

Oracle Maximum
Availability Architecture

Oracle 白皮书
2014 年 4 月

数据库整合高可用性最佳实践

数据库即服务基础

概述	2
引言	3
操作系统虚拟化 — 虚拟机.....	3
模式整合	3
Oracle Database 11g 的数据库整合	4
使用 Oracle Multitenant 的数据库整合	4
Oracle Exadata — 针对整合和 DbaaS 进行了优化	6
有力证据 — 使用 Exadata 获得高整合密度.....	8
高可用性参考架构	10
青铜级 — 单实例.....	11
白银级 — 具有自动故障切换的高可用性	15
黄金级 — 全面的高可用性/灾难恢复	18
白金级 — 白金级应用程序零停机.....	23
数据库整合规划.....	27
迁移到多租户架构	32
文件移动	34
迁移和 Data Guard.....	35
Oracle 资源管理	35
规划.....	35
控制资源	37
监视资源	41
高可用性与数据保护	44
管理意外停机	46
管理计划维护	47
DbaaS 的生命周期管理.....	49
总结	50
附录 A: Exadata 整合密度和性能	51
附录 B: RMAN 活动数据库复制迁移	55

概述

企业目前承受着巨大的压力，它们需要花更少的钱办更多的事，需要降低风险并提高敏捷性。为了实现上述这些目标，积极整合信息技术 (IT) 基础架构并在公有和私有云上部署数据库即服务 (DBaaS) 是许多企业都在寻求的一种战略。

需要多项关键要素才能实现借助数据库整合和 DBaaS 而降低成本的全部潜能。要求具有高整合密度并简化管理才能实现硬件和管理成本的最大降低。然后，还必须将这些特质与能够满足可用性、性能和数据保护要求服务级别协议 (SLA) 的智能软件基础架构相结合。

随 Oracle Database 12c 引入的 Oracle Multitenant 代表着对 Oracle 数据库的根本性重新设计，从而使它成为数据库整合的最佳平台。Oracle Multitenant 是一种进化，它可以轻松整合现有 Oracle 数据库以降低成本。Oracle Multitenant 还是一个变革，它支持最大的整合密度，并且与之前的整合战略相比，可明显简化对整合后数据库环境的管理。

Oracle Multitenant 与 Oracle 最高可用性架构 (MAA) 一起为整合的数据库环境提供高可用性和数据保护，否则所有中断产生的影响将会被放大多次。虽然不管底层硬件平台或操作系统环境如何，Oracle Multitenant 和 MAA 都能够提供独一无二的优势，但是当它们部署于 Oracle 工程化系统时可提供最大优势。Oracle 集成硬件和软件解决方案通过使用标准的高性能平台降低了总生命周期成本，这些平台在以下多个维度上实现整合环境的规模经济：性能、可靠性、可管理性和支持。例如，与传统系统相比，已经证实 Oracle 数据库云服务器在整合密度上最高可获得 5 倍的优势。

本白皮书介绍使用 Oracle Multitenant 实现的数据库整合 MAA 最佳实践。它介绍了作为 DBaaS 基础的标准高可用性架构。它最适用于以下技术读者：负责将传统数据库部署整合和迁移到 DBaaS 的架构师、IT 主管和数据库管理员。推荐的最佳实践同样适用于 Oracle 数据库支持的任何平台，除非已明确注明优化或示例仅适用于 Oracle 工程化系统。

引言

本白皮书首先讨论传统数据库整合方法及其利弊权衡，从而为 Oracle Multitenant 提供的高价值提供参考。白皮书的剩余部分将呈现数据库整合的高可用性最佳实践，并且包括实现高可用性和数据保护的服务级别所驱动的标准参考架构：青铜级、白银级、黄金级和白金级。

操作系统虚拟化 — 虚拟机

早期的整合工作重点是使用操作系统虚拟化进行服务器整合。操作系统虚拟化提供简单的模板驱动的部署，并且能够将单个物理计算机分为多个虚拟机 (VM) 来提高服务器利用率。提高服务器利用率是数据库整合的一个目标。不是在各自具有不同利用率的十台独立物理计算机上运行十个数据库，而是通过将同样的数据库部署在一个物理系统上的十个虚拟机 (VM) 中，从而减少服务器占用空间。

然而从数据库整合的角度来看，虚拟机本省能够实现的操作存在以下诸多缺陷：

- 由于运行操作系统和数据库软件的多个实例，每一个都有它自己伴随的内存和系统占用，因此效率低下，导致整合密度有限。随着更多的虚拟机和数据库被部署到给定的物理计算机，这将导致性能变差。
- 虚拟机控制管理成本的能力有限，因为它们无法减少必须管理的数据库和操作系统环境的总数。
- 虚拟机无法满足任务关键型应用程序（不容许停机或数据丢失）的高可用性要求。虚拟机限制于中断后冷启动，并且它们依赖于与外部复制技术集成来提供实时数据保护。
- 虚拟机在为 Oracle 数据库提供动态资源管理上功能有限：在数据库整合环境中，由于业务周期的原因，一天中不同用户组或数据库对系统资源（例如 CPU 或 IO）的消耗可能会有所不同。

模式整合

模式整合描述将独立数据库转换为模式（整合到单个数据库）的过程。这解决了将虚拟机用作数据库层的整合战略内在的整合密度和管理效率缺陷。虽然有许多成功的部署案例，但是模式整合有它自己的一套权衡方法。实施起来比较困难，尤其是未针对模式整合进行设计的现有应用程序和数据库环境。与使用虚拟机实现的情况相比，模式整合在模式级别也没有固有的可移植性和隔离简单性。

Oracle Database 11g 的数据库整合

使用 Oracle Real Application Clusters (Oracle RAC) 和 Oracle Resource Management 的 Oracle 数据库提供了第一个针对 Oracle 数据库进行优化的整合平台，它是 Oracle Database 11g 的 MAA 最佳实践。Oracle RAC 能够将多个 Oracle 数据库轻松地整合到单个 Oracle RAC 集群中。Oracle Resource Management 支持跨 Oracle RAC 集群上整合的多个数据库和用户组控制系统资源的优先级，以确保达到响应时间服务级别目标。

Oracle 数据库云服务器通过将可伸缩的性能与独特的资源管理功能相结合，为整合环境提供了额外的价值。有关更多详细信息，请参阅[“Oracle 数据库云服务器上的数据库整合最佳实践”](#)。

Oracle RAC、Exadata 和 Resource Management 提供了一个具有大量可管理性、性能和高可用性优势的简单整合方法，但是它们无法实现像模式整合一样的效率。Oracle RAC 集群中的每个数据库（驻留在磁盘上的系统和用户数据）都有一个或多个数据库实例（专用后台进程和内存区域）。每个数据库实例都可以挂载和打开单个数据库。随着更多的数据库整合到单个集群中，每个数据库实例必须专用的内存和 CPU 数量对可实现的整合密度造成了实际限制。

使用 Oracle Multitenant 的数据库整合

Oracle Multitenant 通过引入多租户容器数据库 (CDB) 和可插拔数据库 (PDB) 的概念，从根本上改变了 Oracle 数据库架构。现有数据库可以轻松地转换为 PDB。通过将多个 PDB“插入”单个 CDB¹ 来实现整合。带有 Oracle Multitenant 的 Oracle Database 12c 的设计在数据库整合的每个方面都能提供最有效的平台。

CDB 有一组后台进程和共享内存区域 (SGA) 供所有 PDB 使用。相对于将多个独立数据库整合到单个物理计算机、多个虚拟机或 Oracle RAC 集群的传统方法，此架构需要的 CPU 和内存更少。虽然 CDB 可以部署到物理或虚拟环境中，但是在物理计算机上部署时，它能实现数据库层的最高管理和性能效率。CDB 本身成为数据库层的虚拟化技术，从而消除了多个虚拟机和来宾操作系统的开销。

已通过在运行 Oracle Database 12c 的 Sun T5-8 服务器上执行的一系列测试证实了 Oracle Multitenant 架构的优势。该 T5-8 配置有 128 核、2 TB 内存和 8 个 Exadata 存储服务器。使用整合的 252 个 OLTP 数据库（33% 的 1 GB 小数据库，33% 的 5 GB 中等数据库以及 33% 的 15 GB 大数据库）运行测试。测试设计为 将多租户架构中的可插拔数据库与单实例数据库的部署进行比较。

¹ 在 Oracle Database 12c 第 1 版 (12.1) 中，一个 CDB 最多可包含 252 个 PDB。计划在将来的版本中提高此限制。

测试证明，与使用相同系统资源（CPU、内存和 I/O）的单实例数据库相比，Oracle Multitenant 可以增加整合密度。多租户可实现：

- 整合密度提高 50%（整合的数据库数量），同时每个数据库可达到相同的吞吐量
- 整合相同数量的数据库时，总吞吐量增加 80%

从另一个角度看时，这些测试证实了 Oracle Multitenant 如何带来真正的硬件和软件许可成本节省：

- 要整合同等数量的数据库 (252)，且吞吐量水平相同，多租户架构要求的 CPU 内核数减少 64 个，并且只要求 1/3 的 IOPS（每秒 I/O）

图 1 对测试结果进行了总结。有关更多详细信息，请参阅 Oracle 技术网上发布的 [Oracle Multitenant 整合研究](#)。

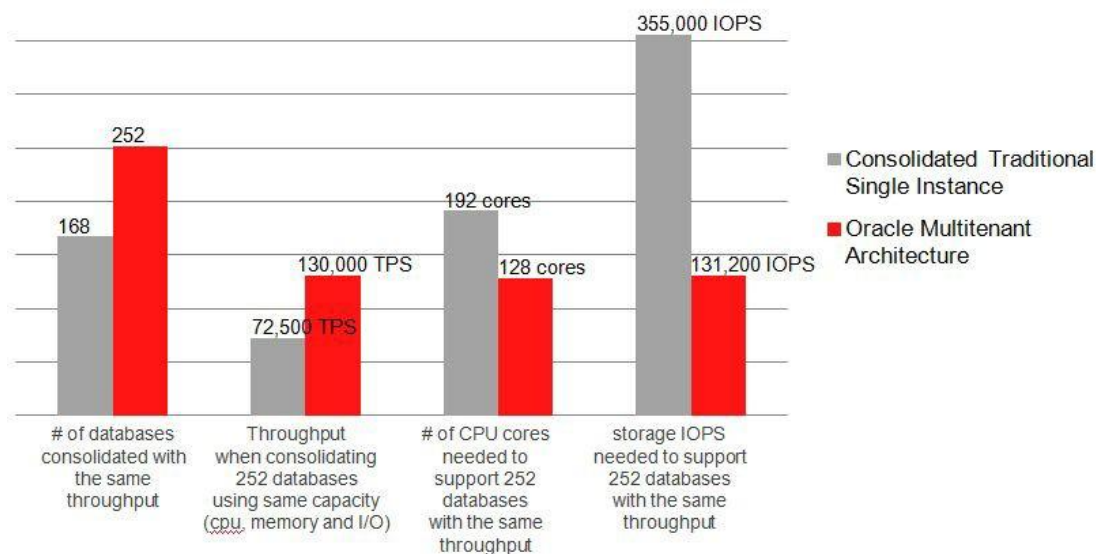


图 1：Oracle Multitenant 架构的整合优势

Oracle Multitenant 还提供了多合一管理的简单性和效率。图 2 对此进行了简单说明。不管在 CDB 中整合多少 PDB，仍然只需管理一个数据库（多租户容器数据库）：单个备份、用于灾难恢复 (DR) 的单一副本以及供升级和打补丁的单个数据库。如果将此原则应用于图 2 中表示的相对较小的整合环境，Oracle Multitenant 将通过减少要求管理的单独数据库的数量来实现 4 倍的优势。随着更多数据库整合到单个 CDB 中，这一优势将继续增加。

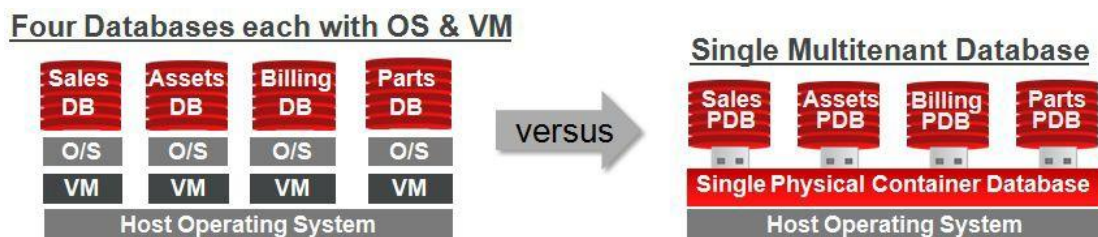


图 2：使用虚拟机的整合与使用 Oracle Multitenant 的数据库虚拟化比较

Oracle Multitenant 还能提供高度隔离。PDB 可以简单地从一个 CDB 中拔出并插入到另一个 CDB，以便允许数据库管理员根据需要选择对个别 PDB 执行维护工作。可以供应、打补丁、克隆、整合、恢复或移动个别 PDB 而不会影响同一 CDB 中的其他 PDB。

Oracle Multitenant 非常独特，它可实现其他整合方法的积极特质，同时又能避免它们的每个缺陷。Oracle Multitenant 可以实现：

- 虚拟机的简单性和灵活性，不限制整合密度、性能，也不增加管理复杂性
- 模式整合的高整合密度，且不存在实施复杂性，也不限制灵活性和隔离度
- 使用 Oracle RAC 与 Oracle Database 12c 达到的简单数据库整合的高可用性、高可伸缩性和自动负载管理，且不存在整合密度限制，也不存在对每个应用程序的单个数据库（每个数据库都有其自己的运营开销）进行管理的复杂性

Oracle Multitenant 与 Oracle 数据库的高可用性和数据保护功能无缝集成。此集成与 Oracle 最高可用性架构 (MAA) 最佳实践相结合，为数据库整合提供了一条通往革命性技术的进化升级之路。

Oracle Exadata — 针对整合和 DbaaS 进行了优化

[Oracle 工程化系统](#)是针对 Oracle 数据库实施了各种优化的一个系统体系。它们包括：Oracle 数据库云服务器、Oracle SuperCluster、Oracle 数据库机、Oracle 虚拟计算设备。Oracle 工程化系统通过在预先集成的优化平台上对 Oracle 数据库及 Oracle 支持的硬件和软件实施标准化来降低生命周期成本。

[Oracle 数据库云服务器](#)是专为 Oracle 数据库提供最佳性能和可管理性而打造的工程化系统。它可伸缩的架构和高级软件功能让它成为适宜作为整合和 DBaaS 的标准数据库平台的理想选择。

Exadata 存储软件的独特功能包括：

- 存储的数据库处理分流

通过将数据库表达式求值推送到存储单元进行处理，查询运行速度得以显著提高。仅仅满足谓词、指定列和声明列条件的行才会返回到数据库服务器，从而消除了到数据库服务器的无效数据传输。使用全配置的 Oracle 数据库云服务器，可以产生每秒 100 GB 的 SQL 数据吞吐量。

- 数据库优化存储

混合列压缩将读取密集型和存档数据压缩 10 倍或更多，从而降低整体存储占用空间要求。

- 数据库优化的 PCI 闪存

Exadata 智能闪存缓存透明地将经常访问的数据缓存到快速固态存储中，从而提高查询响应时间和吞吐量。由闪存（而非磁盘）提供服务的写入操作被称为“写回闪存缓存”。该缓存具有智能，它不缓存备份和不频繁的表扫描。从 Exadata 11.2.3.3 开始，Oracle Exadata Storage Server 软件根据读取对象的频率将表和分区扫描负载读取的对象自动缓存到闪存缓存中，而不会影响联机事务处理 (OLTP) 或让闪存缓存超负荷，同时还会压缩这些对象。PCI 闪存消除了磁盘控制器瓶颈。

Exadata 智能闪存日志在每个闪存磁盘上仅使用 512 MB 空间来提供一致的低延迟日志写入。一致的低延迟日志写入可以稳定并改进 OLTP 负载，甚至混合负载。

Exadata 智能闪存缓存压缩（Exadata 11.2.3.3 或更高版本）可以透明地压缩存入闪存缓存的用户数据，从而动态增加闪存缓存的逻辑容量。这样，便可以在闪存中存放更多数据，从而降低在磁盘驱动器上访问数据的需求。闪存中数据的 I/O 比磁盘上数据的 I/O 要快几个数量级。压缩和解压缩操作对应用程序和数据库来说完全透明，并且它们没有性能开销，即使以每秒数百万 I/O 的速率运行时也是如此。

- 数据库优化的全面资源管理

Oracle 数据库资源管理在 Oracle 数据库持的所有平台上均可用。例如，这些功能包括根据优先级保证 PDB 和数据库负载的 CPU，管理和分布运行并行操作，以及检测和控制失控查询，如[使用 Oracle 数据库资源管理器管理资源](#)中所述。但是，Exadata 支持额外的独特资源管理功能，而它们是支持 DBaaS 所必需的。这些功能包括 I/O 资源管理 (IORM) 和网络资源管理。

IORM 支持多个数据库、PDB 和数据库负载共享同一个存储，同时确保它们根据其优先级利用 I/O 资源。Oracle Exadata Storage Server 软件与 IORM 和 Oracle 数据库资源管理器一起确保满足客户定义的策略，即使在多个数据库、CDB 或 PDB 共享同一个数据库集群和存

储网格时也是如此。因此，一个数据库或应用程序服务不会以不可预测的方式独占 I/O 带宽和降低其他数据库的性能。从 Exadata 11.2.3.2.1 开始，IORM 在 Exadata 上最多支持 1024 个数据库。从 Exadata 12.1.1.1 开始，IORM 已扩展为支持 Oracle Database 12c 中的多租户架构中的各个 PDB。

从 Exadata 11.2.3.3 和 12.1.1.1 或更高版本开始，网络资源管理将自动并透明地排列关键数据库网络消息的优先级，从而确保不允许延迟的操作的快速响应。在数据库节点、数据库 InfiniBand 适配器、Oracle Exadata Storage Server 软件、Exadata 存储单元 InfiniBand 适配器和 InfiniBand 交换机中实施优先级排列，以确保在整个 InfiniBand 结构中进行优先级排列。对延迟敏感的消息（例如 Oracle RAC 缓存融合消息和关键的后台节点间通信）的优先级高于批处理、报告和备份消息。日志文件写入操作的优先级最高，以确保事务处理的低延迟，即使在存在大型批处理负载、备份、恢复读取和写入以及可能会让网络处于饱和状态的密集报告和查询时也是如此。

Exadata 的高级特性和独特的服务质量功能让它成为能为从应用程序到数据库再到网络和存储提供端到端优先级的唯一平台，以便智能地管理数据库整合环境中的负载和 DBaaS 负载。

有力证据 — 使用 Exadata 获得高整合密度

Oracle 进行了测试以验证 Exadata 是否能够比相似配置的非 Exadata 系统大幅实现更高的整合密度。使用完全相同的负载和 Oracle 数据库配置，在 Oracle 数据库云服务器上运行了两个系列的测试。

- 第一个测试系列禁用了所有特定于 Exadata 的特性，以提供比较基准。虽然它的目的是模拟非 Exadata 系统，但鉴于即使不启用 Exadata 特性的情况下 Exadata 系统也具有更高的 I/O 和网络带宽及稳定的性能特点，因此这种方法提供的比较基准较为保守。这种“非 Exadata”系统的性能远远超过了一般 Linux 系统。
- 第二个测试系列与第一个测试系列运行相同的负载并且使用相同的计算机，但这次启用所有仅 Exadata 提供的特性，包括 Exadata 智能闪存日志、智能闪存缓存、智能扫描、智能闪存缓存压缩、存储索引、网络资源管理器和 I/O 资源管理器。这样便可以评估 Exadata 独有的特性是否能够提高数据库负载的整合密度。
- Oracle Multitenant 未用于任一测试系列中，以便独显 Exadata 的独特数据库整合功能的价值。

测试结果

图 3 中提供的汇总结果回答了以下两个基本问题：

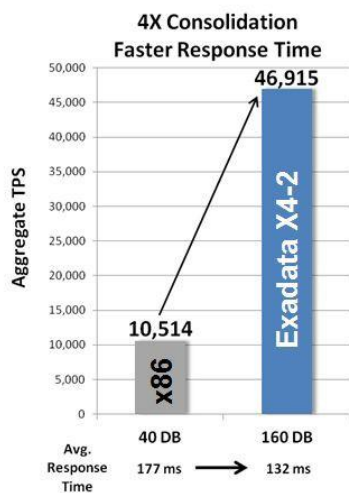
- 与相似配置的非 Exadata x86 系统相比，在 Exadata 系统上可以更多整合多少数据库？

第一个测试模拟了 OLTP 负载，整合的数据库数量逐渐增加，直到达到瓶颈。Exadata 提供了更快的响应时间，同时由于相比同等 X86 硬件上的仅 40 个数据库（受 I/O 限制）能够支持 160 个数据库（受 CPU 限制），因此展示了 4 倍的整合密度。在性能服务级别协议允许超载的整合环境中，Exadata 系统的整合密度优势增加为相似配置的 x86 系统的 5 倍。

- 与相似配置的 x86 系统相比，Exadata 运行整合环境通常所具有的混合负载的速度可以快多少？

此测试中模拟的负载包括 OLTP 负载和报告数据仓库。测试显示，Exadata 具有 15 倍的事务响应速度，2 倍的整合密度以及超过 6 倍的事务量。这项测试附带的一个有趣方面是，非 Exadata 系统在 40 个数据库时出现了 I/O 瓶颈，而 Exadata 系统有额外的 I/O 和 CPU 容量来应对负载中的峰值或整合其他数据库。

How many more databases can be consolidated on Exadata?



How much faster can Exadata execute mixed workloads?

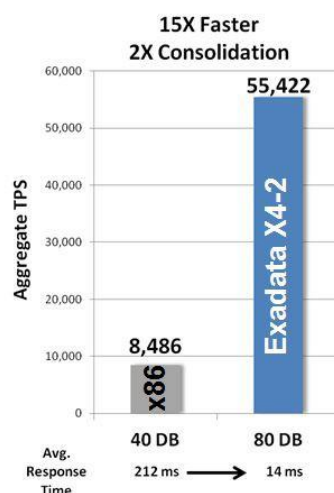


图 3：Exadata 整合密度和性能

整合密度的增加让 Exadata 可以产生巨大的成本优势：购买更少的硬件、管理更少的系统、能耗降低并且需要的 Oracle 数据库许可也更少。

在[附录 A](#) 中提供了本节所描述测试所使用的方法、配置和结果的详细信息。附录 A 还从一组扩展的测试方案提供了其他发现。

高可用性参考架构

MAA 最佳实践定义了四个高可用性服务级别分层来解决所有规模和行业的企业所需的全系列可用性和数据保护。这些分层指定为白金级、黄金级、白银级和青铜级。它们分别提供图 4 所述的服务级别。

每一层都使用不同的 MAA 参考架构，以部署最佳的一组 Oracle 高可用性功能，从而让这些功能以最低的成本针对停机和潜在数据丢失可靠地实现给定的服务级别。每一层都能解决所有类型的意外中断，包括数据损坏、组件故障、系统和站点中断以及计划内维护。每一层针对最大允许停机的服务级别指定为恢复时间目标 (RTO)。每一层针对最大允许数据丢失的服务级别指定为恢复点目标 (RPO)。

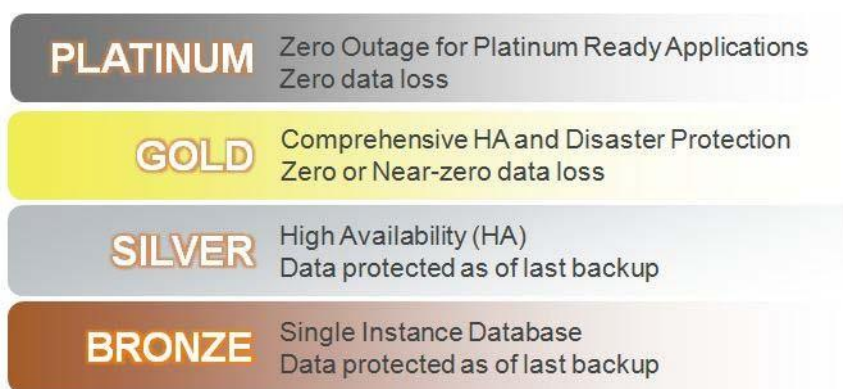


图 4：高可用性服务级别分层

高可用性分层独立于选择的整合方法或部署模型。但是，这不妨碍将归于 Oracle Multitenant 的高可用性功能在适当的情况下包含在讨论中。例如，如果需求发生变化，则在升级 Oracle 数据库时，或者将数据库（多租户架构中的 PDB）从一个高可用性层移到另一个高可用性层时，简单拔出/插入操作提供的灵活性在减少计划内停机和管理开销方面带来了价值。

参考架构作为一个指南进行提供。Oracle 认为，对数据库层的四个架构进行标准化所获得的益处将超出部署一个较大集合带来的益处。但是，Oracle 高可用性功能的灵活性让客户能够自定义四个层中的任何一个，以满足他们独一无二的业务需求。表 1 概要描述了每一层实现的各种中断类别的服务级别。

表 1：按层划分的高可用性和数据保护属性

中断类别/高可用性层	计划外中断（本地站点）	计划内维护	数据保护	不可恢复的本地中断和灾难恢复
白金级	白金级应用零应用停机	零应用停机	全面运行时验证和手动检查相结合	白金级应用零应用停机、保留运行中事务、零数据丢失
黄金级	全面的高可用性/灾难恢复	全部滚动或联机处理	全面运行时验证和手动检查相结合	实时故障切换、零或近乎为零的数据丢失
白银级	具有自动故障切换的高可用性	有些滚动、有些联机、有些脱机处理	基本运行时验证和手动检查相结合	从备份中恢复，可能丢失自上次备份以来生成的数据
青铜级	单实例、自动重启可恢复实例和服务器故障	有些联机，大多数脱机处理	基本运行时验证和手动检查相结合	从备份中恢复，可能丢失自上次备份以来生成的数据

后面的几部分将详细介绍每一层的高可用性架构和服务级别。

青铜级 — 单实例

青铜级层以最低绝对成本提供基本数据库服务。为了降低成本和实施的复杂性，接受降低的高可用性和数据保护级别作为交换。图 5 对青铜级层进行了概述。

青铜级使用单实例 Oracle 数据库；如果运行 Oracle 数据库实例的服务器出现中断，没有任何集群技术可用于自动故障切换。当服务器变得不可用或数据库无法恢复时，RTO 受制于更换系统供应的速度或备份恢复的速度。在整个站点中断的最坏情况下，可能需要额外的时间才能在辅助位置执行这些任务，并且在某些情况下可能需要几天。

使用 Oracle Recovery Manager (RMAN) 来执行 Oracle 数据库定期备份。如果出现无法恢复的中断，RPO 便只能恢复到自上一次备份以来生成的数据。还需要在远程位置或云上保留数据库备份的副本以实现存档和灾难恢复的双重目的，以防灾难降临到主数据中心。

Bronze Tier: Single Instance

RTO of Minutes to Days, RPO From Last Backup

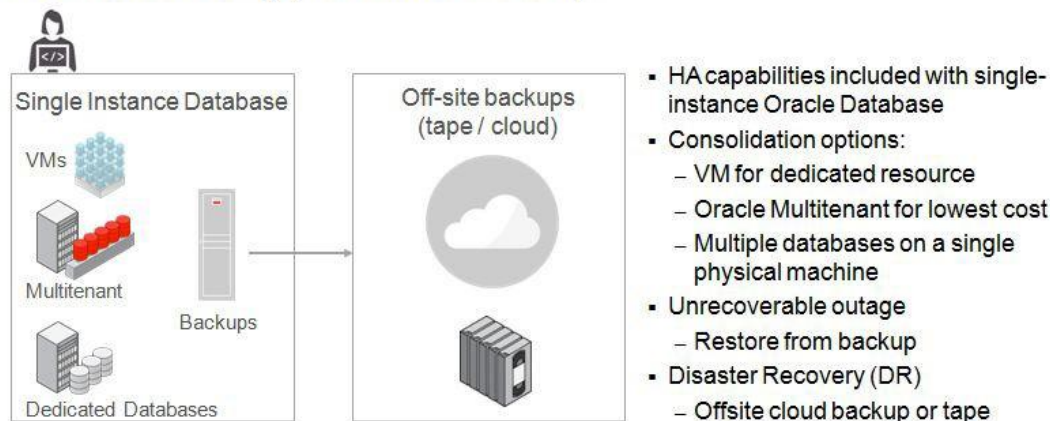


图 5：青铜级层高可用性参考架构

青铜级层由以下各节中描述的主要组件组成。

Oracle 数据库高可用性和数据保护

青铜级利用 Oracle 数据库企业版包含的高可用性和数据保护功能，无需额外的成本，这些功能包括

- 在硬件或软件故障后，或者不管任何时候重新启动数据库主机时，[Oracle Restart](#) 自动重启数据库、监听器和其他 Oracle 组件。
- Oracle 损坏保护检查有无物理损坏和逻辑块内部损坏。检测内存中损坏并防止将它写入磁盘，在许多情况下均可自动修复。有关更多详细信息，请参阅[防止、检测并修复块损坏：Oracle Database 11g](#)。
- [自动存储管理](#) (ASM) 是一个 Oracle 集成文件系统和卷管理器，它包括本地镜像以防止磁盘故障。
- [Oracle 闪回技术](#) 以适用于修复单个事务、表或完整数据库的粒度级别提供快速纠错。
- [Oracle Recovery Manager](#) (RMAN) 支持针对 Oracle 数据库优化的低成本且可靠的备份和恢复。
- [联机维护](#) 包括联机重新定义和重新组织数据库维护、联机文件移动和联机打补丁。

数据库整合

在青铜级层部署的数据库包括开发和测试数据库，以及支持较小工作组和部门应用程序（通常是数据库整合和 DBaaS 的第一候选）的数据库。

[Oracle Multitenant](#) 是从 Oracle Database 12c 开始的 MAA 数据库整合最佳实践。

生命周期管理和 DBaaS

[Oracle Enterprise Manager Cloud Control](#) 让业务用户可以自助部署 IT 资源，还提供了可满足各种多租户架构需求的资源池模型。需要这些功能才能实施数据库即服务 (DBaaS)，后者是一种范式，最终用户（数据库管理员、应用程序开发人员、质量保证工程师、项目负责人等）可以通过它来请求数据库服务，在项目生命周期内对其加以利用，然后在使用后自动取消数据库服务供应并将其返回资源池。Cloud Control 数据库即服务 (DBaaS) 提供：

- 一个用于供应数据库服务的共享整合平台
- 一个用于供应这些资源的自助式模型
- 自由伸缩数据库资源的弹性
- 基于数据库使用的计费方式

Oracle 工程化系统

Oracle 工程化系统在所有层上均是数据库整合和 DBaaS 的一个高效部署选项。Oracle 工程化系统通过在预先集成的优化平台上对 Oracle 数据库及 Oracle 支持的硬件和软件实施标准化来降低生命周期成本。Oracle 工程化系统包括：

- [Oracle 虚拟计算设备](#)从根本上简化了客户为任何 Linux、Oracle Solaris 或 Microsoft Windows 应用程序安装、部署和管理虚拟基础架构的方式。
- [Oracle 数据库机](#)是一个包含软件、服务器、存储和网络的完整低成本系统，旨在通过简化对数据库和应用程序负载的部署、维护和支持来提供简单性并节省时间和金钱。Oracle 数据库机同时支持物理和虚拟部署。
- [Oracle 数据库云服务器](#)是运行 Oracle 数据库的性能最高、最具有伸缩性且可用性最强的平台。Oracle 数据库云服务器可运行所有类型的数据库负载，包括联机事务处理 (OLTP)、数据仓储 (DW) 以及各种混合负载，并且它是数据库整合的理想基础。
- [Oracle SuperCluster](#) 工程化系统是在单个通用平台上整合数据库和应用程序、私有云部署和 Oracle 软件的理想选择。Oracle SuperCluster 使用基于 SPARC 架构的最快的处理器和 Exadata 存储。
- [Oracle ZFS 存储设备](#)使用网络连接存储 (NAS) 为客户提供立竿见影的空间、管理和成本优势。Oracle ZFS 包括一个可执行管理、监视、故障排除、快照、克隆、复制和高级数据服务（这些都是对所有 Oracle 工程化系统的一个自然补充）的丰富软件套件。

青铜级总结：数据保护、RTO 和 RPO

表 2 总结了青铜级层的数据保护功能。表 2 的第一列表示执行物理和逻辑损坏验证的时间。

- 管理员启动手动检查，或者由执行定期检查的调度作业定期启动。
- 在数据库打开时，由后台进程连续地自动执行运行时检查。
- 以定期计划间隔运行后台检查，但是仅限在资源处于空闲状态期间运行。

每一项检查均为使用特定 Oracle 数据块和重做结构知识的 Oracle 数据库所独有。

表 2：青铜级层数据保护

类型	功能	物理块损坏	逻辑块损坏
手动	Dbverify、分析	物理块检查	逻辑检查块内和对象间一致性
手动	RMAN	在备份和恢复期间进行物理块检查	块内逻辑检查
运行时	数据库	内存中块和重做校验和	内存中块内逻辑检查
运行时	ASM	使用本地范围对自动进行损坏检测和修复	
运行时	Exadata	写入时进行 HARD 检查	写入时进行 HARD 检查
后台	Exadata	自动硬盘清理和修复 ²	

请注意，HARD 验证与自动硬盘清理和修复（表 2 的最后两行）是 Exadata 存储所独有的。HARD 验证确保 Oracle 数据库不会以物理方式将损坏的块写入到磁盘。在出现空闲资源时，自动硬盘清理和修复将定期检查并修复具有受损或已坏磁盘扇区（存储簇）或具有其他物理或逻辑缺陷的硬盘。Exadata 向 ASM 发送请求，以通过从另一个镜像副本读取数据来修复坏扇区。默认情况下，每两周运行一次硬盘清理。

² 在 Exadata 11.2.3.3 和更高版本以及 Oracle Database 11g 第 2 版 (11.2.0.4) 和更高版本中提供。

表 3 总结了各种计划外和计划内中断的青铜级层的 RTO 和 RPO。

表 3：青铜级层恢复时间 (RTO) 和数据丢失可能 (RPO)

类型	事件	停机时间	数据丢失可能
计划外	数据库实例故障	数分钟	零
计划外	可恢复的服务器故障	数分钟至一小时	零
计划外	数据损坏、不可恢复的服务器故障、数据库故障或 站点故障	数小时至数天	自上次备份
计划内	联机文件移动、联机重新组织和重新定义、联机 打补丁	零	零
计划内	无法联机执行的硬件或操作系统维护和 数据库打补丁	数分钟至数小时	零
计划内	数据库升级：补丁集和完整数据库版本	数分钟至数小时	零
计划内	平台迁移	数小时至一天	零
计划内	修改后端数据库对象的应用程序升级	数小时至数天	零

白银级 — 具有自动故障切换的高可用性

白银级层建立在青铜级基础上，它并入了集群技术以提高计划外中断和计划维护的可用性。白银级使用 Oracle RAC 或 Oracle RAC One Node 来实现数据中心内部的高可用性，方法是在出现不可恢复的数据库实例中断或运行实例所在的整个服务器出现故障时提供自动故障切换。Oracle RAC 还具备了避免多种计划内停机的实质优势，方法是在 Oracle RAC 节点之间滚动执行维护。图 6 对白银级层进行了概述。

白银级包括以下各部分描述的高可用性组件。

Oracle RAC

Oracle RAC 提高了数据库实例或运行数据库实例的服务器出现中断时数据中心的应用程序可用性。Oracle RAC 的服务器故障切换瞬间完成。在正常运行的实例上继续运行服务之前会出现一个非常短暂的中断，故障实例中的用户能够重新连接。另外还消除了计划维护任务造成的停机，可以在 Oracle RAC 节点之间以滚动方式执行这些任务。用户在执行维护的节点上完成他们的工作并终止会话。当他们重新建立连接时，他们被定向到已经在另一个节点上运行的某个数据库实例。

大致了解 Oracle RAC 的工作方式有助于了解其优势。有以下两个组件：Oracle 数据库实例和 Oracle 数据库自身。

- **数据库实例** 定义为在单个节点（或服务器，它让特定数据库可用于客户端）上运行的一组服务器进程和内存结构。
- **数据库** 是驻留在持久存储中且可以一起打开并用于读取和写入数据的一组特定共享文件（数据文件、索引文件、控制文件和初始化文件）。
- Oracle RAC 使用主动-主动架构，使多个数据库实例（每个都运行在不同的节点上）可以同时读取和写入同一个数据库。

Silver Tier: HA with Automatic Failover

RTO of Seconds for Server Failures, Worst-Case RPO From Last Backup

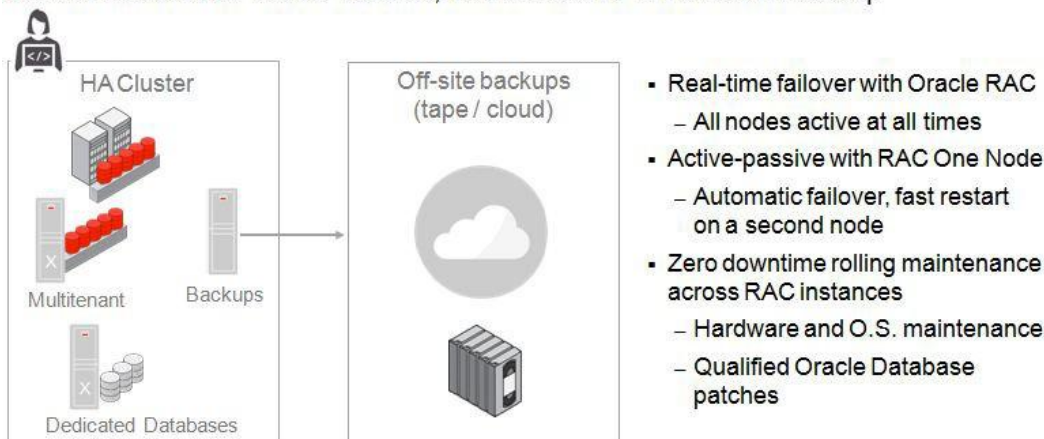


图 6：白银级层高可用性参考架构

Oracle RAC 的主动-主动架构提供诸多优势：

- **增强的高可用性**：如果服务器或数据库实例发生故障，则与正常实例的连接不受影响；与故障实例的连接将快速故障切换至集群中其他服务器已打开和运行的正常实例。
- **可伸缩性**：Oracle RAC 是大容量应用程序或整合环境的理想选择，在这些环境中，需要跨多个服务器动态添加容量或重新调整容量优先级的可伸缩性和能力。单个数据库可以让实例运行在集群的一个或多个节点上。同样，可以在一个或多个数据库实例上提供数据库服务。可以联机供应额外的节点、数据库实例和数据库服务。在集群中轻松分发负载的能力让 Oracle RAC 成为 Oracle Multitenant 的理想补充。
- **可靠的性能**：Oracle 服务质量 (QoS) 可用于为高优先级的数据库服务分配容量，以便在数据库整合环境中提供一致的高性能。可以在负载之间动态转移容量，以便快速响应不断变化的需求。
- **计划维护期间的高可用性**：通过在 Oracle RAC 节点之间以滚动方式实施更改来保持高可用性。这包括：要求服务器脱机的硬件、操作系统或网络维护；为 Oracle Grid Infrastructure 或数据库打补丁的软件维护；或者，如果数据库实例需要移动到其他服务器，以增加容量或平衡负载。

Oracle RAC 是实现服务器高可用性的推荐的 MAA 最佳实践。

Oracle RAC One Node

在白银级层，如果要求服务器高可用性,但不要求可伸缩性和即时故障切换，那么可以选择 Oracle RAC One Node 代替 Oracle RAC。Oracle RAC One Node 许可费只有 Oracle RAC 的一半，如果数分钟的 RTO 对管理服务器故障来说已足够，则它是一个成本较低的替代选择。

Oracle RAC One Node 是一种主动-被动故障切换技术。它建立在与 Oracle RAC 完全相同的基础架构上，但对于 Oracle RAC One Node，在正常操作期间一次只打开一个数据库实例。这样会显著减少内存需求，尤其是在整合大量数据库时。如果托管已打开实例的服务器出现故障，则 Oracle RAC One Node 在第二个节点上自动启动一个新的数据库实例，以迅速恢复服务。

Oracle RAC One Node 相比其他主动-被动集群技术具有多个优势。在 Oracle RAC One Node 配置中，Oracle 数据库高可用性服务、网络基础架构和数据库许可始终在第二个节点上运行。在故障切换时，仅需启动数据库实例和数据库服务，从而减少恢复服务所需的时间，让服务在数分钟内得以恢复。图 7 展示了 Oracle RAC One Node 故障切换相对于其他主动-被动故障切换解决方案所具有的优势。在此例中，Oracle RAC One Node 与虚拟机高可用性进行比较。

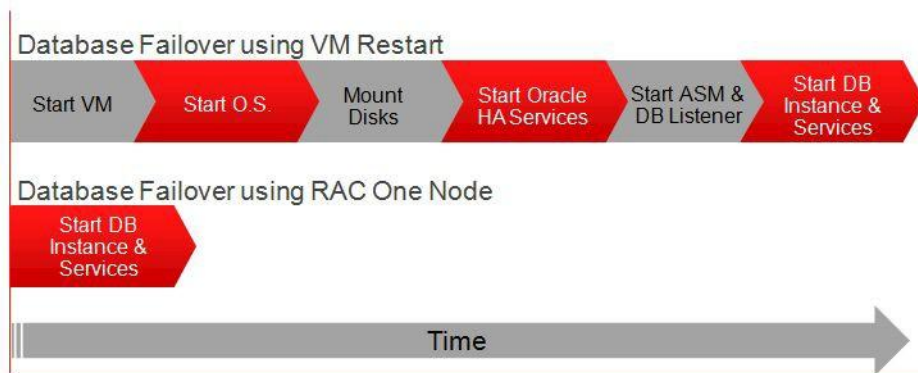


图 7: Oracle RAC One Node 与虚拟机高可用性比较

对于计划维护，Oracle RAC One Node 与 Oracle RAC 提供同样的优势。在计划维护期间，RAC One Node 允许出现两个主动数据库实例，以便无需停机就可以将用户从一个节点从容地迁移到另一个节点。在节点之间以滚动方式执行维护，同时始终保持数据库服务对用户可用。

白银级层总结：数据保护、RTO 和 RPO

与青铜级层相比，数据保护级别没有任何变化。相比青铜级，白银级提供的所有改进均涉及服务器中断和几种频繁执行的计划维护的 RTO。表 4 总结了白银级层支持的 RTO 和 RPO。与青铜级相比有所改进的方面以红色突出显示。

表 4：白银级层恢复时间 (RTO) 和数据丢失可能 (RPO)

类型	事件	停机时间	数据丢失可能
计划外	数据库实例故障	数秒（如果是 RAC） 相对于数分钟	零
计划外	可恢复的服务器故障	数秒（如果是 RAC） 相对于数分钟至一小时 数分钟（如果是 RAC One Node） 相对于数分钟至一小时	零 零
计划外	数据损坏、不可恢复的服务器故障、数据库或站点 故障	数小时至数天	自上次备份
计划内	联机文件移动、联机重新组织和重新定义、联机打补丁	零	零
计划内	无法联机执行但是符合 Oracle RAC 滚动安装资格的硬件或操作系统维护 and 数据库打补丁	零 相对于数分钟至数小时	零
计划内	数据库升级：补丁集和完整数据库版本	数分钟至数小时	零
计划内	平台迁移	数小时至一天	零
计划内	修改后端数据库对象的应用程序升级	数小时至数天	零

黄金级 — 全面的高可用性/灾难恢复

黄金级层建立在白银级的基础上，它使用数据库复制技术消除单点故障，并且为所有类型的计划外中断提供高得多的数据保护和高可用性级别，包括数据损坏、数据库故障和站点故障。复制副本的存在还为在计划维护期间减少停机提供了实质性优势。在图 8 中概述了黄金级层。RTO 降低到了数秒或数分钟，伴随的 RPO 为零或近乎为零，具体取决于实际配置。

请注意，黄金级使用 Oracle RAC 作为服务器高可用性的标准，代替了白银级提供的性能较低的 Oracle RAC One Node。

Gold Tier: Comprehensive HA/DR

RTO of Seconds to Minutes, RPO of Zero or Near-Zero

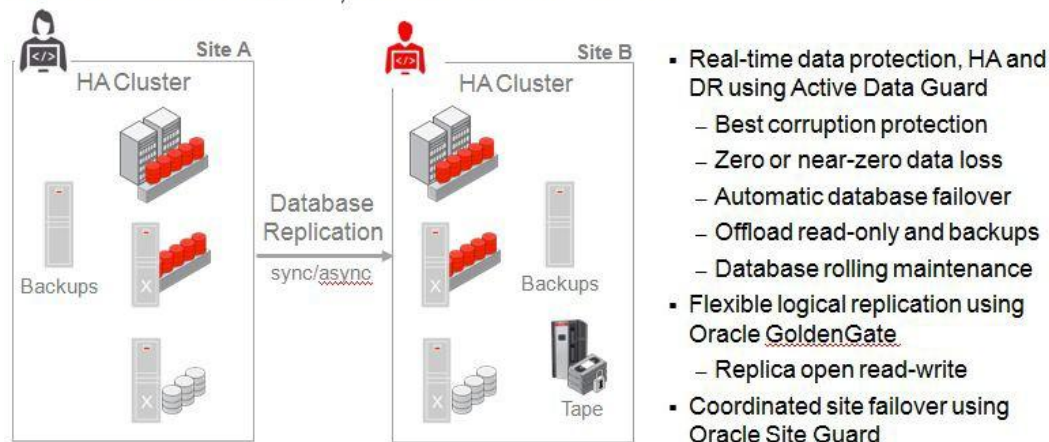


图 8：黄金级层高可用性参考架构

黄金级层添加了高级高可用性组件，以实现在以下部分中描述的改进的服务级别。

Oracle Active Data Guard — 实时数据保护和可用性

[Oracle Active Data Guard](#) 在远程位置维护一个或多个同步物理副本（备用数据库），用于消除生产数据库（主数据库）的单点故障。Oracle Active Data Guard 为黄金级层添加的功能包括：

- 选择零或近乎为零的数据丢失可能。Oracle Active Data Guard 实时执行将更改从主数据库复制到备用数据库。直接从主数据库的日志缓冲区传输更改，以便最大程度降低传播延迟和开销，并且将复制与可能在生产数据库的 I/O 体系中发生的损坏完全隔离。

管理员可以选择同步传输和最高可用性，以保证零数据丢失。或者，他们可以选择异步传输和最高性能，以实现近乎为零的数据丢失。当提供的网络带宽足以容纳传输量时，最高性能可以实现亚秒级数据丢失风险。

Data Guard 是提供零数据丢失保护的唯一一个 Oracle 复制技术。

- 如果发生影响主数据库可用性的数据库或站点中断，Oracle Active Data Guard 备用数据库可以快速接管生产数据库并恢复服务。Oracle 数据库始终处于运行状态，不需要重新启动它即可转换为主数据库角色，并且角色转换可以在不到 60 秒的时间内完成，即使是在负载较重的系统中。

黄金级利用 Data Guard 快速启动故障切换自动完成数据库故障切换。这样将消除通知管理员和响应中断所需的延迟，从而加速恢复时间。快速启动故障切换使用角色特定的数据库服务和 Oracle 客户端通知框架来确保应用程序快速与出现故障的主数据库断开连接，并自

动重新连接到新的主数据库。还可以使用命令行界面或 Oracle Enterprise Manager 手动执行角色转换。

- 透明复制。Oracle Active Data Guard 执行 Oracle 数据库的完整单向物理复制，具有以下特点：高性能，易于管理，支持所有数据类型、应用程序和负载，例如 DML、DDL、OLTP、批处理、数据仓库和整合数据库。Oracle Active Data Guard 与 Oracle RAC、ASM、RMAN 和 Oracle 闪回技术紧密集成。
- 生产数据库分流以实现高投资回报 (ROI)。Oracle Active Data Guard 备用数据库可以在复制处于活动状态时以只读方式打开，可用于分流生产数据库中的即席查询和报告负载。这种分流将利用空闲容量，增加备用系统的 ROI，并提高所有负载的性能。它也提供连续应用程序验证，因为备用系统已准备好支持生产负载。
- 备份分流。主系统和备用系统都是物理副本，从而可以将备份工作从主数据库分流到备用数据库。在备用数据库进行的备份可用于恢复主数据库或备用数据库。这将为管理员提供灵活的恢复选择，而不会因执行备份的开销给生产系统增加负担。
- 减少了计划维护所需的停机。备用数据库可用于以滚动方式升级到新 Oracle 补丁集（例如，补丁版本 11.2.0.2 到 11.2.0.4）或新 Oracle 版本（例如，版本 11.2 到 12.1），方法是首先在备用数据库上实施升级，然后将生产数据库切换为新版本。总停机时间限制在维护完成后将备用数据库切换到主生产数据库角色所需的时间。
- Oracle Active Data Guard 备用数据库执行连续 Oracle 验证，以确保不会传播源数据库中的损坏。它将检测可能在主数据库或备用数据库上独立发生的物理和逻辑块内部损坏。它在支持静默写入丢失损坏（I/O 子系统确认为成功的丢失或偏离写入）的运行时检测方面也独一无二。有关更多详细信息，请参阅 My Oracle Support 说明 1302539.1 — [损坏检测、预防和自动修复最佳实践](#)。
- 自动块修复。Oracle Active Data Guard 自动修复可能独立发生于主数据库或备用数据库上的间歇性随机 I/O 错误导致的块级损坏。它从相对的数据库检索该块的好副本来执行修复。不需要进行任何应用程序更改，并且修复对用户来说是透明的。

上面的几点解释了黄金级层如何利用 Oracle 复制技术维护同步副本，而不是使用存储远程镜像产品（例如 SRDF、Hitachi TrueCopy 等）。有关这些差异的更深入讨论，请参阅 [Oracle Active Data Guard 与存储远程镜像](#)。

Oracle GoldenGate

Oracle GoldenGate 提供了逻辑复制选择，以维护生产数据库（源数据库）的同步副本（目标数据库）。逻辑复制是一个比物理复制更为复杂的过程，但是它提供了处理不同复制方案和异构平台的更大灵活性。

- 从数据分布的角度看，逻辑复制的设计目的是高效复制源数据库的子集，以便将数据分发到其他目标数据库。它也可以用于将多个源数据库中的数据整合到单个目标数据库（例如运营数据存储）。
- 从高可用性的角度看，逻辑复制可用于为高可用性或灾难保护维护源数据库的完整副本，以便在源数据库变得不可用时随时准备好立即进行故障切换。Oracle GoldenGate 使用逻辑复制过程。它读取源数据库磁盘中的更改，将数据转换为独立于平台的文件格式，将文件传输到目标数据库，然后将数据转换为目标数据库原生的 SQL（更新、插入和删除）。目标数据库包含相同的数据，但是它是不同于源数据库的数据库（例如，备份不可互换）。
- Oracle GoldenGate 逻辑复制为以滚动方式执行维护和迁移（使用 Data Guard 物理复制无法这样做）提供了更大的灵活性。例如，Oracle GoldenGate 支持从运行于大字节序平台的源数据库复制到运行于小字节序平台的目标数据库（跨字节序复制）。这样便让执行平台迁移具有额外的优势，从而能够反向复制操作，以便在切换后快速回退到之前的版本。

Oracle GoldenGate 逻辑复制是一个更为复杂的过程，它有许多不适用于 Data Guard 物理复制的先决条件。作为这些先决条件的回报，Oracle GoldenGate 提供了应对高级复制要求的独特功能。有关每种复制技术的利弊权衡的其他见解，以及可能偏向于使用一个而不是另一个，或者以互补方式同时使用两种技术的需求，请参阅 [MAA 最佳实践：Oracle Active Data Guard 和 Oracle GoldenGate](#)。

Oracle Site Guard

[Oracle Site Guard](#) 让管理员能够在生产站点和远程灾难恢复站点之间编排其 Oracle 环境、多个数据库和应用程序的转换（计划事件）和故障切换（响应意外中断）。Oracle Site Guard 包含在 Oracle Enterprise Manager 生命周期管理包中。

Oracle Site Guard 提供以下优势：

- 由于准备好了响应站点故障而减少了错误。Oracle Site Guard 减少了在发生灾难时出现人为错误的可能性。在应用程序中准备的响应中筹划、测试并演练恢复战略。在管理员为灾难恢复发起 Site Guard 操作后，不需要人工干预。
- 协调多个应用程序、数据库和各种复制技术。在启动或停止站点时，Oracle Site Guard 自动处理不同目标之间的相关性。Site Guard 与 Oracle Active Data Guard 集成以协调多个并发的

数据库故障切换。Site Guard 还提供一个与任何存储远程镜像产品相集成的轻松机制。它与存储设备相集成，通过在操作工作流程中向用户指定的任何存储角色对调脚本使用调出，从而执行转换或故障切换。

- 更快的恢复时间。Oracle Site Guard 自动化最大程度减少了恢复活动的手动协调。这样，即使与成功执行所有手动操作的情况相比，也会加速恢复。Site Guard 还可以避免解决人为错误的耗时操作，因为这些错误经常是伴随复杂过程的手动实施而产生的。

黄金级总结：数据保护、RTO 和 RPO

表 5 总结了黄金级层提供的数据保护。黄金级和白银级之间的数据保护差异（标注为红色）直接归因于 Oracle Active Data Guard。

表 5：黄金级层数据保护

类型	功能	物理块损坏	逻辑块损坏
手动	Dbverify、分析	物理块检查	逻辑检查块内和对象间一致性
手动	RMAN	在备份和恢复期间进行物理块检查	块内逻辑检查
运行时	Oracle Active Data Guard	在备用数据库进行物理块检查 主数据库和备用数据库之间的强隔离消除了单点故障 自动修复物理损坏 自动数据库故障切换	检测写入丢失损坏，自动关闭和故障切换 在备用数据库进行块内逻辑检查
运行时	数据库	内存中块和重做校验和	内存中块内逻辑检查
运行时	ASM	使用本地范围对自动进行损坏检测和修复	
运行时	Exadata	写入时进行 HARD 检查	写入时进行 HARD 检查
后台	Exadata	自动硬盘清理和修复	

表 6 总结了黄金级层的 RTO 和 RPO。与白银级相比，黄金级层明显降低了恢复时间和数据丢失可能。与白银级层相比有所改进的方面以红色突出显示。

表 6: 黄金级层恢复时间 (RTO) 和数据丢失可能 (RPO)

类型	事件	停机时间	数据丢失可能
计划外	数据库实例故障	数秒	零
计划外	可恢复的服务器故障	数秒	零
计划外	数据损坏、不可恢复的服务器故障、数据库故障或站点故障	零至数分钟 相对于数小时至数天	近乎为零（如果使用 ASYNC） 相对于自上次备份 零（如果使用 Data Guard 同步传输） 相对于自上次备份
计划内	联机文件移动、联机重新组织和重新定义、联机打补丁	零	零
计划内	无法联机执行但是符合 RAC 滚动安装资格 的硬件或操作系统维护和数据库打补丁	零	零
计划内	数据库补丁集或完整版本升级	数秒 相对于数分钟至数小时	零
计划内	平台迁移	数秒 相对于数小时至一天	零
计划内	修改后端数据库对象的 应用程序升级	数小时至数天	零

白金级 — 白金级应用程序零停机

白金级层建立在黄金级基础上，它使用最新的 Oracle Database 12c 功能，从而为对中断或数据丢失零容忍的应用程序提供最高级别的高可用性和数据保护。白金级屏蔽了中断对应用程序和用户的影响，从而确保甚至在可恢复的故障后保留运行中的事务。它支持零停机维护、迁移和应用程序升级。在主数据库由于任何原因而发生故障时，它保证零数据丢失，不管站点之间的距离如何。最后，白金级还自动管理数据库服务的可用性以及多个站点中的数据库副本之间的负载平衡。图 9 概述了白金级层。

Platinum Tier

Zero Application Outage for Platinum Ready Applications

ORACLE
DATABASE 12^c

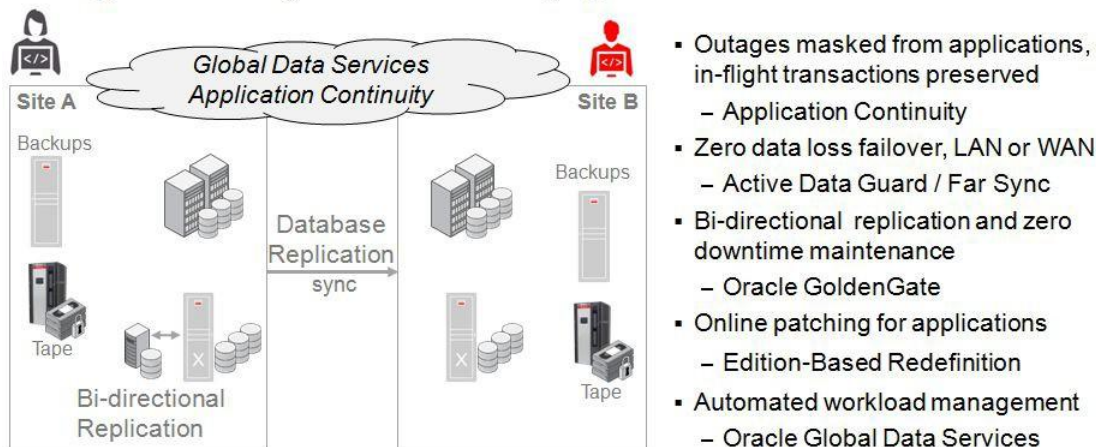


图 9：白金级层高可用性参考架构

白金级应用程序提供的功能

某些应用程序需要具有某种修改级别才能使用白金级层提供的功能而实现零应用停机。这解释了为什么将白金级描述为向白金级应用程序提供零应用停机。请注意，不必做出任何修改即可实现零数据丢失。

白金级层支持以下各部分描述的高可用性功能。

应用程序连续性

[应用程序连续性](#)保护应用程序免受由实例、服务器、存储、网络或任何其他相关组件（甚至是整个数据库故障）造成的数据库会话故障的影响。应用程序连续性重放受故障影响的“运行中”请求，以便在应用程序看来该故障只是一次稍有延迟的执行，从而对用户屏蔽了中断。

如果整个 Oracle RAC 集群出现故障，从而使数据库不可用，则应用程序连续性将在 Oracle Active Data Guard 故障切换后重放会话，包括事务。将应用程序连续性用于备用数据库要求采用 Data Guard 最高可用性模式（零数据丢失）和 Data Guard 快速启动故障切换（自动数据库故障切换）。

虽然在许多情况下，需要对现有应用程序代码进行某些修改才能使用应用程序连续性，但是通过透明地处理可恢复故障，它简化了新应用程序的开发。

Oracle Active Data Guard 远程同步

Oracle Active Data Guard 是为 Oracle 数据库提供零数据丢失故障切换的唯一一个 Oracle 感知复制技术。在 Data Guard 最高可用性模式下使用同步传输实现零数据丢失。使用同步传输时，主站点和备用站点间的网络延迟将影响数据库性能。随着站点间的距离增加，延迟及对

数据库性能的影响也会增加。因为主数据中心和备用数据中心经常长距离分隔，所以对于许多数据库来说，实现零数据丢失故障切换不切实际。

Oracle Database 12c 的 Oracle Active Data Guard 远程同步消除了之前的限制，即使主数据库和备用数据库相隔数百或数千英里，它也支持零数据丢失故障切换，而不会影响主数据库的性能。它使用易于部署且对 Oracle Active Data Guard 故障切换或转换操作透明的轻型转发机制实现这一点。将远程同步与 Oracle Advanced Compression 选件结合使用时，还支持主机外传输压缩，以节省网络带宽。

将远程同步与 Data Guard 快速启动故障切换（自动数据库故障切换）相结合，应用程序连续性便可以屏蔽运行中事务的中断，不管主站点和备用站点之间的距离有多远。因此，远程同步支持白金级层提供的两个重要增强：任何数据库的零数据丢失故障切换，以及使用应用程序连续性的能力（不管站点之间的距离有多远）。不需要做出任何应用程序修改即可利用远程同步。

Oracle GoldenGate 零停机维护和主动-主动复制

白金级层使用 Oracle GoldenGate 的高级复制功能，通过双向复制来实施零停机维护和迁移。在这种情况下：

- 首先在目标数据库上实施维护。
- 使用 Oracle GoldenGate 逻辑复制在各版本之间同步源和目标。这将处理跨字节序平台迁移。它也处理复杂应用程序升级，这些升级修改后端对象，从而复制机制必须能够将数据从旧版本转换为新版本，反之亦然。
- 在新版本或平台经过同步并稳定后，双向复制能够在用户终止之前版本上的会话并重新连接时将其逐渐迁移到新平台，从而提供零停机体验。在迁移过程中，Oracle GoldenGate 双向复制会保持旧版本和新版本的同步。这样也会在新版本添加负载时出现任何意外问题的情况下，提供快速回退选择。

在需要与同一数据的多个副本进行连续读写连接时，也可以使用主动-主动双向复制增加可用性服务级别。

双向复制不对应用程序透明。在同时对多个数据库的相同记录做出更改时，它要求进行冲突检测并解决之。它还要求仔细考虑不同故障状态和复制延迟的影响。将 GoldenGate 双向复制用于修改后端数据库对象的应用程序升级时，要求对新版本修改或添加的数据库对象具有开发人员层次的了解，以便让 GoldenGate 能够在各版本之间复制。要求为每个新应用程序版本实施跨版本映射。

就定义看，GoldenGate 复制是一个异步过程，它不能提供零数据丢失保护。因此，当远程副本必须提供零数据丢失保护时，如果主数据库或主站点遇到意外中断，白金级层不使用 Oracle GoldenGate 在站点之间复制。白金级结合使用 Oracle Active Data Guard 和 GoldenGate 双向复制，以满足零数据丢失要求。使用本地 GoldenGate 副本来零停机执行计划维护，同时

Oracle Active Data Guard 备用数据库提供连续的零数据丢失故障切换保护，以防在进行维护的过程中出现意外中断。

基于版本的重新定义

[基于版本的重新定义](#) (EBR) 支持在不中断应用程序可用性的情况下而修改后端数据库对象的联机应用程序升级。完成升级安装后，升级前应用程序和升级后应用程序可以同时使用。现有会话可以继续使用升级前的应用程序，直到其用户决定终止它们，而所有新会话可以使用升级后的应用程序。当不再有任何会话使用升级前应用程序时，升级前应用程序便可以报废了。通过以这种方式使用 EBR，便可以从升级前版本热翻滚到升级后版本。

EBR 通过以下方式支持联机应用程序升级：

- 代码更改在新版本中暗自安装。
- 通过仅写入旧版本看不到的新列或新表中，安全地进行数据更改。版本控制视图将表以不同方式投射到每个版本中，从而使每个版本只能看到自己的列。
- 交叉版本触发器将旧版本进行的数据更改传播到新版本的列中，或（以热翻滚方式）将新版本进行的数据更改传播到旧版本的列中。

与 Oracle GoldenGate 零停机应用程序升级相似，使用 EBR 也要求在开发人员层面上深入了解应用程序和重要工作，以便将其包含在内。与 Oracle GoldenGate 不同的是，利用 EBR 仅需一次投资。此后，为后续的新版本应用程序使用 EBR 所需的投资最少。经证实，即使是针对最复杂的应用程序也可以实施 EBR，例如 Oracle E-Business Suite 12.2 使用 EBR 联机打补丁。EBR 是作为零成本选项包含在 Oracle 数据库中的一个特性，为的是鼓励应用程序开发人员采用它。

Oracle 全局数据服务

针对使用 Oracle Active Data Guard 或 Oracle GoldenGate 的复制数据库，[Oracle 全局数据服务](#) (GDS) 是一个完整的自动化负载管理解决方案。GDS 可实现更好的系统利用率，并且为运行于复制数据库上的应用程序负载提供更高的性能、可伸缩性和可用性。GDS 为一组复制数据库提供以下功能：

- 基于区域的负载路由
- 连接时负载平衡
- 为 Oracle 集成客户端提供运行时负载平衡顾问
- 数据库间服务故障切换
- 针对 Oracle Active Data Guard 的基于复制延迟的负载路由

- 针对 Oracle Active Data Guard 的基于角色的全局服务
- 集中式负载管理框架

白金级总结：数据保护、RTO 和 RPO

白金级层与黄金级层提供相同的损坏保护。白金级和黄金级层之间的差异在于白金级应用程序的恢复时间 (RTO) 和数据丢失可能 (RPO)。表 7 总结了白金级层的 RTO/RPO。

表 7：白金级应用程序的恢复时间 (RTO) 和数据丢失可能 (RPO)

类型	事件	停机时间	数据丢失可能
计划外	数据库实例故障	零应用停机 相对于数秒	零
计划外	可恢复的服务器故障	零应用停机 相对于数秒	零
计划外	数据损坏、不可恢复的服务器故障、数据库故障或站点故障	零应用停机 相对于零至数分钟	零 相对于近乎为零
计划内	联机文件移动、联机重新组织和重新定义、联机打补丁	零	零
计划内	无法联机执行但是符合 RAC 滚动安装资格的硬件或操作系统维护和数据库打补丁	零应用停机	零
计划内	数据库补丁集或完整版本升级	零应用停机 相对于数秒	零
计划内	平台迁移	零应用停机 相对于数秒	零
计划内	修改后端数据库对象的应用程序升级	零应用停机 相对于数小时至数天	零

数据库整合规划

对任何数据库而言，为整合数据库规划高可用性的第一步都始于同一个地方。进行业务影响分析，以便对作为整合对象的每个数据库进行评估，评估其对数据丢失（恢复点目标，或称为 RPO）和停机时间（恢复时间目标，或称为 RTO）的容忍程度。此分析还可以确定可能存在于两个或多个数据库之间的所有相关性，即一个数据库的不可用或数据丢失会影响其他数据库高效服务于它们所支持的应用程序的能力。

将整合对象分组到不同的高可用性层

第二步，基于其 RTO 和 RPO 确定适合进行相互整合的数据库集。MAA 最佳实践是，根据上节介绍的一组标准高可用性层整合具有相似 RTO 和 RPO 的数据库。请注意，在数据库之间存在相关性的情况下，应将每个数据库分配给具有最严格高可用性要求的数据库适用的高可用性层。

根据 RTO 和 RPO 要求将数据库分组到标准高可用性层的过程实现以下三个目标：

- 标准化一组有限的高可用性层将降低复杂性并实现规模经济。
- 通过在高可用性基础架构和过程中避免过度投资或不必要的复杂性而实现高效整合。例如，如果要整合数据库的 RTO 和 RPO 可通过使用另一个数据库（它的 RTO 和 RPO 需要额外的高可用性基础架构来实现实时高可用性和灾难恢复）恢复备份来实现，则将其视为效率低下。
- 为 DBaaS 建立[服务目录](#)的高可用性和数据保护组件。服务目录描述 IT 组织为其用户群（开发人员、架构师和最终用户）提供有关如何提供和管理数据库服务的细节的各种服务。

选择整合方法

每个高可用性层的参考架构都支持所有整合方法和部署模型。但是，针对将独立数据库迁移到整合环境的详细最佳实践受使用的整合方法影响。

- 从数据库的角度看，使用虚拟机部署模型的服务器整合类似于将多个独立 Oracle 数据库从分散的计算机迁移到单个物理计算机。就这点而言，包含在本白皮书中的高可用性最佳实践同样适用于物理或虚拟环境。但是，虚拟机还需要考虑容量规划、性能、系统管理和高可用性，从而带来了额外的复杂性。这些考虑事项特定于所使用虚拟技术的供应商。为虚拟机提供高可用性或者将虚拟机用作整合方法的详细最佳实践超出了本白皮书的讨论范围。
- 对现有应用程序实施模式整合具有一定的挑战性。这些挑战包括名称冲突、安全问题、难以进行按模式时间点恢复、打补丁和克隆。将独立数据库迁移到模式整合环境的最佳实践超出了本白皮书的讨论范围。
- 将 Oracle RAC 与 Oracle Resource Manager 结合使用的简单数据库整合是 Oracle Database 11g 的 MAA 最佳实践。除了明确说明 Oracle Database 12c 特性的情况外，本白皮书记录的高可用性层同样适用于 Oracle Database 11g 和 Oracle Database 12c。
- Oracle Database 11g 的 Exadata 整合最佳实践超出了本白皮书的讨论范围，但是在“[Oracle 数据库云服务器上的数据库整合最佳实践](#)”中对这些最佳实践进行了详细介绍。

- 从 Oracle Database 12c 开始，Oracle Multitenant 成为最佳数据库整合方法。多租户架构结合了之前所有整合方法的最佳品质，去除了其伴随的弊端。

本白皮书假设从 Oracle Database 12c 开始，Oracle Multitenant 成为使用的数据库整合方法。MAA 最佳实践同样适用于物理或虚拟环境，但是它们不考虑虚拟机所独有的其他注意事项。

将高可用性层与硬件池保持一致

在较高的层面上看，使用 Oracle Multitenant 来虚拟化 Oracle 数据库并实现整合的过程非常简单，只需将每个数据库转换为 PDB 并将其整合到给定高可用性层的一个或多个 CDB 中即可。最终，您必须执行供应足够容量所需的分析，以便为要整合到每一层的数据库提供所需的服务级别。在容量规划实践中，一部分工作要求：根据每层上部署的 CDB 数量，计划您的最终整合密度。有多个原因支持在一个高可用性层中部署多个 CDB，即使在给定层的所有 PDB 都有相似的服务级别要求时也是如此。

- 底层硬件可能有容量限制，它妨碍了将分配给特定高可用性层的所有 PDB 整合到单个服务器或服务器集群中。必须评估性能要求，以确定是否适合将数据库作为数据库整合的对象，然后将要整合的现有托管服务器与将整合到的服务器进行匹配。
- 通常需要一次支持多个 Oracle 数据库版本/补丁集。例如，一个 CDB 可能采用 Oracle Database 12c 第 1 版 (12.1.0.1)，而另一个 CDB 在 Oracle Database 12.1.0.2 发布后可能采用 Oracle Database 12.1.0.2。这样便支持从 Oracle 12.1.0.1 CDB 中“拔出”单个 PDB 并将其“插入”到 Oracle 12.1.0.2 CDB，以便在其他 12.1 PDB 无法升级时实施升级。例如，由于供应商提供的支持新版本的应用程序出现延迟，便可能发生这种情况。
- 保守起见，您可以偏向于让部署的 CDB 具有更有限数量的 PDB，以便获取多租户架构的经验。鉴于可以轻松地增加整合密度（只需从一个 CDB 中拔出 PDB 并将其插入另一个 CDB 即可，从而最终减少每个高可用性层中的 CDB 数量），因此此策略对未来没有影响。

还有一个典型问题，首先应将哪些数据库迁移到 Oracle Multitenant。青铜级层具有符合整合条件的最大数量的数据库。Oracle 建议从青铜级层开始，假设它将快速产生投资回报且几乎没有风险。

表 8 概述了整合规划过程。

表 8：整合规划

关键整合步骤	建议	优势
1. 标准化高可用性 SLA 并将应用程序及其数据库分配给相应的高可用性层	遵循 MAA 高可用性层（白金级、黄金级、白银级、青铜级）	通过标准化减少运营支出 (OPEX) 并降低风险。
2. 设计高可用性架构以满足 SLA，确定配置和运营最佳实践	每个高可用性层都遵循建议的高可用性参考架构为 Oracle 数据库使用 MAA 配置和运营最佳实践 (www.oracle.com/goto/maa)	降低运营支出 提高整体稳定性和可用性。 MAA 验证的配置最佳实践可降低风险。
3. 减少 Oracle 数据库平台的硬件平台和操作系统的个数	对 Linux 或 Solaris x86-64，数据库云服务器提供一个标准的可伸缩数据库平台以实现最大整合密度。 Oracle 数据库机为较小的环境提供具有高整合密度的标准平台 对 SPARC 和 Solaris 或多用途数据库和应用程序处理，使用 Oracle SuperCluster（包括 Exadata 存储和 Exadata 特性） Oracle 虚拟计算设备已针对为任何 Linux、Oracle Solaris 或 Microsoft Windows 应用程序部署虚拟基础架构进行了优化	已经为工程化系统平台优化并预先配置了 MAA 最佳实践 减少设置时间并降低运营支出 系统和软件的集成支持降低了运营支出 由于硬件平台更少，因此降低了资本支出。 Exadata 软件与 Oracle 资源管理和压缩技术相结合已经针对数据库整合进行优化 高整合密度减少了总生命周期成本。
4. 建立作为数据库整合目标的硬件池	创建适当大小的硬件池。建议半机架至最高两个全机架间系统作为典型硬件池。 将每个硬件池分配给高可用性层。由于每一层都有不同的高可用性架构，因此每个硬件池只有一个高可用性层。	降低了伴随系统泛滥的复杂性和变化性 降低运营支出
5. 选择标准映像软件体系	在内部验证后选择 2 或 3 个最新 Oracle 补丁集。示例包括 11.2.0.3、11.2.0.4 和 12.1.0.1 与建议的 PSU 或补丁包。 为每个数据库补丁集版本选择不超过 1 个的其他变体。 建议一个 Oracle Grid Infrastructure 和最多五个 Oracle 主目录。	降低软件泛滥和风险。降低运营支出并提高整体稳定性和可用性。

6. 选择数据库整合方法	<p>在 Oracle Database 12c 中使用多租户架构作为数据库整合的最有效形式。</p> <p>如果需要专用资源来实现严格隔离，并且产生的运营支出和资本支出成本可以接受，则使用操作系统虚拟化，例如 OVM 和 Oracle Solaris Zones。</p> <p>对于 Exadata 数据库整合，请参阅附录 2, 使用 Oracle Multitenant 的 Exadata 整合最佳实践。</p>	实现最大的运营支出和资本支出节省
7. 系统选型、资源需求和性能预期	<p>评估每个应用程序当前和未来的 CPU、I/O、内存和存储容量消耗。</p> <p>使用 EMCC 整合规划程序（请参阅文档和演示）或联系 Oracle 咨询部门或 Oracle 高级客户支持服务 (ACS) 获取选型服务。</p>	<p>在迁移到目标整合解决方案后降低资本支出和运营支出</p> <p>使用数据将数据库分配给硬件池并配置资源管理器分配和限制。</p>
8. 使用相同的高可用性层将数据库分配给相应的硬件池	<p>如果整合到已经托管主动数据库的硬件池，监视当前 CPU、I/O、内存和存储消耗以了解其可用容量。</p> <p>当硬件池没有空闲容量时，将下一个数据库分配给新硬件池。由于重要程度和资源消耗，会出现硬件池中有单个数据库的情况。</p> <p>选择 EM12c 进行供应</p>	
9. 配置资源分配和限制	<p>使用资源管理最佳实践保证并限制每个数据库和 PDB 的资源。</p> <p>对于需要专用资源进行隔离的数据库，请考虑使用 PROCESSOR_GROUP_NAME 参数将实例绑定到专用的 CPU 或 NUMA 节点。</p> <p>有关更多详细信息，请参阅本白皮书的“资源管理”部分。</p>	高效资源利用率
10. 部署监视和自助服务基础架构	Oracle Enterprise Manager Cloud Control 12c	降低运营支出，改善服务

迁移到多租户架构

在完成整合规划并且将数据库整合对象分配给相应的硬件池和高可用性层后，下一步是将现有数据库迁移到多租户架构。使用的方法取决于：

- 源数据库类型（非 CDB、CDB 或 PDB）
- 源数据库版本
- 源和目标硬件平台和数据库高可用性要求

本节提供将数据库迁移到多租户架构的战略。有关常规迁移选项，请参阅[数据库平台或位置迁移](#)，有关到非多租户数据库的 Exadata 迁移最佳实践，请参阅[迁移到 Exadata 数据库云服务器的最佳实践](#)。

对于版本 Oracle Database 12c，三种数据库类型如下所列：

- 非容器数据库（非 CDB）。这是不是多租户容器数据库 (CDB) 或可插拔数据库 (PDB) 的任何数据库。它包括 Oracle 12c 版本之前的所有数据库、已经升级到 Oracle 12c 但尚未迁移到 PDB 的 Oracle 12c 之前的任何数据库，以及 Oracle 12c 创建但未创建为 CDB 或 PDB 的任何数据库。
- 多租户容器数据库 (CDB)。CDB 是在 Oracle Database 12c 中将 ENABLE_PLUGGABLE_DATABASE 初始化参数设置为 TRUE 创建的数据库。它创建后将成为 0、1 或许多个 PDB 的容器。无法将非 CDB 转换为 CDB，反之亦然，因为无法使用 CDB 创建模式对象或处理用户 SQL。Oracle 建议将所有新 Oracle Database 12c 数据库创建为 CDB。
- 可插拔数据库 (PDB)。这是一系列可移植模式、模式对象和非模式对象，它们对于应用程序来说显示为独立数据库。非 CDB 可转换为 PDB，但是 PDB 无法转换为非 CDB。

推荐的迁移战略

多租户架构的迁移选项高度依赖于源数据库版本以及所需的高可用性层（本白皮书之前讨论过）。表 9 总结了各迁移选项。也请参阅 My Oracle Support 说明 1576755.1：[使用 ASM 进行文件存储迁移非 CDB 和 PDB 的逐步示例](#)。

表 9. 多租户架构迁移方法

SLA 层和源	迁移路径	注意事项	DATA GUARD 备用数据库影响	停机时间估计
白银级、黄金级或白金级 任何数据库类型 任何数据库版本 任何平台	Oracle GoldenGate	要求在 GoldenGate 实例化之前从种子预先创建 PDB 如果使用数据泵或任何逻辑实例化，则可以重新组织并重新优化对象（例如，进行压缩）。 快速回退选项可用	在主数据库上实例化 Oracle GoldenGate 对象可能要求执行其他步骤来维护备用 CDB	近乎为零（基于手动转换操作）
白银级 Oracle Database 12c 非 CDB 和同字节序平台	RMAN 活动数据库复制	执行描述/非 CDB 插入 支持增量文件迁移 不需要任何临时区域 这是物理迁移，因此不会重新组织或重新优化对象（例如，进行压缩）。	将文件复制到备用 CDB MRP 找到文件时继续，并且 MRP 将继续	45 分钟，受上次增量时间和 noncdb_to_pdb.sql 执行所需时间影响
青铜级或白银级 任何 Oracle Database 11.2.0.3 或更高版本 任何平台	完整数据库 TTS/数据泵	要求在数据泵导入之前从种子预先创建 PDB 支持增量文件迁移 需要用于增量备份的临时区域 这是物理迁移，因此不会重新组织或重新优化对象（例如，进行压缩）。	使用 ASMCMD cp 或基于主机的复制命令将文件复制到备用数据库 遵循 Data Guard 文档处理数据文件	2 小时，受数据泵元数据导入所需时间影响
青铜级 任何 Oracle 数据库版本 任何平台	可传输表空间 (TTS) 和数据泵导出/导入 自定义脚本，以复制或提取并加载数据。	要求在数据泵导入之前预先创建种子 PDB 如果使用数据泵或任何逻辑实例化，则可以重新组织并重新优化对象（例如，进行压缩）。	N/A	取决于导出和导入的大小和时间 数小时至数天。

虽然可以使用可传输表空间和数据泵从 Oracle Database 12c 之前的 Oracle 数据库版本直接迁移到 PDB，但是首选方法是首先将数据库升级到 Oracle Database 12c。从 Oracle Database 12c 源转换为多租户架构的迁移优势包括：

- 简单性。需要的步骤更少并且可用更多的工具，从而简化了过程
- 在迁移到多租户架构之前验证 Oracle Database 12c 数据库的性能和功能

如果您选择首先升级数据库，请注意 Oracle 数据库版本 10.2.0.5、11.1.0.7 和 Oracle 11g 第 2 版 11.2.0.2 或更高版本可以直接升级到 Oracle Database 12c。另外所有其他版本需要进行两次升级，首先将数据库升级到前面列出的最低中间版本之一。请参阅 My Oracle Support 说明 1462240.1：[Oracle Database 12c Upgrade Companion](#) 了解有关从之前版本升级到 Oracle Database 12c 的详细信息。

文件移动

考虑到源和目标上给定的数据库大小、网络带宽和可用系统资源，将数据库从一个数据中心移到另一个数据中心或者从一个系统移到另一个系统可能具有一定的挑战性。只要可能，应使用以下选项之一降低文件移动操作和停机的影响：

- 在插入之前将文件复制到目标位置。从 Oracle Database 12c 开始，RMAN 复制命令可感知 CDB 和 PDB。[附录 B](#) 提供了使用 RMAN 复制进行迁移的一个示例。在 My Oracle Support 说明 1576755.1 中提供了其他迁移用例的其他详细信息和示例：[使用 ASM 进行文件存储迁移非 CDB 和 PDB 的逐步示例](#)。

此外，可以将 RMAN 备份增量应用于初始复制，从而最大程度减少整体停机。某些选项包括使用 RMAN 备份增量并适用于相同的源和目标平台，或者使用在[使用不一致的备份执行表空间的跨平台传输](#)中描述的步骤（对于跨平台迁移）。对于 Oracle Database 12c 之前的版本，可以使用 My Oracle Support 说明 1389592.1 中的步骤：[使用跨平台增量备份缩短可传输表空间的停机时间](#)。需要一小段额外的时间进行拔出/插入操作，以应用最终增量；但是，源文件保持原封不动，从而允许在出现问题时立即进行回退。

- 配置存储，以便源和目标环境都可以访问数据文件。如果源数据文件存在于目标 CDB 可访问的共享存储上，则不需要移动文件，并且可简化插入命令。原位使用源文件提供最快的拔出/插入操作；但是，这样将修改源文件。如果需要回退到之前的环境，则需要数据库还原和恢复，或者至少一个 RMAN 开关以处理数据丢失。

使用这些选项之一，可以在插入语句中指定 NOCOPY 子句。

迁移和 Data Guard

在迁移到包含 Data Guard 的预配置黄金级或白金级架构时，作为插入过程的一部分，还必须确保在备用位置可以使用您的数据文件的副本。在执行插入操作之前，这些文件必须处于备用位置上的适当位置，以便允许介质恢复自动查找文件并继续不间断地应用重做。Oracle 数据库将使用 DB_FILE_NAME_CONVERT 设置或在 DB_FILE_CREATE_DEST 目录下自动搜索文件。

如果您计划向同一个 CDB 进行许多迁移，则最好延迟向备用站点的重做传输，并在所有迁移完成后重新实例化备用站点，以简化处理过程。这样，您便可以一步将所有文件复制到备用站点，而不是在每次插入发生时为其执行复制。此选项的缺点是，在 PDB 备用数据库实例化完成之前，主数据库不会得到全面保护。

Oracle 资源管理

所有数据库、CDB 和 PDB 将争用硬件池中的有限 CPU、I/O 和内存资源。这一部分介绍如何规划、配置和管理整合的多租户架构环境中的资源。现有的 Oracle 数据库管理工具均可感知多租户架构，应该使用这些工具来保证数据库或 PDB 具有充足的资源量并且不会对其他应用程序带来不利影响。这在 DBaaS 环境中尤为重要，因为该环境中的客户需要付费使用特定服务或访问固定的系统资源量。

可以在非 Exadata 和 Exadata 系统中使用资源管理工具，但是 Exadata I/O 资源管理器 and 网络资源管理器除外。有关最新推荐的补丁、监视脚本和分步实施指南，请参阅 [Master Resource Manager MOS 说明 1339769.1](#)。

规划

规划整合性能的第一步是确定数据库的性能要求。图 10 描述了性能隔离层：白金级、黄金级、白银级和青铜级。由于减少了使用资源管理的性能隔离的级别，因此将增加整合密度。



图 10：整合和性能隔离层

通常，具有黄金级层性能要求的数据库或 PDB 也同时具有黄金级层高可用性服务要求，但是不需要严格执行高可用性服务要求。处于较高性能隔离层的数据库要求较高份额的资源保障。像对 CPU 和 Exadata IORM 进行 CDB 资源管理这样的资源管理工具允许共享资源，同时保证每个数据库具有指定的资源量。对这些工具，在可用资源不断变化的系统上，处于较低性能隔离层且具有较少资源分配的数据库和 PDB 可能会发生性能波动。为了提供更为一致的性能，请使用资源管理器限制来限制资源。资源限制也可能成为鼓励 DBaaS 环境中的客户付费升级到较高性能层的有效方式。

最简单的规划往往是最佳规划。

- 在同一硬件池中运行的所有数据库和 CDB 应该处于相同的高可用性服务和性能隔离层，即青铜级、白银级、黄金级或白金级。此外，一个 CDB 中的所有 PDB 也应该具有相同的高可用性服务级别和性能隔离层。
- 使用实例囚笼、内存参数和 Exadata IORM 来管理共享一个硬件池的所有数据库和 CDB。
- 需要不同数据库版本时创建单独的 CDB，但是将 Oracle 数据库主目录的数量限制到最多 5 个以降低运营成本。
- 为不同应用程序类型（例如 OLTP 和 DW）创建单独的 CDB。这样分组便于管理 CDB，因为 OLTP 和 DW 的资源需求往往迥然不同。例如，OLTP 数据库受益于较大的缓冲区缓存和闪存缓存，而数据仓库则受益于较高的磁盘 I/O 吞吐量。
- 在使用 Oracle 数据库云服务器时，将应用程序划分为受益于闪存缓存的 CDB 与没有闪存缓存执行起来也可接受的 CDB。这样便可以为需要利用 Exadata 智能闪存缓存或智能闪存日志的 OLTP 数据库 (PDB) 提供更好的闪存命中率。例如，可以将能够忍受较长响应时间的许多测试和开发数据库插入到通过 Exadata IORM 规划禁用了闪存缓存的 CDB 中。可以

将 I/O 敏感应用程序的关键数据库插入到通过 Exadata IORM 规划启用了闪存的单独 CDB 中。另一个示例是可能不需要闪存缓存但最能受益于 Exadata 的智能分流功能和存储索引的纯 DW 或分析数据库。

对其中可能存在其性能需求有较大变化的数据库和应用程序的青铜级层，此额外配置建议最为相关。对于所有应用程序都很重要且对性能敏感的更高层，Exadata 智能闪存缓存针对所有应用程序的 I/O 请求（日志写入、OLTP 或较小 I/O 读取和写入以及表扫描）动态做出调整。

- 从开始便使用 CDB 资源管理来管理共享某个 CDB 的 PDB。这样便可以确保客户访问相同的一组资源并体验一致的性能，不管在硬件池中运行的数据库是 10 还是 100 个。通过从开始便使用资源管理，您的 CDB 资源规划将允许您向每个 PDB 分配有保障的 CPU 和磁盘 I/O 资源。另外，CDB 资源规划允许您对一个 PDB 可以使用的 CPU 或磁盘 I/O 量设置硬性限制。这在客户付费获得性能的多租户架构环境中非常有用。
- 重点关注具有最高可能整合密度的青铜级层，并一次分步处理一小批数据库，以便获得多租户架构的经验和专业知识。

控制资源

表 10 总结了在整合环境或多租户架构环境中控制共享资源的建议。这些建议适用于所有高可用性服务级别和性能层。随后的表中给出了针对每个层的特定建议。

表 10：控制共享资源的指导原则

资源	指导原则
内存	<p>独立于高可用性层（青铜级、白银级、黄金级、白金级），内存决不应过度使用。应该在将数据库迁移到硬件池之前执行容量规划和趋势分析。</p> <ul style="list-style-type: none"> • 根据您的目标或当前共享内存要求设置 HugePages。 • 不要禁用 PGA_AGGREGATE_LIMIT 初始化参数。PGA_AGGREGATE_LIMIT 对 PGA 内存使用施加硬性限制，它对于避免过量分页（可能会导致性能低下或实例被逐出）来说非常重要。如果不同时降低 PGA_AGGREGATE_TARGET，则不要降低它的值。可以动态调整该参数。 • 使用以下公式计算数据库的初始占用空间。通过遵守表 10 中的性能层最佳实践，确保不过度使用内存。一旦实际峰值内存使用的 PGA 和按进程内存统计数据可用，则可以将这些值替代为这些初始近似值。 <ul style="list-style-type: none"> o OLTP 数据库：SGA_TARGET + PGA_AGGREGATE_TARGET + 4 MB *（最大进程数） o DW/BI 数据库：SGA_TARGET + 3 * PGA_AGGREGATE_TARGET • 对于 64 位系统，将 LOG_BUFFER 设置为最大 256 MB • 参考： <ul style="list-style-type: none"> o 有关 HugePages 的更多详细信息，请参阅 MOS 361468.1 和 MOS 401749.1

CPU 实例囚笼数据库和 CDB 运行于同一服务器上，以避免实例间的 CPU 争用。实例囚笼可防止失控查询和其他负载激增而消耗所有可用的系统资源。

- 根据表 10 中的性能层最佳实践配置实例囚笼。
- 要配置实例囚笼，请将“spfile”中的 CPU_COUNT 设置为实例可以使用的最大 CPU 线程数（最小值为 2）。然后，通过设置有效 RESOURCE_MANAGER_PLAN 来启用实例囚笼，例如为非 CDB 设置 DEFAULT_PLAN，为 CDB 设置 DEFAULT_CDB_PLAN。如果资源需求发生变化，可以动态更改 CPU_COUNT 参数，这将立即调整“实例囚笼”大小。
- 参考
 - o MOS 1362445.1 –“配置和监视实例囚笼”
 - o 有关配置操作系统信号的通用最佳实践，请参阅“数据库云服务器上的数据库整合最佳实践”– 仅限 Exadata。

使用 CDB 资源管理器管理共享 CDB 的 PDB 之间的 CPU 使用和争用。在 CDB 资源规划中，您可以为每个 PDB 配置 SHARES 和 UTILIZATION_LIMIT 指令，如下所述。

- 使用 SHARES 配置“公平性”。份额表示 PDB 的相对重要性。具有更多份额的 PDB 将被允许使用更多 CPU。如果所有 PDB 同等重要，则将其份额设置为相同的值或使用 DEFAULT_CDB_PLAN。如果一个 PDB 的重要性是另一个 PDB 的两倍，则将其份额加倍。您可以使用“默认指令”为 PDB 指定一个默认的份额。

将现有数据库迁移到 CDB 时，将各个数据库的 CPU_COUNT 转换为 SHARES，以便创建等效的 CDB 资源规划。

- 使用 UTILIZATION_LIMITS 配置“按性能付费”。在公有云中，根据订阅率限制一个 PDB 可以使用的 CPU 数量。限制将防止“青铜级”PDB 使用空闲系统上的所有 CPU，正如实例囚笼防止数据库使用空闲系统上的所有 CPU。从插入 PDB 的时候就设置 DBRM“利用率限制”有助于确保从一开始便正确设置预期。
- 如果为 CDB 启用实例囚笼，则除了“实例囚笼”大小还应应用利用率限制（例如 cpu_count），以确定一个 PDB 可以使用的最大 CPU 数。

- 在 RAC 数据库中，必须为所有实例设置相同的 CDB 资源规划。如果不相同，“并行语句队列”和“Exadata IORM”将以不可预测的方式运行。

- 参考
 - o MOS 1567141.1 –“将使用实例囚笼的数据库迁移到 CDB”

处理器和会话 在大多数情况下，PDB 使用在 CDB 级别设置的初始化文件参数设置，但是许多参数可以在 PDB 级别进行设置。请注意，这些设置不存储在参数文件中；它们存储在 CDB 中的内部表中。如需可以修改的所有 PDB 特定设置的完整列表，请执行以下语句：

```
SQL> SELECT NAME FROM V$SYSTEM_PARAMETER WHERE

ISPDB_MODIFIABLE='TRUE'

ORDER BY NAME;
```

以下列表描述将 PDB 添加到 CDB 时需用更改的 PDB 和 CDB 的设置。

- SESSIONS:

要控制 CDB 及其 PDB 可用的会话数，请在根级的初始化参数文件中设置 CDB 的 SESSIONS 总数。对于每个 PDB，您可以选择为 SESSIONS 参数设置一个值，以便对 PDB 可以启动的会话数施加硬性限制。初始化参数 SESSIONS 设置 CDB 的最大值，PDB 中的 SESSIONS 设置则特定于该 PDB。

- PARALLEL_MAX_SERVERS:

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, 这会导致数据库节点性能降低且变得不稳定。

在 Exadata 上:

- o 对于 X2-2、X3-2 和 X4-2: 所有实例和 PDB 的 sum(PARALLEL_MAX_SERVERS) <= 240。
- o 对于 X2-8 和 X3-8: 所有实例和 PDB 的 sum(PARALLEL_MAX_SERVERS) <= 1280

在 CDB 中, 所有 PDB 共享并行服务器。要限制 PDB 任何时候可以使用的最大并行服务器数量, 请将 CDB 资源规划中的 PARALLEL_SERVER_LIMIT 指令设置为 PDB 任何时候可以使用的 PARALLEL_MAX_SERVERS 的最大百分比。

- 限制进程数量以及到数据库服务器的连接数:

拥有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, 但最重要的是获得系统可靠性。

可保守地使用最多 2 倍 CPU 内核数的**活动进程数**以及**整个数据库节点最多 10-12 倍 CPU 内核数的总进程数**。通过使用以下一项或多项技术, 可以降低进程总数并提高性能和稳定性:

- o 使用连接池并将最大连接数设置为稍高于估计活动工作会话的值, 从而限制进程数量以及到数据库服务器的连接数。通过将 min=max 连接设置为相同而消除动态连接池, 从而避免成本高昂的进程分配和取消分配。

注意: 如果在 Exadata 上运行, 请参阅 [Exadata 整合最佳实践](#) 了解有关使用共享服务器/MTS、连接池等的建议。

- o 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。

- 限制重做应用并行度 (如果这是 Active Data Guard)

- 参考

- o 有关减少进程数的详细信息, 请参阅[“数据库云服务器上的数据库整合最佳实践”](#)— 仅限 Exadata

IORM
(仅限
Exadata)

- 使 Exadata IORM 能够确保跨所有 CDB 和 PDB 公平访问磁盘。默认情况下, IORM 在运行时采用“基本”IORM 目标, 并且仅执行名义数量的调度以限制关键磁盘 I/O 的延迟不会达到极限级别。为了完全提供支持, 请将“IORM 目标”设置为“自动”而不是默认的“基本”。设置目标后, IORM 执行数据库间 IORM 规划及所有 CDB 和数据库资源规划。
- 当 OLTP 数据库和数据仓库共享相同的 Exadata 存储单元时, 将从闪存提供大多数 OLTP I/O, 并且将从磁盘提供大多数数据仓库 I/O。如果 OLTP 数据库发出到磁盘的 I/O, 则来自数据仓库负载的争用可能会导致不可接受的磁盘延迟增加。通过启用 IORM, 您可以在数据库间 IORM 规划和 CDB 资源规划中向 OLTP 数据库和 PDB 提供较大的资源分配, 从而降低 OLTP I/O 的磁盘延迟。如果磁盘延迟继续过高或不一致, 则可以将 IORM 目标修改为“平衡”或“低延迟”。
- 当整合的许多应用程序执行智能扫描时, 请在数据库间、CDB 或数据库资源规划中配置 UTILIZATION_LIMIT。
- 将不同负载隔离到不同 CDB 有助于确保数据仓库应用程序不会消耗重要 OLTP 数据库中的闪存资源, 从而允许 OLTP 应用程序维护其所需的 SLA。

- 参考：请参阅 MOS 1363188.1 –“为常见情形配置 Exadata I/O 资源管理器”

网络	<ul style="list-style-type: none">网络资源管理（仅限 Exadata）自动且透明地排列关键数据库网络消息的优先级，从而确保不允许延迟的操作的快速响应。在数据库、数据库 InfiniBand 适配器、Oracle Exadata Storage Server 软件、Exadata 存储 InfiniBand 适配器和 InfiniBand 交换机中实施优先级排列，以确保在整个 InfiniBand 结构中进行优先级排列。对延迟敏感的消息（例如 Oracle RAC 缓存融合消息）的优先级高于批处理、报表和备份消息。日志文件写入操作的优先级最高，以确保事务处理的低延迟。从数据库版本 11.2.0.4/12c 上 IB 交换机软件版本为 2.1.3-4 的 Exadata 11.2.3.3.0 开始，网络资源管理始终开启且可用。通过在 RAC 数据库中最小数量的实例上激活 PDB，避免出现大量缓存融合网络流量。使用 PDB 服务确定特定 PDB 处于活动状态的实例，以避免不必要的数据分发。另外，确保跨 CDB RAC 数据库的所有实例平衡负载。
存储 网络	<ul style="list-style-type: none">为每个硬件池配置一个共享存储网络。管理一个共享存储网络更为简单，且需要的管理成本更少。通过共享存储，空间和带宽的利用也更高效。如果这是一个黄金级或白金级硬件池，则为 DATA 和 RECO 磁盘组使用 ASM 高冗余，从而在 Exadata 单元滚动升级期间和遇到存储类型故障时实现最佳数据保护和冗余 – 仅限 Exadata。由于硬件池指定到一个层，因此共享存储网络仅应服务一个层。
集群	<ul style="list-style-type: none">每个硬件池使用一个集群。所有数据库服务均由一个 Oracle Clusterware 安装进行管理，该安装应该用于进一步负载平衡并将应用程序路由至集群中的特定数据库实例或 PDB。使用 Oracle Multitenant，建议每个集群保持最多 10 个 CDB 的数量，这表示每个数据库节点最多 10 个实例。每个 CDB 实例最多可以具有 252 个 PDB。 <p>您为每个数据库节点或集群创建的实际数据库实例或 PDB 数量取决于每个实例/PDB 的应用程序负载和系统资源占用。</p>

表 11 提供了根据性能层配置数据库的指导原则。

表 11：按性能层划分的资源管理指导原则			
性能层	CPU	内存	EXADATA I/O
白金级	<p>实例囚笼，没有过量使用。所有数据库的 CPU_COUNT 总和不应超过服务器 CPU 线程的 75%。</p> <p>通过配置 Linux cgroups 或 Solaris 资源池并设置 PROCESSOR_GROUP_NAME 参数，考虑将每个实例绑定到专用 CPU 或 NUMA 节点。请参阅 MOS 说明 1585184.1。</p>	<p>为所有数据库配置 SGA_TARGET 和 PGA_AGGREGATE_TARGET</p> <p>PGA、SGA 和客户端进程的内存总和不应超过系统总内存的 75%。</p> <p>注意与其他 11g 数据库整合。如果不使用 PGA_AGGREGATE_LIMIT 参数（仅在 12c 中可用），便不</p>	<p>配置数据库间 IORM。如果存储单元托管多个性能层中的数据库，则为较高层的数据库提供较多分配或份额。</p>

可能严格限制 数据库的 PGA 使用。			
黄金级	实例囚笼，没有过量使用。所有数据库的 CPU_COUNT 总和不应超过服务器 CPU 线程的 90%。	为所有数据库配置 SGA_TARGET 和 PGA_AGGREGATE_TARGET PGA、SGA 和客户端进程的内存总和不应超过系统总内存的 75%。	配置数据库间 IORM。如果存储单元托管多个性能层中的数据库，则为较高层的数据库提供较多分配或份额。
白银级	实例囚笼。如果数据库在不同的时间出现峰值负载，则可以允许适度的过量使用。	为所有数据库配置 SGA_TARGET 和 PGA_AGGREGATE_TARGET PGA、SGA 和客户端进程的内存总和不应超过系统总内存的 80%。	配置数据库间 IORM。如果存储单元托管多个性能层中的数据库，则为较高层的数据库提供较多分配或份额。
青铜级	实例囚笼。过量使用。	为所有数据库配置 SGA_TARGET 和 PGA_AGGREGATE_TARGET PGA、SGA 和客户端进程的内存总和不应超过系统总内存的 90%。	配置数据库间 IORM。如果存储单元托管多个性能层中的数据库，则为较高层的数据库提供较多分配或份额。 考虑对青铜级数据库设置“限制”，以便获得更加一致的性能并限制智能扫描。 如果闪存缓存不足以大到可容纳所有数据库，则考虑通过数据库间规划禁用非关键数据库的闪存缓存，例如测试和开发数据库。有关监视闪存缓存未命中率的信息，请参阅下一部分。

监视资源

将数据库迁移到硬件池之前就开始监视和分析数据库性能。在决定数据库是否为合适整合对象之前，请务必从源头了解它的平均和峰值资源消耗。

在迁移到目标整合环境后，密切监视性能指标对于了解整合规划是否可行仍很重要。始终建议首先使用测试环境，以便可以在转入生产环境之前能够对迁移过程以及对将迁移的每个数据库的影响进行验证。

应该分 3 个级别进行监视：

- 系统监视。应监视服务器和存储，以查看 CPU、内存和存储的使用是否处于可接受的水平。如果您计划将更多数据库整合到服务器或存储中，还应该监视可用容量或空间。
- CDB 或数据库监视。应该监视数据库以查看它实际正在使用多少资源。这项检查对于无法使用资源管理器限制的资源尤为重要，例如 PGA。还应该监视数据库以查看多少资源管理器正在限制它。如果您看到由于资源管理器而等待 CPU 或 I/O，并且数据库的性能不可接受，则可以调整资源管理器或将数据库移至具有更多资源的系统。
- PDB 监视。与数据库一样，应以相同的方式监视每个 PDB。如果其实际资源使用超过当前 CDB 资源规划保证的资源，那么随着将更多 PDB 添加到 CDB，则 CDB 的性能可能会下降。如果性能的下降不可接受，则不要将更多 PDB 添加到此 CDB 实例。

表 12 重点介绍几个关键性能指标。

表 12：关键性能指标

监视/管理	指导原则
内存	<p>对于系统：</p> <ul style="list-style-type: none"> • 不应该看到分页。使用 <code>vmstat</code> 命令监视零或非常低的页入和页出率。 • 决不应过量使用内存。在 Linux 上使用 <code>/proc/meminfo</code> 获取系统总内存。将它与为不同 Oracle 数据库和客户端进程实际分配的内存进行比较。可以从每个数据库的 <code>SGA_TARGET</code> 参数计算 SGA 使用率。可以使用以下统计信息确定 PGA 使用率。 <p>对于数据库或 CDB：</p> <ul style="list-style-type: none"> • 通过 <code>v\$pgstat</code> 监视每个实例的实际 PGA 使用率。“分配的最大 PGA 数”统计信息是实例自启动以来分配的最大 PGA 数。统计信息“分配的 PGA 总数”是实例当前分配的 PGA 数量。监视这些值可以让 DBA 知道实例实际正在使用多少 PGA。应该将它们与 <code>PGA_AGGREGATE_TARGET</code> 进行比较，以查看数据库是否超出该参数设置以及超出多少。
CPU	<p>对于系统：</p> <ul style="list-style-type: none"> • 使用操作系统工具或 <code>v\$sysmetric_history</code> 中的“主机 CPU 利用率”指标监视系统的 CPU 利用率。如果 CPU 利用率接近 100%，请使用操作系统工具或 <code>v\$sysmetric_history</code> 中的“操作系统负载”指标确定系统过量使用的程度如何。如表 10“按性能层划分的资源管理指导准则”所述，应避免出现过度过量使用的情况。使用实例囚笼避免过量负载。 <p>对于数据库或 CDB：</p> <ul style="list-style-type: none"> • 如果已启用实例囚笼或 CDB 资源管理器，则监视实例实际使用的 CPU 数量以及需要但阻止使用的 CPU 数量。按照 MOS 说明 1338988.1 所述使用 <code>v\$srcmgrmetric_history</code>。所有用户组和 PDB 的“avg_running_sessions”总和指定实际使用的 CPU 数量。所有用户组和 PDB 的“avg_waiting_sessions”总和指定资源管理器由于 CPU 不足而执行的限制。它对应于额外需要的 CPU 数量。 • 对于 CDB，监视可用 CPU 容量或空间，以确定是否可以添加更多 PDB。如果已启用 CDB 资源管理器，请通过从所

有用户组和 PDB 的 `v$srcmrgmetric_history` 中求“avg_running_sessions”和“avg_waiting_sessions”的总和来计算需要的 CPU 总数。如果此和小于 CPU_COUNT，则此和与 CPU_COUNT 之间的差距是 CDB 上的 CPU 可用空间，并且您可以考虑添加更多 PDB。如果此和大于 CPU_COUNT，则表明此 CDB 已经满负荷运转，没有任何可用空间。添加更多 PDB 只会增加对 CPU 的争用。因此，仅应为青铜级 CDB 添加额外的 PDB，但前提是现有 PDB 及其应用程序可以容忍性能的下降。应该在高峰时段执行监视。有关更多详细信息，请参阅 MOS 说明 1338988.1。

对于 PDB：

- 监视每个 PDB 的实际 CPU 利用率和 CPU 等待时间。使用 `v$srcmrgmetric_history` 的“avg_running_sessions”指标查看一个 PDB 实际使用了多少 CPU。将实际 CPU 利用率和 PDB 的保证 CPU（基于按实例上打开的所有 PDB 的总份额划分的 PDB 份数）进行比较。例如，如果 PDB 有 1 个份额，总份额是 5，则为该 PDB 保证提供 CPU_COUNT 的 1/5。如果 PDB 的实际 CPU 利用率高于保证的 CPU 利用率，则在添加额外 PDB 时，该 PDB 的性能可能会下降。使用 `v$srcmrgmetric_history` 的“avg_waiting_sessions”指标查看 PDB 的 CPU 等待时间。如果为非零，可通过在 CDB 资源规划中增加其份额或提高利用率限制，从而提高 PDB 的性能。有关更多详细信息，请参阅 MOS 说明 1338988.1。

磁盘/闪存

对于 Exadata 存储单元。使用 MOS 说明 1337265.1 中的脚本监视 Exadata I/O 指标。在大多数情况下，也可以使用 Enterprise Manager 12c 查看这些指标。

- 使用 `CD_IO_ST_RQ` 指标监视 OLTP I/O（例如缓冲区缓存读取）的磁盘延迟。如果延迟较高并且导致 OLTP 数据库的性能问题，请启用 Exadata IORM。在数据库间 IORM 规划中，为高性能层中的 OLTP 数据库提供较多的分配。如果延迟仍然不可接受，则考虑将 IORM 目标从“自动”更改为“平衡”或“低延迟”。
- 使用 `CD_IO_LOAD` 指标监视磁盘负载。对于具有 OLTP 数据库的系统，高于 5 的值被视为高，因为较高的值将导致延迟增加。如果不可接受较高的延迟，则不应再添加更多数据库。对于数据仓库，较高的值则非常好，因为高负载将导致磁盘吞吐量得到改善。但是，如果负载指标一致地超过 20，则不应添加更多数据库。
- 使用 `DB_IO_UTIL_SM + DB_IO_UTIL_LG` 指标监视各数据库的实际磁盘 IO 利用率。这项比较将显示哪些数据库使用磁盘最多。
- 使用 `FC_IO_RQ_DISK_WRITE` 监视闪存写入 IOPS。如果总写入 IOPS 已达到发布的最大值，并且闪存性能的降低不可接受，则不要添加使用闪存缓存的新数据库或 CDB。可通过在数据库间 IORM 规划中使用“flashCache=off”指令禁用某个数据库或 CDB 的闪存缓存。
- 比较 `FC_IO_RQ_R` 与 `FC_IO_RQ_R_MISS`，以计算 OLTP 闪存命中率。如果当前性能应保持不变，请在添加更多使用闪存的数据库之前通过监视闪存命中率来确保闪存有空间。例如，如果可以接受 80% 或更高闪存命中率的性能，则在添加更多数据库时性能可能会降低，并且闪存命中率将下降到 80% 以下。

对于数据库或 CDB：

- 使用 AWR 监视 I/O 等待事件。如果有问题的等待事件是“数据库文件顺序读取”，则监视并调整数据库的闪存缓存命中率（见上）、存储单元的磁盘延迟（见上）以及数据库对小请求的 IORM 限流时间（见下）。如果有问题的等待事件是“单元智能表扫描”，则监视并调整数据库的限流时间（见下）。
- 如果已启用 IORM，则使用 `DB_IO_WT_SM_RQ` 和 `DB_IO_WT_LG_RQ` 指标监视每个请求的平均 IORM 限流时间。如果等待时间较长且数据库的性能令人不满，则需要增加数据库规划中的数据库分配/份额或提高利用率限制。如果所有数据库的等待时间都较长，则表明磁盘已达到其最大容量。在这种情况下，不应再添加任何新数据库，除非现有数据库可以容忍性能下降。

- 对于 CDB，使用 PDB_IO_UTIL_SM + PDB_IO_UTIL_LG 指标监视各 PDB 的实际磁盘 IO 利用率。这项比较将显示 CDB 中的哪些 PDB 使用磁盘最多。

对于 PDB：

- 如果已启用 IORM，则使用 PDB_IO_WT_SM_RQ 和 PDB_IO_WT_LG_RQ 指标监视每个请求的平均 IORM 限流时间。如果等待时间较长且 PDB 的性能令人不满，则需要增加 CDB 资源规划中的 PDB 份额或提高利用率限制。如果所有 PDB 的等待时间都较长，则需要增加数据库间 IORM 规划中的 CDB 分配和/或限制。在这之前，不应再添加任何新 PDB，除非现有 PDB 可以容忍性能下降。

数据库性能指标	<ul style="list-style-type: none"> • 为数据库负载配置文件使用自动负载信息库和/或 Enterprise Manager 12c，下钻以查看主要等待事件。从此处下钻到特定等待。 • 使用 ASH 分析，按用户组或 PDB 查看 CPU 利用率细分
应用程序性能指标	<ul style="list-style-type: none"> • 为数据库负载配置文件使用自动负载信息库和/或 Enterprise Manager 12c，下钻以查看主要等待事件。从此处下钻到特定等待。 • 使用 ASH 分析，按用户组或 PDB 查看 CPU 利用率细分
通用信息	<ul style="list-style-type: none"> • MOS 说明 1070954.1 中的 Exachk（仅限 Exadata）

高可用性与数据保护

非整合环境的服务级别预期与整合环境相比大不相同。以开发人员或部门使用的独立数据库为例。服务器故障导致的中断级别仅局限在通常在解决故障时能找到其他方法保持高效工作的较小用户群中。而现在考虑这同一个数据库与 100 个其他数据库（每个都支持不同的部门和用户群）整合后面临的情况。影响整合数据库的故障的企业中断级别被放大了 100 倍，从而使高可用性和数据保护成为高得多的优先级。

Oracle Multitenant 使用 Oracle 最高可用性架构 (MAA) 满足整合环境的高可用性和数据保护要求。除了 MAA 的通用高可用性和数据保护目标外，还有一些多租户架构上下文特定的目标：

- 多合一管理的简单性。MAA 最佳实践必须轻松伸缩大型整合环境的管理，以便在实现高可用性和数据保护的同时实现最大成本优势（资本和运营成本）。
- 隔离性。MAA 最佳实践必须防止影响单个 PDB 的问题影响 CDB 中其他 PDB 的可用性。

从表面上看，这两个目标似乎存在矛盾。对“n”个环境的隔离经常会导致必须单独管理“n”个不同的环境。Oracle MAA 可以利用多租户架构，因此能够以最小的隔离性牺牲为代价实现高可用性，同时实现多合一管理的巨大优势。

不管一个 CDB 包含多少个 PDB，Oracle MAA 都允许作为一个数据库管理该 CDB。对于许多维护和操作任务来说，具有 100 个 PDB 的 CDB 其任务量可以减少为原来的百分之一：一个 RMAN 备份、一个 Oracle Active Data Guard 灾难恢复备用数据库以及给一个数据库升级或打补丁。Oracle MAA 还根据需要提供与其他 PDB 相隔离来管理某个 PDB 的灵活性。例如：

- 如果必须恢复单个 PDB，RMAN 可以执行此操作而不影响同一个 CDB 上打开的其他 PDB。请注意，对于刚刚插入的 PDB，应该在插入操作后立即对它进行备份，以确保在下次定期执行计划的 CDB 备份之前出现问题时，可以对它进行恢复。
- 如果需要进行快速时间点恢复，则第一版的 Oracle Multitenant 支持在 CDB 级别使用闪回数据库。未来版本的 Oracle Multitenant 计划支持在各个 PDB 上使用闪回数据库，而不影响其他 PDB 的可用性。
- 如果单个 PDB 遇到管理员认为可能会影响该 CDB 中其他 PDB 的问题，则可以轻松拔出可疑 PDB 并将其插入另一个 CDB，从而可以隔离解决该问题。拔出/插入操作还提供了应用补丁而不影响原始 CDB 中运行的其他 PDB 的灵活性。您可以在解决问题后将 PDB 留在新 CDB 中，或者可以在安装新补丁后将 PDB 拔出并重新插入回到原始 CDB。
- 如果突然出现负载高峰、错误或某个其他更改消耗模式的事件，Oracle 资源管理器将防止 PDB 消耗多于为其分配的系统资源份额的资源。
- Oracle GoldenGate 逻辑复制可用于复制 CDB 中的一个或多个 PDB。这样便提供了最小或零（使用双向复制）停机情况下将 PDB 迁移到其他 CDB 的灵活性。在这一过程开始时，先在新 CDB 中创建该 PDB 的一个克隆，以便支持计划维护的所有用途（平台迁移、Oracle 数据库升级、修改后端数据库对象的应用程序升级或者通常需要停机的其他数据库维护）。当克隆的 PDB 在新版本上运行时，使用 GoldenGate 复制将其与自克隆原始源后发生的新事务进行同步。用户在验证完成后过渡到新版本。处理此过渡时有两个选项：
 - 通过终止用户会话，允许 Oracle GoldenGate 完成复制所有提交的事务，关闭源数据库以及允许用户连接到新版本，从而实现最小停机时间。
 - 可以使用双向复制实现零停机。这样便可以随着用户自然断开连接，然后重新建立连接而逐渐迁移用户，从而创造零停机的用户体验。这样还可以在随着负载的增加而在新版本上出现意外问题时立即实施故障恢复。双向复制要求管理员使用 Oracle GoldenGate 提供的基本功能（首次更改占优），或者通过扩展 Oracle GoldenGate 实施更复杂的冲突解决模式，实施冲突检测和解决。要求了解应用程序。

管理意外停机

表 13 列出了会对多租户架构中的数据库产生影响的各种意外停机。它还列出了可以解决该停机的 Oracle 高可用性解决方案，该解决方案可在本白皮书之前描述的各个高可用性层中找到。

表 13. 多租户架构的意外停机对照表

事件	青铜级、白银级、黄金级和白金级的解决方案	恢复时间 (RTO)	数据丢失 (RPO)
实例故障	青铜级: Oracle Restart	数分钟 (如果服务器可以重新启动)	零
	白银级: Oracle RAC 或可选 Oracle RAC One Node	数秒 (使用 Oracle RAC) 数分钟 (使用 Oracle RAC One Node)	零
	黄金级: Oracle RAC	数秒	零
	白金级: Oracle RAC 与应用程序 连续性	零应用停机	零
永久 节点故障 (但存储 可用)	青铜级: 还原和恢复	数小时至一天	零
	白银级: Oracle RAC	数秒	零
	白银级: Oracle RAC One Node	数分钟	零
	黄金级: Oracle RAC	数秒	零
	白金级: Oracle RAC 与应用程序连续性	零应用停机	零
存储故障	所有: 自动存储管理	零停机	零
数据损坏	青铜级/白银级: 基本保护 某些损坏要求还原和恢复 PDB 或整个 CDB	一小时至数天	自上次备份 (如 果不可恢复)
	黄金级/白金级: 全面损坏保护及自动块修复与 Oracle Active Data Guard	零 (使用自动块修复) 数秒至数分钟 (如果损坏由于写 入丢失所致且使用 Data Guard 快速启动故障切换)。	零 (除非损坏由于 写入丢失所致)
人为错误	所有: 逻辑故障通过闪回删除、闪回表、闪回事 务、闪回查询和撤销来解决。	取决于检测时间, 但隔离到使用 这些对象的 PDB 和应用程序。	取决于逻辑故障
	所有: 影响整个数据库和 PDB 的广泛逻辑故 障要求进行 RMAN 时间点恢复 (PDB) 或闪回 数据库	取决于检测时间	取决于逻辑故障
	黄金级/白金级: 使用 Oracle GoldenGate, 只 能故障切换一个 PDB	取决于检测时间, 但实际故障切 换可能花费数秒	取决于逻辑故障
数据库不 可用、	青铜级/白银级: 还原和恢复	数小时至数天	自上次备份

系统、站点或存储故障、广泛的损坏或灾难	黄金级：故障切换到备用数据库（Oracle Active Data Guard 或 Oracle GoldenGate） 白金级：Active Data Guard 故障切换与应用程序连续性	数秒 零应用停机	零至近乎为零 零
性能下降	所有：数据库资源管理器和调优	无停机，但服务会降级	零

注意：为确保 PDB 可恢复，应在 PDB 插入操作后立即执行 PDB 或 CDB 备份。要恢复 PDB，相关 CDB 必须可用且可操作。

虽然可以在不影响其他 PDB 的情况下使用单个 PDB 修复许多问题，但是仍有些情况需要进行隔离。例如，您可能需要向 CDB 正在使用的 Oracle 主目录应用补丁。在此情形下，建议您创建新的 Oracle 主目录和 CDB，然后拔出有问题的 PDB 并将其插入新 CDB。然后，您就可以解决问题，而且对原始 CDB 中的其他 PDB 完全没有影响。解决问题后，您可以拔出 PDB 并将其插回到原始 CDB 中，或将其留在新 CDB 中，直到将来的某个时间点。

管理计划维护

从计划维护的角度看，所有可用于非 CDB 的传统解决方案在多租户架构环境中均可用。另外，在有些情况下，管理员可以决定是只对一个 PDB 还是对同一容器中的所有 PDB 进行维护。Oracle 具有很高的灵活性，可以处理任何一种情况。

表 14 着重介绍了每个计划事件的主要高可用性解决方案。

表 14：多租户架构的计划内维护对照表

事件	青铜级、白银级、黄金级和白金级的解决方案	预期停机
迁移	请参阅 使用 MAA 服务层迁移到多租户架构	各不相同
动态联机资源供应或联机重新组织和重新定义	所有：联机重新组织并重新定义每个 PDB 中的选定对象 文档： 动态联机资源供应 和 联机重新组织和重新定义	零
联机补丁	所有：如果相关，可对整个 CDB 联机打补丁	零

数据库和 Grid Infrastructure 补丁和一次性补丁	所有：PDB 可以拔出并插入到具有目标软件版本的单独 CDB	估计数秒至一小时（无数据文件复制选项）
	白银级：如果相关，整个 CDB 可以利用 Oracle RAC One Node 滚动升级	零（通过重新定位服务）
	黄金级/白金级：如果相关，整个 CDB 可以利用 Oracle RAC 滚动升级。在白金级层中，应用程序连续性将提供补充。	零（通过重新定位服务） 零应用停机 数秒至数分钟
	黄金级：整个 CDB 可以利用 Data Guard 备用数据库优先打补丁并发起 Data Guard 切换 白金级：整个 CDB 可以利用 Data Guard 备用数据库优先打补丁并发起 Data Guard 切换和应用程序连续性	零应用停机
数据库补丁集	所有： PDB 可以拔出并插入到具有目标软件版本的单独 CDB	估计数秒至一小时（无数据文件复制选项）
	黄金级/白金级：对于补丁集和主要数据库版本，整个 CDB 可以利用 Data Guard 数据库滚动升级	数秒至数分钟
	白金级：CDB 或 PDB 可以故障切换到驻留在目标软件版本上的备用 GoldenGate 副本	零停机
应用程序升级	白金级：基于版本的重新定义要求开发人员在设计时利用此特性	零
	白金级：PDB 可以故障切换到具有目标应用程序更改的 GoldenGate 副本	零至近乎为零的停机（使用移动服务）
	文档： 联机应用程序维护和升级	

打补丁和升级

打补丁和升级是受多租户架构影响最大的方面。使用多租户架构提供的“多合一管理”功能，只需升级一次，CDB 中的所有 PDB 都会升级到最新版本。也可以使用现有工具（如 Data Guard 备用数据库首先打补丁、Data Guard 临时逻辑滚动升级或 Oracle GoldenGate）将单次操作升级整个 CDB 的停机时间缩短至零或近乎为零。

如果只是想升级 CDB 中的一个 PDB 子集，则可以创建一个具有升级后版本的全新 CDB，然后拔出/插入 PDB。这样便可以提供满足特定应用程序需求所需的灵活性。而且，此操作耗用的时间会比升级单个现有数据库用时更少。通过在不同 CDB 之间复制，Oracle GoldenGate 也可以用最少的停机时间或者不停机就能升级 PDB。

DbaaS 的生命周期管理

在管理多租户架构和数据库即服务 (DBaaS) 中数据库的不同生命周期阶段方面，Oracle Enterprise Manager 12c 发挥着重要的作用。Oracle Enterprise Manager 12c 让业务和技术用户可以自助部署 IT 资源，还提供了可满足各种多租户架构需求的资源池模型。DBaaS 是一种范式，最终用户 (DBA、开发人员、QA 工程师、项目负责人等) 可以通过它来请求数据库服务，在项目生命周期内对其加以利用，并在使用后自动取消数据库服务供应后将其返回资源池。

DBaaS 提供：

- 一个用于供应数据库服务的共享整合平台
- 一个用于供应这些资源的自助式模型
- 自由伸缩数据库资源的弹性
- 基于数据库使用的计费方式

对于客户已在运行数据库的现有实施，Oracle Enterprise Manager 12c 可以自动发现这些数据库，建立使用基准，以及为迁移到多租户架构提供整合建议。然后，它还提供将非 CDB 数据库迁移到多租户架构的引导式流程。在迁移的过程中，它还将升级早期版本的数据库 (即 Oracle Database 10g 或 Oracle Database 11g)。

为了推行新数据库，Oracle Enterprise Manager 12c 利用原生插入和拔出机制，提供了用于供应 PDB 的现成工作流。相同的步骤将显示在自助服务界面上供自助服务 DBaaS 用户使用。

Oracle Enterprise Manager 提供了大规模管理 PDB 所需的自动化。配置管理功能 (如资产清单管理和报告) 有助于阻止任何不希望出现的泛滥问题。Oracle Enterprise Manager 具有行业领先的配置偏差管理功能，因此可以根据黄金基线比较整个体系并检查其一致性。在此情形下，还应当关注的另一个功能是合规性管理。Oracle Enterprise Manager 有现成的规则，可以按照众所周知的行业标准和最佳实践检查数据库的配置。

最后，在数据中心对数据库打补丁和升级是既耗时又费力的一项活动，尤其是在管理员得处理成百上千个 PDB 和底层 CDB 的情况下。Oracle Enterprise Manager 12c 提供现成的打补丁和升级自动化功能，并补充有运行前检查和修补后报告功能。

总的来说，多租户架构中数据库的生命周期管理包括：

- 从非 CDB 迁移
- 初始供应和克隆 PDB
- 资产清单跟踪、配置偏差跟踪和拓扑映射
- 配置合规性管理

- 自动打补丁和升级

有关详细信息，请参阅：

- [数据库管理](#)
- [数据库生命周期管理](#)
- [数据库云管理](#)

总结

实施数据库整合和 DBaaS 需要整体分析，以便为实现一组目标（例如减少系统占用空间）而设计的战略不会在整合环境的其他方面（例如性能、高可用性、数据保护或管理成本）造成新问题。

Oracle Database 12c 与 Oracle Multitenant、Oracle MAA 最佳实践和 Oracle Enterprise Manager 提供有效、可靠的数据库整合和 DBaaS 所需的必要的整体解决方案。虽然这些解决方案能够在任何平台上实施整合和 DBaaS，但是使用 Oracle 工程化系统可在提供最佳服务的同时全面实现生命周期总成本的最小化。

附录 A: Exadata 整合密度和性能

Oracle 在数据库云服务器上实施了一系列验证能否达到高整合密度的测试。已经使用完全相同的负载和 Oracle 数据库配置，在 Oracle 数据库云服务器上运行了两个系列的测试。第一个测试系列禁用了所有特定于 Exadata 的特性，以提供比较基准。虽然它的目的是模拟非 Exadata 系统，但鉴于即使不启用 Exadata 特性的情况下 Exadata 系统也具有较高 I/O 和网络带宽及稳定的性能特点，因此这种方法提供的比较基准较为严格。第二个系列的测试是在相同的机器上运行相同的负载，但是启用了所有特定于 Exadata 的特性。这样便可以评估 Exadata 独有的特性是否能够提高数据库负载的整合密度。

系统配置

[Oracle 数据库云服务器 X4-2](#) 全机架配置如下：

- 8 个数据库服务器，总计有 192 个 CPU 内核和 4TB 的内存
- 14 个 Exadata 存储服务器，有 168 个内核专用于存储层中的 SQL 处理
- 44 TB 的 Exadata 智能闪存缓存
- 40 Gb/秒的内部 InfiniBand 网络

数据库配置

Oracle Database 12c 的配置如下：

- Oracle RDBMS 12.1.0.1 (PSU 2) 和 Oracle Grid Infrastructure 12.1.0.1 (PSU 2)
- DBFS_DG 为常规冗余（双重镜像），DATA 和 RECO ASM 磁盘组为高冗余（三重镜像）
- 尝试使用 Oracle 单实例数据库（非 RAC）、Oracle RAC One Node 和 Oracle RAC
- 使用了 Oracle 数据库的标准 Exadata 配置最佳实践，例如使用 HugePages、合适的系统和数据库设置以及 MAA 设置
- 每个 OLTP 数据库都使用 4 GB 的 SGA
- DW/报告数据库在全部 8 个节点上运行，有 7 GB 的 SGA

分离用于比较非 Exadata 和 Exadata 整合密度的 Exadata 特性

- 智能闪存日志
- 智能闪存缓存
- 闪存缓存压缩
- 网络资源管理
- 分流扫描
- I/O 资源管理

- Exadata 存储索引

测试运行和负载

每个测试运行 30 分钟，包括数据库的重新启动和预热时间。使用 Swingbench 的订单录入应用程序生成 OLTP 负载。带有各种查询功能的数据仓库报告应用程序用于展示混合用例，但是在所有情况下都对数据仓库数据库使用实例囚笼来限制数据仓库和报告查询。

Exalogic 服务器用于驱动 OLTP 应用程序负载。每个 Swingbench 应用程序都有自己独立的连接池。

测试结果

所有测试用例都采用了相同的 MAA 整合最佳实践来获得最大吞吐量。对于非 Exadata 测试，我们不断增加数据库个数，直到达到资源限制且平均响应时间保持在 < 200 毫秒。在 I/O 受限且平均响应时间为 177 毫秒之前，我们能够整合 40 个 OLTP 数据库。主要等待事件的等待时间为 47 毫秒、Dbtime 为 55% 的单元单块物理读取（例如，等同于 db_file_sequential reads）和等待时间为 122 毫秒、Dbtime 为 31% 的日志文件同步。平均日志文件并行写入为 23 毫秒，20% 的异常情况超过 32 毫秒，最多达到 1 秒。请注意，对于只有 1 个 OLTP 数据库或最多有 5 个 OLTP 数据库的情况，等待分布和 I/O 等待时间大不相同。在只有少数 OLTP 数据库和没有系统资源瓶颈的情况下，数据库等待分布包括平均等待时间为 4 毫秒、DBtime 为 60% 的单元单块物理读取、DBtime 为 30% 的 DB CPU 以及 1 毫秒这样非常低的日志文件同步。如果有多个数据库争用 IO、CPU 和网络资源，则等待时间将迅速延长。

启用 Exadata 特性后，我们运行相同的测试，在不牺牲响应时间或稳定性的情况下，整合尽可能多的数据库。下例显示了在 Exadata 上整合 160 个 OLTP 数据库的结果。

量度	非 EXADATA 仅 OLTP 负载	EXADATA 仅 OLTP 负载	结论
累计 TPS	10514	46915	提高 4.4 倍
平均响应时间	177 毫秒	132 毫秒	降低 25%
数据库数量	40	160	增加 4 倍
主要等待事件与平均等待时间	单元单块物理读取：47 毫秒，55% DBtime 日志文件同步：122 毫秒，31% DBtime 日志文件并行写入：23 毫秒	单元单块物理读取：11 毫秒，31% DBtime 日志文件同步：28 毫秒，10% dbtime 日志文件并行写入：5 毫秒	Exadata 确定了基本数据库 I/O 的优先级，以确保低延迟。Exadata 智能闪存明显扩展了 IO 带宽。

Exadata 智能闪存缓存特性和网络资源管理确定了数据库读取和写入操作的优先级（尤其是日志写入操作），从而确保了低延迟和高带宽。即使有 160 个数据库，平均日志文件并行写入也仍为 5 毫秒，日志文件同步为 28 毫秒，单元单块物理读取为 11 毫秒。在此示例中，负载受 CPU 的限制，系统非常稳定。

在有 200 个数据库的情况下，数据库再多 40 个，系统也仍能保持稳定，但是响应时间会略微超过 40 个非 Exadata 数据库的时间，如下所示。

量度	非 EXADATA 仅 OLTP 负载	EXADATA 仅 OLTP 负载	结论
累计 TPS	10514	48309	提高 4.5 倍
平均响应时间	177 毫秒	196 毫秒	增加 11%
数据库数量	40	200	增加 5 倍
主要等待事件与平均等待时间	单元单块物理读取：47 毫秒，55% DBtime 日志文件同步：122 毫秒，31% DBtime 日志文件并行写入：23 毫秒	单元单块物理读取：4 毫秒；11% DBtime 日志文件同步：31 毫秒，10% dbtime 日志文件并行写入：6 毫秒 DBWR 类型但没时间进行调优的等待（如写入完成等待、可用缓冲区等待和缓冲区繁忙等待）的 DBtime 为 45%。	Exadata 确定了基本数据库 I/O 的优先级，以确保低延迟。Exadata 智能闪存明显扩展了 IO 带宽。

这两个示例显示，如果青铜级层只有 OLTP 负载，则数据库整合密度非常好，可以提高 4 到 5 倍。

在实际应用中，OLTP 应用程序混合着一些报告、批处理和 ETL 活动。在有 40 个或更多个数据库和应用程序的大型整合环境中，经常可以发现像非 OLTP 这样的活动。这些其他活动会对 OLTP 响应时间和吞吐量产生不利影响。在本例中，向包含各种长期运行的查询的报告负载添加了一个 DW。在所有运行中，为数据仓库和报告负载使用实例囚笼，以防止这类负载淹没可用的 I/O 和 CPU 资源。对于非 Exadata 情况，40 个 OLTP 数据库加上 1 个数据仓库/报告数据库便达到 IOPS 上限；对于 Exadata 情况，160 个 OLTP 数据库加上 1 个数据仓库/报告数据库达到 CPU 资源使用上限。Exadata 特性再次确定关键数据库操作的优先级，并允许更高的吞吐量（6.3 倍），大幅缩短了响应时间（降低 50%）并提高了整合密度（4 倍）。

量度	非 EXADATA OLTP + DW/报表	EXADATA OLTP + DW/报表	结论
累计 TPS	8486	53662	提高 6.3 倍
平均响应时间	212 毫秒	104	缩短 50%
数据库数量	41	161	增加 4 倍
主要等待事件与平均等待时间	单元单块物理读取: 54 毫秒; 59% DBtime 日志文件同步: 104 毫秒, 23% DBtime 日志文件并行写入: 21 毫秒	单元单块物理读取: 4 毫秒; 13% DBtime 日志文件同步: 38 毫秒, 21% dbtime 日志文件并行写入: 4 毫秒	Exadata 可以更高效地进行 OLTP I/O 和分流扫描

对于白银级、黄金级和白金级应用程序，Oracle 建议留足 CPU 和内存空间。在本例中，未超过 Exadata 上的 CPU 和内存限制。整合密度较低，但是响应时间的缩短和吞吐量的倍增更为显著。

量度	非 EXADATA OLTP + DW/报表	EXADATA OLTP + DW/报表	结论
累计 TPS	8486	55422	提高 6.5 倍
平均响应时间	212 毫秒	14 毫秒	缩短 93%
数据库数量	41	81	增加 2 倍
主要等待事件与平均等待时间	单元单块物理读取: 54 毫秒, 59% DBtime 日志文件同步: 104 毫秒; 23% DBtime 日志文件并行写入: 21 毫秒	DB CPU 为主要等待对象, DBtime 为 42% 单元单块物理读取: 1 毫秒; 23% DBtime 日志文件同步: 3 毫秒, 12% dbtime 日志文件并行写入: 1 毫秒	不受 CPU 限制时, 可以提高性能并可预测响应时间。 在本示例中, 白银级层以上级别的整合密度为 2 倍

为保持环境的稳定性，应用了 Exadata 整合最佳实践。在很大程度上，这些配置最佳实践是固有的，已应用于我们的工程化系统。但是，以下更改和自定义将产生最大的影响：

1. 设置 HugePages（需要调整，以容纳所有数据库）。
2. 最大程度减少进程数并使用连接池。
3. 设置 ASM 进程数。
4. 调整 semmsl 和 semmns 设置。
5. 对数据仓库负载设置实例囚笼。
6. 使用 iorm 目标=AUTO。

附录 B：RMAN 活动数据库复制迁移

本附录简要介绍用最短的停机时间将 Oracle Database 12c 非 CDB 迁移到 PDB 的步骤。这一过程利用最新的 12c RMAN 复制活动数据库功能，使用 FROM SERVICE 执行活动数据库备份，并在整个网络中执行增量应用，所有这些操作都不需要额外的空间来临时存储任何备份文件。这些文件使用 Oracle Net 进行复制，并存储在新 CDB 的目标位置中。此示例假设使用 ASM 存储数据文件。目前，只能在源和目标环境字节序相同时执行非 CDB 到 PDB 插入。跨字节序的迁移将需要 Oracle GoldenGate 或某个导出/导入方法（传统导出/导入、跨字节序可传输表空间或跨字节序完整数据库可传输表空间）。

下面的过程着重介绍将 Oracle Database 12c 非 CDB 作为 PDB 插入到 CDB 的主要步骤。My Oracle Support 说明 1576755.1 中介绍了以下过程的详细示例以及传统迁移方法：[使用 ASM 进行文件存储迁移非 CDB 和 PDB 的逐步示例](#)。

将 Oracle Database 12c 非 CDB 数据库作为 PDB 插入：

- 将数据库升级到 Oracle Database 12c。此数据库将成为非 CDB 数据库。
- 创建一个 CDB 或选择现有 CDB 作为迁移目标。要创建 CDB，可以使用 DBCA 或通过 SQL*Plus 使用 CREATE DATABASE 命令完成。请注意，无需预先创建 PDB，PDB 将在插入操作期间创建。
- 选择性地创建物理备用数据库。
- 使用 Oracle Database 12c RMAN，同时连接到原始的非 CDB 和新 CDB。这些连接必须都使用 Oracle Net，并且非 CDB 应当作为目标进行连接，CDB 则应作为克隆数据库进行连接。

```
RMAN> connect target <user>@non-cdb as sysbkup
RMAN> connect clone <user>@cdb as sysbkup
```
- 执行初始活动复制，将源数据库的映像文件复制到目标站点。如果使用 OMF，RMAN 会将文件恢复到正确位置。
- 在目标 CDB 上创建临时实例。
- 将非 CDB 控制文件恢复到目标环境中的临时位置。
- 将刚刚恢复的数据文件编录到临时实例上的控制文件中，并发出 SWITCH DATAFILE TO NEW 命令。
- 创建插入 SQL 语句，用于将非 CDB 作为 PDB 插入。在创建语句中使用 SOURCE_FILE_DIRECTORY 子句，并指定在步骤 5 中恢复文件所在的目录位置。
- 可以根据需要多次使用 FROM SERVICE 子句进行恢复，该子句将执行非 CDB 的增量备份，并将更改应用于目标 CDB 上的文件中。
- 当您做好了执行拔出操作的准备时，执行以下步骤：

- 1. 彻底关闭非 CDB，并以只读方式打开。
- 2. 在非 CDB 上创建清单 XML 文件。
- 3. 执行从非 CDB 到目标副本的最终增量应用。
- 4. 将清单复制到目标主机。
- 5. 使用清单 XML 文件创建 PDB。
- 6. 打开新 PDB。
- 7. 备份新 PDB。
- 8. 为 PDB 设置应用程序访问权限。

推荐的参数设置	相同的参数实践仍然适用；但是，大部分默认系统设置应当足够了。
维护和管理考虑事项	<p>Oracle 软件和文件系统空间</p> <p>采用一些 Oracle 数据库软件版本的相同维护实践。采用 Exadata 所有者指南中所述的相同回收空间实践。</p> <p>对于 MAA，请参阅本白皮书的以下部分：</p> <p>高可用性参考架构 — 青铜级、白银级、黄金级和白金级</p> <p>意外停机的高可用性</p> <p>计划维护的高可用性</p>

甲骨文（中国）软件系统有限公司

北京远洋光华中心办公室

地址：北京市朝阳区景华南街5号远洋光华中心C座21层
邮编：100020
电话：(86.10) 6535-6688
传真：(86.10) 6515-1015

北京汉威办公室

地址：北京市朝阳区光华路7号汉威大厦10层1003-1005单元
邮编：100004
电话：(86.10) 6535-6688
传真：(86.10) 6561-3235

北京甲骨文大厦

地址：北京市海淀区中关村软件园24号楼甲骨文大厦
邮编：100193
电话：(86.10) 6106-6000
传真：(86.10) 6106-5000

北京国际软件大厦办公室

地址：北京市海淀区中关村软件园9号楼国际软件大厦二区308单元
邮编：100193
电话：(86.10) 8279-8400
传真：(86.10) 8279-8686

北京孵化器办公室

地址：北京市海淀区中关村软件园孵化器2号楼A座一层
邮编：100193
电话：(86.10) 8278-6000
传真：(86.10) 8282-6401

上海名人商业大厦办公室

地址：上海市黄浦区天津路155号名人商业大厦12层
邮编：200001
电话：(86.21) 2302-3000
传真：(86.21) 6340-6055

上海腾飞浦汇大厦办公室

地址：上海市黄浦区福州路318号腾飞浦汇大厦508-509室
邮编：200001
电话：(86.21) 2302-3000
传真：(86.21) 6391-2366

上海创智天地10号楼办公室

地址：上海市杨浦区淞沪路290号创智天地10号楼512-516单元
邮编：200433
电话：(86.21) 6095-2500
传真：(86.21) 6107-5108

上海创智天地11号楼办公室

地址：上海市杨浦区淞沪路303号创智天地科教广场3期11号楼7楼
邮编：200433
电话：(86.21) 6072-6200
传真：(86.21) 6082-1960

上海新思大厦办公室

地址：上海市漕河泾开发区宜山路926号新思大厦11层
邮编：200233
电话：(86.21) 6057-9100
传真：(86.21) 6083-5350

广州国际金融广场办公室

地址：广州市天河区珠江新城华夏路8号合景国际金融广场18楼
邮编：510623
电话：(86.20) 8513-2000
传真：(86.20) 8513-2380

成都中海国际中心办公室

地址：成都市高新区交子大道177号中海国际中心7楼B座02-06单元
邮编：610041
电话：(86.28) 8530-8600
传真：(86.28) 8530-8699

深圳飞亚达科技大厦办公室

地址：深圳市南山区高新南一道飞亚达科技大厦16层
邮编：518057
电话：(86.755) 8396-5000
传真：(86.591) 8601-3837

深圳德赛科技大厦办公室

地址：深圳市南山区高新南一道德赛科技大厦8层0801-0803单元
邮编：518057
电话：(86.755) 8660-7100
传真：(86.755) 2167-1299

大连办公室

地址：大连软件园东路23号大连软件园15号楼502
邮编：116023
电话：(86.411) 8465-6000
传真：(86.755) 8465-6499

苏州办公室

地址：苏州工业园区星湖街328号苏州国际科技园5期11幢1001室
邮编：215123
电话：(86.512) 8666-5000
传真：(86.512) 8187-7838

沈阳办公室

地址：沈阳市和平区青年大街390号皇朝万鑫国际大厦A座39层3901&3911室
邮编：110003
电话：(86.24) 8393-8700
传真：(86.24) 2353-0585

济南办公室

地址：济南市冻源大街150号中信广场11层1113单元
邮编：250011
电话：(86.531) 6861-1900
传真：(86.531) 8518-1133

南京办公室

地址：南京市玄武区洪武北路55号置地广场19层1911室
邮编：210018
电话：(86.25) 8579-7500
传真：(86.25) 8476-5226

西安办公室

地址：西安市高新区科技二路72号西安软件园零壹广场主楼1401室
邮编：710075
电话：(86.29) 8834-3400
传真：(86.25) 8833-9829

重庆办公室

地址：重庆市渝中区邹容路68号大都会商厦1611室
邮编：400010
电话：(86.23) 6037-5600
传真：(86.23) 6370-8700

杭州办公室

地址：杭州市西湖区杭大路15号嘉华国际商务中心810&811室
邮编：310007
电话：(86.571) 8168-3600
传真：(86.571) 8717-5299

福州办公室

地址：福州市五四路158号环球广场1601室
邮编：350003
电话：(86.591) 8621-5050
传真：(86.591) 8801-0330

南昌办公室

地址：江西省南昌市西湖区沿江中大道258号
皇冠商务广场10楼1009室
邮编：330025
电话：(86.791) 8612-1000
传真：(86.791) 8657-7693

呼和浩特办公室

地址：内蒙古自治区呼和浩特市新城区迎宾北路7号
大唐金座19层北侧1902-1904室
邮编：010051
电话：(86.471) 3941-600
传真：(86.471) 5100-535

郑州办公室

地址：河南省郑州市中原区中原中路220号
裕达国际贸易中心A座2015室
邮编：450007
电话：(86.371) 6755-9500
传真：(86.371) 6797-2085

武汉办公室

地址：武汉市江岸区中山大道1628号
武汉天地企业中心5号大厦23层2301单元
邮编：430010
电话：(86.27) 8221-2168
传真：(86.27) 8221-2168

长沙办公室

地址：长沙市芙蓉区韶山路159号通程国际大酒店1311-1313室
邮编：410011
电话：(86.731) 8977-4100
传真：(86.731) 8425-9601

石家庄办公室

地址：石家庄市中山东路303号石家庄世贸广场酒店14层1402室
邮编：050011
电话：(86.311) 6670-8080
传真：(86.311) 8667-0618

昆明办公室

地址：昆明市三市街六号柏联广场写字楼11层1103A室
邮编：650021
电话：(86.871) 6402-4600
传真：(86.871) 6361-4946

合肥办公室

地址：安徽省合肥市蜀山区政务新区怀宁路1639号平安大厦18层1801室
邮编：230022
电话：(86.551) 6595-8200
传真：(86.551) 6371-3182

广西办公室

地址：广西省南宁市青秀区民族大道136-2号华润大厦B座2302室
邮编：530028
电话：(86.771) 391-8400
传真：(86.771) 577-5500



数据库整合高可用性最佳实践

2014 年 4 月

作者: Frank Kobylanski, Rene Kundersma

Sue K. Lee, Joe Meeks, Lawrence To

公司网址: <http://www.oracle.com> (英文)

中文网址: <http://www.oracle.com/cn> (简体中文)

销售中心: 800-810-0161

售后服务热线: 800-810-0366

培训服务热线: 800-810-9931

欢迎访问:

<http://www.oracle.com> (英文)

<http://www.oracle.com/cn> (简体中文)

版权© 2014 归 Oracle 公司所有。未经允许, 不得以任何形式和手段复制和使用。

本文的宗旨只是提供相关信息, 其内容如有变动, 恕不另行通知。Oracle 公司对本文内容的准确性不提供任何保证, 也不做任何口头或法律形式的其他保证或条件, 包括关于适销性或符合特定用途的所有默示保证和条件。本公司特别声明对本文档不承担任何义务, 而且本文档也不能构成任何直接或间接的合同责任。未经 Oracle 公司事先书面许可, 严禁将此文档为了任何目的, 以任何形式或手段(无论是电子的还是机械的)进行复制或传播。

Oracle 是 Oracle 公司和/或其分公司的注册商标。其他名字均可能是各相应公司的商标。