Oracle 最高可用性架构

Oracle 白皮书
2013 年 1 月

Exadata 数据库云服务器上的数据库整合最佳实践

执行概要

当您在目标系统上托管多个模式、应用程序或数据库时，通过整合能够最大限度减少闲置资源并降低成本。整合是在公有云和私有云上部署 Oracle 数据库的核心驱动因素。

本白皮书为您提供 Exadata 数据库云服务器（简称 Exadata）整合最佳实践，从而使您能够设置和管理系统和应用程序以实现最大稳定性和最高可用性。

引言

在 IT 部门中，在组织致力于实现更高运营效率的过程中，整合是一项重要的战略。通过整合，组织可以提高 IT 资源利用率，以便最大限度地缩短空闲周期。相应的，由于能够以更少的资源实现相同的产出，从而可降低成本。例如，在一天中不同时间经受峰值负载的各应用程序可以共享同一硬件，而不是使用专用的、在非高峰期将会空闲的硬件。

根据涉及的系统和环境，可通过多种不同方式实现数据库整合。在单个数据库中运行多个应用程序模式、在单个平台上托管多个数据库或这两种配置的混合都是数据库整合的有效形式。

Exadata 针对 Oracle 数据仓库和 OLTP 数据库负载进行了优化，其平衡的数据库服务器和存储网格基础架构使其成为用于数据库整合的理想平台。Oracle 数据库、存储和网络网格架构与 Oracle 资源管理特性相结合，为您提供一种比其他虚拟化战略（如硬件或操作系统虚拟化）更简单、更灵活的数据库整合方法。Exadata 目前依靠简化 Oracle 资源管理来实现高效数据库整合。但 Exadata 目前尚不支持虚拟机或 Solaris 区域。

本白皮书为您介绍推荐用来在 Exadata 上整合数据库的方法，其中包含必需的初始规划阶段、设置阶段和管理阶段，当系统支持多个应用程序和数据库使用共享资源时，可通过此方法最大限度地提高系统稳定性和可用性。

本白皮书补充完善了其他 Exadata 最佳实践和最高可用性架构 (MAA) 最佳实践，同时也引用了其他适用的相关白皮书。本白皮书不讨论 SPARC Supercluster、Exalogic 或虚拟机上的数据库整合。

Exadata 整合规划

1. 定义高可用性 (HA)、计划维护和业务需求。

   在 Exadata 上进行整合时，应该应用一些核心原则。首先，各目标数据库的可用性和计划维护目标应该类似。如果不是这样，那么不可避免，会对其中的一些应用程序在某些方面产生不利影响。例如，如果任务关键型应用程序与开发系统或测试系统共享同一环境，那么开发和测试系统中频繁的变更会对任务关键型应用程序的可用性和稳定性造成影响。考虑到管理通用基础架构（例如 Oracle 网格基础架构（包括 Oracle Clusterware 和 Oracle ASM）以及 Exadata 存储单元）和运营系统环境的效率和相对简单性，Oracle 将采用整合相似服务级别目标数据库的方法。如果各目标应用程序的服务级别要求并不相似，那么整合的优势将不复存在。

   企业应首先定义以下关键高可用性要求：恢复时间目标（RTO 或应用程序的目标恢复时间）、恢复点目标（RPO 或应用程序的最大数据丢失容限）、灾难恢复 (DR) 的恢复时间以及指定的计划维护时段。

   还必须考虑其他事项，例如性能目标（预计的峰值负载时段、平均负载时段和空闲负载时段）、系统要求、安全性和组织边界，以确保各候选应用程序之间的兼容性。

2. 按类别将数据库分成多个组。

   根据步骤 1 中确定的高可用性要求将计划进行整合的数据库分成组。例如可按类别将数据库分成以下几个组：

   - 关键：与核心业务相关的面向客户的创收型数据库
   - 标准：其他非关键生产数据库
   - 非生产型：开发和测试数据库

   每个组都可进一步细分（如果需要），以使所有应用程序的高可用性要求和计划维护时段不会彼此冲突。

3. 创建 Exadata 硬件池。

   硬件池 一词描述的是用作目标整合平台的一台或一组机器。企业可以创建多个硬件池，以使得每个整合目标平台更易于管理。推荐的最小硬件池为半机架 Exadata X3-2，推荐的最大硬件池为两台全机架 Exadata 数据库云服务器（如果需要，还可以加上

额外 Exadata 存储扩展机架）。此范围的硬件池是最常见的用于整合的 Exadata 配置，它们能够提供足够的容量，可有效实现数据库整合目标。

对于初级整合，还可以选择部署一个只包含四分之一机架或八分之一机架 Exadata X3-2 的小型硬件池。如果在单独的 Exadata 上部署了一个备用数据库，那么对于关键应用程序，上述方法是可接受的。 四分之一机架或八分之一机架 Exadata X3-2 在高可用性方面的一个小缺点是，没有足够的 Exadata 单元可供表决磁盘驻留在任何高冗余磁盘组中。 表决磁盘需要 5 个故障组或 5 个 Exadata 单元，这就是为何我们推荐将半机架 Exadata 作为最小硬件池的一个主要原因。如果因为存储故障卸载了（驻留表决磁盘的）ASM 磁盘组，那么会有 Oracle ASM 和 Oracle Clusterware 出故障的风险。这种情况下，所有数据库都会出现故障，但是可以按照 My Oracle Support (MOS) 说明 1339373.1 中的方法手动重新启动这些数据库（请参阅*“DBFS_DG 丢失产生的影响”一节*）。

您还能够部署一个包含 8 台或更多全机架 Exadata 数据库云服务器的硬件池。但不推荐这样做，因为调度和管理这样一个整合环境的复杂性会抵消整合带来的优势。

通过给节点和 Exadata 存储单元（也称为 Exadata 单元）分区，还可以在一个 Exadata 数据库云服务器上共存多个硬件池。但也不建议使用此配置，因为分区方法可能会导致资源利用效率低下，并有以下缺点：

- 限制对整个服务器和存储网格带宽的访问
- 增加复杂性
- 对于 InfiniBand 结构、Cisco 交换机和物理机架本身等常用组件，缺乏完备的故障和维护隔离机制

4. 将数据库组和应用程序组映射到特定硬件池。

继续上面的示例，每个数据库组和应用程序组都将部署到它们自己的硬件池上：关键、标准和非生产型。不要将不同组的数据库整合到同一硬件池中。如果一个组需要的容量超过了单个硬件池提供的容量，则将该组中的目标数据库分为两个单独的组，然后再将每个组部署在它自己的硬件池中。任何一个数据库，如果它需要的容量超过了两台全机架 Exadata 数据库云服务器和附加 Exadata 存储扩展机架（此为推荐的最大硬件池）提供的容量，那么不应将该数据库视为能从整合获益的候选数据库；如果需要，这种数据库最好部署到由多个 Exadata 系统组成的专用集群上。

下表提供了一个使用 Exadata 硬件池时，不同高可用性要求可能需要哪些不同架构的示例。

表 1. 高可用性要求及其推荐的架构

高可用性要求	推荐的架构
具有以下要求的任务关键型应用程序： • RTO < 5 分钟 • RPO=0 • 每季度有 4 小时的计划维护时间（周末）	三个关键硬件池，其中包含 • 主关键池 • 本地 Data Guard 关键池 • 远程 Data Guard 关键池 • 测试池 • 以 Data Guard 故障切换和应用程序故障切换为基础的 RTO/RPO。
具有以下要求的关键应用程序： • 5 分钟 > RTO < 2 小时 • 2 秒 > RPO < 2 小时 • 每季度有 8 小时的计划维护时间（周末）	两个关键硬件池，其中包含 • 主关键池 • 远程 Data Guard 关键池 • 测试池 • 以 Data Guard 故障切换和应用程序故障切换为基础的 RTO/RPO。 应用程序故障切换占用大多数时间。
具有以下要求的标准应用程序： • 12 小时 > RTO < 24 小时 • 12 小时 > RPO < 24 小时 • 每季度有 48 小时的计划维护时间	一个标准硬件池： • STDPOOL1 • 以从备份还原和恢复为基础的 RTO/RPO。 备份频率和还原速率影响 RTO 和 RPO。
具有以下要求的开发应用程序： • 供开发使用的高可用性 • 24 小时 > RTO < 72 小时 • 每月有 48 小时的计划维护时间	一个非生产型硬件池： • DEVPOOL1 • 以从生产备份还原和恢复为基础的 RTO/RPO。 备份频率和还原速率影响 RTO 和 RPO。
具有以下要求的测试应用程序： • 测试系统应验证系统更改和补丁，并评估新功能、性能和高可用性 • 这是针对所有生产应用程序的建议	建议与生产环境相同 或最小为 • TESTPOOL1

5. 在迁移到 Exadata 硬件池之前评估大小要求。

   如果您正在从现有系统迁移数据库，那么可根据业务推断当前 CPU、I/O 和内存的利用率，并获得未来增长预测。然后您可以利用这些计算结果，评估可将多少数据库和

应用程序合理地安装到一个硬件池中。有关大小的详细信息，请联系 Oracle 咨询部门和 Oracle 高级客户支持服务 (ACS) 部门获取。

其他考虑事项包括：

- 预留系统容量，以供各种高可用性 (HA) 和滚动升级活动使用，例如 Oracle Real Application Clusters (Oracle RAC) 和 Exadata 单元滚动升级活动。
- 为关键硬件池预留系统容量，以实现最高稳定性。例如，常用的最佳实践是，将关键硬件池配置为有 25% 的未分配资源，以适应高峰期负载。
- 请记住，如果关键硬件池使用 Oracle Automatic Storage Management (ASM) 高冗余磁盘组，则可用的数据存储空间会更小。
- 评估应用程序之间及数据库之间的业务或负载周期是否允许进一步整合。例如，应用程序 A 可能会在应用程序 B 相对空闲的时候拥有峰值负载。

6. 收集每个应用程序和数据库的准确性能要求。

   收集每个应用程序对吞吐量和响应时间的准确性能预期。使用 Oracle Enterprise Manager (EM) Grid Control 或 EM Cloud Control 监视关键应用程序量度，包括应用程序、数据库和系统性能统计信息的历史记录。在调试任何未来性能问题时会需要这些数据。

设置和管理关键资源以实现稳定性

完成初始规划和大小调整活动后，将转到设置 Exadata 硬件池阶段。本节为您提供与在 Exadata 上整合数据库相关的建议，从初始部署到特定配置设置都涵盖其中。

推荐的存储网格（磁盘组）配置

推荐的存储配置是每个硬件池配有**一个共享 Exadata 存储网格。**此存储网格包含所有 Exadata 单元和 Exadata 磁盘，并配置为具备 ASM 高冗余或常规冗余（将在后面进一步讨论 ASM 冗余）。

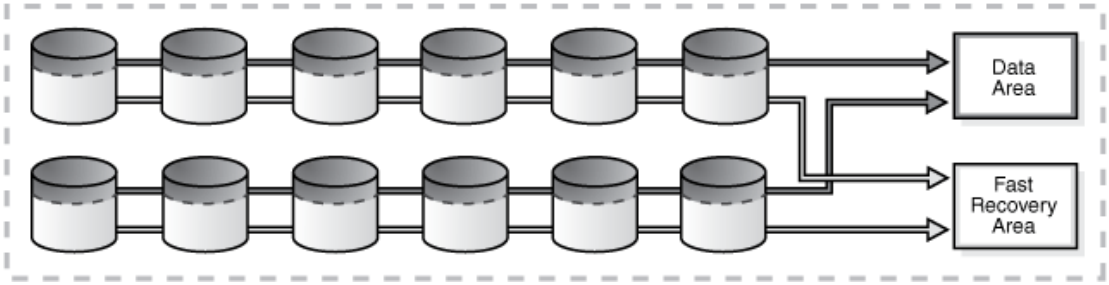

图 1. 共享 Exadata 存储网格

主要优势是：

- 管理简单且方便
- 是最常用的经过充分验证的配置
- 平衡配置，通过该配置，应用程序拥有对 I/O 带宽和存储的完全访问权限
- 支持容错和滚动升级

管理一个共享的 Exadata 存储网格比较简单，只需很少的管理成本。通过共享存储，空间和带宽的利用也更高效。如果存储空间和 I/O 带宽需求超过一个全机架 Exadata 所提供的容量，您可以添加一个 Exadata 存储扩展机架，也可以将应用程序迁移到其他硬件池。

如果该硬件池是关键硬件池，则强烈建议针对 DATA 和 RECO 磁盘组使用 ASM 高冗余，从而提供 Exadata 存储单元滚动升级和其他维护活动期间特殊需要的对存储故障的最佳容错能力。有关 DATA 和 RECO 磁盘组配置、MAA 存储网格配置以及四分之一和八分之一机架 Exadata X2-2 或 X3-2 限制的详细信息，请参阅 *Oracle Exadata 存储服务器软件用户指南* 中的“关于通过 Oracle ASM 实现最高可用性”。当您的空间受限时，如果还部署了一个使用 Oracle Data Guard 的备用硬件池（备用池），那么可考虑针对 DATA 和 RECO 磁盘组使用 ASM 常规冗余。使用备用池，在数据库、集群或存储出现故障时，可实现最全面的数据损坏保护及快速故障切换，从而降低未在主硬件池上使用高冗余所带来的高可用性和数据保护方面的风险。

如果您的空间受限，而且又无法部署备用池，那么还有第二种方法，即针对 DATA 磁盘组使用高冗余，而针对 RECO 磁盘组使用常规冗余。请注意，在发生存储故障时，此方法可能会影响可用性和简便性。有关在丢失 DATA 或 RECO ASM 磁盘组时应如何进行恢复的详细信息，请参阅 My Oracle Support (MOS) 说明 1339373.1。

您应该使用 OneCommand 配置过程中的磁盘组配置（在 *Exadata 数据库云服务器所有者指南* 中有所介绍）。默认情况下，第一个高冗余磁盘组存储联机日志、控制文件、服务器参数文件、集群件和表决设备。DATA 磁盘组存储数据库文件，而 RECO 磁盘组则包含针对快速恢复区 (FRA) 的与恢复相关的文件。 如果要求隔离应用程序，可创建单独的 DATA 和 RECO 磁盘组。请参阅本文后面第 26 页上的“安全性和管理角色”一节或 Oracle Exadata 数据库云服务器整合：分离数据库和角色 MAA 白皮书。

此外，在创建 ASM 磁盘组时，确保将 ASM 磁盘组的 COMPATIBLE.RDBMS 属性设置为硬件池中 Oracle 数据库软件的最低版本。

推荐的数据库网格（集群）配置

推荐的 Oracle 网格基础架构（包括 Oracle Clusterware 和 Oracle ASM）设置是**每个硬件池使用一个集群**。应使用由 Oracle Clusterware 管理的 Oracle 数据库服务来进一步对应用程序进行负载平衡，从而将应用程序路由到集群中特定的 Oracle 数据库实例上。主要优势有：

- 管理简单且方便
- 平衡配置，通过该配置，应用程序拥有对数据库 CPU 和内存带宽的完全访问权（如果需要）
- 支持容错功能
- 支持 RAC 和网格基础架构滚动升级功能

当特定应用程序资源增长或收缩时，可方便、透明地对服务进行负载平衡并路由到可用的数据库实例和节点上。

Oracle Grid Infrastructure 软件的版本必须等于或高于您计划在集群中使用的 Oracle RAC 软件的最高版本。虽然可随时升级 Grid Infrastructure 软件，但是如果各数据库按不同时间表升级，则需要进行规划以降低修补 Grid Infrastructure 的频率。所有 Grid Infrastructure 升级应是滚动升级。

应为关键硬件池分配一个 Data Guard 硬件池（备用池），以避免其受到集群故障、数据库故障、数据损坏和灾难的影响。您还可以切换到备用池以进行站点、系统或数据库维护。

> **注意：**对于 Oracle 11*g* 第 1 版以及更高版本，每个集群中数据库实例的数量最多为 512。建议将每个 X2-2 或 X3-2 数据库节点的数据库实例数量的上限限定为 128，而将每个 X2-8 或 X3-8 数据库节点的数据库实例数量的上限限定为 256。每个数据库节点或集群的实际数据库实例数量取决于应用程序负载及其相应的系统资源占用。

推荐的参数设置

以下参数设置对于每种硬件池都尤为重要。它们有助于高效地分配系统资源和限制系统资源的使用。随着负载的变化或数据库的添加或弃用，需要定期进行监视以调整和调优这些参数设置。

操作系统参数

- **如果 /proc/meminfo 中的 PageTables 设置为大于物理内存的 2%，则将操作系统参数 HugePages 设置为所有共享内存段的总和。（仅限 LINUX）**

  HugePages 是一种允许 Linux 内核使用多种页面大小功能的机制。Linux 将页作为内存的基本单元，按基本页单元划分和访问物理内存。默认页面大小是 4096 字节。Hugepages 支持以低开销使用大量内存。Linux 使用 CPU 架构中的转换旁路缓冲区

(TLB)。这些缓冲区中包含了虚拟内存到实际物理内存地址的映射。因此具有较大页面大小的系统可提供更高密度的 TLB 条目，从而减少了页表空间。 此外，每个进程使用较小的私有页表，而随着进程数量的不断增多，由此实现的内存节省将非常明显。

如果 /proc/meminfo 中的 PageTables 大于物理内存的 2%，则通常需要设置 HugePages。设置时，HugePages 应等于所有数据库实例使用的共享内存段的总和。当所有数据库实例都在运行时，可通过分析 *ipcs -m* 命令的输出来计算被使用的共享内存的数量。MOS 说明 401749.1 提供了一个可用于确定在用共享内存数量的脚本。MOS 说明 361468.1 介绍如何设置 HugePages。

从版本 11.2.0.2 开始，在所有实例上设置数据库初始化参数 USE_LARGE_PAGES=ONLY，可防止在没有足够大页面的情况下启动实例。

> **注意：**如果数据库参数 SGA_TARGET 有变化，或每当数据库实例数量变化时，都必须重新计算要使用的 HugePages 的值。Hugepages 只可用于 SGA，因此不要过度分配。此外，如果启用了 HugePages，则数据库参数 MEMORY_MAX_TARGET 和 MEMORY_TARGET 互不兼容。
>
> **注意：**在 Solaris 上，通过锁定共享内存 (ISM) 自动配置和使用 HugePages。

- **将共享内存段的数量 (kernel.shmmni) 设置为大于数据库的数量。**

  应将共享内存标识符的数量设置为大于节点上运行的数据库实例的数量。可使用 sysctl 命令（如，/sbin/sysctl -a | grep shmmni）检查 kernel.shmmni 的设置。如果需要，可在 Linux 系统上通过设置 /etc/sysctl.conf 中的 kernel.shmmni 来进行调整。SHMMNI 的默认设置 (4096) 应能满足所有情况的需求。

- **将最大共享内存段大小 (kernel.shmmax) 设置为物理内存大小的 85%，这是默认设置**

  应将最大共享内存段大小设置为数据库服务器物理内存大小的 85%。使用 sysctl 命令检查 kernel.shmmax 的设置。如果需要，可在 Linux 系统上通过设置 /etc/sysctl.conf 中的 kernel.SHMMAX 来进行调整。

- **将系统信号的最大总数 (SEMMNS) 设置为大于所有数据库进程的总数。**

  系统中信号的最大数量必须大于系统上运行的所有数据库实例的进程的总数（如数据库参数 PROCESSES 所指定的数值）。可使用 sysctl 命令检查 kernel.sem 的设置。kernel.sem 的设置包含一个由 4 个数字值组成的列表。系统上信号总数的设置，

也称为 SEMMNS，是列表中的第二个值。如果需要，通过设置 /etc/sysctl.conf 中的 kernel.sem 进行调整。 SEMMNS 的默认设置 (60000) 应能满足所有情况的需求。

- **将一个信号集内的最大信号数量 (SEMMSL) 设置为大于任何一个数据库内的最大进程数量。**

  应将一个信号集内的最大信号数量设置为大于数据库服务器上运行的任何一个数据库实例的最大进程数量。使用 sysctl 命令检查 kernel.sem 的设置。一个信号集内最大信号数量的设置，也称为 SEMMSL，是上述列表中的第一个值。如果需要，可在 Linux 系统上通过设置 /etc/sysctl.conf 中的 kernel.sem 来进行调整。

数据库内存设置

Exadata 系统提供以下物理内存：

- 基于 Sun Fire X4170 Oracle Database Server 的 Exadata X2-2（也称为 V2）的每台数据库服务器配有 72 GB 内存
- Exadata X2-2 默认配置配有 96 GB 内存，可以选择扩展为 144 GB 内存（通过 Exadata 内存扩展工具包实现）。
- Exadata X3-2 配有 256 GB 内存。
- Exadata X2-8 的每台数据库服务器配有 1 TB 内存（对于 X4800）或 2 TB 内存（对于 X4800M2）。
- Exadata X3-8 的每台数据库服务器配有 2 TB 内存。

应始终避免过度的内存分页（或称“页出”），因为这会导致应用程序性能变差及节点不稳定。应定期监视系统内存以查看是否有页出情况。

注意：

- 使用 Linux *free* 命令显示系统上的空闲内存和已用内存信息。
- 使用 *vmstat* 命令监视页出值是否为零或较低。
- 使用操作系统 *ps top* (Linux) 或 *prstat* (Solaris) 命令查看更多进程级信息，包括私有和共享内存利用率。

  如果 PGA_USED_MEM 超过应用程序的阈值，针对 V$PROCESS 的数据库查询会向您发出警报。在某些情况下，您可能需要终止进程以防止因节点不稳定而影响到该数据库节点上的所有数据库和应用程序。 例如，任何进程，如果其 OLTP 占用的 PGA 内存超过 1 GB，以及数据仓库报告占用的内存超过 10 GB，则终止掉该进程是合理的。确切的阈值取决于已知的应用程序行为和要求，但是应小于 PGA_AGGREGATE_TARGET。

  下面是一个查询示例和操作流程：

```
SELECT s.inst_id, s.sid, s.serial#, p.spid, s.username, s.program, p.pga_used_mem
  FROM  gv$session s JOIN gv$process p ON p.addr = s.paddr AND p.inst_id = s.inst_id
  WHERE  s.type != 'BACKGROUND'  and s.program not like '%(P%'   and

  p.pga_used_mem > <APPLICATION_MEMORY_THRESHOLD>
  order by s.inst_id, s.sid, s.serial#;
```

每 15 分钟执行一次上述查询，用应用程序内存阈值的数值替代 <APPLICATION_MEMORY_THRESHOLD>。

调查任何正以不公平方式占用内存的可疑进程。如果此元凶进程占用的内存超过了内存阈值，且该进程不是关键进程，则有必要终止该进程以维护节点稳定性和其他客户端的性能，这点也许非常重要。例如，您可以使用 SQL 语句 ALTER SYSTEM KILL SESSION '<S.SID>,<S.SERIAL#>,@<S.INST_ID>' 终止该会话，如果万不得已，也可以使用操作系统命令 KILL -9 <SPID>。

对于初始部署和将应用程序迁移到硬件池的情况，请使用以下公式，确保为 SGA、PGA 和各服务器进程提供足够内存。如果某个应用程序需要更多 SGA 或 PGA 内存分配以满足其性能需求，而又没有足够物理内存，则可利用另一个硬件池。对关键硬件池而言，我们建议采用更保守的方法，即不要超过每个数据库节点的物理内存的 75%。

OLTP 应用程序：

数据库的总数 (SGA_TARGET + PGA_AGGREGATE_TARGET) + 4 MB *(最大进程数) < 每个数据库节点的物理内存

DW/BI 应用程序：

数据库的总数 (SGA_TARGET + 3 * PGA_AGGREGATE_TARGET) < 每个数据库节点的物理内存

注意：

每个进程使用的内存会不同，不过在我们针对 Exadata 进行的应用程序 MAA 研究中，观测到内存分配为 4 MB。在监视到实际的进程内存使用情况后，可在您的计算中重新调整此值。

应持续监视自动负载信息库 (AWR) 报告中的内存统计信息，以获得最准确的计算。例如，下表取自描绘 PGA 增长的 AWR 报告。

表 2. PGA 内存

COMPONENT	BEGIN SNAP SIZE (MB)	CURRENT SIZE (MB)	MIN SIZE (MB)	MAX SIZE (MB)	OPER COUNT	LAST OP TYP/MOD
PGA Target	16,384.00	**16,384.00**	**16,384.00**	**29,384.00**	0	STA/

表 3. 内存统计

	BEGIN	END
Host Mem (MB)	96,531.5	96,531.5
SGA Use (MB)	24,576.0	24,576.0
PGA Use (MB)	577.7	578.2
% Host Mem used for SGA+PGA	26.06	26.06

从上例可观测到，PGA_AGGREGATE_TARGET 不是一个硬性限制。在一些数据仓库应用程序中，已观测到累积的 PGA 内存使用是 PGA_AGGREGATE_TARGET 的三倍。对于 OLTP 应用程序，内存使用超出的可能性通常非常低，除非应用程序使用大量 PL/SQL。查询 V$PGASTAT 提供 PGA 内存使用的实例级统计信息，其中包括：当前目标设置、内存使用总计、已分配内存总计以及分配的最大内存。如果观测到 PGA 增长，您必须确定此增长是否是所预期的，且数据库服务器上是否有足够内存可用于满足此增长的需求。如果答案是否定的，您可能需要提高 PGA_AGGREGATE_TARGET 设置并降低 SGA_TARGET 设置，以满足更大 PGA 内存分配的要求，或将应用程序迁移到其他数据库服务器或硬件池以从根本上解决此问题。

数据库 CPU 设置和实例囚笼

实例囚笼是一个重要的工具，用于管理和限制每个数据库对 CPU 的使用。通过实例囚笼，可防止失控进程和高应用程序负载产生非常高的系统负载，从而导致系统不稳定或导致其他数据库的性能变差。

要启用实例囚笼，请为服务器上的每个实例执行以下操作：

1. 通过为 RESOURCE_MANAGER_PLAN 初始化参数分配一个资源计划来启用 Oracle Database Resource Manager。此资源计划必须包含 CPU 指令才能启用实例囚笼。有关说明，请参阅 Oracle Database 11g 第 2 版管理指南中的“启用 Oracle Database Resource Manager 和切换计划”。如果您不打算管理数据库内的负载，只需将 RESOURCE_MANAGER_PLAN 设置为“DEFAULT_PLAN”即可。

```
SQL> ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'DEFAULT_PLAN'
SID='*' SCOPE='BOTH';
```

2. 将 CPU_COUNT 初始化参数设置为实例随时会使用的最多 CPU 的个数。默认情况下，将 CPU_COUNT 设置为服务器 CPU 的总个数。对于超线程 CPU，CPU_COUNT 包含 CPU 线程数。CPU_COUNT 是一个动态参数，因此其值会随时变化，但是最好在启动实例时就设置此参数，因为它会影响其他 Oracle 参数和内部结构（例如 PARALLEL_MAX_SERVERS、缓冲区缓存以及闩锁结构分配）。

服务器可能的峰值负载等于该服务器上所有数据库实例的 CPU_COUNT 参数的总和。通过将可能的峰值负载与每台服务器的 CPU 个数进行比较，可评估数据库实例间潜在的资源争用情况。Exadata 的每台服务器配有的 CPU 数量如下所述。请注意，这些计数取决于 CPU 线程数，而不取决于内核数或插槽数。

- Exadata V2 的每台数据库服务器配有 8 个内核或 16 个 CPU。
- Exadata X2-2 的每台数据库服务器配有 12 个内核或 24 个 CPU。
- Exadata X3-2 的每台数据库服务器配有 16 个内核或 32 个 CPU。
- Exadata X2-8 的每台 X4800 数据库服务器配有 64 个内核或 128 个 CPU。Exadata X2-8 的每台 X4800M2 数据库服务器配有 80 个内核或 160 个 CPU。
- Exadata X3-8 的每台数据库服务器配有 80 个内核或 160 个 CPU。

配置实例囚笼最重要的步骤是确定每个数据库实例的 CPU_COUNT 值以及整个服务器的 CPU_COUNT 值的总和。有以下两种方法：

- **针对任务关键型硬件池采用分区方法。**如果目标数据库服务器上所有数据库实例的 CPU_COUNT 值的总和不超过该服务器的 CPU 的个数，则服务器已自然分区。这种情况下，数据库实例间应不存在 CPU 资源争用。但是如此一来，即使某个数据库实例未使用分配给它的 CPU 资源，其他数据库实例也无法使用这些 CPU 资源。

  分区方法的优势是不存在 CPU 争用，但是有可能 CPU 未得到充分利用。因此，建议将此分区方法用于关键硬件池中的任务关键型数据库。

  建议使用以下特定大小调整方法：通过将 CPU_COUNT 值的总和限制为小于服务器 CPU 总数的 75%，使得 CPU 资源可用于其他进程（PMON、SMON、LMS、LMON、LGWR 等）。此外，由于 Exadata 使用超线程 CPU，因此上述限制能够最大限度减少共享一个内核的各 CPU 线程之间的资源争用。

(CPU_COUNT) 总数 < CPU 总数 * 75%

- **针对非关键硬件池采用过度使用方法。**如果目标数据库服务器上所有数据库实例的 CPU_COUNT 值的总和超过服务器 CPU 的数量，则服务器被过渡使用。这种情况下，如果同时加载所有数据库则会超载，将导致 CPU 资源争用和性能降低。

  过渡使用方法的优势是能够更好地利用资源，但是存在潜在的 CPU 争用现象。因此建议将此过渡使用方法用于测试硬件池或峰值时段不重叠的非关键数据库硬件池。

  建议将 CPU_COUNT 值的总和限制为不超过 CPU 数量的三倍。

  (CPU_COUNT) 总和 <= 3 * CPU 总数

  例如，使用上述公式，目标数据库服务器（配有 32 个 CPU 的 X3-2）上所有数据库实例的 CPU_COUNT 值的总和最大为 3 * 32 = 96。因此您可以有 4 个数据库实例，每个实例的 CPU_COUNT 设置为 24，也可以有 8 个实例，每个实例的 CPU_COUNT 设置为 12，或将每个实例的 CPU_COUNT 设置为不同值，但是其总和应小于等于 96。使用实例囚笼时，此 CPU_COUNT 值可确保任何一个数据库实例使用的 CPU 数量都不会超过为其指定的数量。

如果配置了实例囚笼，那么您可以使用 MOS 说明 1362445.1 中的脚本来监视每个实例实际使用的 CPU。如果需要，可根据获得的实际使用信息，通过调整每个实例的 CPU_COUNT 值来对实例囚笼进行调优。对于非常活跃的 Oracle RAC 系统，您应为每个数据库实例至少分配 2 个 CPU，以便后台进程（例如 SMON、PMON、LMS 等）能够持续高效运行。有关实例囚笼推荐补丁的信息，也请参阅 MOS 说明 1340172.1。

进程设置

以下 Oracle ASM 和 Oracle 数据库进程设置和操作实践可帮助您避免常见的进程配置错误。

- **针对 Oracle ASM 实例，请进行如下设置，PROCESSES = 50 MIN（数据库节点上的数据库实例数 + 1, 11）+ 10 MAX（数据库节点上的数据库实例数 – 10, 0）**

  如果每个数据库节点有多个实例，或如果添加或删除了实例，则需要调整 Oracle ASM 初始化参数 PROCESSES。通过遵循上述指导，在尝试请求 ASM 分配以及请求取消分配时，可避免 Oracle ASM 出现错误。通过上述公式可得出，5 个数据库实例需要 300 个 Oracle ASM 进程，而 20 个数据库实例需要 650 个 Oracle ASM 进程。

- **限制 PARALLEL_MAX_SERVERS**
  - X2-2 和 X3-2：所有实例的 (PARALLEL_MAX_SERVERS) 总和 <= 240

- X2-8 和 X3-8：所有实例的 (PARALLEL_MAX_SERVERS) 总和 <= 1280

PARALLEL_MAX_SERVERS 指定一个实例的并行执行进程和并行恢复进程的最大数量。随着需求的增长，Oracle 数据库可将并行执行进程的数量增加至 PARALLEL_MAX_SERVERS。确保以较低的值就能满足每个应用程序的性能要求。如果 PARALLEL_MAX_SERVERS 设置得过高，在高峰期可能会出现内存资源短缺现象，这会导致应用程序性能降低，并使数据库节点变得不稳定。

- **限制重做应用的并行度**

对于 Oracle Data Guard，默认的重做应用并行度被隐式设置为 CPU 的数量，这样做会造成不必要的系统资源消耗，而收效甚微。与之相对，可通过以下命令显式地将恢复并行度仅设置为 16（或更小值）：

SQL> RECOVER MANAGED STANDBY DATABASE … PARALLEL 16。

- **限制进程数量以及到数据库服务器的连接数**

拥有较低的进程数或合适的进程数会带来很多优势，例如可避免或减少内存、CPU 和数据库闩锁争用，缩短*日志文件同步*等待时间以及应用程序故障切换的总时间。进程数和连接数的减少还会提高性能和吞吐量。请参见“实际性能”培训视频，以了解进程数和连接数与性能之间的相关性，网址是：
http://www.youtube.com/watch?v=xNDnVOCdvQ0

Oracle 性能专家建议，可保守地将**活动**进程数设置为 CPU 内核数的 5 倍，或者，可将 Siebel 等 CPU 密集型应用程序的活动进程数设置为 CPU 内核数的 1 倍或 2 倍。通过使用以下一项或多项技术，可以降低进程总数，从而提高性能：

1. 具有最小连接数设置和最大连接数设置的**应用程序连接池**

   通过使用应用程序连接池，可以避免代价高昂的打开和关闭连接的过程，从而节省宝贵的系统资源。通过使用连接池中的最小连接数设置和最大连接数设置，可以实现快速的响应速度、高效的内存和 CPU 使用，并能避免登录风暴。该技术可显著减少连接的总数，而不会对任何性能目标产生影响。

[1] 通过查询 V$SESSION where STATUS=”ACTIVE” 或监视 V$ACTIVE_SESSION_HISTORY，可获得活动的进程数量。Enterprise Manager 的性能页提供与此相关的详细信息以及对任何潜在争用的有益见解。

连接池是数据库连接对象的缓存。这些对象表示可被应用程序用来连接到数据库的物理数据库连接。在运行时，应用程序从连接池请求连接。如果连接池包含满足此请求的连接，则将该连接返回给应用程序。如果找不到可用的连接，则创建一个新连接，并将它返回给应用程序。应用程序使用此连接在数据库上执行一些任务，完成任务后将此连接对象返回给连接池。之后此连接可供下一个连接请求使用。

连接池促进了连接对象的重用，从而减少创建连接对象的次数。创建连接对象在时间和资源上的开销都很大，而连接池减少了这些方面的开销，因此可显著提高数据库密集型应用程序的性能。创建连接对象需要执行多项任务，例如网络通信、读取连接字符串、身份验证、事务登记以及内存分配等，所有这些都会占用时间和资源。此外，因为这些连接都是事先创建好的，所以应用程序只需等待很少时间即可获得连接。

连接池通常会有一些属性用来优化自身的性能。这些属性控制连接或连接池的行为，例如连接池内允许的最少和最多连接数，或一个连接在返回到连接池之前可保持为空闲状态的时间。最佳配置的连接池能够使快速响应时间与维护连接池中的连接所占用的内存之间取得平衡。通常需要尝试多种不同设置才能让一个特定应用程序达到最佳平衡。

此外，应用程序连接池的配置会因应用程序而异，因此可能无法用于其他应用程序。有关详细信息，请参阅您的应用程序文档。

2. 数据库共享服务器

在 Oracle 数据库共享服务器架构中，调度程序将多个传入网络会话请求导向一个共享服务器进程池，从而避免了为每个连接指定一个专用服务器进程的需求。此池中的空闲共享服务器进程从公共队列中选择请求。

如果无法改变应用程序层，则可针对大量短请求和事务使用共享服务器架构，由此可显著减少进程总数。有关使用 Oracle 数据库共享服务器架构的所有限制或具体指南，请查看您的应用程序文档。

在首次评估数据库共享服务器时，请查看以下文档并采用这些实践：

- Oracle 数据库管理指南中的“配置 Oracle 数据库以支持共享服务器”

- Oracle 数据库性能调优指南中的“共享服务器和调度程序的性能考虑事项”
- 最初每 500 个会话使用 20-30 个共享服务器，随后可进行调优（Oracle Net Services 最佳实践中的第 44 页幻灯片)
- 最初每 50-100 个会话使用 1 个调度程序，然后可进行调优
- 对于长时间运行的查询、报告任务、Data Guard 传输以及管理用连接，请继续使用专用服务器。您可始终混合使用共享服务器和专用服务器，当转变为使用数据库共享服务器时，这样做将非常有效。
- 在 sqlnet.ora 文件中，进行如下设置，USE_ENHANCED_POLL=on（Oracle Net Services 最佳实践中的第 46 页幻灯片）

3. 数据库驻留连接池 (DRCP) — Oracle Database 11*g* 中的新特性

   DRCP 使用池化服务器，相当于将专用服务器进程（而不是共享服务器进程）和数据库会话相结合。通过池化服务器模型，可避免为每个不需要长时间访问数据库的连接指定一个专用服务器进程。从 DRCP 获取了连接的客户端将连接到被称为“连接代理”的 Oracle 后台进程。连接代理实现 DRCP 池功能，并在客户端进程的入站连接中实现池化服务器的多路复用。

   虽然 DRCP 比共享服务器更高效，但是它仍需要少量应用程序代码更改才能获取和释放会话。请参阅 OTN 上的“数据库驻留连接池 (DRCP)”白皮书，网址为

   http://www.oracle.com/us/products/database/oracledrcp11g-1-133381.pdf。

   有关使用 DRCP 的所有限制或具体指南，请查看您的应用程序文档。

- **11.2.3.1 或更高版本的 Exadata 软件最多可支持源自一个硬件池中的一个或多个数据库服务器的 60,000 个并发连接。这意味着同时连接到一个 Exadata 单元并执行 I/O 操作的进程数不会超过 60,000。在 Exadata 11.2.2.4. 版本中，此限制是 32,000 个连接。而在 Exadata 11.2.2.4 之前，此限制是 20,000 个连接。 这里的上限目标取决于内存或 CPU 消耗，不过对于 X2-2 或 X3-2，每个节点大约可容纳 7,500 个进程，而对于 X2-8 或 X3-8，每个节点大约可容纳 30,000 个进程。**

要计算每个硬件池的进程数，需通过以下命令查询包含 Oracle ASM 实例、数据库文件系统 (DBFS) 实例以及应用程序数据库在内的每个数据库：

```
SQL> SELECT COUNT(1) FROM GV$PROCESS;
```

.

较高的进程数会导致内存分页，从而导致节点不稳定、CPU 争用、数据库闩锁争用以及长时间应用程序故障切换或中断。在增加进程数之前，进行测试非常重要。

- **可将 Oracle 监听器配置为限制传入连接数，以避免在数据库节点或实例出现故障后产生登录风暴。**

  通过 Oracle Net Listener 中的连接率限制器特性，数据库管理员 (DBA) 可以限制监听器处理的新连接的数量。如果启用该特性，Oracle Net Listener 会将监听器每秒处理的最大新连接数强制设定为用户指定的值。

  根据配置，此连接率会应用到一组端点或一个特定端点。

  通过以下两个 listener.ora 配置参数控制此特性：

  - CONNECTION_RATE_*<listener_name>=number_of_connections_per_second* 设定所有受连接率限制的监听端点的总连接率。如果指定了此参数，其优先级高于任何指定的端点级的连接率数值。
  - RATE_LIMIT 表示某特定监听端点受连接率限制。在监听器端点配置的 ADDRESS 部分指定此参数。如果将 RATE_LIMIT 参数设定为大于 0 的值，则在该端点级实施连接率限制。

  示例：限制新连接数以防止登录风暴。

```
APP_LSNR= (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521)(RATE_LIMIT=10))
    (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1522)(RATE_LIMIT=20))
    (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1523))
    )
```

在上面示例中，在端点级实施了连接率限制。端口 1521 每秒最多处理 10 个连接。端口 1522 每秒处理的连接数限制为 20。端口 1523 处理的连接数没有限制。 如果超过设定的连接数，系统会记录一条 TNS-01158 错误消息：Internal connection limit reached is logged.

请参阅 Net Services 参考指南。

其他设置

Exadata 配置和参数的最佳实践已记录在 MOS 说明 1274318.1 和 1347995.1 中。

启动对系统的监视和检查

需要通过 Enterprise Manager、由自动负载信息库 (AWR) 收集的统计信息或针对 Active Data Guard 环境的 Active Data Guard Statspack 来监视和管理您的数据库和系统资源。有关详细信息，请参阅以下文档：

- MOS 说明 1070954.1 中的 Exachk
- Oracle Enterprise Manager 12c: Oracle Exadata Discovery Cookbook 中的“通过 Enterprise Manager 12*c* 实现 Exadata 监视最佳实践”和 MOS 说明 1110675.1
- 位于 http://www.oracle.com/asr 的“Exadata 自动服务请求”
- MOS 说明 454848.1 中的“Active Data Guard statspack”
- Oracle Clusterware 管理和部署指南中的“集群运行情况监视”。

资源管理

Oracle Database Resource Manager（简称 Resource Manager）) 是一个对负载和数据库之间的系统资源进行精细控制的基础架构。您可以使用 Resource Manager 管理 CPU、磁盘 I/O 以及并行执行。Resource Manager 在以下两个不同场景下对您很有用：

- 对于数据库内整合，该工具可用来管理应用程序间的资源使用和争用。
- 对于数据库间整合，该工具可用来管理数据库实例间的资源使用和争用。

将 Resource Manager 用于数据库内（模式）整合

在数据库内（模式级）整合中，您可以使用 Resource Manager 控制应用程序如何在一个数据库内共享 CPU、I/O 和并行服务器。它将依据数据库管理员指定的**资源计划**将资源分配给用户会话。资源计划指定如何将资源分发给各**资源用户组，**资源用户组是按照资源需求对用户会话进行分组而形成的。对于模式整合，通常需要为每个应用程序创建一个用户组。**资源计划指令**将资源用户组和资源计划关联到一起，从而指定如何将 CPU、I/O 和并行服务器资源分配给用户组。

CPU 资源管理另外一个优势是，可使关键的后台进程（例如 LGWR、PMON 和 LMS）不会遇到资源缺乏的情况。从而可提高 OLTP 负载的性能，并降低 Oracle RAC 数据库实例被逐出的风险。

要管理每个应用程序的资源使用，必须配置并启用资源计划。有关如何进行这些操作的详细信息，请参阅 Oracle 数据库管理员指南中的“使用 Oracle Database Resource Manager 管理资源”、MAA 白皮书“使用 Oracle Database Resource Manager”，并可从 MOS 说明 1339769.1 中获取设置示例脚本。

要管理每个应用程序的磁盘 I/O，必须启用 **I/O 资源管理 (IORM)。**用于管理 CPU 的资源计划也可用于管理磁盘 I/O。IORM 以单元为单位管理 Exadata 单元的 I/O 资源。只要 I/O 请求会导致单元磁盘过载，IORM 就会依照已配置的资源计划对 I/O 请求进行调度。IORM 调度 I/O 请求的方法是，立即发出一些 I/O 请求，而让其他请求排队。IORM 根据资源计划中的资源分配来选择要发出的 I/O 请求：相对于那些获得较少资源分配的数据库和用户组，获得较多资源分配的数据库和用户组将得到更频繁的调度。如果未达到单元的运行容量限制，IORM 不会让 I/O 请求排队等候。

如果启用了 IORM，它会自动管理后台 I/O。日志文件同步以及控制文件的读写等关键后台 I/O 的优先级较高，而非关键后台 I/O 的优先级较低。

将 Resource Manager 用于数据库整合

Resource Manager 可在两个方面对数据库整合提供帮助。首先，Resource Manager 可通过实例囚笼来帮助控制 CPU 使用和管理 CPU 争用。其次，Resource Manager 可通过 IORM 的数据库间资源计划来控制磁盘 I/O 使用和争用。

通过**数据库间 IORM 计划，**您可以对共享 Exadata 单元的多个数据库进行管理。您可通过单元控制命令行界面 (CellCLI) 实用程序来配置数据库间 IORM 计划。通过数据库间 IORM 计划，您可以对每个数据库的以下几方面做出规定：

- 磁盘 I/O 资源分配：获得较多资源分配的数据库能够更快地发出磁盘 I/O 请求。通过数据库资源计划指定数据库内各负载的资源分配。如果未启用数据库资源计划，则数据库中的所有用户 I/O 请求都将得到平等对待。但是后台 I/O 仍自动具有高优先级。
- 磁盘利用率限制：除了指定资源分配外，您还可以指定每个数据库的最高磁盘利用率。例如，如果生产数据库 OLTP 和 OLTP2 共享 Exadata 存储，那么您可以为这两个数据库设置最高利用率限制。通过设置数据库磁盘利用率限制，您能够获得更可预测的、一致的性能，而这在托管环境中通常很重要。

  如果指定了最高利用率限制，数据库将无法使用超额容量。同时，如果指定了最高利用率限制，磁盘就不会满负荷运行。
- 闪存缓存使用：自 11.2.2.3.0 起，在该版本和之后的 Exadata 软件中，IORM 均支持闪存缓存管理。可将 ALTER IORMPLAN 闪存缓存属性设置为“off”，以防止数据库使用闪存缓存。这样可将闪存缓存留给任务关键型数据库使用。只有您确认该数据库对闪存缓存的使用影响了关键数据库对闪存缓存的命中率时，您才能禁用闪存缓存。禁用闪存缓存

会带来增加磁盘 I/O 负载这样的负面影响。

> **注意：**在 11.2.3.1 或更高版本的 Exadata 存储服务器软件中，数据库间 IORM 计划可以包含用于多达 1023 个数据库的指令，而在之前版本中，只包含用于 31 个数据库的指令。可将未使用指令的所有数据库作为一个实体来管理。

按类资源管理是一个高级特性。该特性允许您主要按照正在执行的任务的类别来分配资源。例如，假设所有数据库都拥有三种负载：OLTP、报告和维护。要根据这些负载类别分配 I/O 资源，您将使用按类资源管理特性。 请参阅“Oracle Exadata 存储服务器软件用户指南”中的“关于按类资源管理”。

本节重点介绍针对 Exadata 的关键整合实践。有关全面 Exadata 资源管理的**先决条件、最佳实践和补丁**，请参阅 Oracle Database Resource Manager 的主要说明（MOS 说明 1339769.1）。

管理一个硬件池中 CPU 和 I/O 资源的一些指导原则

- 启用实例囚笼。根据大小分析，为每个实例设置 CPU_COUNT。
- 对于关键硬件池，请使用分区方法：(CPU_COUNT) 总和 <= CPU 数量的 75%。
- 对于非关键硬件池和测试硬件池，请使用过度使用方法：(CPU_COUNT) 总和 <= (3) * CPU 数量，具体由可接受的响应时间或吞吐量要求决定。
- 配置并启用数据库间 IORM 计划。
- 要为数据库中的负载配置 Resource Manager，请参阅 Oracle 数据库管理员指南中的“数据库资源管理和任务调度”一节。

监视和调优

主要建议如下：

- 检查每个应用程序的性能指标是否已达到要求。这些指标是最能说明问题的。
- 针对实例囚笼，使用以下查询监视每个实例的 CPU 利用率。也请参阅 MOS 说明 1338988.1。

```
SQL> select to_char(begin_time, 'HH24:MI') time,
max(running_sessions_limit) cpu_count, sum(avg_running_sessions)
avg_running_sessions, sum(avg_waiting_sessions)
avg_waiting_sessions    from v$rsrcmgrmetric_history group
by begin_time order by begin_time;
```

如果 **avg_running_sessions** 一直小于 CPU_COUNT 的值，那么您可以在不影响性能的条件下降低 CPU_COUNT 的值，而对于其他需要额外资源的实例，您可以增加 CPU_COUNT 的值。

如果 **avg_waiting_sessions** 一直大于 CPU_COUNT 的值，并且响应时间和吞吐量让人无法接受，您可以通过增大 CPU_COUNT 来改进性能（如果您有可用的 CPU 资源）。否则，可以考虑将此应用程序或数据库迁移到其他硬件池。

- 使用过度使用实例囚笼方法时，应对系统进行监视。如果系统 CPU 运行队列大于 9，则应考虑降低所有实例的 CPU_COUNT 总和。
- 使用 MOS 说明 1337265.1 中的脚本监视 I/O 指标。

  通过此脚本，可对数据库和用户组每分钟使用的磁盘和闪存指标进行监视。还能监视每分钟的磁盘 I/O 延迟指标。如果启用了 IORM，还能提供对每分钟 I/O 限制指标的监视。

  注意：对于高性能磁盘，每秒支持的 I/O 操作次数 (IOPS) 最高为 300 IOPS，而对于高容量磁盘，则为 150 IOPS。使用高性能磁盘的 Exadata 单元支持的最大吞吐量约为 1.5- 2.0 GB/秒，而使用高容量磁盘的 Exadata 单元的最大吞吐量则为 1 GB/秒。I/O 吞吐量问题的典型表现是：高 I/O 利用率（iostat 命令中的 %util 参数）、高磁盘队列大小（iostat 命令中的 avgqu-sz 参数）以及糟糕的响应时间、I/O 延迟（通过 iostat 命令监视）和吞吐量。

调优以实现低延迟负载

对于 OLTP 负载，低延迟极为重要。请进行如下检查：

1. 高缓冲区缓存命中率 (> 98%)。根据情况对缓冲区缓存进行相应调优。
2. 高闪存缓存命中率 (> 90%)。如果需要，将常用表保存在闪存缓存中。
3. 低磁盘利用率 (< 60%)。磁盘利用率较高时，将看到良好的吞吐量，但延迟却较高。因此，如果磁盘利用率较高，请使用 I/O 指标确定哪个数据库和应用程序正在生成此负载。
4. 检查以下数据库等待事件是否是低延迟：*log file sync*（例如 < 10 ms）、*db file sequential read*（例如 < 15 ms）以及 *cell single block physical read* （例如 < 15 ms）。
5. 每个数据库的磁盘利用率。如果一个数据库的磁盘利用率有很大提高，可能会对其他数据库产生影响。可通过 I/O 指标或 AWR 和 ASH 报告中的顶级 I/O 事务进行监视。

您还可以使用 IORM 改进 OLTP 延迟，方法如下：

1. 为具有 OLTP 负载的数据库和用户组分配更多资源。
2. 如果延迟仍不足够低，可将 IORM 的目标设置为“低延迟”。
3. 如果延迟仍不足够低，可能是因为您的性能目标太高，以至于不允许 OLTP 和 DSS 负载共享存储。吞吐量和延迟之间存在一种内在平衡。对于极低的磁盘延迟，磁盘利用率可能也需要这么低，以至于使得 OLTP 和 DSS 负载之间共享存储变得没任何意义。这种情况下，可考虑为 OLTP 和 DSS 创建单独的存储或硬件池。

案例 1：OLTP 数据库的整合

此方案说明如何在三个关键 OLTP 数据库中分配 I/O 资源以及如何在 Exadata 数据库云服务器 X3-2 上配置实例囚笼。通常，我们建议使用简单的单级资源计划。除了指定资源分配外，以下资源计划根据其他数据库的负载情况，使用 I/O 限制指令将金级数据库的 I/O 利用率限定为 50%，而将银级数据库的 I/O 利用率限定为 30%，以使得性能不会有太大变化。当其他数据库空闲时，要允许每个数据库使用 100% 的磁盘资源，可删除此限制指令。

表 4. OLTP 整合

数据库名称	说明	级别	资源分配	限制	CPU_COUNT
OLTP-A	金级数据库	1	35	50	8
OLTP-B	金级数据库	1	35	50	8
OLTP-C	银级数据库	1	20	30	6
其他	（任何其他数据库）	1	10	30	4

```
ALTER IORMPLAN  -
   dbplan=((name=oltp-a, level=1, allocation=35, limit=50), - (name=oltp-b, level=1, allocation=35,
      limit=50), - (name=oltp-c, level=1, allocation=20, limit=30), - (name=other,      level=1,
      allocation=10, limit=30))
```

要启用实例囚笼，需设置每个实例的 cpu_count，然后再启用 CPU Resource Manager。 例如，可通过以下语句为 OLTP_A 实例启用实例囚笼：

```
SQL> ALTER SYSTEM SET CPU_COUNT = 8 SCOPE=SPFILE;
SQL> ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'DEFAULT_PLAN'
SID='*' SCOPE='BOTH';
```

案例 2：混合负载的整合

此案例说明如何将 OLTP 负载的优先级设置为高于数据仓库负载的优先级，以保持 I/O 低延迟。

表 5. 具有 OLTP 优先的混合整合

数据库名称	说明	级别	资源分配

OLTP-A	金级数据库	1	50
OLTP-B	银级数据库	1	20
DW-X	铜级数据仓库	1	10
DW-Y	铜级数据仓库	1	10
其他	（任何其他数据库）	1	10

因为 OLTP 数据库分配的资源比数据仓库数据库多，因此 IORM 会自动优化以实现低磁盘延迟。要显式控制此目标，请使用 ALTER IORMPLAN 将此目标设置为“低延迟”。在混合负载环境中，IORM 可将磁盘延迟降低到约 15 毫秒。

案例 3：数据仓库的整合

此案例说明如何在三个数据仓库数据库中分配 I/O 资源。通常，我们建议使用简单的单级资源计划。除了指定资源分配外，以下资源计划根据其他数据库的负载情况，使用 I/O 限制指令将金级数据库的 I/O 利用率限定为 50%，而将铜级数据库的 I/O 利用率限定为 20%，以使性能不会有太大变化。在要求使用“按性能付费”模型的托管环境中，利用率限制非常有用。

表 6. 数据仓库整合

数据库名称	说明	级别	资源分配	限制
DW-X	金级数据库仓库	1	40	50
DW-Y	金级数据库仓库	1	40	50
DW-Z	铜级数据仓库	1	10	20
其他	（任何其他数据库）	1	10	50

资源管理最佳实践

要了解包含脚本及推荐的特用于实现有效资源管理的软件版本在内的最新的资源管理最佳实践，请参阅 MOS 说明 1339769.1。

维护和管理考虑事项

Oracle 软件和文件系统空间

只保留少量活动的 Oracle 主目录版本，因为大量不同的数据库版本会增加维护复杂性。还请记住，Oracle Grid Infrastructure 软件版本必须大于等于 Oracle 数据库软件的最高版

本。只要可能，尽量使用共享的 Oracle 主目录，并保留固定数量的终端版本。

为确保根分区空间能够最大化以容纳所有 Oracle 主目录、Oracle Grid Infrastructure 主目录以及各种日志目录，请执行以下操作：

- 确保在选择首选操作系统后已回收磁盘空间。请参阅“Oracle Exadata 数据库云服务器所有者指南”中的“选择操作系统后回收磁盘空间”一节。
- 评估是否可以为非根分区 (/u01) 增加更多空间。 请参阅“Oracle Exadata 数据库云服务器所有者指南”中的“*调整 LVM 分区大小*”一节。
- 有关分步说明，请参阅“*如何扩展 Exadata 计算节点文件系统*”（MOS 说明 1357457.1）。
- 通过 cron 管理审计文件目录的增长（MOS 说明 1298957.1）。
- 或者您可将日志目录和文件复制和移除到 DBFS 或外部 NFS 目录中。

安全性和管理角色

如果一个硬件池中整合了多个数据库，则可能需要隔离某些数据库组件或功能，以便将管理和权限明确分开。有关如何通过使用操作系统和数据库范围的安全功能来防止管理员或最终用户的未授权访问的详细说明，请参阅 Oracle Exadata 数据库云服务器整合：隔离数据库和角色白皮书。

注意：在 Oracle 11*g* 中，每个存储网格最多允许有 63 个 ASM 磁盘组。

下图描述了如何利用一些与默认推荐的单 DATA 和 RECO 磁盘组配置不同的安全实践来限制访问权限。

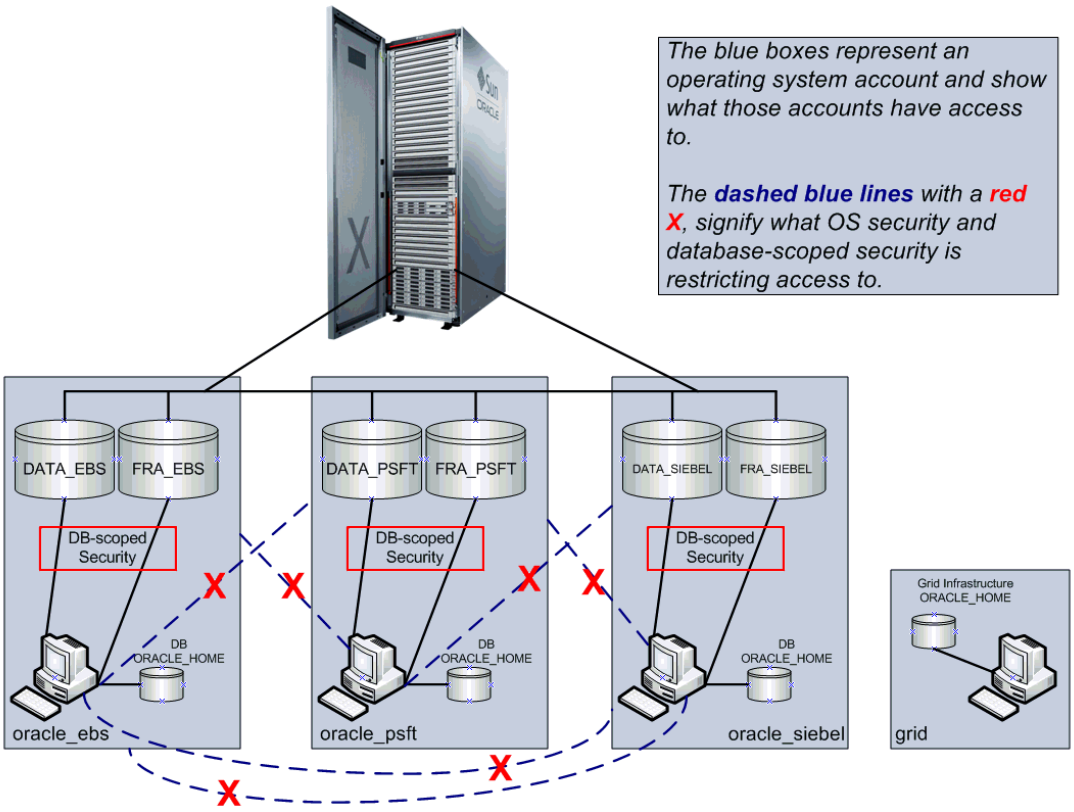

图 2. 用于访问限制的安全实践

用于整合的 Exadata MAA

Oracle 最高可用性架构 (MAA) 与整合环境密切相关，在整合环境中，计划停机和意外停机产生的影响要比应用程序运行在单独系统上时大很多。有关 Exadata 的通用最佳实践，请参阅 Oracle Exadata 数据库云服务器 MAA 最佳实践白皮书。以下各节以整合环境为背景，重点讲述一些主要 MAA 建议。

业务关键应用程序的整合 MAA 环境由以下部分组成：

- 生产硬件池
- 备用池。备用池是生产硬件池的精确同步副本，使用 Data Guard 来维护它。根据生产硬件池的重要程度，它可能会拥有一个采用同步零数据丢失保护措施的本地备用池，还可能会拥有一个采用异步传输方式的远程备用池，以便能够提供地域保护，防止大范围的服务中断。理想情况下，备用池的容量配置应与生产硬件池类似，以在需要故障切换时能够提供等同的服务级别。您可以通过一些策略来有效利用处于备用角色的备用池。有关详细信息，请参阅 MAA 白皮书：Oracle Exadata 数据库云服务器的灾难恢复。通过对备用数据库首先打补丁（请参阅 MOS 说明 1265700.1），可使用备用池来验证补丁和软件更改。

- 开发/测试硬件池，它们也非常重要，因为所有更改都应该先在开发和测试系统中进行验证，以便在正式应用时能最大限度保持生产硬件池和备用硬件池的稳定性和可用性。

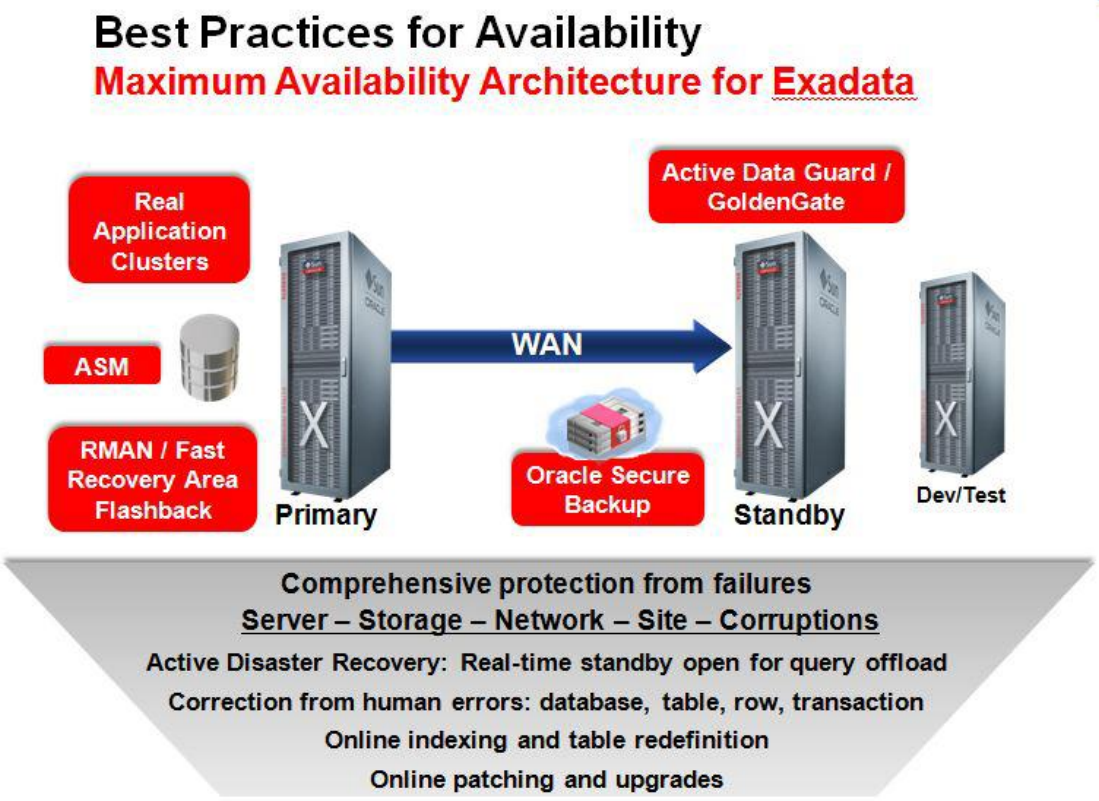

图 3. Exadata 的最高可用性架构：保护整合数据库环境

最高可用性架构（图 3）为整合环境提供以下高可用性功能：

- Oracle Real Application Clusters (RAC) — 提供针对节点和实例故障的容错功能。
- Oracle ASM 和 Oracle Exadata — 提供针对各种存储故障的容错功能。
- Active Data Guard — 自动修复物理块损坏，防止备用池出现可能影响到主硬件池的数据损坏或服务中断，并且在个别数据库或整个硬件池因某些原因出现故障时支持快速故障切换以保持高可用性。
- 损坏检测、预防和自动修复最佳实践 — 在 Data Guard 配置中 (MOS 1302539.1)

- Oracle 联机补丁安装、Oracle Grid Infrastructure 和 Oracle Exadata 单元滚动补丁安装、Oracle RAC 滚动维护、Oracle Data Guard 备用数据库先打补丁、Oracle Data Guard“临时逻辑”数据库滚动升级以及 Oracle GoldenGate 异构迁移和零停机维护 — 这些都最大限度减少或消除了计划维护所需的停机。
- 闪回技术，当出现逻辑损坏时，通过该技术可实现快速具体时间点恢复。

有关上述高可用性功能的最佳实践都记录在技术白皮书 Oracle Exadata 数据库云服务器 MAA 最佳实践中。

以下 Exadata MAA 操作实践对于实现整合数据库环境的高可用性目标来说也非常重要：

1. 将高可用性服务级别协议文档化。
2. 验证不同服务中断的修复策略和滚动升级解决方案。
3. 定期测试并升级到 MOS 说明 888828.1 中建议和传达的软件版本和补丁，并参考 MOS 说明 1461240.1 中介绍的常规硬件和软件维护建议。
4. 遵循 MOS 说明 1262380.1 中介绍的测试和修补实践，包括备用数据库先打补丁（MOS 说明 1265700.1）。
5. 在计划维护活动的前后运行最新版本的 exachk 实用软件（MOS 说明 1070954.1），并且每个月至少运行一次。
6. 定期执行 Data Guard 角色转换，例如每 6 个月执行一次。
7. 配置 Exadata 监视功能、Oracle Configuration Manager (OCM) 以及自动服务请求最佳实践（MOS 说明 1110675.1 和 1319476.1）。
8. 请参阅位于 http://www.oracle.com/goto/maa 的最新 MAA 最佳实践和白皮书。也请参阅 Oracle Exadata 数据库云服务器 MAA 最佳实践技术白皮书中的通用 Exadata MAA 实践。

以下各节重点介绍数据库整合的其他 Exadata MAA 考虑事项。

打补丁和升级

在最初配置 Exadata 硬件池中的软件时，请参阅 MOS 说明 888828.1 以了解受支持的最新 Exadata 软件配置。除了滚动升级情况之外，各 Exadata 单元的版本应相同。所有数据库节点应使用相同的 Oracle Grid Infrastructure 软件版本，并且有一些节点共享 Oracle 数据库主目录。

在更新 Oracle 数据库软件版本时，最好按照异地打补丁和升级程序进行。此方法尤其适用于数据库整合环境，因为这样做可以在不影响其余数据库的情况下将目标数据库迁移到新版本，而且如果出现意外性能问题，还能回退到上一版本。您可以利用各种工具，例如针对数据库服务器使用 OPlan 和 OPatch，而针对 Exadata 单元则使用 patchmgr。

备份和恢复

您的备份、还原和恢复策略不会随整合而改变。首先应该清楚如何针对不同恢复情况使用备份，并清楚您预期的恢复时间目标 (RTO)、恢复点目标 (RPO) 和备份时窗。您的硬件池应包含具有相似高可用性要求的应用程序和数据库。对于采用 MAA 硬件池架构的关键硬件池，可能会将备份用于数据库节点裸机恢复、应对双重灾难以及一些损坏或逻辑故障情况。其次，您应该了解通过 Exadata 上基于磁盘的备份（在系统内部或通过 Exadata 存储扩展机架备份）或将备份存放到磁带上可以实现什么目标。有关可能的备份和还原速率以及配置实践的相关信息，请参阅 Exadata 数据库云服务器的备份与恢复性能和最佳实践和 使用 Sun ZFS 存储设备备份和恢复 Oracle Exadata 数据库云服务器时的备份与恢复性能以及最佳实践技术白皮书。

下一步您应该了解的是，备份和还原数据库与可用系统资源密切相关，尤其是以下资源：

- 用于备份和还原的 Exadata 磁盘吞吐量
- 网络带宽：InfiniBand 用于基于 Exadata 的备份，外部网用于 Exadata 以外设备的基于磁盘的备份或磁带备份
- 第三方 I/O 吞吐量：磁带吞吐量，用于基于磁带的备份；存储吞吐量，用于非 Exadata 存储目标。

你可以选择同时备份或还原一个或多个数据库。瓶颈问题将可能由上述的一个元凶引起。

示例 1：将全机架 Exadata 数据库云服务器备份到磁盘

在本示例中，在使用高性能 SAS 磁盘的全机架 Exadata 数据库云服务器 X2-2 上有 5 个数据库。所有数据库占用的空间总和约为 50 TB。

根据我们的 MAA 研究，当 RMAN 操作利用所有数据库节点，并且为每个数据库节点分配 2 到 8 个 RMAN 通道时，预计备份速率可高达 20 TB/小时。您可能会选择限制备份速率，以确保在执行备份操作期间不会影响应用程序性能。假设只使用 1 或 2 个 RMAN 通道，备份速率是 10 TB/小时，那么可在 5 个小时内备份完所有数据库。可以选择一次只备份一个数据库，也可以选择同时备份多个数据库。关键是将任务分配给所有节点，以便使得各数据库节点受到的影响是均衡的。例如，可以针对两个不同数据库执行两个

RMAN 操作，其中一个备份服务利用数据库节点 1-4，而第 2 个备份服务利用数据库节点 5-8。此组合方法实现的备份速率仍然是 10 TB/小时。而使用我们推荐的增量备份方法，备份时间可缩短为不到 1 个小时。

示例 2：将全机架 Exadata 数据库云服务器备份到磁带

在本示例中，在使用高性能 SAS 磁盘的全机架 Exadata 数据库云服务器 X2-8 上有 10 个数据库。这些数据库采用基于磁带的备份解决方案，以使 DATA 区域有更多可用空间。所有数据库占用的空间总和约为 100 TB。本示例提供 2 个介质服务器和 16 个磁带驱动器（例如速率为 200 MB/秒）磁带驱动器的数量以及它们的吞吐量是造成瓶颈的原因。使用此基于磁带的基础架构，备份/还原吞吐量接近 11 TB/小时。将备份服务任务分配给两个数据库节点，并为每个数据库节点分配 8 个 RMAN 通道。根据备份集合的实际大小，整个 100 TB 数据库的完整备份可在 10 小时内完成，而日常的累积增量备份可在 2 小时内完成。

此外，您还可以使用资源管理来排定不同备份和应用程序负载的优先级。

Data Guard

有关针对 Exadata 的 Data Guard 实践的信息，请参阅 Oracle Data Guard：Exadata 数据库云服务器灾难恢复最佳实践白皮书。针对备用池的主要整合考虑事项如下：

- 确保主池和备用池不在同一个 InfiniBand 结构上。
- 如果要求零数据丢失，那么您可能需要使用本地备用池，并需要进行性能评估，以评估 Data Guard 同步传输对性能的影响。
- 需要足够的网络带宽来处理所有主数据库和备用数据库之间的所有重做数据流量。如果主池和备用池所处的广域网 (WAN) 没有足够的网络带宽，则应考虑采用重做压缩技术。重做压缩可减少主池和备用池 5-10% 的 CPU 开销。
- 执行角色转换测试，包括应用程序与 Data Guard 切换操作和故障切换操作。针对一个或多个数据库，最坏情况下针对整个硬件池执行此测试，以确保所有环境都满足您的 RTO 和 RPO 要求。
- 确认备用池上有足够的系统资源，以便在整个生产硬件池出现故障时，备用池能够支持所有目标生产负载。
- 鉴于主池和备用池中众多数据库管理的复杂性，我们强烈建议您使用 Data Guard Broker 和 Enterprise Manager。如果要求使用自动故障切换功能以降低 RTO 以及避免在检测和响应一些故障上的时间开销，可考虑使用 Data Guard 快速启动故障切换和 MAA 客户端故障切换最佳实践（Data Guard 11g 第 2 版客户端故障切换最佳实践）。

模式整合环境的恢复

对于模式级整合环境（在一个数据库中运行多个应用程序模式），主要的考虑事项是各应用程序模式是否适合共存在同一个数据库中。如果模式级整合是可行的，则应按照所有 Exadata MAA 标准建议对一个数据库进行联机配置和管理。此外，以下建议和注意事项也适用：

- 每个应用程序模式应包含在一个单独的表空间或一个表空间集合中。从而实现方便且高效的空间管理。如此还能促进使用表空间时间点恢复 (TSPITR) 或闪回表操作来恢复某特定应用程序模式，而不会对其他应用程序产生影响。
- 每个应用程序模式应使用单独的数据库服务。
- 调优您的备份和恢复实践，以便将维护各应用程序模式或表空间时对其他应用程序产生的影响降到最低。其中包括了解不同方法的开销和需求，并确保所需系统资源可用，以支持实现期望的目标。不同的修复方法包括 TSPITR、闪回表或闪回事务、导出和导入操作或数据泵操作。
- 如果考虑使用 TSPITR，那么还应考虑是否能够使用映像副本以支持更快的恢复。这种情况下，RMAN 不需要从备份恢复数据文件。
- 请参阅“Exadata 整合环境最佳实践中的 MAA 模式恢复可选方案”（MOS 说明 1386048.1）

总结

Exadata 是最佳的数据库整合平台，它经过完全集成设计，可为 OLTP 和数据仓库数据库提供卓越性能和可伸缩的容量。Oracle 数据库、存储和网络网格架构与 Oracle 资源管理相结合，为您提供一种比其他虚拟化策略（例如硬件或操作系统虚拟化）更简单、更灵活的数据库整合方法。当系统支持多个应用程序和数据库在一个数据库整合环境中使用共享资源时，遵循本文和相关参考文件中记录的最佳实践将能够最大限度地提高系统稳定性和可用性。