ORACLE

资源管理，优化数据库资源管理效率，提高用户体验

Exadata资源管理

FDDN，IO latency capping，ASM数据保护

1 Exadata 资源管理

2 FDDN，IO latency capping及ASM数据保护技术

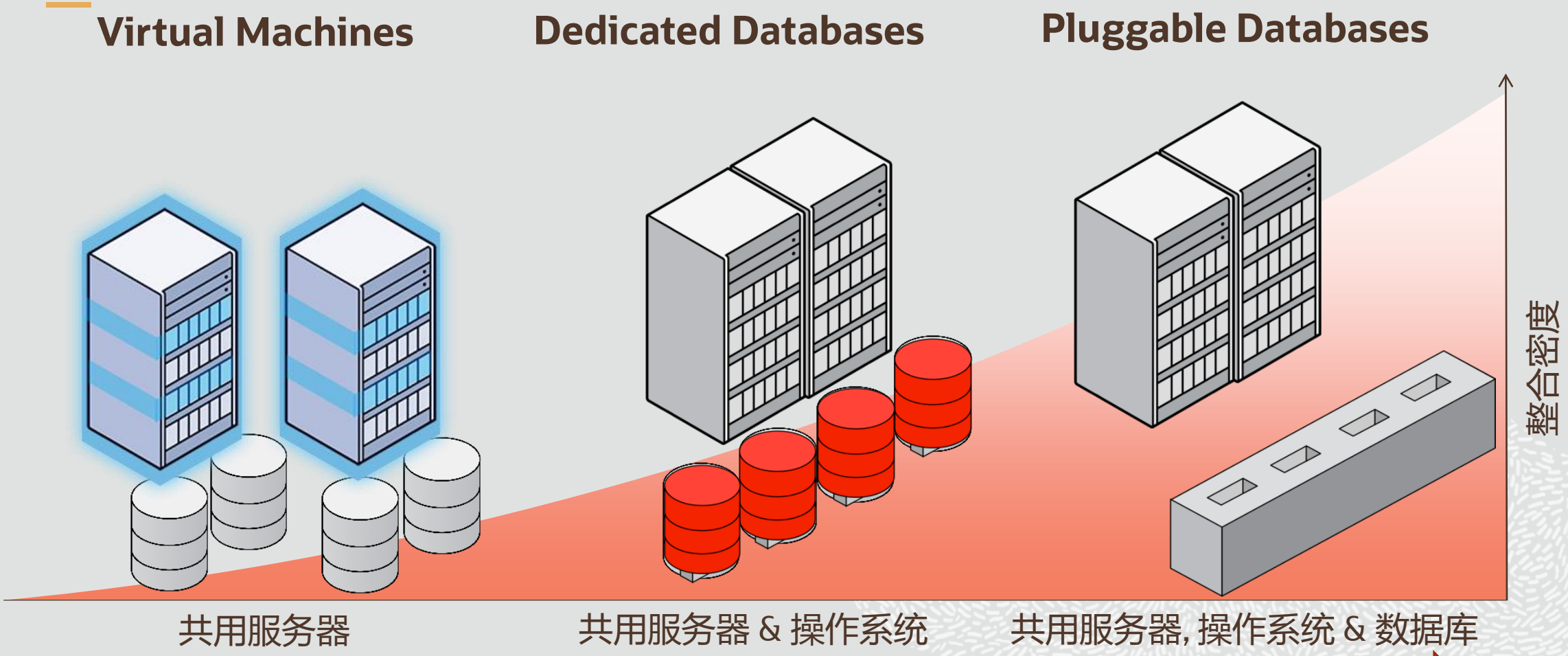
传统数据库部署模式

传统专用模式导致服务器和软件泛滥

- 曾是唯一切实可行，能保证好的服务质量的办法
  - 性能，扩展性，可靠性
- 导致了很高的费用
  - 硬件，软件，人工成本
- 需要管理大量的非标准部件
  - 服务器，数据库，操作系统
- 一般通过专用的服务提供
  - 配置一个新的方案往往比较复杂，导致部署缓慢

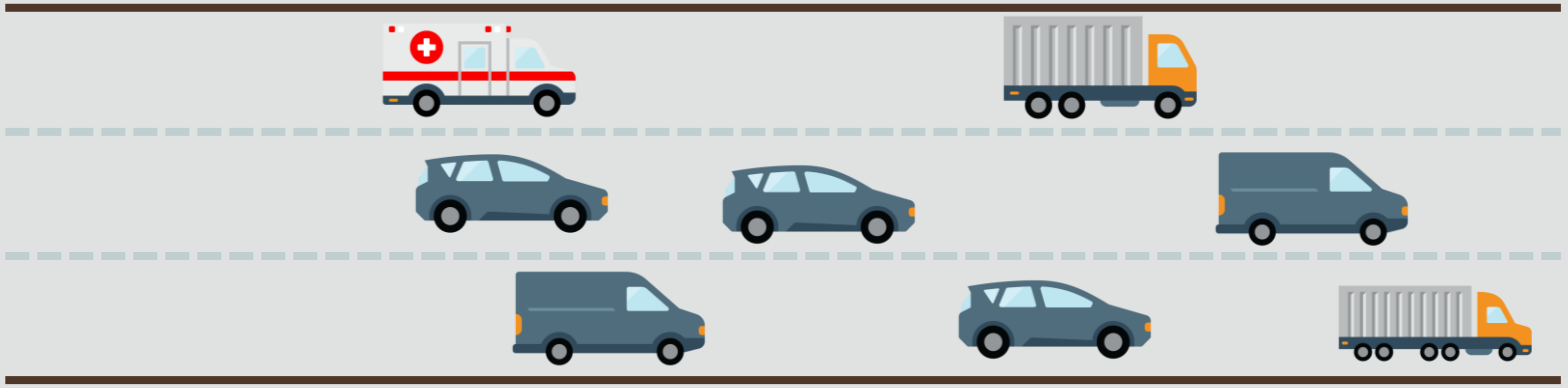
数据库服务提供架构

基于Oracle Database 12c



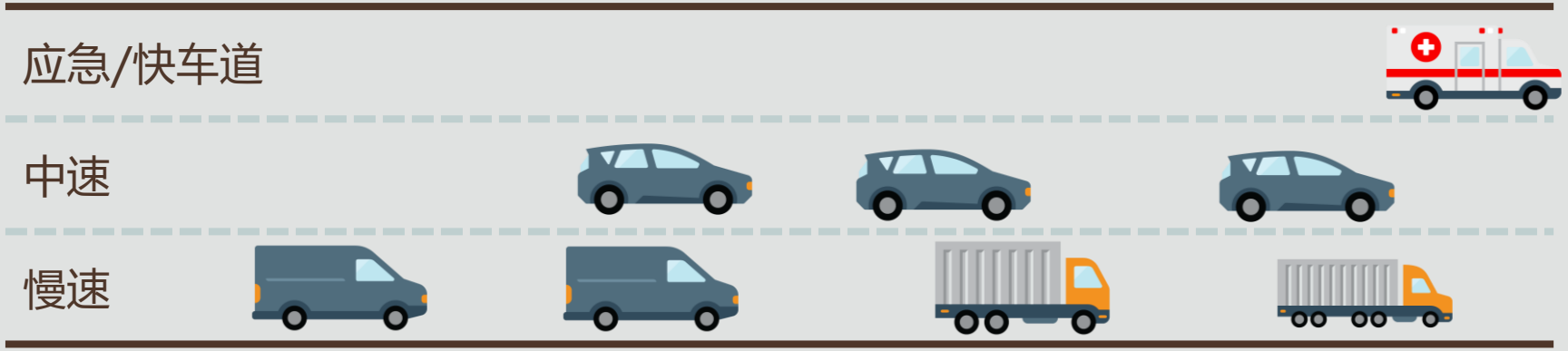
资源管理

对于业务系统的意义

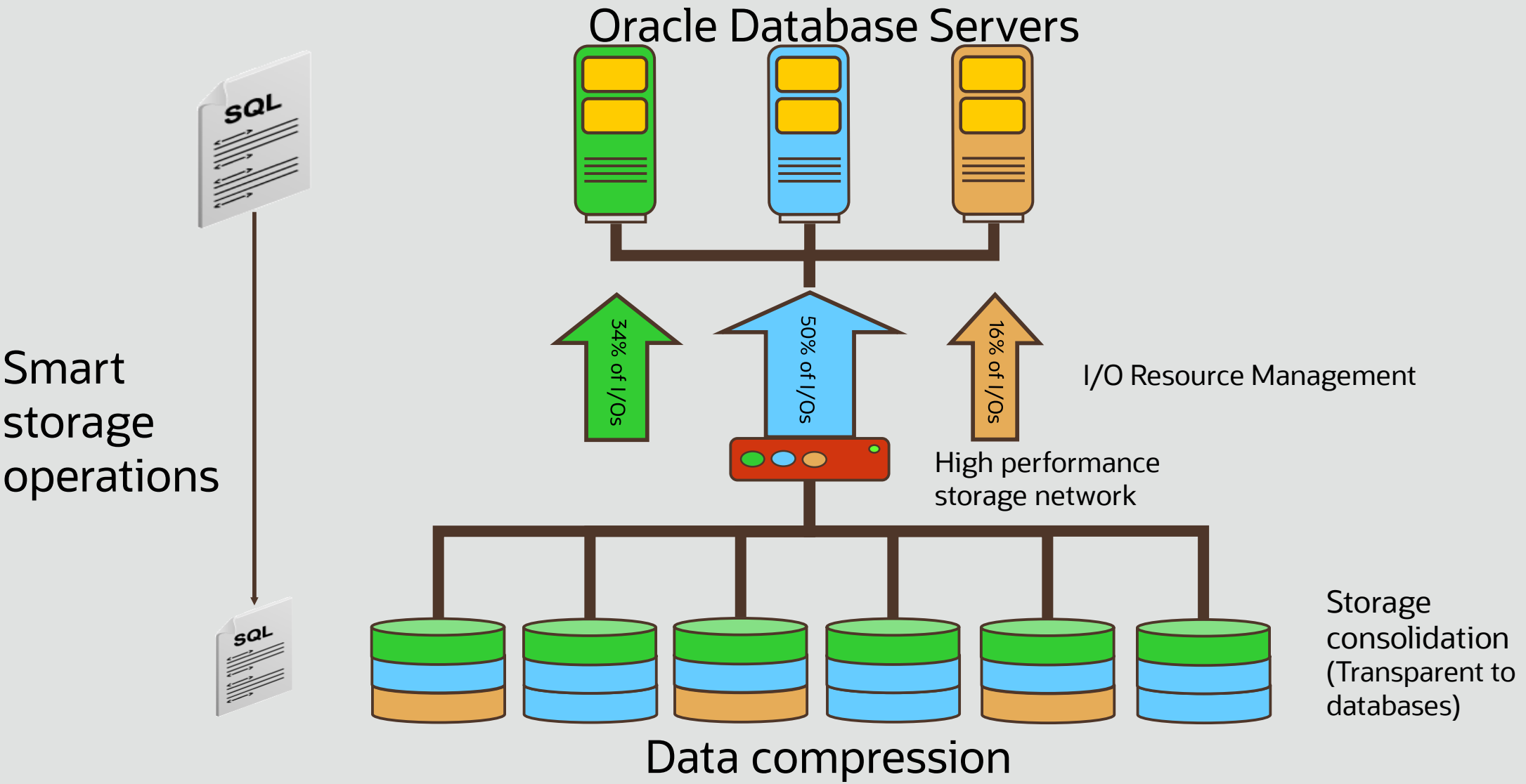


当没有资源管理时，任意数据库可能拖慢其它库

Exadata 对于资源的管理可以让数据库跑得更快.

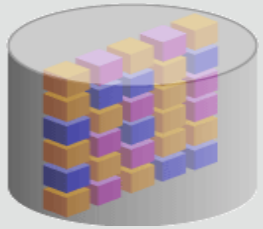


Exadata的资源共享和资源控制



传统的I/O 调度方式

传统的存储，磁盘服务是FIFO方式
IO按照针对磁盘效率的方式进行重新排序
无法根据应用的SLA进行调整其方式



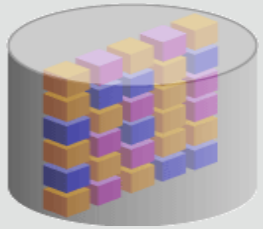
先发生的突发性的报表IO会排在OLTP IO之前
自然也先于OLTP IO被服务

Exadata采用了IORM后的I/O 调度方式

I/O Resource Manager 控制到磁盘的IO顺序

IORM 产生足够的IO保持磁盘的效率和负荷

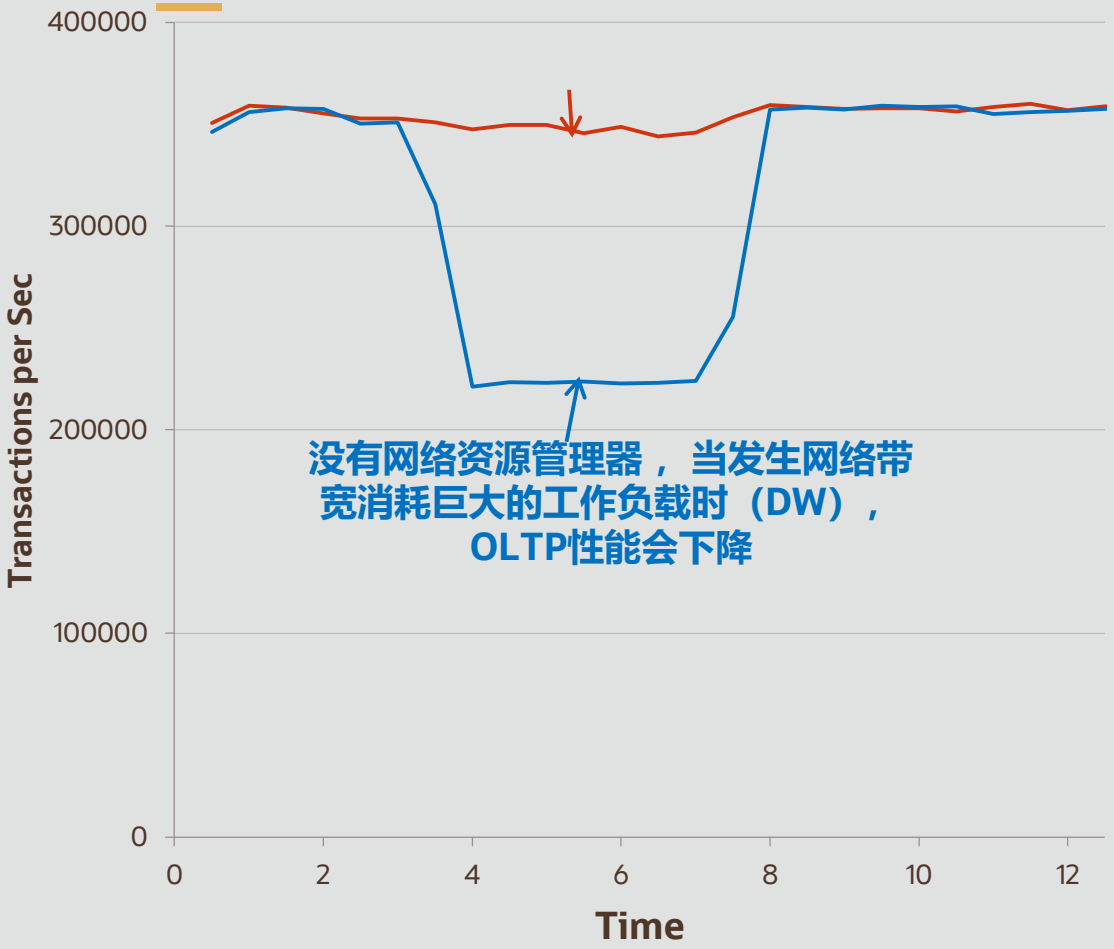
I/Os 根据需要在Exadata内部根据需要排序



- ✓ 通过使用Resource Plan来确定IO请求的顺序
- ✓防止由于某些业务/应用的IO请求过高导致整个数据库SLA无法保证

Exadata 网络资源管理器

Network Resource Management 维持稳定的性能



DB Version 11.2.0.4 or 12c, Switch 2.1.3-4

只有Exadata 网络资源管理器能在整个内部网络优先保证重要的数据库消息传递

- 从数据库到IB卡，再穿过IB交换机到存储的IO消息
- 对于延迟敏感的消息优先于批处理、报表和备份的数据流
- Log文件写有最优先的处理级别以保证OLTP的响应时间

和Exadata的CPU和IO资源管理器结合，以保证多数据库/工作负载整合的性能完全自动化和透明

Exadata: 服务质量保证

带有IO资源管理器的智能存储

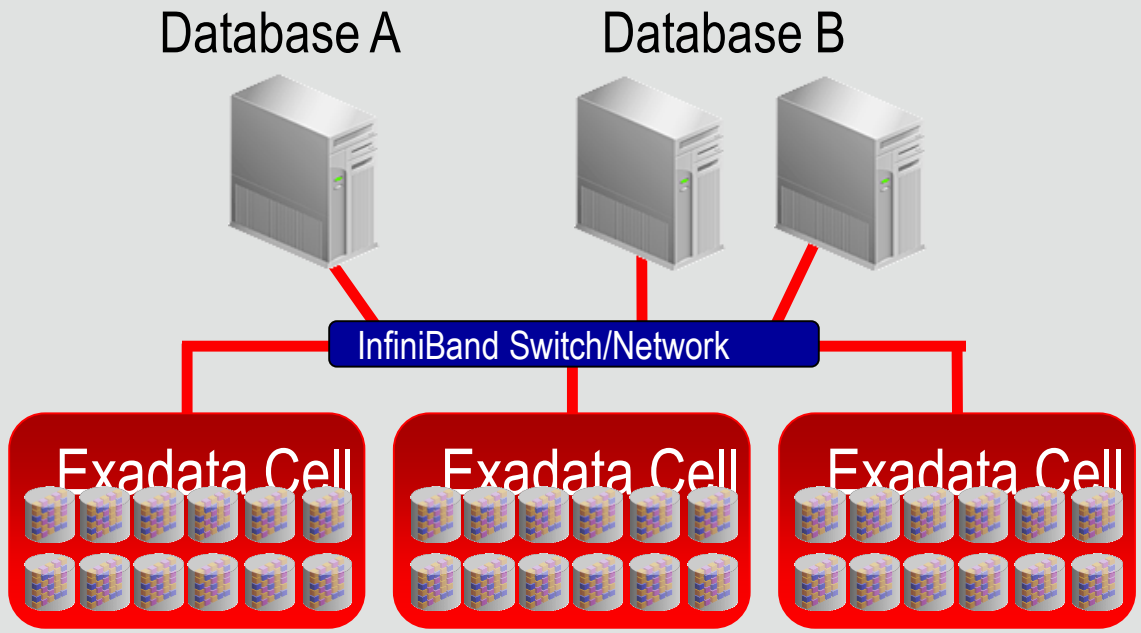
每个IO请求都带标记：谁发起的，目的和优先级

保证在复杂工作负载、多应用整合，高负荷下的性能

IO任务	Exadata存储软件采取的动作
关键数据仓库的表扫描	高优先级，IORM优先于其他全表扫描对其提供服务，同时从flash和磁盘扫描
即席查询表扫描	低优先级，资源消耗大查询。只有flash有空闲空间时提供服务，降低磁盘或flash的IO优先级.
DBWR 写 – 没有 “free buffer wait”	不急： – 大量空闲buffer. IORM降低其IO优先级
DBWR 要解决 “free buffer wait”等待	紧急 – 用户操作被阻止. IORM优先服务此 I/O
LGWR redo 写	高优先级I/O. 通过Exadata Flash Log加速!
OLTP应用的Buffer Cache读IO	中等优先级I/O. 主要通过Flash服务，一般比其他用户IO优先级高，基于IO资源计划调整

Exadata数据库云特点：多租户资源管理

- 确保共享的不同数据库被分配正确的I/O 资源
  - Database A: 33% I/O资源
  - Database B: 67% I/O资源
- 确保库间、库内的不同用户和任务被分配正确的I/O资源
  - Database A:
    - 报表: 60% of I/O资源
    - ETL: 40% of I/O资源
  - Database B:
    - 交互作业: 30% of I/O资源
    - 批处理: 70% of I/O资源



Resource Controls资源控制

什么是保证What is a Guarantee?

ISP必须保证我100 Mbps的带宽

我这次度假至少需要5千块钱

其它PDB不管有多忙，我的PDB也需要至少8

保证用来保证用户的公平份额，保护不受恶邻的骚扰

什么是限制What is a Limit?

ISP限制我的带宽为100 Mbps（即使ISP有大量的带宽资源）

这次度假的花费限制为1万

我的PDB限制为16 CPU，因为这些都是付费的

限制不是为了公平的份额，限制是为了可预测的性能，为性能付费

New in 12.2!

Exadata管理 Flash & Disk I/O's

数据库控制

不再需要去配置一个IORM计划!

根据CPU_COUNT进行自动配置，CPU_COUNT代表了数据库的优先级

第一步：设置IORM的目标为"AUTO"

第二步：对于每个数据库设置CPU_COUNT

共享Shares是根据CPU_COUNT和RAC节点数自动配置的

Database	CPU_COUNT	# RAC Instances	Shares	Guaranteed Disk/Flash Utilization
Support	16	2	16 x 2 = 32	32 / 128 = 25%
Marketing	16	2	16 x 2 = 32	32 / 128 = 25%
Sales	64	1	64	64 / 128 = 50%

需要一个IORM计划去配置闪存利用率和闪存空间的限制

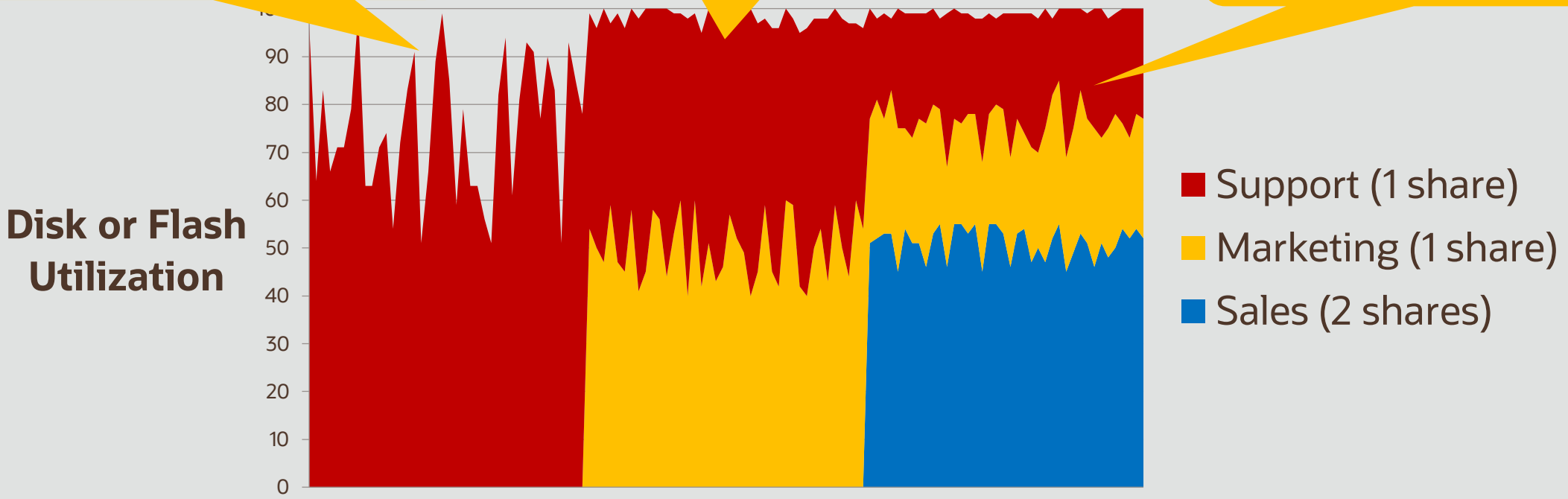
Exadata管理 Flash & Disk I/O's

数据库控制

即使有共享，当只有SUPPORT 单库处于活跃状态时，它会尽可能的所需要的IO资源

当SUPPORT 和MARKETING库都处于活跃状态时，由于它们具有相同的共享数（都是1），因此使用相同比例的disk/flash 的IO资源.

当全部的数据库都激活状态时，它们基于共享的比例使disk/flash等IO资源用.



New in Exadata 19.2!

Exadata管理 Disk and Flash I/O's

数据库控制利用率限制

数据库间的IORM 计划	
Database	**闪存利用率限制**
Support	50%
Marketing	50%
Sales	100% (default)

Exadata 19.2及以上版本，利用率限制不再适用于磁盘disk！

Disk只是I/O的一小部分，并且具有相对较长的延迟.

Disk使用率限制会导致极大的延迟!

Exadata管理闪存空间

数据库控制

数据库的最小保证空间。对于有时不活动的关键数据库很有用。

软限制
仅在闪存已满时应用。

大小限制，并为数据库保留空间。 谨慎使用！

Inter-Database IORM Pla			
Database	Flash Cache Min	Flash Cache Limit	Flash Cache Size
Support	100 GB		
Marketing		2 TB	
Sales			10 TB

在每个存储单元指定设置

对于KEEP数据也同样适用！

IORM优化

最佳实践

如果系统有大量带宽可用，不要指望IORM对优化有什么作用!

- 利用率限制（utilization limit）是例外
- 主要在性能计费的情况下设置利用率限制

对于OLTP系统，主要关注点：

- 闪存命中率
- 平均延迟
- 延迟直方图

对于数据分析负载，主要关注点：

- 闪存命中率
- 闪存和磁盘吞吐量 (MBps)

Exadata IORM监控

New 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

I/O是来自闪存还是磁盘?

每个数据库产生了多少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

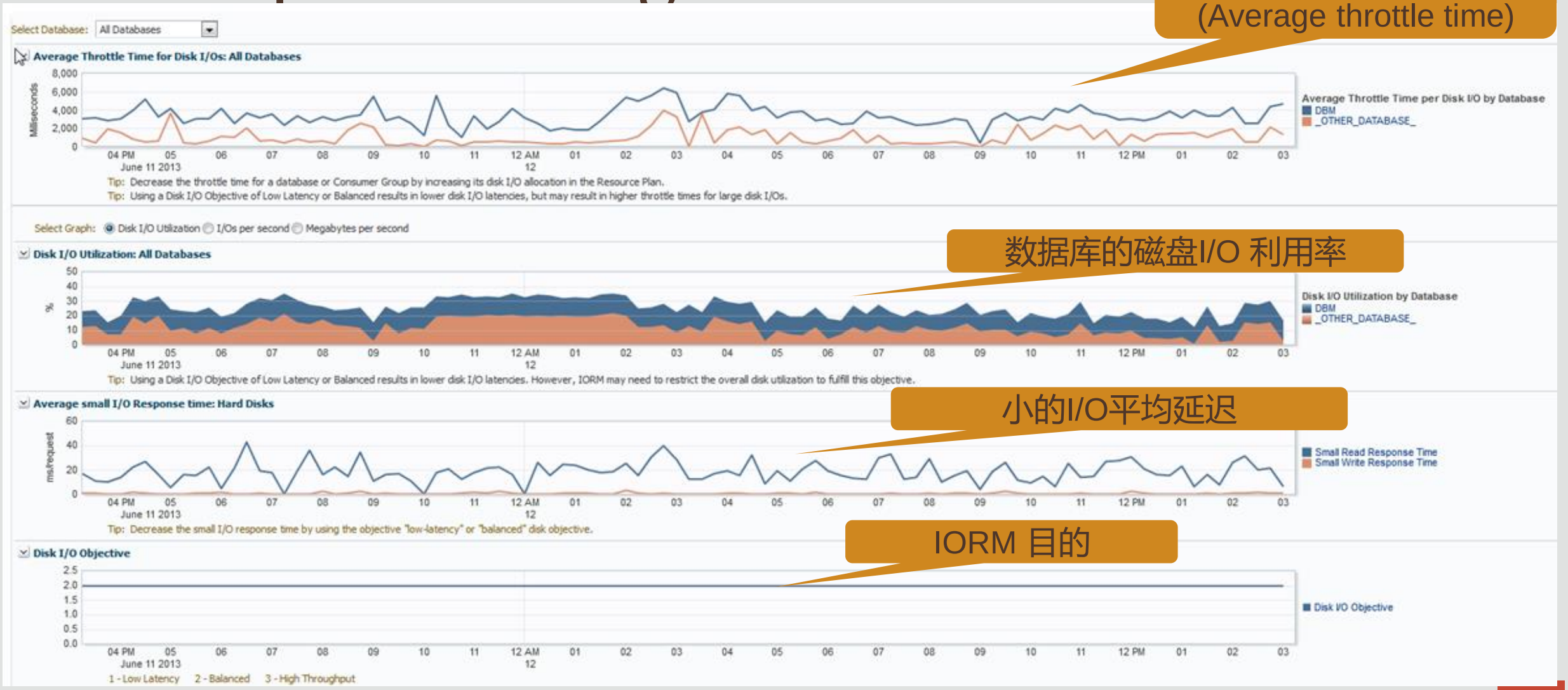
IORM限制了多少I/O?

- Request details for the top databases by IO requests

			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

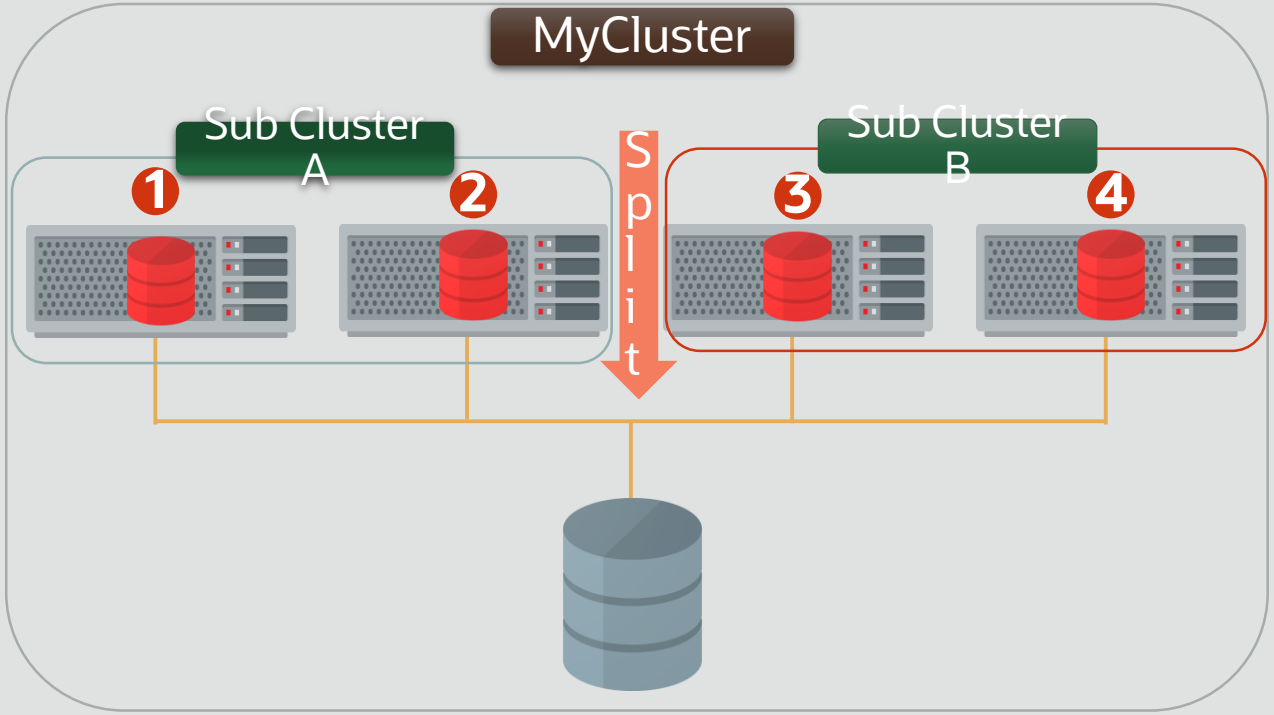
Enterprise Manager IORM UI



1 Exadata 资源管理

2 FDDN，IO latency capping及ASM数据保护技术

RAC节点驱逐基础(通用系统)

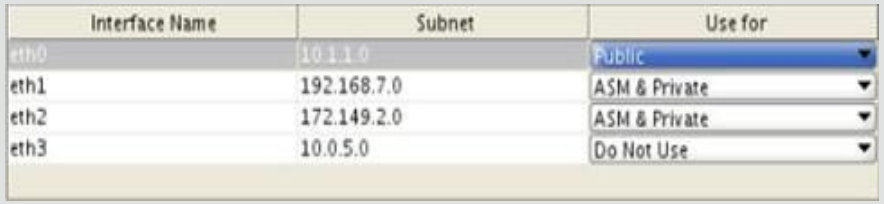


更多详细信息，请查看关于新Oracle RAC 12c第2版节点加权功能的博客文章

节点成员资格取决于

NHB (Network Heart Beat)

```
$oifcfg getif
eth0  XX.XXX.XX.0  global  public
eth1  X0.XXX.0.0     global
cluster_interconnect
eth2  X0.XXX.0.0     global  asm
```



Interface Name	Subnet	Use for
eth0	10.1.1.0	Public
eth1	192.168.7.0	ASM & Private
eth2	172.149.2.0	ASM & Private
eth3	10.0.5.0	Do Not Use

每个HB卡每秒每个接口 (协议：UDP and GIPC for Cache fusion)

PollingThread and SendingThread

Misscount 缺省是30 秒 (升级后是60 秒)

```
$crsctl get css miscount
 CRS-4678: Successful get misscount 30 for Cluster Synchronization
Services
```

网络心跳

2018-03-13 17:00:20.023: [CSSD][4096109472]clssnmSendingThread: sending status msg to all nodes

2018-03-13 17:00:22.818: [CSSD][4106599328]clssnmPollingThread: node anair2 (2) at 50% heartbeat fatal, removal in 14.520 seconds

2018-03-13 17:00:22.818: [CSSD][4106599328]clssnmPollingThread: node anair2 (2) is impending reconfig, flag 132108, misstime 15480

网络心跳发给其它节点
每5条消息输出一次信息

从节点2 (anair2) 来的网络心跳连续15秒(15480 ms)丢失

CSS Logging (私有网络故障)

2018-03-13 17:00:29.833: [CSSD] [4106599328] clssnmPollingThread: node anair (2) at 75% heartbeat fatal, removal in 7.500 seconds

… heartbeat fatal, removal in 2.490 seconds

2018-03-13 17:00:37.337: [CSSD][4106599328] clssnmPollingThread: Removal started for node anair2 (2), flags 0x2040c, state 3, wt4c 0

2018-03-13 17:00:37.340: [CSSD][4085619616] clssnmCheckSplit: Node 2, anair2, is alive, DHB (1281744040, 1396854) more than disk timeout of 27000 after the last NHB (1281744011, 1367154)

从节点2(anair2)的网络心跳丢失

**对节点2(anair2)
启动了节点驱逐程序**

注意：节点2 (anair2)正在更新磁盘(SplitBrain)

磁盘心跳用来解决脑裂(SplitBrain)

Disk Heart Beat aka DHB

```
#crsctl query css votedisk
##  STATE    File Universal Id                          File Name Disk group
1. ONLINE   ee0c34dfb13f4f31bfc36d551f919c96       (AFD:DATA1) [DATA]
2. ONLINE   ee0334dfedu3213b4s3cc33e9303751       (AFD:RECO1) [RECO]
3. ONLINE   ee033dfe58564f32b43g94d683e93f1       (AFD:DATA2) [DATA]
```

每个表决(Voting)文件每秒读一次写一次

对于外部冗余的表决文件，每个节点都有一个写入两个读取

Disktimeout 缺省是200秒

```
 #crsctl get css disktimeout
CRS-4678: Successful get disktimeout 200 for Cluster Synchronization Services.
```

当表决盘的I/O出现问题时CSS logging

2010-08-13 **18:31:19.787**: [SKGFD][4131736480]ERROR: -9(Error 27072, OS Error (Linux Error: 5: Input/output error

2010-08-13 18:31:19.787: [CSSD][4131736480](:CSSNM00060:) clssnmvReadBlocks: read failed at offset 529 of /dev/XXXX

…. *(message repeated)*

2010-08-13 18:31:23.782: [CSSD][150477728] clssnmvDiskOpen: Opening /dev/sdb8

….

2010-08-13 **18:34:37.429**: [CSSD][150477728] (:CSSNM00060:)clssnmvReadBlocks: read failed at offset 17 of /dev/sdb8

写入表决磁盘时发生错误、(OS error 5)

错误消息….重复

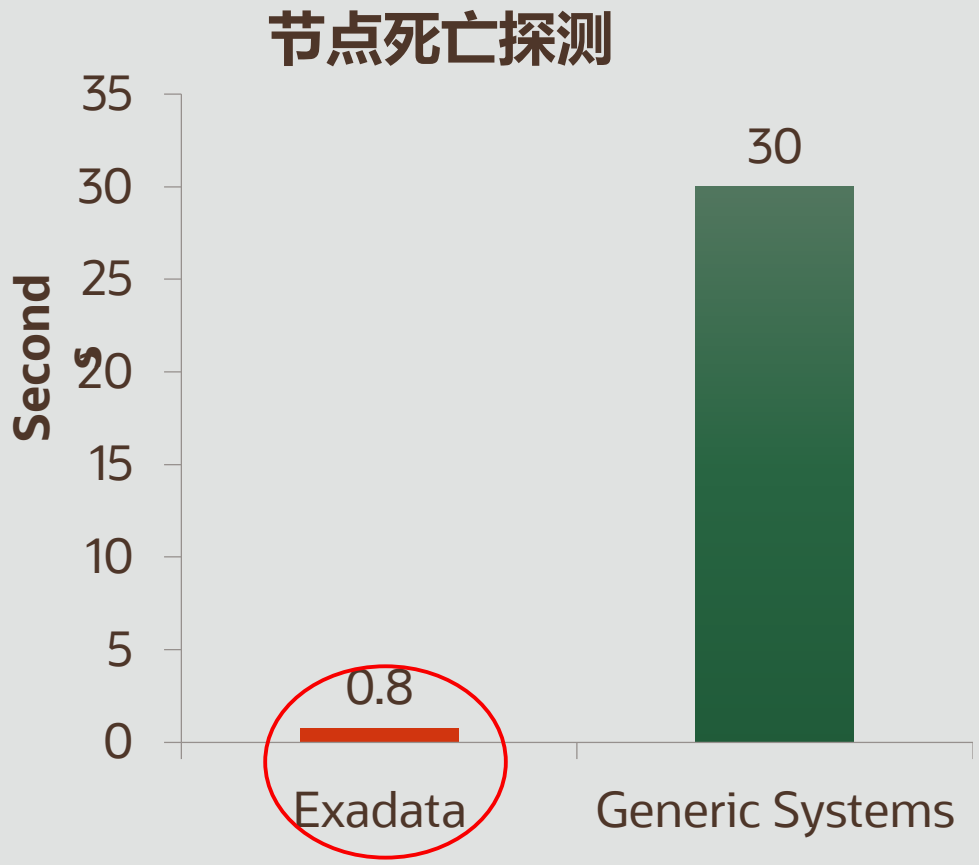
18:31:19.787

18:34:37.429

~200 Seconds

Exadata的快速节点死亡探测FNDD(Fast Node Death detection)

- CSS和网络的subnet manager集成，在不到2秒的时间声明节点死亡
  - 不需要等待漫长的心跳超时
  - 减少了应用的停顿
  - 开箱即用
    *(Subnet manager在switch上运行)
- 通过GIPS协议把节点死亡的故障信息发布到CSS
  - 导致RAC更快的重新配置和应用快速故障切换



Fast Node death detection Logging (Exadata)

2018-02-03 **13:33:27.724** : CSSD:866105088: clssnmeventhndlr: EXADATA NodeDeath detection from GIPC for node xyzadm1 number 1

2018-02-03 13:33:27.724 :GIPCHTHR:988976896: gipchaWorkerProcessClientDelEndp: workerThread set nodeDeath req, hendp 0x7f9c04088f70 [0000000000000bca]

{ gipchaEndpoint : port 'gm2_xyzadXX/ba3d-0156-2b59-0f07', peer 'xyzadm01:cff5-5bd0-d868-2361',

2018-02-03 **13:33:28.457** : CSSD:971667200: clssnmvDiskEvict: Begin: Kill block write, file o/192.168.175.192/DBFS_CD_09_xyzadm1

CSS收到GIPC的节点死亡事件

13:33:27.724

13:33:28.457

00:00:00.733

在不到1秒的时间里写入kill 块并开始驱逐进程

Fast Node Death detection (DHB) ** Diskmon

Exadata上的表决磁盘通过iDB协议：

- 不走常规OS IO路径
- 不依赖操作系统的路径重试

附加的diskmon进程：

- 一个主Diskmon和一个实例级Diskmon从属进程
- 如果Diskmon无法到达存储单元，它将发出CSS信号
- 快速检测DHB丢失

```
2017-09-25 22:37:14.417 [OCSSD(63908)]CRS-
1605: CSSD voting file is online:
o/ZZ.XXX.YYY.9;ZZ.YYY.BB.10/DBFSC3_CD_02_a
dmxxx;
o/ZZ.XXX.YYY.7;
ZZ.YYY.BB.8/DBFSC3_CD_02_admxxx;
o/ZZ.XXX.YYY.5;
ZZ.YYY.BB.6/DBFSC3_CD_02_admxxx;
```

```
ps -ef | egrep "diskmon|dskm" | grep -v grep
oracle    3205     1  0 Feb16 ?        00:01:18
ora_dskm_sales2
oracle   10985     1  0 Feb16 ?        00:32:19
/u01/app/12.2.0/grid/bin/diskmon.bin -d -f
oracle   17542     1  0 Feb16 ?        00:01:17
asm_dskm_+ASM2
oracle   24738     1  0 Feb18 ?        00:00:21
ora_dskm_orcl1
```

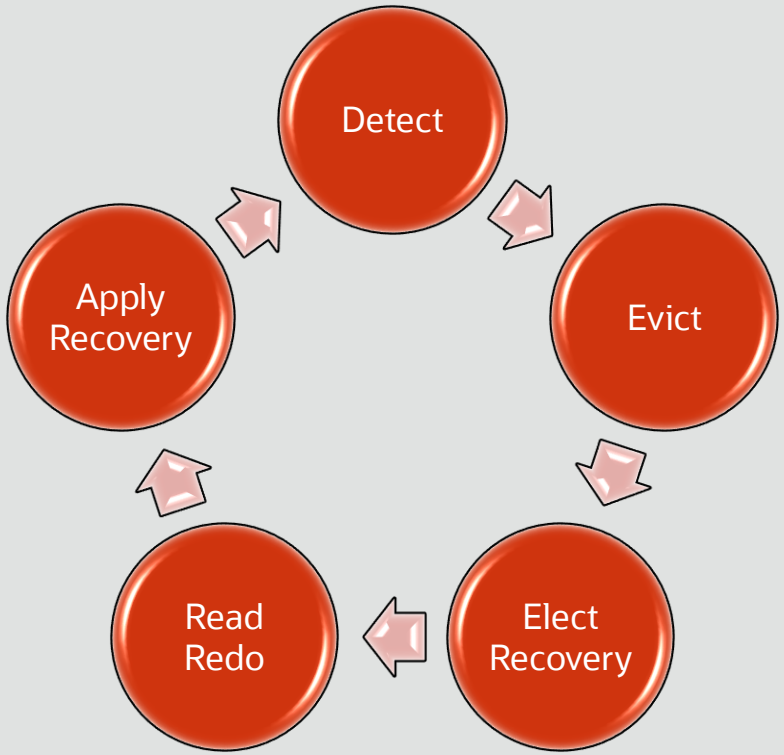
通过恢复伙伴(Recovery Buddy)重新配置接近零DOWN机时间

- Recovery Buddy
  - 跟踪伙伴实例上的块更改
  - 快速识别重新配置期间需要恢复的块
  - 允许快速处理新交易
  - *通过恢复伙伴（Recovery Buddy）和优化的（Singleton）重新配置时间，将速度提高四倍

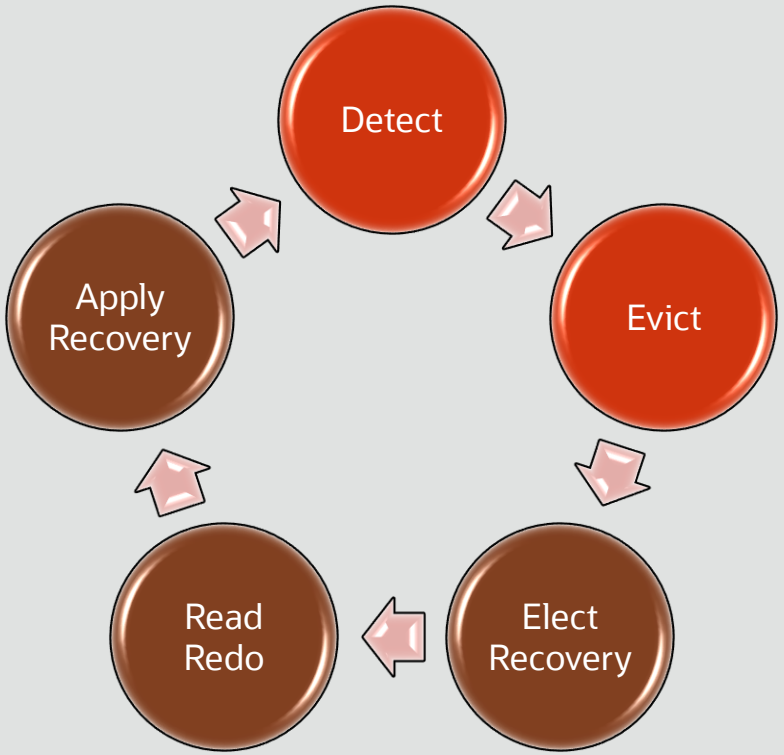
关于Recovery Buddy的更多细节请单击链接阅读blog文章

快速重新配置

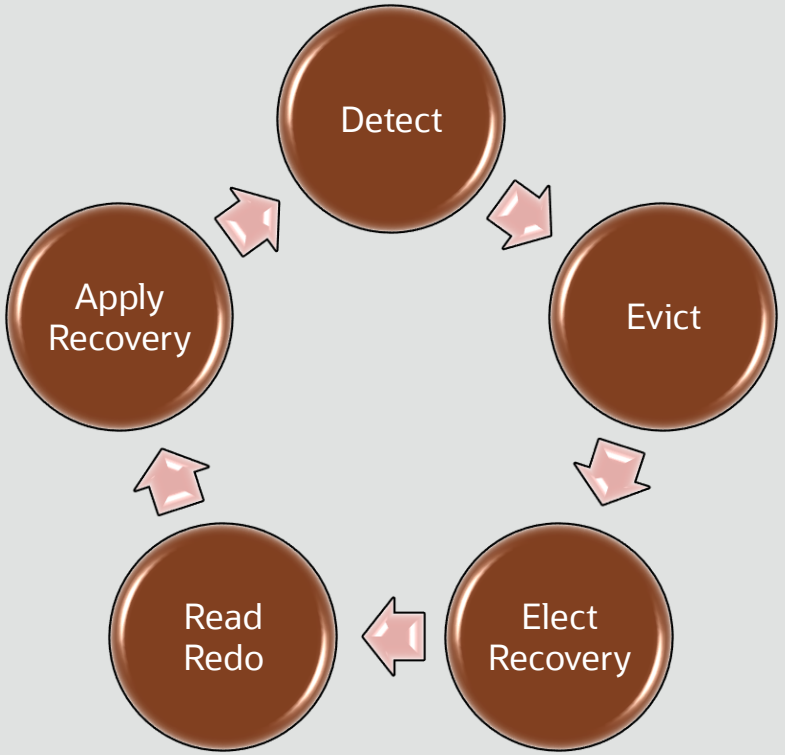
12c



12.2



ORACLE EXADATA



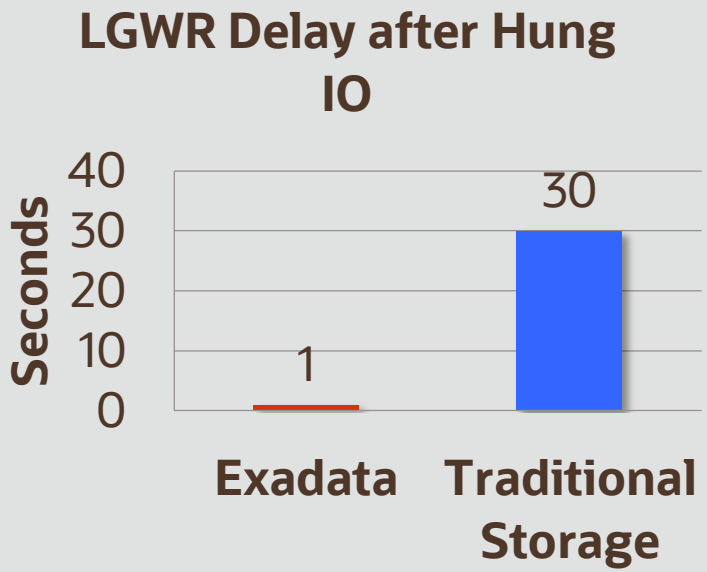
高可用性

- 计算节点与存储服务器的即时故障检测
  - 无需等待漫长的心跳超时
  - 如果某服务器在InfiniBand的两个端口都消失了，系统声明这个服务器死亡
- IO延迟上限(IO latency capping)
  - 当某个存储设备慢时，读取操作被重定向到它的镜像数据，写IO被取消，并临时写到同个存储中的其它正常的flash中

Exadata: 服务质量保证

为了最佳性能

- 存储端IO延迟上限Cell Side IO Latency Capping (Hard Disk & Flash)
  - 当某个存储执行过量的IO时：
    - 读IO被重定向到其它关联的存储
    - 写IO被取消并临时写到同个存储中的正常的flash中
- 存储侧磁盘限制
  - 当某磁盘损坏并脱机时
    - 自动在此磁盘运行诊断，以确定运行状况
    - 如果运行状况良好，则磁盘恢复联机状态并重新同步数据
    - 如果不正常，则执行drop操作，然后数据重新平衡



ASM 与Exadata集成

从存储到存储直接的卸载操作，支持ASM范围内的安全性

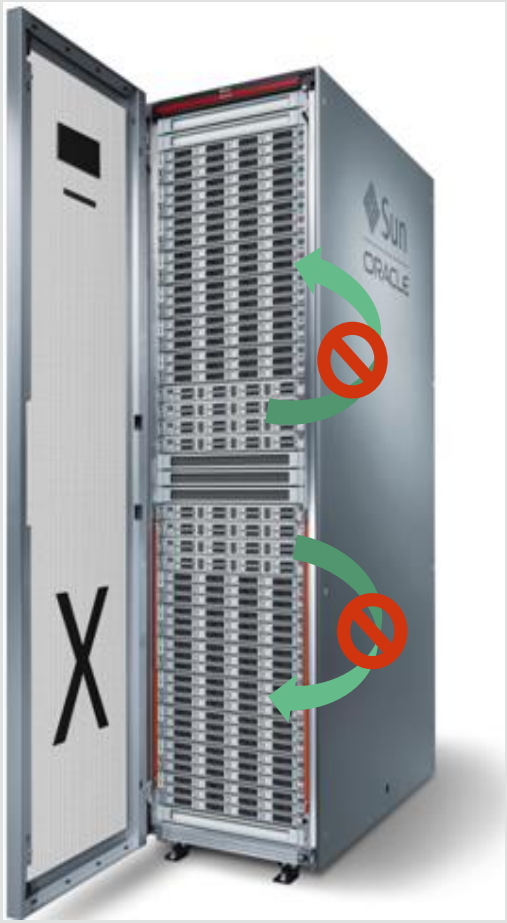
- 适用于ASM的重新同步(re-sync) rebalance, resilver, rebuild和数据库的高吞吐量写操作

智能重定向可实现高吞吐量的写入

- 自动探测并隔离任何异常操作，并使用备用的IO路径

与ASM集成的数据保护

坏块检测，预防与修复



当应用程序更新数据遇到坏块时

数据库从ASM的镜像读取

使用好的拷贝修复坏块

✓此种修复不会对应用程序及数据库其它进程有影响

当数据库服务器与存储之间的IO路径上的网络数据包发生损坏时

存储单元阻止写操作

ASM尝试重新发送这个网络包

✓应用程序不会遇到这种网络包损坏的错误

- 当磁盘空闲时自动进行磁盘检查和修复
- 硬件辅助弹性数据Hardware Assisted Resilient Data (HARD)兼容检查

谢谢