Oracle Resource Manager
资源管理器

甲骨文数据库与云系列公益讲座之五十六期

Nick Peng 彭立诚

Senior Solution Engineer

Data Management SEHUB

Safe Harbor

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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

数据库整合形态的发展

-数据库整合即多个数据库共享同一硬件资源，提高利用率

- 独立数据库与整合数据库一直在发展
- 云时代下数据库形态都是在整合资源
- 不要过度隔离，限制虚拟蔓延
- 如何阻止干扰：使用资源管理器

“虚拟化+容器”是当前最佳数据库隔离方式，资源管理变得不可或缺!

为什么要整合数据库?

会存在所需要的场景:

- **降低成本**
- **简单**
- **安全**

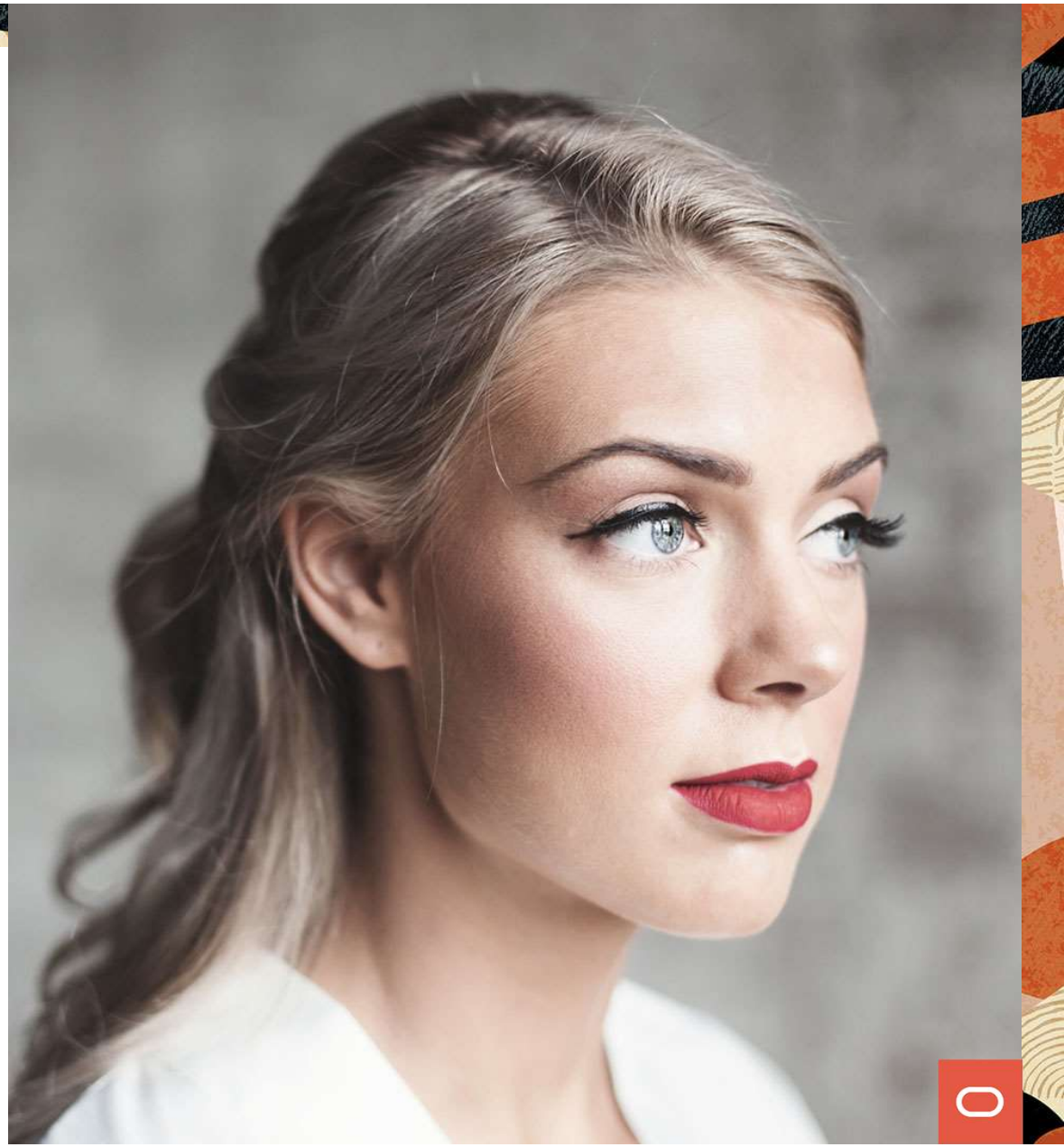
整合**成本**更低，因为相同的数据库在更少的硬件上运行。这也适用于云部署，因为公司仍在为使用底层硬件付费。在某些云环境中可能更难跟踪和可视化，但更多的硬件意味着更高的成本。无论您的数据库是部署在本地数据中心还是云数据中心的硬件上，使用这些系统都会增加成本。

数据库“整合”和“隔离”我们都需要！资源分配与管理变得非常重要

Agenda

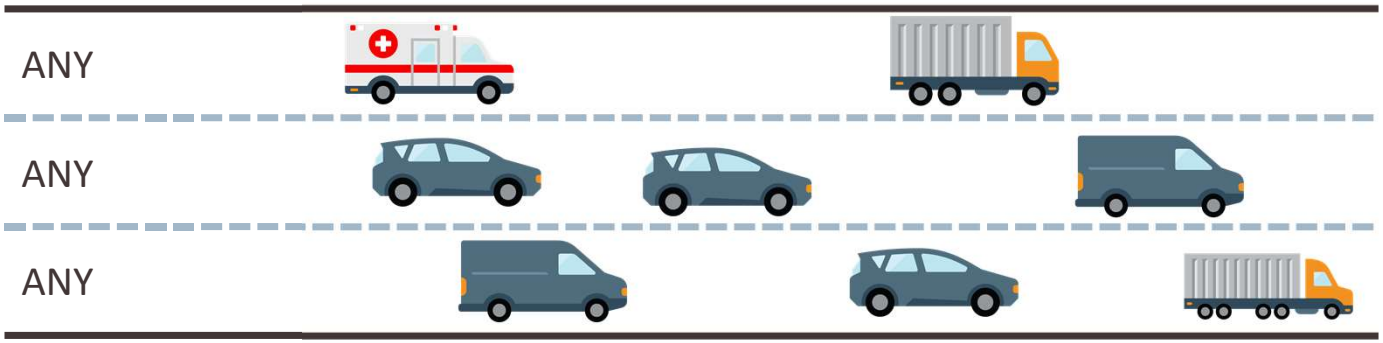
- **Resource Management 概念**
- **资源控制策略**
- **I/O 资源控制**
- **资源监控**
- **案例演示**

Resource Management
– 概念



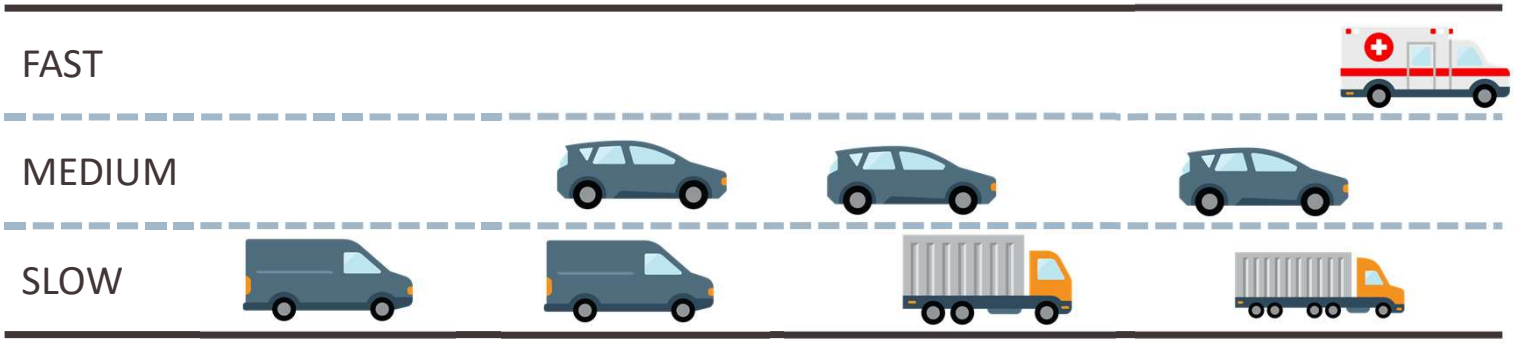
Resource Management – 概念

CPU, Memory, Processes, Storage Network, Flash Storage, I/O



如果没有资源管理，任何数据库都可能减慢其他数据库的运行速度

RM允许完全控制哪个数据库运行得更快。



资源管理的主要场景在于数据库整合，其它友商没有具备oracle 完善的资源控制功能

方法论

数据库整合方法

1. CPU

- 启用数据库资源管理器默认计划
- 为每个数据库设置CPU_COUNT
- 在19c及以上版本中使用CPU_MIN_COUNT

2. I/O

- Set Exadata IORM Objective = “auto”
- 从CPU设置继承I/O资源比率

3. Memory

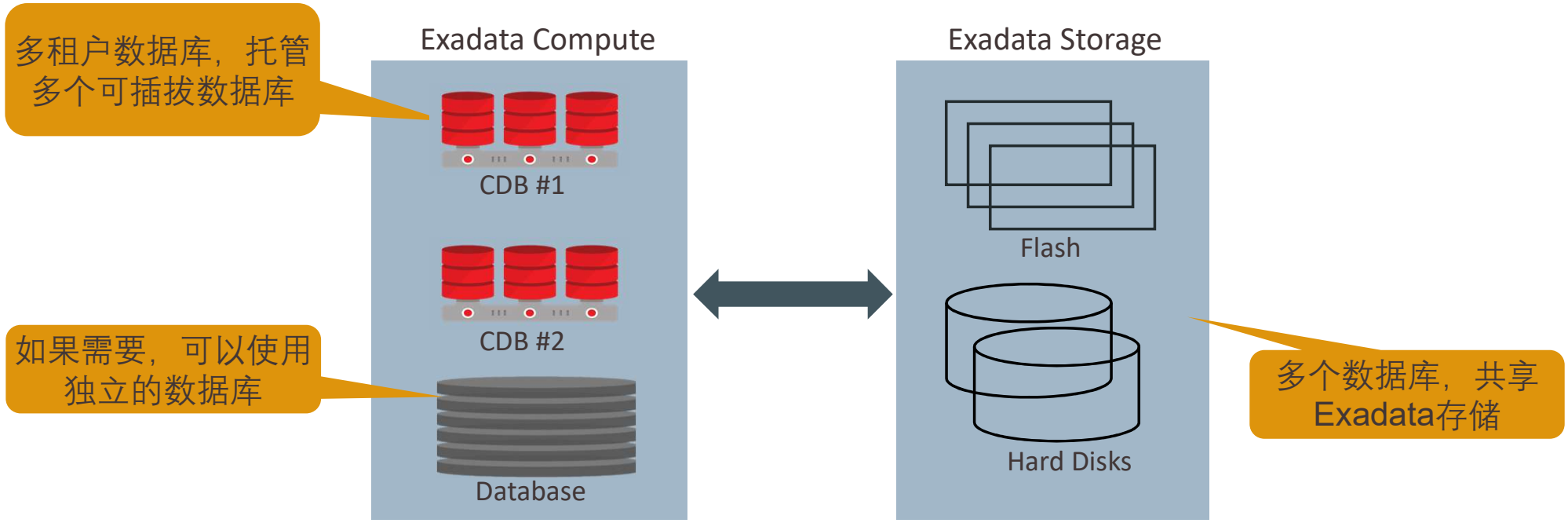
- 使用 Huge Pages
- 分配SGAs和PGAs到物理内存的大小

4. Processes

- 建立系统范围的进程数量预算
- 尽可能保持在core-count和thread-count范围内
- 越少越好，但要注意对应用程序的影响!
- 杀死闲置阻滞进程

最佳实践整合架构

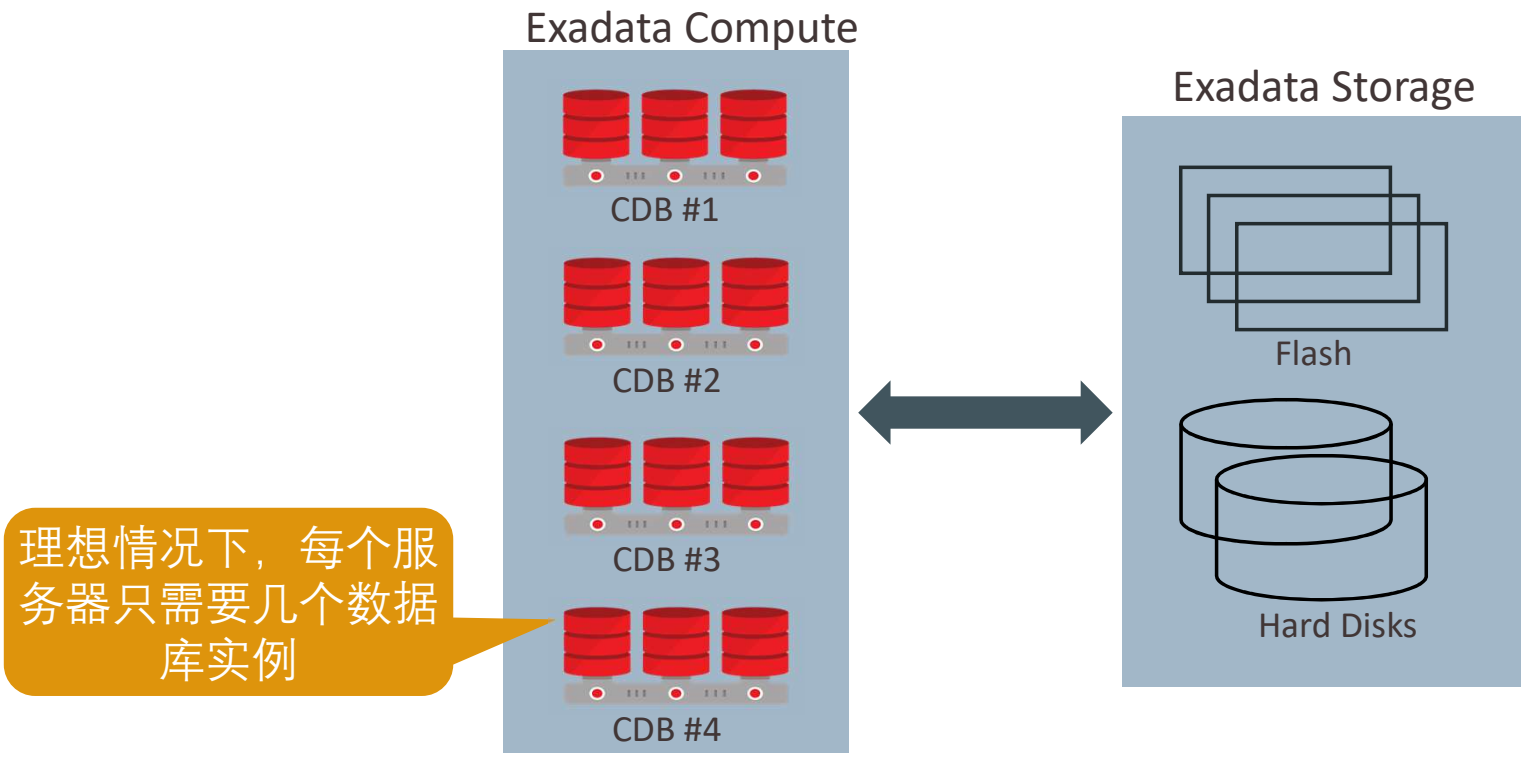
整个堆栈的完全控制- CPU到I/O



Oracle支持独立数据库和多租户整合。

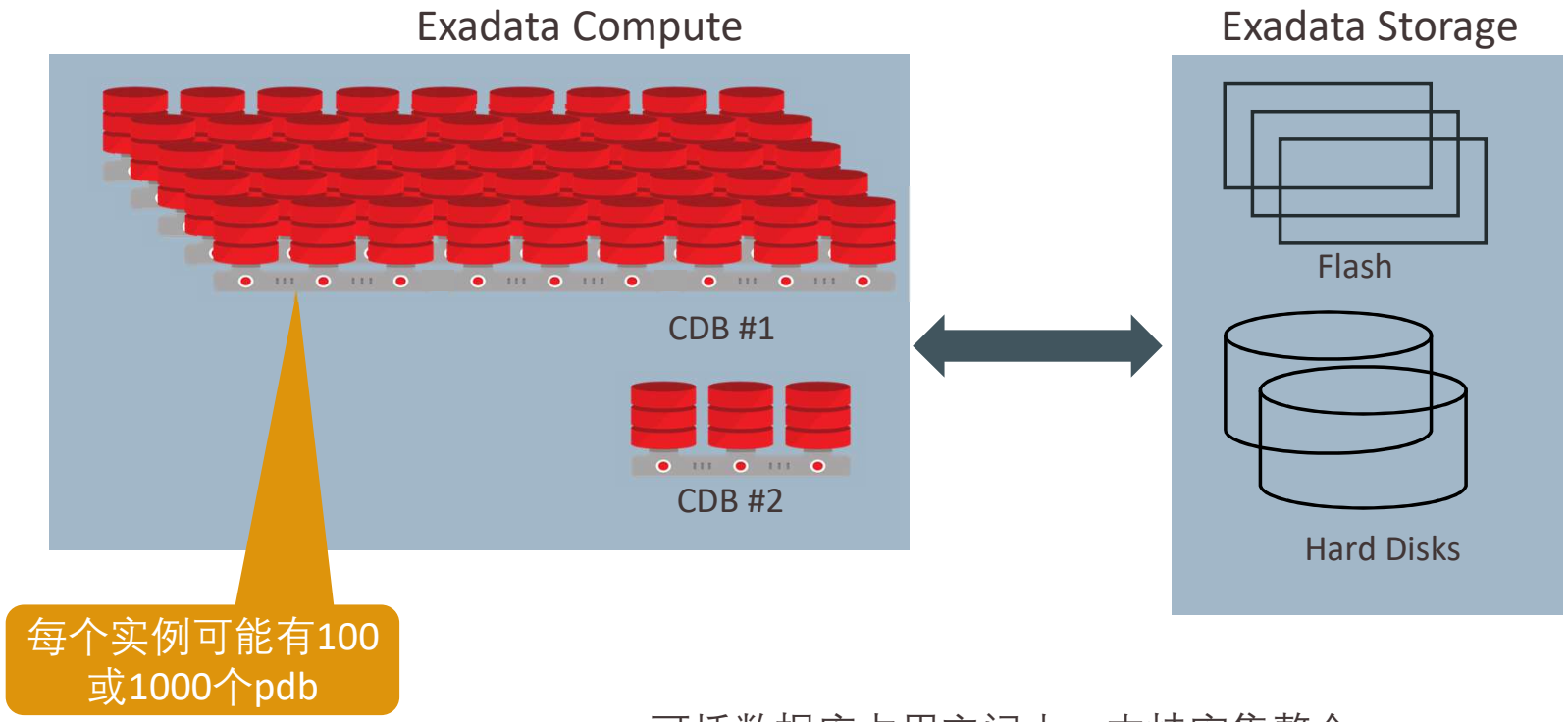
仅在Exadata中可用的I/O资源控制

数据库整合架构



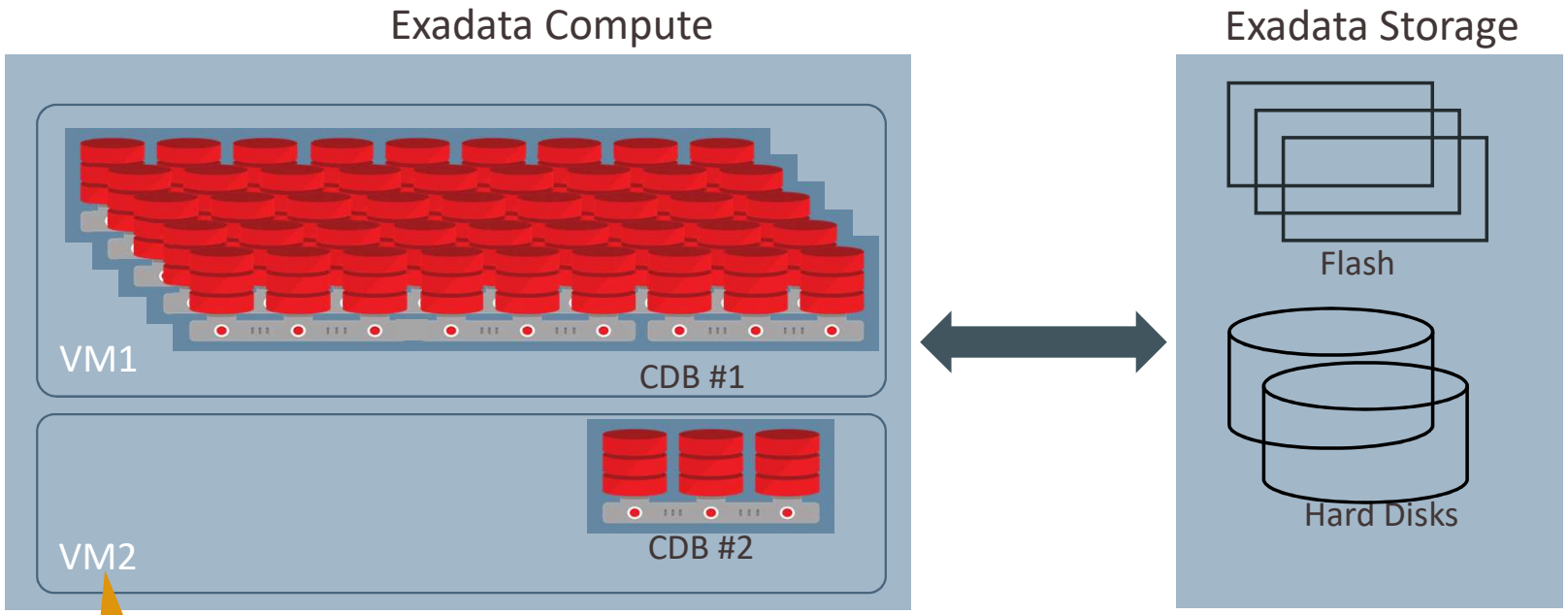
数据库实例占用空间很大。限制每个服务器的数据库实例数量!

数据库整合架构



可插数据库占用空间小，支持密集整合。
使用多租户进行大规模的整合!

数据库整合架构



一种管理方法
跨数据库、容器和虚拟机的资源

Oracle PDBs 整合密度

数据库整合密度比单DB / VM方法高出10倍以上

更少的实时处理

- 操作系统
- 集群软件
- 数据库管理系统后台进程

更小内存占用

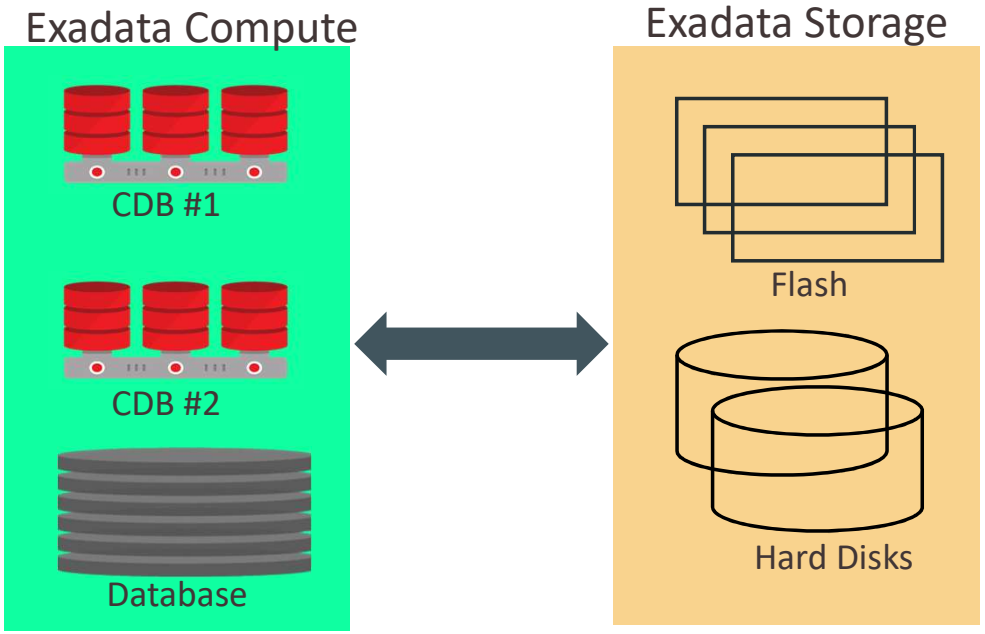
- 单一共享SGA

资源控制策略

Resource Controls 资源控制

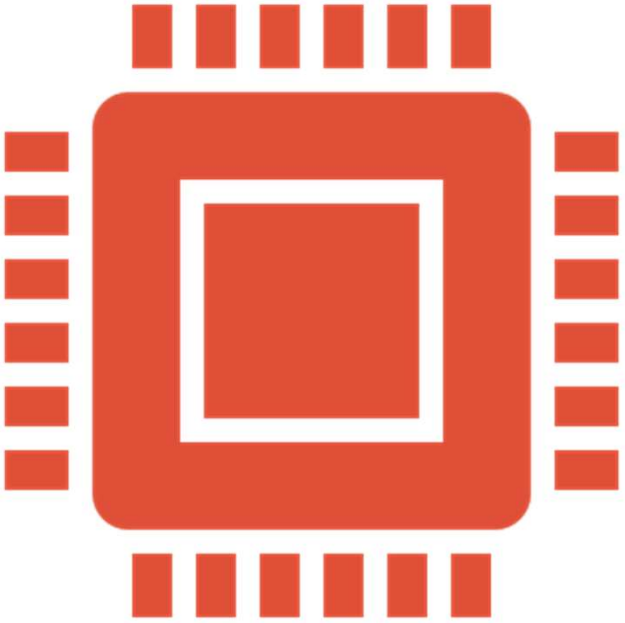
在统一环境中管理的资源

数据库服务器
CPU
SGA
PGA
Sessions
Parallel servers



Exadata Storage
Flash Cache space
Flash I/Os
Disk I/Os

CPU



资源控制

什么是Guarantees担保?

我的ISP必须保证100mbps的带宽

即使我的税涨了，我每个月也需要1000美元度假!

我的数据库需要8个cpu，即使其他数据库的工作负载激增

保证用户的“公平份额”，
防止“邻居吵闹”

什么是Limit限制?

我的ISP把我的带宽限制在100mbps(尽管他们有足够的带宽)

每年2千美元是假期的上限(即使我加薪了!)

我的PDB被限制为16个cpu，因为这是我支付的全部

限制不是为了“公平分享”!
限制提供可预测的性能，
按性能支付报酬

资源控制

2种类型的资源参数:担保和限制

大多数参数都是极限。如何配置guarantees担保?

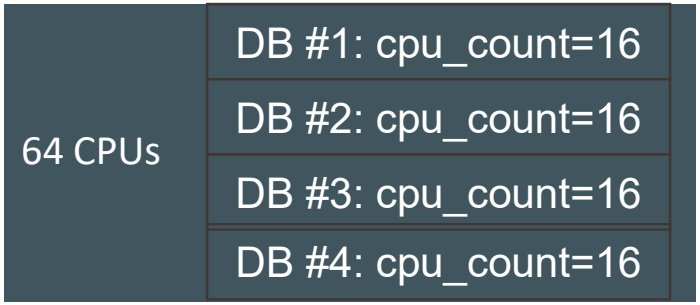
1. **通过分区担保**

通过分区担保

限制可以划分资源

什么是共享的!

Limit == guarantee



资源控制

2种类型的资源参数:保证和限制

大多数参数都是极限。如何配置保证?

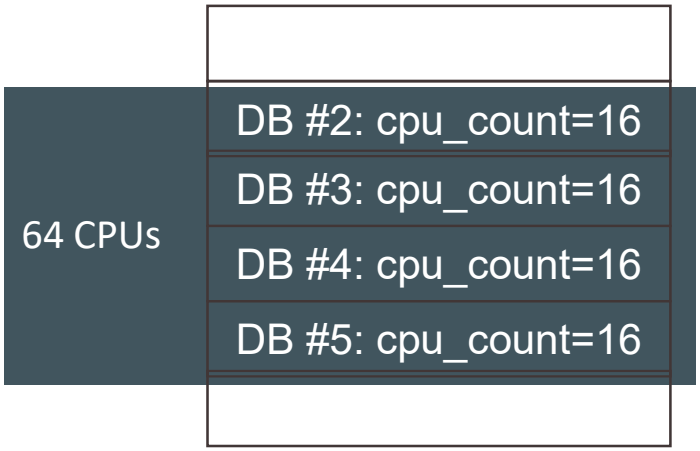
1. 通过分区担保
2. **通过过度订阅来担保，有时...**

Guarantees with Over-Subscription

只是为了一些资源。有些资源是“先到先得”的。

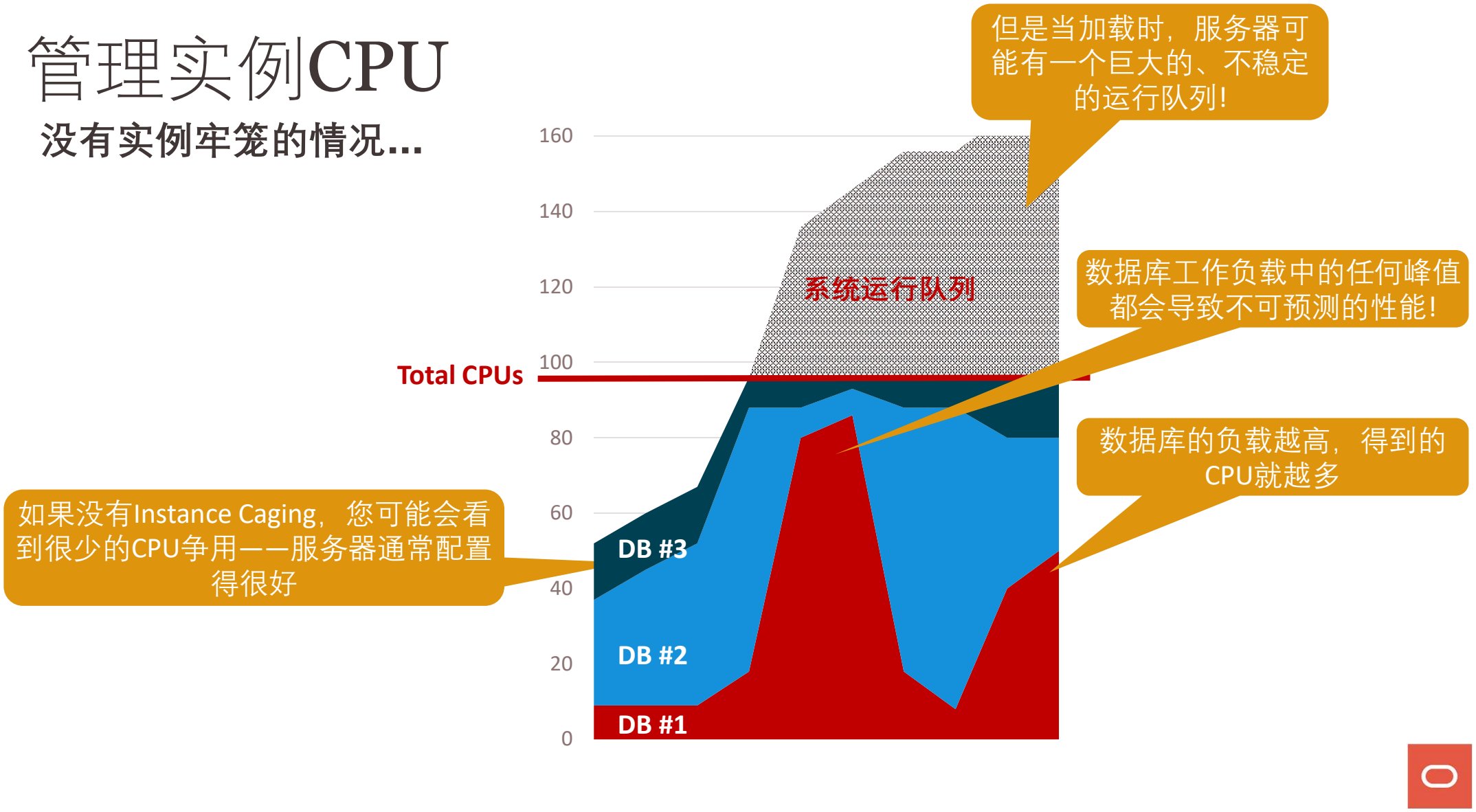
限制可以过度认购资源

限额就像股份一样，保证了资源的公平分配



管理实例CPU

没有实例牢笼的情况...



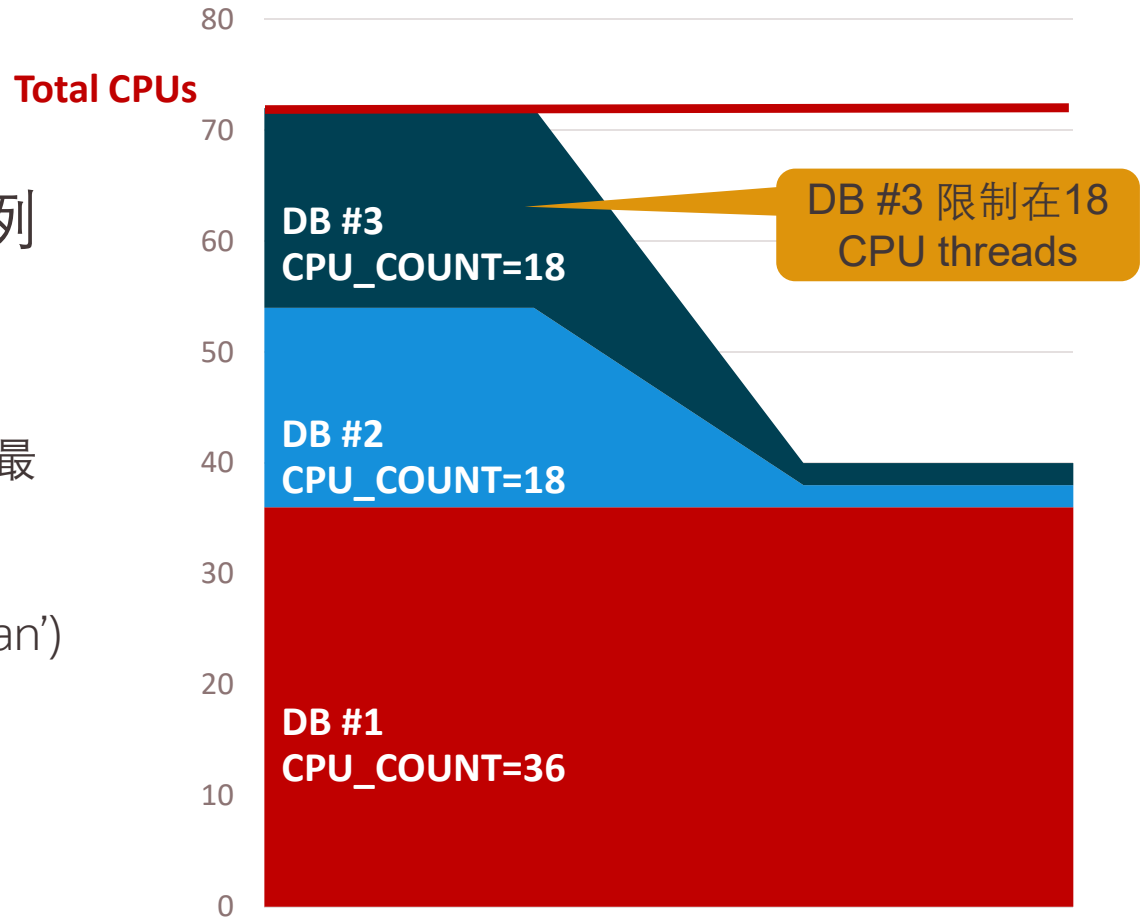
管理实例CPU

数据库实例

实例笼通过限制每个数据库实例的CPU使用来管理CPU。

仅2步:

1) 将**CPU_COUNT**设置为实例可以使用的最大CPU线程数(不是Core)
2) 配置 **RESOURCE_MANAGER_PLAN**
(e.g. to 'default_plan' or 'default_cdb_plan')



For examples and tuning, see MOS note 1362445.1

管理实例CPU

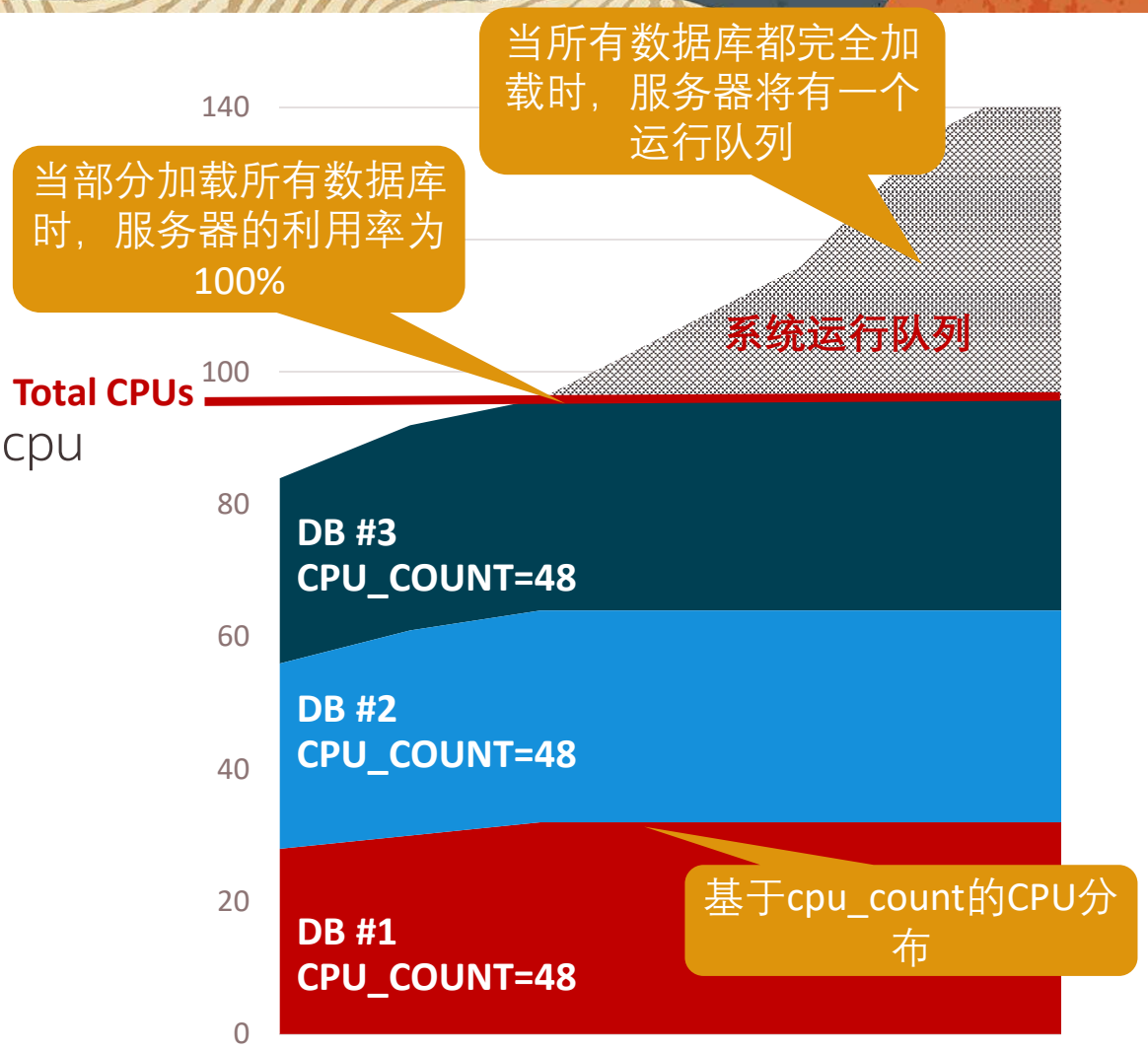
Instance Caging w/ Over-Subscription
实例牢笼 / 过度订阅

sum(cpu_counts) > total cpus on server

过度订阅Over-Subscribing可以有效地利用cpu

- 基于cpu_count的CPU使用率
- 实例可能会争夺CPU …
- 服务器可能有一个运行队列…
- 取决于超额订阅因素和总负载

!!最适合非关键数据库



For examples and tuning, see MOS note 1362445.1

管理实例CPU

为数据库分配CPU资源

1) 决定是对Exadata计算服务器进行**分区**还是**过度订阅**

   Partition: SUM(CPU_COUNT) <= System Threads

   Over-Subscribe: SUM(CPU_COUNT) > System Threads

2) 配置**RESOURCE_MANAGER_PLAN**
   (e.g. to “default_plan” or “default_cdb_plan”)

3) 将**CPU_COUNT**设置为每个实例可以使用的最大CPU线程数(不是 CORES)

4) 监控CPU使用率, 用 v$rsrcmgrmetric_history 视图和DBA_HIST_RSRC_METRIC视图

5) 根据需要动态调整cpu_count (no database bounce required!)

For examples and tuning, see MOS note 1362445.1

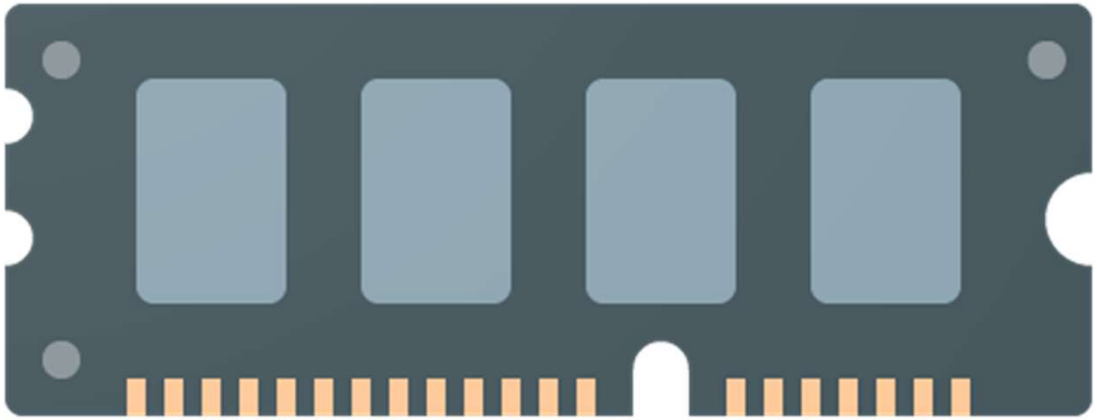
CPU 资源 Shares & Limits (PDBs)

- **高界限与低界限**
  - SHARE 是低界限(可用CPU的相对百分比)
  - LIMIT 是高界限(可用CPU百分比)
- **Shares（股份） 是相对于其他数据库的**
  - Database #1 Share = 10, Limit = 30%
  - Database #2 Share = 50, Limit = 90%
  - Total shares = 60
  - Database #1 gets 16% guaranteed (10/60)
  - Database #2 gets 83% guaranteed (50/60)
- **在添加或删除数据库时重新计算共享**
  - 设置LIMIT允许空间添加数据库
  - 考虑一下上面所示示例的含义!

动态CPU扩展（PDB）

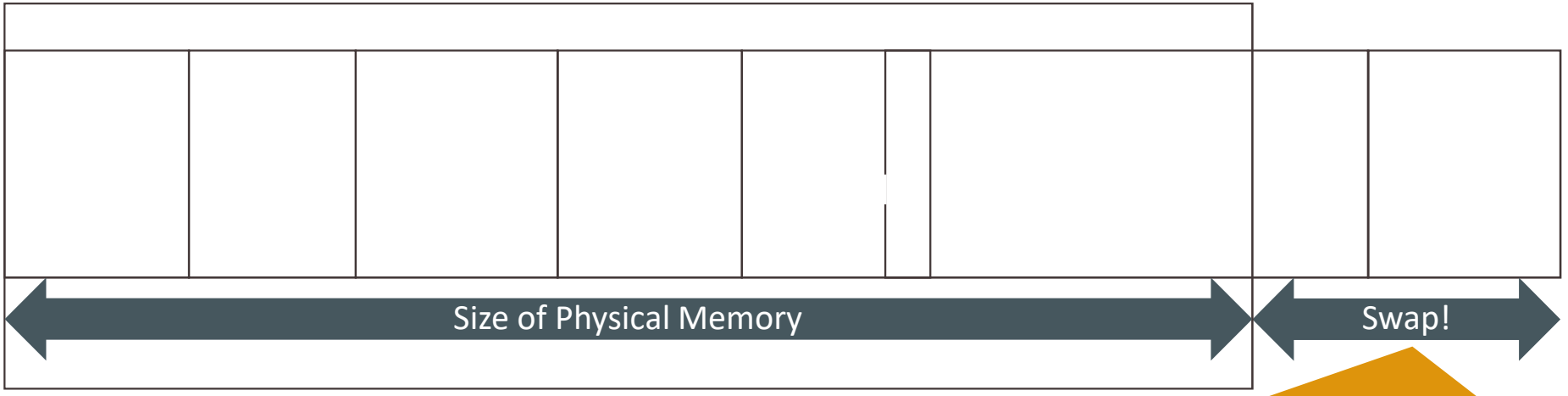
- **PDBs 可以在数据库19c(19.4)中使用动态CPU扩展**
  - 用于Oracle自治数据库的自动缩放功能
  - 数据库自动获得更多CPU(动态且即时)
  - 现在可用于任何运行Database 19c(19.4)的环境
- **最小和最大设置(在PDB内)**
  - CPU_MIN_COUNT – 每个数据库的最低担保CPU数
  - CPU_COUNT –每个数据库的最大担保CPU数
- **推荐配置**
  - 以下经验法则是基于历史趋势分析得出的
  - 下限占用率(MIN)一般为最大值的1/3
  - CPU_MIN_COUNT (为每个数据库建立所需的最小值)
  - 配置 CPU_COUNT (maximum) 为3倍CPU_MIN_COUNT
  - CPU_MIN_COUNTs (pdb的)的总和不能超过CDB的CPU_COUNT

MEMORY



管理物理内存

预防 Swapping

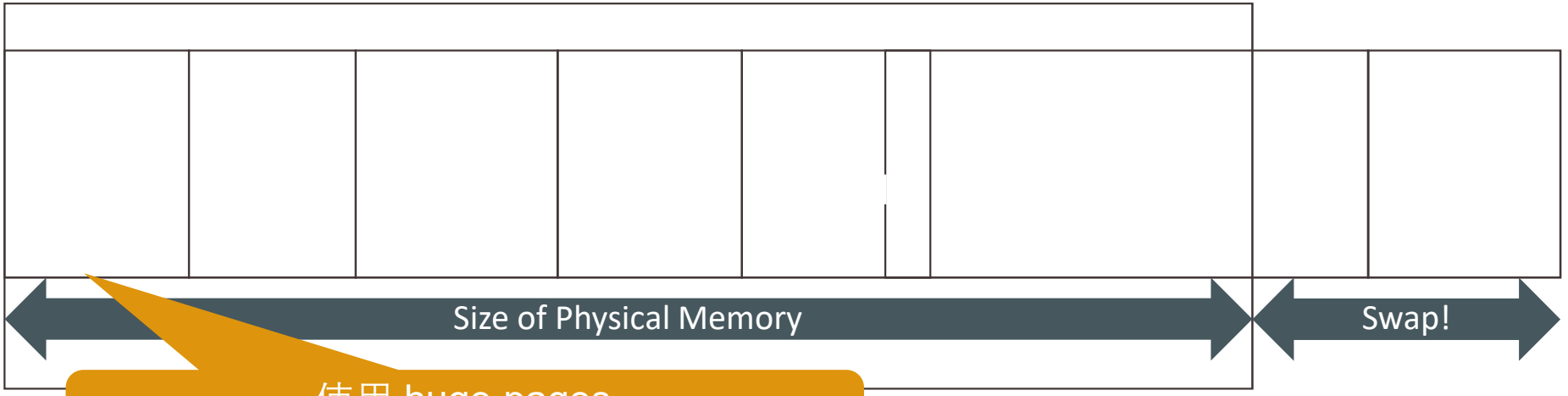


所有的内存分配应该适合物理内存!
如果没有，服务器将分页，导致性能问题和RAC驱逐。

如果 你看见*“WARNING: Heavy swapping observed on system”* 在alert log中, 要注意

管理物理内存

Linux Huge Pages



Use Huge Pages!

See MOS notes #361468.1 and #401749.1.

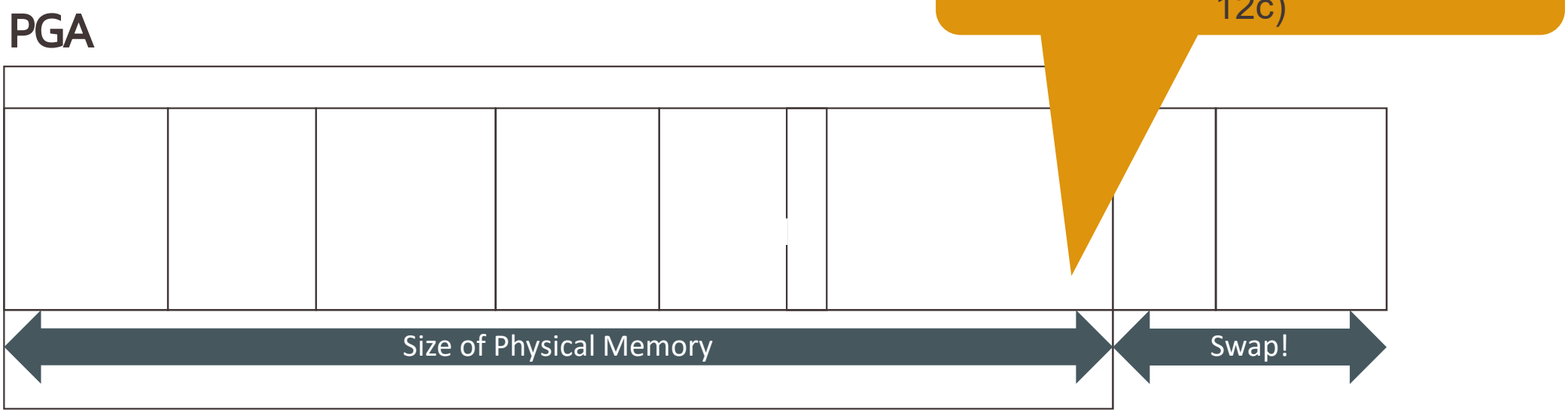
管理物理内存

PGA_AGGREGATE_LIMIT

- 谁通常是目标?
  PL/SQL 操作 内存分配过多
  BUG 数据库操作正在泄漏内存
  大多数数据密集型操作(e.g.排序和哈希连接) 使用temp临时表空间将不会是被针对

- 对“终止请求”和“杀掉会话”感到紧张吗?
  最好是针对几个行为不端的会话，而不是拥有一个不稳定的服务器!
  没有其他选项控制PGA的使用

管理物理内存

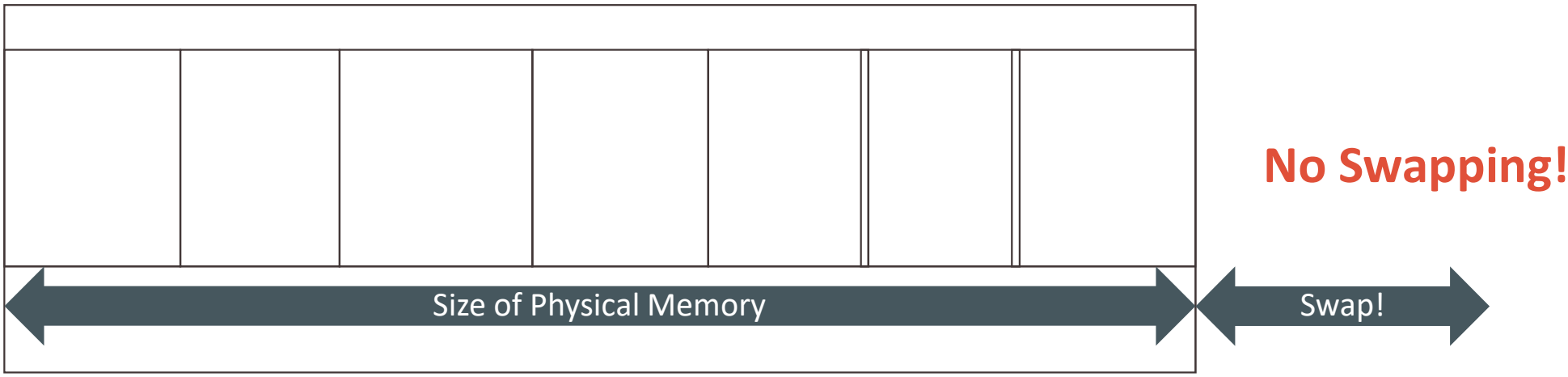


- Monitor v$pgastat
  - “maximum PGA allocated” –历史上最大
  - “total PGA allocated” –当前的使用率

确保PGA_AGGREGATE_LIMIT >= PGA_AGGREGATE_TARGET x 2! *(start in 9i)*

成功的内存管理

Living within your memory limits!



No Swapping!

- 监控 v$pgastat
  - “maximum PGA allocated” –历史上最大
  - “total PGA allocated” –当前的使用率

确保 PGA_AGGREGATE_LIMIT >= PGA_AGGREGATE_TARGET x 2!

管理物理内存

Best Practices

- **避免分配过多的内存**
  过多的交换可能导致服务器不稳定
- **预先计划内存配置!**
  Huge 面在服务器启动时配置
  SGAs不能收缩
- **使PGA限制!**
  杀死逃跑的PGA用户比破坏整个服务器的稳定要好
- **仅在小型数据库上使用MEMORY_TARGET !**
  全自动调整SGA和PGA大小
  使用SGA和PGA单独配置代替

New in 12.2!

管理CDB中的内存

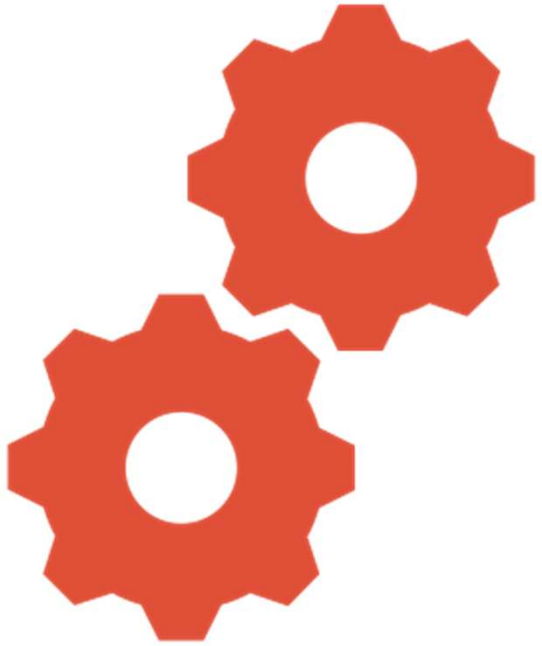
- **为什么使用多租户?**
  灵活的内存管理!
- **与单独的数据库区别**
  SGA和PGA不共享。内存大小被有效锁定!
  如果一个数据库不使用它的内存，其他数据库就不能使用它
- **使用CDB中的pdb**
  一个共享SGA和PGA
  一个PDB中未使用的内存可以被另一个PDB使用
  一个PDB的冷数据可以被另一个PDB的热数据替换
  PDBs可以快速调整大小!
  非常适合高密度整合的数据库场景!
  使用资源管理器来控制pdb如何共享内存!

Introduced in Database 12.2

管理CDB中的内存

PDB 参数	描述	何时设置
PGA_AGGREGATE_LIMIT	Maximum PGA size	针对难以控制的PDBs 验证 the limit >= 2x target. --云上总是被设置
PGA_AGGREGATE_TARGET	Target PGA size	针对难以控制的PDBs --云上总是被设置
SGA_MIN_SIZE	Minimum SGA size	用于小的pdbs or 和关键的pdbs
DB_CACHE_SIZE	Minimum buffer cache size	设置很少–如果PDB的共享池正在收缩其缓冲缓存
SHARED_POOL_SIZE	Minimum shared pool size	设置很少- 如果PDB的缓冲缓存正在收缩它的共享池
SGA_TARGET	Maximum SGA size	针对难以控制的PDBs 云上总是被设置

O/S PROCESSES



O/S Processes

- **每个2-Socket服务器“差不多”预算3168个进程**
  - 这个**3168**数字多少有些随意(假设无空闲会话的数量)
  - 我们观察到2套接字Exadata服务器运行**20,000**多个进程
  - 平均活动会话(AAS)往往在核心计数和线程计数之间达到峰值
  - 更少的DB会话在技术上更有效，但是应用程序需要管理会话
- **3,168 / 48 cores = 66 processes per core**
  - 每个线程33个进程(CPU_COUNT=1 is equal to 1 thread)
  - 会话在RAC集群中跨节点总和
- **分割会话和并行查询之间**
  - OLTP示例:95%用于会话，5%用于并行查询
  - DW示例:会话50%，并行查询50%

管理空闲会话

- **Idle or "stuck" 会话是有问题的!**
  无限期地占用PGA、锁资源、TEMP临时空间和UNDO撤消空间
  并行操作占用更多的资源
- **Solution #1**
  为每个工作负载类型(消费者组)配置不同的限制
- **Solution #2**
  诊断并杀死空闲会话
  - MAX_IDLE_TIME (Database Parameter):会话在终止前可以空闲的最长时间
  - MAX_IDLE_BLOCKER_TIME (Resource Plan Directive pre-19c):仅适用于正在阻塞其他会话的会话

Init Parm in 19c

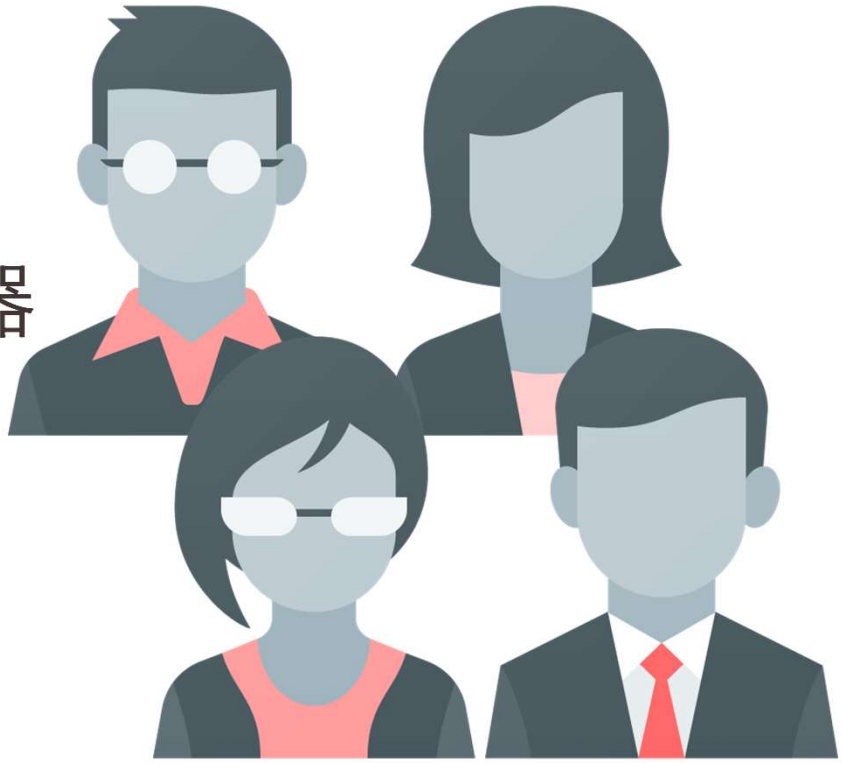
New in 12.2 and 18.1!

PDB-并行服务器管理

- Parallel servers（并行资源）是pdb中的共享资源
- 为了防止一个PDB使用所有并行服务器
  - 配置 PARALLEL_MAX_SERVERS 限制PDB的并行服务器使用
  - 配置 PARALLEL_SERVERS_TARGET启用并行语句队列
  - 通过配置PDB的CPU_COUNT来控制PDB
  - 不需要重启数据库

	Set in PDB?	默认值
Max DOP		PDB's CPU_COUNT
PARALLEL_SERVERS_TARGET	Yes! New in 18.1	从PDB's CPU_COUNT计算
PARALLEL_MAX_SERVERS	Yes! New in 12.2	从PDB's CPU_COUNT计算

Oracle数据库资源管理器
-使用者组



Resource Consumer Groups

用于更大的数据库，如数据仓库

- **Consumer Groups（消费者组）**
  - 资源用户组的标识符
- **用户到消费者组的映射（MAPPING）**
  - 数据库用户、数据库角色
  - 数据库服务
  - Application Module & Action
  - 客户端程序，客户端O/S用户，客户端机器，客户端ID
- **资源计划（Resource Plans）**
  - 为每个消费组分配资源
    - Shares（股份），Percentages（百分比），Utilization Limits（利用率）等数据库级的资源
    - Parallelism平行度 (bypass queue, Max DOP, Parallel Server Limit, PQ Timeout)
    - Runaway Query Control 失控查询 (限制或采取行动)
    - Idle Time 空闲时间 (Max Idle and Max Idle Blocker)

管理失控的SQL查询

失控查询可能由以下原因引起

- 写得很糟糕的SQL
- 糟糕的执行计划

严重影响行为良好的查询的性能

- 消费PQ Server

很难完全根除!

- 在“更大”的数据库中管理这些

管理失控查询

Configure by Consumer Group

定义 失控的查询阈值:

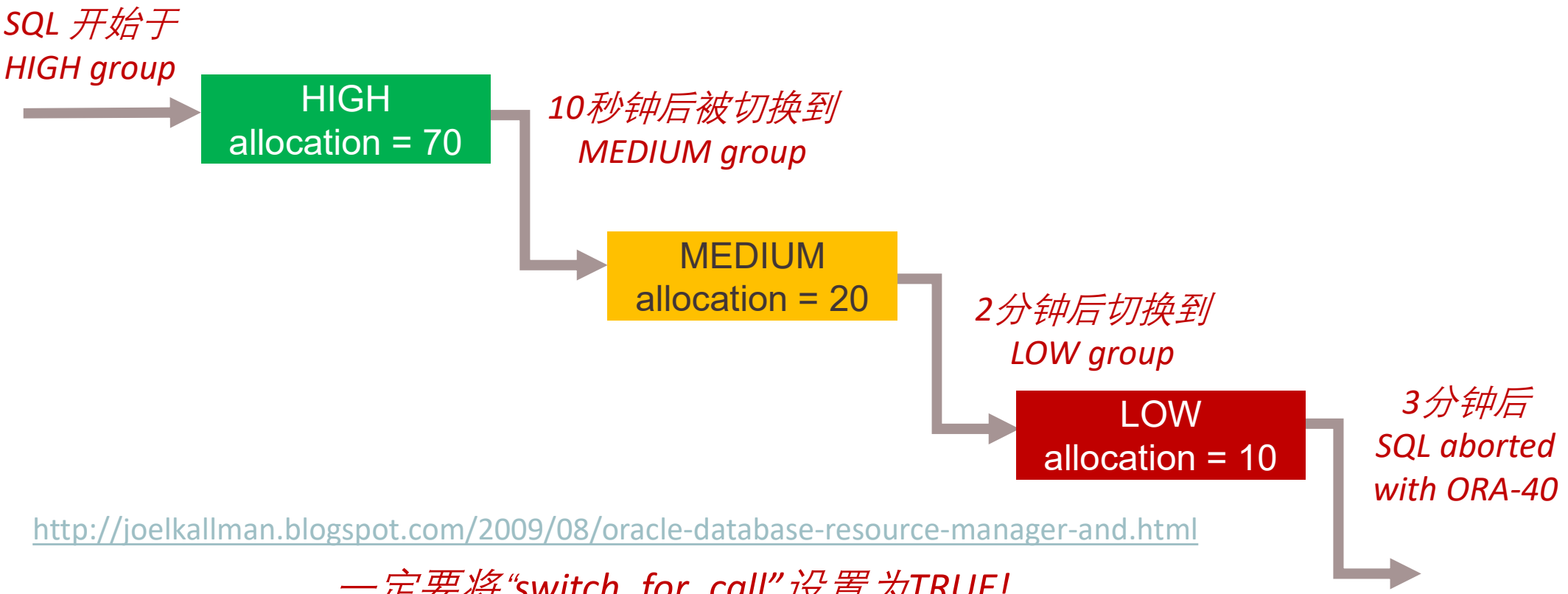
- ✓估计执行时间
- ✓运行时间 **New in 12c**
- ✓所使用的CPU时间
- ✓I/O请求数量
- ✓I/O请求字节
- ✓逻辑I/O数量 **New in 12c**

管理 失控的查询:

- ✓切换到较低的消费群体
- ✓中止请求
- ✓杀掉会话
- ✓使用SQL Monitor New in 12c

使用资源管理器自动检测和管理失控查询

管理失控查询



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

一定要将“switch_for_call”设置为TRUE!

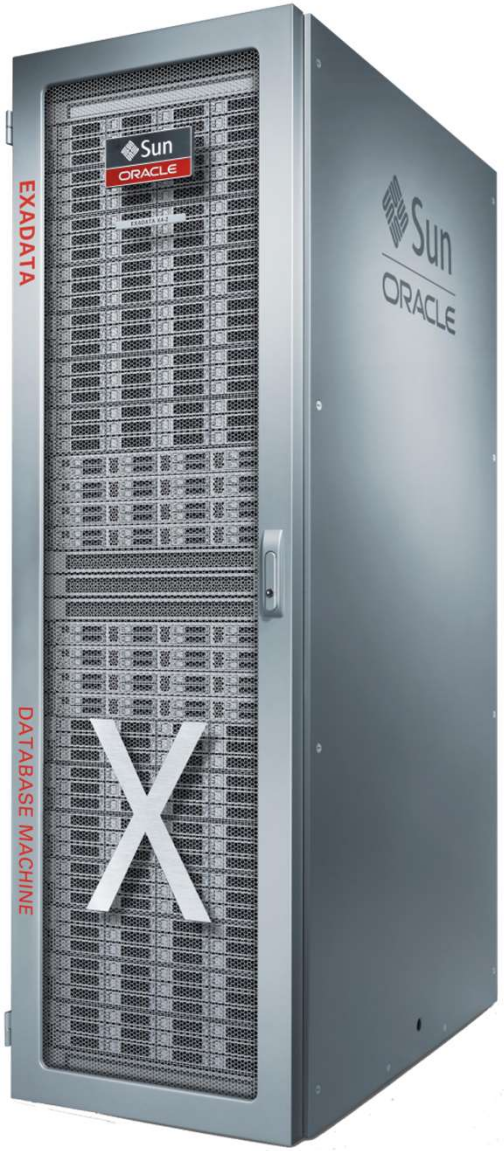
New in 12c!

管理失控查询

查看v$sql_monitor

Column	Description
RM_CONSUMER_GROUP	当前用户组名
RM_LAST_ACTION	采取的行动(if any): SWITCH TO <consumer group name> CANCEL_SQL KILL_SESSION LOG_ONLY New in 12c
RM_LAST_ACTION_REASON	采取上述行动的原因:: e.g. SWITCH_CPU_TIME
RM_LAST_ACTION_TIME	采取这种行动的时间

I/O Resource Manager



Exadata IORM-继承CPU比率

基于CPU分配比率管理I/O

配置的Exadata IORM

- 设置IORM目标为“auto” IORM objective = “ AUTO ”
- 这导致IORM在所有RAC实例中继承为CPU指定的相同比率
- IORM自动计算这些比率(CPU设置在实例级别)

你也会得到...

- 包含与IORM objective = “ basic ”（默认值）相同的特性
- OLTP自动优先于扫描
- 基于优先级的flash和I/O调度

Oracle的自治数据库云使用这种方法

- 自治数据库云支持数千个统一数据库

但如果你需要更多的控制……

FlashCache和PMEM缓存控件

I/O资源的高级控制

- **FlashCache控制**
  - Minimum – flashcachemin 最小值
  - Limit – flashcachelimit 限制
  - Space – flashcachesize SIZE
- **PMEM的控制与FlashCache控制相似**
  - Minimum - pmemcachemin
  - Limit - pmemcachelimit
  - Space – pmemcachesize
- **这些被认为是“高级”控件**
  - 大多数情况下不需要使用
  - 对最关键的数据库使用这些控件

Exadata IORM

管理Flash Space

Minimum 保证数据库的空间. 对于有时不活动的关键数据库很有用。

Soft **Limits**仅在Flash Cache已满时使用

Size limits并为数据库预留空间。使用谨慎!

Inter-Database IORM Plan			
Database	Flash Cache Min	Flash Cache Limit	Flash Cache Size
DB-1	100 GB		
DB-2		2 TB	
CDB-1			10 TB

"设置"是每个cell需要指定的。

也适用于KEEP数据!

Exadata存储的优势

唯一为整合而建的存储!

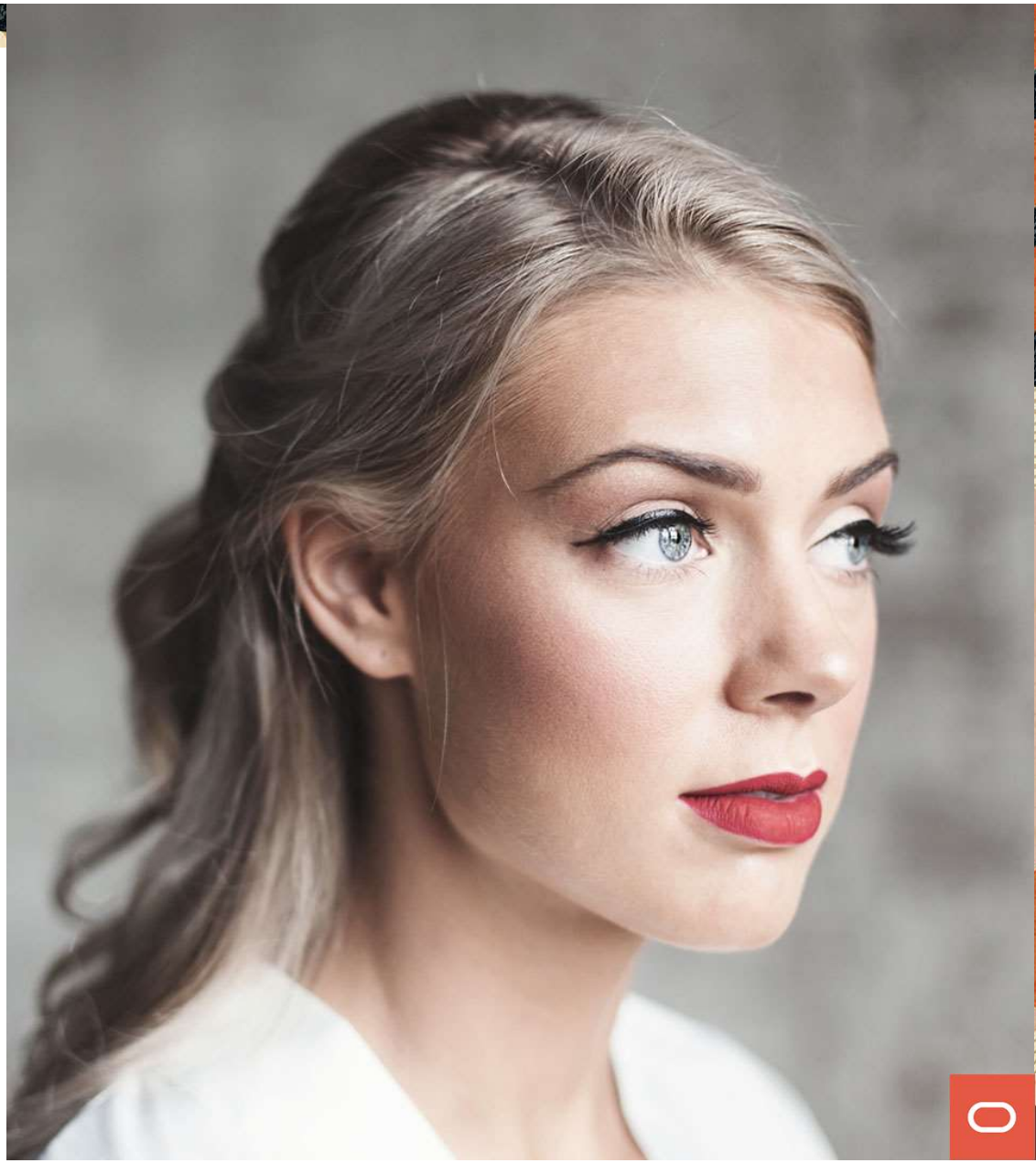
- **普通存储无法控制数据库的磁盘或flash带宽**
  - 你需要使用操作系统技术或虚拟机技术来限制数据库的IOPS
  - 或者为每个数据库使用专用磁盘
  - 无需多租户数据库的工具!
- **Exadata IORM提供IO优先级，基于您的策略!**
  - 您可以配置磁盘和闪存的最小带宽和限制
  - 您可以配置**flash space**最小带宽和限制**New in 12.1.2.1!**
  - 自动将**OLTP优先于扫描**(数据仓库)工作负载
  - 支持cdb、pdb和独立数据库
  - Easy to configure!

调优IORM

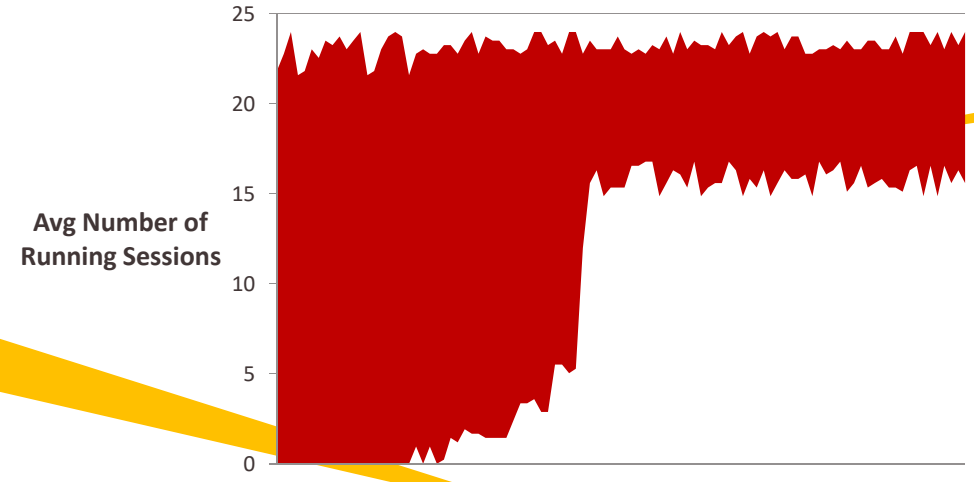
Best Practices

- **如果有足够的带宽可用，不要指望IORM做任何事情!**
  利用率限制是例外
  配置利用率限制主要是为了满足性能需求
- **对于OLTP，请关注**
  Flash Cache命中率
  Average latency 平均延迟
- **对于OLAP，请关注**
  Flash Cache命中率
  闪存和磁盘吞吐量(MBps)

Resource 监视

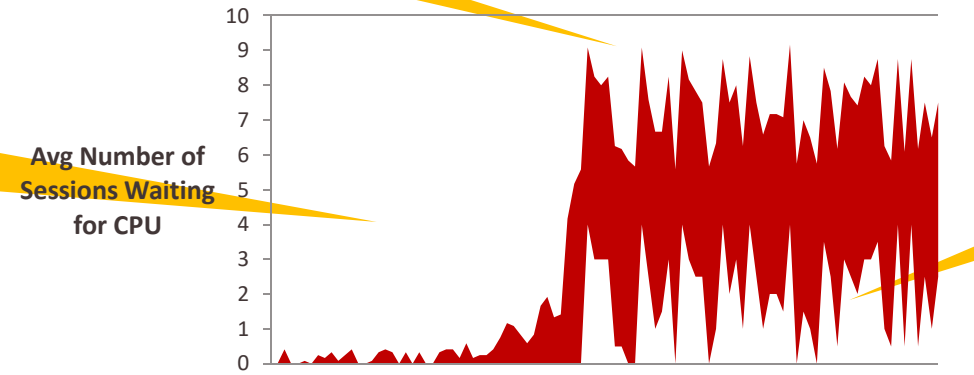


监视和调整CPU资源管理器



监视CPU是如何实际分布的

如果有任何会话正在等待CPU，则数据库实例将受益于更大的服务器。

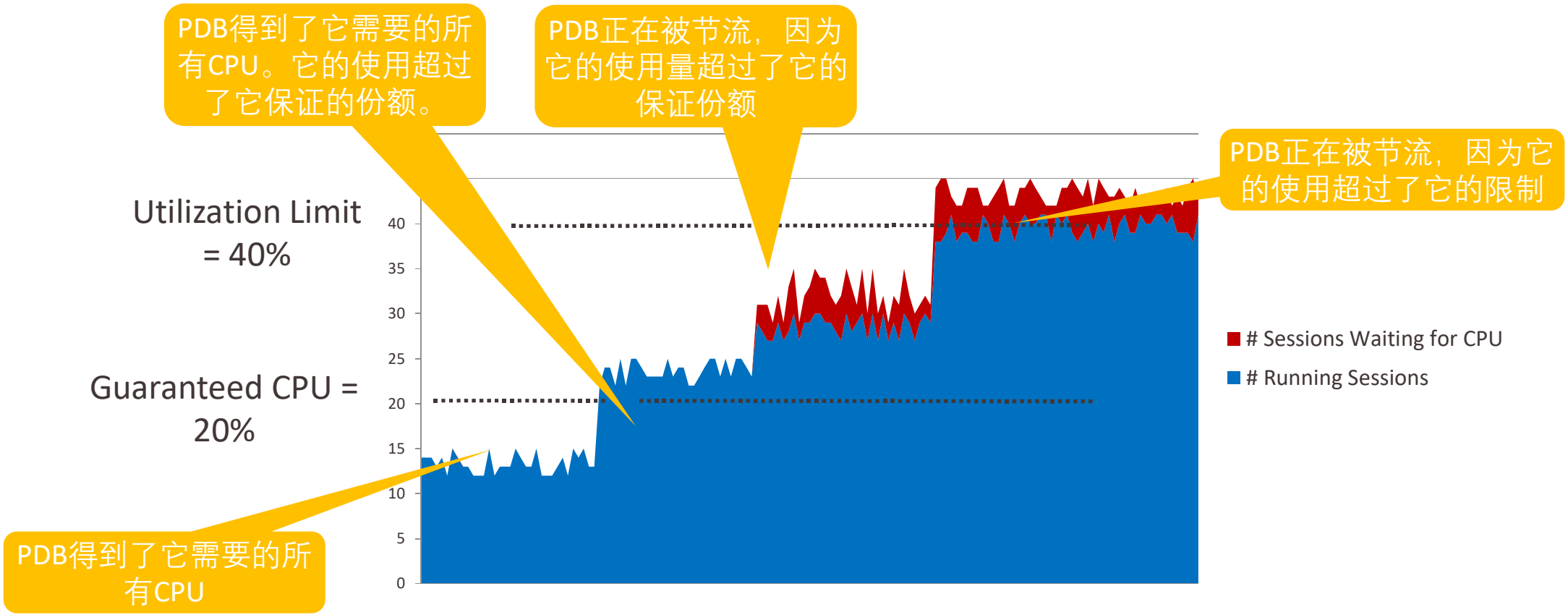


如果会话没有等待CPU，资源管理器就不会积极管理

如果SALES会话正在等待CPU，那么它们将受益于更大的资源分配。

Monitor using v$rsrcmgrmetric_history or Enterprise Manager

监视和调整PDB



查看PDB如何受其资源计划设置的影响。

监控视图v$rsrcmgrmetric_history.

监视和调整CPU资源管理器

- **配置资源计划是一个迭代过程**
  - 创建资源计划
  - 监视应用程序性能和资源管理器指标
  - 调整资源分配并重新监控
- **会话等待CPU是否不好?**
  - 相当于在O/S运行队列上等待的进程
  - 大量等待会话意味着服务器过载!

  如果表现不令人满意，增加对该消费群体的资源分配
  有了资源管理器，关键的后台进程和O/S不会匮乏

监视PDB资源使用情况

v$rsrcpdbmetric_history: CPU

Metric	Description
CPU_CONSUMED_TIME	Milliseconds of CPU consumed
CPU_WAIT_TIME	Milliseconds that this PDB waited to be scheduled. Equivalent to waiting in the OS run queue.
NUM_CPUS	CPU_COUNT for the CDB
RUNNING_SESSIONS_LIMIT	Maximum number of sessions that can be concurrently on CPU. CPU_COUNT for the PDB. PDB's utilization_limit x CDB's CPU_COUNT.
AVG_RUNNING_SESSIONS	Average number of sessions that are on CPU
AVG_WAITING_SESSIONS	Average number of sessions that are waiting on CPU
CPU_UTILIZATION_LIMIT	Maximum CPU utilization, with respect to the host server
AVG_CPU_UTILIZATION	Average CPU utilization, with respect to the host server

监视PDB资源使用情况

v$rsrcpdbmetric_history: Memory

Metric	Description
SGA_BYTES	SGA used, in bytes
BUFFER_CACHE_BYTES	Buffer cache used, in bytes
SHARED_POOL_BYTES	Shared pool used, in bytes
PGA_BYTES	PGA used, in bytes

监视PDB资源使用情况

v$rsrcpdbmetric_history: I/O

Metric	Description
IOPS	I/O requests per second
IOMBPS	Megabytes of I/Os per second
AVG_IO_THROTTLE*	Average number of milliseconds each I/O was throttled, due to I/O rate limits. *Note: 20c feature will be backported to 19c.

Non-Exadata Platforms (use MAX_IOPS, MAX_MBPS init parameters)

Metric	Non-Exadata Parm	Description
IOPS_THROTTLE_EXEMPT	MAX_IOPS	I/O requests per second that are exempted from being throttled. (Primarily DBWR writes.)
IOMBPS_THROTTLE_EXEMPT	MAX_MBPS	Megabytes of I/Os per second that are exempted from being throttled. (Primarily DBWR writes.)

监视PDB资源使用情况

v$rsrcpdbmetric_history: Parallel Execution

Metric	Description
AVG_ACTIVE_PARALLEL_STMTS	Average number of active parallel statements
AVG_QUEUED_PARALLEL_STMTS	Average number of queued parallel statements
AVG_ACTIVE_PARALLEL_SERVERS	Average number of active parallel servers
AVG_QUEUED_PARALLEL_SERVERS	Average number of queued parallel servers
PARALLEL_SERVERS_LIMIT	Maximum number of parallel servers that can be used

监视PDB资源使用情况

使用这些资源管理器指标监视PDB历史内存使用情况

提供CPU、内存、I/O和并行执行的统计信息

v$rsrcpdbmetric_history

过去一小时的每分钟数据

dba_hist_rsrc_pdb_metric

用AWR快照捕获

默认情况下，过去两周的每小时统计数据

监控 IORM

Introduced in
RDBMS 12.1.0.2
OSS 12.1.2.1

AWR Reports

Top Databases by IO Requests

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

Are the I/Os from flash or disk?

每个数据库生成多少I/O？

		IO Requests					IO Throughput (MB)			
DB Name	DBID	%Captured	Total Requests	per Sec	Flash	Disk	Total MB	per Sec	Flash	Disk
ESJ1POD	4036668196	99.75	26,480,820	81,730.93	26,461,139	19,681	1,651,610.58	5,097.56	1,651,140.47	470.12
ASM	1	0.23	62,269	192.19	2,520	59,749	97.13	0.30	9.84	87.28
OTHER	0	0.02	4,882	15.07	4,850	32	53.47	0.17	46.42	7.05

Back to Exadata Top Database Consumers
Back to Exadata Statistics

Top Databases By Requests - Details

- Request details for the top databases by IO requests

IORM抑制了多少

			Small Requests							Large Requests						
			Reqs/s			Latency		Queue Time		Reqs/s			Latency		Queue Time	
DB Name	DBID	IOs/s	Total	Flash	Disk	Flash	Disk	Flash	Disk	Total	Flash	Disk	Flash	Disk	Flash	Disk
ESJ1POD		81,730.93	223.93	164.52	59.41	83.85us	104.48us	31.41us	3.93us	81,507.00	81,505.66	1.34	371.20us	471.00us	5.35ms	3.97us
ASM		192.19	192.07	7.78	184.29	102.58us	94.42us		2.67us	0.12	0.00	0.12		140.46us		3.69us
OTHER		15.07	14.49	14.49	0.00	62.59us				0.58	0.48	0.10	334.35us	121.88us		

Back to Exadata Top Database Consumers
Back to Exadata Statistics

Introduced in RDBMS 12.1.0.2, Exadata 12.1.2.1

Exadata IORM

Managing Flash Space

检查闪存缓存问题，使用Exadata部分在AWR报告考虑根据这些结果调优闪存缓存设置

Exadata Write-Back Flash Cache (WBFC) is enabled by default (since April 2017).

Flash Cache Performance Summary

I/O Summary

- average requests/second and MB/s per disk

	Requests		Throughput	
Disk Type	Reads/s	Writes/s	Read MB/s	Write MB/s
F/1.5T	559.75	1,794.56	12.29	13.22
H/3.6T	1.93	226.63	0.11	3.83

Flash Cache Savings

- Disk write savings (overwrites) - writes absorbed by flash cache that would have otherwise gone to disk
- Database Flash Cache Hit% - for the database, not restricted to an instance
- Cell OLTP and Cell Scan Flash Cache Hit% - for the cells, not restricted to this database or instance

Database Flash Cache Hit %	98.86
Cell OLTP Hit %	98.71
Cell Scan Hit %	97.8
Disk Write savings/s	48,071.38

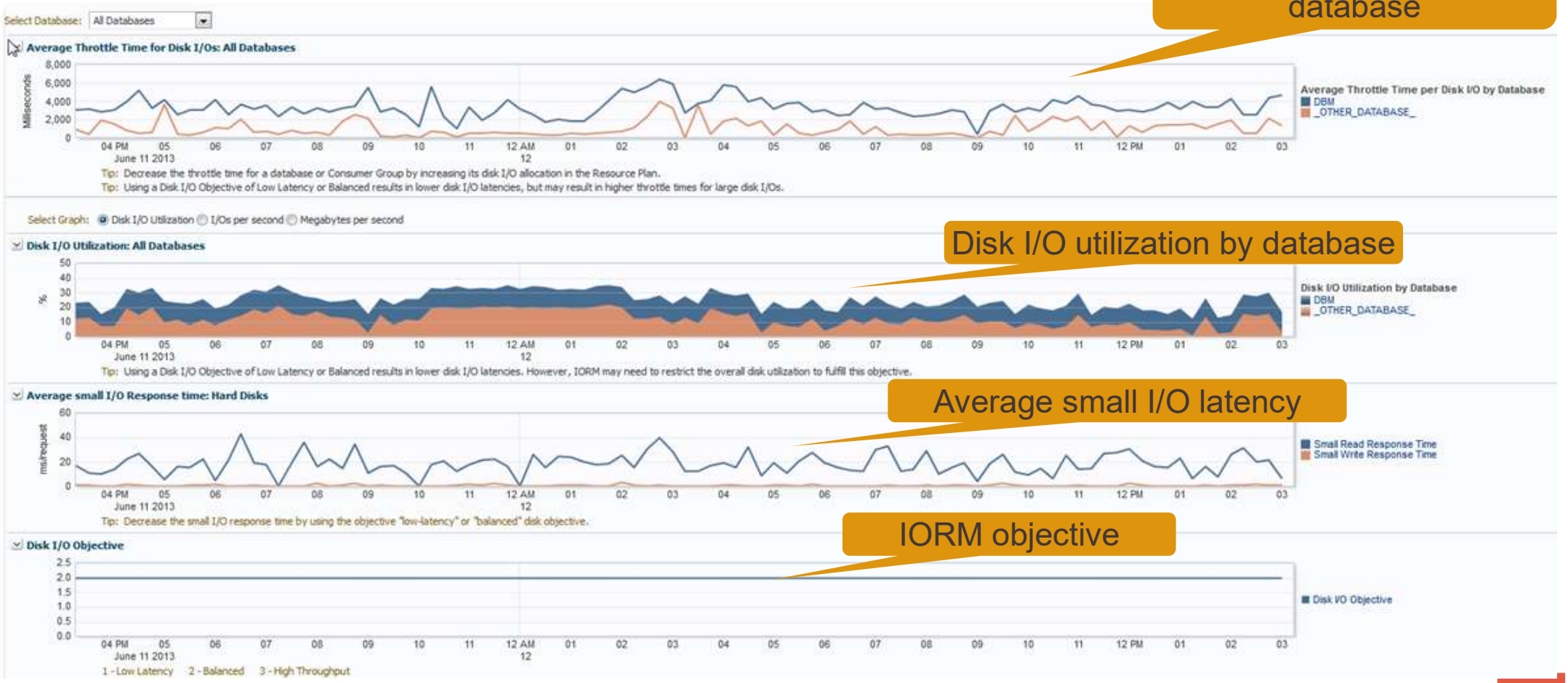
Disk Activity

- Flash cache activity resulting in disk I/O
- Read Misses - reads from disk, writes to flash
- Disk Writer - reads from flash, writes to disk

I/O per second	
Read Misses	303.96
Disk writer	539.92
% Utilization	
H/3.6T	2.20

Exadata Write-Back Flash Cache - FAQ (Doc ID 1500257.1)

Enterprise Manager IORM UI



Target将超过350个系统整合到Exadata上

Target 百货公司是美国僅次於沃尔玛的第二大零售百货集团

$1.33M
降低购置成本

10x
更快的备份时间

250%
订单流程改进

CUSTOMER PERSPECTIVE

“我们已经在Exadata基础设施上整合了350多个系统。这带来了更大的灵活性。这提高了速度。这也大大改善了我们的操作框架。我们能够更快地对事故作出反应，并提高了整体稳定性。”

— Tony Kadlec, Senior VP, Infrastructure and Operations, Target

ORACLE

DEMO演示

Oracle Resource Manager Demo #1
- Pluggable Databases

每周五 11:00 ~ 12:00
扫描二维码安装手机Zoom进入直播
https://oracle.zoom.com.cn/s/99636023345
Zoom ID: 996 3602 3345 密码: 20212021

行业大咖、技术大拿专业直播分享

甲骨文数据库与云系列公益讲座

甲骨文数据库、一体机、云服务最新前沿技术、优秀案例及解决方案分享

加入19c 公益课微信群，获取更多技术咨讯

甲骨文云技术
官方微信公众号