Consolidation Best Practices

Oracle Database 12*c* plugs you in to the cloud

ORACLE WHITE PAPER | APRIL 2017

Table of Contents

Introduction

Traditionally, IT organizations have deployed individual databases and applications onto dedicated server infrastructure to support different departments or line of businesses (LOBs). This kind of segmented alignment of technology to business functions results in severe underutilization of the technology infrastructure and inefficient utilization of administrative resources managing such deployments. Additionally, such siloed deployments inhibit the ability of IT organizations to respond quickly to changing business needs.

To address these challenges, organizations are pursuing enterprise clouds to deliver cost savings while increasing business agility. This movement to a Cloud Computing model involves several transitions. Consolidation is one of the key steps in this journey, and allows organizations to achieve greater efficiencies in their operations by improving resource utilization, and lowering both capital and operational expenditures. Key to achieving these savings is standardization and reducing the number of distinct environments to manage.

Oracle Database 12c Release 2 offers significant advantages for consolidating application workloads. These benefits include:

1. Simplified Management – Reduce the number of distinct environments to manage. Manage many as one.
2. Streamlined Provisioning and Patching
3. Ease of Consolidation - .Consolidation without application changes is possible.

In this paper, we will review these capabilities and show how Oracle Database 12c aids in consolidation and accelerates your journey to the cloud.

Journey to Enterprise Cloud

You sometimes hear the statement that cloud computing is synonymous with consolidation. It is not! Consolidation is a cornerstone of cloud computing, but the enterprise cloud offers so much more.

The promise of cloud computing—greater agility, less risk, and lower costs—is real, but realizing that promise depends on the approach you adopt. Oracle offers choice and flexibility with the most comprehensive, modern, and secure portfolio of cloud products and services to meet all your business needs and deliver the full benefits of cloud computing.

Making the full transformation to an enterprise cloud may take several years, and will affect many aspects of organizations and roles, processes, policies and service delivery. Oracle has seen many enterprises organize their transformation into a phased approach -- a journey to enterprise cloud. The journey implements discrete steps that are each achievable and provide significant benefits. So however fast or far each organization chooses to go, they derive immediate value from even their initial steps.

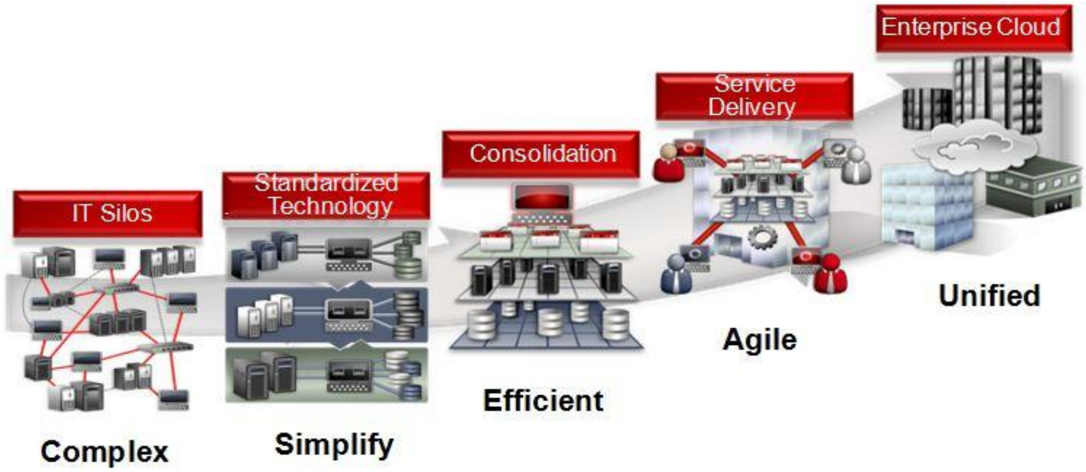


Figure 1. Journey to Enterprise Cloud

Oracle Database 12c is designed to enable enterprise clouds and brings new features that deliver key benefits at each phase of the journey. An Oracle whitepaper*, Journey to Enterprise Cloud*[1] , describes this phased approach.

The focus of this whitepaper is the consolidation phase, highlighting benefits offered by Oracle Database 12c, though we will briefly touch on standardization and the benefits to be gained from reducing complexity.

Standardization reduces complexity

Standardization is fundamental to the success of the enterprise cloud.

In the standardization phase the goal is to reduce complexity by moving from custom deployments to highly standardized deployments. The majority of service requests should be able to be fulfilled using a few standardized services.

It is important to understand your current silo'ed IT environment, and all that it includes, and then look for opportunities to simplify and standardize: infrastructure components (hardware and software), services, processes, and vendors.

[1] http://www.oracle.com/technetwork/database/database-cloud/journey-to-enterprise-cloud-wp-1959164.pdf

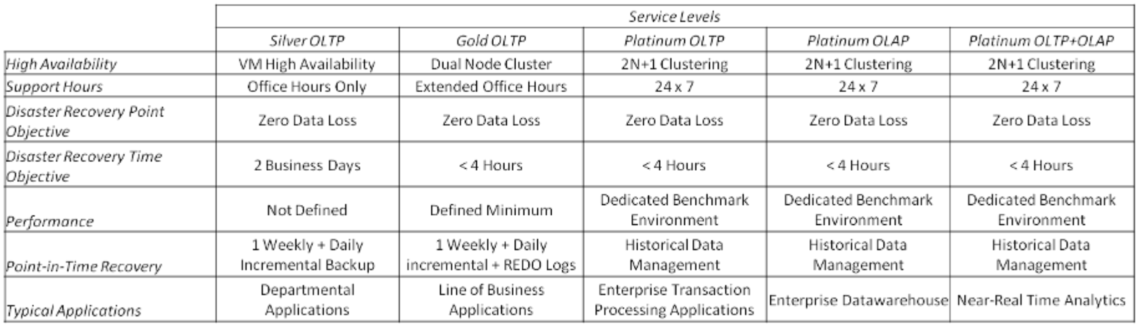
One of the key benefits of standardization is reducing the variety of environments you manage, thus reducing your administrative overhead. In the majority of cases this leads directly to operational expense (OpEx) savings. The impact on OpEx derives from how effectively you standardize your environment.

Oracle Enterprise Manager 12c can assist in this phase through the use of Consolidation Planner.

Standardized Services – creating building blocks

Defining a minimal catalog of standardized services to support the majority of business requests is the key deliverable of this stage. A standardized service is a simple to support, basic building block. A standardized service can be combined with other standardized services to build higher level business services.

The following example of standardized services at a major financial institution defines Platinum, Gold, and Silver Database Services.



	Service Levels				
	Silver OLTP	*Gold OLTP*	*Platinum OLTP*	*Platinum OLAP*	*Platinum OLTP+OLAP*
High Availability	VM High Availability	Dual Node Cluster	2N+1 Clustering	2N+1 Clustering	2N+1 Clustering
Support Hours	Office Hours Only	Extended Office Hours	24 x 7	24 x 7	24 x 7
Disaster Recovery Point Objective	Zero Data Loss	Zero Data Loss	Zero Data Loss	Zero Data Loss	Zero Data Loss
Disaster Recovery Time Objective	2 Business Days	< 4 Hours	< 4 Hours	< 4 Hours	< 4 Hours
Performance	Not Defined	Defined Minimum	Dedicated Benchmark Environment	Dedicated Benchmark Environment	Dedicated Benchmark Environment
Point-in-Time Recovery	1 Weekly + Daily Incremental Backup	1 Weekly + Daily incremental + REDO Logs	Historical Data Management	Historical Data Management	Historical Data Management
Typical Applications	Departmental Applications	Line of Business Applications	Enterprise Transaction Processing Applications	Enterprise Datawarehouse	Near-Real Time Analytics

Figure 2. Standardized services offered by a major financial institution

Note that even though standardized services offer greater efficiencies there does have to remain a degree of flexibility, through which custom services can be provided (when and if needed).

The above example is one of several we studied, and after identifying the common themes and variations in actual implementations, we decided to create a standardized service catalog for Oracle Database services in private clouds. We have leveraged customer experiences plus Oracle's best practices and key lessons learned to define four service offerings for Oracle Database services[2]. Customers can adopt or adapt these to quickly create the foundation for delivering Database as a Service with the service catalog model.

Consolidation lowers costs and improves manageability

The decision to consolidate is often arrived at as a consequence of other initiatives, often directed by cost saving. These could include a corporate "green" initiative to reduce power consumption. Offset capital expenditure through raising hardware utilization; reduce operational costs: power, space, administration.

All of these goals are achievable via a good consolidation strategy, which is all well and good, but much more can be achieved if consolidation is viewed as building a platform on which an enterprise cloud can be developed.

Database consolidation may start out as the primary driver of a business initiative. For many customers the outcome of the consolidation phase is all that is desired; the cost savings, both initial and on-going are sufficient in themselves. However, once consolidation has been achieved, and practices to support this approach in to the future implemented, it need not be the end-goal. Consolidation, along with standardization of the IT environment and the services it offers, is just the beginning.

[2] http://www.oracle.com/technetwork/database/database-cloud/private/service-catalogs-for-dbaas-2041214.pdf

Cost saving will come as a consequence. By consolidating your database services[3] onto shared infrastructure, you increase utilization of that infrastructure and reduce the hardware server footprint. By consolidating your standardized database services onto a shared environment, you can further improve utilization and also significantly reduce management costs by reducing the number of distinct environments that you have to manage. This lowers both capital and operational expenditures. Lower power consumption, a smaller datacenter footprint, and lower IT management expenses are just some examples of the savings.

Consolidation with Oracle Database 12*c*

Oracle Database 12c introduces features that are designed to facilitate database consolidation.

A new architecture, Oracle Multitenant, was introduced with Oracle Database 12*c* which greatly simplifies consolidation of multiple applications onto a shared database environment. By removing all limitations that previously existed with schema consolidation, such as namespace collisions, certification difficulties and so forth, Oracle Multitenant readily allows the creation of a single container database (CDB) that contains one or more pluggable databases (PDBs).

Oracle Multitenant is a licensed option of Oracle Database 12c Enterprise Edition that helps customers reduce IT costs through simplifying consolidation, streamlining provisioning, providing faster upgrade and migration options, to name a few. Improvements in the Advanced Security Option protect consolidated environments, without sacrificing performance. Consolidating your dedicated environments into pluggable databases within an Oracle Real Applications Cluster (RAC) container database improves availability, and provides flexibility to help meet fluctuating business needs.

Key Benefits of Multitenant Architecture

High consolidation density. The many pluggable databases in a single container database share its memory and background processes, letting you operate many more pluggable databases on a particular server platform than when deploying dedicated single-instance databases. This is the similar to the benefit that schema consolidation brings. Notably, though, the multitenant architecture removes all the barriers to adopting schema-based consolidation, such as namespace collision, shared schemas or users and so forth, and removes the ongoing operating problems that come with that approach.

Streamlined provisioning and cloning using SQL. A pluggable database can be unplugged from one container database and plugged into another. Alternatively, you can clone a pluggable database, within the same container database, or from one container database to another. These operations, together with creating a pluggable database, are done with new SQL commands and take just seconds. When the underlying filesystem supports thin provisioning, many terabytes can be cloned almost instantaneously simply by using the keyword snapshot in the SQL command.

New paradigms for rapid patching and upgrades. Patching one container database results in patching all of its pluggable databases. To patch a single pluggable database, you simply unplug/plug across a version boundary of the Oracle Database software.

[3] Database services represent groups of applications with common attributes, service level thresholds, and priorities. Database services provide a single system image to manage competing applications, and allow each workload to be managed as a unit. A database service can span one or more instances of an Oracle database, multiple databases in a global cluster, and a single instance can support multiple database services.

Manage many databases as one. By consolidating existing databases as pluggable databases, administrators can manage many databases as one. For example, tasks like backup and disaster recovery are performed at the container database level.

Dynamic between-pluggable database resource management. Oracle Database 12c Resource Manager is extended with specific functionality to instantly control the competition between the pluggable databases within a container database.

Improved Availability and Resiliency. Hardware migrations that require downtime can be performed faster using the unplug/plug functionality of PDBs. The failover of a container database, and its many pluggable databases, is faster than the failover of a set of single dedicated databases.

Improved Security. Easier configuration, better performance enhancements for all major security features (including Oracle Database Vault, Transparent Data Encryption, Unified Auditing and Database Firewall). Oracle Database Vault, Unified Auditing and Transparent Data Encryption can be configured at the PDB level.

Choosing a Consolidation Model

The business drivers for consolidation do not take in to account the requirements of individual applications. The task of choosing which applications can be housed in a shared environment is not always an easy one, but by careful consideration of a few key factors, the benefits of consolidation can be realized for the majority of applications.

Before we proceed, let us first consider an important point. Consolidation does not mean placing all of your databases into virtual machines. Virtualisation, in many cases, replaces physical silos with virtual silos. That does not reduce your IT complexity.

How PDBs solve the IT complexity problem

According to the IT Infrastructure Library (ITIL) a Configuration Item (CI) is defined as any component that needs to be managed to deliver an IT service.

Consider how many CIs are present in a data center with 8 databases, each running in its own dedicated environment. There are 8 Oracle database instances, 8 operating system images and 8 separate machines. That is 24 CIs that need to be managed, monitored, patched, and backed up. And that is saying nothing about the average utilization of these assets. This is shown in Figure 3, eight dedicated databases, each running in its own silo.

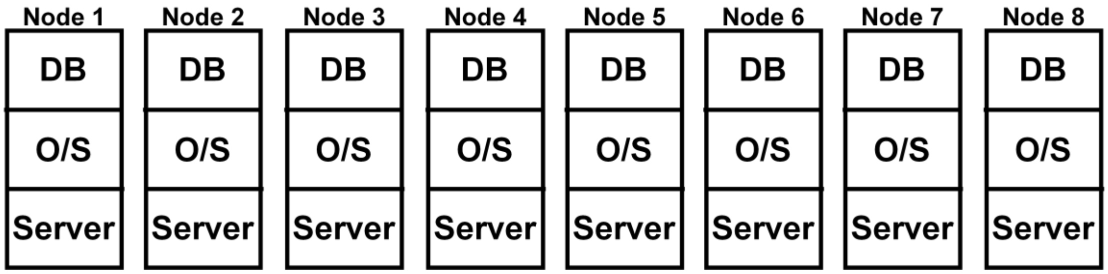


Figure 3. Eight Dedicated Database Environments

In Figure 4, below, we have consolidated the eight dedicated databases in to a single Oracle RAC CDB, running on a 2 node cluster. This CDB contains 8 PDBs, where we have opened, as an example, 4 PDBs on each instance. In this example we have reduced our CIs to 14: 8 PDBs, 2 CDB instances, 2 Operating System images, and 2 nodes, giving a significant reduction in managed components.

Further to this, while we have 14 CIs to monitor and manage, patching and backup are done at the CDB level , meaning management of even fewer CI components.

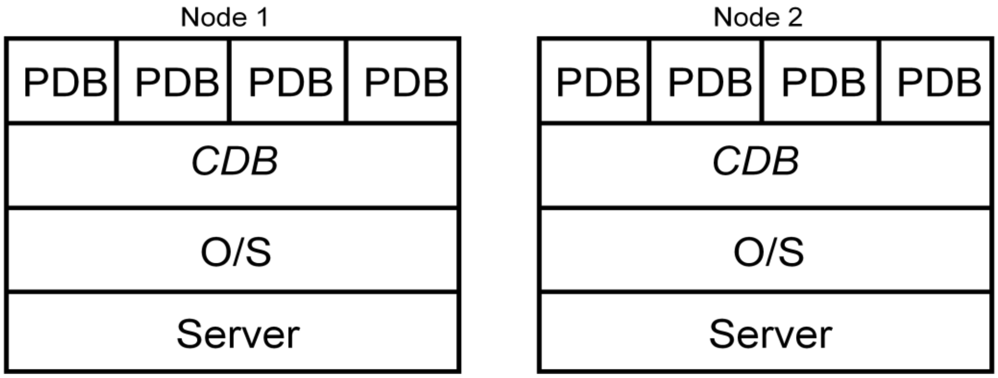


Figure 4. Oracle RAC CDB with 8 PDBs open on each instance

Selecting the appropriate level of isolation

Inter-tenant isolation requirements are a major influence in selecting the method of consolidation and have significant bearing on the degree of consolidation that can be achieved. Isolation and flexibility, when it comes to sharing resources, are opposed to one another. There is a trade-off between shared environments, which offer greater flexibility, lower administrative overheads, and greater resource utilization, and higher isolation, which minimizes security concerns, and reduces inter-application consolidation considerations (which can influence when the database can be upgraded, for example).

The general recommendation is to isolate only as needed, to meet the tenant isolation requirements of the specific deployment. Typically the isolation requirements relate to governmental compliance or regulatory mandates, but these can also include such things as database or operating system version restrictions.

Isolation and its influence on consolidation

Isolation can be physical or logical and can be considered in the context of four areas: fault, security, resource and operational.

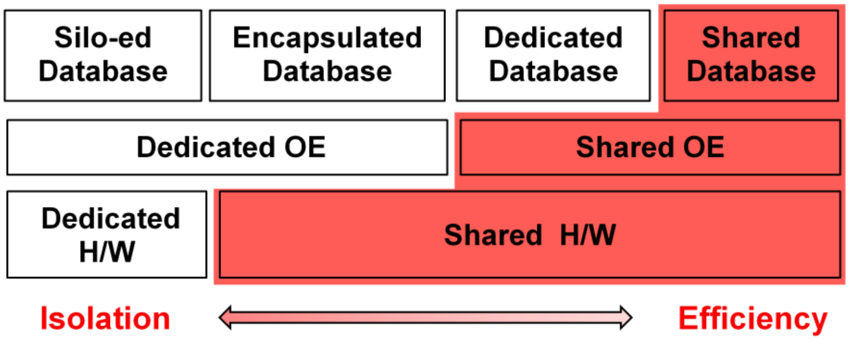


Figure 5. Sharing promotes greater efficiency

Choosing to consolidate multiple applications in a single database using pluggable databases, hosting multiple databases on a single platform, or a combination of both approaches depends on the level of isolation that your consolidation strategy demands. Each consolidation model deals with isolation slightly differently, using a combination of Oracle Database 12c and operating system capabilities, in combination with other advanced features and products.

This section will focus on tenant isolation in private database clouds, comparing database, schema, and pluggable database deployment models.

Pluggable Database Consolidation

In this model applications are consolidated as PDBs running in a single CDB. Multiple CDBs can be provisioned such that each CDB can run on one or more servers in a cloud pool[4]. The granularity of the tenancy is the PDB.

Oracle Database 12c supports up to 252 PDBs in a single CDB. In addition, up to 1024 Dynamic Database Services can be created on a given CDB, bringing greater economies of scale than consolidating dedicated databases on to a shared platform, or in individual VMs that are consolidated on to a shared server.

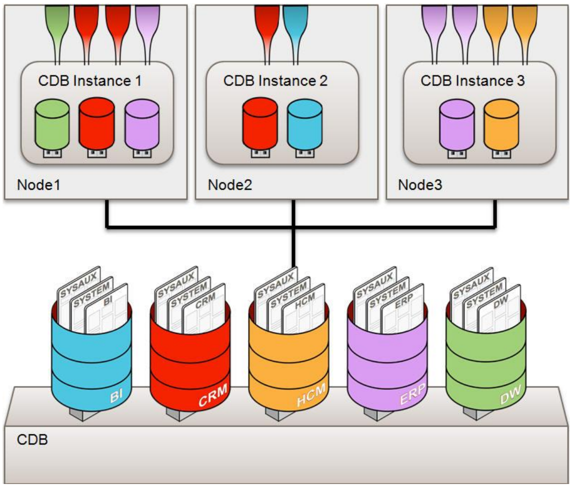


Figure 6. Consolidating on to multitenant architecture

Figure 6 shows a single container database with five pluggable databases. The CDB is running in an Oracle RAC cluster, in which each PDB is accessed through an associated service (the colored funnels), being offered on one or more instances.

Fault Isolation

An application fault in a given PDB will not affect other PDBs in the same CDB. The application's activities are constrained to the PDB that it is connected to.

Resource Isolation

Resource isolation deals with the allocation and segregation of system resources.

If multiple CDBs are active on the same node, then contentious resources include CPU, memory and I/O (both storage capacity and IOPs). As per the database consolidation approach:

Memory – Set appropriate SGA_TARGET values on a per instance basis, noting that these values must be consistent for all instances of the same database. SGA_TARGET does not enforce a hard limit for PGA values. PGA limits are set via PGA_AGGREGATE_LIMIT which will impose a hard limit on PGA memory use and can help avoid issues related to excessive paging.

[4] A cloud pool refers to a collection of servers that have shared storage and that share a private network. A cloud pool can therefore be an Oracle Clusterware cluster or a pool of virtual servers. A private cloud is an aggregation of cloud pools.

Use HugePages when operating on systems with large physical memory.

It is highly discouraged to over-commit memory resources.

A conservative target would not exceed 75% of available memory for all databases on the one node. On Exadata hardware:

OLTP: SUM of CDBs (SGA_TARGET + PGA_AGGREGATE_TARGET) + 4 MB * (Maximum PROCESSES) < Physical Memory per Database Node

DW: SUM of CDBs (SGA_TARGET + 3 * PGA_AGGREGATE_TARGET) < Physical Memory per Database Node

CPU – Set appropriate values through the use of CPU_COUNT or enabling instance caging[5]. The latter is the recommended approach as the use of instance caging enforces a Database Resource Manager (DBRM) resource plan which provides more control over CPU consumption.

The multitenant architecture extends Resource Manager to allow a CDB-level plan to manage the contention of resources between the PDBs. It is good practice to use UTILZATION_LIMIT within the Resource Manager profile when consolidating applications. By limiting resources, users of applications that were initially in a sparsely populated CDB, will not experience performance change as additional applications are added to the CDB.

Oracle Database Quality of Service Management (QoS Management) manages resources that are shared across applications, and adjusts the system configuration to keep the applications running at the performance levels required by your business.

Consolidating PDBs in to a single CDB can lead to an increase in the number of client sessions directed to a given node (or instance). A large number of simultaneous connection attempts, a logon storm, due to failover, for example, or improperly configured mid-tiers, can impact other consolidated applications. A logon storm will present itself as a resource shortage, but in severe cases can lead to scheduling conflicts and other cascading failures.

Minimizing the impact of logon storms requires appropriate configuration of mid-tier connection pools, and care taken as to where services are offered. In addition, the Oracle Listener can be configured to limit the rate of connections.

Operational Isolation

Minimizing the impact of recovery/restore of lost data is important in the PDB model, as is minimizing the impact of patch application and upgrade.

The manage many-as-one approach that PDBs offer allows for backups to be scheduled automatically for each CDB, whilst a given PDB can be restored individually. The PDB point-in-time restore operation has no impact on peer PDBs.

Flashback of a PDB is not available in Oracle Database 12c Release 1.

Patching

Patching databases in a consolidated environment involves planning for the patch application and the actual applying of the patch itself. Expectations should be set that one-off or unscheduled patching is inefficient, and should be discouraged. A schedule for patch application (for example, Oracle Patch Set Updates (PSU)) should be pre-defined, and the tenants must acknowledge this. Having said that, provision must be made for the application of one-off patches, but these patches need to be evaluated for the impact they have on the entire CDB.

[5] Refer to the section entitled Cloud Pool Design, in this paper, for additional information on Instance Caging

If a one-off patch is needed urgently by only one or a few applications (PDBs) then the most efficient method of applying this patch would be to create a new CDB containing the additional patches, and them migrate individual PDBs to this CDB using the unplug/plug operations. Note that this would increase management complexity, but may be tolerable for a short period.

Security

In any consolidated environment taking the approach of “least privilege” will provide the most benefit to tightening security. In most cases a single Oracle Home will host all databases running on that node. This means that any O/S user that is part of the DBA group for that Oracle Home has SYSDBA access to all database instances running from that home. This is a good approach from a manageability standpoint, but it does open security issues.

It is recommended to implement the following best practices:

- Minimize access to the database server. Only allow SQL*Net pipe access.
- Used named accounts for all DBAs and provide sudo access for privileged commands
- Create common users for administrative tasks that span more than one PDB
    - Common users should not own objects outside of those required to run administration tasks (certain procedures for example).
- Use roles to restrict privilege; maintain limited access to SYSDBA and SYSOPER. Additional administrative roles are available with Oracle Database 12c for backup management, Oracle Data Guard management, and encryption key management: SYSBACKUP, SYSDG, and SYSKM respectively. Use where appropriate.
- Enable Oracle Database Vault to provide role separation and control data access
    - Pre-defined Data Realms are available for E-Business Suite, Siebel and PeopleSoft.
    - Encrypt data at rest

Database Consolidation

With the Database Consolidation model, individual databases are consolidated on physical servers clustered together into cloud pools. Any server in the pool can host one, or more, Oracle database instances.

By utilizing Oracle RAC or Oracle RAC One Node, databases inherit high availability through server redundancy. Elasticity and scalability can be achieved through adding more servers to a pool, or adding additional CPU, memory, or I/O cards to a given node or instance.

Despite the importance of standardization, exceptions do need to be allowed for, and as such the addition of larger (higher capacity) nodes, and forming heterogenous clusters (from a node’s CPU or memory configuration perspective) is also possible.

Fault Isolation

The granularity of this consolidation approach is the database. Each database and its associated instance is isolated from other databases in the same pool. The failure of a given instance is generally constrained to that instance, even though the databases may run from the same Oracle Home. Situations may arise where the unresponsiveness of a single instance may lead to a conclusion that impacts multiple instances; typically when a node reboot is required. Application design and implementation of best practices can limit the impact of instance or node failure. For example the use of dynamic database services and connection pools, when combined with Fast Application Notification (FAN), allow applications to respond more quickly to outages, thus limiting their impact.

Resource Isolation

Resource isolation deals with the allocation and segregation of system resources. In the database consolidation model competing resources include CPU, memory and I/O (both storage capacity and IOPs).

Memory – Set appropriate SGA_TARGET values on a per instance basis, noting that these values must be consistent for all instances of the same database. SGA_TARGET does not enforce a hard limit for PGA values. PGA limits are set via PGA_AGGREGATE_LIMIT which will impose a hard limit on PGA memory use and can help avoid issues related to excessive paging.

Use HugePages when operating on a system with large physical memory.

It is highly discouraged to over-commit memory resources.

On Exadata hardware:

- OLTP: SUM of CDBs (SGA_TARGET + PGA_AGGREGATE_TARGET) + 4 MB * (Maximum PROCESSES) < Physical Memory per Database Node
- DW: SUM of CDBs (SGA_TARGET + 3 * PGA_AGGREGATE_TARGET) < Physical Memory per Database Node

CPU – Set appropriate values through the use of CPU_COUNT or enabling instance caging[6]. The latter is the recommended approach as the use of instance caging enforces a Database Resource Manager (DBRM) resource plan which provides more control over CPU consumption.

Operational Isolation

Operational isolation ensures that management or maintenance activities performed on a database or its operating environment do not affect other running databases in the same cloud pool. These activities include startup and shutdown of instances, patching, and backup or restore operation.

Startup/Shutdown

Commonly only one, or some minimum number of Oracle Homes is used for all consolidated databases. Named users should be created for each Cloud DBA, and those users are then added to the database password files (REMOTE_LOGIN_PASSWORDFILE must be set to EXCLUSIVE). These named users are granted the SYSDBA role. By having separate password files for each database, users can only gain SYSDBA privileges for the database(s) they administer.

Patching

Similarly as for the PDB consolidation approach, patching databases in a consolidated environment involves planning for the patch application and the actual applying of the patch itself. Expectations should be set that one-off or unscheduled patching is inefficient, and should be discouraged. A schedule for patch application (for example, Oracle Patch Set Updates (PSU)) should be pre-defined, and the tenants must acknowledge this. Having said that, provision must be made for the application of one-off patches, but these patches need to be evaluated for the impact they have on the entire database.

The most efficient method for patching is to clone the Oracle Home, apply the patch, and then switch the database instance to the new home. In RAC environments, rolling patches should be used where provided.

Security Isolation

Similar to that discussed in the PDB consolidation section.

[6] Refer to the section entitled Cloud Pool Design, in this paper, for additional information on Instance Caging

Consolidating Multiple Container Databases

At the database level, the database workload consolidation and sizing guidelines described above apply to the consolidation of multiple CDBs and dedicated databases onto shared cloud pools. However, you must augment the server and operating system sizing guidelines based on the number of PDBs per CDB, and mix of CDBs and dedicated databases that will be consolidated onto these cloud pools.

The general guidelines for consolidating databases are driven by the service-level-agreements (SLAs) with your database cloud tenants. At the infrastructure level, it boils down to mapping these SLAs to the cloud pools most suited to deliver those SLAs. The cloud pools themselves should be built using standardized hardware and software components and sized appropriately to support pre-defined consolidation density and isolation policies.

For example, Figure 7 shows a database cloud deployment, where a given pool of servers in a cloud pool has been divided into three server pools hosting three policy-managed CDBs[7]. Each CDB, in this example, could represent a database with a different characterset, at a different patch level, belonging to a different line of business, or similar consideration.

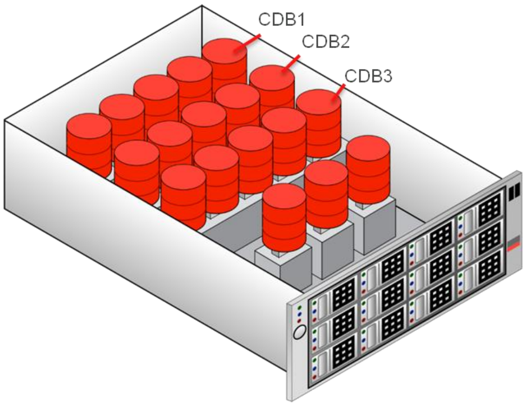


Figure 7. Database Consolidation with Oracle Database 12*c*

While pursuing consolidation of Container Databases, we recommend that you pre-define policies for each database and enforce these policies while provisioning services.

- Have a maximum number of PDBs per CDB
- Define a maximum number of CDBs and/or dedicated databases per cloud pool

During deployment, strict enforcement of these policies will result in uniform environments. Such environments lend themselves to easier automation and lifecycle management.

Schema Consolidation

In this model the consolidated database consists of one or more application schemas running across one or more instances in a cloud pool. Customers who have implemented this approach typically limit the number of applications (schemas) to less than 20. The tenancy granularity is the schema.

Fault Isolation

[7] Policy-managed deployment is based on *server pools*, where database services run within a server pool as singleton or uniform across all of the servers in the server pool. Databases are deployed in one or more server pools and the size of the server pools determine the number of database instances in the deployment.

An application fault in one schema will not affect applications in other schemas. Logon storms or improperly configured mid-tiers can impact multiple applications. The use of properly configured mid-tier connection pools is important to limit this impact. Poorly written database resident code, PL/SQL for example, can affect other unrelated applications. Strong code review during development and thorough testing prior to application deployment are essential..

Resource Isolation

Resource management is a necessity for schema consolidation. Oracle Database Resource Profile Limits provide a basic approach to limiting resource usage. By setting resource limits, users are prevented from performing operations that tie up the system and prevent other users from performing operations. Note that you can also use resource limits for security to ensure that users log off the system and do not leave the session connected for long periods of time. DBRM and QoS Management supplement the resource profile.

Applications can be grouped in to consumer groups with appropriate DBRM resource plan directives. The resource plan controls the allocation of CPU, I/O (in Exadata deployments) and parallel server process resources to the consumer group.

Storage usage can be controlled through the use of tablespace quotas.

Operational Isolation

Minimizing the impact of recovery and restore operations is a major goal of operational isolation in this model, as is patch application management.

For the most efficient data restore possible, a careful design of the backup policy is needed. The backup method should include the restore granularity appropriate for the application. Nightly backups as well as Data Pump exports of individual schemas is a typical approach. Lost or deleted data can be recovered using Flashback methodology (table, query, or transactional) to provide the least invasive restore procedure. If data has aged from the flashback area, the restore table package is also useful for lost data.

The approach to patching is similar to that discussed in other consolidation methods.

Security Isolation

Security isolation between schemas is one of the most important aspects of schema consolidation. Oracle Database profiles can be used to limit access to data, but in most cases more stringent security measures and policies must be enacted. Always protect data at rest (encryption), provide granular access control, and implement security auditing. These practices imply the use of Transparent Data Encryption, Database Vault Realms, and Oracle Audit Vault and Database Firewall for runtime audit management.

The following should also be implemented as best practice:

- Limit `SYSDBA`, `SYSOPER`, and `SYSASM` access. Take advantage of `SYSBACKUP` and `SYSDG` roles where required.
- Ensure the use of private synonyms. Avoid use of public synonyms.

Mandate strong database passwords and set appropriate values for `PASSWORD_LOCK_TIME` and `FAILED_LOGIN_ATTEMPTS.`

Cloud Pool Design

The initial size of the cloud pool will depend on the type of applications being housed and their individual capacity requirements. In most cases the private cloud will be made up of several smaller cloud pools, it will not be a single pool.

Cloud pools should be aligned with the following requirements:

- **Business**
  - Build separate cloud pools for lines of business or departments
  - Create separate cloud pools for different SLAs, compliance requirements or test and development
- **Functional**
  - Build a cloud pool for similar functioning applications. For example, separate internal versus external applications
- **Technical**
  - Separate cloud pools based on OS type, database version, or isolation requirements
  - Consider the complementary nature of workloads
  - Cloud pools with similar High Availability (HA) goals

Cloud pools are generally built for specific configurations and support specific business requirements. It is best practice for applications with similar SLA requirements to co-exist in a consolidated environment, but it is generally not suitable for critical and non-critical applications to reside in the same cloud pool.

The number of applications that can be consolidated depends on the size, resource consumption and SLAs of the applications to be grouped. Furthermore, a pre-defined threshold for system resource usage should be set, which will dictate the capacity of the cloud pool.

It is recommended to build cloud pools using standardized modular building blocks. Many customers have standardized on a four-node cluster as this provides flexibility for service placement and application load growth, as well as increased availability through more options for planned outage and protection against component failure.

The following describes the configuration of physical resources:

CPU

Initial CPU sizing will depend on what applications will be housed; allow an additional 10% for operational tasks, such as backup, or scheduled tasks, and 15% to account for failover of workload. Operating nodes in a cloud pool at 75% CPU capacity provides a good balance between general usage and headroom.

The initial size of the cloud pool will depend on the type of applications being housed and their individual capacity requirements. In most cases the private cloud will be made up of several smaller cloud pools, it will not be a single pool.

Cloud pools should be aligned with the following requirements:

- **Business**
  - Build separate cloud pools for lines of business or departments

    - Create separate cloud pools for different SLAs, compliance requirements or test and development

- **Functional**
    - Build a cloud pool for similar functioning applications. For example, separate internal versus external applications
- **Technical**
    - Separate cloud pools based on OS type, database version, or isolation requirements
    - Consider the complementary nature of workloads
    - Cloud pools with similar High Availability (HA) goals
- Cloud pools are generally built for specific configurations and support specific business requirements. It is best practice for applications with similar SLA requirements to co-exist in a consolidated environment, but it is generally not suitable for critical and non-critical applications to reside in the same cloud pool.
- The number of applications that can be consolidated depends on the size, resource consumption and SLAs of the applications to be grouped. Furthermore, a pre-defined threshold for system resource usage should be set, which will dictate the capacity of the cloud pool.
- It is recommended to build cloud pools using standardized modular building blocks. Many customers have standardized on a four-node cluster as this provides flexibility for service placement and application load growth, as well as increased availability through more options for planned outage and protection against component failure.
- The following describes the configuration of physical resources:
- CPU
- Initial CPU sizing will depend on what applications will be housed; allow an additional 10% for operational tasks, such as backup, or scheduled tasks, and 15% to account for failover of workload. Operating nodes in a cloud pool at 75% CPU capacity provides a good balance between general usage and headroom.

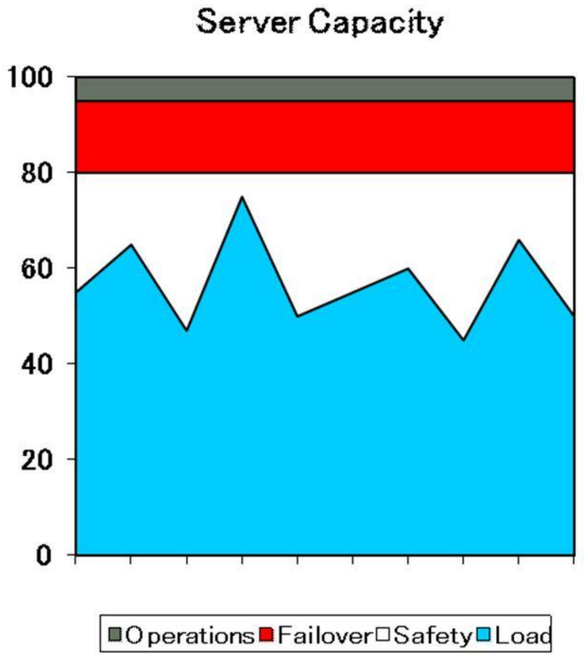


Figure 8. CPU Capacity

A database, CDB or non-CDB[8], that is provisioned in a cloud pool should get a minimum of two CPUs. This can be enforced using CPU_COUNT (set in the init.ora file) or by using the instance caging feature[9]. Instance caging, whilst also set using the CPU_COUNT parameter, has the additional benefit of enforcing resource usage limits via Database Resource Manager.

Instance Caging is used to segregate CPU usage amongst databases on the same machine. These databases could be two or more CDBs, non-CDBs, or a combination of both.

Partitioning versus Over-Provisioning

Instance caging is a means of reducing contention for CPU resource between multiple database instances sharing the same server. This is done by setting a maximum on the number of CPUs on which the processes associated with a given instance are scheduled. This is applicable to both CDBs and non-CDBs. There are two approaches to determining the maximum number of CPUs: partitioning and over-provisioning.

Partitioned CPU

In the partitioned approach the aggregate of the `CPU_COUNT` settings across all of the database instances on a given node does not exceed the total number of CPUs. This approach is recommended for mission-critical applications, and is also required if QoS Management is to be used.

With this configuration there should be no contention for CPU resources between database instances. However, if an instance does not use its CPU allocation then these resources are unavailable for use by other instances.

The following recommendation should be used for partitioning CPUs, this ensures CPU resources are available for critical processes such as Oracle Clusterware and its agents, ASM, and so forth:

[8] Note that the term non-CDB refers to the database architecture familiar to those who have used Oracle Database 11g or earlier. Oracle Database 12*c* supports both multitenant and non-CDB architectures

[9] Instance caging is discussed in detail in http://www.oracle.com/technetwork/database/focus-areas/performance/instance-caging-wp-166854.pdf

```
SUM(CPU_COUNT) < 75% Total CPUs
```

Over-Provisioned CPU

With the over-provisioned approach the sum of the CPU_COUNT across all instances can exceed the number of CPUs. Better resource utilization is provided with over-provisioning, however, it is possible for contention to occur if multiple databases are heavily loaded at the same time. Consequently, performance will degrade.

Over-provisioning is recommended for systems running non-critical applications, test or development, or any databases that do not have strict SLA requirements. Avoid using over-provisioning for databases with high I/O requirements or those with high transaction rates.

The following recommendation is made:

```
SUM(CPU_COUNT) <= 2x(Total CPUs)
```

The allowing of twice the allocation of Total CPUs is based on hyper-threading of 2 CPU threads per core. Care should be taken when over-provisioning on systems where the CPU:thread ratio is higher than two (for example, Sun T4 systems can have thread counts of up to 64 threads per CPU).

Databases running on Exadata Database Machines can be configured with higher levels of over-provisioning. Refer to the current Exadata Best Practice papers for the most up-to-date recommendation.

In addition, limit the Parallel Max Query servers to less than or equal to 20 times the number of Total CPUs or threads.

Memory

Configuring memory can be more straightforward in the cloud pool than CPU sizing. Do not over-commit memory resources.

For existing databases that will be consolidated, determine SGA and PGA memory using AWR reports, `V$SGASTAT/ V$PGASTAT` or EM Automatic Memory Advisor. For new applications, after the basic sizing exercise is performed, set conservative values `for MEMORY_TARGET` and aggressive values for `MEMORY_MAX_TARGET`.

The following are general guidelines for the memory footprints:

Once the SGA/PGA information is obtained, evaluate the following before each migration or placement into the Private Cloud.

OLTP applications:

```
SUM of databases (SGA_TARGET + PGA_AGGREGATE_TARGET) < 80% Physical Memory per
                                Database Node
```

DW/BI applications:

```
SUM of databases (SGA_TARGET + 3 * PGA_AGGREGATE_TARGET) < 80% Physical Memory per
                                Database Node
```

Note: Database Consolidation does not make the most efficient use of memory, as each database instance will consume its own SGA and PGA. PDB consolidation is the most efficient memory usage model, since a single large database instance with a consolidated SGA is configured in the cloud pool.

Storage

Cloud DBAs should review existing applications' storage use before consolidation. Silo’ed databases are typically given more storage than they will use, thus one of the key items to review is how much of the allocated storage is actually in use; i.e., how much is free and how much contains data. If there is overallocation of storage, then this may be a good opportunity to consolidate the storage space. This can be performed using Oracle Data Pump or any mechanism that will logically migrate the data. DBAs should research data growth patterns by using EM 12c to evaluate a schema's storage growth patterns on a per tablespace/datafile basis. Furthermore, before migrating applications into a private cloud, application owners should consider cleansing the database of obsolete or unneeded data. This not only improves storage efficiency, but also improves overall migration time.

Storage IOPS is probably the most overlooked area in consolidated environments. DBAs should look at average and peak IOPS for each database or application to be consolidated as part of the consolidation planning exercise.

Use AWR reports to collect the following I/O metrics.

```
  IOPS = "physical reads total I/O requests" + "physical writes total I/O requests"

       MBytes/s = "physical reads total bytes" + "physical writes total bytes"
```

These metrics will aid in determining storage throughput needed to support the application. Aggregate the IOPS or MBytes/s for all nodes if the existing application is running on RAC.

Complementary Workloads

One of the key aspects of a successful private database cloud deployment is to ensure that only complementary workloads are housed. Poor performance in a consolidated environment, missed SLAs, and outages can be a consequence of mixing non-complementary, or antagonistic workloads.

When consolidating workloads ensure that the consolidated workloads' peak CPU utilisation does not greatly exceed the average CPU utilisation. The gap between peak and average should be kept to a minimum, which ensures that the CPUs are being utilized as fully as possible.

Evaluating workloads for inclusion in the consolidated environment involves looking at the changes the new workload causes on the peak and average CPU utilisation. A complementary workload will cause the average load to increase more than the peak, as shown in Figure 9. An optimal situation would be where the peak usage remains unchanged, while the average increases.

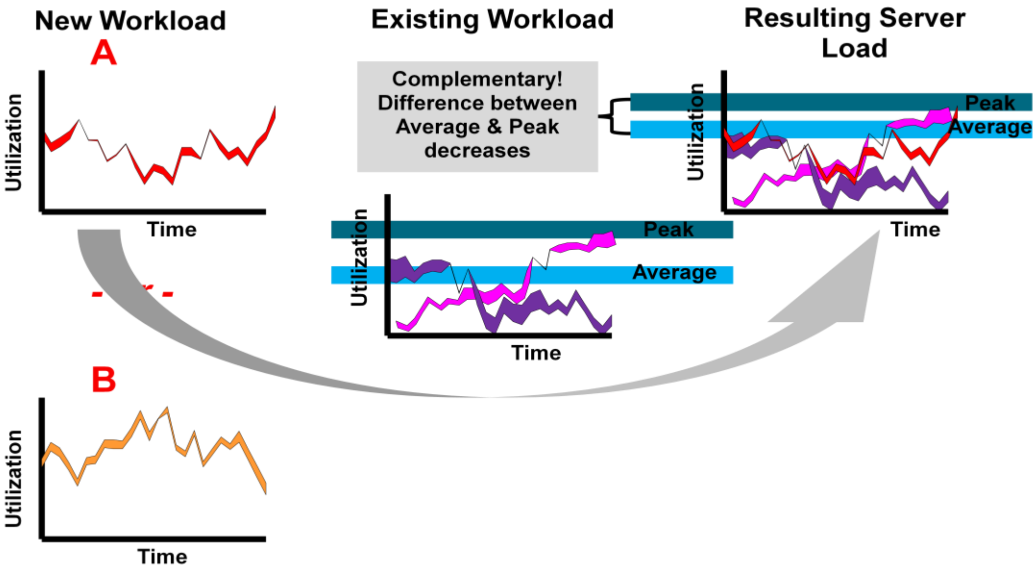


Figure 9. Complementary Workloads

In the antagonistic case, Figure 10, the peak increases more than the average. This means that the CPUs are underutilized during periods of low usage; they must be kept available for peak loads.

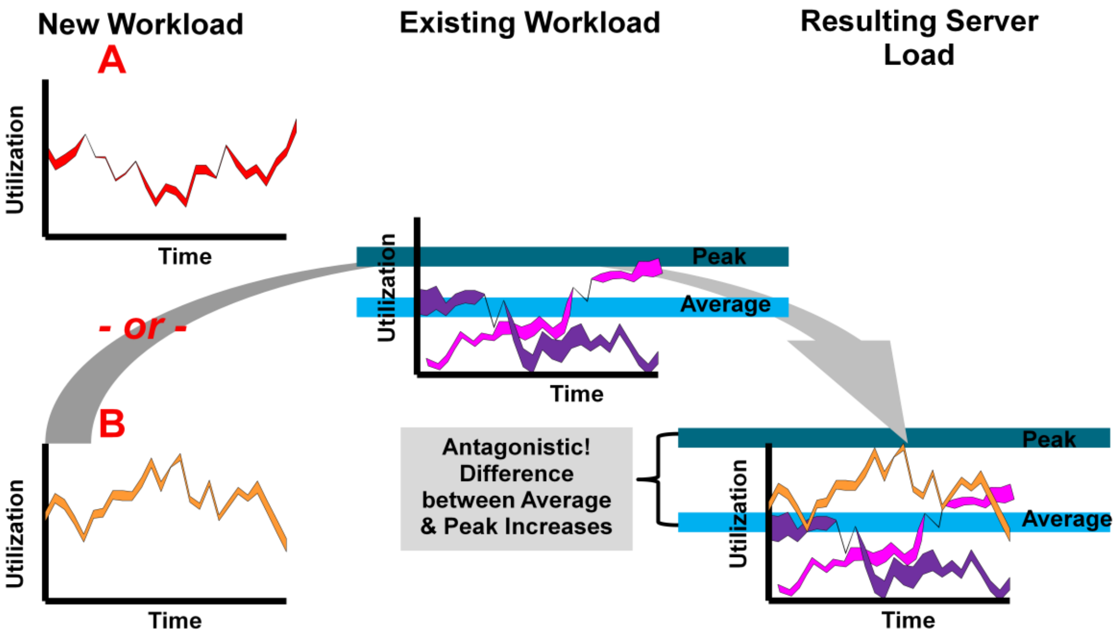


Figure 10. Antagonistic Workloads

Real Application Testing (RAT) is useful in determining the impact that new workloads will have on an existing system.

RAT works for all consolidation approaches: database, schema, PDBs and non-CDB consolidation. Workloads captured on different databases can be replayed concurrently using Consolidated Database Replay. This enables you to consolidate multiple workloads captured from one or multiple systems and replay them concurrently on a single test system. Using Consolidated Database Replay will help you assess how the database consolidation will affect the production system and if a single machine, for example an Oracle Exadata Machine, can handle the combined workloads from the consolidated databases.

Workload Folding allows you to combine the workloads and replay them simultaneously, while time-shifting allows the workload peaks to be aligned.

More information on Real Application Testing can be found in *Oracle® Database Testing Guide 12c Release 1 (12.1).*

Oracle Enterprise Manager 12*c* Cloud Management Pack

Oracle Enterprise Manager 12c is Oracle’s integrated enterprise IT management product, and provides the industry’s first complete cloud lifecycle management solution. Oracle Enterprise Manager’s business-driven IT management capabilities allow you to quickly set up, manage and support Enterprise Clouds and traditional Oracle IT environments all the way from applications to disk.

Oracle Cloud Management Pack for Oracle Database delivers capabilities spanning the entire Database cloud lifecycle.

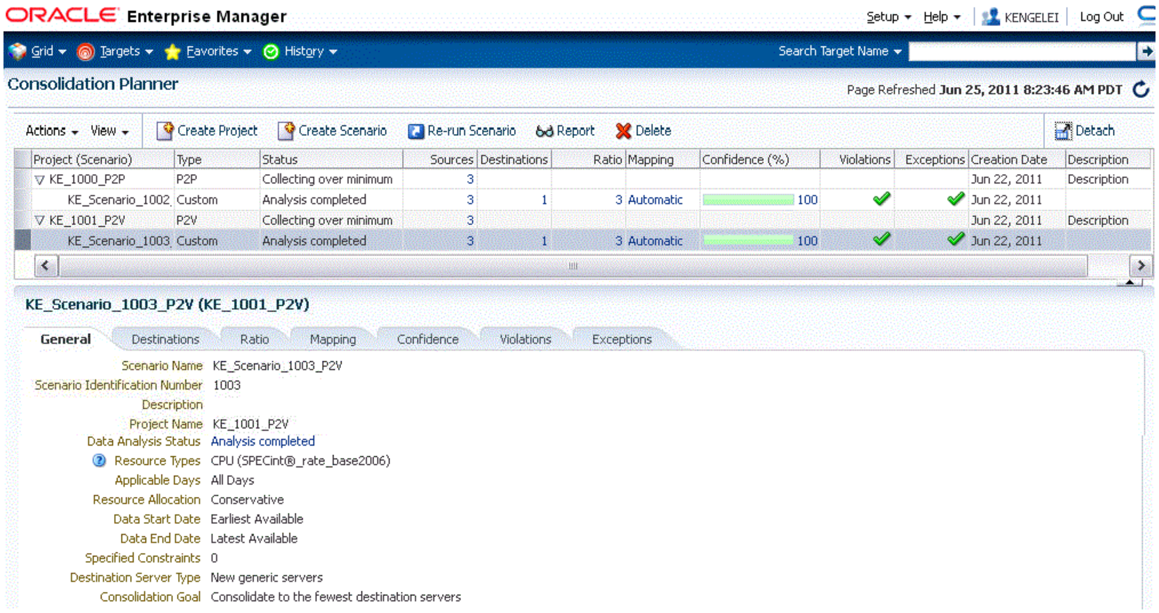
Consolidation Planner

To build a cloud, administrators must first understand what their existing resources are and their utilization. Enterprise Manager automatically discovers their infrastructure and its topology, and helps then understand the current workloads in the environment.

Administrators can then use Consolidation Planner to run different scenarios for redistributing workloads onto existing systems or new environments (utilizing a what-if capability), and determine if this will result in SLA violations. These scenarios can be tested either on engineered systems, such as Exadata, or off-the-shelf hardware.

A variety of consolidation choices are offered; as mentioned consolidation onto Exadata can be studied, as can consolidation on other physical or virtual environments. Consolidation Planner can also provide consolidation advice for application consolidation on to Exalogic engineered systems.

Consolidation Planner provides a guided migration path to the Enterprise Cloud. Consolidation advice is based on both technical and business reasons. For example, processor architecture, line-of-business ownership, location, and lifecycle can all be considered.



Consolidation Planner

Page Refreshed **Jun 25, 2011 8:23:46 AM PDT**

Project (Scenario)	Type	Status	Sources	Destinations	Ratio	Mapping	Confidence (%)	Violations	Exceptions	Creation Date	Description
▽ KE_1000_P2P	P2P	Collecting over minimum	3							Jun 22, 2011	Description
KE_Scenario_1002	Custom	Analysis completed	3	1	3	Automatic	100	✓	✓	Jun 22, 2011	
▽ KE_1001_P2V	P2V	Collecting over minimum	3							Jun 22, 2011	Description
KE_Scenario_1003	Custom	Analysis completed	3	1	3	Automatic	100	✓	✓	Jun 22, 2011	

KE_Scenario_1003_P2V (KE_1001_P2V)

Scenario Name	KE_Scenario_1003_P2V
Scenario Identification Number	1003
Description	
Project Name	KE_1001_P2V
Data Analysis Status	Analysis completed
Resource Types	CPU (SPECint®_rate_base2006)
Applicable Days	All Days
Resource Allocation	Conservative
Data Start Date	Earliest Available
Data End Date	Latest Available
Specified Constraints	0
Destination Server Type	New generic servers
Consolidation Goal	Consolidate to the fewest destination servers

Figure 11. EM 12*c* Consolidation Planner

Database Provisioning Console for all Provisioning Activities

The Database Provisioning console is a starting point for your database provisioning activities. The console displays information about provisioning setup, profiles, deployment procedures, and information about getting started with provisioning.

You can provision container databases or pluggable databases, in addition to the non-CDB options that existed prior to Oracle Database 12c. Oracle Real Applications Clusters, or Oracle RAC One Node Databases are options for both the container database or the non-CDB. Any of these databases can be provisioned using database templates, installation media, database entities, or through the use of provisioning profiles to standardize deployments.

Chargeback

Chargeback is used to allocate the costs of IT resources to the people or organizations who consume them. While it can be applied in situations where IT resources are dedicated, it is particularly relevant in situations where resources are shared, as without some way to meter and charge for consumption there will be a tendency for users to use more resources than they need. This problem is exacerbated in cloud environments where users are able to provision their own resources using self-service.

Chargeback functions enable you to track the use of business-critical resources or metrics by consuming entities (for example, cost centers) and enable businesses to report back the usage charges to the consuming entities. IT departments can accurately share or report costs with business users or business units commensurate with the usage of the resources.

Recent enhancements to Cloud Control mean you can now share an Oracle RAC database not only by instance, but also by service. Services within an Oracle RAC database can be assigned to different cost centers, but must all be assigned to the same charge plan. This aligns well with pluggable databases, as all access to a PDB is via a service.

Chargeback has three basic metrics against which to compute resource consumption: CPU usage, and memory and storage allocation. These metrics comprise a universal charge plan that can be applied to any target type configured for Chargeback.

Adopting a Metering and Chargeback (or Showback) model can deliver significant benefits to both IT and Line of Business users:

- Providers and consumers understand the cost of the services delivered and establish accountability for the consumption of resources
- Metering gives visibility into how the environment is being used and provides opportunity to make improvements in the environment and service catalog offerings
- Chargeback benefits users by placing them in control of their IT costs.

Rapid Home Provisioning

Rapid Home Provisioning (RHP) is a new feature of Oracle Grid Infrastructure Release 12.1.0.2 that allows for centralized software deployment and maintenance. Software only need be installed once, then stored on the RHP server, and from there can be provisioned to any node or cluster in the private cloud, as many times as needed.

RHP will

- Eliminate the need to patch individual databases
- Update any number of databases with a single command
- Ensure standardization through gold image lineage
- Provision and maintain both Oracle Database and Grid Infrastructure homes

Rapid Home Provisioning Server

The Rapid Home Provisioning Server is a highly available software provisioning system. The Rapid Home Provisioning Server acts as a central server for provisioning Oracle homes and Grid Infrastructure homes, making them available to Rapid Home Provisioning Clients; clients can be other Grid Infrastructure clusters or single nodes within your enterprise, or cluster domain.

The main features of the Rapid Home Provisioning Server are:

- Efficiently stores gold images for the managed homes, including separate binaries, and metadata related to users, roles, and permissions.
- Provides a list of available homes to authorized clients upon request.
- Automate all software maintenance tasks (provision, patch and upgrade) for database homes, and their associated databases, and grid infrastructure homes.
- Provides the ability to report on existing deployments.

More information on Rapid Home Provisioning is available in Oracle Clusterware Administration and Deployment Guide, 12c Release 1 (12.1)

Conclusion

Oracle Database 12c adds value at every phase of the journey to the Enterprise Cloud, and provides the most efficient platform for consolidation of standardized database services.

The multitenant architecture delivers the highest consolidation density without the overhead of changing application code and with improved security access controls. Standardization is easier than it has ever been.

Pluggable Databases provide improved business agility; new capabilities in migration and cloning make operations faster and simpler. Provisioning tasks can be streamlined.

Ensuring application administrators and operators are granted only necessary privilege to perform their tasks can now be validated through runtime reports. All native database security features can be established at the PDB level.

Oracle Database 12*c* is an ideal platform for consolidation and service delivery that can streamline your business operations. Consolidating on to Oracle Real Application Clusters for the flexibility and availability features that Oracle RAC offers, and then combining this with the new features of the database, and those offered by Oracle Enterprise Manager 12*c* Cloud Control, means you can accelerate your journey to realizing the full benefits of cloud computing.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/oracle

oracle.com

Integrated Cloud Applications & Platform Services

Consolidation Best Practices
April 2017
Author: Troy Anthony
Contributing Authors: : Bernard Clouse, Michael Timpanaro-Perrotta

Oracle is committed to developing practices and products that help protect the environment