An Oracle White Paper
March 2012

Best Practices for Database Consolidation in Private Clouds

Executive Overview

Today's business world is an ever-changing environment of increasing complexity and challenges, requiring corporations to build agility and flexibility into their IT infrastructure to swiftly adapt to changes in the marketplace. IT organization's key initiatives include consolidation of systems, standardization of business processes, the move to shared services, and corporate compliance. Consolidation is one of the major strategies that organizations are pursuing to achieve greater efficiencies in their operations. Consolidation allows organizations to increase the utilization of IT resources, which lowers costs since fewer resources are required to achieve the same outcome. Operational costs are also reduced since fewer components and objects need to monitored, managed, and maintained.

In the "Database Consolidation onto Private Clouds" White Paper we describe the different database cloud models and application workloads that work best for each model, as well as the different phases that need to be executed to successfully consolidate applications. In this paper we will describe the system resources that require special sizing considerations, define workloads that should be consolidated, and address tenant isolation. We will focus on the Database Cloud models: Database Consolidation and Schema Consolidation.

Business Drivers for Consolidating Databases onto Private Cloud

Four key business drivers typically motivate database consolidation onto a private cloud.

Reduced Cost

IT budgets are under constant scrutiny, so IT departments need solutions that reduce both capital expense and operating expense without compromising key business requirements. Consolidating onto shared resources effectively replaces siloed, underutilized infrastructures with a shared resource pool, which lowers overall costs and increases resource utilization. Capital expenditure can be reduced beyond simply shrinking server footprint by creating a higher density of databases per server through multi-tenancy configurations. Operational expenditure can be reduced by improving efficiency through automation, improved management productivity and fewer elements to manage.

Reduced Complexity

IT departments can simplify their environments by reducing the number of supported configurations and services through rationalization, standardization, and consolidation. By standardizing on a common set of building blocks, IT departments can easily deploy predefined configurations and scale-out using modular components. One of the keys to reducing complexity is centralized management: as the environment becomes more homogenous, it becomes easier to manage. And having a central management hub keeps operational costs low and further promotes the automation of routine tasks.

Increased Quality of Service

IT departments are not only trying to drive down costs, they are also looking for solutions that will improve performance, availability, and security. In a private cloud, database performance can be monitored and managed via shared Centers of Excellence. Databases also benefit from the high availability built into the private cloud. And consolidation helps enforce a unified identity and security infrastructure as part of standardized provisioning and management processes.

Improved Agility

IT departments are increasingly looking to develop more agile and flexible environments that will enable faster time to market and rapid responses to changing business requirements. This provides efficient rollout of new business strategies as well as the capability to quickly deploy applications without a huge lag time due to infrastructure setup. The key aspects of agility are:

- **Fast deployment.** Building a private cloud infrastructure using standardized hardware components, software configurations and tools enables an automated and simplified deployment process.
- **Rapid provisioning.** Resources in a cloud can be rapidly provisioned, often via self-service, providing quicker application go-to-market. This reduces overall time in deploying production applications, development platforms, and creating test bed configurations.
- **Resource elasticity.** The ability to grow and shrink the capacity of any database, both in terms of size and compute power, offers applications the flexibility to meet the dynamic nature of business workloads.

Initial Cloud Pool sizing

Common questions when building a Private Cloud are: "What should be the size of the Cloud Pool?" and, "How many databases or applications can be consolidated?"

The initial Cloud Pool size will depend on the type of applications housed and their capacity requirements. Note, in most cases you will not be building a single "gig-normous" Private Cloud, but rather many Cloud Pools.

To avoid the overloaded term "Server Pool", we use the term Cloud Pool to refer to a collection of servers that have shared storage and that share a private network. Thus a Cloud Pool can be a RAC cluster or virtualization pool. Furthermore, we define a Private Cloud as an aggregation of Cloud Pools.

Cloud Pools should be built and aligned with the following requirements:

- **Business**
    - Build separate Cloud Pools for lines of business (LOB) or departments
    - Create separate Cloud Pools for different application service levels or governance compliance
- **Functional**
    - Build a Cloud Pool for similar functioning applications; e.g., Internal vs. External facing applications
- **Technical**
    - Separate Cloud Pools based on OS type, database version, or isolation requirements
    - Cloud Pools for applications with complementary workloads
    - Cloud Pools built around specific High Availability goals

Most customers we have worked with have created multiple Cloud Pools. These Cloud Pools are built for specific configurations and support specific business requirements. For example, customers can build a Cloud Pool for their 11gR1 databases, one for their compliance regulated data (PCI, PII, HIPAA, etc.), and another for their business critical 11gR2 applications. It is a best practice for applications with similar SLA requirements to co-exist in a consolidated environment; i.e., do not mix mission critical applications with non- critical applications in the same Cloud Pool.

The number of applications that can be consolidated will depend on size, resource consumption, and SLAs of the applications that will be consolidated. Furthermore a pre-defined threshold of the system resource usage will also dictate how much can be consolidated. The Cloud Pool Capacity Management section covers this topic.

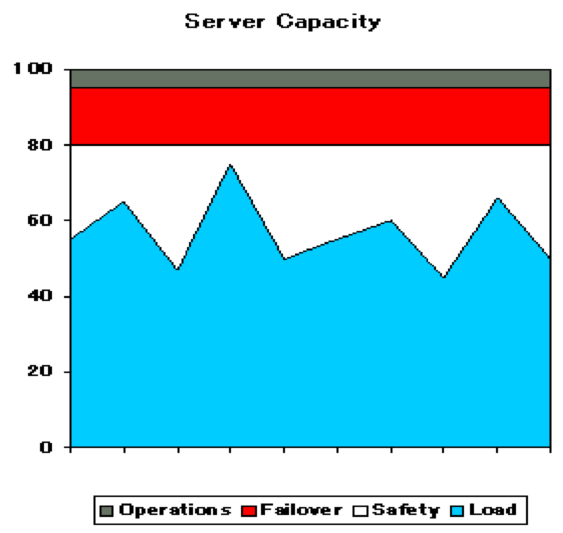
It is recommended to build a Cloud Pool based on standardized modular building blocks; e.g., many customers will standardize on a four node cluster consisting of dual sockets, with 8 cores per socket. Oracle recommends a minimum of three servers in a Cloud Pool, as this will tolerate the failure of one or two servers and protects from server failure during planned outages, such as rolling upgrades scenarios. However there should be careful consideration for CPU, memory and storage configurations.

Sizing applications that will be placed in the Private Cloud depends on if these are new or existing applications. For new applications, initial sizing of the pool is the same as any current capacity planning exercise. For existing applications use tools such as AWR reports, *vmstat* and *iostat* to determine consumption. EM 12c Consolidation Planner provides a streamlined approach for gathering this dataset.

The next section will describe considerations when configuring physical resources for the Private Cloud.

CPU

When doing the initial sizing for CPU, determine what applications will be housed and scope for this base, include 10% overhead for operational tasks, such as backups, *crontab* jobs, etc., and 15% overhead to account for failover of workload. This generally leaves the CPU capacity at 75%. Oracle recommends that operating nodes in a Cloud Pool at 75% CPU capacity provides a good balance between system usage and headroom.



A database that gets provisioned in the Cloud Pool should get a default minimum of two CPUs. This can be enforced using the CPU_COUNT init.ora parameter or using the Instance Caging feature (for which also you set the CPU_COUNT). There is a difference between implementing Instance Caging and setting CPU_COUNT:

- Setting CPU_COUNT simply limits the number CPUs that the database "sees" and implicitly sets internal SGA memory structures as well as database background process settings. There are numerous internal and external parameters that are derived from this CPU_COUNT setting; e.g., dbwr_processes or library_cache_latches. However, CPU_COUNT does not enforce any resource limits.
- Instance Caging leverages CPU_COUNT as well as Database Resource Manager (DBRM) to enforce resource usage limits. For example, if a database instance is using Instance Caging with CPU_COUNT set to four, then no more than four active user processes will consume CPU at the same time. Note Instance Caging does not enforce resource limits on background processes (DBWR, LGWR, PMON, etc.). Oracle strongly recommends using Instance Caging in Cloud configurations. The following paper discusses Instance Caging is greater detail: http://www.oracle.com/technetwork/database/focus-areas/performance/instance-caging-wp-166854.pdf

Partitioning vs. 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. There are two approaches to determining the maximum number of CPUs, partitioning or over-provisioning.

Partitioned CPU Approach

The Partitioned CPU approach ensures that the aggregate CPU_COUNT setting across all database instances on a given node does not exceed the number of CPUs. The partitioned approach is therefore recommended for mission-critical databases.

With this configuration there should be no contention for CPU resources between database instances. However, if the database instances do not consistently use their CPU resources, then these resources cannot be utilized by the other database instances, making it inefficient from a utilization perspective.

The following recommendation is for the Partitioned CPU approach. This ensures that CPU resources are available for other critical clusterware resources and processes, such as ASM, CSS, CRS agents, etc.

Sum (CPU_COUNT) < 75% x Total CPUs

Over-provisioning approach

With the Over-provisioning approach, a.k.a. over-commitment, the sum of the CPU_COUNT across all instances on a node can exceed the number of CPUs. The advantage of the over-provisioning approach is better resource utilization; however, there can be potential CPU contention if all databases are heavily loaded at the same time, which will degrade database performance. This approach should be deployed for test/development, non-critical or any databases that do not have stringent SLA requirements. Furthermore, databases that consume heavy I/O resources or are highly transactional should not be consolidated into an over-provisioned Cloud Pool.

For this model, use the following recommendation:

Limit the sum of CPU_COUNTs to two times the number of CPUs.
    Sum (CPU_COUNT) <= up to 2 x Total CPUs

The above calculation is based on hyper-threading of two CPU-threads per core, typical in x86 systems. Care should be taken when over-provisioning CPUs on Sun T2 or T4 systems, where CPU-thread ratio can range from 8 to 64 threads.

Note: the extreme capabilities of Oracle Exadata Database Machine allow for even higher levels of CPU over-provisioning. Please see Best Practices For Database Consolidation on Exadata Database Machine for more details.

Additionally, limit the Parallel Max Query Servers to be <= 20 x total CPU (or threads)

It is important to note that Oracle determines the number of CPUs on the system based on the value returned back from the OS. In cases where the server has enabled hyper-threading (HT), the value returned could be threads and not actual cores. Threads are not real CPUs and thus over-provisioning CPUs based on thread count may result in inefficient and unpredictable results when workloads peak at the same time.

RAC databases, as well as ASM, use a real-time (RT) mode process called LMS for cache fusion communication. The number of LMS processes started by an instance is dependent on the CPU_COUNT value. Since these LMS processes execute in real-time (RT) mode, excessive LMS processes can cause performance issues when over-provisioning CPU resources. There is a general guideline that limits the number of LMS RT processes per server to one less than the number of cores on the server. This guideline can limit the number of databases consolidated. There is a new feature introduced in 11.2.0.3 that will automatically lower these LMS processes to standard time sharing (TS) mode. Please note this feature is only applicable for 11.2.0.3 RAC and ASM instances. Cloud DBAs should review MOS Note 1392248.1 when managing LMS processes in consolidated environments.

Memory

Configuring memory can be more straightforward in the Cloud Pool than CPU sizing. Note, unlike for CPU configuration, it is highly discouraged to 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, Server consolidation and Database Consolidation Models do not make the most efficient use of memory, as each database instance will consume its own SGA and PGA. Schema 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 gross over allocation 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 the Private Cloud, application owners should ensure that the database is cleansed 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 database consolidation After all, consolidating databases is essentially the aggregation of IOPS. DBAs should look at average and peak IOPS for each database 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. Mixing non-complementary workloads leads to missed SLAs, outages and a poor consolidation deployment. When consolidating workloads, ensure that the consolidated workloads' peak CPU usage does not overwhelmingly exceed the average CPU usage; i.e., the gap between peak and average CPU usage should be minimal. Satisfying this goal means that the CPUs are being utilized as fully as possible.

- When complementary workloads are combined, the average overall load increases more than the peak. See Figure1.
- When optimally complementary workloads are combined, only the average increases; the peak remains unchanged.
- In antagonistic workloads, the peak increases more than the average; i.e., during low usage periods, CPUs are underutilized but must be kept available for the peak loads. See Figure 2.

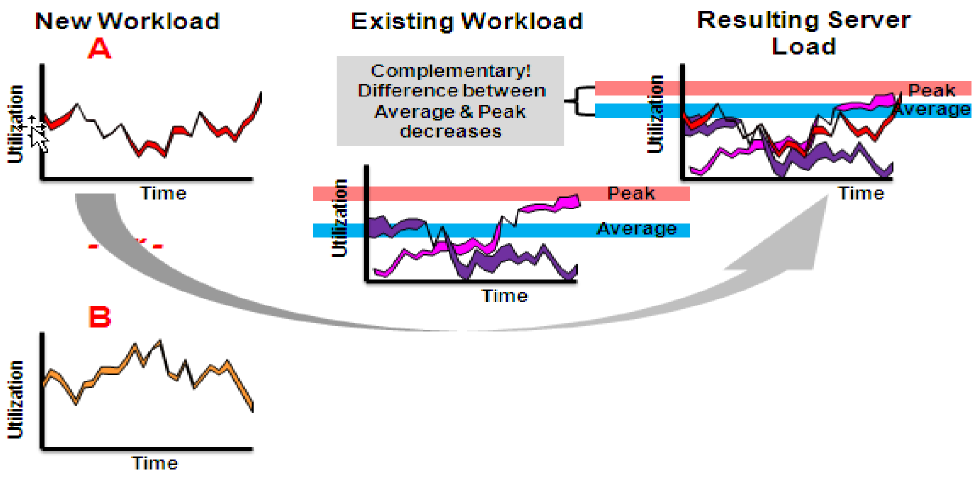


Figure1

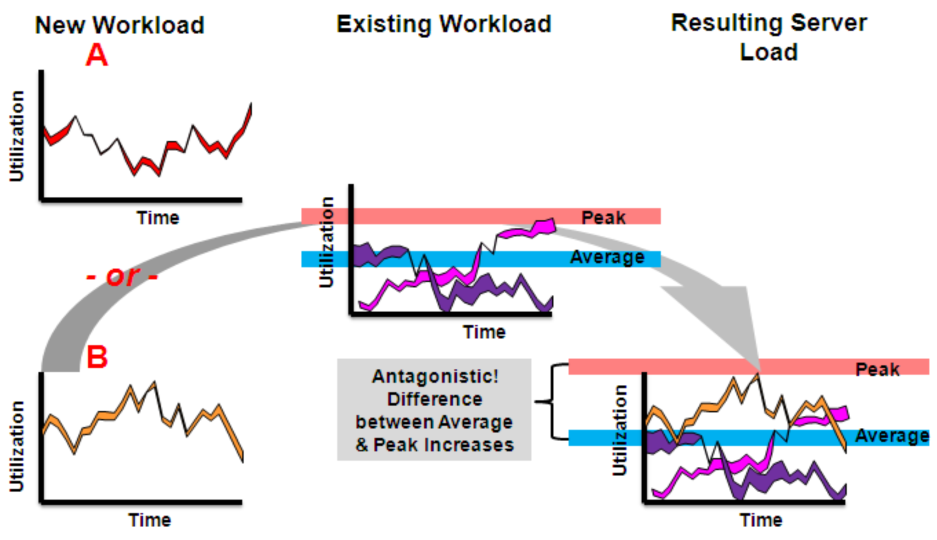


Figure 2

Cloud Pool Capacity Management

When sizing the Cloud Pool you must also pre-establish the rules or thresholds for Cloud Pool fullness; i.e., how will you determine when a Cloud Pool is full. Two recommended and generally adopted approaches are: CPU usage or the number of databases in the Cloud Pool. A customer we worked with had a defined threshold of 40 databases per Cloud Pool; whereas another customer stated a 75% CPU average usage was their threshold. Once these thresholds are reached there are different approaches for growth:

One approach is to add additional nodes to the pool, or additional resources to the existing nodes; e.g., add more memory or CPU. This approach adds additional capacity to the existing Cloud Pool. Another approach is to build out a completely new Cloud Pool and begin the consolidation effort into the new Cloud Pool, and this step is repeated each time a Cloud Pool is filled. The latter method is best if you have an ongoing effort to consolidate, and it simplifies management as each Cloud Pool becomes a "POD" of consolidated databases.

Operating System Parameters

When implementing Database Consolidation, it is important to define how many potential databases will be consolidated. This will drive the appropriate setting of many OS and database parameters required to support the environments. These parameters include:

- Hugepages
    - Hugepages is generally required if 'PageTables' in /proc/meminfo is > 2% of physical memory
    - If used should equal the sum of shared memory from all databases
    - See MOS note 401749.1 for precise computations;
- Shared Memory settings
    - Shared Memory Identifiers - set greater than the number of potential databases that will consolidated
    - Size of Shared Memory Segments – OS setting for max size = 85% of physical memory
    - Number of system semaphores – must be larger than the sum of processes.
    - Maximum semaphores in a semaphore set should be the largest value for the processes parameter in any of the databases.
- ASM instance parameters
    - Size ASM Processes parameters based the on expected number of databases that will be managed by ASM on that instance and keep values consistent across instances

      50 x MIN (# instances on db node+ 1, 11) + 10 x MAX (# instances on db node – 10, 0)
    - Size memory_target (or explicitly define shared pool) to accommodate the large Processes setting

      MEMORY_TARGET should be increased by 12MB per 100 PROCESSES increment when the parameter PROCESSES is set to be more than 600

Isolation

Isolation requirements can influence the method or degree of consolidation possible. Whether you consolidate multiple application schemas in a single database, or host multiple databases on a single platform, or some combination of both approaches depends on the level of isolation the system demands. Isolation can be categorized into four areas: Fault, Resource, Security, and Operational. Each Cloud model deals with isolation slightly differently, using OS and/or Database built-in capabilities, often in combination with advanced features or products to provide a more complete solution, commensurate with the risk.

The section will focus on tenant isolation in the Private Database Cloud models.

Database Consolidation

Before building out a Database Consolidation environment it is important to define what applications will be hosted in the Cloud Pool; i.e. production, non-critical production, or test/dev. Applications of differing criticality or SLA should not be consolidated together

Fault Isolation

With Database Consolidation the multi-tenancy granularity is the database, thus each database (and each database instance) is isolated from the other databases in the Cloud Pool. Although all the databases may run from the same Oracle Home installation, database faults are generally isolated to a failing instance; i.e., fault isolation is maintained by fencing off the offending instance. For example, if a database instance becomes unresponsive, then one of the neighboring node's LMS process will request CSS to perform a "member kill" operation against the offending instance. In rare cases, where the unresponsive instance cannot be killed, an invasive operation such as node reboot is invoked; in these rare cases other database instances will be affected. However, proper application design and implementation of documented best practices can limit the impact of instance or node failure. For example, the use of dynamic services, such as Fast Application Notification (FAN) and connection pools, provide a faster mechanism for event notification (node or database instance down events) to the application. This allows the application to reconnect more quickly, minimizing the overall impact of the outage.

Operational Isolation

Operational Isolation ensures that any management or maintenance performed on a database or its operating environment does not affect other running databases in the pool, including startup/shutdown of instances, patching and backup/recovery.

Startup/Shutdown

In most consolidation configurations, the number of Oracle Homes is kept to a minimum, and typically one Oracle Home is used for all the consolidated databases. To provide operational isolation, create named users for each Cloud DBA for the database, and add those users to the password file (needs REMOTE_LOGIN_PASSWORDFILE set to EXCLUSIVE). Then grant SYSDBA privileges to those named users. By having different password files for each database, users can gain SYSDBA privileges only to their database. To perform operational functions, such as startup or shutdown, the Cloud DBA should connect to the appropriate database user with the SYSDBA privilege. Typically this is performed via EM, therefore the necessary database credentials need to be established in EM.

Patching

Patching databases in a consolidated environment involves two tasks, logistics (planning for the patch application) and the actual applying of the patch. When building Private Cloud environments, expectations should be set that one-off or unscheduled patching is not efficient and should be discouraged, as the patching may affect many databases. A schedule for patch application should be pre-defined and acknowledged by all participating tenants. For example, a schedule for Oracle Patch Set Updates (PSU) should be well defined. One-off patches should be evaluated for priority and relevance with respect to the entire database consolidated community. When patches need to be applied, the most efficient method is to stage the patch, this involves cloning the Oracle Home, apply the patch to the cloned home, and finally switching the Oracle Home. Rolling patch application should be leveraged where possible.

If the patch management logistics across databases is not practical or if sharing of SYSDBA across databases is not desired, then a separate Oracle Home for a group databases can be an alternative. For these cases, each Oracle Home should use distinct username and OSDBA. However, running database instances from different Oracle Homes does increase complexity.

Resource Isolation

Resource isolation deals with the allocation and segregation of system resources. In Database Consolidation the competing resources include CPU, memory and I/O (storage capacity as well as IOPs).

- Memory – Appropriate memory_target and memory_ max_target values need to be set on a per instance basis for each node, and must be maintained consistently across all instances of the same database. Note that memory_target does not enforce a hard ceiling for PGA values. It is highly discouraged to over-commit.
- CPU – Like memory settings, the CPU values should be set appropriately. This can be done by setting CPU_COUNT to a specific value or enabling the 11gR2 Instance Caging feature. The latter is recommended as it also enforces a Database Resource Manager (DBRM) resource plan providing better control of CPU consumption. Follow the best practices outlined in the CPU section.

Security Isolation

In any consolidation environment, a security implementation should use the "least privileges" approach to harden the environment. In most cases a single Oracle Home will be used by all database instances on a given node. If several databases are sharing the same Oracle Home, then any user that is part of the OSDBA group for that Oracle Home will have SYSDBA access to all database instances running from that home. This is a good approach to reduce manageability overhead; however, it does open security issues.

It is recommended to implement the following best practices for this type of configuration:

- Minimize access to the database server; i.e., Sql*Net Pipe only access
- Use named user accounts for DBAs with sudo access for privileged commands
- Implement and enable Database Vault to provide role separation to control user data access
    - Database Vault using Realms should be used to protect application and schema data from unauthorized access
    - Upon provisioning the database, the Security Admin should enable Realms for each application
    - Use Pre-defined Data Vault Realms for E-Business Suite, Siebel and PeopleSoft
    - Encrypt data where necessary

Schema Consolidation

In this deployment model, the consolidated database essentially consists of one or more application schemas running across one or more servers in a Cloud Pool. Customers who we have worked with in deploying this model typically consolidate 15-20 applications (schemas). In the Schema Consolidation model the tenancy granularity is the schema. Therefore the schema needs to have proper isolation.

Although Schema Consolidation provides the highest level of consolidation, careful consideration and planning must be done to ensure that the consolidated applications can co-exist. For example, check for schema namespace collisions and confirm certification of packaged applications to run in these consolidation configurations.

Fault Isolation

In this model, an application fault in one schema will not cause other applications to fail. However, login storms or improperly configured application/mid-tiers can impact other consolidated applications. To minimize the impact of login storms, configure mid-tier connection pools appropriately. In some cases login-rate limiters can be defined for the Oracle Listener. Poorly written database resident code, such as PL/SQL, can also affect other unrelated applications. A through testing of the application as well as code review is necessary to prevent application faults.

Operational Isolation

Operational isolation for this model includes minimizing the impact of the recovery/restore of lost data or patch 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. Typically for Schema Consolidation, nightly backups, as well as Data Pump exports of the schema are needed. If data is lost or deleted, then features such as Flashback Table, Flashback Query, or Flashback Transaction should be used to provide the least invasive approach for restore.
- Patching – The patching issues in Schema Consolidation are similar to those in Database Consolidation.

Resource Isolation

With multiple applications contending for the same database and systems resources, resource management is a necessity for Schema Consolidation. Oracle Database Resource Profile Limits provide basic “knobs” to control consumption, and can be supplemented with Oracle Database Resource Manager along with Oracle QoS. Applications can be grouped in consumer groups with appropriate resource plan directive mapping with a resource plan. This will specify how CPU, I/O, and parallel server resources are to be allocated to the consumer group.

- Storage Capacity – Cloud DBAs can cap the storage consumed by applications using tablespace quota. There should be close monitoring of the tablespace space usage by each schema so that growth patterns and thresholds can be managed.

Security Isolation

Security isolation between schemas is one of the most important aspects of Cloud management. Out of the box security Oracle database profiles can be used to limit access to data. However, many times deeper security measures and policies must be put in place. This may include protecting data at rest, granular access control, as well as security auditing. For these cases encryption should be implemented where necessary, via Advanced Security, Realm based access control, via Database Vault, and Audit Vault for runtime audit management. The following are also security best practices for Schema Consolidation

- Only Cloud DBA should have SYSDBA, SYSOPER and SYSASM access
- Guest DBAs only have schema level access and V$ view access
- Ensure use of private synonyms
- Use strong database user passwords
- Set appropriate values for PASSWORD_LOCK_TIME and FAILED_LOGIN_ATTEMPTS

Summary

Deploying databases with private clouds is a proven model for the delivery of database services. Consolidation onto shared resources in a private cloud enables IT departments to improve quality of service levels—as measured in terms of database performance, availability, and data security—and reduce capital and operating costs. Private clouds consolidate servers, storage, operating systems, databases, and mixed workloads (schemas) onto a shared hardware and software infrastructure. The higher the consolidation density achieved, the greater the return on investment. Oracle's complete technology stack of hardware and software, showcased in Oracle's Engineered Systems, offers unique capabilities for all levels of consolidation.

This paper described the best practices for consolidating Oracle Databases in the Database Consolidation and Schema Consolidation private cloud deployment models. Cloud Pool sizing, selection of workloads to consolidate, resource management and tenant isolation were discussed in detail, preparing you for an efficient and effective consolidation plan. Further details are available for planning database consolidation on Oracle Database Appliance and Oracle Exadata Database Machine.

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

Best Practices for Database Consolidation in Private Clouds
March 2012
Author: Nitin Vengurlekar
Contributing Authors: Burt Clouse, Raj K. Kammend

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Hardware and Software, Engineered to Work Together