ORACLE

Best Practices for Database Consolidation

A Guide for Implementation
Including MAA Reference Architectures

September 2021 | Version [4.0]

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of Contents

Executive Overview

Information Technology (IT) organizations are under pressure to manage costs, increase agility, ensure availability, reduce security risks, and deliver the proper level of system performance to meet business objectives. Consolidation of databases and other systems has emerged as a common strategy used by IT organizations to meet these objectives. This paper outlines best practices for consolidation of Oracle databases, including use of key technologies including O/S virtualization, resource management, Oracle's Multitenant database capabilities, Oracle Exadata Database Machine, Oracle Exadata Cloud@Customer, Oracle Exadata Cloud Service, and Oracle Autonomous Database which also resides on Exadata.

On-Premise and Cloud Oracle Exadata Database platform provides the ideal platform for consolidation of databases at lower cost, with simplified management, industry-leading availability, enterprise-grade security, and performance capabilities that can be employed and managed to meet business objectives. Exadata incorporates the full range of Oracle Maximum Availability Architecture (MAA) best practices, and is combined with standardized operational practices, enabling customers to achieve the highest levels of reliability possible in a consolidated database environment.

Oracle Exadata Database Machine can be deployed in traditional on-premises environments, in the Oracle Public Cloud, and in the Oracle Cloud@Customer model. Database consolidation as outlined in this paper applies to all of these deployment models. In addition, Oracle Autonomous Database automatically incorporates the principals outlined in this document, allowing customers to consolidate databases with greater ease.

This paper will explore these topics in detail to give IT organizations the best possible guidance for consolidating Oracle databases.

Overview of Database Consolidation

This paper outlines best practices for consolidation of Oracle databases to achieve the primary business objectives of managing costs, improving manageability of databases, ensuring availability, improving security, and delivering the proper level of performance for business processes that rely on Oracle databases. While many of the recommendations included in this document can be applied to any platform, special attention will be given to the Oracle Exadata Database Machine and how it delivers on these business objectives on-premise and in the Cloud.

Database Consolidation Defined

Database consolidation involves placing multiple databases onto a single set of computing infrastructure, either on-premise or in the Oracle Cloud. Database consolidation is similar to but different from server consolidation, which involves consolidating multiple physical servers into a single physical server running Virtual Machines. The primary purpose of database consolidation is to reduce the cost of database infrastructure by taking advantage of more powerful servers that have come onto the market that include dozens of processor cores in a single server. Databases consolidation may involve any of the following configurations:

- Multiple Databases on a Single Physical Server
- Multiple Databases on a Single Virtual Machine
- Multiple Databases on a Cluster of Physical Servers
- Multiple Databases on a Cluster of Virtual Machines

Database consolidation allows more databases to run on fewer servers, which reduces infrastructure, but also reduces operational costs. Each physical server represents an Information Technology asset or "Configuration Item" (CI) that must be placed into a data center rack, connected, secured, and maintained. Each Virtual Machine represents another Configuration Item (CI) that also must be deployed, connected (virtually), secured, and maintained. Database Consolidation also applies in Cloud environments such as the Oracle Database Cloud Service and Oracle Exadata Cloud Service.

Oracle databases can also be consolidated into a cluster of physical servers or virtual machines. Oracle Database clustering capabilities include active/passive clustering, as well as active/active clustering using Oracle Real Application Clusters. Clustering is used to increase availability of Oracle databases, enable rolling patching, and to provide for scalability across multiple physical servers or virtual machines.

Oracle databases can also be further consolidated into what are known as Container Databases using the Oracle Multitenant option that was first introduced in Oracle Database 12c. The Multitenant option allows multiple databases to be managed under a single Container Database, with a single backup, single disaster recovery configuration, and single clustering configuration.

Database Consolidation Goals

While database consolidation is intended to reduce the cost of computing infrastructure and the cost of Information Technology operations, database consolidation must also take into account the following factors to meet business requirements:

- Reduce Cost
- Simplify Administration
- Improve Security and Isolation
- Meet Availability Goals
- Deliver Required Performance

Cost reduction is the primary business driver of database consolidation. The cost saving advantages of consolidation apply in traditional on-premise data centers, as well as when using Cloud services. Consolidation of databases enables

higher utilization of on-premise computing assets as well as systems deployed using Cloud services. Consolidation also enables dynamic, automatic reallocation of computing resources (whether on-premise or Cloud) using the Resource Management capabilities of the Oracle database.

Simplified Administration of databases in a consolidated environment begins with standardization and a reduction in the variety of configurations that need to be maintained. Database consolidation enables enterprises to manage many databases as one, reducing management of physical servers, virtual machines, and even the number of databases via the use of Container Databases with Oracle Multitenant. Simplifying administration enables enterprises to focus efforts on ensuring availability of databases and business applications.

Improved Security and Isolation of databased is achieved in a consolidation environment through a reduction in the number of points of vulnerability, an increase in standardization across databases, and a reduction in the administrative workload required to secure those databases. Database consolidation can also raise the stakes in security, since an attacker can potentially gain greater access to more data than in a non-consolidated environment. Oracle Database includes a wide array of security capabilities itself, and these capabilities are enhanced as part of the Exadata Database Machine to further mitigate security vulnerabilities.

Meeting Availability Goals of databases and applications as dictated by business requirements is critical in a consolidation environment due to the mere fact that multiple databases reside on the same infrastructure. This paper outlines the proper configuration to harden systems against failures, deploy multiple layers of isolation to reduce the scope of impact (or "blast radius") of failures, as well as ensuring fault tolerance and proper operational practices that can achieve continuous availability during maintenance events, upgrades, or in response to failure of infrastructure components.

Delivering the Required Performance of databases and applications is a key goal of any consolidation environment. Performance of one database should not be impacted by issues that arise on a separate database. Databases and the business applications serviced by those databases must receive the proper amount of computing resources assigned to those databases and business applications. The Oracle Database Resource Manager (DBRM) and Exadata I/O Resource Manager (IORM) provide the resource management tools necessary to provide the required level of control to ensure the proper level of database and application performance.

The Role of Virtual Machines in Database Consolidation

While virtual machines (VMs) are universally used for server consolidation, VMs play a specific role in consolidation of Oracle databases. Virtual Machines (also known as Logical Partitions or Logical Domains on some platforms) allow a single physical server to simultaneously run multiple Operating System (O/S) images, keeping the content of those O/S images isolated from each other. Each O/S is allocated its own disk, memory, CPU, and virtual network interfaces, and each O/S has its own separate system processes and is isolated from other O/S images running on the same physical server. The primary benefit of Virtual Machines is to provide isolation at the O/S layer when multiple systems are consolidated onto the same physical server.

Virtual Machine (VM) technology plays an important role in database consolidation to provide isolation where necessary for security purposes. For example, Virtual Machines can be used to provide separation between Development and Test environments, or between multiple production databases. Virtual Machines can also be used to address isolation for data security requirements including PCI (Payment Card Industry) data and HIPAA (Healthcare Insurance Portability and Accountability Act) data, including use of distinct VLAN-tagged networks for each VM Cluster.

Virtual Machines also provide advantages during system software maintenance by isolating groups of databases during upgrade, as well as providing more flexible fallback methods such as O/S snapshots. However, each Virtual Machine brings additional administrative overhead, including installation, patching, security, and management. Oracle recommends against the approach of deploying one database per VM due to the increased administrative overhead. Rather, Oracle recommends a more judicious use of Virtual Machines, including deployment of small numbers of VMs per physical machine, with each VM hosting multiple databases, container databases, and/or pluggable databases. Bare Metal machines (without VM) are recommended for cases that require the highest level of performance as well as for extremely large-scale databases with hundreds of terabytes of data and hundreds or thousands of users.

The Role of Real Application Clusters in Database Consolidation

Oracle Real Application Clusters (RAC) provide key features for availability and scalability in a database consolidation environment. Server clustering involves grouping together two or more servers (or Virtual Machines) that work together in a cooperative fashion under the control of software known as clusterware (such as Oracle Clusterware). Oracle Real Application Clusters (RAC) allows an Oracle database to span the servers of a cluster and actively operate on ALL servers in the cluster simultaneously in what is often referred to as active/active clustering. Oracle RAC allows databases to scale "horizontally" by adding servers (or VMs) to the cluster. RAC also provides the ability to continue operating during server (or VM) maintenance or when a server (or VM) failure occurs. RAC allows a database to continuing operation in the event of:

- Server Hardware Maintenance
- O/S Maintenance
- VM Hypervisor Maintenance
- Server Hardware Failure
- O/S Failure (ex: Kernel Panic)
- Database Software Updates
- Oracle Grid Infrastructure Updates

RAC facilitates hardware and software maintenance including quarterly security patches, database software fixes, and Oracle Grid Infrastructure Updates. RAC provides for high availability in combination with Oracle Application Continuity, which allows user sessions to continue operating (without loss of in-flight transactions) during server failures by moving any active sessions to other surviving servers within the RAC cluster.

Non-RAC databases can be used in a database consolidation environment for lower-tier, non-critical databases that do not require RAC for scalability. However, these databases will not have protection from the events outlined above and may represent a greater operational burden on administrators due to this lack of clustering benefits. Even non-critical databases should use Real Application Clusters if possible, simply to reduce administrative burden.

The Importance of Resource Management in Consolidation

Oracle Database Resource Manager (DBRM) and I/O Resource Manager on Exadata systems (IORM) are critical components in the operation of any Oracle database and are even more critical in database consolidation. The Oracle Resource Managers (DBRM and IORM) provide the controls necessary to apportion resources across databases and prevent resource conflicts between databases that can result in performance issues of one database impacting other databases on the same system. Without DBRM and IORM, administrators use a combination of tools that are more difficult to manage and don't provide the same degree control over resource allocations for Oracle Databases. The Oracle Resource Managers (DBRM and IORM) allow administrators to:

- Allocate and manage system resources for databases
- Ensure proper level of database & business application performance
- Ensure stability of databases and systems

Oracle Database Resource Manager and I/O Resource Manager (for Exadata systems) enable database administrators to allocate system resources (CPU, memory, processes, I/O, and network) to databases and manage those allocations over time with changing business demands. The Oracle Resource Managers allow system resources to be re-allocated easily or even dynamically to deliver the level of performance needed for business applications. Properly configured resource management also improves system stability by preventing resource starvation caused by misbehaving applications, databases or systems.

The Oracle database also provides I/O resource management for non-Exadata systems, although without the deep integration found on Exadata. Please refer to the Resource Management section of this document for more information on this topic.

Methods for Database Consolidation

There are 3 methods for consolidating Oracle databases onto physical machines or Virtual Machines as discussed below. Virtual Machines are used to provide isolation at the O/S layer, while consolidation refers to multiple databases running within a Virtual Machine, within a non-virtualized physical sever, or within a cluster comprised of physical servers or Virtual Machines. Multiple databases can be consolidated using one of these methods:

- Multiple Databases per Physical Server, Virtual Machine, or Cluster
- Oracle Multitenant
- Schema Consolidation

These methods can be used separately or in combination to consolidate databases onto a set of physical machines or virtual machines.

Multiple Databases per Physical or Virtual Machine

Oracle databases can be consolidated onto a single physical machine or Virtual Machine, and multiple RAC (Real Application Cluster) databases can be consolidated onto a cluster of physical machines or cluster of Virtual Machines. These can be the same or different versions of the Oracle database and can use either shared or dedicated Oracle software (shared or dedicated ORACLE_HOMEs). The Instance Caging feature of Oracle Database Resource Manager (DBRM) enables the allocation of CPU across multiple consolidated databases on a single server or cluster of servers. The I/O Resource Manager (IORM) features of Exadata extends these resource management features into the storage layer of Exadata systems. Please refer to the related section of this document for more information on Oracle Resource Manager.

Database Consolidation Using Oracle Multitenant

Oracle Multitenant provides the efficiency gains achieved previously with schema consolidation (see below for more information), while also providing the needed isolation between applications. Oracle Multitenant introduced the concept of Pluggable Databases (PDB) that reside within a Container Database (CDB). Pluggable Databases can be plugged into one Container Database, then moved to another Container Database through "un-plug" and "plug" operations. This architecture greatly increases the number of databases that can reside on a single server as shown below.

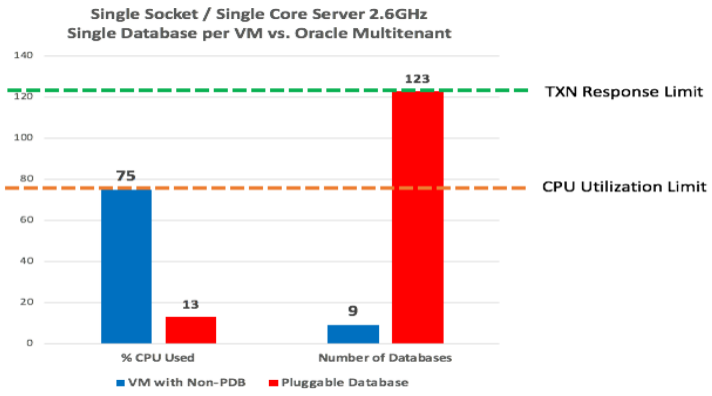

Figure 1: Consolidation using Virtual Machines compared to Database Virtualization using Oracle Multitenant

Oracle internal tests have demonstrated that Oracle Multitenant increases consolidation density compared to single-instance databases running on dedicated Virtual Machines. In Figure 1, 123 databases were consolidated in the same server using Oracle Multitenant, while the same server would only service 9 databases using the single database per Virtual Machine approach. The single database per VM approach quickly reached the CPU utilization limit (75%), whereas 123 databases were consolidated onto the server using Oracle Multitenant before reaching the transaction

response time limit. Oracle Multitenant also reduces administrative burden by allowing many databases to be managed as a single database.

Schema Consolidation

Schema consolidation is an approach that was used prior to the advent of Oracle Multitenant to combine schemas from multiple databases into a single database, making more efficient use of system resources and enabling a much higher consolidation density. While schema consolidation makes more efficient of computing resources, this approach often requires application changes because it does not provide the necessary degree of isolation that Oracle Multitenant provides between applications and does not facilitate isolation for maintenance operations such as upgrade. Oracle Multitenant has become the preferred solution for high density consolidation because it addresses the complexities found with schema consolidation. Schema consolidation is a useful approach in some circumstances, such as combining multiple databases belonging to a single application but will not be addressed further in this document.

Why Exadata for Consolidation

Oracle Exadata Database Machine for on-premise and for Oracle cloud is an engineered system purpose-built to provide optimal performance, availability, and manageability for Oracle Databases. Its scalable architecture and advanced software capabilities make it ideally suited as a standard database platform for consolidation of Oracle databases. Exadata addresses each of the business goals outlined earlier including:

- Reduce Cost
- Simplify Administration
- Improve Security and Isolation
- Meet Availability Goals
- Deliver Required Performance

While Oracle databases can certainly be consolidated onto non-Exadata platforms, there are significant advantages to using Exadata for consolidation.

Reducing Costs with High Consolidation Density

Oracle's benchmark testing has shown that substantially higher consolidation densities can be achieved with Exadata than similarly sized non-Exadata systems. The workload in these tests included from 150 to 300 OLTP databases running Oracle Real Application Clusters (RAC) on an Exadata X8-2 Quarter Rack with Oracle Multitenant. Exadata delivered transaction response times under the defined 3 millisecond cap in these tests, as the number of databases was increased from 150 to 300 as seen in Figure 2. Newer generations of Exadata such as X9M will exceed the results shown in this test.

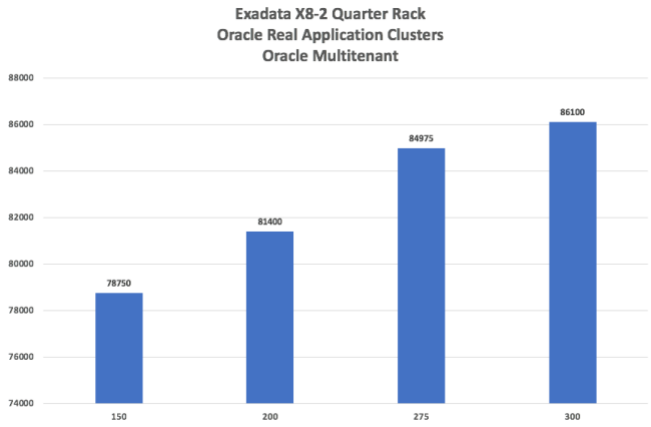

Figure 2: Exadata Consolidation Density and Performance

Exadata offers high I/O bandwidth, high IOPS (Input/Output Operations Per Second), and high throughput rates, combined with offloading of database processing into storage to deliver higher consolidation density than traditional platforms. Exadata Elastic Expansion provide the ability to add storage with zero downtime as more databases are added to the system or when data volumes grow.

Simplification through Standard Configuration & Management Practices

Thousands of Oracle customers run Exadata Database Machines, and standardized practices have been developed for the management and administration of Exadata. Standardization decreases complexity of the operational environment, improves security, enhances availability, and reduces costs. Exadata is a single platform that can run workloads of any scale and complexity, making Exadata the only single platform that can be used for all workloads.

Enhanced Security with Exadata

The Exadata platform includes a number of features that deliver improved security out of the box, without placing additional burden on system administrators. Consolidation of databases results in fewer systems to secure, and Exadata provides a more secure platform by default. The capabilities delivered with Exadata include:

- Defense In-Depth
- Principal of Least Privilege
- Pre-Integrated Updates & Security Fixes
- Minimal Attack Surface
- Pre-Scanned Software
- Automated Intrusion Detection Environment (A.I.D.E.)

Oracle follows the approach of defense in-depth in the security design of Exadata, which includes strong authentication, encryption capabilities, database users and roles, and system-level (O/S) security. Oracle implements the principal of "least privilege", enabling customers to assign separate administrative roles such as for infrastructure administration, and multiple sub-divisions of database administrator roles.

Exadata includes security fixes that are integrated with each quarterly release, monthly security releases, and emergency fixes to address zero-day vulnerabilities. These updates are pre-integrated across the entire stack including O/S, Virtual Machines, Clusterware, Logical Volume Manager, drivers, storage software, networking, and the database software itself. Exadata incorporates a minimal attack surface by only including the software components required specifically to run the Oracle database. Oracle conducts multiple security scans of each Exadata software release using industry leading security scanners. These scans include malware scanning, vulnerability scanning, and configuration compliance. Customers should also run their own scans on deployed systems to ensure against any deviations from the delivered configurations.

Exadata also includes the Advanced Intrusion Detection Environment (A.I.D.E.) as of the 19.1 Exadata software release. AIDE automatically detects modification of critical files by checking SHA256 hash signatures against a database of expected values. For more information, please refer to the Exadata Security Guide: https://docs.oracle.com/en/engineered-systems/exadata-database-machine/dbmsq/security-guide-exadata-database-machine.pdf

Ensuring Availability with Exadata and MAA

Exadata is the leading platform for Maximum Availability Architecture (MAA) due to extensive development investment in the Exadata platform to achieve the highest levels of availability, data protection and ensuring best performance for Data Guard and Oracle GoldenGate. These capabilities include fast node death detection, automatic storage failure detection and rebalance, automatic network fault detection, as well as fast detection and reconfiguration during instance failure. Oracle has made continuous improvements in all areas of availability with each Exadata software release. The high performance of Exadata contributes to both higher consolidation density as well as the HA and DR features such as faster RAC instance recovery and DR standby recovery. Please refer to the Exadata MAA white paper for more information (https://www.oracle.com/technetwork/database/features/availability/exadata-maa-131903.pdf).

Exadata Performance

The high-performance capabilities of Exadata are often used to deliver high performance, but those same performance advantages are also used to drive higher consolidation density that cannot be achieved with other platforms. Exadata reduces or eliminates most common bottlenecks and optimizes database transactions through smart Flash, smart high-bandwidth internal InfiniBand network, and smart storage that offloads Oracle database processes and SQL processing into the Exadata storage layer. The advanced resource management capabilities of Exadata (including both DBRM and IORM) allow customers to devote the proper amount of resources to each database. Exadata therefore allows customers to manage performance and deliver the desired level of performance required by the business. While Exadata is often known for delivering the highest levels of performance, Exadata can also be used to deliver performance consistent with a configured service level through the use of resource management.

In addition to enabling greater consolidation density using stand-alone or Multitenant databases, the Exadata platform also brings comprehensive resource management capabilities under unified control. Performance of databases can be easily

managed on the Exadata platform through simple allocation and management of system resources such as CPU, memory, processes, and I/O. Exadata extends the Oracle database-level resource management capabilities to include the Exadata storage network and I/O resources as well. Please refer to the section of this document on Resource Management for discussion of resource management capabilities generally, as well as specifically for Exadata environments.

Exadata Deployment Models

Exadata Database Machine can be deployed using three deployment models, all of which use the same underlying Exadata technology as follows:

- Exadata On-Premises
- Exadata Cloud@Customer
- Exadata Cloud Service
- Autonomous Database

All of these deployment models bring the same benefits in terms of managing costs, simplifying operations, ensuring security, meeting availability goals, and providing the needed level of database performance. Deployment in the Oracle Cloud Service and Cloud@Customer bring additional capabilities that further simplify operations through Oracle's Cloud tooling and Cloud Service offerings. The Oracle Cloud also enhances security further through use of mandatory data encryption, and Cloud network security.

Establishing the Foundation for Consolidation

Consolidation of databases relies on a solid foundation of standardized system configurations and operational practices to meet the business goals of cost containment, simplified operations, and improved security, while delivering the required level of application performance. Standardized configurations must also provide sufficient flexibility to meet the needs of the business.

In this section, we will address a number of critical configuration standards that allow organizations to meet these goals. However, these are not intended to be comprehensive recommendations for all configuration settings for each component. Please refer to the additional information section for a complete set of recommendations in each area.

Exadata Cloud and Exadata Cloud@Customer

Exadata Cloud Service, Exadata Cloud@Customer and Autonomous Database incorporate the best-practice recommendations included this section, providing the foundation needed for consolidation. Both of these offerings include infrastructure management services as well as automation needed to accomplish the business goals of database consolidation.

Highly Available Storage

Consolidation of databases increases the performance and availability requirement for storage associated with those systems. Any downtime of storage has the potential to impact many databases at once. For Exadata systems HIGH redundancy storage (triple mirroring) should be used in any consolidation environment to provide improved protection against hardware failures, as well as to allow continued operation during maintenance. All Exadata systems in Oracle Cloud has configured with high redundancy storage (disk groups).

Virtual Machine Configuration

In cases where virtual machines (VMs) are used, each physical server can be configured with 2-4 Virtual Machines to provide isolation where necessary without adding excessive complexity and administrative overhead. Virtual Machines are optional in on-premises Exadata configurations but are integral to the configuration of Exadata Cloud Service and Exadata Cloud@Customer, where they provide administrative isolation between Oracle and customer operations teams. In the case of multi-VM configurations, it is important to understand the maximum number of Virtual Machines that can be configured for each deployment type. For more information on use of Virtual Machines with Exadata, please refer to the following:

OVM (Oracle Virtual Machine): https://www.oracle.com/technetwork/database/availability/exadata-ovm-2795225.pdf

KVM (Kernel Virtual Machine): https://www.oracle.com/a/tech/docs/exadata-kvm-overview.pdf

Memory Configuration

While this section does not include a comprehensive treatment of server configuration, special attention should be paid to the topics below in a database consolidation environment. Memory management is especially important in a database consolidation environment due to the potential for impact across multiple databases. Customers should be using the following memory management settings of Linux and the Oracle database:

- Linux Huge Pages
- Oracle Database USE_LARGE_PAGES parameter

A portion of system memory should be allocated to HugePages in Linux in proportions designed for the specific workload type (OLTP or Data Warehouse), and the Oracle SGA should be placed within the Huge Pages area. Specifying USE_LARGE_PAGES=ONLY in the Oracle database will ensure the SGA is placed in the HugePages area. While this prevents the affected database instance from starting, the purpose is to prevent memory starvation that can potentially impact other databases on the system. Please refer to MOS 361468.1 and MOS 401749.1 for more details on use of Linux Huge Pages.

Multitenant Container Databases

Each physical or virtual machine running Oracle Multitenant databases should include provisions for at least 2 Container Databases (CDBs) to facilitate upgrades. Pluggable Databases (PDBs) can be moved between CDBs during upgrade, or an entire CDB can be upgraded to reduce administrative complexity. One CDB on each server (physical or virtual) is used for the current database version, while the 2nd CDB is used for the next database version during upgrade. Databases can be moved from one container to the next during upgrade using relocate operations.

Configuration Checking Tools

Oracle provides a number of configuration checking tools that are consistently updated with the latest best practices configuration checks. These tools include the following:

- DB Security Assessment Tool (DBSAT)
- exachk

The Oracle Database Security Assessment tool (DBSAT) is used for checking security vulnerabilities of Oracle databases. The exachk (EXA Check) tool is a holistic health check of the entire Exadata system evaluating storage, network operating system, Grid Infrastructure and database. It highlights when key Exadata MAA configuration practices or software vulnerabilities are exposed in target system.

Data Guard & Active Data Guard (Read Replicas)

Oracle Data Guard has provided industry leading Disaster Recovery (or "standby" database) capabilities for more than two decades, beginning with Oracle9i. Data Guard standby databases are kept in sync with a "primary" source (production) database through automatic apply of database transactions contained in redo logs. Data Guard standby databases normally consume a fraction of the resource consumption found on the source (or "primary") database until the database is activated during a disaster scenario.

While Data Guard standby databases are intended primarily for use during disaster recovery, Active Data Guard (ADG) enables a standby database to be actively queried as a "Read Replica" while the database is being updated from the stream of redo changes generated by the primary database. Data Guard also provides additional protection, especially with automatic block repair of physical data corruption.

Exadata Infrastructure & DB Version Interoperability

The Exadata Infrastructure layer consists of the software layers that underly databases running on the Exadata system. Oracle Databases, Container Databases, and Pluggable Databases are installed on top of these infrastructure layers. Multiple Oracle Database versions are supported with each release of the underlying Exadata Infrastructure layers. Version interoperability is outlined in MOS Note # 888828.1 and Exadata Cloud Service Software Version 2333222.1.

Consolidation of databases onto a common set of infrastructure raises the stakes in terms of availability. Oracle Maximum Availability Architecture (MAA) best practices define four HA reference architectures that address the complete range of availability and data protection required to address these needs in a database consolidation environment. The architectures, or HA tiers, are designated PLATINUM, GOLD, SILVER, and BRONZE, delivering the service levels described in Figure 3 below.

Figure 3: HA and Data Protection Service Levels

Each tier uses a different MAA reference architecture to deploy the optimal set of Oracle HA capabilities that reliably achieve a given service level at the lowest cost and complexity. These architectures explicitly address all types of unplanned outages including data corruption, component failure, system and site outages, as well as planned outages due to software maintenance, migrations, or database upgrades.

The Bronze MAA reference architecture provides basic database service at the lowest possible cost. A reduced level of high availability and data protection is accepted in exchange for reduced cost and implementation complexity. This architecture may be suitable for databases used for test, development, and less critical production applications and databases. The Bronze architecture uses the high availability capabilities included in Oracle Enterprise Edition. Bronze defaults to the Oracle Database single-instance or multitenant architecture. Oracle Restart or Oracle Clusterware high availability capabilities are used to restart a failed instance, database server, or any relevant managed service. In the worst-case scenario of a complete site outage, there is additional time required to perform these tasks at a secondary location which may result in days of downtime.

In the Bronze reference architecture, a local backup within the same data center is always recommended for the fastest recovery. Oracle also recommends maintaining a second copy of backups in a remote data center to protect against site outages and disasters. You can use Oracle Cloud Database Backup Service to maintain a cloud-based backup of on-premises databases. Silver provides an additional level of availability for databases that require minimal or zero downtime in the event of database instance or server failure as well as many types of planned maintenance such as operating system, Database, and Grid Infrastructure software updates including required security updates. Silver adds clustering technology using Oracle Real Application Clusters (RAC) for high availability. RMAN provides database-optimized backups to protect data and restore availability should an outage prevent the cluster from being able to restart. Application Continuity provides reliable replay of in-flight transactions, masking outages from users.

Gold substantially raises the service level for business-critical applications by providing low downtime solutions for all unplanned outages and major database software upgrades. Gold adds database-aware replication technologies using Oracle Active Data Guard to synchronize one or more replicas of the production database, providing data protection and availability. Far Sync extends real time data protection and availability to ensure support for long distances between production and disaster recovery sites. Database-aware replication greatly increases HA, disaster recovery, and data

protection beyond what is possible with storage replication technologies. This also reduces cost while improving return on investment by enabling active use of all replicas at all times.

Platinum builds upon the Gold tier of protection by adding Oracle GoldenGate to provide bi-directional capabilities that enable zero downtime upgrades and migrations. Global Data Services in the Platinum Tier provides automated service management and workload balancing for replicated database environments. In addition, the platinum tier includes optional configurations for Edition-Based Redefinition (EBR), which enables application upgrades to be performed online, and sharding, which provides intelligent horizontal partitioning of data across a farm of independent databases using a common key (i.e. shard key). While each of these technologies requires additional effort to implement at the platinum tier, these technologies deliver substantial value for the most critical applications where downtime and data loss are not an option.

The MAA reference architectures provide a standard infrastructure optimized for Oracle Database that enable enterprises to select the level of HA appropriate for different service level requirements. Standardization reduces cost and simplifies movement of databases from one HA tier to the next, or from one hardware platform to another should business requirements change. It should be noted that all tiers also included Oracle Enterprise Manager Cloud Control for configuration, monitoring, alerting, and management in addition to the specific set of Oracle HA capabilities that are described for a given tier.

Please refer to the MAA architecture documentation (https://www.oracle.com/webfolder/technetwork/tutorials/architecture-diagrams/high-availability-overview/high-availability-reference-architectures.html), Exadata MAA architecture presentation (https://www.oracle.com/a/tech/docs/exadata-maa.pdf) and Exadata Cloud MAA architecture presentation (Oracle Cloud: Maximum Availability Architecture Presentation or https://www.oracle.com/a/tech/docs/cloud-maa-overview.pdf) for further details.

Service level expectations in a non-consolidated environment are quite different compared to a consolidated environment. For example, with a stand-alone database used by a developer or a department, the level of disruption caused by a system down event is limited to a smaller user base. However, when this same database is consolidated with 100 other databases, the impact is much greater if each database supports different departments and user communities. The level of disruption to the enterprise of an outage that impacts the consolidated database is magnified by 100 times, making HA and data protection a higher priority. The principals outlined in this section apply equally to on-premise database deployments on traditional hardware architectures, as well as to Exadata deployed in on-premises, Cloud@Customer, or Exadata Cloud Service environments.

Virtual Machines for Isolation

On systems that use Virtual Machines, Oracle recommends using approximately 2-4 Virtual Machines per physical server at most in order to minimize the additional administrative overhead that comes with each Virtual Machine. Virtual Machines can be used to isolate groups of databases according to environment (isolating DEV from TEST) or isolating databases to meet governance requirements (such as PCI data from non-PCI data). Virtual Machines can also be used to isolate a subset of databases from others to reduce impact of upgrades and other O/S level system maintenance tasks. Real Application Clusters allows user sessions to be relocated to another node of a Virtual Cluster, enabling zero downtime maintenance. RAC therefore eliminates the approach of deploying one database per VM for maintenance isolation, eliminating the administrative overhead associated with large numbers of VMs.

Patching and Upgrades in Consolidated Environments

Patching and upgrades are a particular challenge in a consolidated environment. With multiple databases and business applications relying on a common set of hardware and common software stack, it is critical to provide the proper level of isolation to facilitate upgrades. For example, a single upgrade at the CDB level affects all PDBs within that container. This ability to "manage many as one" significantly reduces administrator time but increases the need for coordination of upgrades. Therefore, databases should be grouped according to SLA and software update cycles to ease coordination. Capabilities including Grid Infrastructure Rolling Patching, Real Application Cluster Rolling Patching, Data Guard Standby First Patching, Data Guard Transient Logical Rolling Upgrade, or Oracle GoldenGate can also be used to reduce downtime to near zero or zero to upgrade an entire CDB in a single operation.

To upgrade individual PDBs within in a CDB, a new CDB can be created with the upgraded version and the PDB(s) can be relocated to the new CDB. This results in downtime during the PDB movement operation but provides the flexibility that may be required to address specific application needs. Although more complex to operate, Oracle GoldenGate also provides the option to upgrade PDBs with minimal or zero downtime by replicating between different CDBs and between databases of different versions.

Application Continuity for Availability

Consolidation of databases involves multiple databases running on each physical server and/or Virtual Machine. Oracle Real Application Clusters (RAC) and Active Data Guard (ADG) provide for high availability for both planned and unplanned outages. Oracle Database also includes the ability to provide continuous application service availability with the following capabilities:

- Transparent Application Failover (TAF)
- Transparent Application Continuity (TAC)

Sessions are drained & rebalanced across nodes of a RAC cluster or Active Data Guard standby database for planned maintenance, while in-flight (active) sessions can be relocated across nodes of the RAC cluster using TAF or TAC. The automatic relocation of sessions across RAC nodes also occurs in the event of physical server or Virtual Machine failures. Databases in a consolidation environment that require high availability should be configured to enable these capabilities. Please refer to the following Continuous Availability white paper for more information on session drain and rebalance, as well as use of TAF and TAC:

https://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/adb-continuousavailability-5169724.pdf

Considerations for Multiple Databases per Server

Consolidation can be accomplished by placement of multiple non-PDB (non-Multitenant) databases on each physical server, Virtual Machine or Cluster of physical machines or virtual machines. These databases will not benefit from the advantages of Oracle Multitenant such as reduction in background processes, reduced shared memory usage, dynamic sharing of system resources, and administrative benefits from managing many-as-one. Multiple non-Multitenant databases can share a single installation of Oracle software, in what is known as a shared Oracle Home ($ORACLE_HOME) configuration. Systems should include provision for at least 2 Oracle Homes to facilitate upgrades.

Multitenant Considerations

Oracle Multitenant uses the Oracle Maximum Availability Architecture (MAA) to addresses the HA and data protection requirements of consolidated environments. In addition to MAA's customary objectives for HA and data protection there are objectives that are specific to a multitenant architecture context:

Manage Many-as-One Simplicity. MAA best practices must easily scale the management of large consolidated environments so that HA and data protection objectives are achieved while realizing maximum cost benefit (capital and operating costs).

Isolation. MAA best practices must prevent problems that impact a single PDB from impacting the availability of other PDBs in the CDB.

On the surface the first two objectives appear contradictory. Isolation for "n" environments often results in "n" different environments that must be individually managed. Oracle MAA can leverage the multitenant architecture so that HA is achieved with minimal compromise in isolation while achieving substantial benefits from being able to manage-as-one.

Oracle Multitenant enables the management of a CDB as a single database regardless of how many PDBs it contains. A CDB with 100 PDBs can achieve 100 to 1 reduction for many maintenance and operational tasks: a single RMAN backup, a single Oracle Active Data Guard standby for disaster recovery, and a single database to upgrade or patch. Oracle MAA also provides the flexibility to manage a PDB in isolation from other PDBs when appropriate. For example:

If an individual PDB must be restored, RMAN can do so without impacting other PDBs that are open in the same CDB. Note that a PDB that has just been plugged in should be backed up immediately after the plugin operation to ensure that it can be recovered should a problem arise before the next regularly scheduled CDB backup.

If fast point-in-time recovery is required, the first release of Oracle Multitenant (in 12cR1) enabled using Flashback Database at the CDB level. Oracle Database 12c Release 2 introduced the ability to use Flashback Database on an individual PDB without impacting the availability of other PDBs.

If an individual PDB experiences a problem that the administrator believes has the potential to impact other PDBs in the CDB, the suspect PDB can be easily unplugged and plugged into another CDB where the issue can be resolved in isolation. An unplug/plug operation also enables the flexibility of applying a patch without impacting other PDBs running in the original CDB. Once the problem is resolved, the PDB can be left in the new CDB, or the PDB can be relocated back to the original CDB after the new patches have been installed.

If one PDB is corrupted on the primary CDB database and there's an existing standby database where the PDB is healthy, a single PDB failover can be executed by unplugging and plugging the PDB from the standby to a new CDB. In Oracle Database 12c Release 2, this functionality is automated through Data Broker command line interface. Refer to My Oracle Support Note: Unplugging a Single Failed PDB from a Standby Database and Plugging into a New Container (Doc ID 2088201.1).

Oracle Resource Manager will prevent a PDB from consuming more than its assigned share of system resources if there is a sudden spike in workload, a bug, or some other event that changes consumption patterns.

Managing Unplanned Outages

Table 1 identifies various unplanned outages that can impact a database in multitenant architecture. It also identifies the Oracle HA solution to address that outage that is available in each of the HA tiers described earlier in this paper.

Table 1: Unplanned outage matrix for multitenant architecture

EVENT	SOLUTIONS FOR BRONZE, SILVER, GOLD & PLATINUM	RECOVERY WINDOWS (RTO)	DATA LOSS (RPO)
Instance failure	BRONZE: Database Restart	Minutes after server restart	Zero
	SILVER, GOLD, PLATINUM: Oracle RAC Application Continuity	Seconds with Oracle RAC Zero with Application Continuity	Zero
Permanent DB node failure	BRONZE: Restore and recover	Hours to Day	Zero
	SILVER, GOLD, PLATINUM: Oracle RAC Application Continuity	Seconds with Oracle RAC and Exadata Fast Node Death Detection. Zero with Application Continuity	Zero
Storage failure	ALL Tiers: Automatic Storage Management on Exadata and with Exadata HIGH redundancy (triple-mirroring)	Zero database downtime	Zero
Physical Data Corruption	BRONZE, SILVER: Basic protection Some corruptions require restore and recover of PDB or entire CDB	Hours if restore and recovery needed	Varies depending on time of last backup.
	GOLD, PLATINUM: Comprehensive corruption protection and Auto Block Repair with Oracle Active Data Guard	Zero with Active Data Guard automatic block repair for physical corruptions. Seconds to minutes if corruption due to lost writes when using Data Guard Fast Start Failover (FSFO) for repair.	Zero with Automatic Block Repair for Physical Corruptions. Zero with SYNC or FAR SYNC Seconds with ASYNC or when encountering lost writes on primary.
Logical Data Corruption	ALL Tiers: Logical failures resolved by flashback drop, flashback table, flashback transaction, flashback query and undo.	Varies based on detection time	Varies based on extent of failure
	BRONZE, SILVER: Some logical block corruptions require restore and recover of PDB or entire CDB.	Hours to day	Varies based on extent of failure
	GOLD, PLATINUM: Standby prevents and detects logical corruptions during redo apply with DB_BLOCK_CHECKING enabled. Once detected, a failover can be	Seconds after manual failover is executed	Seconds

	completed in seconds with minimal downtime.		
Cluster, Database or complete site failure	BRONZE, SILVER: Restore and recover	Hours to Day	Since last backup. For Oracle Cloud, it can be 30 minutes or less since the last archive backup
	GOLD, PLATINUM: Fail over to secondary using Oracle Active Data Guard Fast Start Failover	Seconds to couple mins	Zero to Near Zero
Single PDB Failure	BRONZE, SILVER: Restore and recover PDB	Minutes to hours	Recovery point as of last backup. For Oracle Cloud, it can be 30 minutes or less since the last archive backup
	GOLD, PLATINUM: Fail over to CDB secondary or perform PDB Failover (Note 2088201.1)	Seconds to minutes	Zero for Data Guard and PDB failover
Performance degradation	ALL Tiers: Database Resource Manager to isolate impact	Degraded service on affected database.	Zero

While it is possible to remedy many issues with an individual PDB without impact to other PDBs, there are situations in which isolation is required. For example, you may need to apply a patch to the Oracle Home being used by the CDB. In this scenario it is recommended that you create a new Oracle Home and CDB, then unplug the PDB and plug it into the new CDB. You can then resolve the problem with 100% certainty of zero impact to other PDBs in the original CDB. Once the problem is resolved you can unplug/plug the PDB back into its original CDB or leave it in the new CDB until some future point in time.

Managing Planned Maintenance

From a planned maintenance perspective all of the traditional solutions available to non-CDBs will work in a multitenant architecture environment. Additionally, there are cases where the administrator can decide if the maintenance should be done to just one PDB or all PDBs in the same container. Oracle provides the flexibility to handle either situation. Oracle HA solutions for planned maintenance are provided in table 2 below.

Table 2: PLANNED MAINTENANCE MATRIX FOR MULTITENANT ARCHITECTUERE

EVENT	SOLUTIONS FOR BRONZE, SILVER, GOLD AND PLATINUM	EXPECTED DOWNTIME
Dynamic and Online Resource Provisioning **or** **Online reorganization and redefinition**	ALL Tiers: Online Reorganization and Redefinition of select objects within each PDB Documentation: Dynamic and Online Resource Provisioning and Online Reorganization and Redefinition	Zero
Database and Grid Infrastructure Software Updates	ALL Tiers: Entire CDB can be online patched if relevant SILVER, GOLD, PLATINUM: Entire CDB can leverage Oracle RAC rolling upgrade with Application Continuity	Zero Seconds to minutes

	GOLD, PLATINUM: With Gold and Platinum, apply software changes to the standby first by following standby-first patch recommendations to evaluate the software changes as additive precaution, then apply RAC rolling upgrades with Application Continuity on the primary.	Zero Zero by relocating services Zero application outage
Database Upgrades (e.g. 18c to 19c)	ALL Tiers: PDB can relocate to CDB with targeted software release GOLD, PLATINUM: Entire CDB can leverage Data Guard database rolling upgrade for patchsets and major database releases PLATINUM: Upgrade CDB to target release and then switchover connections to new upgraded CDB/PDB	Estimated seconds to hour with no datafile copy option. PDB meta data still needs to be upgraded after relocated. Seconds to Minutes Zero downtime
Application upgrades	PLATINUM: Edition-Based Redefinition requires developers to design to leverage this feature PLATINUM: PDB can switch over to GoldenGate replica with the targeted application changes Documentation: Online Application Maintenance and Upgrades	Reduce software sprawl and risk. Reduce OPEX and increase overall stability and availability.

Measuring Capacity Utilization Prior to Consolidation

This section addresses the measurement of capacity utilization prior to migrating to a consolidation environment rather than methods for ongoing monitoring of capacity usage. Monitoring implies ongoing operational support, related tools (such as Oracle Enterprise Manager), and implementation of those tools, which is beyond the scope of this document. Proper measurement of system resource utilization is critical to resource planning and configuration of Resource Management, which will be covered in detail later in this document.

It is critical to note that database performance is closely related to system capacity allocated to the database, but these ultimately are separate aspects of system management. Performance can often be improved by allocating more resources to a database, while performance tuning of the database can often reduce resource usage. Therefore, performance and capacity are linked but must be addressed separately. This section addresses the amount of resources (or capacity) assigned to each database and does not address database and application performance tuning.

Oracle Automatic Workload Repository (AWR) provides detailed data for analysis of capacity usage for Oracle databases and the systems that host Oracle databases for databases using the Oracle Diagnostic Pack. While AWR is a valuable tool for this analysis, this section provides a "least common denominator" approach that can be used in virtually any environment.

Measuring capacity utilization begins before each database is migrated to the consolidation environment. It is important to understand the database's average and peak resource consumption at the source while planning migration to a consolidated environment. Capacity utilization should be tracked (monitored) after migration on the target consolidation environment to ensure capacity allocations are adjusted as needed when workloads change. It is always recommended to validate performance and capacity needs in a test environment before moving databases into production.

Capacity usage measures should be gathered at 3 levels:

System Level. The servers and storage should be monitored to see if CPU, memory, and storage utilization (storage space and I/O) are at acceptable levels. Customers planning to consolidate more databases onto a set of servers and storage should also monitor the available capacity or headroom.

Database (or CDB) Level. Databases should be evaluated to determine resource usage patterns. Databases should also be evaluated to determine whether Resource Manager (if used currently) is restricting or throttling them. Waits for CPU or I/O due to Resource Manager constraints may indicate the need to increase resources assigned to the Database or Container Database. Monitoring of database performance is also important to ensure proper resource allocation or if tuning is required.

PDB monitoring (Multitenant only). Each PDB should be monitored in the same way as a database. If actual resource usage surpasses the resources guaranteed by the CDB Resource Plan or individual PDB settings, then performance will likely degrade as more PDBs are added to the CDB. If the anticipated performance degradation is not acceptable, additional PDBs should not be introduced into the CDB.

Key capacity indicators are highlighted in the table below.

Table 3: key CAPACITY indicators

MONITOR /ADMINISTER	GUIDELINE

CPU	**System (Physical Server or Virtual Machine):**
	
	Monitor the system's CPU utilization, using available tools or using the "Host CPU Utilization" metric from v$sysmetric_history. If CPU utilization is near 100%, use OS tools or the "OS Load" metric from v$sysmetric_history to determine the degree of over-subscription on the system. Excessive over-subscription should be avoided, as specified in Table 4, Guidelines for Over-Subscription by Environment. Use Instance Caging to avoid excessive loads.
	
	Database CPU utilization on existing systems is used as a baseline for consolidating databases onto the new systems. Databases moved from legacy hardware will often require less CPU resources on an Exadata system. However, the degree of CPU savings when moving to Exadata will vary depending on the specific workload of each database. Therefore, the existing system CPU utilization is used as a baseline for the initial deployment. The following query can be used on databases with Resource Plans enabled to determine the CPU resource consumption for the database, including the running and waiting sessions:
	
	`select to_char(begin_time, 'HH24:MI') time,`
	`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;`
	
	The average running sessions from this query determines the CPU_COUNT required. However, if Average Waiting Sessions is greater than zero with any regularity and performance is not acceptable, the CPU_COUNT should be raised higher.
	
	Database or CDB:
	
	If Instance Caging, CDB Resource Manager, or per PDB CPU management is enabled, monitor the amount of CPU that the instance actually used and the amount of CPU it needed but was prevented from using. Use v$rsrcmgrmetric_history, as described in MOS note 1338988.1. The sum of 'avg_running_sessions' across all Consumer Groups and PDBs specifies the number of CPUs actually used. The sum of 'avg_waiting_sessions' across all Consumer Groups and PDBs specifies the throttling performed by Resource Manager due to insufficient CPU. This value corresponds to the additional number of CPUs needed.
	
	For CDBs, evaluate the available CPU capacity or headroom to determine if more PDBs can be added. If CDB Resource Manager or per PDB management is enabled, calculate the total amount of CPU needed by summing 'avg_running_sessions' and 'avg_waiting_sessions' from v$rsrcmgrmetric_history across all Consumer Groups and PDBs. If this sum is less than CPU_COUNT for the entire CDB instance, then the gap between this sum and the instance CPU_COUNT is the CPU headroom on the CDB and you can consider adding more PDBs. If this sum is greater than the instance CPU_COUNT, then this CDB is already operating at capacity and has no headroom. Adding more PDBs will only increase contention for CPU. Therefore, additional PDBs should only be added for BRONZE CDBs if the existing PDBs and their applications can tolerate some drop in performance. Monitoring should be done during peak hours. For more details, see MOS note 1338988.1.
	
	PDBs:
	
	Monitor the actual CPU usage and CPU wait time of each PDB. Use v$rsrcmgrmetric_history to see how many CPUs a PDB actually used, using the 'avg_running_sessions' metric. Compare the actual CPU usage with the PDB's guaranteed CPU, which is based off the PDB's shares divided by the total number of shares across all PDBs that are open on the instance. For example, if the PDB has 1 share and the total shares is 5, then the PDB is guaranteed 1/5th of CPU_COUNT. If the PDB's actual CPU usage is above its guaranteed CPU usage, then the PDB's performance may suffer if additional PDBs are added. Use v$rsrcmgrmetric_history to view the PDB's CPU wait time, using the

	'avg_waiting_sessions' metric. If non-zero, the PDB's performance could be improved by increasing its shares or utilization limit in the CDB Resource Plan. For more details, see MOS note 1338988.1.
Memory	**System (Physical Server or Virtual Machine):** No paging should be seen. The vmstat command or other tools can be used to monitor for zero or very low page-in and page-out rates. Memory should never be oversubscribed. On Linux systems, /proc/meminfo can be evaluated to determine the total system memory allocated. Compare the memory allocated at the O/S level to the memory allocated to Oracle databases and client processes. SGA usage can be computed from each database's SGA_TARGET parameter. PGA usage can be determined, using the statistics below. **Database or CDB:** SGA should reside in HugePages to avoid impacting performance of the database or other databases on the system. Setting USE_LARGE_PAGES=ONLY forces SGA into HugePages and prevents databases from starting if HugePages space is not sufficient. Otherwise, SGA size simply needs to be captured for capacity utilization purposes. PGA usage is highly dependent on PGA_AGGREGATE_TARGET and can be monitored via v$pgastat or using monitoring tools. PGA size will be hard-limited to PGA_AGGREGATE_LIMIT (if configured), or the "maximum PGA allocated" statistic will show the maximum PGA that the instance has allocated since its startup. The statistic "total PGA allocated" is the amount of PGA that the instance has allocated currently. Monitoring these values enables the DBA to know how much PGA is actually being used by the instance. They should be compared to PGA_AGGREGATE_TARGET to see if the database is exceeding it and by how much. PGA_AGGREGATE_LIMIT will impose a hard limit on PGA usage by reducing PGA where possible without disruption, as well as terminating sessions using excessive PGA.
I/O	For Exadata storage cells. Monitor Exadata I/O metrics using the script in MOS note 1337265.1. These metrics can also be viewed using Enterprise Manager. Monitor the Exadata FlashCache hit ratio for OLTP I/Os and evaluate increases in FlashCache allocations for the database if hit ratios are unacceptable. If the latency is still not acceptable, then consider changing the IORM objective from "auto" to "balanced" or "low latency". Monitor the actual disk IO utilization across databases using the DB_IO_UTIL_SM + DB_IO_UTIL_LG metrics. This comparison shows which databases are most heavily using the disks. Compare FC_IO_RQ_R with FC_IO_RQ_R_MISS to calculate the OLTP flash hit ratio. If the current performance should remain the same, make sure the flash has headroom by monitoring its flash hit ratio before adding more databases to use flash. For example, if performance is acceptable at an 80% or higher flash hit ratio, performance may deteriorate if more databases are added and the flash hit ratio drops below 80%. **For the database or CDB:** Monitor I/O wait events, using AWR. If the problematic wait event is "db file sequential read", then monitor and tune the database's flash cache hit rate (see above), the storage cell's disk latency (see above), and the database's IORM throttle time for small requests (see below). If the problematic wait event is "cell smart table scan", then monitor and tune the database's throttle time (see below). If IORM is enabled, monitor the average IORM throttle time per request, using the DB_IO_WT_SM_RQ and DB_IO_WT_LG_RQ metrics. If the wait times are large and the database performance is unsatisfactory, either the database's allocation/share or utilization limit in the inter-database plan needs to be increased. If the wait times are large across all databases, then the disks have reached their maximum capacity. In this case, no new databases should be added unless the existing databases can tolerate a drop in performance.

	For CDBs, monitor the actual disk IO utilization across PDBs using the PDB_IO_UTIL_SM + PDB_IO_UTIL_LG metrics. This comparison shows which PDBs within the CDB are most heavily using the disks. **For the PDB:** If IORM is enabled, monitor the average IORM throttle time per request, using the PDB_IO_WT_SM_RQ and PDB_IO_WT_LG_RQ metrics. If the wait times are large and the PDB performance is unsatisfactory, either the PDB's share or utilization limit in the CDB Resource Plan needs to be increased. If the wait times are large across all PDBs, then the allocation and/or limits for the CDB in the inter-database IORM plan need to be increased. Until then, no new PDBs should be added unless the existing PDBs can tolerate a drop in performance.
Database performance indicators	Use Automatic Workload Repository and/or Enterprise Manager for the database load profile, drill down to see top wait events. From here drill down into specific waits. Use ASH analytics to view breakdown of CPU usage by consumer group or PDB
Application performance indicators	Use Automatic Workload Repository and/or Enterprise Manager for the database load profile, drill down to see top wait events. From here drill down into specific waits. Use ASH analytics to view breakdown of CPU usage by consumer group or PDB

Resource Management for Consolidation

Database Consolidation is the placement of multiple databases onto shared computing and storage infrastructure primarily for the purposes of reducing infrastructure and operational costs. However, shared infrastructure typically raises performance concerns among those responsible for the business applications that rely on those shared resources. How do business owners know their applications are receiving the proper amount of system resources required to deliver acceptable performance? The answer is resource management.

Oracle database includes a number of powerful tools for managing resources that allow multiple databases to share a single set of compute infrastructure, while still ensuring the proper amount of resources required to support critical business functions. These capabilities allow IT organizations to properly manage resources without the administrative overhead of deploying multiple physical servers or Virtual Machines. Exadata extends these resource management capabilities into the Exadata storage tier, providing comprehensive management of the complete set of resources that databases rely upon. The resource management approach outlined in this chapter applies to Exadata on-premises, Exadata Cloud Service, and Exadata Cloud@Customer using the same settings discussed in the sections below.

Resource Management Defined

Computing resources in either a Cloud environment or a traditional data center environment represent a significant cost to any organization. Proper management of those computing resources controls cost ensures performance of business applications and prevents performance issues in one business application from impacting another. Oracle's resource management tools include the ability to manage system resources so that each database (and therefore each business application) receives the desired minimum amount of system resources, and each database does not exceed the amount of allocated resources. Oracle allows control over the following resources:

- Compute (CPU)
- Memory
- Processes (Database Sessions & Parallel Query Processes)
- I/O Operations and Bandwidth
- Exadata Storage & Interconnect Network
- Storage Space (Disk and/or Flash)
- Exadata Smart FlashCache

Compute, memory, and process controls are provided by the Oracle Database Resource Manager (DBRM) across all platforms, while I/O and storage related controls are provided by the Exadata I/O Resource Manager (IORM) specifically on Exadata platforms. Oracle also offers I/O rate limit controls for non-Exadata environments, but Exadata storage includes deep integration with the Oracle database, allowing much more robust control over storage and I/O.

Resource Management protects databases from what is known as the "noisy neighbor" problem, where one application (or database) can consume more than the desired amount of resources. For example, runaway SQL statements within one database could impact other databases sharing the same compute and storage resources. Oracle Resource Manager prevents one database (including runaway SQL statements within that database) from impacting other databases on the same system. Resource Manager Consumer Groups can also limit the impact of runaway SQL within an individual database.

Exadata Network Resource Management is automatic and does not require administrator intervention to manage network traffic on the Exadata storage and cluster interconnect network. This feature automatically prioritizes critical network messages over less critical messages to ensure performance of Exadata systems.

Resource Starvation (Performance Issues)

Resource starvation occurs when a database attempts to use more resources than allowed by the resource plan. The most common causes of resource starvation are application coding errors (including SQL coding errors), accidentally dropped or

otherwise missing indexes, and incorrect optimizer statistics. These issues result in what are called "runaway" SQL statements, which can cause the database to consume (or attempt to consume) more than the amount of CPU, I/O, and other system resources allowed by the Resource Plan. The affected database is therefore constrained by Resource Manager, which may impact ALL users of the database, but will not impact other databases on the same system. Resource Manager Consumer Groups can be created to limit the impact of runaway SQL statements within a database.

When resource starvation occurs due to runaway SQL statements or other unexpected heavy workloads, DBRM and IORM are not the root cause of performance issues. Rather, the runaway application SQL statement or workload is the cause, and DBRM and IORM are simply imposing the defined resource limits to prevent the database from impacting performance of other databases on the system.

Resource Manager Improves Availability

The Oracle Resource Managers (DBRM and IORM) ensure databases do not consume more than the provisioned amount of system resources. As outlined in this section, these resource controls govern use of CPU, memory, O/S processes, and I/O. Resource Manager prevents applications and databases from consuming excessive resources, which can result in availability issues and potential downtime of those systems. Proper resource management as outlined in this section can protect against application failures such as connection storms, memory leaks in PL/SQL applications, and runaway application SQL statements.

Hierarchy of Resource Allocations

The resources of a system are subdivided through successive layers in a hierarchy according to business needs of each application. Resources can be allocated and managed down to the level of granularity required to control costs and ensure databases are isolated from each other to prevent "noisy neighbor" issues, where one database, application service, microservice, job task, or user from affecting others on the system. The hierarchy of resource allocation includes the following:

- Physical Servers
- Virtual Machines
- Databases or Container Databases
- Pluggable Databases
- Consumer Groups
- Users and Jobs

Systems running Oracle databases or group of Oracle databases may be comprised of multiple physical servers. Each physical server includes a given amount of resources (CPU, Memory, and effective process limits) that can be sub-divided into multiple Virtual Machines. The resources of a Virtual Machine can be sub-dived and allocated across one or more databases or container databases. In the case of Container Databases, the resources assigned to each can also be sub-divided and allocated to 1 or more Pluggable Databases. Finally, the resources within one database can be allocated to various consumer groups, which are effectively individual users, groups of users, job classes, etc.

Over-Subscription of Resources

When system resources are carefully divided among databases, administrators tend to allocate more than the resources needed by a given database at any specific point in time. Resources are often allocated for PEAK processing periods but are under-utilized during periods of lower demand. For example, retail businesses often allocate resources for peak seasons, tax-related businesses allocate resources based on user needs during tax filing season, and financial accounting systems have resources allocated to meet end-of-month, end-of-quarter, and end-of-fiscal year processing needs.

Over-subscription is a technique for taking advantage of the peaks and valleys of processing requirements that may differ between applications. Over-subscription effectively means allocating MORE resources than are physically present on a system under the assumption that users and databases will not experience simultaneous peak demands for those resources. In this section, we will explore the topic of over-subscription, including guidelines for each resource type, guidelines by workload, and by environment such as development, test, production, and disaster recovery.

Over-Subscription Among Pluggable Databases

Container Databases provide a mechanism to allow over-subscription of ALL resources among the Pluggable Databases within the Container without causing system stability issues. Performance of databases may suffer, but the operating system will not be impacted because the contained databases are not exceeding what can be supported at the system level.

Over-Subscription of Memory

Memory should never be over-subscribed at the VM level, database level, or between Container Database due to the severe performance and stability issues that will result, including node evictions, node failures, and extreme paging/swapping. Memory should be carefully divided among Virtual Machines, then sub-divided in-turn between databases, and container databases. Operating stability or severe performance issues can result due to paging of memory. Pluggable databases within a container can use memory over-subscription without such performance impacts.

Over-Subscription of Processes

Each operating system image (whether running on bare metal or within a Virtual Machine) can only support a certain number of O/S processes based on the compute resources (processor cores) assigned. Oracle recommends a maximum in the range of 64 database sessions for each processor core assigned to the database. The optimal number of sessions for a particular database will vary based on workload. Over-subscription of processes assumes some databases will be idle while others are active, such as multiple test, QA, or development databases residing on a single sever, Virtual Machine or Cluster.

Over-Subscription of Compute and I/O

It is possible to over-subscribe compute and I/O resources assigned at the Virtual Machine, database, or Container Database level without causing system stability issues. However, this represents a trade-off between capacity utilization and potential application performance impact. Operating systems are able to effectively manage the prioritization of critical work without suffering stability issues when compute (processor cores) use a moderate level of over-subscription.

Over-Subscription by Criticality

Over-subscription of resources ensures full use of available resources but does not provide adequate control over resource usage. For example, ALL available CPU within a bare metal server or Virtual Machine can be allocated to 2 databases. If one database experiences high levels of activity while the other database has less activity, the over-provisioning ensures all of the resources can be utilized. However, if BOTH databases experience high levels of activity simultaneously, the databases will be in contention with each other for those resources and one or both databases will suffer performance issues that are difficult to control. In extreme circumstances, the server can become unstable or unresponsive due to over-committed system resources.

For any mission-critical or "tier-1" systems, over-subscription should be used judiciously to avoid system performance issues. As shown in the table below, memory should only be over-subscribed when using Pluggable Databases because the Container guards against over-subscription at the system level. CPU, processes, and I/O may be over-subscribed depending on business needs, especially for less critical environments. Table 4 (below) shows 5 systems with 4 classification levels. Both TEST and QA are considered the same level of criticality in this example.

Table 4: Guidelines for Over-Subscription by Environment

CRITICALITY	ENVIRONMENT	CPU	PROCESSES	MEMORY	I/O
1	Critical Production	Not to exceed 75% of CPU Threads	64 x Cores	NO	Limit 2x I/O Share
2	Non-Critical Production	Equal CPU Threads	64 x Cores	NO	Limit 2x I/O Share

3	Test	2 x CPU Threads	128 x Cores	NO	Limit 2x I/O Share
4	QA	2 x CPU Threads	128 x Cores	50% (PDB only)	Limit 4x I/O Share
5	Development	2 x CPU Threads	128 x Cores	50% (PDB only)	Limit 4x I/O Share

Enabling Database Resource Manager (DBRM)

While it is possible to create detailed resource management plans using DBRM, the vast majority of non-CDB database deployments should simply use the Default Plan. Enabling the default resource management plan is done by executing the following command in each non-CDB database:

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

Container Databases use the DEFAULT_CDB_PLAN as shown in the following command. This setting covers all Pluggable Databases included under the Container Database.

```
SQL> ALTER SYSTEM SET
           RESOURCE_MANAGER_PLAN = 'DEFAULT_CDB_PLAN'
           SID = '*'
           SCOPE=both;
```

The above commands assume an SPFILE is being used for all databases (as indicated by the SCOPE=both clause), which sets the resource manager plan in memory of each instance as well as in the spfile. The SID='*' clause applies the setting to all instances in a non-RAC or RAC environment. Use of SPFILE is considered "best practice" for managing Oracle databases.

Enabling PDB Performance Profiles

The DEFAULT_CDB_PLAN at the container level is sufficient in cases where the "allocation method" is used at the Pluggable Database level. However, taking full advantage of the resource sharing capabilities of Pluggable Databases requires use of PDB Performance Profiles.

```
DECLARE
  v_plan VARCHAR2(30) := 'cdb_profile_plan';
BEGIN
  DBMS_RESOURCE_MANAGER.clear_pending_area;
  DBMS_RESOURCE_MANAGER.create_pending_area;

  DBMS_RESOURCE_MANAGER.create_cdb_profile_directive(
    plan                  => v_plan,
    profile               => 'xxsmall',
    shares                => 1,
    utilization_limit     => 4,
    parallel_server_limit => 4);

  DBMS_RESOURCE_MANAGER.validate_pending_area;
```

```
  DBMS_RESOURCE_MANAGER.submit_pending_area;
END;
/
```

The above example creates the PDB Performance Profile named "xxsmall" and the pluggable database is altered to use this PDB Performance Profile as follows:

```
ALTER SESSION SET CONTAINER=pdb1;

ALTER SYSTEM SET DB_PERFORMANCE_PROFILE=xxsmall SCOPE=BOTH;
```

All performance profiles required by PDBs within a Container Database should be created and made available to the Pluggable Databases within that Container. The Performance Profile can be included in a lockdown profile (in Oracle 12cR2 and high releases) in cases where PDBs are managed by a separate administrator other than the CDB administrator.

Pluggable Database Lockdown Profiles

Large scale deployments of Pluggable Databases often involve two layers of administrators, with a "fleet administrator" who manage the Container Database level, and one or more administrators who separately manage Pluggable Databases within each Container Database. Lockdown Profiles (available in Oracle 12cR2 and later releases) allow the CDB administrator to control resources allocated to Pluggable Databases, preventing PDB administrators from allocating beyond the approved amount of resources for each database.

Enabling Exadata I/O Resource Manager (IORM)

By default, the Exadata I/O Resource Manager (IORM) is enabled for all environments, but uses the "basic" I/O resource plan, which prioritizes I/O based on size and criticality, irrespective of the database generating the I/O. The following command sets the IORM object to "auto", which enables use of IORM Plans & Profiles:

```
dcli -g ~/cell_group
     -l root cellcli
     -e alter iormplan objective = auto
```

This example shows use of the "dcli" command to set the IORM Plan Objective to "auto" across all Exadata storage cells in the system. Alternatively, it is also possible to execute the commands on each cell using the CELLCLI utility.

Exadata IORM Plans and Profiles

The Exadata I/O Resource Manager (IORM) uses PLANS for individual databases, or (with Oracle 12cR2 and later releases) PROFILES assigned to multiple databases with similar resource requirements. IORM allows control of the following resources in an Exadata system:

- Share of I/O relative to other databases
- Limit of I/O available on the system
- Flash Cache Limit (upper limit)
- Flash Cache Size (reservation of space)

The SHARE clause gives each database a MIMIMUM share of I/O resources on the system relative to the total number of databases on that system. For example, if 10 databases are each assigned share=10, each database receives at least 10% of the total I/O resources. Databases can consume MORE than the assigned share of resources if other databases are idle.

While SHARE sets the minimum resources available to a database, the LIMIT clause sets the maximum I/O resources for databases. Assigning LIMIT to each database results in greater consistency of resource allocation, resulting in more

consistent performance. However, use of LIMIT prevents databases from using idle resources, resulting in under-utilized IT assets.

IORM plans for individual databases use a plan name equal to the DB_UNIQUE_NAME of the database. Profiles are DBPLANS that include the type=profile clause to designate the directive as a profile rather than a plan for an individual database as shown in the example below:

The following Exadata dcli command creates or alters an IORM plan for the database named ORCL across all cells in the configuration. IORM Plans are based on the DB_UNIQUE_NAME of each database.

```
dcli -g ~/cell_group
     -l root cellcli
     -e alter iormplan DBPLAN=((name=ORCL, share=10, limit=21))
```

The following Exadata dcli command uses type=profile to create or alter an IORM profile named "myprofile" across all cells in the configuration.

```
dcli -g ~/cell_group
     -l root cellcli
     -e alter iormplan DBPLAN=((name=myprofile, share=10, limit=21,
     type=profile))
```

Every database using the same IORM profile will receive the same SHARE, LIMIT, and FlashCache allocations. It is recommended to assign databases to a small number of profiles (or "shapes") such as SMALL, MEDIUM, and LARGE for manageability. See the section on Resource Manager Shapes for more discussion of this concept.

Setting the Database Performance Profile (Exadata IORM)

In an Exadata environment, databases (including non-PDB and PDB) can use an IORM Performance Profile (in Oracle 12cR2 and above), which eliminates the need to set the IORM Plan for each database individually. All databases using the same profile will receive the same IORM resource allocations set in the IORM Profile (see following section).

```
SQL> ALTER SYSTEM SET
           DB_PERFORMANCE_PROFILE = 'myprofile'
           SID = '*'
           SCOPE=both;
```

The DB_PERFORMANCE_PROFILE setting in the database must simply match the profile name set in the Exadata storage IORM plan as outlined in the previous section.

I/O Resource Management for Non-Exadata Platforms (PDB only)

For Oracle database 12c Release 2 and above using Oracle Multitenant on non-Exadata platforms, the following settings provide basic I/O Resource Management capabilities for Pluggable Databases within a Container Database:

- MAX_IOPS
- MAX_MBPS

While these database parameter settings provide some I/O Resource Management capabilities, it is important to note that non-Exadata storage does not include Exadata Smart FlashCache, and there is no substitute in non-Exadata systems for the IORM settings that govern Flash Cache Size and Flash Cache Limit.

It is important to note that non-Exadata systems do not have the storage network resource management features of Exadata (found even in the "basic" I/O Resource Plan) that ensure time-critical I/O is prioritized above non-critical I/O. Examples of time-critical I/O events includes log file sync, log file parallel write, file header writes, data block writes performed during checkpoints, and data block writes performed to free space in the buffer cache. Criticality of I/O is

related to the object being written as well as activity within the database. This capability depends on integration of hardware with database software that is only possible in Exadata.

O/S Process Controls

To prevent excessive numbers of database processes, database administrators should target an active session count less than or equal to the CPU_COUNT for each database. In addition, administrators should target an upper bound on the total session count for all databases within a physical server or Virtual Machine to 64 times the total cores in the physical server or Virtual Machine (32 times CPU_COUNT in a server with 2 hyper-threads per core). High process counts on any operating system can lead to memory paging, CPU contention, and even node instability. High numbers of database connections do not always result in higher application throughput and can actually increase contention and decrease application performance. The number of sessions should also be decreased when using Parallel Query in order to devote more system resources to parallelism.

In addition to the overall count of processes in a system, the arrival rate of connections in what is known as a "login storm" can also result in system stability issues as the O/S attempts to service those connections By using one or more of the following techniques, the overall process count and connection arrival rates can be reduced to improve performance and stability of the system.

Configure Oracle listeners to throttle incoming connections to avoid logon storms after a database node or instance failure.

Limit the number of processes and connections to the database servers by using connection pools and setting the maximum number of connections to a value slightly above estimated active working sessions.

Avoid the expensive allocation and de-allocation of processes by eliminating dynamic connection pools by setting MIN and MAX connections to be the same.

Oracle listeners can be configured to throttle incoming connections to avoid logon storms after a database node or instance failure. The connection rate limiter feature in the Oracle Net Listener enables a database administrator (DBA) to limit the number of new connections handled by the listener. When this feature is enabled, Oracle Net Listener imposes a user-specified maximum limit on the number of new connections handled by the listener every second.

Depending on the configuration, the rate can be applied to a collection of endpoints, or to a specific endpoint. This feature is controlled through two listener.ora configuration parameters as follows:

CONNECTION_RATE_<listener_name>=number_of_connections_per_second sets the global rate that is enforced across all listening endpoints that are rate-limited. When this parameter is specified, it overrides any endpoint-level numericrate values that might be specified.

RATE_LIMIT indicates that a particular listening endpoint is rate limited. The parameter is specified in the ADDRESS section of the listener endpoint configuration. When the RATE_LIMIT parameter is set to a value greater than 0, the rate limit is enforced at that endpoint level.

Example: Throttle new connections to prevent logon storms.

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

In the preceding example, the connection rates are enforced at the endpoint level. A maximum of 10 connections are processed through port 1521 every second. The connections through port 1522 are limited to 20 every second. Connections through port 1523 are not limited. When the rate is exceeded, a TNS-01158: Internal connection limit reached is logged. Please refer to the Oracle Net Services Reference guide for more information.

Idle Session and Idle Blocking Session Timeouts

Oracle Database Resource Manager (DBRM) can be used to control timeout of idle sessions and idle blocking sessions to ensure overall system stability. The following settings are used to control both idle sessions as well as idle sessions that are blocking other sessions:

- MAX_IDLE_TIME is used to control idle sessions in generally (blocking or not)
- MAX_IDLE_BLOCKER_TIME is used to control timeouts for blocking sessions

The MAX_IDLE_TIME setting can also be controlled using a database parameter or as a Resource Plan Directive. The MAX_IDLE_BLOCKER_TIME can only be set as a Resource Plan Directive. Oracle recommends use of MAX_IDLE_BLOCKER_TIME rather than MAX_IDLE_TIME in cases where applications commonly leave idle sessions in the system, or where applications do not easily tolerate termination of idle sessions.

Database & PDB Resource Controls

Resources for each database are controlled using a core set of controls within each Pluggable Database (PDB) or stand-alone (non-PDB) database. DBRM (and IORM in Exadata) uses these settings to ensure proper management of available system resources to ensure database performance at the desired level as well as to ensure system stability. Table 5 shows the set of core inter-database resource management controls and the enhancements in these controls from release to release from 12cR1 through 19c.

Table 5: Database Resource Controls by Release (12cR1 – 19c)

	12CR1	12CR2	18C	19C
RESOURCE_MANAGER_PLAN	•	•	•	•
CPU_COUNT	•	•	•	•
SGA_TARGET	•	•	•	•
PGA_AGGREGATE_TARGET	•	•	•	•
PARALLEL_MAX_SERVERS	•	•	•	•
SESSIONS	•	•	•	•
PGA_AGGREGATE_LIMIT	•	•	•	•
MAX_IDLE_BLOCKER (Resource Plan Directive)	•	•	•	•
DB_PERFORMANCE_PROFILE		•	•	•
MAX_IDLE_TIME		•	•	•

As seen in the table above and detailed below, new resource management settings were introduced in Oracle 12c Release 1, 12c Release 2, and in Oracle 19c. These settings improve manageability of resources and improve availability issues that might arise due to resource starvation at the database or system level.

The PGA_AGGREGATE_LIMIT setting was introduced in 12cR1 to provide control over the total amount of PGA memory allocated by all sessions in the database. This setting improves overall system availability by limiting excessive memory allocations in the PGA, which can arise from a variety of conditions including overallocation of memory in PL/SQL code.

The DB_PERFORMANCE_PROFILE setting was introduced in 12cR2 to allow use of IORM profiles, which allow a single profile to control resources for many databases. In earlier releases prior 12cR2, IORM plans are set for each individual database according to DB_UNIQUE_NAME of the database.

The MAX_IDLE_TIME setting was introduced in 12cR2 to enable automatic termination of idle databases sessions, which improves database and system availability by terminating idle sessions that are holding system resources. Some applications don't tolerate termination of idle sessions well. Use of MAX_IDLE_BLOCKER_TIME might be preferable in those circumstances.

The MAX_IDLE_BLOCKER_TIME setting was introduced in Oracle10g to enable automatic termination of idle databases sessions that are blocking other sessions, which improves database and system availability by terminating idle sessions that are also blocking other sessions.

The CPU_COUNT is an instance-level setting that specifies the number of threads (on hyperthreaded systems) that can be used by the instance. Each Real Application Clusters (RAC) database includes one or more instances across nodes of a cluster. It is important to note that the other parameters discussed in this section (with the exception of DB_PERFORMANCE_PROFILE) are also instance-level settings.

The memory and O/S process controls above should be used even in non-consolidated environments to prevent overallocation of system resources, which can lead to system availability issues. Overallocation of PGA memory (such as in a PL/SQL application) can lead to memory starvation. Applications that cause connection storms over-allocate O/S processes and lead to starvation of resources at the O/S level.

For Pluggable Databases, these settings can also be configured in a "lockdown profile" at the CDB level, preventing individual PDB database administrators from modifying these settings. This capability is important in large scale consolidation environments with multiple database administrators.

Intra-Database Resource Management - Consumer Groups

While this document deals primarily with provisioning and management of resources ACROSS databases, the Oracle Database Resource Manager also provides the ability to manage resources within a single database. Resource Manager Consumer Groups can be used to assign system resources to users, sessions, or connections to those consumer groups using various rules and policies and manage use of those resources. For more information on intra-database Resource Management, please refer to MOS Note# 1484302.1 (Master Note: Overview of Oracle Resource Manager and DBMS_RESOURCE_MANAGER, Doc ID 1484302.1).

Runaway Queries

Resource Manager Consumer Groups provide the ability to manage sessions that consume more than the allowed amount of resources, such as sessions executing "runaway" queries. This capability is important in any environment that allows ad-hoc queries, such a Data Warehouse and Analytic systems. Resource Manager allows the following action on any session or query consuming excessive resources:

- Switch Group
- Cancel SQL
- Kill Session
- Log Only

Resource Manager Plan Directives are used to switch a session to a different (usually lower priority) consumer group, cancel the specific SQL statement, kill (cancel) the user session, or simply LOG the event for later analysis by administrators. While addressing runaway queries is an important topic, it is beyond the scope of this document. Please refer to MOS Note# 1484302.1 (Master Note: Overview of Oracle Resource Manager and DBMS_RESOURCE_MANAGER, Doc ID 1484302.1) for more information.

Using Resource Controls

The following table provides a summary of system resources and the controls for those resources in Oracle Database Resource Manager (DBRM) and the I/O Resource management capabilities of Exadata, as well as settings for non-Exadata systems.

Table 6: Controlling System Resources (12c Release 2 and later)

Resource	DB (Stand-alone)	CDB	PDB
CPU	**Instance Caging Method** CPU_COUNT (with a minimum of 2) RESOURCE_MANAGER_PLAN = 'DEFAULT_PLAN'	RESOURCE_MANAGER_PLAN = 'DEFAULT_CDB_PLAN'	**Control via Parameters:** CPU_COUNT **Control via CDB Resource Plan:** SHARE = share of CPU resources per PDB UTILIZATION_LIMIT = Limit of resources for a PDB
Memory	SGA_TARGET PGA_AGGREGATE_TARGET PGA_AGGREGATE_LIMIT (defaults to 2X target)	**CDB Level:** Applies to entire container. SGA_TARGET PGA_AGGREGATE_TARGET PGA_AGGREGATE_LIMIT (defaults to 2X target)	**Control via Parameters:** SGA_TARGET PGA_AGGREGATE_TARGET PGA_AGGREGATE_LIMIT (defaults to 2X target) **Control via CDB Resource Plan:** SHARE = share of CPU resources per PDB UTILIZATION_LIMIT = Limit of resources for a PDB
Processes and Sessions	**Database Settings:** SESSIONS MAX_IDLE_TIME MAX_IDLE_BLOCKER_TIME PARALLEL_MAX_SERVERS MAX_IDLE_BLOCKER_TIME PARALLEL_TARGET_SERVERS **SQL Net Controls:** CONNECTION_RATE_listener_name RATE_LIMIT	**CDB Settings** SESSIONS MAX_IDLE_TIME MAX_IDLE_BLOCKER_TIME PARALLEL_MAX_SERVERS MAX_IDLE_BLOCKER_TIME PARALLEL_TARGET_SERVERS **SQL Net Controls:** CONNECTION_RATE_listener_name RATE_LIMIT	**Control via Parameters:** SESSIONS MAX_IDLE_TIME MAX_IDLE_BLOCKER_TIME PARALLEL_MAX_SERVERS MAX_IDLE_BLOCKER_TIME PARALLEL_TARGET_SERVERS **Control via CDB Resource Plan:** SHARE = share of CPU resources per PDB UTILIZATION_LIMIT = Limit of resources for a PDB
I/O Resource Management	**Exadata** IORM Objective = 'auto'		

	IORM Share – Specifies a share of I/O resources relative to other databases on the system. IORM Limit – Places an upper bound (in %) on I/O used by the database. FlashCache Min – Minimum amount of FlashCache for the specified database. FlashCache Limit – Can use up to this size depending on other activity in the system. FlashCache Size – Reserves this space for the database. **Non-Exadata Systems** 12cR2 and above – PDB Only (NOT on Exadata) » MAX_IOPS » MAX_MBPS
Network	**Exadata** Network resource management is automatic – no settings required **All Platforms (including Exadata)** Activate RAC databases on a minimum number of instances necessary for the workload

Use of parameters to control PDB compute resources (CPU_COUNT) gives greater control over those resources compared to use of shares and limits in a CDB Resource Plan.

Standardized DB Resource Shapes

The use of standardized resource shapes greatly simplifies resource management. Standardized shapes use a common definition for the relationship among resources. All resources in a system can be allocated as simple functions of the CPU count used for each database running on that system. For example, if a database is allocated 10% of the CPU, it should also receive 10% of the memory, 10% of the I/O, and 10% of the available FlashCache in Exadata systems.

Shapes

Oracle recommends establishing 2 types of resource shapes that contain the full range of resource allocations by workload type. The same overall ratios will be applied to each, but with a number of variations needed to meet the resource demands of each. The recommended types of resource shapes are:

- Data Warehouse (DW)
- Online Transaction Processing (OLTP)

Data Warehouse workloads typically require larger PGA (Process Global Area) and smaller SGA (System Global Area), whereas OLTP workloads require larger SGA and smaller PGA. Likewise, Data Warehouse workloads tend to have fewer concurrent user sessions, with heavier use of Parallel Query processing. OLTP workloads are normally “mixed” workloads with primarily non-parallel operations, and a smaller percentage of Parallel Query work.

While the overall size of the DW and OLTP shapes are nearly identical in terms of CPU, Memory, Process, and I/O allocations, the distribution of memory and processes differs as shown in the tables below.

Multiples

The standardized shapes discussed above establishes common ratios between all of the system resources. These shapes are then scaled using multiples to allocate resources to databases. Using standardized shapes and multiples in this manner makes simplifies system resource allocations, while still allowing administrators to provision the proper amount of resources for each application.

Database Resource Shape Examples

The standardized database resource shapes outlined in this section will be used later during the bin-packing process to determine the size of the "bins" (Server Resource Shapes and Container Databases) as well as arranging databases into those bins. The CPU_COUNT required for each database should be used to establish the proportion of all other system resources allocated to each database. Note that these are DATABASE resource shapes as opposed to Virtual Machine or Bare Metal shapes. Database resource shapes are layered on top of underlying Virtual Machine or Bare Metal shapes, and one Virtual Machine or Bare Metal Exadata server may include more than 1 database.

Tables 7 and 8 show standardized database resource shapes for OLTP and DW workloads using the following Exadata configuration:

- Exadata Cloud@Customer X9M-2
- 8 Database Servers
- 62 OCPUs per Database Server (124 vCPU per Database Server)
- 2 Virtual Machines per Server, 31 OCPUs per VM (62 vCPU per Virtual Machine)
- 12 Exadata Storage Servers (25.6 TB FlashCache per Storage Server)

These Database Resource Shapes apply to stand-alone databases as well as at the Container Database level. While these same database resource shapes can be used with Pluggable Databases (PDBs), it is also possible to use SHARE and LIMIT settings for greater flexibility and greater resource sharing with PDBs. Please refer to the section on PDB Resource Shapes for more information. Each database or container database should be configured to run on a minimum of 2 database nodes using Oracle Real Application Clusters (RAC) to provide high availability. It is important to note that memory and process settings are per-node settings, whereas IORM applies to databases regardless of the number of nodes.

Table 7: OLTP DB Resource Shapes – Exadata Cloud@Customer X9M-2

				Memory				Processes		IORM		
Shape	CPU_COUNT	Nodes	Total	DB Memory	SGA	PGA Target	PGA Limit	Sessions	PQ Processes	Share	Limit %	FlashCache Limit
OLTP Small 2V	4	VM 2	8	30 GB	15 GB	7.5 GB	15 GB	128	4	8	5%	3.7 TB
OLTP Medium 2V	8	VM 2	16	60 GB	30 GB	22 GB	30 GB	256	8	16	5%	7.4 TB
OLTP Large 2V	16	VM 2	32	120 GB	60 GB	30 GB	60 GB	512	16	32	5%	14.9 TB
OLTP 2X Large 2V	32	VM 2	64	240 GB	120 GB	60 GB	120 GB	1024	32	64	8%	29.8 TB
OLTP 3X Large 2V	48	VM 2	96	344 GB	172 GB	86 GB	172 GB	1536	48	96	12%	44.8 TB
OLTP 3X Large 4V	48	VM 4	192	344 GB	172 GB	86 GB	172 GB	1536	48	192	25%	89.6 TB
OLTP 6X Large 2V	96	VM 2	192	694 GB	347 GB	172 GB	347 GB	3072	96	192	25%	89.6 TB
OLTP 12X Large 4V	96	VM 4	384	694 GB	347 GB	172 GB	347 GB	3072	96	384	50%	179.2 TB
OLTP 15X Large 4V	124	VM 4	492	1390 GB	695 GB	347 GB	695 GB	3072	96	384	50%	179.2 TB
OLTP 24X Large 8V	124	VM 8	992	1390 GB	695 GB	347 GB	695 GB	3072	96	768	100%	358.4 TB

As outlined in the previous section, Data Warehouse (DW) workloads typically use a smaller SGA than OLTP systems, with more memory allocated to the PGA to optimize for aggregations. Additionally, DW workloads typically involve fewer users and heavier use of parallelism. Consequently, DW systems use lower session limits, with more PQ processes

than OLTP systems. In all of these example shapes, RAC is used with a minimum of 2 instances (or nodes) per RAC cluster for the purposes of scaling and high availability.

Table 8: DW DB Resource Shapes – Exadata Cloud@Customer X9M-2

				Memory				Processes		PDB & IORM		
Shape	CPU_COUNT	Nodes	Total	DB Memory	SGA	PGA Target	PGA Limit	Sessions	PQ Processes	Share	Limit %	FlashCache Limit
DW Small 2V	4	VM 2	8	30 GB	10 GB	10 GB	20 GB	32	16	8	5%	3.7 TB
DW Medium 2V	8	VM 2	16	60 GB	20 GB	20 GB	40 GB	64	32	16	5%	7.4 TB
DW Large 2V	16	VM 2	32	120 GB	40 GB	40 GB	80 GB	128	64	32	5%	14.9 TB
DW 2X Large 2V	32	VM 2	64	240 GB	80 GB	80 GB	160 GB	256	128	64	8%	29.8 TB
DW 3X Large 2V	48	VM 2	96	350 GB	120 GB	120 GB	240 GB	384	192	96	12%	44.8 TB
DW 3X Large 4V	48	VM 4	192	350 GB	120 GB	120 GB	230 GB	384	192	96	25%	89.6 TB
DW 6X Large 2V	96	VM 2	192	690 GB	230 GB	230 GB	460 GB	768	384	192	25%	89.6 TB
DW 12X Large 4V	96	VM 4	384	690 GB	230 GB	230 GB	460 GB	768	384	384	50%	179.2 TB
DW 15X Large 4V	124	VM4	496	1390 GB	460 GB	460 GB	920 GB	768	384	576	75%	268.8 TB
DW 24X Large 8V	124	VM 8	992	1390 GB	460 GB	460 GB	920 GB	768	384	768	100%	358.4 TB

Use of extremely low values for UTILIZATION_LIMIT (below 5%) should be avoided to prevent thrashing on the system. The above examples show the lowest utilization limits at 5%, with higher values being equal to 2X the share value.

Resource Sharing Between Pluggable Databases

Using Oracle Multitenant, resources within a Container Database can be shared across Pluggable Databases residing within the Container. Sharing of compute resources within a CDB is controlled by the following settings of the resource plan for each PDB rather than using the strict CPU_COUNT control (also known as Instance Caging) outlined in the previous section:

- Share
- Utilization Limit
- Parallel Max Servers
- Parallel Servers Target

The SHARE of compute resources for a database is relative to other databases within the container. For example, if there are 10 Pluggable Databases within a container, where each has a SHARE value equal to 10, each PDB receives at least 10% of the CPU resources of the Container. While SHARE provides a minimum allocation of available resources, Pluggable Databases within a Container can use MORE CPU resources if other databases are not active and consuming those resources.

The UTILIZATION_LIMIT for a database places an upper bound on the amount of compute resources available to a PDB. The primary purpose of UTILIZATION_LIMIT is to prevent or limit the variability in resources available to a database, which can lead to variability in end-user performance. Utilization limits will prevent use of idle system resources, which typically reduces overall system utilization and does not allow users to take advantage of those resources but will provide a more consistent end-user experience.

The PARALLEL_MAX_SERVERS setting places a limit on the amount of parallel server processes that can be utilized by a Pluggable Database. The PARALLEL_SERVERS_TARGET setting establishes a target for shrinking the number of parallel servers to when demand is low.

Pluggable Database Resource Shapes

As noted previously Pluggable Databases can use the allocation method outlined above for stand-alone databases and Container Databases, or Pluggable Databases can use PDB Performance Profiles as detailed in the table below. Using either method, Pluggable Databases receive a portion of the resources allocated to the Container Databases. Hard allocations of CPU, memory, and processes results in under-utilization of system resources overall because the PEAK amount of resource must be allocated to each database. Using the Shares & Limits method, the resources are shared between PDBs, with limits placed on the resource consumption of each Pluggable Database.

Table 9: PDB Performance Profiles – Exadata X9M-2

		Compute & Memory & Process Shares			IORM Shares		
Profile Name	**Nodes**	**Shares**	**Utilization Limit**	**Parallel Server Limit**	**Share**	**Limit %**	**FlashCache Limit**
PDB XX Small 2	VM 2	1	5%	4%	1	5%	9 TB
PDB Extra Small 2	VM 2	2	8%	8%	2	5%	1.8 TB
PDB Small 2	VM 2	4	16%	16%	4	5%	3.7 TB
PDB Medium 2	VM 2	8	32%	33%	8	5%	7.4 TB
PDB Large 2	VM 2	16	64%	66%	16	5%	14.9 TB
PDB 2X Large 2	VM 2	32	100%	100%	32	8%	29.8 TB
PDB 3X Large 2	VM 2	48	100%	100%	48	16%	44.8 TB
PDB 3X Large 4	VM 4	48	100%	100%	96	32%	89.6 TB
PDB 6X Large 2	BM 2	96	100%	100%	96	32%	89.6 TB
PDB 12X Large 4	BM 4	96	100%	100%	192	64%	179.2 TB
PDB 24X Large 8	BM 8	96	100%	100%	384	100.00%	358.4 TB

In the example shown in Table 9, each PDB resource shape is given a "share" of resources, as well as utilization limits equal to DOUBLE the amount of the share in what is known as a PDB Performance Profile. For example, in the "PDB XX Small" Performance Profile, a share of "1" is approximately 2% of the 48 cores on the configured Virtual Machine. LIMIT should not be set lower than 5% to prevent thrashing of system resources. Using the "Shares & Limits" method for PDB's is identical for OLTP and DW workloads because the appropriate resource allocations are assigned at the Container Database level. PDBs are simply taking a share (and limit) of the resources assigned to the container.

Notice that IORM shares and limits for PDBs follow the same settings for stand-alone databases, since Exadata I/O Resource Manager is independent of the compute tier.

Implementing Database Consolidation

This chapter outlines the process for implementing database consolidation using the techniques outlined in this document. Consolidating databases follows the same process when consolidating onto Exadata, Exadata Cloud@Customer, or Exadata Cloud Service. Implementing database consolidation should follow a standardized process that determines resource capacity needs for each database and arranges databases into groups for simplified management. The following steps outline the process:

1. Inventory Databases to be Consolidated
2. Gather Resource Utilization & Growth Metrics
3. Map Resource Requirements to New Platform
4. Determine Database Isolation Requirements & Methods
5. Select Consolidation Method for Databases
6. Group Databases into HA Tiers
7. Perform Bin-Packing of Databases into Resource Shapes
8. Create Resource Plans for Databases

Inventory of Databases to be Consolidated

Database consolidation begins with an inventory of all databases to be consolidated. The following information should be collected for databases that are candidates for consolidation:

- Database Name
- Database Version
- Physical location, region, or data center
- Server Name
- Server Platform O/S
- Server O/S Version
- Associated Application(s)
- High Availability Tier (Bronze, Silver, Gold, Platinum)
- Environment (DEV, TEST, QA, Production, DR, etc.)

This list of candidate databases will be used throughout the consolidation planning process to determine which database will be consolidated together on the same physical and/or virtual servers.

Gather Resource Utilization & Growth Metrics

Resource utilization metrics are required for each database in the list of consolidation candidates. These metrics can be collected from the Oracle Automatic Workload Repository (AWR) for each database:

- CPU Utilization
- Processes & Sessions
- Memory
- Storage Space
- I/O

Metrics collected on existing systems are the same metrics collected for capacity monitoring and planning purposes. Please refer to the section of this document on measuring capacity utilization for more information on how to collect these metrics for planning migration to a database consolidation environment. Current system resource utilization analysis shows the current state as well as some amount of historical resource utilization, but growth can be more complex.

All organizations experience growth in data volumes and processing workloads. This growth results in higher demand for storage space, computing power, system memory, and I/O rates. Historical growth rates can often be used for forward planning, but certainly do not reflect anticipated changes in business conditions, changes in application functionality, or other external forces that impact capacity. For these reasons, it is often necessary to look beyond historical system statistics during the capacity planning process.

The planning horizon refers to the period for which capacity planning is being conducted. Most organizations perform capacity planning on a 3-5 year basis, and purchase hardware or acquire needed services for that period of time. For example, capacity planning might show a 10% compounded annual growth over 5 years, or 61% growth in total over that period. Year to year growth is not always consistent due to changes in business climate, corporate acquisitions, application functional changes, regulatory changes, and other factors. However, historical growth rates in prior years can often be used to forecast future growth rates.

Systems that are hitting resource limits do not provide an accurate view of resource utilization. Anytime demand is higher than supply, the true level of demand is unknown and will be higher than the supply of resources. Customers should be aware of this phenomenon and take this into account during capacity planning.

Map Resource Utilization to New Platform Characteristics

Resource utilization is gathered on an existing system but will typically be applied to a separate system during the consolidation process. Newer computing platforms will normally have faster processors, more memory, faster I/O, greater I/O throughput, and additional increased capacity and performance compared to older systems. Each system vendor uses their own methodology for determining system capacity requirements for a given set of applications or databases. Oracle uses system performance metrics known as M-Values to track performance differences across multiple generations of servers.

Consolidation onto a new system might also be a different architecture than older systems, such as when moving from traditional server & storage environments to Exadata Database Machine. Platform architectural differences often have an impact on system capacity requirements for databases. For example, Exadata typically requires less relative compute power compared to traditional systems due to offloading of database processing into the Exadata storage tier. This is referred to as the Exadata Database Server Optimization and varies according to the workload and degree of storage offload for that workload. The benchmark shown earlier in this documented showed a 2X greater consolidation density, or 50% lower compute resource consumption.

Pent-up-Demand is a phenomenon that should always be taken into account when mapping resource requirements to a new platform. This phenomenon is especially prevalent in Data Warehouse or Analytic systems, where use of those systems is often limited by performance of those systems. Improved performance of these systems often leads to increased use of those systems, resulting in unanticipated growth in resource requirements. This is especially true when moving Data Warehouse and Analytic databases to the Exadata platform.

Determine Isolation Requirements & Method

All IT organizations have requirements to provide some degree isolation of databases from others. The most common isolation requirements are:

- Environment (DEV, TEST, QA, PROD, DR, etc.)
- Data Governance (PII, PCI, HIPAA, etc.)
- Upgrade & Patching

Non-production environments (DEV, TEST, QA, etc.) typically must be separated from production environments, and data that is subjected to special governance regulations such as Payment Card Industry (PCI) requirements typically must

be separated from data that is not subjected to those regulations. Two isolation methods are available for Oracle databases on Exadata systems:

- Physical Isolation
- Virtual Machine Isolation
- Isolation by Container Database

Although physically separate machines provide complete isolation between environments, Virtual Machines provide many of the same benefits within the same physical machine. Non-Production databases (DEV, TEST, QA, etc.) should be isolated from production databases (PROD and DR) using physically separate machines. Isolation for data governance purposes can also be achieved with either physical or virtual machine isolation.

Use of multiple Container databases within a physical server, Virtual Machine, or physical/virtual cluster provides a further level of database isolation. Multiple Container Database may reside on the same O/S, but can use different Oracle DBMS patches or version levels.

Select Consolidation Method for Databases

The reference architecture for each HA tier supports all consolidation methods and deployment models. However, detailed best practices for the migration of standalone databases to a consolidated environment are influenced by the consolidation method used.

Multiple Databases per Server using Oracle RAC combined with Oracle Resource Manager is the MAA best practice for Oracle Database 12c and later releases. Oracle RAC provides availability and scalability, while Resource Manager provides isolation of resources.

Oracle Multitenant is applicable for Oracle Database 12cR1 and later releases. The multitenant architecture delivers the highest possible consolidation density and improves manageability of large numbers of databases.

MAA best practices are equally relevant to physical or virtual environments. Schema consolidation has largely been replaced by Oracle Multitenant but is still an option if customers choose to combine multiple applications or microservices into a single database. The resulting database is then consolidated with others using one of the methods above.

When stand-alone databases are migrated to Oracle Multitenant, the process is as simple as converting each database into a PDB and moving that database to an appropriate CDB for the given HA tier.

Group Databases into HA Tiers

All databases should be classified according to one of the HA Tiers outlined earlier in this document, including business tolerance for data loss (recovery point objective, or RPO) and downtime (recovery time objective, or RTO) for each database that is a candidate for consolidation. Any dependencies that might exist between databases should also be well understood to ensure those databases are properly grouped together to meet business objectives.

The MAA best practice is to consolidate databases that have similar RTO (Recovery Time Objective) and RPO (Recovery Point Objective) according to the standard set of HA tiers described in the preceding section. Note that in cases where there are dependencies between databases, each database is assigned to the HA tier appropriate for the database having the most stringent HA requirement.

The process of grouping databases into standard HA tiers according to RTO and RPO requirements accomplishes three objectives:

Standardization on a limited set of HA tiers reduces complexity and achieves economies of scale.

Efficient consolidation by avoiding over-investment or unnecessary complexity in HA infrastructure and processes. For example, it would be inefficient to consolidate a Bronze Tier database (where RTO and RPO that can be achieved by restoring a backup) with Silver or Gold Tier databases where RTO and RPO requirements dictate additional HA infrastructure.

Establishing the HA and data protection component of the Service Catalog for DBaaS. The service catalog describes the services that an IT organization provides its user community - developers, architects, and end-users - with specifics on how database services are delivered and managed.

Perform Bin-Packing of Databases into Resource Shapes

Once the isolation requirements, consolidation method, and HA Tier groupings (outlined above) have been completed, databases can then be arranged into Resource Shapes in a process known as bin-packing. The process of bin-packing involves determining which databases should be consolidated together into what is known as a Resource Shape (or "bin" in this context), which can be one of the following configurations:

- Physical Server
- Virtual Machine
- Physical Server Cluster
- Virtual Machine Cluster
- Container Databases

The bin-packing process determines the size requirement and configuration of each Resource Shape (or "bin"), including the size of Virtual Machines and number of physical or Virtual Machines combined into a cluster. The bin-packing process should always be performed in order from largest to smallest, where size is measured by the COMPUTE resources assigned to each database, not by storage volume. The volume of storage space required for a Resource Shape is determined AFTER the bin-packing process is completed based on the size of the associated databases. The required amount of storage is then allocated to each Resource Shape (physical server, Virtual Machine, or physical/virtual cluster) according to the total size of databases assigned to that Resource Shape.

Any databases to be consolidated as Pluggable Databases (PDB) within a Container Database (CDB) should all belong to the same HA Tier (Bronze, Silver, Gold, Platinum).

The underlying hardware may have a capacity limit that prevents all PDBs assigned to a particular HA tier from being consolidated into a single server or cluster of servers. Performance requirements must be assessed to determine the suitability of a database as a candidate for database consolidation, and to match existing managed servers that you want to consolidate with the servers that you will consolidate to.

It is often necessary to support more than one Oracle Database release/patch set at a time. For example, one CDB may be at Oracle Database 12c Release 1 (12.1.0.2) and a second CDB at Oracle Database 18.1.0.0. This enables an individual PDB to be 'unplugged' from an Oracle 12.1.0.2 CDB and 'plugged' into the Oracle 18.1.0.0 CDB to implement an upgrade should other 12.1 PDBs be unable to upgrade. This can happen, for example, due to a delay in a vendor-supplied application supporting the new release.

Bronze Tier of databases typically includes the largest number of databases eligible for consolidation and are the most likely candidates for use of Oracle Multitenant. Oracle recommends starting any consolidation effort with Bronze Tier databases due to lower criticality of those applications and higher return on investment.

Container Databases represent a "bin" as well, with databases packed into the Container. Resources for a Container Database are sized to contain all of the Pluggable Databases the CDB will include. The CDB provides isolation from a version/patching perspective, but also serves to constrain resources used by the PDBs under the CDB.

Resource Manager Planning for Databases

The Oracle Database Resource Manager (DBRM) and I/O Resource Manager (IORM) on Exadata serve to ensure databases and the applications they service receive the proper amount of system resources to meet business needs.

- Individual Database Resource Plans
- Resource Profiles for Groups of Databases

Individual Resource Manager PLANS are created for any larger, more critical databases that require closer attention and more active management. Resource Manager PROFILES are used for smaller, less critical databases that often have more uniform resource requirements. Please refer to the Resource Manager section of this document for more information on use of the Oracle Resource Managers (DBRM and IORM).

An overview of the consolidation planning process is described in Table 10.

Table 10: Consolidation Planning

Key Steps for Consolidation	Recommendations	Benefits
1. Inventory databases to be consolidated	The list of databases to be consolidated includes location, platform, application, availability tier, and other characteristics that guide consolidation decisions.	Determines the overall scope of the consolidation effort.
2. Gather resource utilization & growth	Current system resource utilization (CPU, memory, I/O, etc.) provides a basis for resource planning on the consolidated database platform. Growth also must be taken into account to ensure the system meets resources needs of the business. Resource utilization also determines Resource Manager configuration.	Needed for bin-packing exercise in step #7. Resource Management Plans are defined for each database and/or groups of databases.
3. Map Resource Requirements to New Platform	Consult system vendors to determine the relative performance differences between older and newer generations of hardware.	New computing platforms typically deliver improved performance and higher capacity than older systems. Changes in system architecture such as moving to Exadata can also affect system capacity requirements.
4. Determine Database Isolation Requirements & Methods	Isolation can be done by placement on separate physical machines or into separate Virtual Machines. Development and Test databases are normally isolated from Production databases. Databases subjected to special governance rules (PII, PCI, HIPAA, etc.) are normally isolated from other databases.	Meet business requirements for isolation, while minimizing complexity related to ongoing management of large numbers of physical servers or Virtual Machines.
5. Select Consolidation Method for Databases	Oracle Multitenant Container Databases (CDB) deliver the highest consolidation density possible. Multiple databases can also be consolidated onto Resource Shapes (single servers or VMs, or clusters of physical or virtual servers) without Oracle Multitenant. Consolidating onto Exadata provides a higher degree of consolidation density in conjunction with either approach. A combination of Multitenant and stand-alone databases can also be used.	Higher consolidation density reduces system infrastructure costs as well as operational costs for maintaining systems and databases.
6. Group Databases into HA Tiers	Follow MAA HA Tiers (PLATINUM, GOLD, SILVER, BRONZE). All databases within a CDB should be of the same tier.	Reduce operational expense and lowers risk through standardization.
7. Perform Bin-Packing of Databases into Resource Shapes	Create appropriately sized Resource Shapes or "bins". Half Racks up to one full rack (Cloud and Cloud at Customer), or two full multi-rack systems (on-premise only) for a typical resource shape. Assign each resource shape to an HA Tier. Since each tier has a different HA architecture, there is only one HA tier per resource shape. Pack databases into "bins" starting from largest to smallest.	Bin-packing exercise determines Resource Shape configurations.
8. Resource Manager Plans for Databases	Individual Resource Manager plans are created for each of the largest or most critical databases. Resource Manager profiles are used for smaller and less critical databases.	Ensures that the proper amount of system resources are provided to each database according to business needs. Resource Management also serves to ensure system stability is not jeopardized by faulty applications or databases.

Conclusion

Database consolidation is a useful approach to meeting the goals of the modern enterprise, and is designed to balance 5 key business goals including:

- Reducing **costs**
- **Simplifying** administration
- Improving **security**
- Ensuring database and application **availability**
- Delivering the levels of **performance** required for business applications

This document outlined the strategy for meeting these goals using Oracle Database, Virtual Machines, Oracle Multitenant, and Exadata, whether it is deployed in on-premises, Cloud Service or in the Cloud@Customer model. In all of these deployment models, Exadata provides a higher level of consolidation density than other platforms, which reduces cost of the overall platform. Administration is simplified through consolidation of databases, but also through use of the Exadata platform, which is delivered as an integrated solution of servers, storage, networking, operating system, virtual machines, clusterware, logical volume manager, and the Oracle database software.

Oracle Multitenant provides an even higher level of consolidation density (in addition to the high density provided by Exadata), reducing costs further, while providing simplified "manage many-as-one" administration to large scale consolidation environments.

Oracle's Maximum Availability Architecture (MAA) was outlined in this document, including the application of MAA in the context of database consolidation. Providing the required level of database and application availability is a key goal for consolidation environments, and this document outlined how to address those business needs.

Security is improved through database consolidation by the reduction in complexity of multiple systems and often divergent platforms with varying management practices. Security is enhanced even further in deployments on the Exadata platform, which is delivered in a "secure by default" configuration, includes pre-scanned system stack, and the Advanced Intrusion Detection Environment (AIDE) implementation to ensure against attack.

Finally, any consolidation effort must deliver the proper level of performance required by business applications. Systems need to be architected to achieve consolidation without sacrificing performance, and Exadata has demonstrated performance capabilities that can be leveraged to provide consolidation as well as the performance necessary.

References

My Oracle Support Documents

DocID	Document Title
1338988.1	Scripts and Tips for Monitoring CPU Resource Manager
1337265.1	Tool for Gathering I/O Resource Manager Metrics: metric_iorm.pl
888828.1	Exadata Database Machine and Exadata Storage Server Supported Versions
361468.1	HugePages on Oracle Linux 64-bit
401749.1	Oracle Linux: Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration
1484302.1	Master Note: Overview of Oracle Resource Manager and DBMS_RESOURCE_MANAGER
1585184.1	Using PROCESSOR_GROUP_NAME to bind a database instance to CPUs or NUMA nodes on Linux
1338988.1	Scripts and Tips for Monitoring CPU Resource Manager
1070954.1	Oracle Exadata Database Machine exachk or HealthCheck

White Papers & Presentations

Exadata Security Guide

https://docs.oracle.com/en/engineered-systems/exadata-database-machine/dbmsq/security-guide-exadata-database-machine.pdf

MAA best practices

https://www.oracle.com/technetwork/database/availability/maa-overview-onpremise-2019-5391126.pdf

Virtual Machines with Exadata

https://www.oracle.com/technetwork/database/availability/exadata-ovm-2795225.pdf

Continuous Availability White Paper

https://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/adb-continuousavailability-5169724.pdf

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.

Outside North America, find your local office at oracle.com/contact.

 blogs.oracle.com  facebook.com/oracle  twitter.com/oracle

Best Practices for Oracle Database Consolidation

October 2121

Author: Christian A. Craft

Contributing Authors: Glen Hawkins, Lawrence To, Tina Rose, Dan Norris, Kevin Diehl