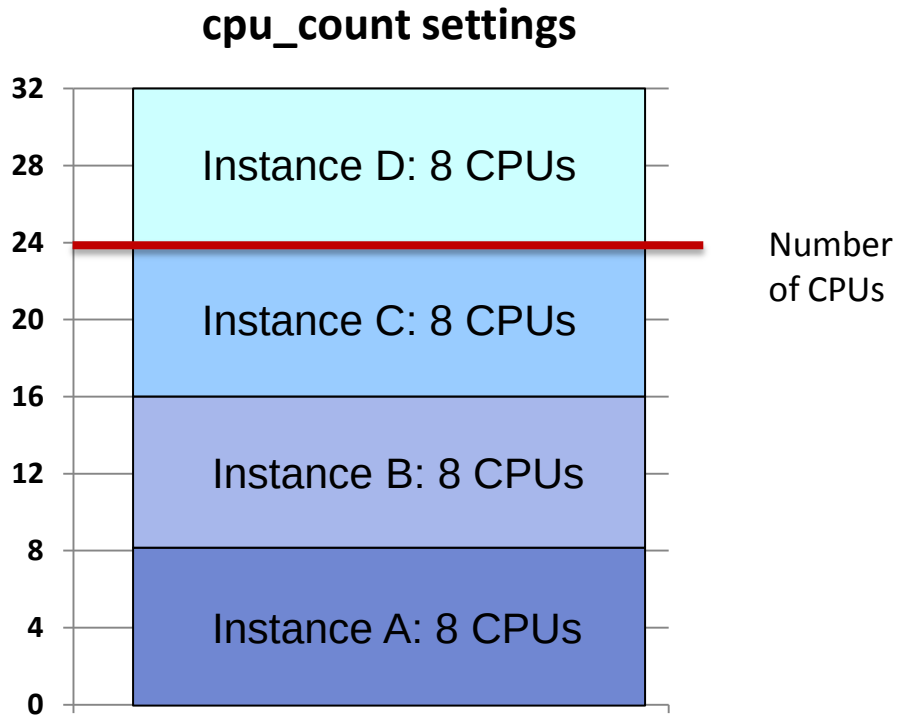
Setting cpu_count: Partition Approach

- Partition CPUs among the database instances
  - sum(cpu_counts) <= # cpu threads
- Partitioning provides maximum isolation
  - No CPU contention between instances
  - But if one instance is idle, its CPU allocation is unused
- Best for performance-critical databases

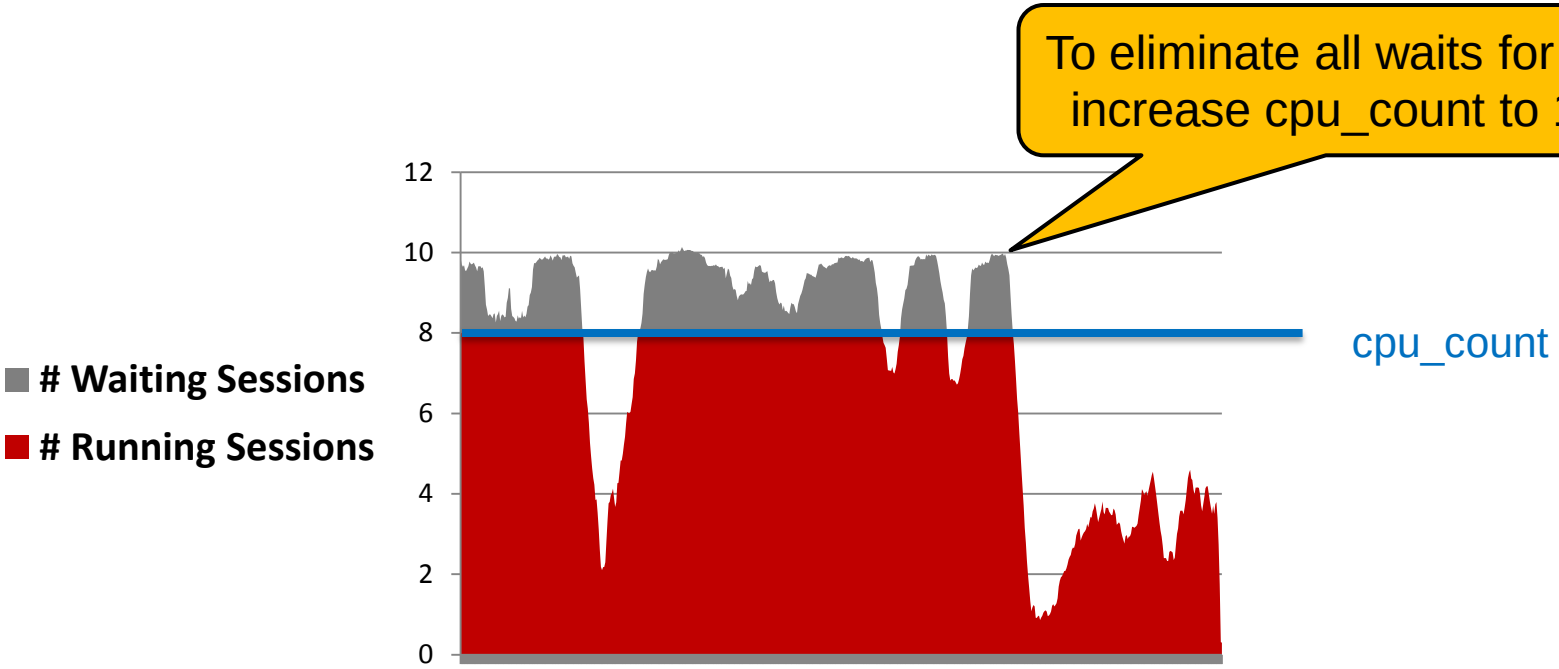


Setting cpu_count: Over-Subscribe Approach

- Over-subscribe the CPUs among the database instances
  - sum(cpu_counts) <= 3 x # cpu threads
  - Monitor CPU utilization to see if there's room!
- Over-subscribing provides efficient CPU utilization
  - Some contention for CPU if databases are sufficiently loaded
  - Contention is controlled, so system is still stable
- Best for non-critical databases



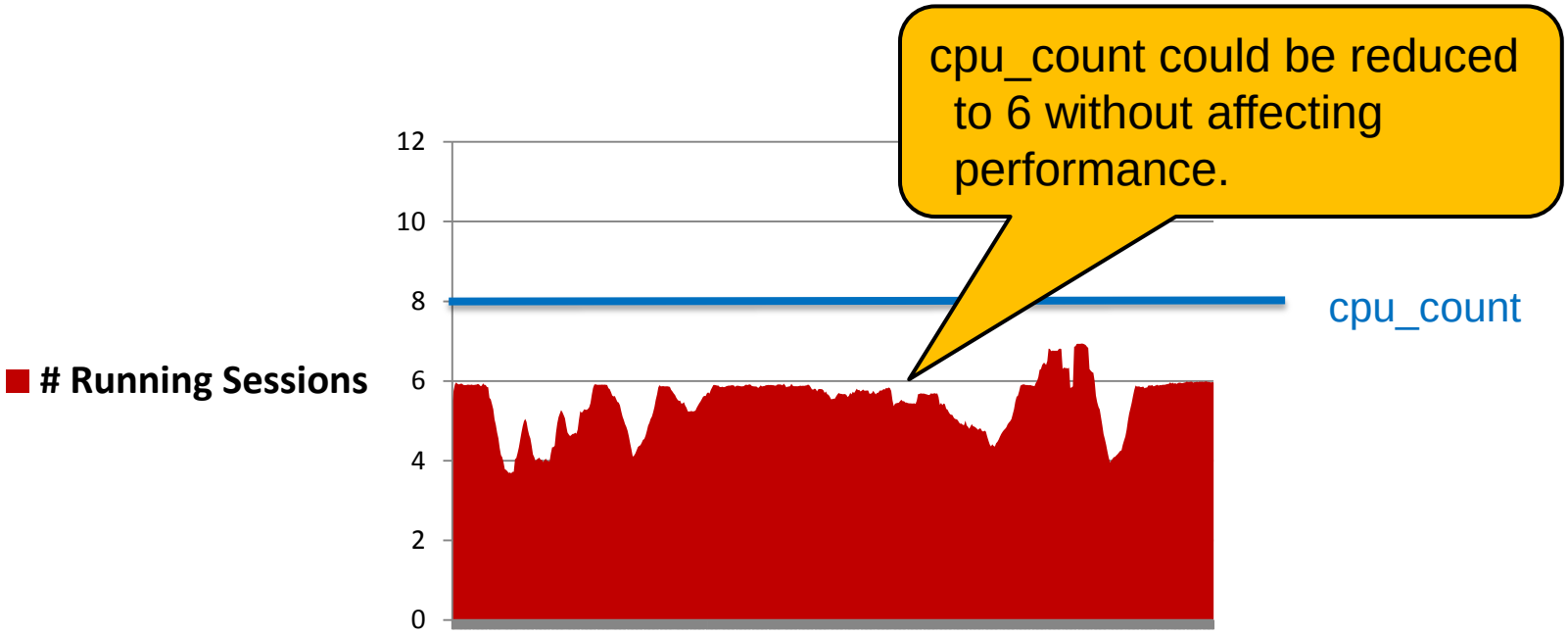
Tune, Based on Instance's CPU Usage



This instance would benefit from a higher cpu_count.

Monitor with v$rsrcmgrmetric_history. See MOS note 1338988.1 for details.

Tune, Based on Instance's CPU Usage



This instance is not using all of its cpu_count allocation

Instance Caging vs O/S Tools

Instance Caging

- Supported on all platforms
- Supported on Exadata
- Resource Isolation
  - No dedicated CPUs
- Over-subscribe support
- Shared infrastructure
  - One CSS per node
  - One ASM server per node

O/S Partitions

- Platform-specific solution
- Resource Isolation
  - Dedicated CPUs
- Over-subscribe sometimes supported
  - AIX micro-partitions
  - Solaris zones with FSS
- Non-shared infrastructure
  - One CSS per partition
  - One ASM server per partition (in 12.1, use FlexASM to access remote ASM servers)

Pros and cons to both approaches – no right answer!

Instance Caging

Keys to Success

- 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
- Keep cpu_count >= 2
- Avoid large changes to cpu_count
- Beware of over-subscribing on SPARC T-Series processors

Migrating to PDBs

Translating Instance Caging to CDB Resource Plans

- Pre-12c: databases managed with Instance Caging
- 12c: PDBs managed with CDB Resource Plans
- To convert from Instance Caging to CDB Resource plans
  - Use cpu_count as PDB's "share"
  - Use cpu_count / num_cpus as PDB's "utilization limit" (optional!)

Database Name	cpu_count
SALES	16
MARKETING	8
SUPPORT	8

is equivalent to

PDB Name	Shares	Utilization Limit
SYS	16	50%
GOLD	8	25%
SILVER	8	25%

MANAGING DISK I/O

IORM for Multiple Databases

- Manage I/Os between databases
- Important for database consolidation
- Configure an inter-database IORM plan

Inter-Database IORM Plan	
Database	Shares
Sales	4
Marketing	1
Support	1
Other	1

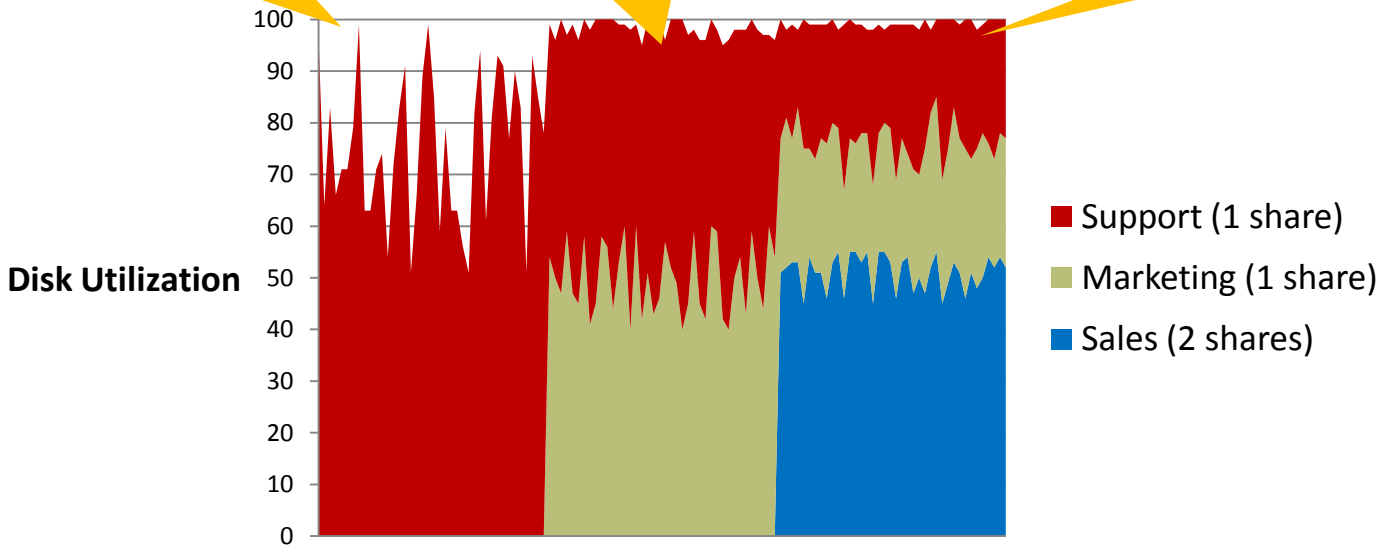
Shares for “Other” are used for all other databases

IORM for Multiple Databases

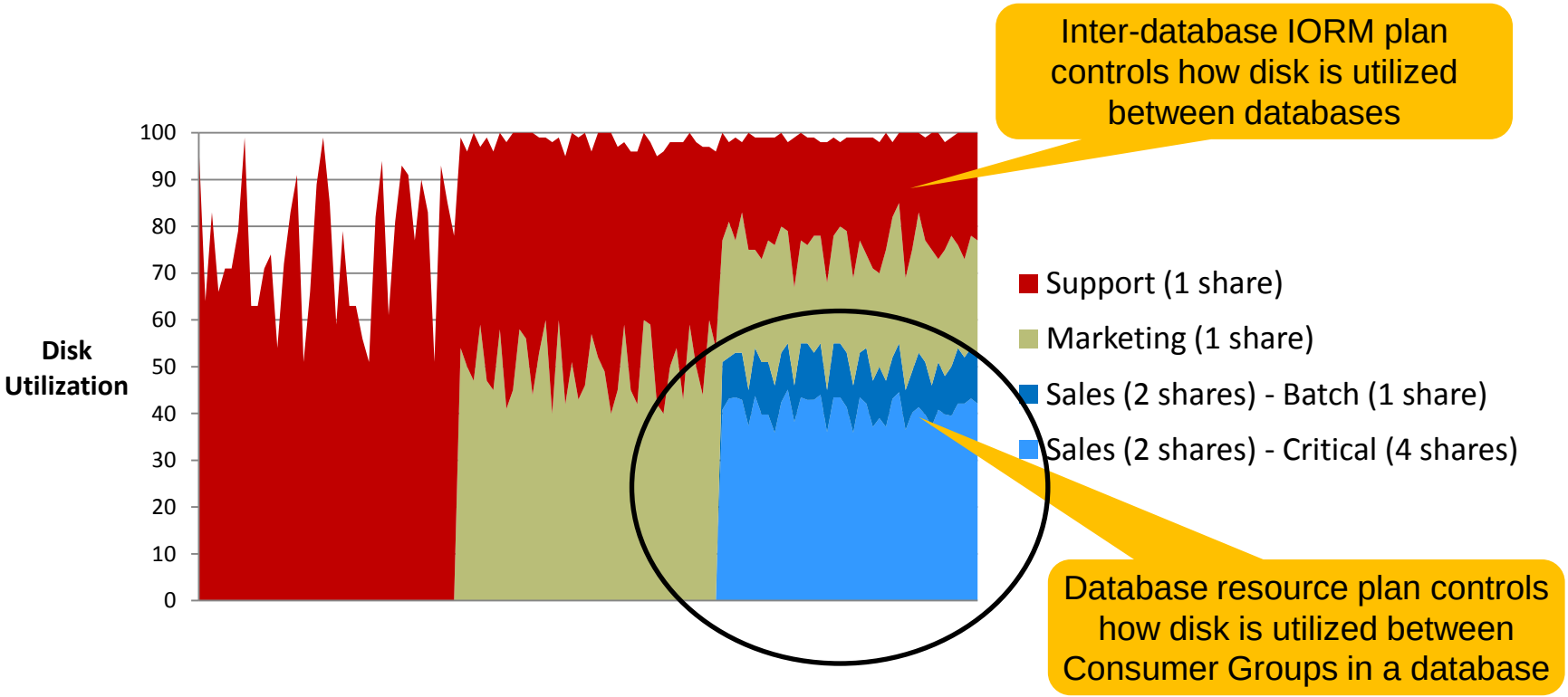
When SUPPORT is the only active database, it issues as many I/Os as it needs to, even with just 1 share.

When both SUPPORT and MARKETING are active, they share the disk since they have an equal number of shares.

When all databases are active, they share the disk, based on their ratio of shares.



Inter and Intra Database IORM



IORM Utilization Limits

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

Inter-Database IORM Plan		
Database	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%**

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!

Manage Runaway Queries

Define runaway query thresholds:

- ✓ Estimated execution time
- ✓ Execution 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**

Manage Runaway Queries

For Tactical consumer group, runaway is:
100+ logical or physical IOs

Switch to "Low Priority" consumer group!

For Reports consumer group, runaway is:
60 minutes elapsed time

Abort query!

For Ad-Hoc consumer group, runaway is:
24+ hour estimated execution time

Don't execute!

Monitor Runaway Queries in 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

Monitor Runaway Queries in v$sql_monitor

Pre 12.1, if Resource Manager canceled the SQL or killed the session, you can monitor the SQL using these columns:

Column	Description
RM_CONSUMER_GROUP	Current consumer group name
ERROR_NUMBER	Error number encountered in case a SQL fails to execute successfully. Resource Manager session kill error numbers are: OER(41), OER(56721), OER(56723) Resource Manager cancel SQL error numbers are: OER(40), OER(56720), OER(56722)
ERROR_MESSAGE	Detailed error message corresponding to ERROR_NUMBER.

MANAGING MEMORY

Manage Memory

- Avoid excessive memory usage
  - Swapping
  - Poor performance
  - Instance eviction
- Ensure memory for kernel, stack space, other applications
- Memory controls
  - sga_target
  - pga_aggregate_target
  - pga_aggregate_limit **New in 12c**

Manage PGA Usage

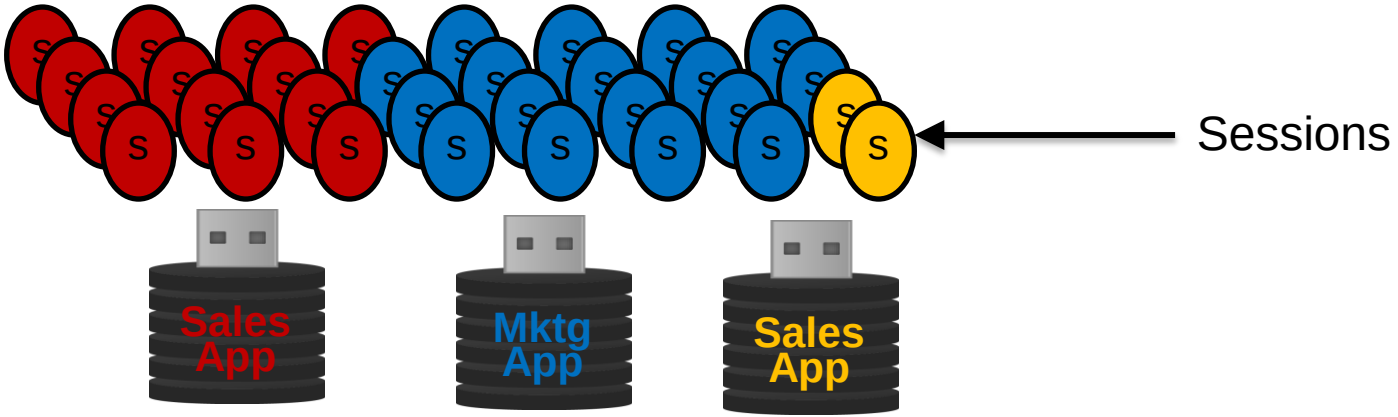
- pga_aggregate_target
  - Only controls “tunable” memory allocations
  - “tunable” means that the operation can opt to use PGA or temp space
  - E.g. hash joins, sorts, etc.
  - Actual PGA usage is often much higher (3x) since operations for “untunable” memory do not heed this parameter
  - Particularly problematic with parallel queries with high DOPs, badly behaved PL/SQL

Manage PGA Usage

- pga_aggregate_limit **New in 12c**
  - Hard PGA memory limit
- When actual PGA usage exceeds PGA_AGGREGATE_LIMIT
  - Calls aborted for sessions using the most memory
  - Memory consumption for parallel operations tracked as a unit
  - SYS and fatal background processes exempted

MANAGING SESSIONS

Manage Sessions



- Sessions are shared between all PDBs
- What happens when some PDBs take all the sessions?
- “sessions” parameter
  - At the root level, maximum number of sessions for the Container Database
  - At the PDB level, maximum number of sessions that a PDB can use at any time

For More Information

- 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
- Best Practices for Database Consolidation on Exadata
  http://www.oracle.com/technetwork/database/features/availability/exadata-consolidation-522500.pdf
- Master MOS document 1339769.1
  - Recommended bug fixes
  - Step-by-step configuration for common scenarios
  - Monitoring and tuning scripts

Hardware and Software

ORACLE

Engineered to Work Together