Database Consolidation

# Overview

## Why consolidate

* To minimize idle resources
* To maximize efficiency
* To lower costs

## Consolidation examples:

* Hosting multiple virtual servers on one physical server
* Hosting multiple database instances on one server
* Supporting different users on one database instance

## Core Principles for Database Consolidation

* Consolidate databases with similar availability and planned maintenance objectives.
* Other considerations include performance, system requirements, security, and organization boundaries.

# Different Consolidation Types

## Database:

* Multiple databases consolidated on to a single platform
* Separate instances for each database
* Separate database management
* Platform resources are shared by database instances

## Application Schema:

* Multiple application schemas consolidated into a single database
* One database instance (or one set of Real Application Clusters [RAC] instances)
* One database to manage

## Virtual Machine:

* Multiple virtual machines consolidated on to a single platform
* One or more databases hosted on each virtual machine
* Platform resources are shared by multiple operating system and database instances
* Separate operating system and database management

## Oracle Multitenant Architecture:

* New database architecture in Oracle Database 12c
* Multiple pluggable databases consolidated into a single container database
* One container database and instance set to manage
* Complete logical separation between pluggable databases

# Recommendation

## Cluster Configuration Options

* Use one cluster per hardware pool.
* Use database services to manage database workloads across the available database servers.
* Use Oracle Data Guard to protect against cluster failure for critical hardware pools.

## Operating System Parameter

* Set the number of shared memory segments (SHMMNI) greater than the number of databases.
* Set the maximum shared memory segment size (SHMMAX) to 85% of physical memory size.
* Set the maximum total number of system semaphores (SEMMNS) greater than the sum of all database processes.
* Set the maximum number of semaphores in a semaphore set (SEMMSL) greater than the largest number of processes in any single database.
* For Linux, if PageTables in /proc/meminfo is > 2% of physical memory size, consider setting HugePages equal to the sum of all shared memory segments

## Database Memory

* Calculate the Oracle Database memory usage as follows:
  + For OLTP: sum((PGA\_AGGREGATE\_TARGET + SGA\_TARGET) + (4 MB \* Maximum Processes))
  + For data warehousing using Oracle Database 11g: sum((3 x PGA\_AGGREGATE\_TARGET) + SGA\_TARGET)
  + For data warehousing using Oracle Database 12c: sum(PGA\_AGGREGATE\_LIMIT + SGA\_TARGET)
* Ensure that the Oracle Database memory usage does not exceed:
  + The database server memory size for non-critical hardware pools 75% of the database server memory size for critical hardware pools
  + Monitor memory usage and maintain a free memory target of at least 5% at all times.

## CPU Management

* Use instance caging to prevent individual databases from consuming too much CPU.
* Recommended CPU\_COUNT parameter settings:
  + For all databases: CPU\_COUNT >= 2
  + For critical pools hardware pools: sum(CPU\_COUNT) <= the total number of CPU cores
  + Otherwise: sum(CPU\_COUNT) <= three times the total number of CPU cores
* Use the database resource manager to control CPU allocation within each database.
  + Use the MAX\_UTILIZATION\_LIMIT directive attribute to limit the CPU utilization for consumer groups.

## Process Management

* Adjust PROCESSES for ASM:
  + For 2 to 10 databases: PROCESSES = 50 + (50 \* number of databases)
  + For 11 or more databases: PROCESSES = 450 + (10 \* number of databases)
* Limit PARALLEL\_MAX\_SERVERS:
  + For 2-socket servers: sum(PARALLEL\_MAX\_SERVERS) <= 240
  + For 8-socket servers: sum(PARALLEL\_MAX\_SERVERS) <= 1280
* Limit Data Guard redo apply parallelism to 16 or less.
* Process and connection guidelines:
  + Limit active processes:
    - For average applications: Active processes <= 5 x CPU\_COUNT
    - For very CPU-intensive applications: Active processes <= CPU\_COUNT
  + Use connection pools to manage application connections.
  + Use database shared server processes to reduce the process count.
  + Use the connection rate limiter to prevent connection storms.

## Other Recommendations

* Configure I/O Resource Management (IORM).
* Set DB\_RECOVERY\_FILE\_DEST\_SIZE:
  + sum(DB\_RECOVERY\_FILE\_DEST\_SIZE) < usable FRA size

## Schema Consolidation

* Isolate application schemas into their own tablespaces:
  + Simplifies space management
  + Enables fast and granular recovery using tablespace point-in-time recovery (TSPITR)
* Tune tablespace backup, restore, and recovery practices to minimize disruption between schemas.
  + Understand the overheads and requirements of different approaches:
    - TSPITR
    - Flashback technologies
    - Logical export and import
    - Others
  + Consider using image copies for faster TSPITR.