Oracle Database

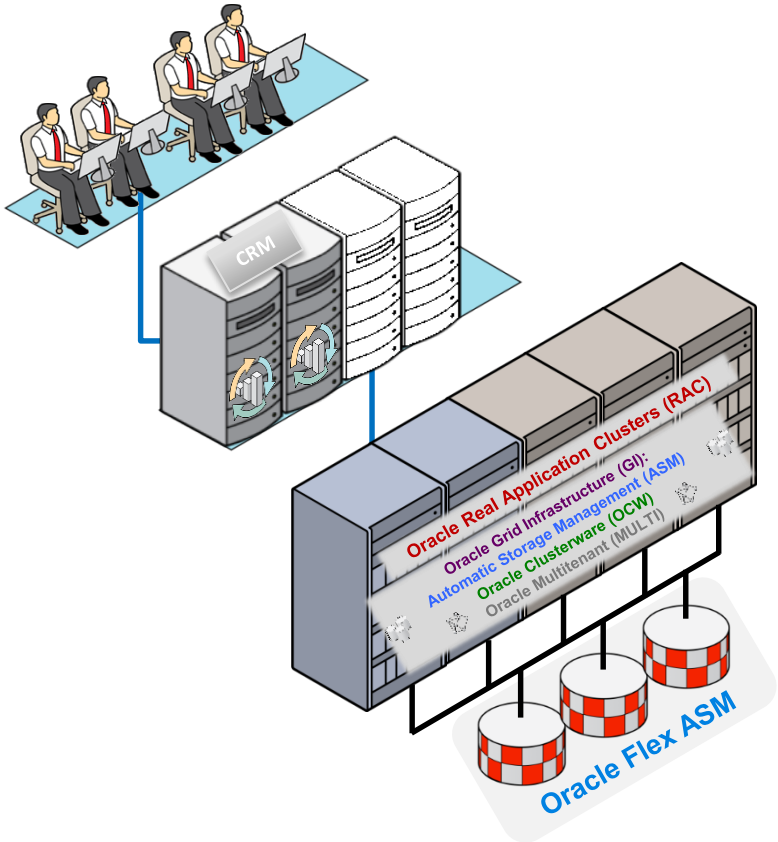
Architecture and New Features

Martin Millstam
Senior Sales Consultant

Topics

- Oracle RAC architecture
  - Clusterware
  - ASM
  - RAC
- Multitenant
- Application Continuity

The New Oracle RAC 12*c*



Oracle RAC 12c provides:

1. Better Business Continuity and High Availability (HA)
2. Agility and Scalability
3. Cost-effective Workload Management

Using

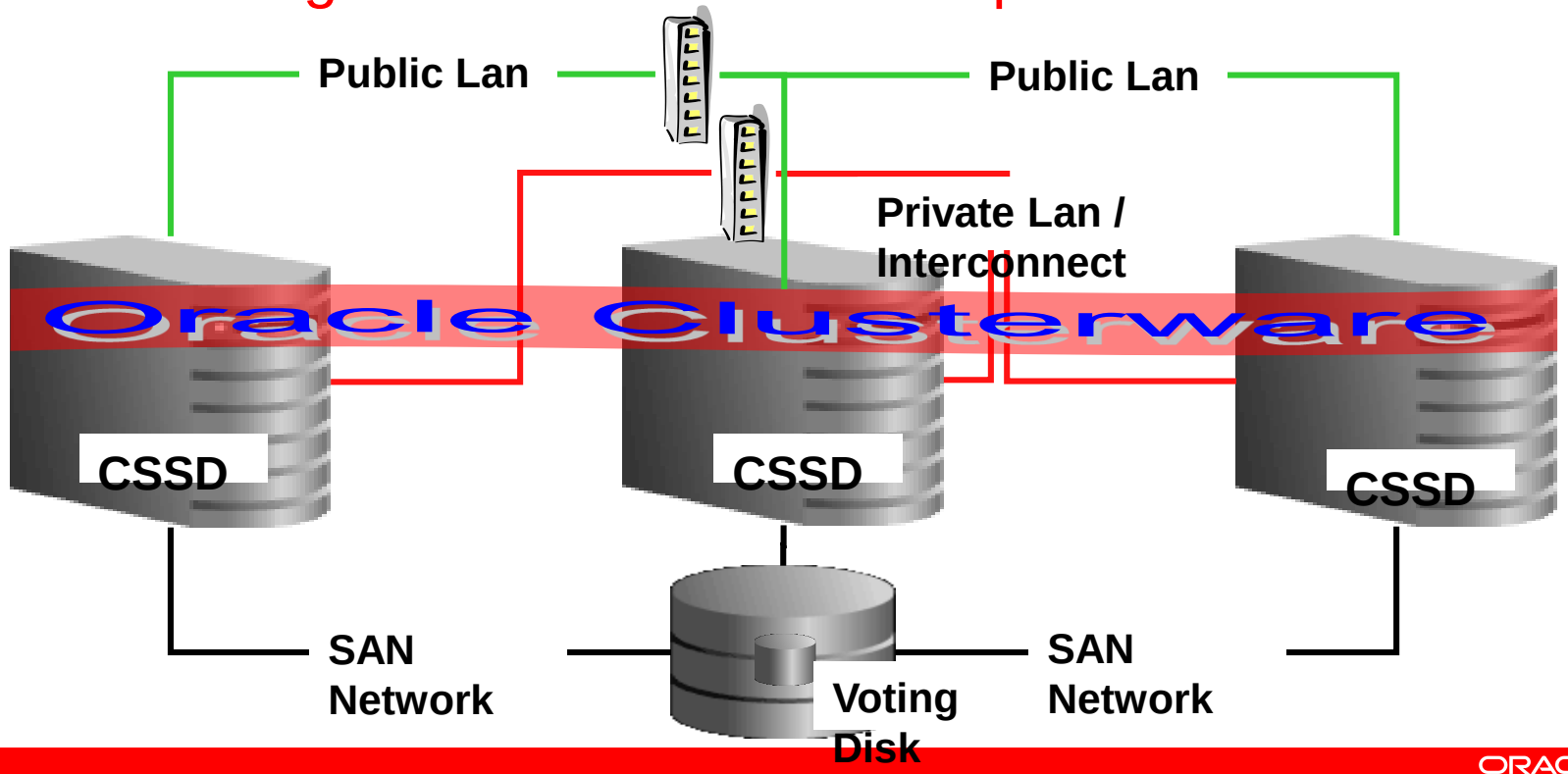
- A standardized and improved deployment and management
- A familiar and matured HA stack

Clusterware

- What is it?
- What does it do?

Basic Hardware Layout Oracle Clusterware

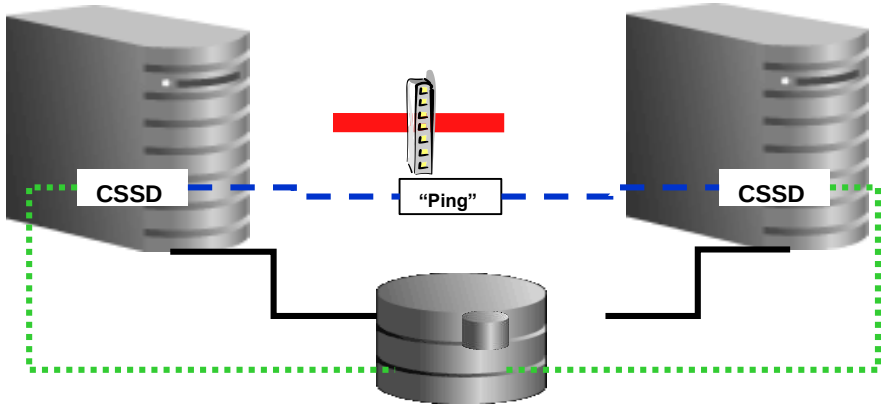
Node management is hardware independent



What does CSSD do?

CSSD monitors and evicts nodes

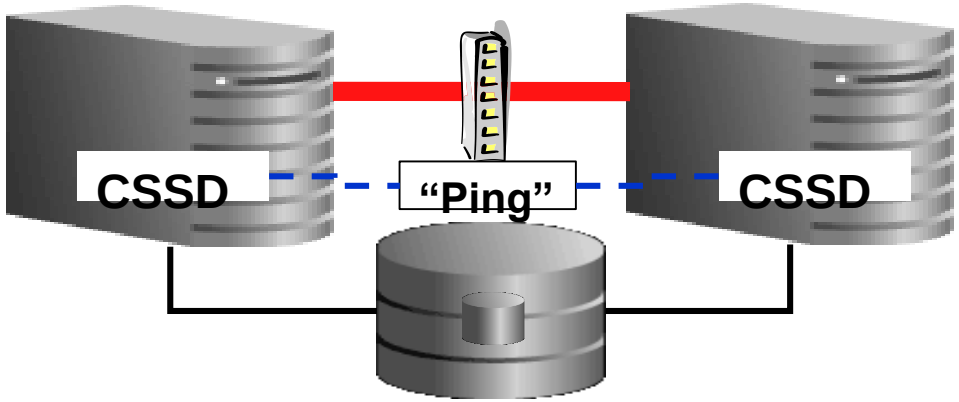
- Monitors nodes using 2 communication channels:
  - Private Interconnect ⇔ Network Heartbeat
  - Voting Disk based communication ⇔ Disk Heartbeat
- Evicts (forcibly removes nodes from a cluster) nodes dependent on heartbeat feedback (failures)



Network Heartbeat

Interconnect basics

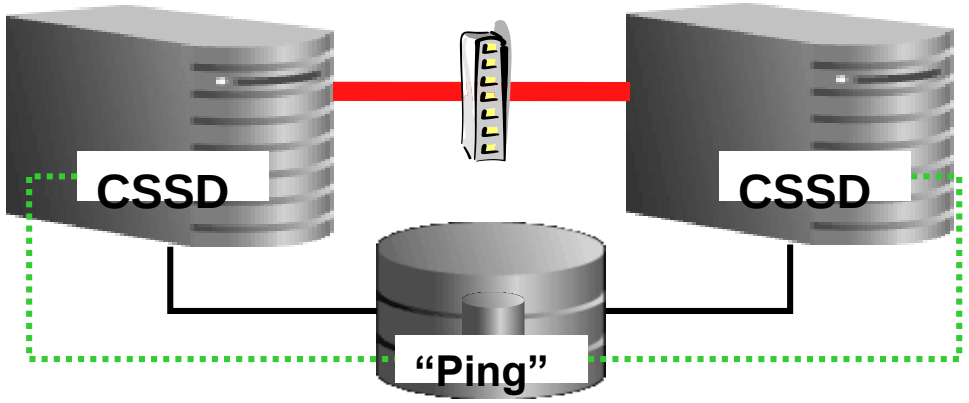
- Each node in the cluster is “pinged” every second
- Nodes must respond in css_misscount time (defaults to 30 secs.)
  - Reducing the css_misscount time is generally not supported
- Network heartbeat failures will lead to node evictions
  - CSSD-log: `[date / time] [CSSD][1111902528]clssnm`**`PollingThread:`** `node` *`mynodename`* `(5) at 75% heartbeat fatal, removal in 6.770 seconds`



Disk Heartbeat

Voting Disk basics

- Each node in the cluster “pings” (r/w) the Voting Disk(s) every second
- Nodes must receive a response in (long / short) diskTimeout time
  - I/O errors indicate clear accessibility problems → timeout is irrelevant
- Disk heartbeat failures will lead to node evictions
  - CSSD-log: … `[CSSD] [1115699552] >TRACE: clssnm`**`ReadDskHeartbeat:`** `node(2) is down. rcfg(1) wrtcnt(1) LATS(63436584) Disk lastSeqNo(1)`

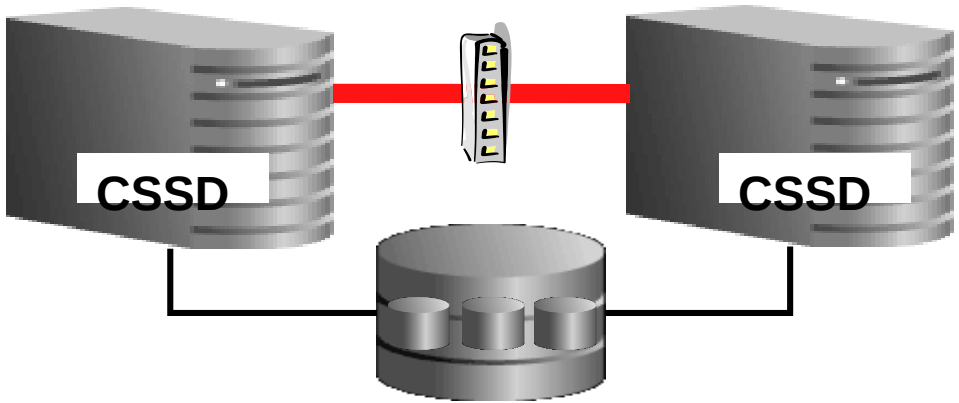


“Simple Majority Rule”

Voting Disk basics

- Oracle supports redundant Voting Disks for disk failure protection
- “Simple Majority Rule” applies:
  - Each node must “see” the simple majority of configured Voting Disks at all times in order *not* to be evicted (to remain in the cluster)

➢ trunc(n/2+1) with *n*=number of voting disks configured and n>=1



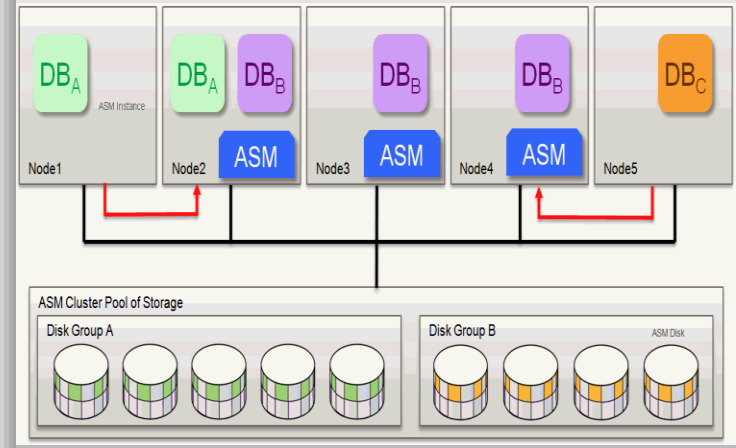
Insertion 2: Voting Disk in Oracle ASM

The way of storing Voting Disks doesn't change its use

```
[GRID]> crsctl query css votedisk
 1.   2 1212f9d6e85c4ff7bf80cc9e3f533cc1 (/dev/sdd5) [DATA]
 2.   2 aafab95f9ef84f03bf6e26adc2a3b0e8 (/dev/sde5) [DATA]
 3.   2 28dd4128f4a74f73bf8653dabd88c737 (/dev/sdd6) [DATA]
Located 3 voting disk(s).
```

- Oracle ASM auto creates 1/3/5 Voting Files
  - Based on Ext/Normal/High redundancy and on Failure Groups in the Disk Group
  - Per default there is one failure group per disk
  - ASM will enforce the required number of disks
  - New failure group type: Quorum Failgroup

Oracle Automatic Storage Management (ASM) 12*c*



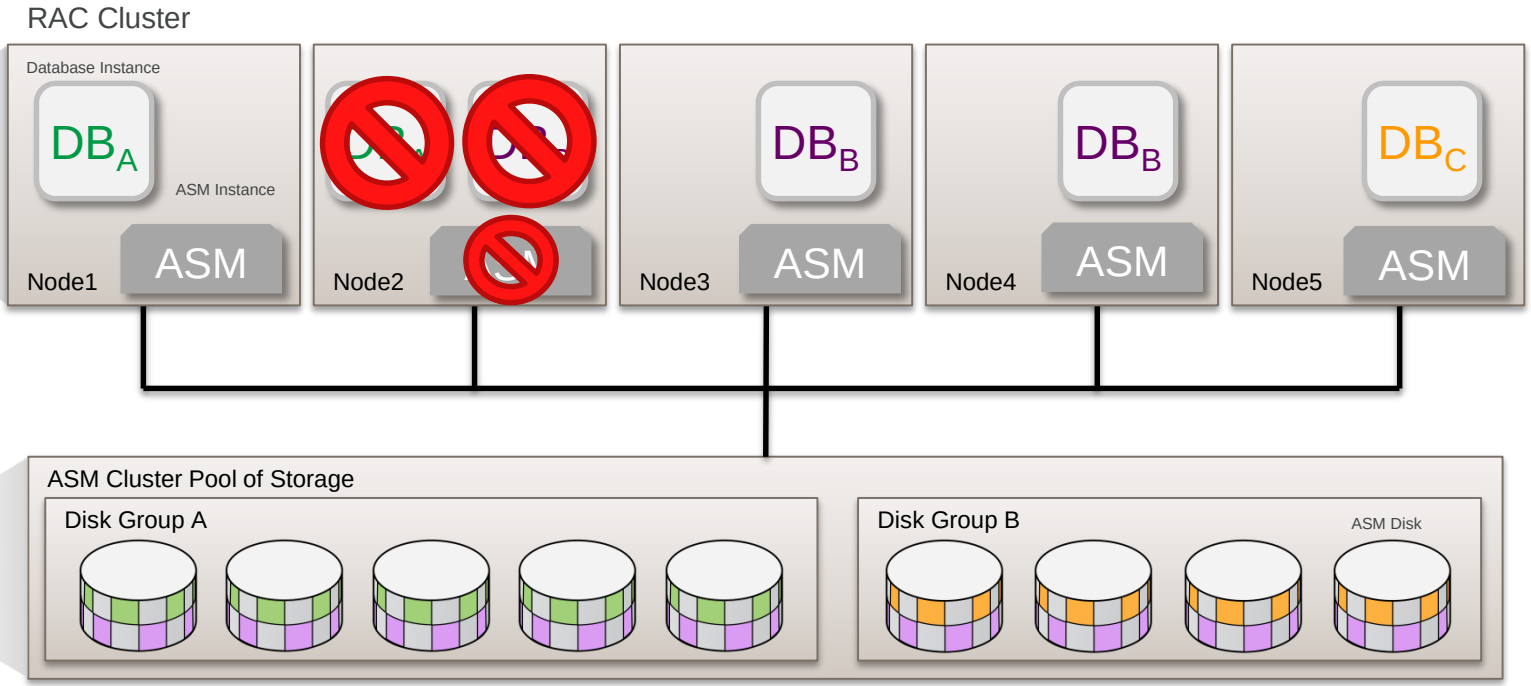
Oracle Automatic Storage Management (ASM)

Oracle Database 11.2 or earlier configuration

One to One Mapping of ASM Instances to Servers

Shared Disk Groups

Wide File Striping



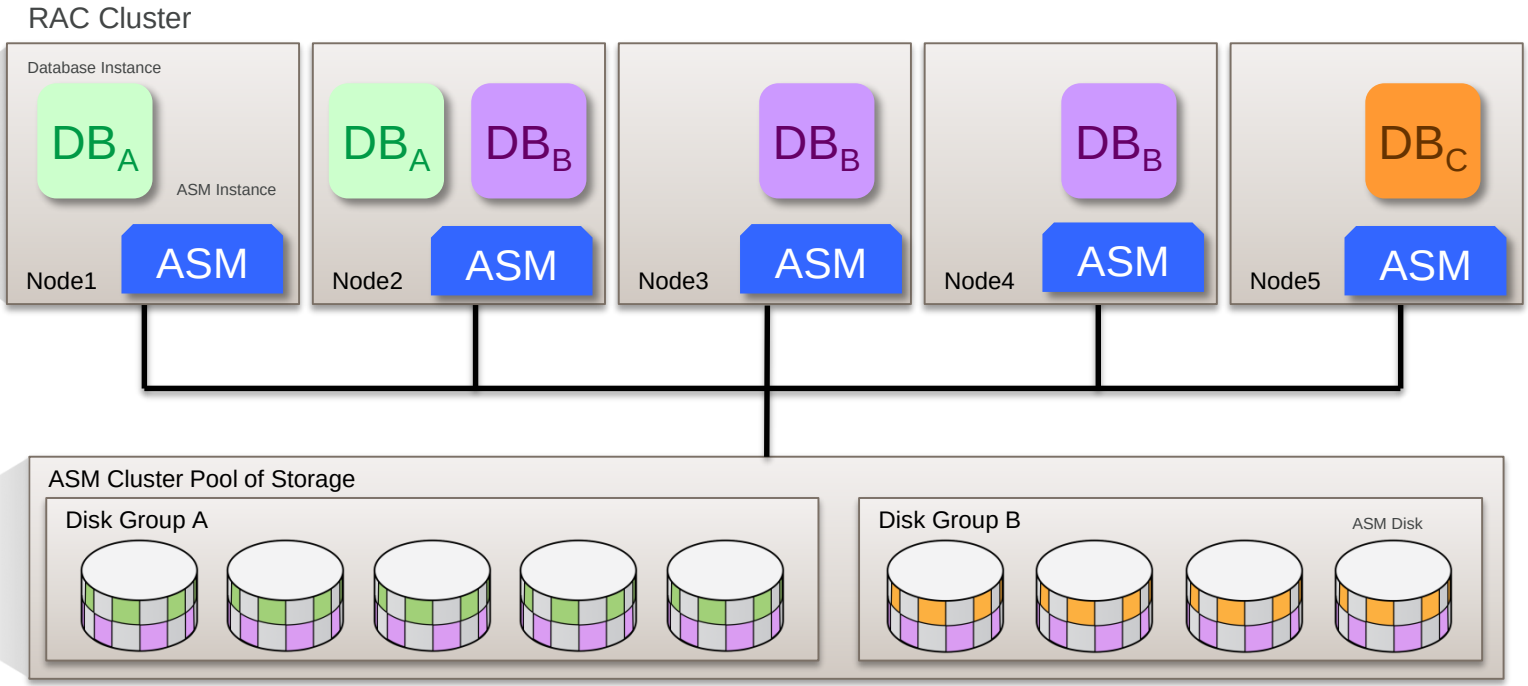
Oracle ASM 12*c* – Overview

Oracle ASM 12*c* Standard Deployment

One to One Mapping of ASM Instances to Servers

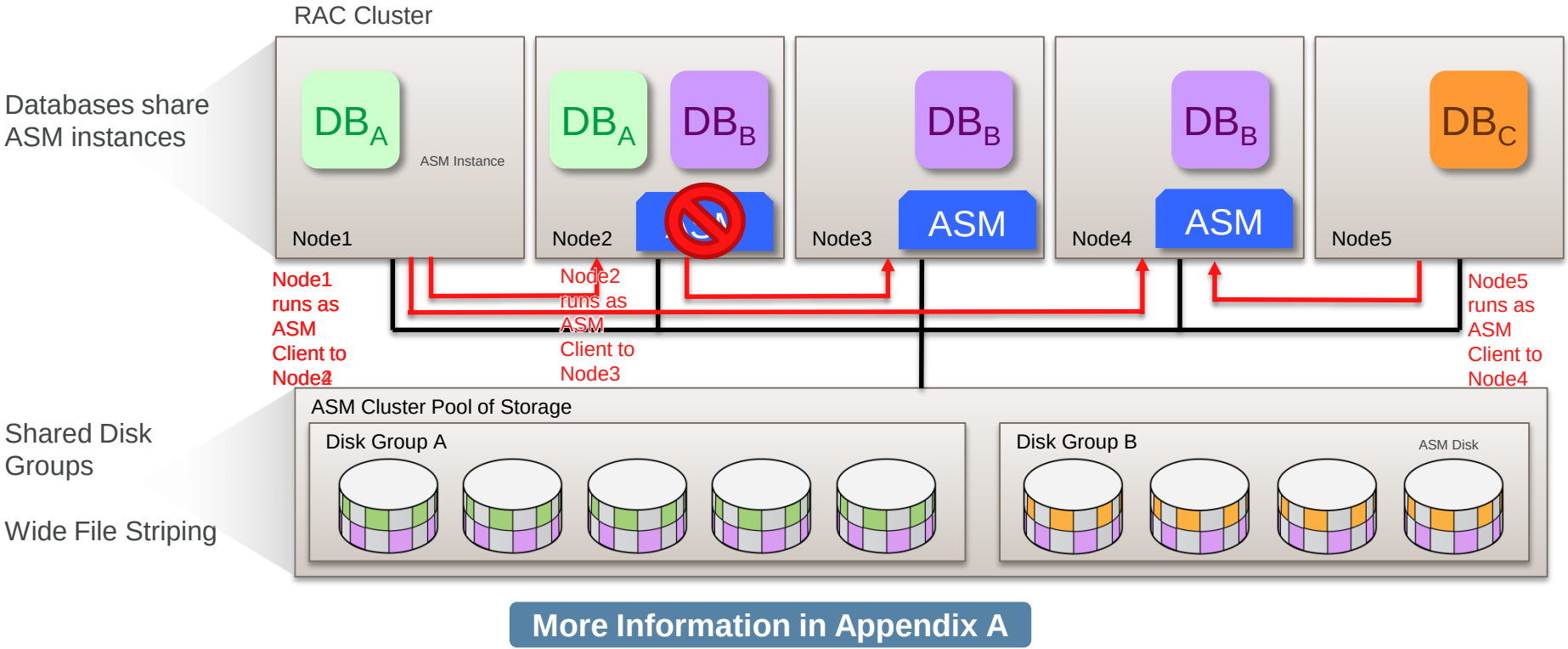
Shared Disk Groups

Wide File Striping



Introducing Oracle Flex ASM

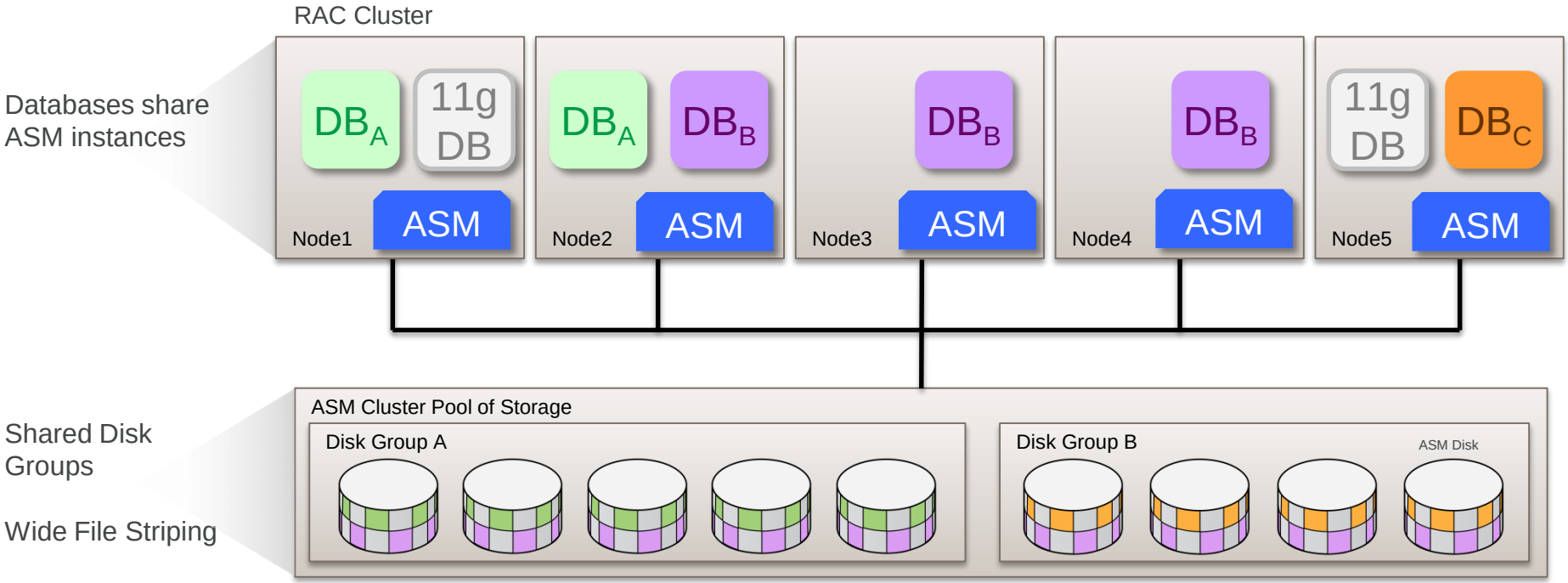
Removal of One to One Mapping and HA



More Information in Appendix A

Supporting Pre-Oracle 12*c* Databases

Pre-Oracle 12*c* Databases require a local ASM instance



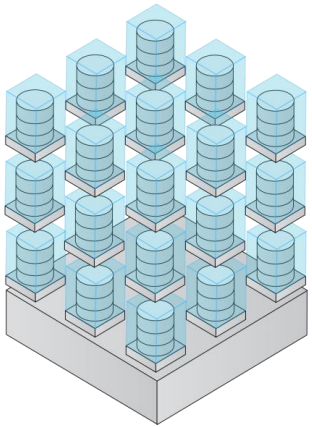
Oracle Multitenant

- Why?
- How?

Private Database Cloud Architectures

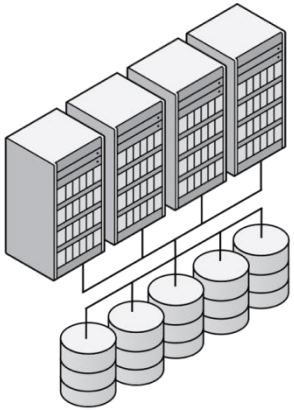
Oracle Database 11g

Virtual Machines



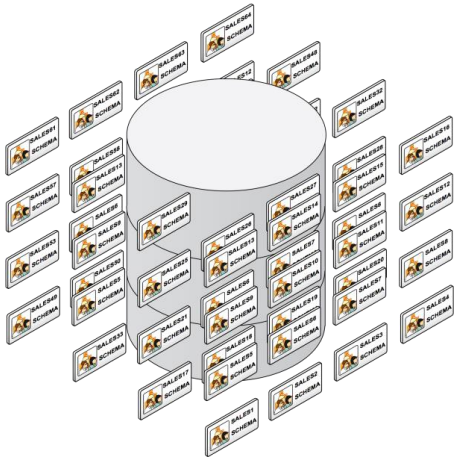
share servers

Dedicated Databases



share servers and OS

Schema Consolidation



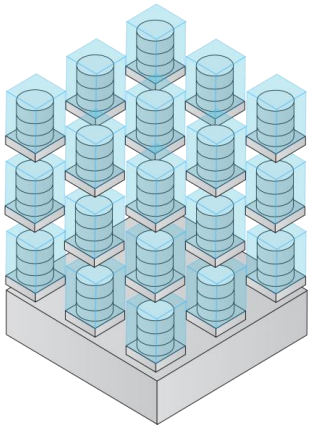
share servers, OS and database

Increasing Consolidation

Private Database Cloud Architectures

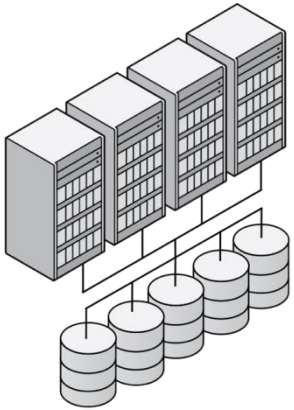
Oracle Database 12*c*

Virtual Machines



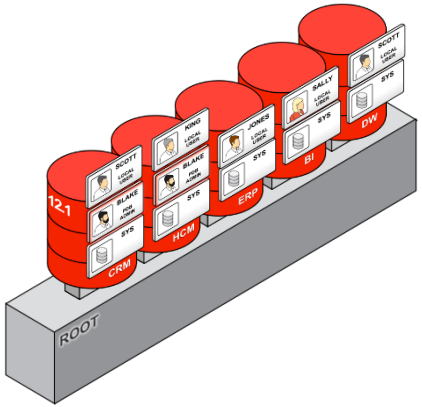
share servers

Dedicated Databases



share servers and OS

Multitenant Database

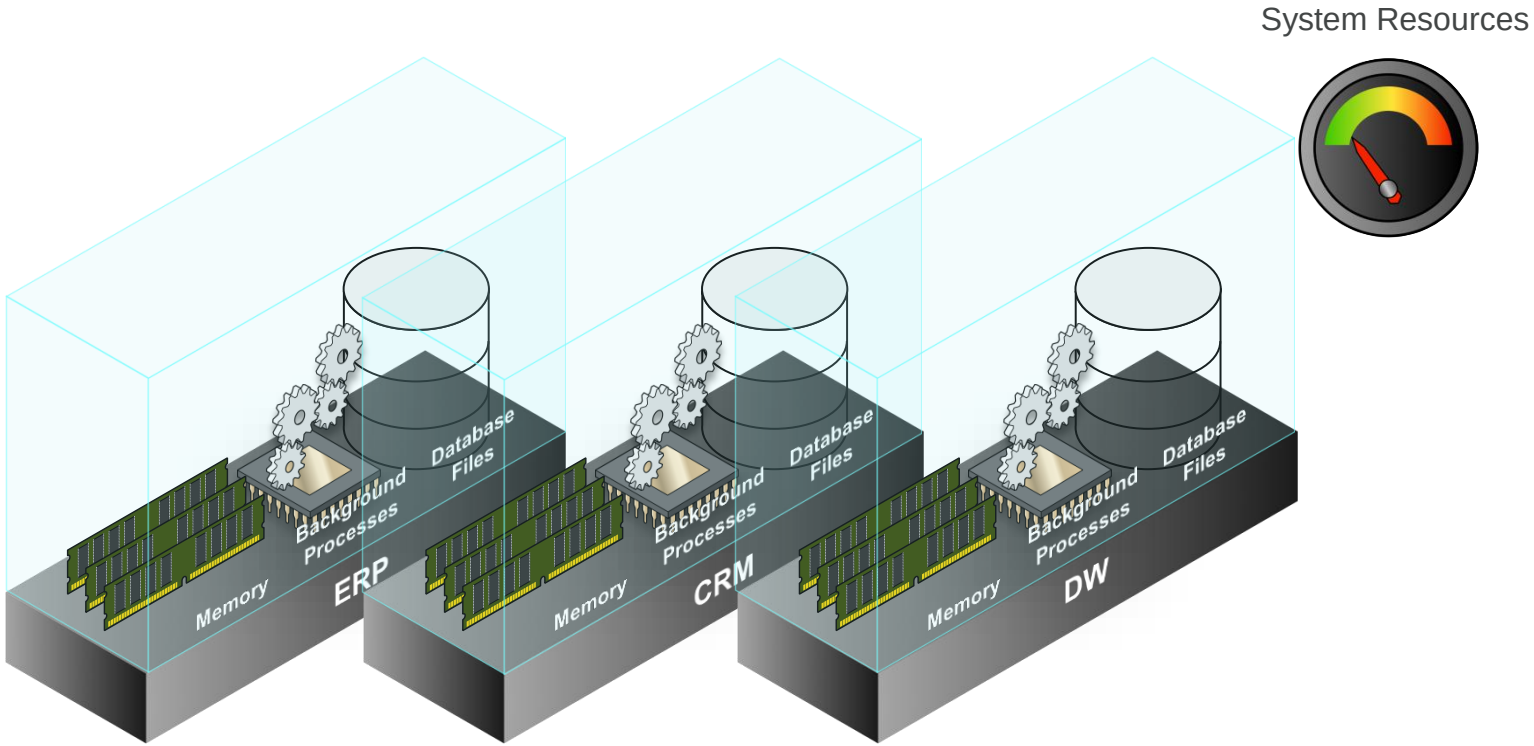


share servers, OS and database

Increasing Consolidation

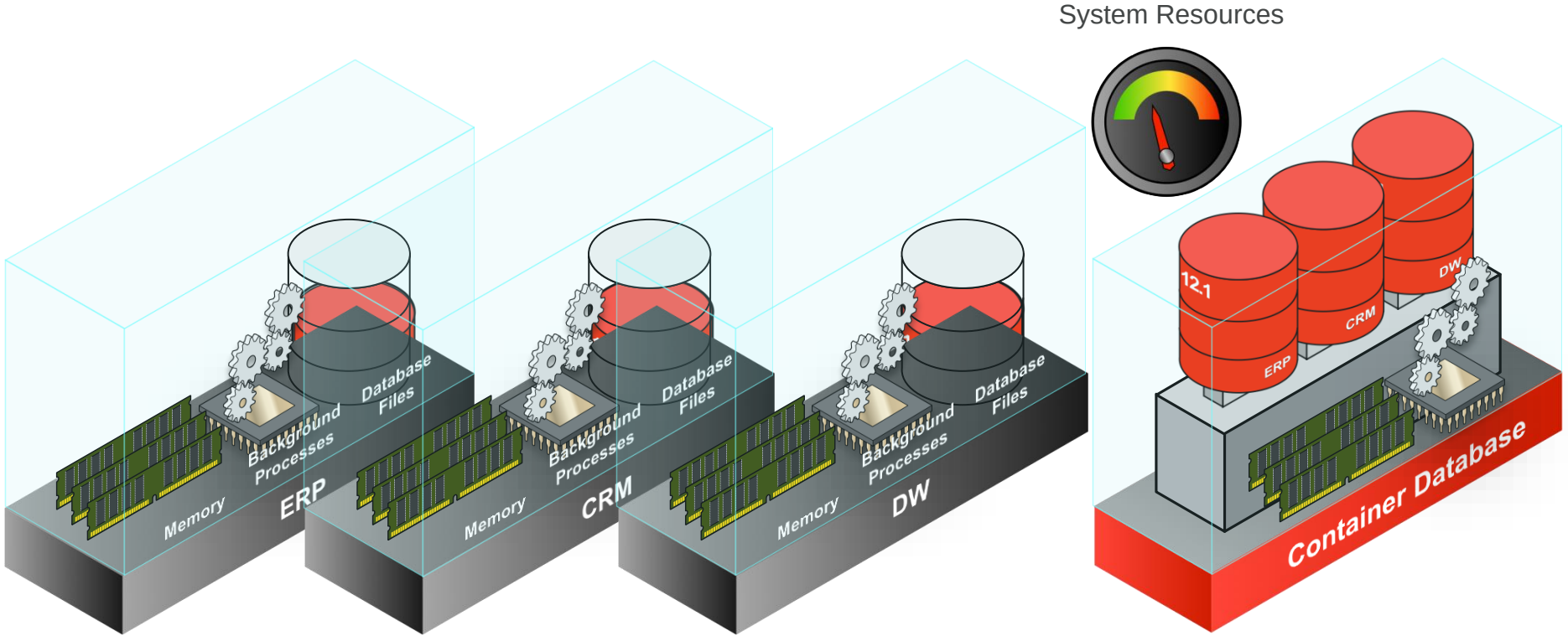
Oracle Database Architecture

Requires memory, processes and database files



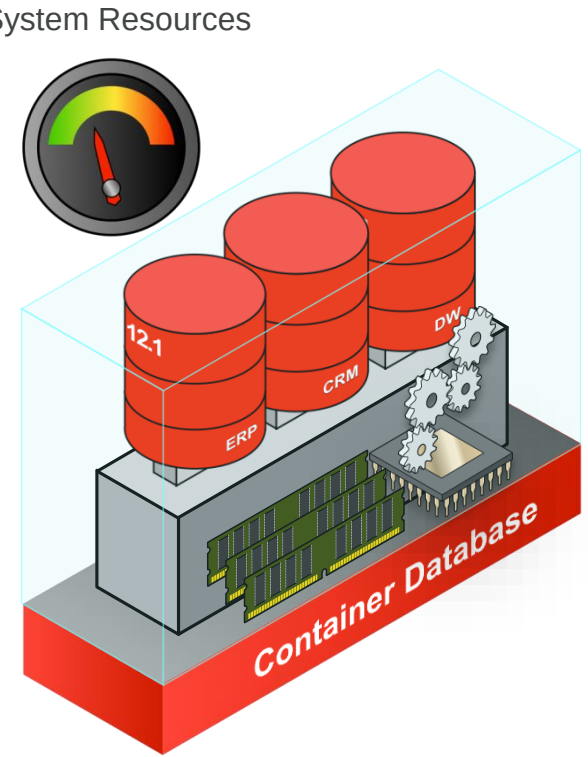
New Multitenant Architecture

Memory and processes required at multitenant container level only



New Multitenant Architecture

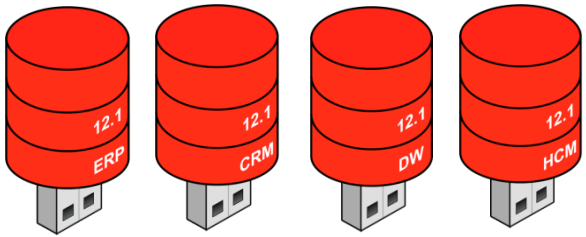
Memory and processes required at multitenant container level only

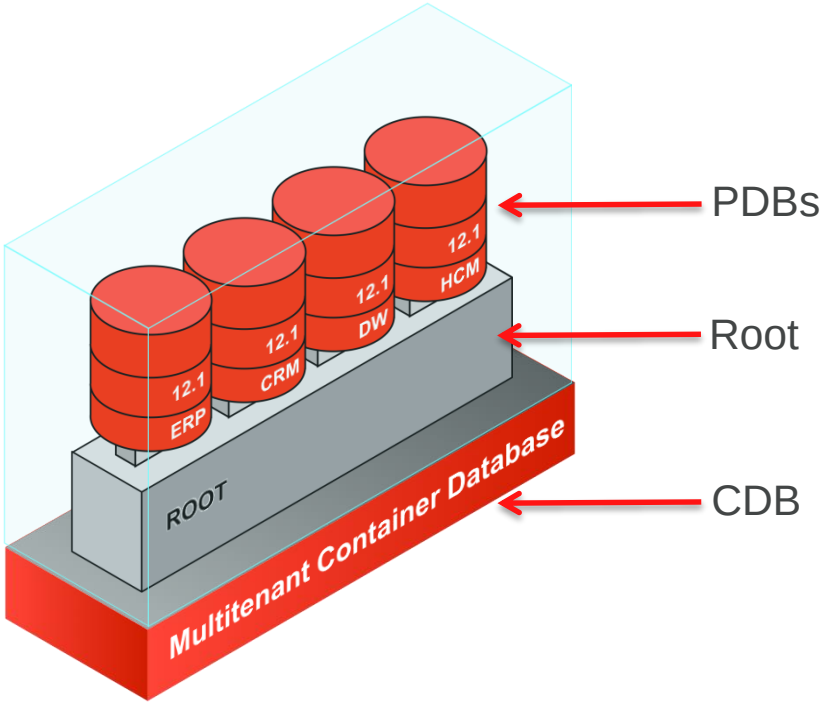


Multitenant Architecture

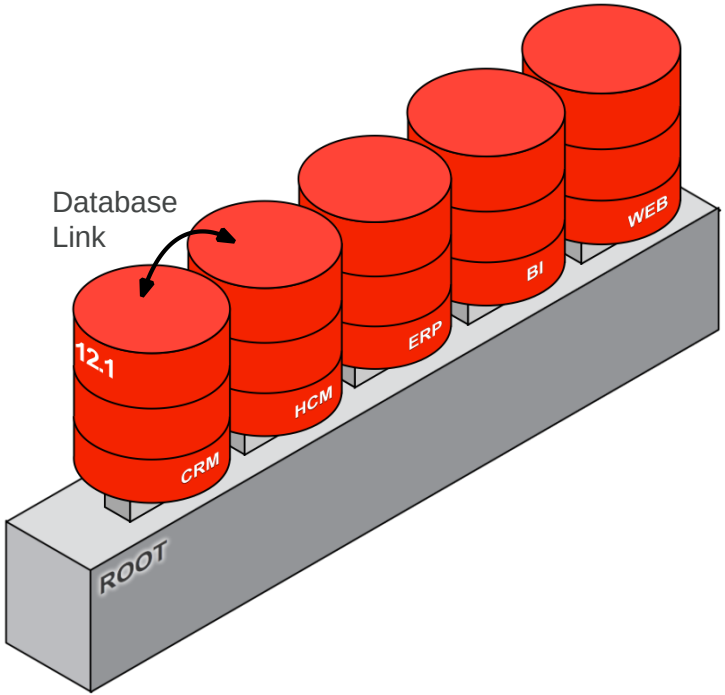
Components of a Multitenant Container Database (CDB)

Pluggable Databases (PDBs)





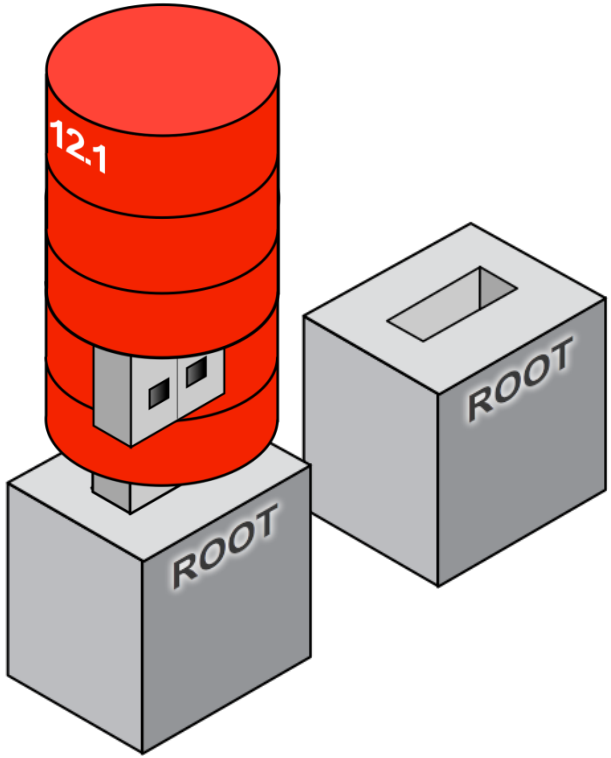
Multitenant Architecture



- Multitenant architecture can currently support up to 252 PDBs
- A PDB feels and operates identically to a non-CDB
- You cannot tell, from the viewpoint of a connected client, if you're using a PDB or a non-CDB

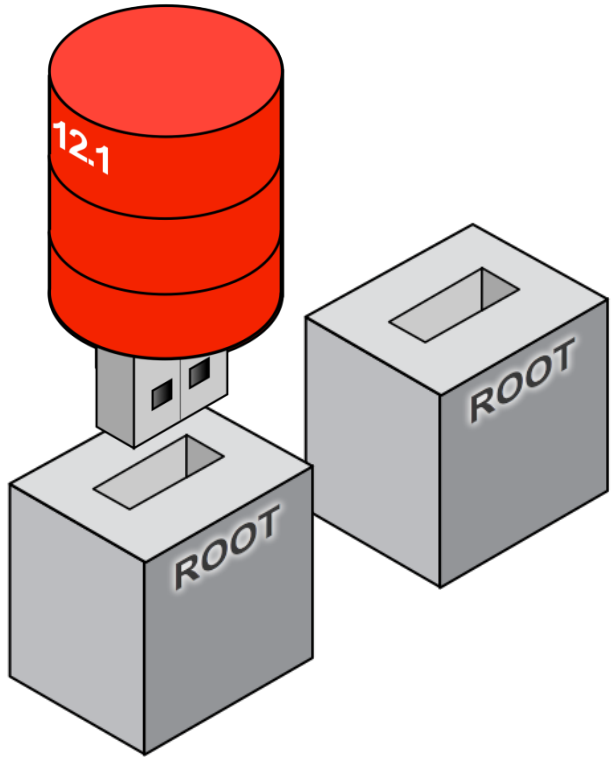
Unplug / plug

Simply unplug from the old CDB…



Unplug / plug

…and plug in to the new CDB…



- Moving between CDBs is a simple case of moving a PDB's metadata
- An unplugged PDB carries with it lineage, opatch, encryption key info etc

Unplug / plug

Example

Unplug

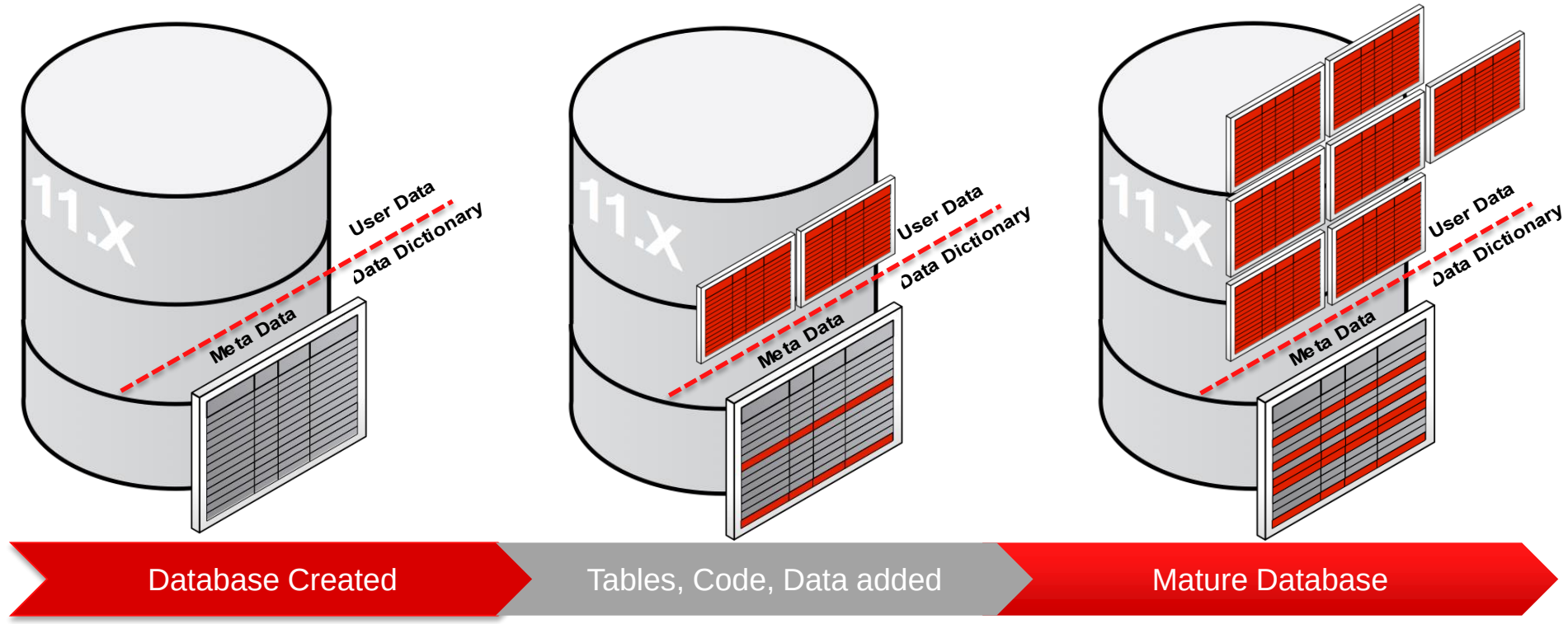
```
alter pluggable database HCM
unplug into '/u01/app/oracle/oradata/…/hcm.xml'
```

Plug

```
create pluggable database My_PDB
using '/u01/app/oracle/oradata/…/hcm.xml'
```

Common Data Dictionary

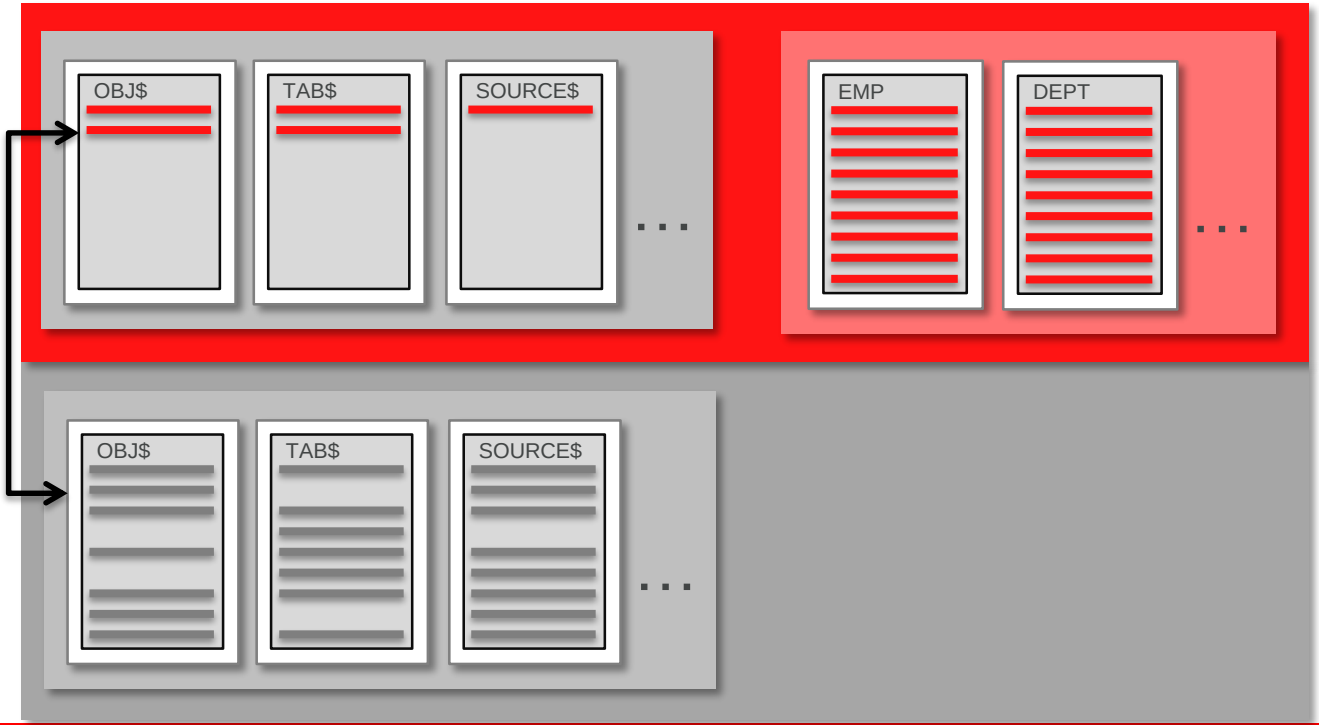
Before 12.1: dilution over time



Oracle Data and User Data

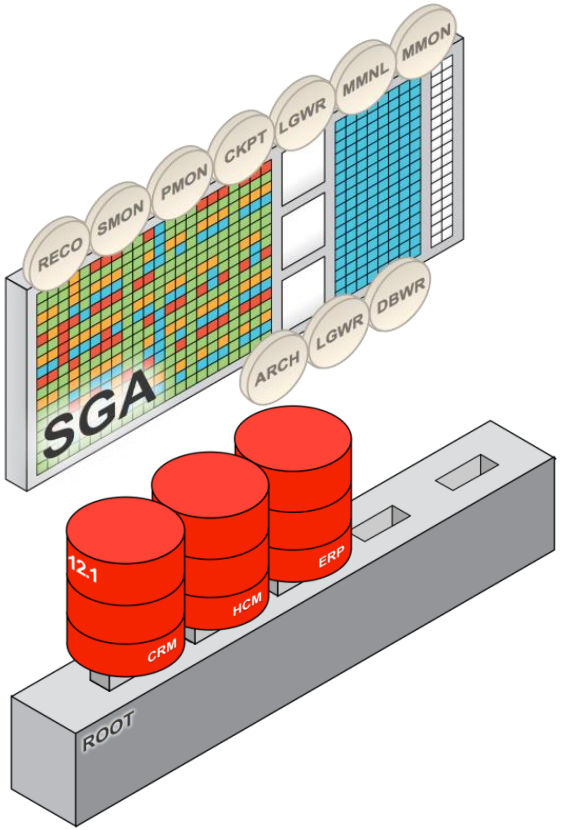
- Multitenant fix: Horizontally-partitioned data dictionary
- Only Oracle system definition remains
- Data dictionary is diluted by customer's metadata

Horizontally Partitioned Data Dictionary



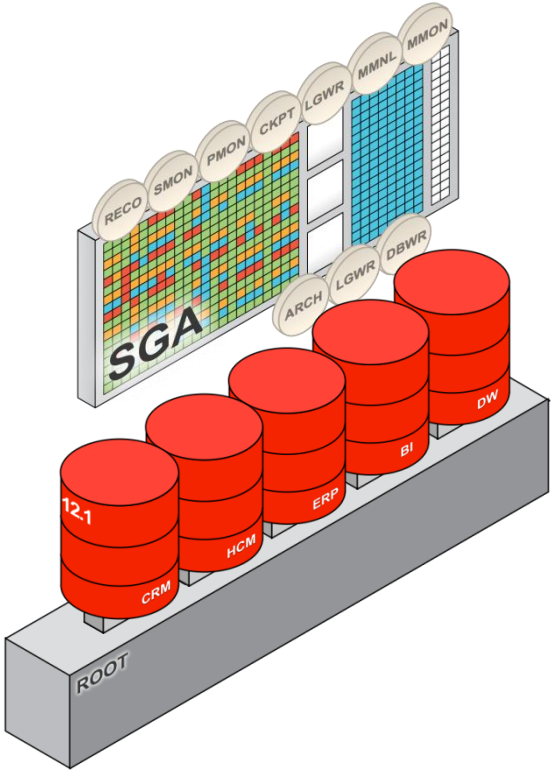
- Oracle-supplied objects such as views, PL/SQL, etc., are shared across all PDBs using object “stubs”
- In-database virtualization

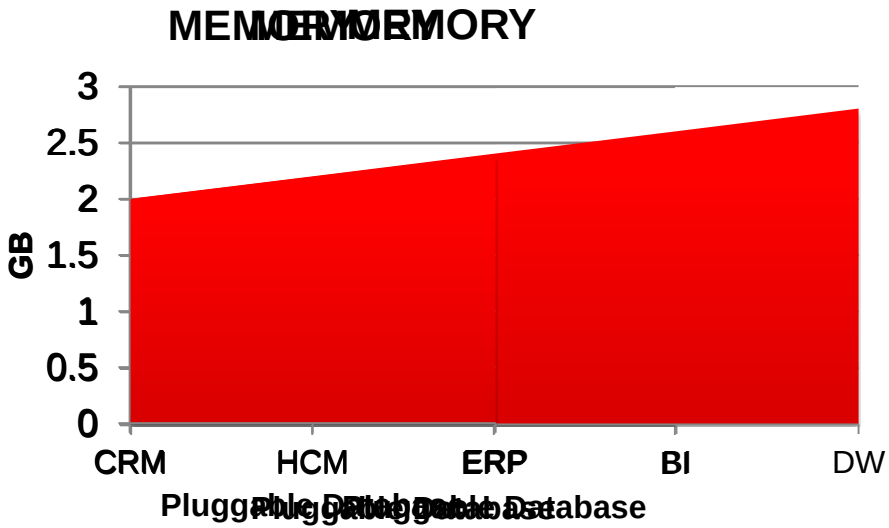
Multitenant Architecture – Dynamics



- PDBs share common SGA and background processes
- Foreground sessions see only the PDB they connect to

Multitenant Scalability

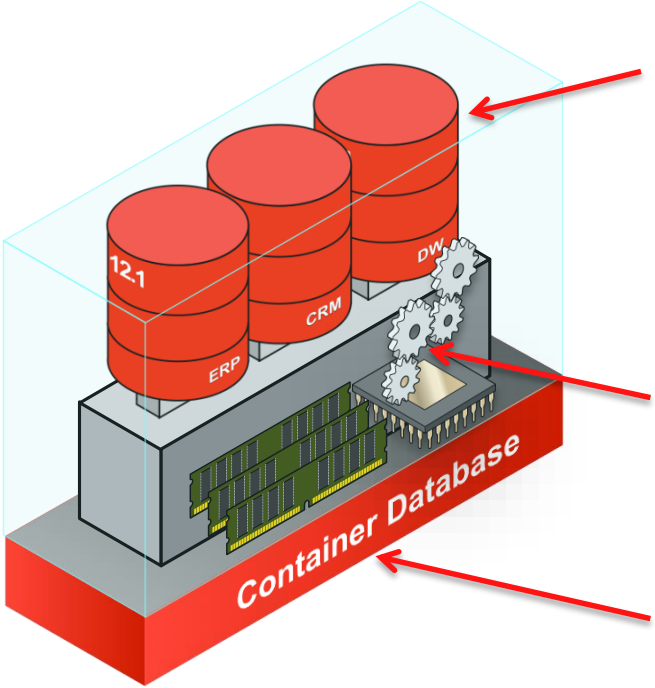




- Only small increments in memory as additional PDBs are added

Advantages of Oracle Multitenant Architecture

Increased Agility, Easy Adoption



Self-contained PDB for each application

- Applications run unchanged
- Rapid provisioning (via clones)
- Portability (via pluggability)

Shared memory and background processes

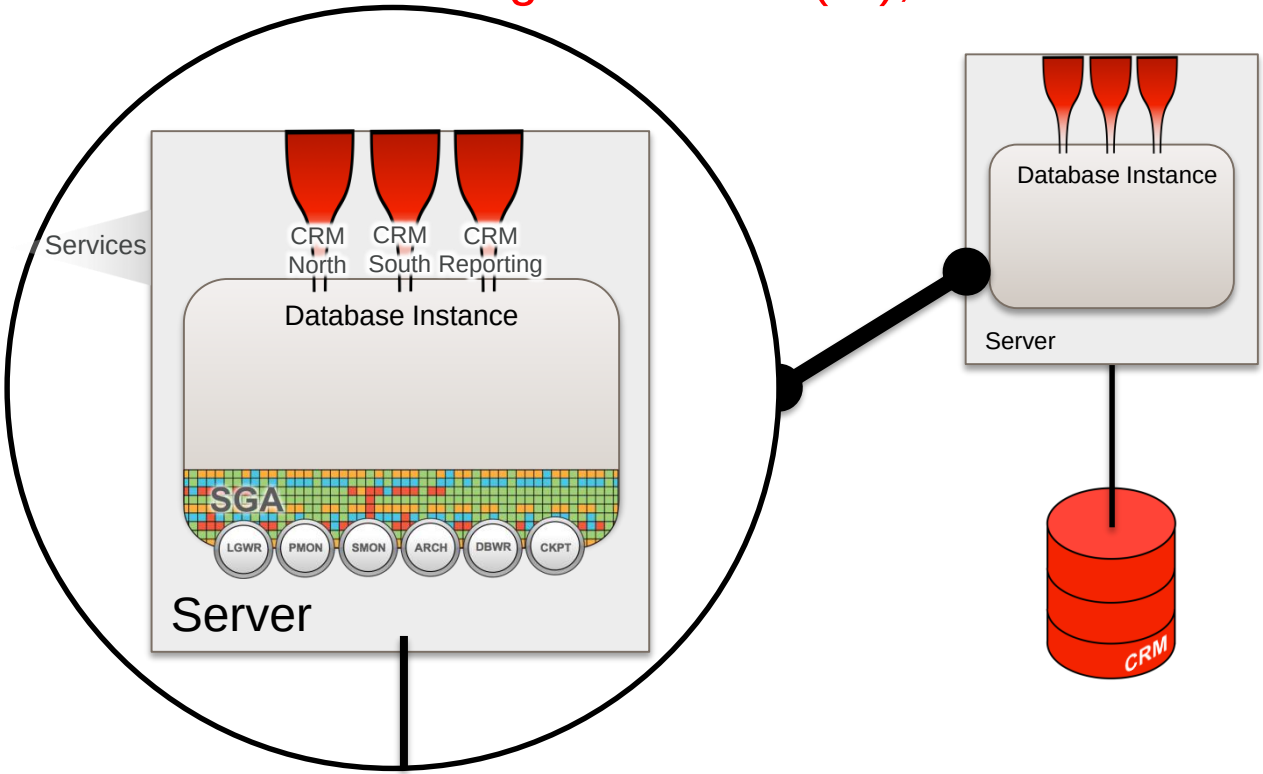
- More applications per server

Common operations performed at CDB level

- Manage many as one (upgrade, HA, backup)
- Granular control when appropriate

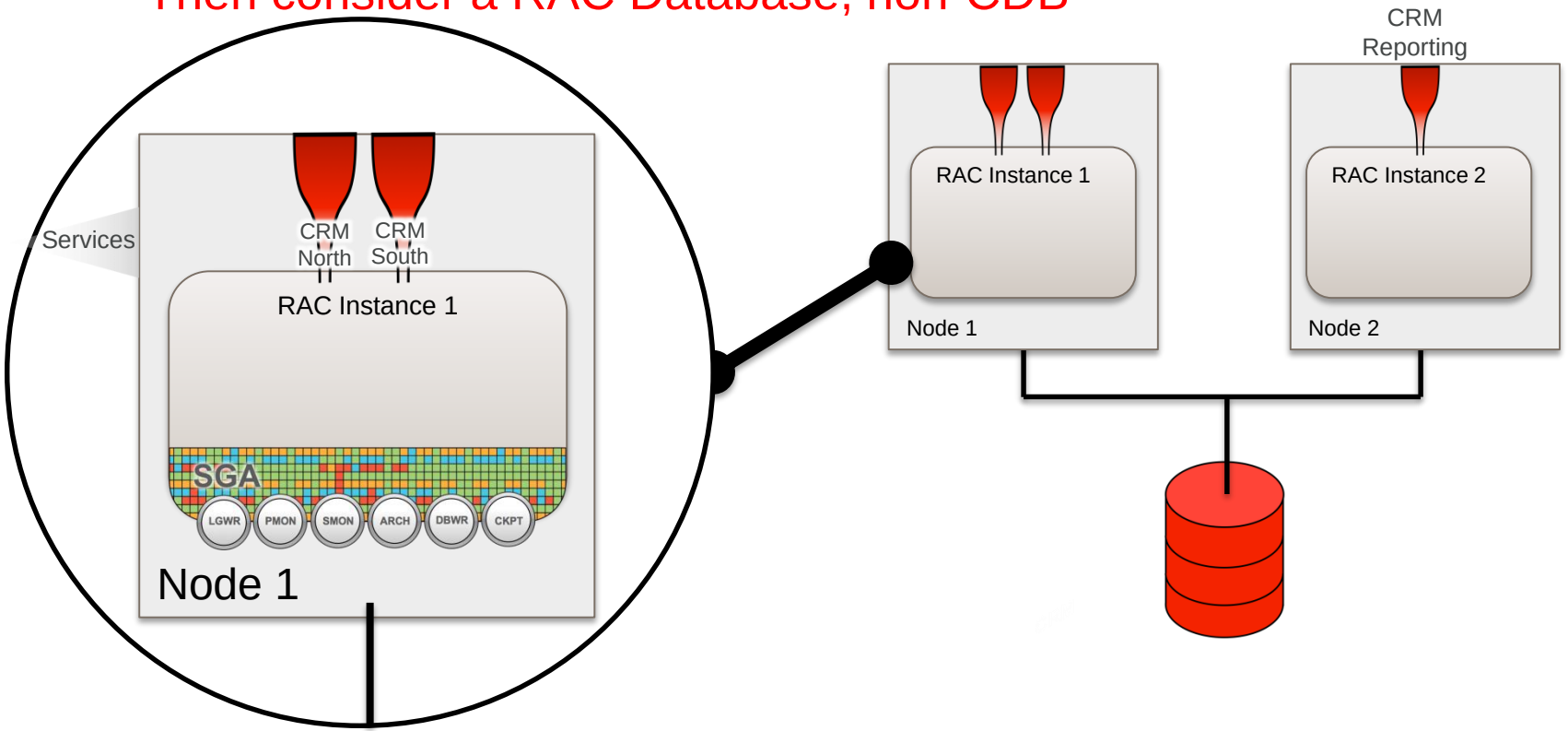
Oracle Multitenant on Oracle RAC

Consider a Single Instance (SI), non-CDB



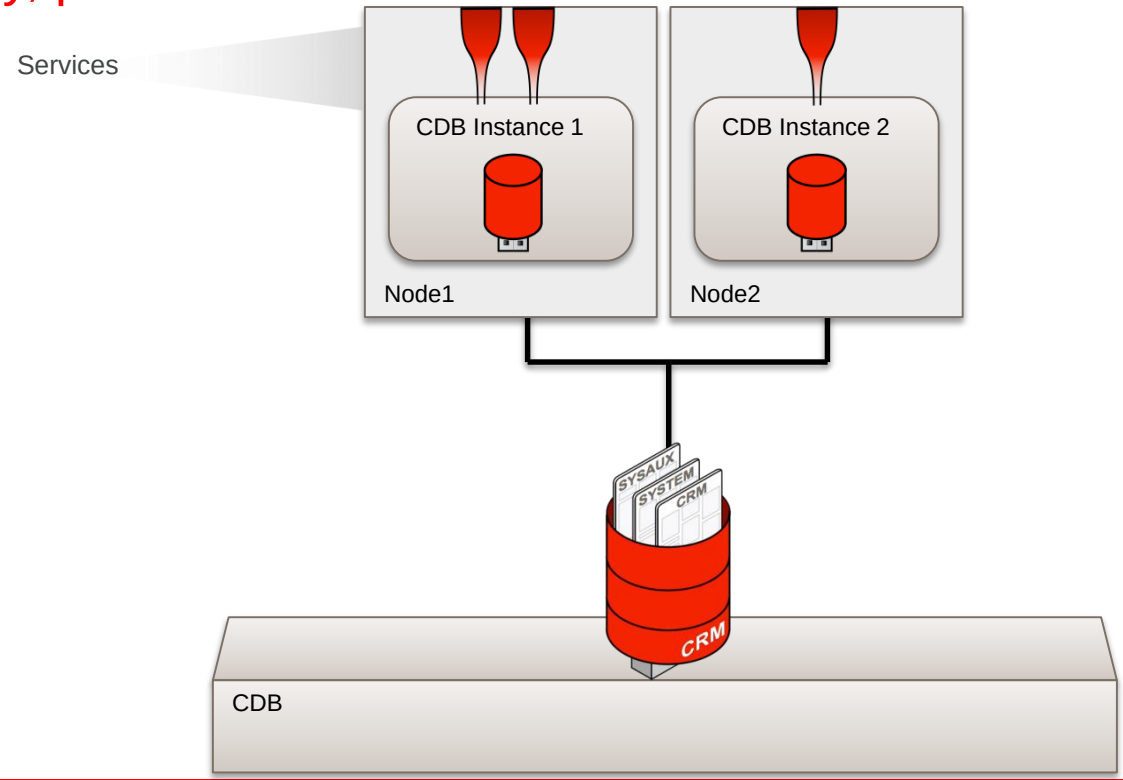
Oracle Multitenant on Oracle RAC

Then consider a RAC Database, non-CDB



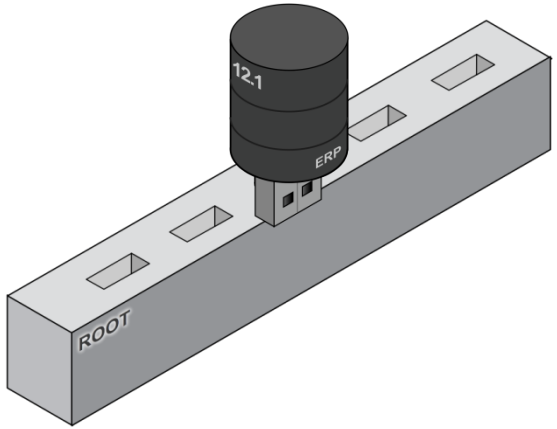
Oracle Multitenant on Oracle RAC

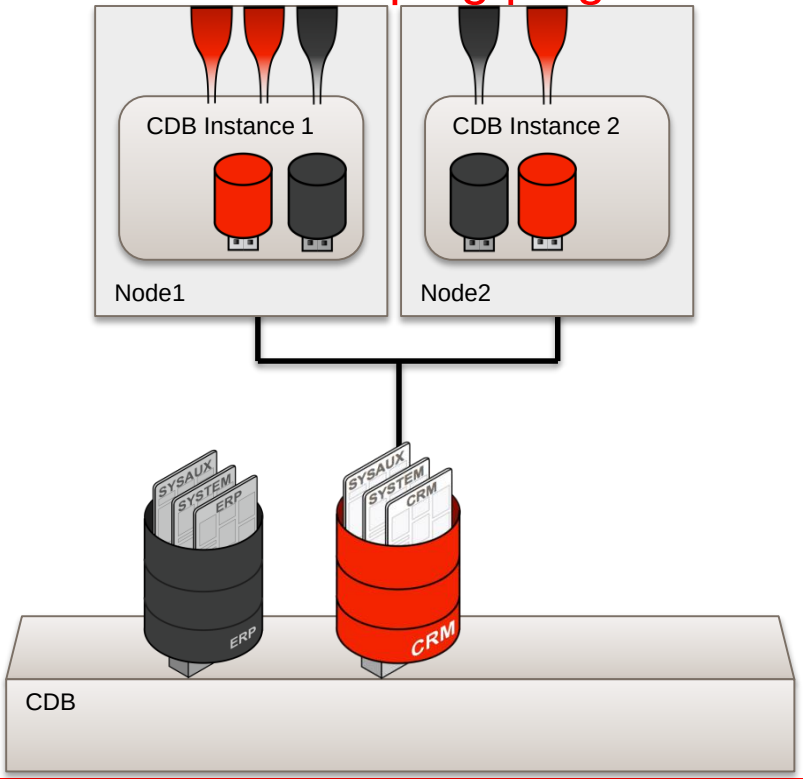
Finally, picture a CDB RAC Database



Oracle Multitenant on Oracle RAC

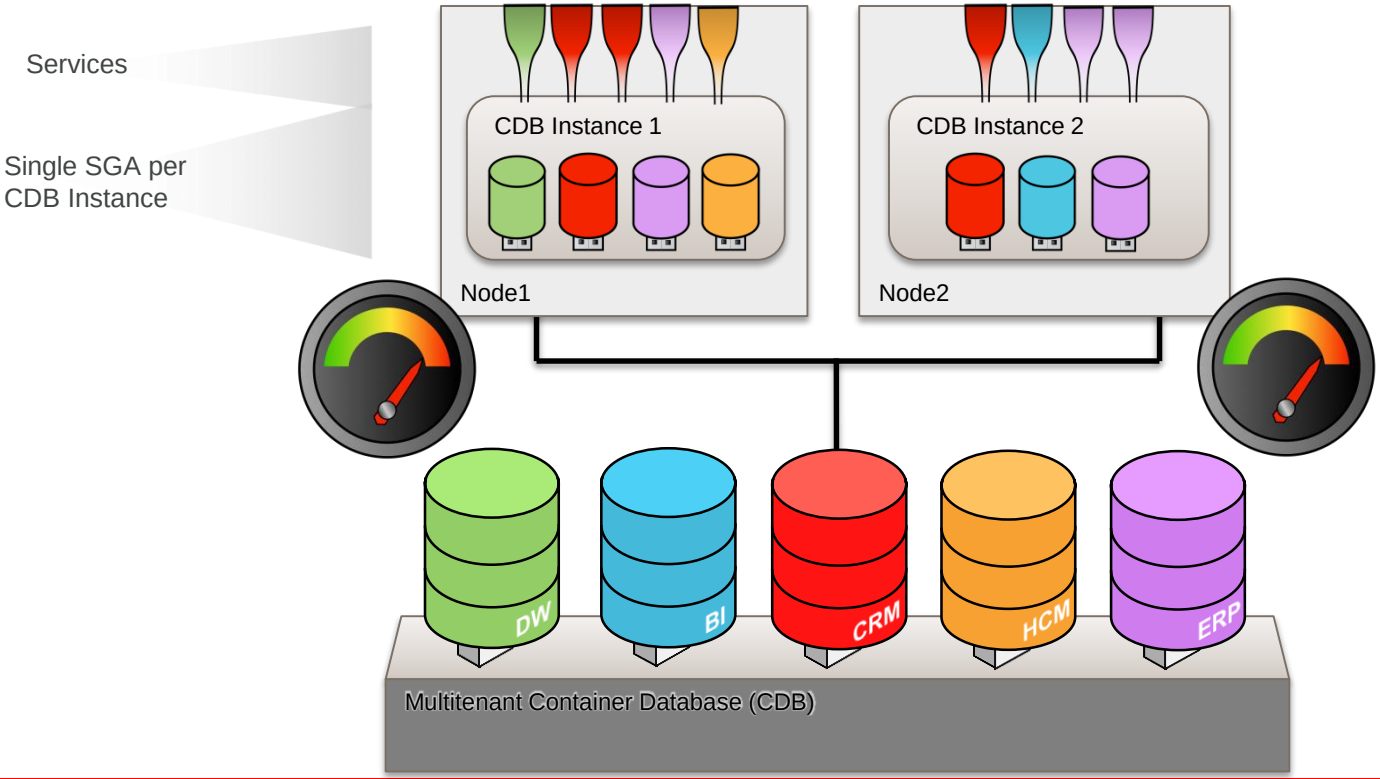
The simplest way of converting a SI PDB to RAC: unplug/plug





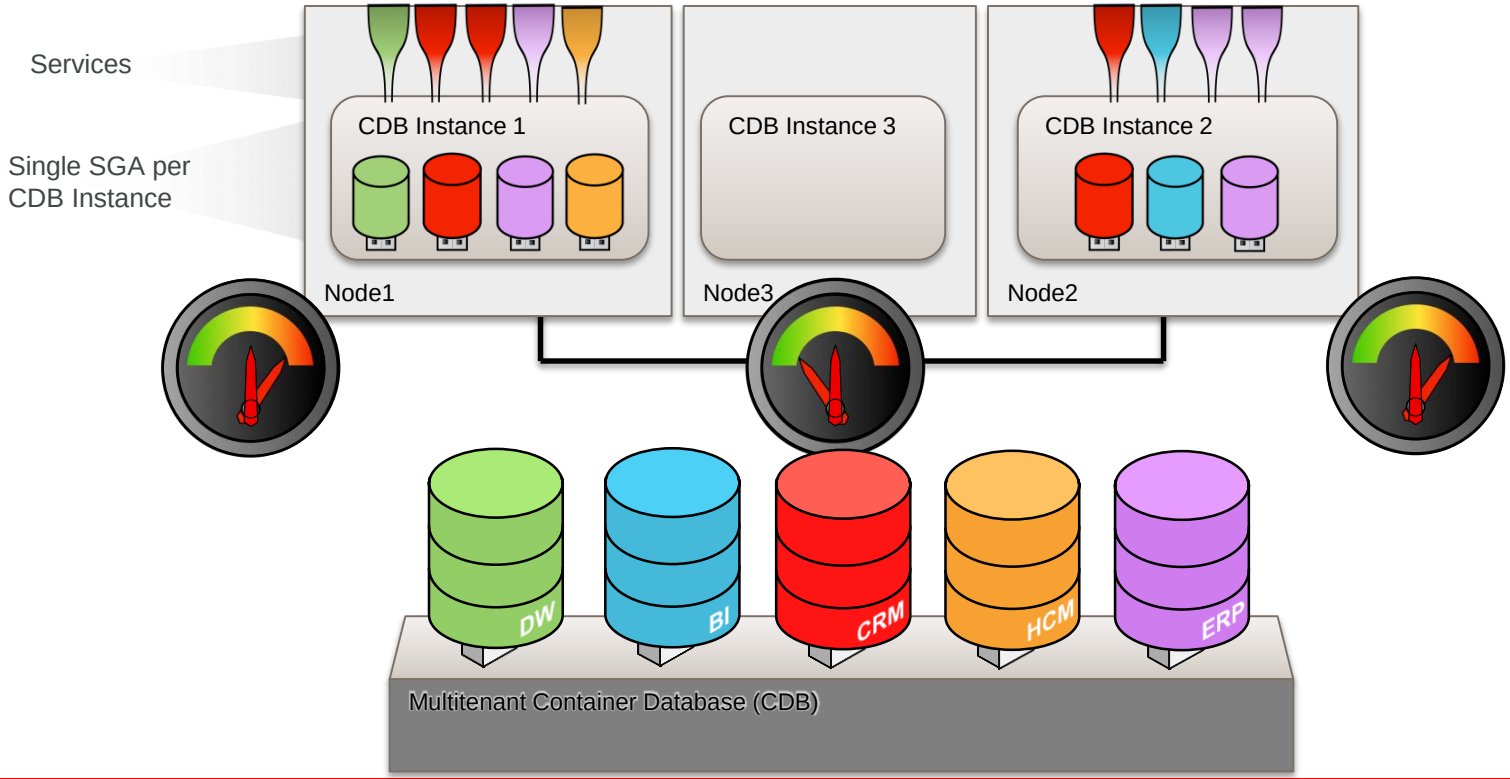
Improved Agility With Changing Workloads

Utilize Nodes in the Cluster to Support Flexible Consolidation Model



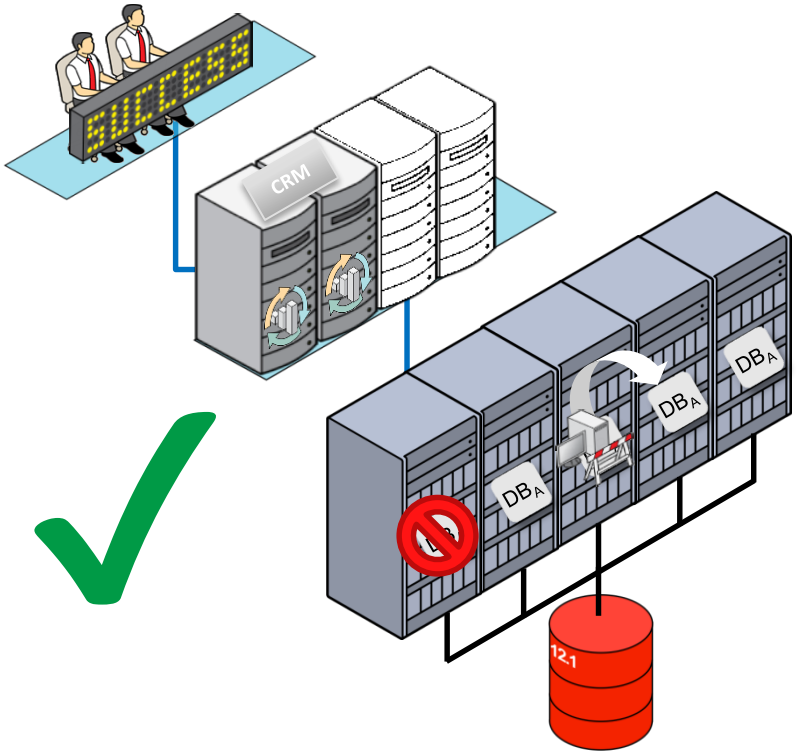
Improved Agility With Changing Workloads

Utilize Nodes in the Cluster to Support Flexible Consolidation Model



Application Continuity

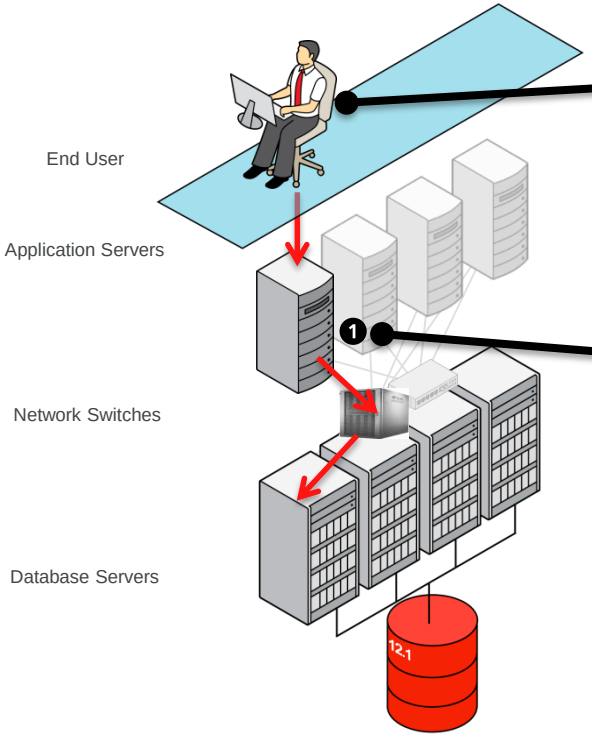
Masks Unplanned & Planned Outages



- Replays in-flight (DML) work on recoverable errors
- Masks many hardware, software, network, storage errors and outages when successful
- Improves end-user experience and productivity without requiring custom application development

Application Continuity – Example

A reliable replay of in flight work

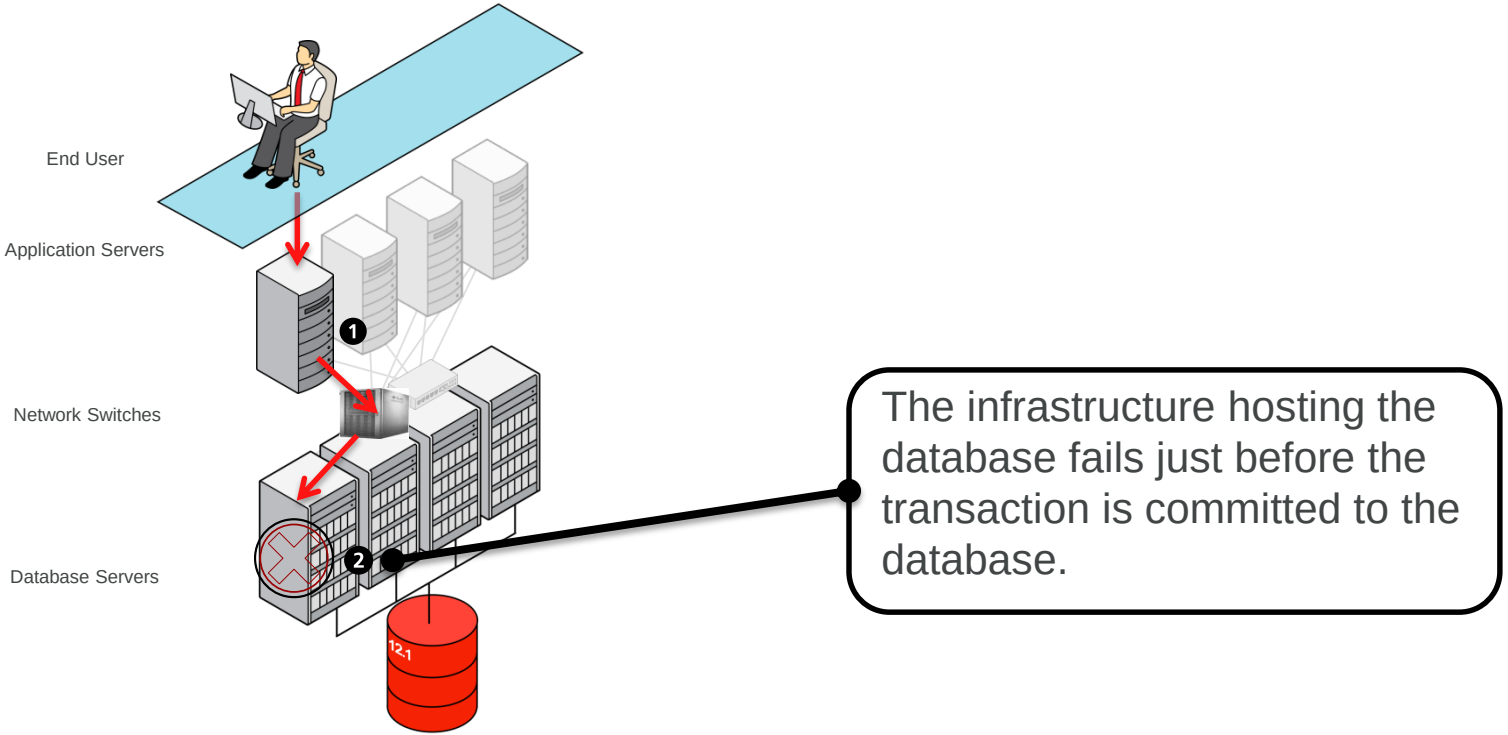


User selects product from application and purchases it from the web checkout

User transaction arrives at application infrastructure. It makes it's way through the application tiers and results in a database transaction being created

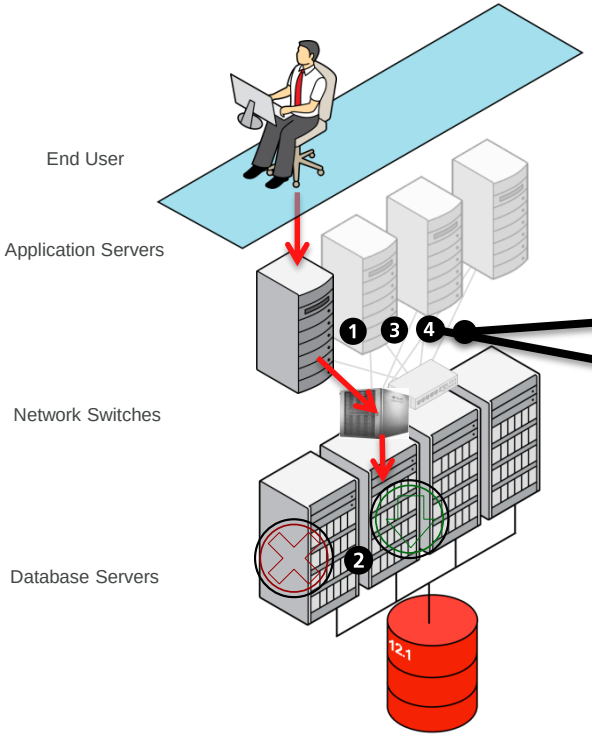
Application Continuity – Example

A reliable replay of in flight work



Application Continuity – Example

A reliable replay of in flight work

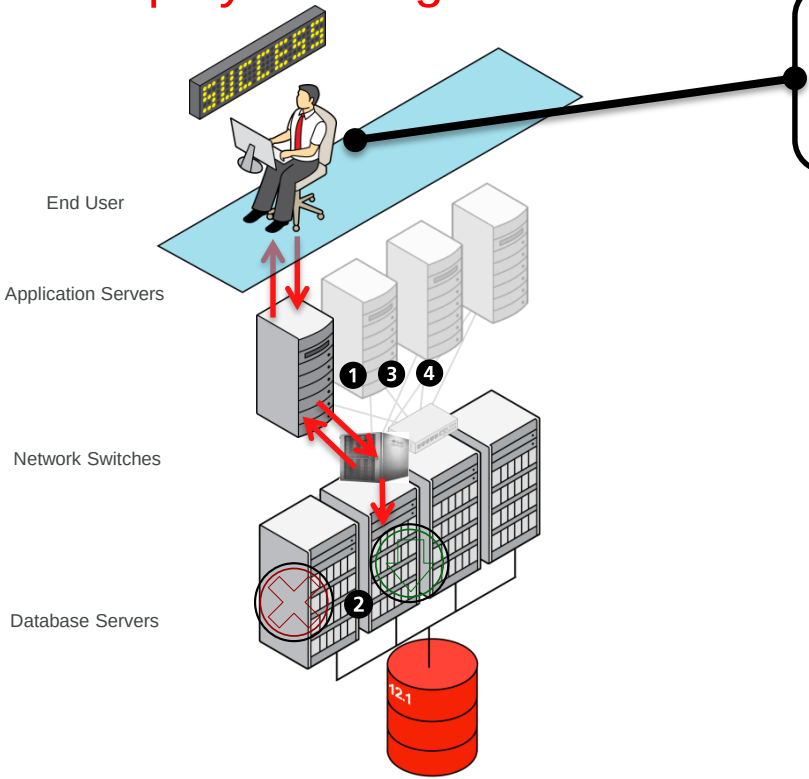


If the transaction needs to be replayed, “Application Continuity” will submit all of the inflight work to a surviving node in the cluster and perform a commit. This all happens transparently to the application

replayed

Application Continuity – Example

A reliable replay of in flight work

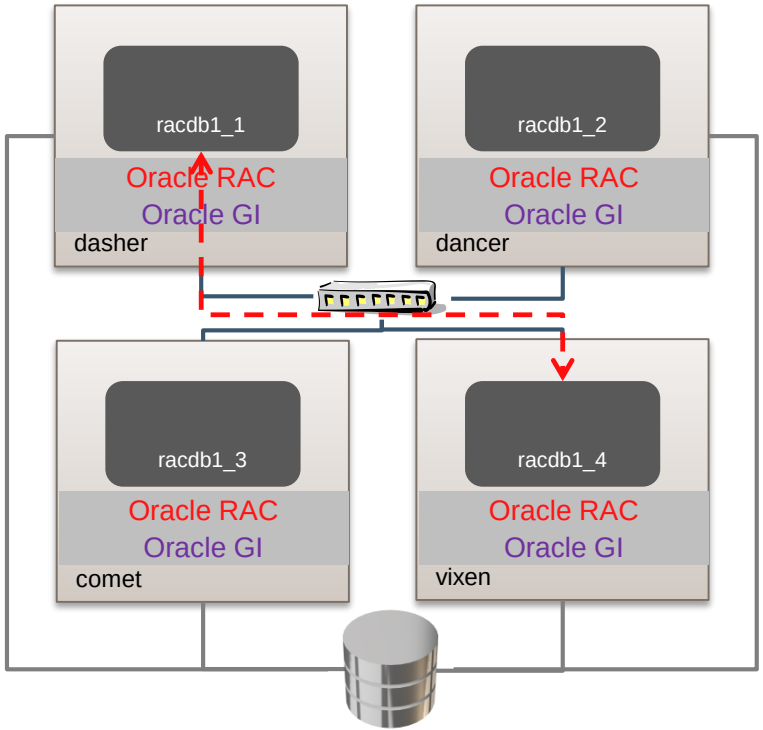


The user receives confirmation that his order has been successfully completed.

Appendix

Oracle RAC Fundamentals

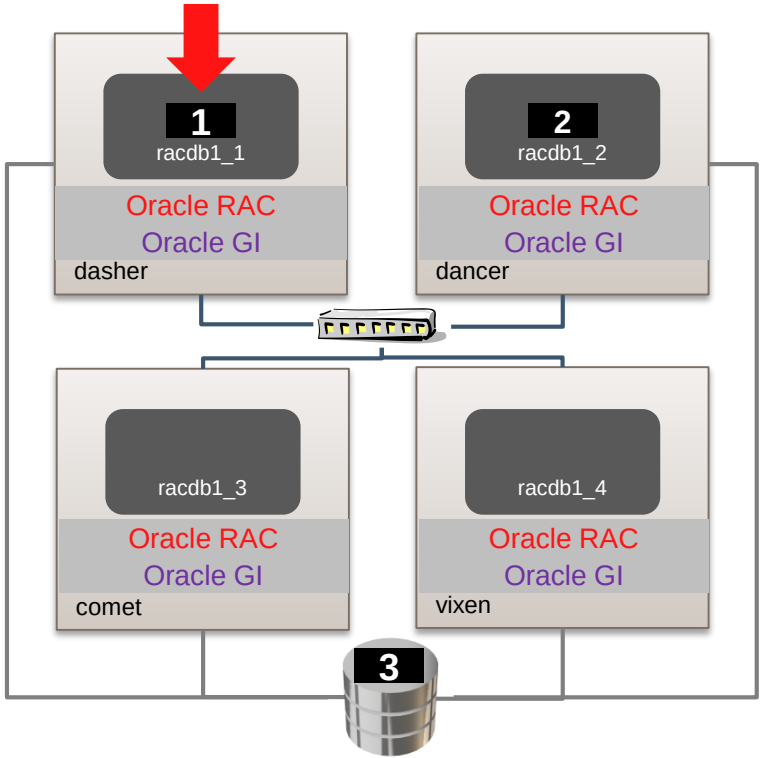
Communication flow in the cluster



- Instances communicate over the private interconnect for 2 purposes:
  1. Function / message shipping
  2. Data shipping (block transfer)
     - In order to minimize spinning disk access

Oracle RAC Fundamentals

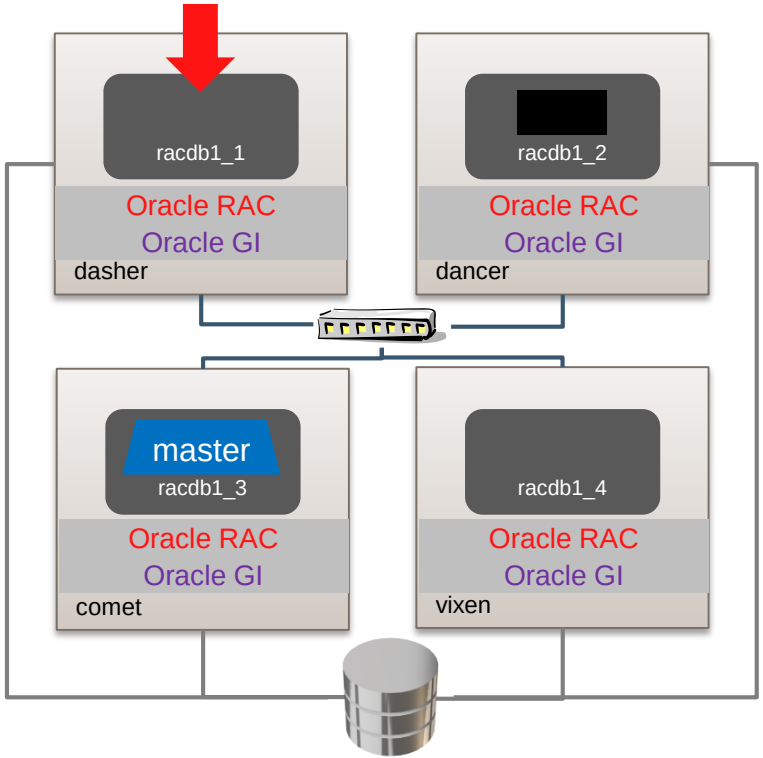
“3” ways of getting access to data



- Data is either stored
  1. Locally (local cache) → access time: nanoseconds
  2. Remote (global cache) → access time: micros.
  3. “On disk”
     - Flash cache → access time: microseconds
     - Disk controller cache → access time: micros.
     - Spinning disk → access time: milliseconds

Oracle RAC Fundamentals

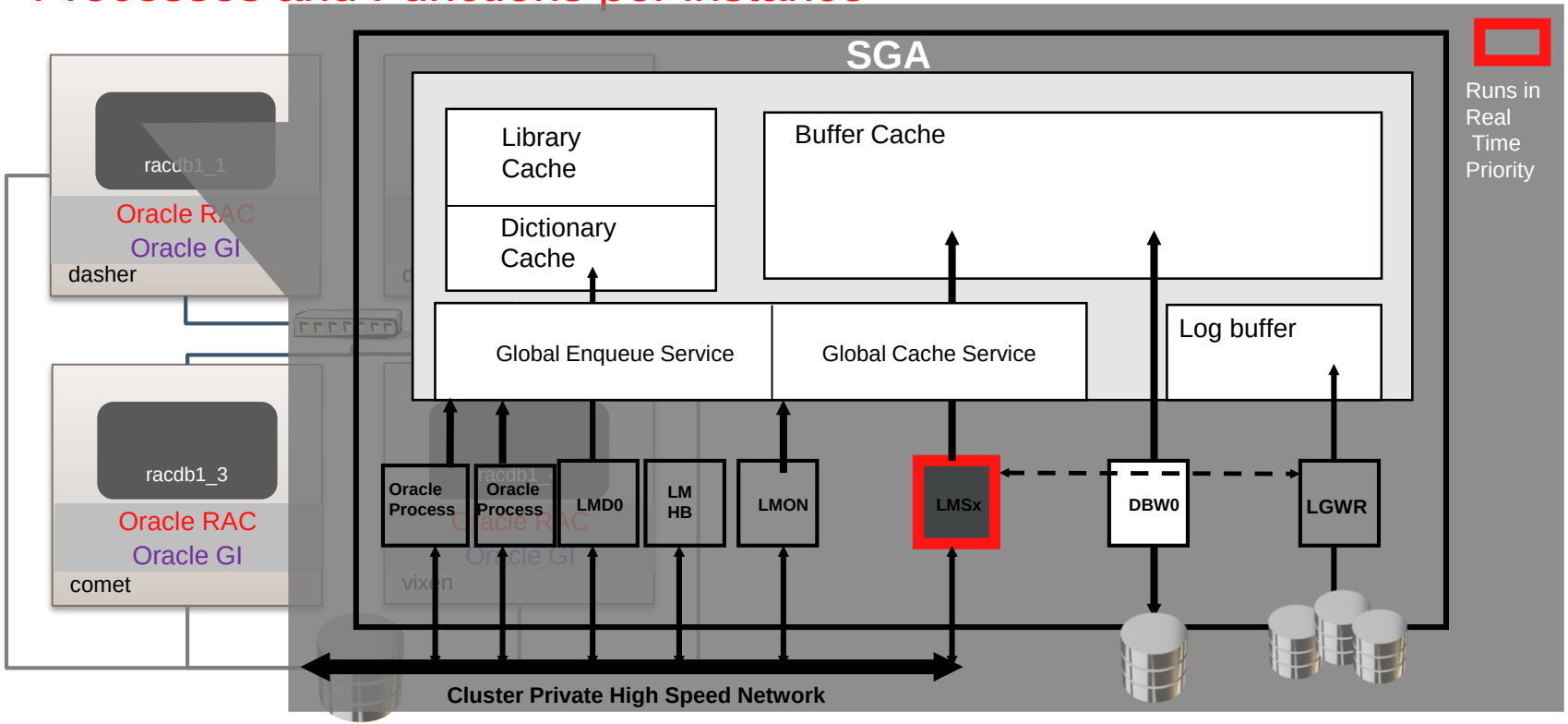
Maximum “3” way communication to access data



- In the worst case,
  - The requester asks
  - for data held in a remote instance
  - mastered in a third instance

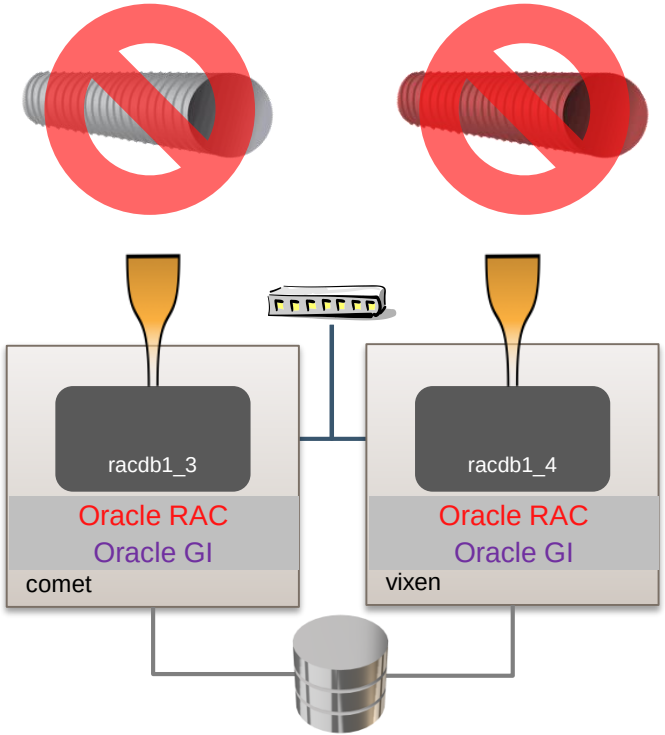
Oracle RAC Fundamentals

Processes and Functions per instance



Application Considerations

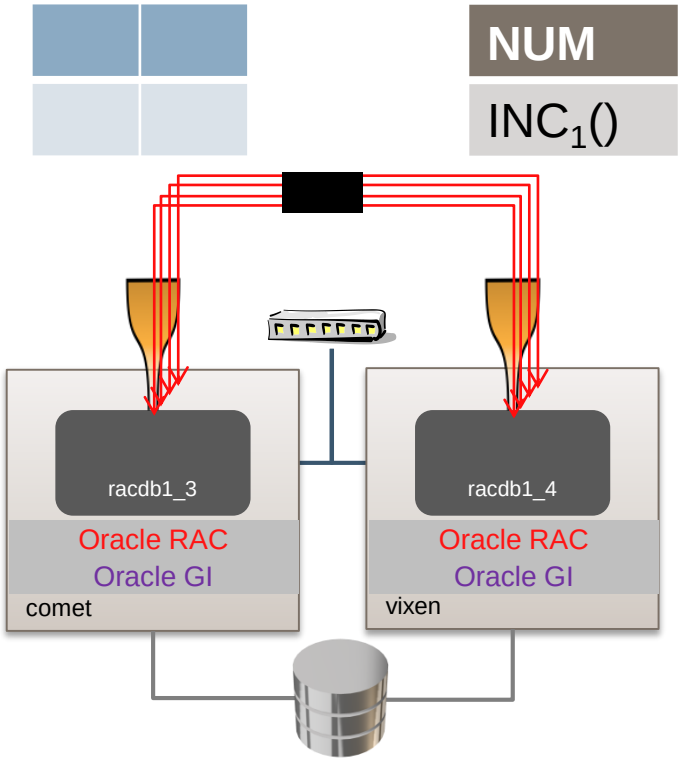
What to avoid in any case …



- Do not use (named) pipes
  - A pipe on one server may not exist on the other

Application Considerations

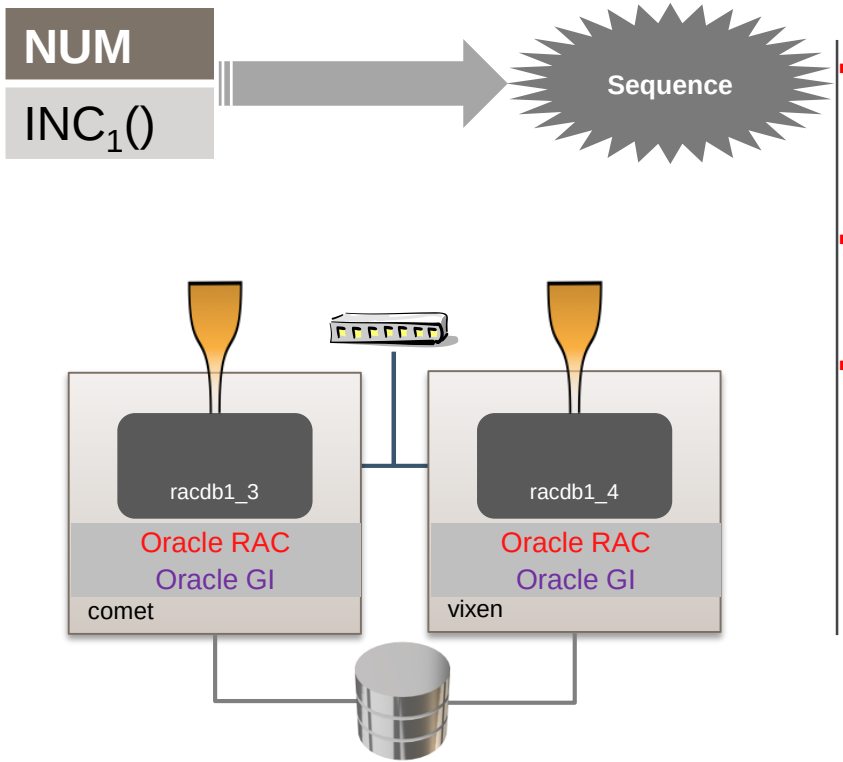
How to avoid “Write Hot Spots” in applications – part 1



- Frequent transactional changes to the same data blocks in all instances may result in “write hot spots”
  - *In 99% of OLTP performance issues, write hot spots occur on indexes*
- Block with pending changes may be “pinged” by other instance
  - Pending redo must be written to log before the block can be transferred
  - Latency for a deferred block transfer becomes dependent on delay for log IO
- Only for very frequently modified data

Application Considerations

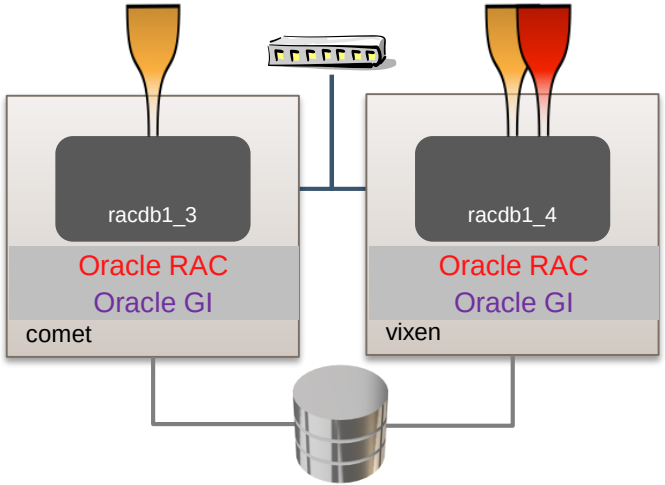
How to avoid “Write Hot Spots” in applications – part 2



- Use **non-ordered & cached** sequences if sequence is used to generate primary key
  - ALTER SEQUENCE S1 … CACHE 10000+
- Symptoms if not cached:
  - EQ or SQ contention
- Ordered Sequences
  - Do not scale well in Oracle RAC
  - Solution: Use them only on one instance in active-passive configuration
    - Create multiple per application

Application Considerations

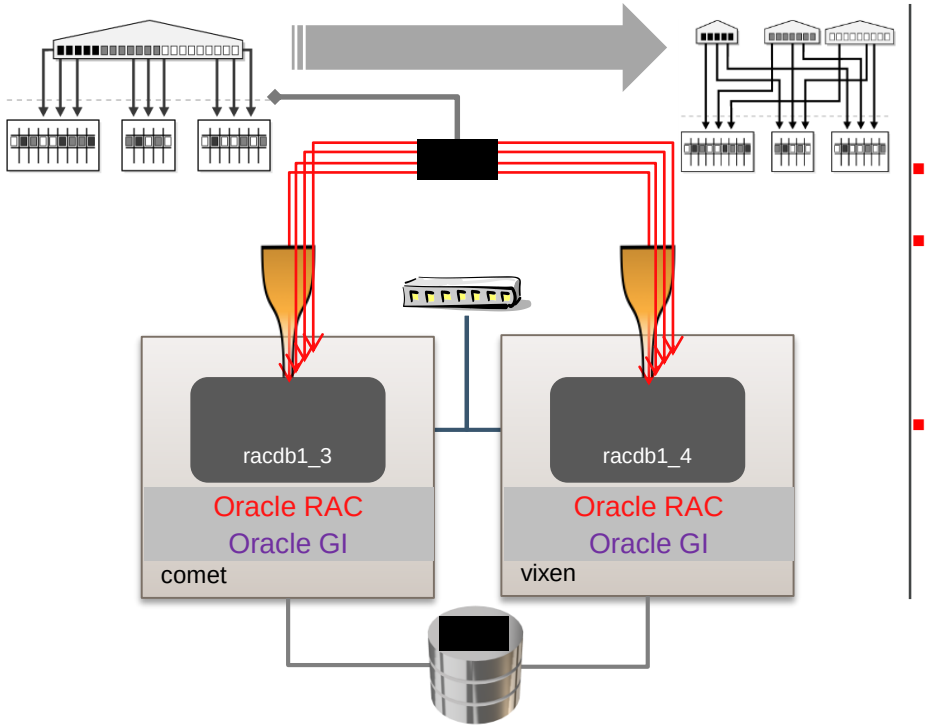
How to avoid “Write Hot Spots” in applications – part 3



- Possible:
  - Consolidate applications to use only one server and route via services
- Optimize log flush :
  - Place redo logs on fast storage if performance critical; e.g. SSDs
  - Separate disks for logs from other IO busy disks
    - Implemented in 11.2.2.4 of Exadata and Oracle Database Appliance by default (Smart Logs and SSDs, respectively)
- Schema tuning only involves minimal modification and is the preferred option

Application Considerations

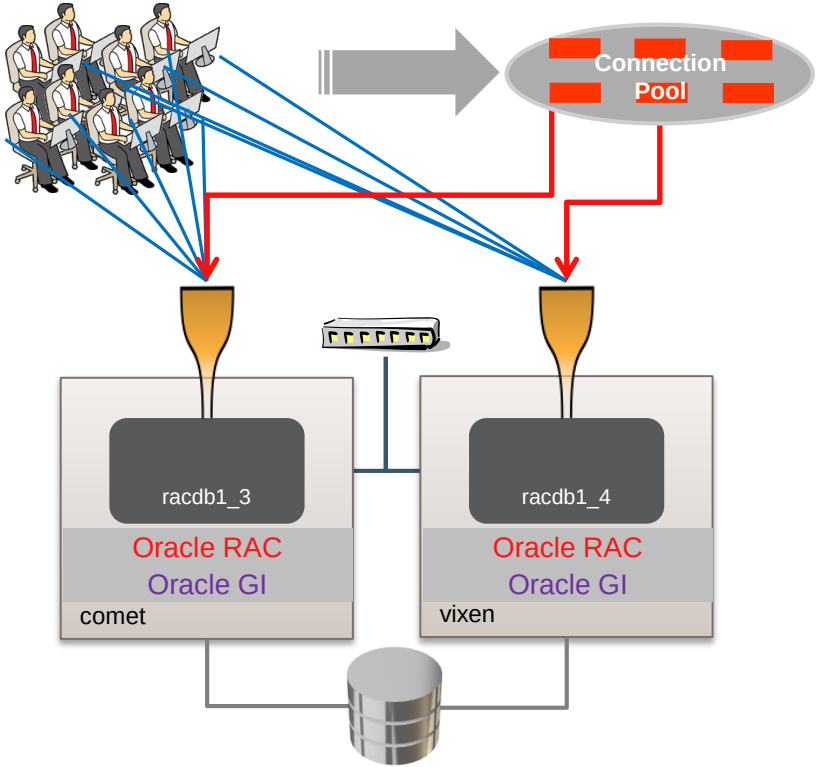
How to avoid “Write Hot Spots” in applications – part 4



- Global hash partitioned indexes
- Locally partitioned indexes
  - **Both solutions achieve better cache locality**
- Drop unused indexes

Application Considerations

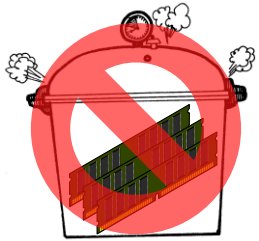
How to avoid number of sessions related resource contention

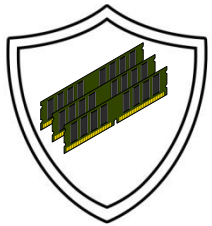


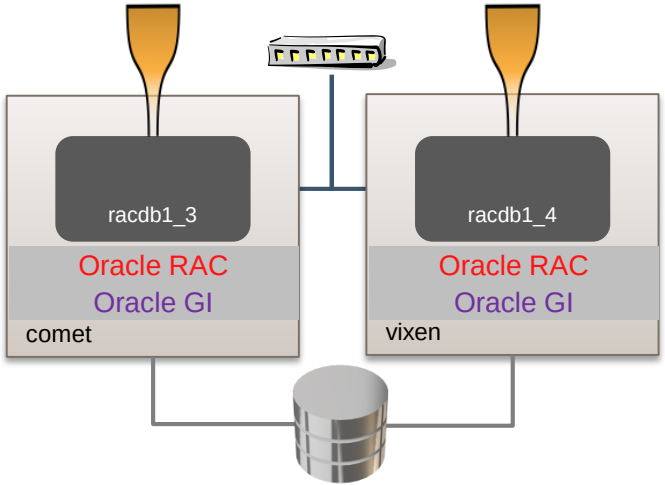
- Control the number of concurrent sessions
  - Foreground processes are in time-share class
  - Scheduling delays on high context switch rates on busy systems may increase the variation in the cluster traffic times
  - More processes imply higher memory utilization and higher risk of paging
- How to control concurrent sessions:
  - Use connection pooling
  - Avoid connection storms (pool and process limits)
- Ensure that load is well-balanced over nodes

Application Considerations

Memory considerations part 1: optimize memory locally





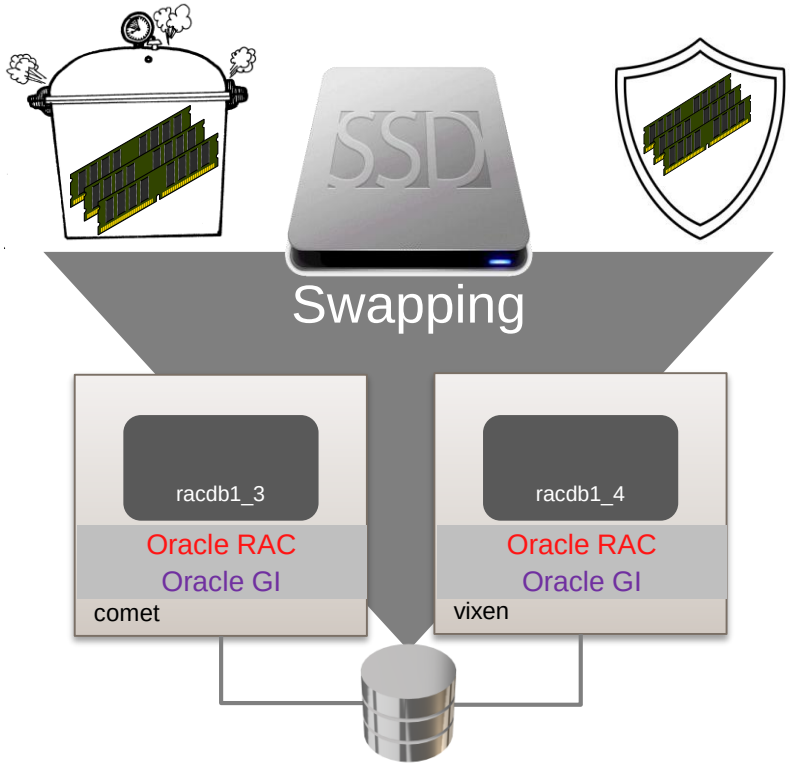


- Avoid memory pressure!
  - Paging and Swapping activity on one node affects performance on all nodes
  - **Severe Paging and Swapping activity on one node can cause instance evictions**
    - *#1 cause for service disruptions in clusters*

- **Use Memory Guard** 
  - QoS feature – available in monitoring only mode
  - Prevents new connections from coming in to a server that is already under memory pressure

Application Considerations

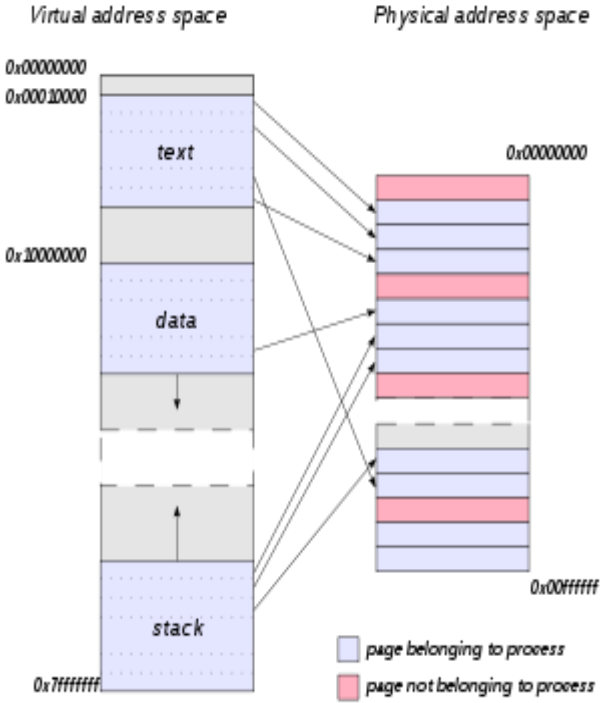
Memory considerations part 2: use Solid State Disks to host swap space



- Use Solid State Disks (SSDs) to host swap space in order to increase node availability
  - Memory pressure can cause node evictions.
  - Preventing memory pressure is the solution.
  - If prevention is not successful and swapping is performed by the Operating System (OS),
    - hosting the swap space can mitigate the impact that extensive swapping can have on cluster operations on the on the affected server(s).
- More information:
  - My Oracle Support Note Doc ID: 1671605.1 – “Use Solid State Disks to host swap space in order to increase node availability”

Application Considerations

Memory considerations part 3: configure Huge Pages for Oracle RAC

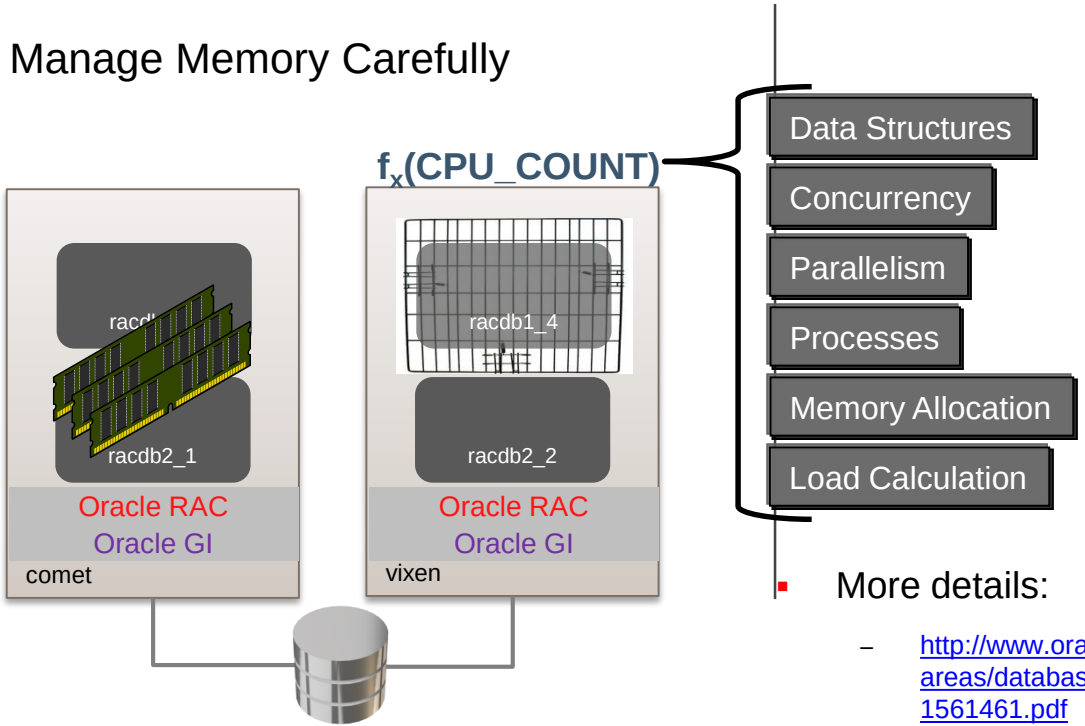


- Use Huge pages for SGA (Linux)
  - Dramatic reduction in memory for page tables
  - SGA pages pinned in memory
- More information:
  - My Oracle Support note 361323.1 – HugePages on Linux: What It Is... and What It Is Not...
  - My Oracle Support note 401749.1 – Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration
- Engineered systems provide templates for pre-configuration of huge pages for the SGA

Consolidation Tips and Tricks

What to consider when using more than one instance per server – part 1

1. Manage Memory Carefully



2. Use Instance Caging / set CPU_COUNT
3. Number of real-time processes needs to be taken into consideration

- More details:
  - http://www.oracle.com/technetwork/database/focus-areas/database-cloud/database-cons-best-practices-1561461.pdf