TECHNICAL WHITE PAPER – NOVEMBER 2021

Architecting Oracle Workloads on VMware Hybrid Multi-Clouds

BEST PRACTICES GUIDE

vmware®

Table of contents

Introduction . . . 5

Purpose . . . 5

Target Audience . . . 5

Technology Overview . . . 6

VMware vSphere . . . 6

VMware vSAN . . . 6

Oracle Database Architecture . . . 7

Oracle Memory Management . . . 7

Oracle Multitenant . . . 7

Oracle ASM, ASMLIB and ASMFD . . . 8

Linux Device Persistence and udev Rules . . . 9

Oracle Real Application Clusters (RAC) on VMware Platform . . . 9

Oracle Requirements Considerations . . . 11

Understand Oracle Workloads . . . 12

Measure Oracle Workloads . . . 12

Tools to Measure Oracle Workloads . . . 13

Migrate Oracle Workloads . . . 14

High Availability for Oracle Workloads on VMware Hybrid Clouds . . . 15

Business Continuity and Disaster Recovery for Oracle Workloads . . . 15

Monitor and Troubleshoot Oracle Workloads . . . 16

Automate Oracle Workloads Deployment using Database as a Service (DBaaS) . . . 17

General Approach to Best Practices for Oracle on VMware Hybrid Platform . . . 18

VM Best Practices for Oracle . . . 19

VM CPU Best Practices . . . 19

NUMA Considerations . . . 22

CPU Hot Plug and Hot Add . . . 23

CPU Hot Remove . . . 23

VM Memory Best Practices . . . 23

Memory Reservation . . . 23

Memory Hot Plug and Hot Add . . . 26

Memory Hot Remove . . . 26

Persistent Memory . . . 27

VM Storage Best Practices 28
Multiple PVSCSI Controllers and PVSCSI and VMDK Queue Depth 28
Virtual Disks Hot Add and Hot Remove 29
Virtual Disks Hot Extend 29
VM Network Best Practices 29
Interrupt Coalescing 29
Jumbo Frames and Virtual NIC MTU size 30
Load Balancing on VMware vSphere Standard Switch and vSphere Distributed Switch 31
VM Maintenance Best Practices 32
Disabling Time Synchronization 32
VMware Tools 32
In-Guest Best Practices 33
CPU Best Practices 33
OS NUMA 33
numactl Linux utility 34
Automatic NUMA Balancing 35
Memory Best Practices 36
HugeTLB Pages (Static and Traditional HugePages) 36
Transparent HugePages 36
Linux 1GB HugePages 37
Tuned Profile tuned-profiles-oracle 37
Memory Swapping 40
Linux kdump and kernel.panic_on_oops 41
Linux Out-of-Memory (OOM) 42
Storage Best Practices 43
I/O Scheduling 43
Partition Alignment 44
Multipathing 45
Enable SCSI_ID for devices 46
Maximum I/O Size for Block Devices 46
Network Best Practices 47
Enable Jumbo Frames for Oracle RAC Private Interconnect 47
Receive Side Scaling (RSS) and Multiqueue Support 47
TCP Segmentation Offload 48
Large Receive Offload 50

Oracle Best Practices . . . 52
CPU Best Practices . . . 52
Oracle NUMA . . . 52
Oracle CPU_COUNT & CPU_MIN_COUNT . . . 53
Memory Best Practices . . . 54
Oracle and Static and Traditional HugePages . . . 54
PRE_PAGE_SGA and LOCK_SGA Initialization Parameters . . . 56
GIMR and HugePages Memory . . . 56
Storage Best Practices . . . 56
Oracle Optimized Flexible Architecture (OFA) . . . 56
Oracle Database Layout Considerations . . . 57
Oracle ASM Considerations . . . 60
Oracle ASM Allocation Unit (AU) Considerations . . . 61
Oracle ASM SAME and Striping Considerations . . . 62
Oracle ASM 4k Blocksize . . . 63
Oracle ASM Maximum I/O Size . . . 64
Oracle ASM Rebalance and Compact Considerations . . . 64
Oracle ASM Thin Provisioning and Space Reclamation . . . 65
Oracle ASM Disks Online Resizing . . . 65
Oracle ASM Intelligent Data Placement and Physical Disk Placement . . . 66
Network Best Practices . . . 67
Summary . . . 68
Conclusion . . . 76
Resources . . . 76
Acknowledgments . . . 77

Introduction

Oracle is one of the most widely used database platforms in the world, with many organizations having deployed dozens or even hundreds of Oracle single-instances and Real Application Clusters (RAC) into their environments. The flexibility of Oracle, with its rich application capabilities and the relative low cost of x86 computing, has led to a wide variety of Oracle installations. These range from large data warehouses with business intelligence and reporting features to small, highly specialized departmental and application databases. Flexibility at the database layer translates directly into application flexibility, giving end users more useful application features and ultimately improving productivity.

Application flexibility often comes at a cost to operations. As the number of applications in the enterprise continues to grow, an increasing number of Oracle installations are brought under lifecycle management. Each application has its own set of requirements for the database layer, resulting in multiple versions, patch levels, and maintenance processes. For this reason, many application owners insist on having an Oracle installation dedicated to an application. As application workloads vary greatly, many Oracle installations are allocated more hardware resources than they need, while others are starved.

Recognizing this problem over many years, organizations are now virtualizing their most critical applications and embracing a virtualization first policy. This means applications are deployed on virtual machines (VMs) rather than on physical servers by default. Oracle is one of the most virtualized business critical applications (BCAs) of the past few years.

Virtualizing Oracle with VMware vSphere® allows for the best of both worlds, simultaneously optimizing compute resources through server consolidation and maintaining application flexibility through role isolation, taking advantage of the software-defined data center (SDDC) platform and capabilities such as network and storage virtualization. Oracle workloads can be migrated to new sets of hardware in their current states without expensive and error-prone application remediation, and without changing operating systems (OS), application versions or patch levels. For high performance databases, VMware has demonstrated the capabilities of vSphere to run the most challenging Oracle workloads.

Virtualizing Oracle with vSphere enables many additional benefits. For example, vSphere vMotion® enables seamless migration of VMs containing Oracle instances between physical servers and between data centers without interrupting users or their applications. vSphere Distributed Resource Scheduler™ can be used to dynamically balance Oracle workloads between physical servers. VMware vSphere® High Availability (HA) and VMware vSphere® Fault Tolerance (FT) provide simple and reliable protection for VMs containing Oracle and can be used in conjunction with Oracle's built-in HA capabilities. Among other features, VMware NSX® provides network virtualization and dynamic security policy enforcement. VMware Site Recovery Manager™ provides disaster recovery plan orchestration, VMware vRealize® Operations™ manager provides a comprehensive analytic and monitoring engine, and VMware Cloud™ on AWS can be consumed to take the advantages of public cloud. These are only a sampling of the many benefits VMware can provide for virtualized applications.

For many organizations, the question is no longer whether to virtualize Oracle. Rather, it is to determine the best architecture design for achieving business and technical requirements while keeping operational overhead and cost to a minimum.

Purpose

This document provides best practice guidelines for designing and implementing Oracle single-instances and Oracle RAC in VMs to run on vSphere. The recommendations are not specific to a particular hardware set nor to the size and scope of a particular Oracle implementation. The examples and considerations in this document provide guidance only and do not represent strict design requirements, as varying application requirements can result in many valid configuration possibilities.

The successful deployment of Oracle on VMware hybrid multi-clouds is not significantly different from deploying Oracle on physical servers. DBAs can fully leverage their current skillsets to enable the benefits associated with virtualization.

This guide assumes the reader is conversant with the basic administration of a typical VMware vSphere-based infrastructure. References to standard VMware vSphere administration documents are provided as appropriate.

Target Audience

This document assumes a knowledge and understanding of vSphere and Oracle. Architectural staff can use this document to understand how the system will work as a whole as they design and implement various components. Engineers and administrators can use this document as a catalog of technical capabilities. DBA staff can use this document to understand how Oracle can fit into a virtual infrastructure. Management staff and process owners can use this document to help model business processes and take advantage of the savings and operational efficiencies achieved with virtualization.

Technology Overview

This section provides an overview of the technologies used in this solution:

- VMware vSphere
- VMware vSAN™
- Oracle Database Architecture
- Oracle Memory Management
- Oracle Multitenant
- Oracle ASM, ASMLIB and ASMFD
- Linux Device Persistence and udev Rules
- Oracle Real Application Clusters (RAC) on VMware Platform

VMware vSphere

VMware vSphere, the industry-leading virtualization and cloud platform, is the efficient and secure platform for hybrid clouds, accelerating digital transformation by delivering simple and efficient management at scale, comprehensive built-in security, a universal application platform, and a seamless hybrid cloud experience. The result is a scalable, secure infrastructure that provides enhanced application performance and can be the foundation of any cloud.

As the next-generation infrastructure for next-generation applications, vSphere 7.0 has been rearchitected with native Kubernetes, enabling IT admins to use VMware vCenter Server® to operate Kubernetes clusters through namespaces. VMware vSphere with Tanzu allows IT admins to leverage their existing skillset to deliver self-service infrastructure access to their DevOps teams, while providing observability and troubleshooting of Kubernetes workloads. vSphere 7 provides an enterprise platform for both traditional and modern applications, enabling customers and partners to deliver a developer-ready infrastructure, scale without compromise, and simplify operations.

For further information, review:

VMware vSphere 7.0

VMware vSAN

VMware vSAN is a software-defined storage solution, built from the ground up, for vSphere VMs.

It abstracts and aggregates locally attached disks in a vSphere cluster to create a storage solution that can be provisioned and managed from vCenter and the vSphere client. vSAN is embedded within the hypervisor, hence storage and compute for VMs are delivered from the same x86 server platform running the hypervisor.

Hyperconverged infrastructure (HCI) backed by VMware vSAN provides a wide array of deployment options, from a two-node setup to a standard cluster supporting up to 64 hosts. Also, vSAN accommodates a stretched cluster topology to serve as an active-active disaster recovery solution. vSAN includes HCI Mesh, which allows customers to remotely mount a vSAN datastore to other vSAN clusters, disaggregating storage and compute. This allows greater flexibility to scale storage and compute independently.

For further information, review:

- *VMware vSAN*

Oracle Database Architecture

Oracle Database 19c, the latest generation of the world's most popular database, provides businesses of all sizes with access to the world's fastest, most scalable, and reliable database technology. These capabilities enable secure and cost-effective deployment of transactional and analytical workloads in the cloud, on-premises, and in hybrid cloud configurations.

For further information, review:

- *Oracle database architecture*

Oracle Memory Management

Oracle Database can either manage the SGA memory and instance PGA memory completely automatically OR they can be manually configured.

- With automatic memory management (AMM), Oracle Database dynamically exchanges memory between the SGA and PGA as needed to meet processing demands by specifying the target memory size (MEMORY_TARGET) and optional total memory size (MEMORY_MAX_TARGET) to be used by the instance SGA and PGA.
- With the manual memory management, one can exercise more direct control over the sizes of individual memory components by configuring the database for manual memory management, which can be achieved in one of two ways:
  - Automatic shared memory management (ASMM), in which SGA target (sga_target) and SGA maximum size (sga_max_size) is specified and the database then dynamically tunes the sizes of many SGA components
  - Manual shared memory management (MSM), in which we set the sizes of several individual SGA components, thereby determining the overall SGA size and then we manually tune these individual SGA components on an ongoing basis

  For the instance PGA, the sizing can be done in one of two ways:

  - Automatic PGA memory management, in which the target size for the instance PGA (pga_agregate_target) is set up and then the database then sets the size of the instance PGA to the target and dynamically tunes the sizes of individual PGAs.
  - Manual PGA memory management, in which the maximum work area size for each type of SQL operator (such as sort or hash-join) is set. This memory management method, although supported, is not recommended

PGA_AGGREGATE_TARGET specifies the target aggregate PGA memory available to all server processes attached to the instance. PGA_AGGREGATE_LIMIT specifies a limit on the aggregate PGA memory consumed by the instance. PGA_AGGREGATE_TARGET is a soft limit to control PGA memory allocated to work areas whereas PGA_AGGREGATE_LIMIT puts a hard limit on the PGA usage.

For further information, review:

- *Managing Memory*

Oracle Multitenant

Multitenant architecture enables an Oracle database to function as a multitenant container database (CDB).

A *CDB* includes zero, one, or many customer-created pluggable databases (PDBs). A *PDB* is a portable collection of schemas, schema objects, and nonschema objects that appears to an Oracle Net client as a *non-CDB*. All Oracle databases before Oracle Database 12c were non-CDBs.

The many pluggable databases in a single multitenant container database share their memory and background processes, letting you operate many more pluggable databases on a particular platform than single databases that use the old architecture could be operated.

Oracle Multitenant enables you to consolidate data and code without altering existing schemas or applications and is a database consolidation feature.

The best practices for setting up Oracle 12c multitenant architecture are no different than those recommended for setup of Oracle 11g. The key point to consider is when deploying a multitenant architecture, memory, CPU, and IOPS requirements will increase based on individual tenant requirements, which must be provided for individually and adequately.

For further information, review:

- *Introduction to the Multitenant Architecture*

Oracle ASM, ASMLIB and ASMFD

ASM

Oracle Automatic Storage Management (ASM) is a volume manager and a file system for Oracle database files that supports single-instance Oracle Database and Oracle RAC configurations.

Oracle ASM is Oracle's recommended storage-management solution that can be used for both Oracle RAC and single-instance Oracle databases, providing an alternative to conventional volume managers, file systems, and raw devices.

ASM offers many advantages over the traditional Linux storage management solution such as logical volume manager (LVM). The main benefits include the following:

- Automatic file management
- Online data files rebalancing across ASM disks
- Online addition and removal of ASM disks without downtime
- Single solution for both volume and file management that is integrated with Oracle software
- Improved I/O performance because ASM stripes all files across all disks in a disk group

There are multiple types of disk groups based on the Oracle ASM redundancy level. The table below lists the types with their supported and default mirroring levels. The default mirroring levels indicate the mirroring level with which each file is created unless a different mirroring level is designated.

DISK GROUP TYPE	SUPPORTED MIRRORING LEVELS	DEFAULT MIRRORING LEVEL
EXTERNAL redundancy	Unprotected (none)	Unprotected
NORMAL redundancy	Two-way, three-way, unprotected (none)	Two-way
HIGH redundancy	Three-way	Three-way
FLEX redundancy	Two-way, three-way, unprotected (none)	Two-way (newly-created)
EXTENDED redundancy	Two-way, three-way, unprotected (none)	Two-way

TABLE 1. Table 4-2 Mirroring options for Oracle ASM disk group types

For further information, review:

- *Oracle Automatic Storage management (ASM)*

ASMLIB

Oracle ASMLIB maintains permissions and disk labels that are persistent on the storage device so that the label is available even after an operating system upgrade.

The Oracle ASMLIB driver simplifies the configuration and management of block disk devices by eliminating the need to rebind block disk devices used with Oracle ASM each time the system is restarted.

With Oracle ASMLIB, you define the range of disks you want to make available as Oracle ASM disks. Oracle ASMLIB maintains permissions and disk labels that are persistent on the storage device, so that the label is available even after an operating system upgrade

For further information, review:

- *About Oracle ASM with Oracle ASMLIB*

ASMFD

Oracle ASM Filter Driver (Oracle ASMFD) is a kernel module that resides in the I/O path of Oracle ASM disks. Oracle ASM uses the filter driver to validate write I/O requests to Oracle ASM disks.

Oracle ASMFD simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.

Oracle ASM Filter Driver rejects any I/O requests that are invalid. This action eliminates accidental overwrites of Oracle ASM disks that would cause corruption in the disks and files within the disk group. For example, the Oracle ASM filter driver filters out all non-Oracle I/Os which could cause accidental overwrites.

Oracle ASM Filter Driver (Oracle ASMFD) is available on Linux systems starting with Oracle Database 12c Release 1 (12.1.0.2)

For further information, review:

- *Administering Oracle ASM Filter Driver*

Linux Device Persistence and udev Rules

Device names in Linux are not guaranteed persistent across reboots. A device (e.g., /dev/sdb) can be renamed on the next reboot (e.g., /dev/sdc).

Linux udev rules may be used to guarantee device persistence across reboot.

For further information, review:

- *Configuring device persistence for Oracle storage*

Oracle Real Application Clusters (RAC) on VMware Platform

Oracle Clusterware is portable cluster software that provides comprehensive multi-tiered high availability and resource management for consolidated environments. It supports clustering of independent servers so that they cooperate as a single system.

Oracle Clusterware is the integrated foundation for Oracle Real Application Clusters (RAC), and the high-availability and resource-management framework for all applications on any major platform.

There are two key requirements for Oracle RAC:

- Shared storage
- Multicast Layer 2 networking

These requirements are fully addressed when running Oracle RAC on VMware vSphere, as both shared storage and Layer 2 networking are natively supported.

VMware vSphere HA clusters enable a collection of VMware ESXi™ hosts to work together so that, as a group, they provide higher infrastructure-level availability for VMs than each ESXi host can provide individually.

VMware vSphere HA provides high availability for VMs by pooling the VMs and the hosts on which they reside into a cluster. Hosts in the cluster are monitored and in the event of a failure, the VMs on a failed host are restarted on alternate hosts.

When creating a vSphere HA cluster, a single host is automatically elected as the master host. The master host communicates with VMware vCenter Server® and monitors the state of all protected VMs and of the slave hosts.

Oracle RAC and VMware HA solutions are complementary to each other. Running Oracle RAC on a VMware platform provides the application-level HA enabled by Oracle RAC, in addition to the infrastructure-level HA enabled by VMware vSphere.

Extended Oracle RAC provides greater availability than local Oracle RAC. It provides extremely fast recovery from a site failure and enables all servers, in all sites, to actively process transactions as part of a single database cluster.

Running Extended Oracle RAC on a stretched cluster architecture provides the same advantages as traditional Oracle RAC, across data centers and sites, in addition to the site-level protection enabled by stretched cluster architecture.

For further information, review:

- *Oracle Clusterware 19c*
- *VMware vSphere HA*
- *Oracle VMware Hybrid Cloud High Availability Guide*
- *Extended Oracle RAC on vSphere Metro Storage Cluster using Storage Virtualization*
- *Extended Oracle RAC across sites with VMware NSX*
- *Oracle Real Application Clusters on VMware vSAN*

Oracle Requirements Considerations

When considering Oracle deployments as candidates for virtualization, you need a clear understanding of the business and technical requirements for each database instance. These requirements span multiple dimensions, such as availability, performance, scalability, growth and headroom, patching, and backups.

Use the following high-level procedure to simplify the process for characterizing Oracle candidates for virtualization:

- Understand the performance characteristics and growth patterns of the workloads associated with the applications accessing Oracle.
- Understand availability and recovery requirements, including uptime guarantees and disaster recovery for both the VM and the databases.
- Capture resource utilization baselines for existing physical server hosting databases.
- Plan the migration and deployment to vSphere.

With a few subtleties, the Oracle workload on VMware hybrid platform deployment and configuration methodology is no different than that applied to migrating Oracle workloads from one platform (e.g., big endian/little-endian) to another (e.g., big endian/little-endian) as part of a replatforming exercise or hardware refresh cycle.

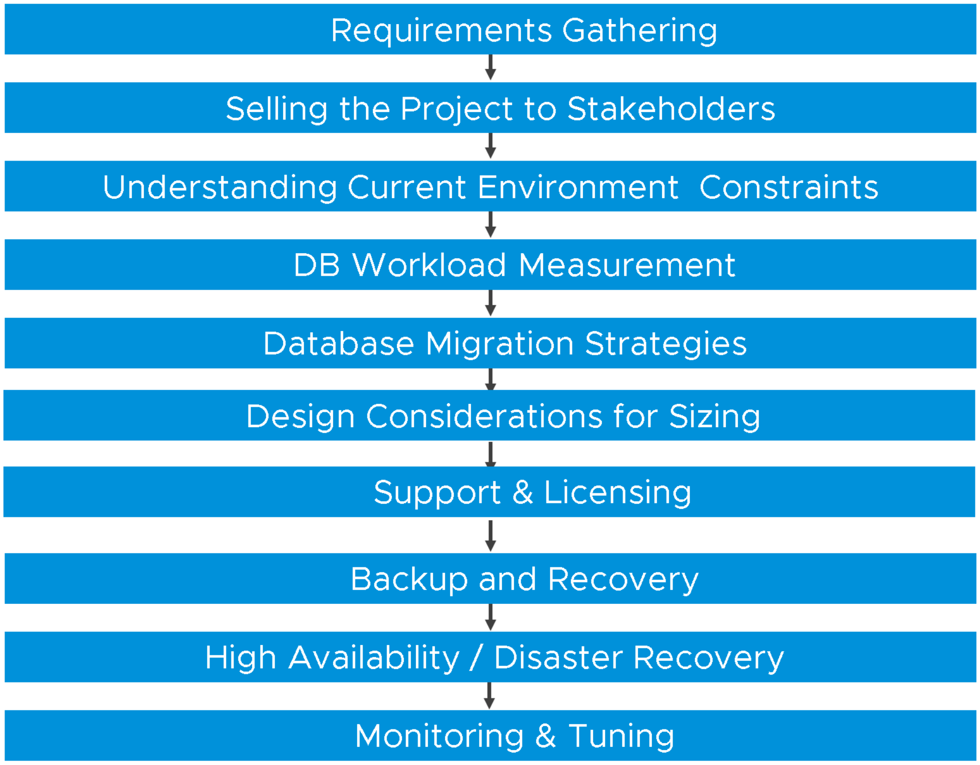

Requirements Gathering
Selling the Project to Stakeholders
Understanding Current Environment Constraints
DB Workload Measurement
Database Migration Strategies
Design Considerations for Sizing
Support & Licensing
Backup and Recovery
High Availability / Disaster Recovery
Monitoring & Tuning

FIGURE 1. Oracle Workload Design Methodology

Understand Oracle Workloads

The application workloads that run against Oracle single-instance and Oracle RAC databases can have different characteristics that influence deployment and other factors, such as feature usage or the availability architecture. These factors influence characteristics like how VMs are laid out on VMware ESXi hosts, as well as the underlying disk configuration.

Before deploying Oracle instances on VMware hybrid multi-clouds, one must understand the business requirements (e.g., service level agreements [SLAs], recovery time objectives [RTOs], recovery point objectives [RPOs]) and the application workload for the Oracle deployments you intend to support.

Each application has different requirements for availability, performance, and capacity. Consequently, each deployment must be designed to optimally support those requirements. Many organizations classify Oracle installations into multiple management tiers based on SLAs, RPOs, and RTOs. The classification of the type of workload an Oracle database runs often dictates the architecture and resources allocated to it. The following are some common examples of workload types.

Mixing application workload types in a VM, whether they are multiple standalone single-instance container databases with pluggable databases OR RAC container databases with pluggable databases, is not recommended unless careful workload analysis is done to determine that running these workloads simultaneously in the same VM will not cause contention.

Typical workload types include:

- Online transaction processing (OLTP) databases are often the most critical databases in an organization. These databases usually back customer-facing applications and are considered essential to the company's core operations. Mission-critical OLTP databases and the applications they support often have SLAs that require very high levels of performance and are very sensitive for performance degradation and availability. Oracle VMs running OLTP mission-critical databases might require more careful resource allocation (e.g., central processor unit [CPU], memory, disk, and network) to achieve optimal performance. They might also be candidates for clustering with Oracle RAC. These types of databases are usually characterized with mostly intensive random reads and writes to disk and sustained CPU utilization during working hours.
- Decision support systems (DSS) databases can be also referred to as data warehouses. These are mission-critical in many organizations that rely on analytics for their business. These databases are very sensitive to CPU utilization and sequential read and write operations from disk when queries are being run. In many organizations, DSS databases are the most critical resource at month, quarter, and year end.
- Batch, reporting services, and ETL databases are busy only during specific periods for such tasks as reporting, batch jobs, and application integration of ETL workloads. These databases and applications may be essential to your company's operations, but they have much less stringent requirements for performance and availability. They may, nonetheless, have other very stringent business requirements, such as data validation and audit trails.
- Other smaller, lightly used databases typically support departmental applications that may not adversely affect your company's real-time operations in the event of an outage. It's often possible to tolerate these types of databases and applications being down for extended periods.

Resource needs for Oracle deployments are defined in terms of CPU, memory, disk and network I/O, user connections, transaction throughput, query execution efficiency and latencies, and database size. Some customers have established targets for system utilization on hosts running Oracle (e.g., 80 percent CPU utilization) leaving enough headroom for any usage spikes or availability.

Understanding database workloads and how to allocate resources to meet service levels helps you to define appropriate VM configurations for individual Oracle databases. Because you can consolidate multiple workloads on a single vSphere host, this characterization also helps you to design a vSphere and storage hardware configuration that provides the resources you need to deploy multiple workloads successfully on vSphere.

Measure Oracle Workloads

A best practice is to perform an analysis of workloads running against single-instance and Oracle RAC databases to determine current utilization in terms of CPU, memory, and disk and network I/O to understand performance characteristics and growth patterns. This helps with the any replatforming exercise and capacity planning.

The above recommendation is independent of the architecture, physical or virtual, and holds strong whether the effort is part of a:

- Hardware refresh – replacing old servers with newer generation servers
- Replatform – moving from big-endian systems (e.g., Solaris, HP-UX, AIX) to little-endian (e.g., x86)
- Architecture change – physical architecture to VMware vSphere platform

Owing to lack of resources or time constraints, customers often may simply move their Oracle workloads from one platform to another (i.e., hardware refresh, replatform or an architecture change). This is considered a lift-and-shift approach.

Once the production workload is stabilized on the target platform, the customer will deploy monitoring tools to check current utilization based on the new platform and then reduce resources allocated to Oracle workloads based on the current utilization. This often leads to a waste of hardware resources, which now have to be repurposed for other projects.

The ideal process is to perform a peak periods analysis of workloads running against single-instance and Oracle RAC databases to determine current utilization in terms of CPU, memory, disk and network I/O to understand performance issues and facilitate future capacity planning. Due diligence in reviewing these metrics helps ensure the target platform is architected appropriately.

Tools to Measure Oracle Workloads

Choosing the right tools is essential to successfully measure Oracle workloads. Customers often run synthetic load generators (e.g., SwingBench, HammerDB) to perform benchmark and load testing against an Oracle database. While these tools offer an understanding of how the system performs under these synthetic loads, they are not an accurate measurement of how the real workload will perform in production.

It’s recommended to capture and run a real production workload against the target system to understand how the system performs. Oracle *Real Application Testing* (RAT) can be used to capture and replay the workload against an Oracle database.

Load testing can be run against different components of the Oracle database:

- IO Subsystem (e.g. *Oracle ORION*, *IOMeter*, *Linux FIO*)
- Oracle database (e.g. *SLOB*, *Oracle Database I/O calibration*)
- Application testing tools (e.g., *Swingbench*, *HammerDB*, *Oracle Real Application Testing*)

The choice of one of the above tools depends on what layer we are testing, and correct expectations of the metrics generated.

The workload utilization capture tools that follow help capture an Oracle workload in order to understand system characteristics and workload utilization:

- Hardware
  - **Standard Performance Evaluation Corporation** (*SPEC*) benchmarks provide standardized benchmarks and tools to evaluate performance and energy efficiency for the newest generation of computing systems.
- Overall Environment
  - *Live Optics* is a free online software one can use to collect, visualize, and share data about workload environment and utilization, thereby helping understand workload performance. It is recommended to use Live Optics on a simulated production system during at least a 24-hour period to include the peak workload. This simulation helps define I/O requirements.
- Operating System
  - The **Linux System Activity Report** (*SAR*) command writes to standard output the contents of selected cumulative activity counters in the operating system. The sar data should be collected for the same time period as the below Oracle Automatic Workload Repository (AWR) reports time frame to understand the OS metrics in detail at the time the workload was run. For Windows, Perfmon can be used to gather Windows OS statistics.
- Oracle
  - Oracle Automatic Workload Repository (AWR) stores performance related statistics for an Oracle database. The Automatic Database Diagnostic Monitor (ADDM) is a diagnostic tool that analyzes the AWR data on a regular basis, locates root causes of any performance problems, provides recommendations for correcting the problems, and identifies non-problem areas of the system. Because AWR is a repository of historical performance data, ADDM can analyze performance issues after the event, often saving time and resources in reproducing a problem.
  - The STATISTICS_LEVEL initialization parameter must be set to the TYPICAL or ALL to enable ADDM. The default setting is TYPICAL. Setting STATISTICS_LEVEL to BASIC disables many Oracles Database features, including ADDM, and is strongly discouraged.

Other useful tools to understand the operating system and Oracle ASM, RDBMS and Clusterware details are:

- *Linux sosreport* is a tool that collects configuration details, system information and diagnostic information from a Linux system (e.g., RedHat [RHEL], Oracle Enterprise Linux [OEL]).

  For any database troubleshooting issue, the Oracle AWR and Linux sosreport are collected as part of the diagnosis phase. To run sosreport, the sos package must be installed. The package is part of the default group and will be installed automatically on most systems. After the package has been installed, issue the following command:

  ```
  #sosreport
  ```

  You can also use sosreport to enable diagnostics and analytical functions. To assist in troubleshooting a problem, sosreport records the information in a compressed file that you can send to a support representative.

  For further information, review *What is a sosreport and how to create one in Red Hat Enterprise Linux?*

- **Oracle Remote Diagnostic Agent** (*RDA*) is a data collection framework, driven from the command line, which can be used to gather diagnostic information related to Oracle products and their environments. This data is used to assist in problem diagnosis. For further information, review *My Oracle Support Doc ID Remote Diagnostic Agent (RDA) - Getting Started (Doc ID 314422.1)*

Migrate Oracle Workloads

Understanding the performance characteristics and growth patters or workloads on the current platform, as well as the resource characteristics and constraints of the target platform, the tools referenced above enable migration of Oracle workloads to vSphere in several ways. An important aspect of the migration process is the endianness of the source and target platform, primarily expressed as big-endian (BE) or little-endian (LE).

A big-endian system stores the most significant byte of a word at the smallest memory address and the least significant byte at the largest. A little-endian system, in contrast, stores the least-significant byte at the smallest address.

Learn more about *endianness*.

For Oracle workloads, changing platforms requires the data be moved using one of the following methods:

- Export or import to include the use of Datapump facilities. All versions support export and import, however Datapump 10.1.0.2 or higher is required
- Transportable tablespaces 10G or later
- RMAN convert database functions. 10G or Later
- RMAN duplicate
- Streams replication
- Create table as select (CTAS)
- Oracle Dataguard heterogeneous primary and physical standbys
- Oracle GoldenGate

Learn more about Oracle data migration across platforms at *My Oracle Support Doc ID Migration Of An Oracle Database Across OS Platforms (Generic Platform) (Doc ID 733205.1)*.

In addition to the above tools, VMware provides tools that can help migrate workloads on a physical x86 platform to a VMware platform:

- *VMware vCenter Converter* converts local and remote physical machines into VMs without any downtime.
- **VMware HCX OS Assisted Migration** (*OSAM*) supports migration of VMs running non-vSphere guest operating systems in Linux or Windows environments.

High Availability for Oracle Workloads on VMware Hybrid Clouds

Deploying Oracle RAC on physical architecture is subjected to challenges like those running Oracle non-RAC on physical architecture. These challenges include, but are not exclusive to, hardware failure due to a failed component, power outage, and complete hardware meltdown.

Providing high availability in these environments presents a significant challenge for business organizations. Hardware issues negate the inherent value proposition of Oracle RAC, which is to provide application-level high availability with sustained infrastructure high availability.

With VMware vSphere, customers have successfully run business-critical, high performance-demanding Oracle workloads for many years. VMware vSphere provides high availability natively at the infrastructure level and is completely complementary to the application-level high availability that Oracle RAC provides.

Extended Oracle RAC provides greater availability than local Oracle RAC. It provides extremely fast recovery from a site failure and enables all servers, in all sites, to actively process transactions as part of a single database cluster. Extended Oracle RAC enables transparent workload sharing, workload balancing, site maintenance without service disruption, and high availability across sites.

Oracle VMware Hybrid Cloud High Availability Guide – REFERENCE ARCHITECTURE describes the configuration and deployment of traditional Oracle RAC on VMware vSphere and VMware vSAN. The paper discusses configuration and deployment of Extended Oracle RAC on VMware vSAN Stretched Cluster and VMware vSphere Metro Storage Cluster.

For further information, review:

- *Oracle VMware Hybrid Cloud High Availability Guide – REFERENCE ARCHITECTURE*

Business Continuity and Disaster Recovery for Oracle Workloads

Customers have successfully run their business-critical Oracle workloads with high-performance demands on VMware vSphere for many years. Virtualization of mission-critical databases adds layers of complexity to the infrastructure, however, making common operations like backup and recovery, cloning, disaster recovery and other day-to-day activities difficult. The most efficient storage operations for mission-critical databases are offloaded to the storage array.

Concerns that often delay virtualization of business-critical database workloads include:

- Rapid database growth and the need to reduce backup windows to meet performance and business SLAs
- The size of modern databases makes it harder to regularly clone and refresh data from production to QA and other environments
- Correct choice of business continuity plan to ensure rapid recovery from significant disruption to the operations
- Correct choice of disaster recovery technology to ensure business needs of RTO and RPO are met

A business continuity plan is a detailed strategy and set of systems for ensuring an organization's ability to prevent or rapidly recover from a significant disruption to its operations. The plan is essentially a playbook for how any type of organization will continue its day-to-day business during a disaster scenario or otherwise abnormal conditions.

Disaster recovery (DR) is an organization's method of regaining access and functionality to its IT infrastructure after events like a natural disaster, cyber-attack, or even business disruptions related to the COVID-19 pandemic. DR is one aspect of business continuity. Disaster recovery relies upon the replication of data and computer processing in an off-premises location not affected by the disaster.

Oracle Business Continuity and Disaster Recovery on VMware Hybrid Multi-Clouds - Reference Architecture describes the configuration and implementation of various business continuity and disaster recovery options across the application, VMware platform, and storage levels of Oracle single-instance and RAC workloads on the VMware vSphere hybrid multi-cloud platform. This includes on-premises and VMware clouds, with an emphasis on VMware Cloud on AWS.

For further information, review:

- *Oracle Business Continuity and Disaster Recovery on VMware Hybrid Multi-Clouds - Reference Architecture*

Monitor and Troubleshoot Oracle Workloads

Choosing the right tool for monitoring Oracle workloads in production on the VMware platform is essential to ensuring stringent SLAs, RTOs and RPOs are met. Customers deploy a number of tools to monitor Oracle workloads on VMware platform to troubleshoot and identify performance-related issues as they occur.

Some of these tools include:

- Oracle
  - *Dictionary tables (v$ / x$)* tables to see real time database statistics
  - The Automatic Workload Repository (AWR) stores performance related statics for an Oracle database. The Automatic Database Diagnostic Monitor (ADDM) is a diagnostic tool that analyzes AWR data on a regular basis, locates root causes of any performance problems, provides recommendations for correcting the problems, and identifies non-problem areas of the system. Because AWR is a repository of historical performance data, ADDM can analyze performance issues after the event, often saving time and resources in reproducing a problem.
  - *Oracle Autonomous Health Framework* (AHF) found at *My Oracle Support Doc ID Autonomous Health Framework (AHF) - Including TFA and ORAchk/EXAchk (Doc ID 2550798.1)* which includes many tools (e.g., ORAchk, oswatcher)
  - *Oracle Enterprise Manager* (OEM) is a management platform that provides a single dashboard to manage all Oracle deployments

- Operating System
  - Linux **System Activity Report** (sar) for OS related metrics.
  - The Linux **sosreport** (RHEL/CentOS) command is a tool that collects configuration and diagnostic information from a Linux system. For instance, the running kernel version, loaded modules, and system and service configuration files. The command also runs external programs to collect further information and stores this output in the resulting archive. For any database troubleshooting issue, the Oracle AWR and Linux sosreport are collected as part of the diagnosis phase.

    To run sosreport, the sos package must be installed. The package is part of the default group and will be installed automatically on most systems. After the package has been installed, issue the following command:

    ```
    #sosreport
    ```

    You can also use sosreport to enable diagnostics and analytical functions. To assist in troubleshooting a problem, sosreport records the information in a compressed file that you can send to a support representative.

- Virtual Machine
  - *VMware vCenter Statistics* to see metrics across the VMware platform
  - *VMware esxtop and resxtop* command-line utilities provide a detailed look at how ESXi uses resources in real time. You can start either utility in one of three modes: interactive (default), batch, or replay.
  - *VMware vRealize® Operations Manager™* – Maintaining and operating virtualized Oracle is the vital part of the infrastructure lifecycle. It's very important that the solution architecture already includes all necessary steps to ensure proper operations of the environment.

    For virtualized Oracle, consider the following requirements for a monitoring tool:

    - Ability to provide end-to-end monitoring from database objects through VM back to the physical hosts and storage in use
    - Ability to maintain, visualize and dynamically adjust the relationships between the components of the solution
    - Ability to maintain mid- and long-term time-series data
    - Ability to collect the data from virtualized and non-virtualized Oracle instances

    VMware vRealize Operations Manager is one of the tools able to fulfil all the requirements mentioned above when combined with vital extensions–VMware vRealize® True Visibility™ Management Pack for Oracle Enterprise Manager and Oracle Database.

When performance or capacity problems arise in your Oracle environment, vRealize Operations Manager can analyze metrics from the application all the way through to the infrastructure to provide insight into problematic components, whether they are compute (physical or virtual), storage, networking, OS, or application related. By establishing trends over time, vRealize Operations Manager can minimize false alerts and proactively alert on the potential root cause of increasing performance problems before end users are impacted.

– *VMware vRealize True Visibility Suite* provides the ability to create a cohesive monitoring strategy and centralized operations management with an end-to-end unified view from data center to applications. True Visibility Suite uses unified data analytics to monitor heterogeneous environments within vRealize Operations, imparting context and data depth and streamlining root-cause analysis.

The VMware vRealize True Visibility Management Pack for Oracle Enterprise Manager is an embedded adapter for vRealize Operations, collecting performance data from your Oracle Enterprise Manager systems and providing predictive analytics and real-time information about problems in your infrastructure—all within the vRealize Operations user interface.

The VMware vRealize True Visibility Management Pack for Oracle Database is an embedded adapter for vRealize Operations, collecting performance and capacity data from your Oracle Database environment and providing predictive analytics and real-time information about problems in your infrastructure—all within the vRealize Operations user interface.

For further information, review:

- *Oracle Automatic Workload Repository (AWR) with ADDM*
- *Linux System Activity Report (SAR)*
- *What is a sosreport and how to create one in Red Hat Enterprise Linux?*
- *Collecting diagnostic information for Linux operating systems (2032614)*
- *Interpreting esxtop Statistics*
- *Troubleshooting ESX/ESXi virtual machine performance issues (2001003)*
- *VMware vRealize True Visibility Suite*

Automate Oracle Workloads Deployment using Database as a Service (DBaaS)

Deploying and managing an Oracle Database environment is not a trivial undertaking. Oracle infrastructure tends to have stringent performance, business continuity, and backup and recovery requirements. To support these requirements, IT departments often are tasked with maintaining sprawling infrastructure for production, disaster recovery, backup, quality assurance, test, training, development, and sandbox.

Enterprise customers have identified database-as-a-service (DBaaS) as a key initiative that will enable higher levels of agility via rapid application development. The idea is to allow database consumers, such as application developers, testers, and architects, to provision databases easily using an on-demand, self-service platform. DBaaS will greatly simplify the deployment and management of a robust Oracle environment, while delivering higher levels of efficiency and flexibility for IT departments throughout the application lifecycle—implementation, migrations, consolidations, upgrades, and ongoing maintenance.

For further information, review:

- *Deploying Database-as-a-Service with Pure Storage and VMware*
- *Moving away from DBA-as-a-Service to Database as a Service (DBaaS) for Oracle Workloads on VMware Hybrid Platform*
- *Database-as-a-Service (DBaaS) Reference Architecture with VMware and Tintri*
- *Accelerating Database-as-a-Service(DBaaS) with VMware vRealize Automation and EMC XtremIO All-Flash Array*
- *Enabling Oracle EM12c-based Database-as-a-Service on VMware with the Blue Medora Plugin for VMware*
- *DBaaS with Oracle 12c, EMC XtremeIO and VMware (vCAC)*
- *DBaaS at EMC using vRealize Suite*
- *A quick reference to VVols (Virtual Volumes)*

General Approach to Best Practices for Oracle on VMware Hybrid Platform

VMware has created separate best practice documents for compute, storage, networking, and performance. Review the latest *Performance Best Practices for VMware vSphere*.

The VMware vSphere guide *Architecting Business Critical Applications on VMware Hybrid Multi-Clouds* provides best practice general guidelines for designing and implementing business-critical application (BCA) workloads on VMware vSphere platforms. The recommendations are not specific to a particular hardware set, nor to the size and scope of a particular BCA workload implementation. The examples and considerations in this document provide guidance only and do not represent strict design requirements, as varying application requirements can result in many valid configuration possibilities.

It's recommended to read the guides below in order they are listed before reviewing the current guide to effectively designing and implementing business-critical Oracle Workloads on the VMware hybrid cloud platform.

- *Performance Best Practices for VMware vSphere*
- *Architecting Business Critical Applications on VMware Hybrid Multi-Clouds*

The sections in the Oracle Best Practices paper draw from the relevant content in *Architecting Business Critical Applications on VMware Hybrid Multi-Clouds*.

The approach described above is summarized in the illustration below:

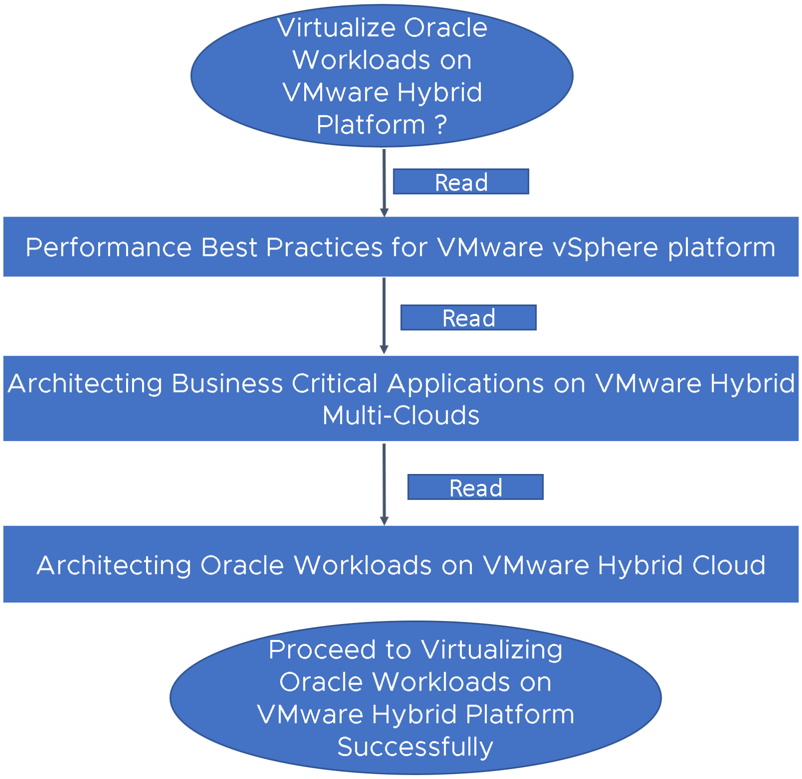

Virtualize Oracle Workloads on VMware Hybrid Platform ?
Read
Performance Best Practices for VMware vSphere platform
Read
Architecting Business Critical Applications on VMware Hybrid Multi-Clouds
Read
Architecting Oracle Workloads on VMware Hybrid Cloud
Proceed to Virtualizing Oracle Workloads on VMware Hybrid Platform Successfully

FIGURE 2. Oracle on VMware Hybrid Cloud Platform – Best Practices

VM Best Practices for Oracle

As a common sizing practice for physical servers, rightsizing also applies when deploying a VM. As its name implies, rightsizing involves the allocation of compute resources (e.g., CPUs, RAM) in only the amounts needed to power the database workload as opposed to allocating more than will be actively utilized.

Rightsizing is imperative when sizing VMs, however the approach to rightsizing a VM differs from that applied to physical server.

It's important to begin with a thorough understanding of your workload. Database server utilization varies widely by application. If the application is commercial, follow published guidelines where appropriate. If the application is custom-written, work with the application developers to determine resource requirements.

By following this guideline, you can gauge performance and utilization within the environment and identify potential excess capacity that could be used for additional workloads. For example, if the physical server on which various BCA workloads currently run equates to 16 physical CPU cores, avoid allocating more than 16 virtual vCPUs for the VMs on that vSphere host during the initial virtualization effort.

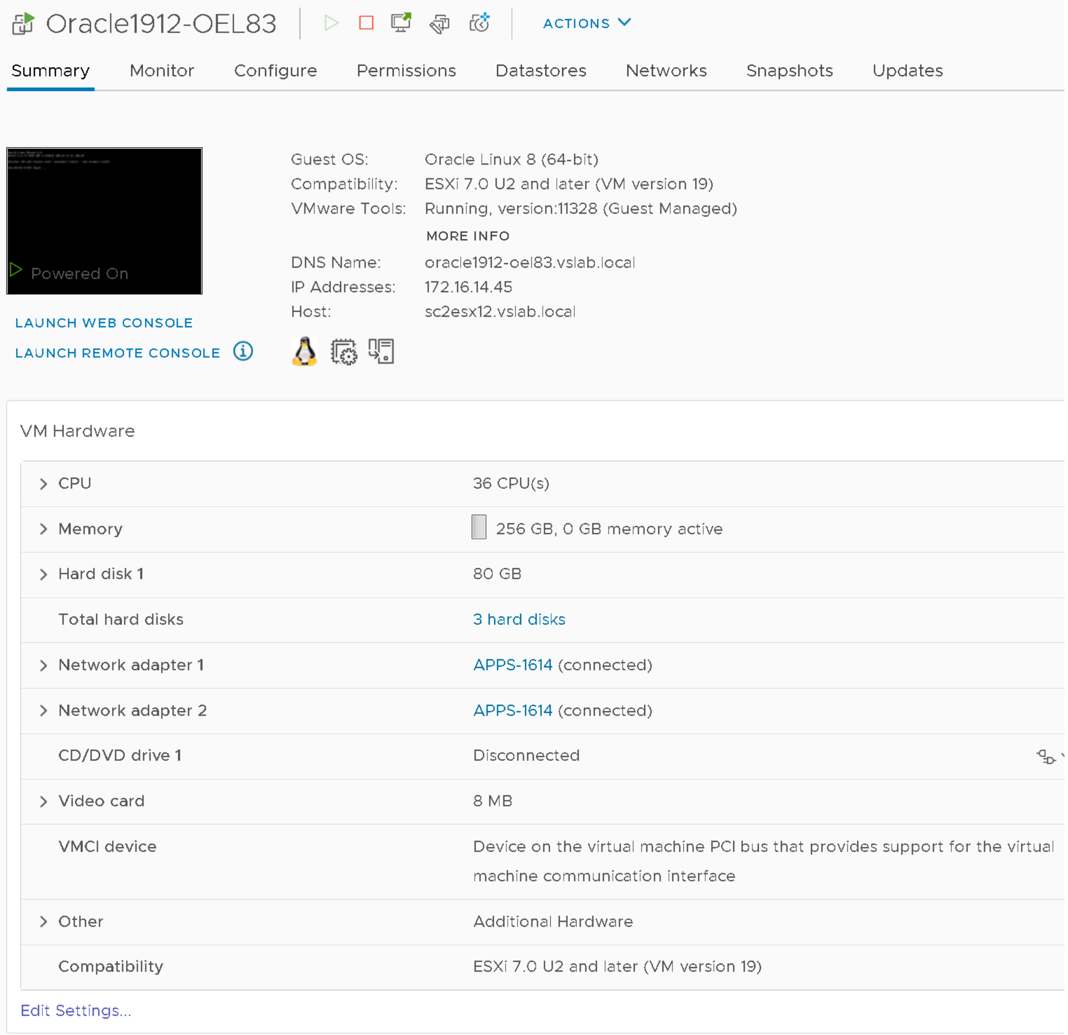
Taking a more conservative sizing approach helps rule out CPU resource contention as a possible contributing factor in the event of sub-optimal performance. After identifying excess capacity, you can consider increasing density by adding more workloads into the vSphere cluster and allocating virtual vCPUs beyond the available physical cores. Consider using monitoring tools capable to collect, store and analyze mid- and long-terms data ranges.

Lower-tier BCA workloads are typically less latency-sensitive so, in general, the goal is to maximize use of system resources and achieve higher consolidation ratios rather than maximize performance. The vSphere CPU scheduler's policy is tuned to balance between maximum throughput and fairness between VMs. For lower-tier databases, a reasonable CPU overcommitment can increase overall system throughput, maximize license savings, and continue to maintain adequate performance.

VM CPU Best Practices

For illustrations purposes, VM **Oracle1912-OEL83** is used in all of the sections that follow.

- VM **Oracle1912-OEL83** is created with 36vCPUs and 256GB vRAM
- VM is hosted on a two-socket, 24 cores-per-socket server with 384GB RAM and HT enabled
- OS is OEL 8.3 UEK with Oracle Grid Infrastructure and RDBMS binaries version 19.12.0.0.0



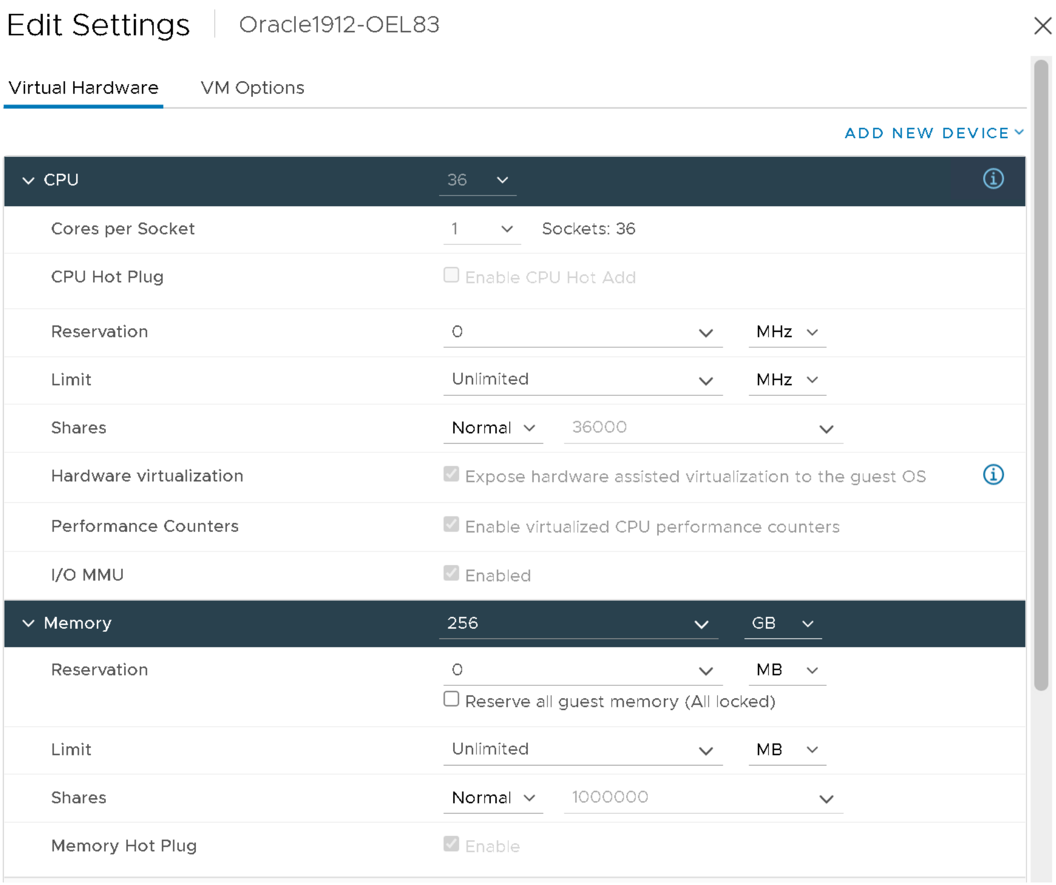
Guest OS: Oracle Linux 8 (64-bit)
Compatibility: ESXi 7.0 U2 and later (VM version 19)
VMware Tools: Running, version:11328 (Guest Managed)
DNS Name: oracle1912-oel83.vslab.local
IP Addresses: 172.16.14.45
Host: sc2esx12.vslab.local

VM Hardware

CPU	36 CPU(s)
Memory	256 GB, 0 GB memory active
Hard disk 1	80 GB
Total hard disks	3 hard disks
Network adapter 1	APPS-1614 (connected)
Network adapter 2	APPS-1614 (connected)
CD/DVD drive 1	Disconnected
Video card	8 MB
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
Other	Additional Hardware
Compatibility	ESXi 7.0 U2 and later (VM version 19)

FIGURE 3. VM Oracle1912-OEL83 Summary

The cores per socket setting for the VM is left at the default.



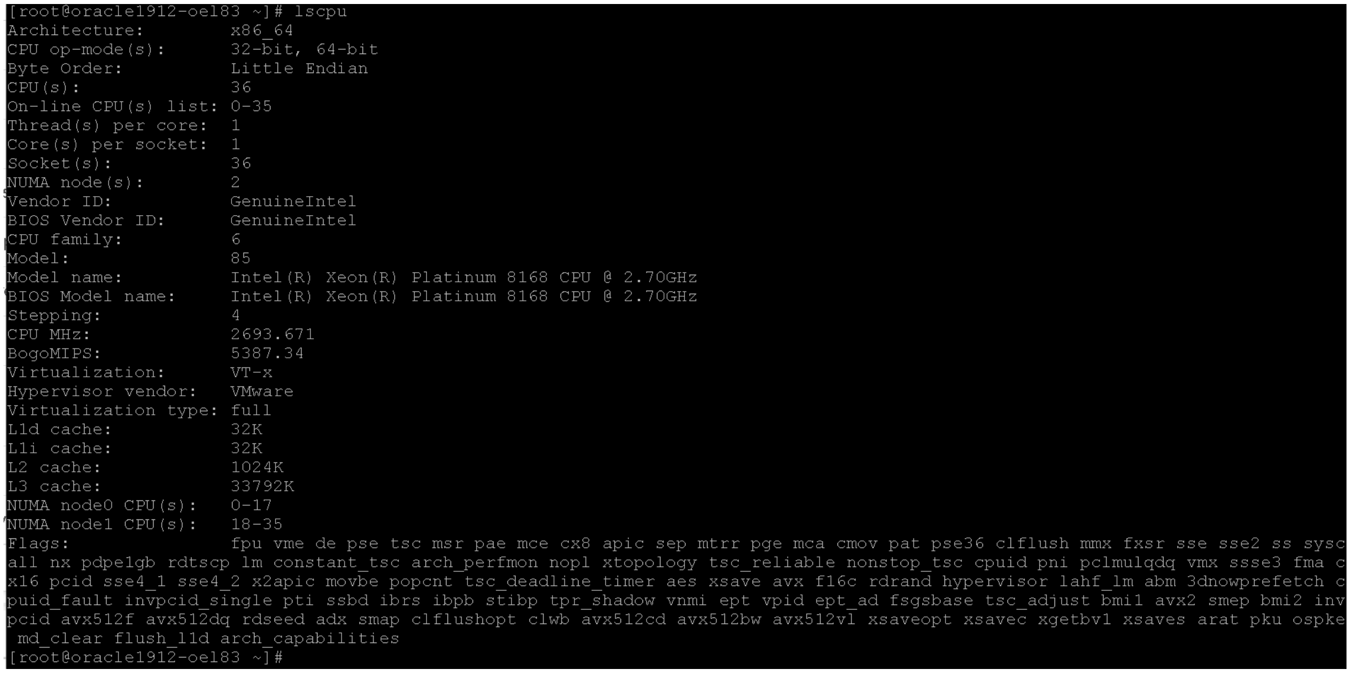
Edit Settings | Oracle1912-OEL83

CPU	36	
Cores per Socket	1	Sockets: 36
CPU Hot Plug	□ Enable CPU Hot Add	
Reservation	0	MHz
Limit	Unlimited	MHz
Shares	Normal	36000
Hardware virtualization	☑ Expose hardware assisted virtualization to the guest OS	
Performance Counters	☑ Enable virtualized CPU performance counters	
I/O MMU	☑ Enabled	
Memory	256	GB
Reservation	0 □ Reserve all guest memory (All locked)	MB
Limit	Unlimited	MB
Shares	Normal	1000000
Memory Hot Plug	☑ Enable	

FIGURE 4. VM Oracle1912-OEL83 Virtual Hardware Settings

The VM vNUMA is enabled as the number of VM vCPUs (36) is greater than 9 and more than the number of physical cores per socket (24).

In this example, in the guest OS, we see 36 sockets and two NUMA nodes with cores per socket set to one.



```
[root@oracle1912-oel83 ~]# lscpu
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              36
On-line CPU(s) list: 0-35
Thread(s) per core:  1
Core(s) per socket:  1
Socket(s):           36
NUMA node(s):        2
Vendor ID:           GenuineIntel
BIOS Vendor ID:      GenuineIntel
CPU family:          6
Model:               85
Model name:          Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
BIOS Model name:     Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
Stepping:            4
CPU MHz:             2693.671
BogoMIPS:            5387.34
Virtualization:      VT-x
Hypervisor vendor:   VMware
Virtualization type: full
L1d cache:           32K
L1i cache:           32K
L2 cache:            1024K
L3 cache:            33792K
NUMA node0 CPU(s):   0-17
NUMA node1 CPU(s):   18-35
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss sysc
all nx pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology tsc_reliable nonstop_tsc cpuid pni pclmulqdq vmx ssse3 fma c
x16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch c
puid_fault invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnmi ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 inv
pcid avx512f avx512dq rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves arat pku ospke
 md_clear flush_l1d arch_capabilities
[root@oracle1912-oel83 ~]#
```

FIGURE 5. VM Oracle1912-OEL83 Virtual Hardware Details

NUMA Considerations

Generally, using NUMA architecture may positively influence the performance of an application if the application is NUMA-aware.

VMware recommends keeping NUMA enabled in server hardware BIOS and at the guest operating system level, which should also be the default settings for NUMA support within most servers and guest operating systems.

Recommendation: As a best practice, consider keeping NUMA enabled in server hardware BIOS and at the guest operating system level.

For further information, review:

- *Setting the number of cores per CPU in a virtual machine (1010184)*
- *Setting corespersocket can affect guest OS topologies (81383)*
- *Virtual Machine vCPU and vNUMA Rightsizing – Guidelines*
- *Does corespersocket Affect Performance?*
- *What is PreferHT and When to Use It*
- *Virtual Machines with preallocated memory might be placed NUMA remote at random power-ons (76362)*

CPU Hot Plug and Hot Add

When CPU hot plug is enabled on a VM, the vNUMA capability is disabled once the VM starts. This feature can be used for Oracle workloads in which NUMA optimization is not needed.

Beginning with RHEL 5.5 and OEL 5.5, once CPU hot plug is enabled on a VM, for example increasing the VM vCPUs online from 8 to 16, for example, will be reflected in the OS immediately.

The SQL statement below can be run to obtain the new value of CPU that the Oracle instance now sees:

```
SQL> select name, value from v$parameter where name = 'cpu_count';
NAME                                  VALUE
-----------------------------------------------
cpu_count                              16
SQL>
```

Oracle is now able to take advantage of the increased number of vCPUs.

Recommendation: As a best practice, consider CPU hot plug when VM NUMA optimization is not needed.

For further information, review:

- RedHat *Hot adding CPU/RAM/NIC is supported for VMware guest?*
- RedHat *Is it possible to "hotplug" a CPU on a running Red Hat Enterprise Linux?*
- *On Demand Scaling up resources for Oracle production workloads – Hot Add CPU and Hot Add Memory*

CPU Hot Remove

Currently vSphere does not support CPU hot remove within its web client. CPU hot remove is an in-guest capability beginning with RHEL 7.1.

RHEL 7.1 and later supports **hot remove** of physical CPU and memory if the underlying hardware supports this function.

In-guest CPU hot remove does not result in a reduction of vCPUs allocated to the VM. This capability is an OS hot online/offline operation.

For further information, review:

- RedHat *How to hot add/remove virtual or real CPUs in RHEL7?*
- RedHat *CPU/Memory "Hot-Add" and "Hot-Remove" Support in RHEL version 7*
- *Hot Remove CPU and Memory for Oracle production workloads*

VM Memory Best Practices

Memory Reservation

Because Oracle databases can be memory-intensive, and to account for situations where performance is a key factor (and to avoid kernel swapping between ESX/ESXi and the guest OS in mission critical production environments), VMware recommends the following:

- Set the VM memory reservation equal to the sum of the size of the Oracle SGA, the Oracle PGA, the Oracle background processes, and the operating system used memory.
- Where the Oracle database is part of a third-party commercial enterprise resource planning application (ERP), follow virtualization guidelines from the ERP vendor.

- Note that setting reservations can limit vSphere vMotion operations. A VM can be live migrated only if the target ESX or ESXi host has free physical memory equal to or greater than the size of the reservation.
- The guest operating system within the VM still needs its own separate swap/page file. Follow the same swap space guidelines given for physical environments.

However, it is acceptable to overcommit more aggressively in non-production environments such as development, test, or QA.

For Oracle workloads, the recommended memory reservation = SGA + PGA + BG process sharable memory + operating system used memory, wherein:

- SGA is system global area
- PGA is program global area
- BG is background processes

The illustration below offers further details:

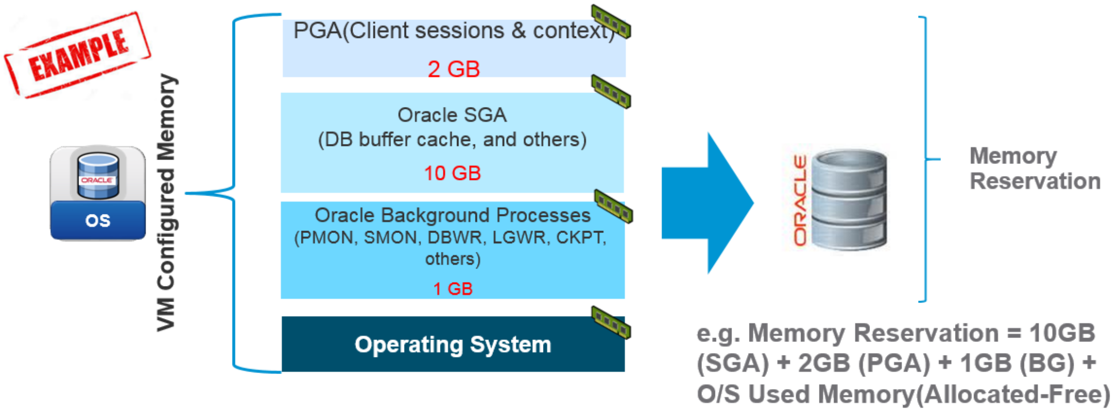

EXAMPLE
OS
VM Configured Memory
PGA(Client sessions & context)
2 GB
Oracle SGA
(DB buffer cache, and others)
10 GB
Oracle Background Processes
(PMON, SMON, DBWR, LGWR, CKPT, others)
1 GB
Operating System
ORACLE
Memory Reservation
e.g. Memory Reservation = 10GB (SGA) + 2GB (PGA) + 1GB (BG) + O/S Used Memory(Allocated-Free)

FIGURE 6. Oracle Memory Reservation Best Practices

The above recommendation holds for both Oracle AMM and ASMM memory management methods.

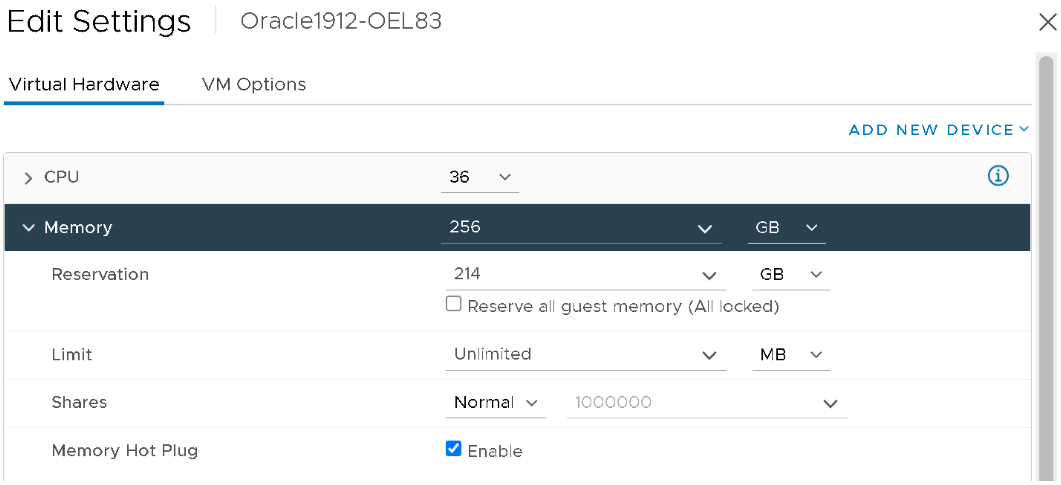
For Oracle ASMM memory management, the recommended memory reservation = SGA (sga_max_size) + PGA (pga_aggregate_limit) + BG process sharable memory + operating system used memory.

For Oracle AMM memory management, the recommended memory reservation = [SGA + PGA (memory_max_size)] + BG process sharable memory + operating system used memory.

For example, VM **Oracle1912-OEL83** created with 36 vCPUs and 256GB has an Oracle Database with Oracle instance parameters as follows:

- pga_aggregate_target=6G
- pga_aggregate_limit=12G
- sga_max_size=192G
- sga_target=192G

For Oracle ASMM memory management, the recommended memory reservation = 192G (sga_max_size) + 12G (pga_aggregate_limit) + 10G (BG process sharable memory + operating system used memory) = 214G.



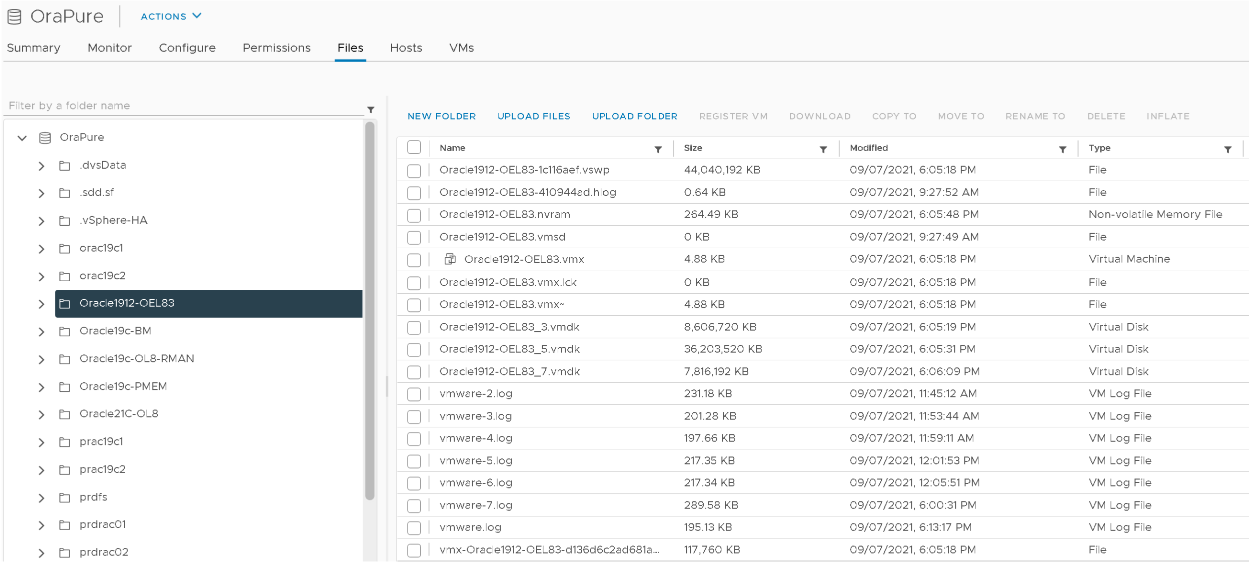
Edit Settings | Oracle1912-OEL83

Virtual Hardware | VM Options

CPU	36	
Memory	256	GB
Reservation	214 ☐ Reserve all guest memory (All locked)	GB
Limit	Unlimited	MB
Shares	Normal	1000000
Memory Hot Plug	☑ Enable	

FIGURE 7. Example of Oracle Memory Reservation

Notice that VM virtual swap disk **Oracle1912-OEL83-1c116aef.vswp** is 42G in size, the difference between VM vRAM (256G) and VM memory reservation (214G).



OraPure | ACTIONS

OraPure
- .dvsData
- .sdd.sf
- .vSphere-HA
- orac19c1
- orac19c2
- Oracle1912-OEL83
- Oracle19c-BM
- Oracle19c-OL8-RMAN
- Oracle19c-PMEM
- Oracle21C-OL8
- prac19c1
- prac19c2
- prdfs
- prdrac01
- prdrac02

Name	Size	Modified	Type
Oracle1912-OEL83-1c116aef.vswp	44,040,192 KB	09/07/2021, 6:05:18 PM	File
Oracle1912-OEL83-410944ad.hlog	0.64 KB	09/07/2021, 9:27:52 AM	File
Oracle1912-OEL83.nvram	264.49 KB	09/07/2021, 6:05:48 PM	Non-volatile Memory File
Oracle1912-OEL83.vmsd	0 KB	09/07/2021, 9:27:49 AM	File
Oracle1912-OEL83.vmx	4.88 KB	09/07/2021, 6:05:18 PM	Virtual Machine
Oracle1912-OEL83.vmx.lck	0 KB	09/07/2021, 6:05:18 PM	File
Oracle1912-OEL83.vmx~	4.88 KB	09/07/2021, 6:05:18 PM	File
Oracle1912-OEL83_3.vmdk	8,606,720 KB	09/07/2021, 6:05:19 PM	Virtual Disk
Oracle1912-OEL83_5.vmdk	36,203,520 KB	09/07/2021, 6:05:31 PM	Virtual Disk
Oracle1912-OEL83_7.vmdk	7,816,192 KB	09/07/2021, 6:06:09 PM	Virtual Disk
vmware-2.log	231.18 KB	09/07/2021, 11:45:12 AM	VM Log File
vmware-3.log	201.28 KB	09/07/2021, 11:53:44 AM	VM Log File
vmware-4.log	197.66 KB	09/07/2021, 11:59:11 AM	VM Log File
vmware-5.log	217.35 KB	09/07/2021, 12:01:53 PM	VM Log File
vmware-6.log	217.34 KB	09/07/2021, 12:05:51 PM	VM Log File
vmware-7.log	289.58 KB	09/07/2021, 6:00:31 PM	VM Log File
vmware.log	195.13 KB	09/07/2021, 6:13:17 PM	VM Log File
vmx-Oracle1912-OEL83-d136d6c2ad681a...	117,760 KB	09/07/2021, 6:05:18 PM	File

FIGURE 8. Example of VM Swap File

In these environments, a DBA can introduce memory overcommitment to take advantage of VMware memory reclamation features and techniques. Even in these environments, the type and number of databases that can be deployed using overcommitment largely depend on their actual workload.

Recommendation: As a best practice, consider setting memory reservations equal to the sum of the size of the Oracle SGA, the Oracle PGA, the Oracle background processes stack space, and operating system used memory in production environments.

For further information, review:

- *Oracle Initialization Parameters*
- *vSphere Resource Management*

Memory Hot Plug and Hot Add

The **hot plug** CPU and memory feature was added in the vSphere 4.0 release but, for best results, use VMs that are compatible with ESXi 5.0 and later.

With vSphere 6.5 and later, when enabling memory hot plug and adding memory to a VM, the memory will be added evenly to all vNUMA nodes, which makes this feature usable for more use cases.

The OS will see the increase in the memory, which can be viewed by running the command below. In this example, the output below indicates the current memory allocated to VM **SB-OL76-ORA19C**.

```
[root@sb_ol76_ora19c ~]# cat /proc/meminfo | grep MemTotal
MemTotal:       32664616 kB
[root@sb_ol76_ora19c ~]#
```

To increase the Oracle sga_max_size and memory_max_target memory initialization parameters and take advantage of the increased memory size, the Oracle spfile/pfile initialization parameter file would have to be modified and the database instance restarted.

The Oracle initialization parameters sga_target, memory_target, pga_aggregate_target and pga_aggregate_limit are online modifiable via the 'ALTER SYSTEM' command.

Memory hot plug does not automatically increase the VM memory reservation set. The VM would have to be powered down and memory reservation changed to take advantage of the increased memory, if needed.

Recommendation: As a best practice, rely on rightsizing rather than memory hot plug and consider memory hot plug for applications which tend to have unpredictable memory requirements.

For further information, review:

- Oracle *Initialization Parameters*
- RedHat *Hot adding CPU/RAM/NIC is supported for VMware guest?*
- *On Demand Scaling up resources for Oracle production workloads – Hot Add CPU and Hot Add Memory*

Memory Hot Remove

Currently, vSphere does not support memory hot remove in its web client. In-guest memory hot remove is not currently supported by VMware.

RHEL 7.0 and later support **hot remove** of physical memory as long as the underlying hardware supports this function. As per RedHat guidance, the hardware or virtualization provider must support hot plug functionality.

RHEL 7.1 and later support **hot remove** of physical CPU and memory if the underlying hardware supports this function.

For further information, review:

- RedHat *How to hot-add and hot-remove memory in RHEL7?*
- RedHat *CPU/Memory "Hot-Add" and "Hot-Remove" Support in RHEL version 7*
- *Hot Remove CPU and Memory for Oracle production workloads*

Persistent Memory

Persistent memory can be consumed by VMs in two different modes:

- Virtual persistent memory (vPMem)– Using vPMem, the memory is exposed to a guest OS as a virtual NVDIMM. This enables the guest OS to use PMem in byte addressable random mode.
- Virtual persistent memory disk (vPMemDisk) – Using vPMemDisk, the memory can be accessed by the guest OS as a virtual SCSI device, but the virtual disk is stored in a PMem datastore.

Both modes could be profitable for an Oracle deployment:

- vPMEM mode could be used with newer Linux operating systems, increasing performance of high-loaded databases. Oracle redo logs files can be placed on the vPMEM device in the guest using it in the DAX mode. One PMem module could be efficiently shared between multiple VMs running on the same host, providing high saving costs by sharing physical NVDIMMs between many consumers.
- vPMemDisk mode could be used with any versions of Linux guest OS or Oracle as a traditional block-based storage device, but with very low latency. Recent use cases have demonstrated the benefits of vPMemDisk for Oracle backup and restore.

Recommendation: As a best practice, consider using persistent memory to accelerate Oracle workloads.

For further information, review:

- *Accelerating Oracle Performance using vSphere Persistent Memory (PMEM)*
- *Accelerating Oracle Performance using vSphere Persistent Memory (PMEM) - Reference Architecture*
- *Oracle and vSphere Persistent Memory (PMEM) – vPMEM v/s vPMEMDisk*
- *Oracle and vSphere Persistent Memory (PMEM) – Oracle Instance Recovery – An Investigation*
- *SAP HANA with Intel Optane Persistent Memory on VMware vSphere*

VM Storage Best Practices

vSphere provides several datastore options for storage configuration for VMs, including:

- VMware Virtual Machine File System (VMFS) datastore
- NFS datastore
- iSCSI datastore
- vSAN datastore
- vVOL datastore

In addition, one could also deploy raw device mapping (RDM), which allows a VM to directly access a volume on the physical storage subsystem without formatting it with VMFS.

Other non-VMware in-guest storage mount options are also available (e.g., in-guest iSCSI mounts, Oracle Direct NFS).

Additionally, VMware VMDKs backed by VMware vSphere Virtual Volumes™ can also be considered when VM or VMDK-level granularity is needed for database operations (e.g., database backup and restore, database refresh).

Recommendation: As a best practice, consider the use of VMDKs for all Oracle workloads unless there is a specific use case calling for RDMs.

For further information, review:

- *Oracle on vSphere – Summary of Storage options*

Multiple PVSCSI Controllers and PVSCSI and VMDK Queue Depth

The PVSCSI controller is the optimal SCSI controller for an I/O-intensive application on vSphere, allowing not only higher I/O rate but also lowering CPU consumption compared with LSI Logic SAS.

For example, VM **Oracle1912-OEL83** is set up with four PVSCSI controllers with maximum queue depth for all of the PVSCSI controllers and VMDKs as shown below:

The two parameters vmw_pvscsi.cmd_per_lun and vmw_pvscsi.ring_pages are set to their maximum respective values.

```
[root@oracle1912-oel83 ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=”$(sed ‘s, release .*$,,g’ /etc/system-release)”
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT=”console”
GRUB_CMDLINE_LINUX=”crashkernel=auto resume=/dev/mapper/ol-swap rd.lvm.lv=ol/root
rd.lvm.lv=ol/swap rhgb quiet net.ifnames=0 biosdevname=0 elevator=none transparent_hugepage=never
vmw_pvscsi.cmd_per_lun=254 vmw_pvscsi.ring_pages=32”
GRUB_DISABLE_RECOVERY=”true”
GRUB_ENABLE_BLSCFG=true
[root@oracle1912-oel83 ~]#
```

Recommendation: As a best practice, increase PVSCSI and VMDK queue depth to the maximum in order to push larger I/O bandwidth, unless the underlying storage vendor recommends otherwise. Use multiple PVSCSI adapters with VMDKs spread across the PVSCSI controllers for load balancing and fairness.

For further information, review:

- *Queues, Queues and more Queues*

Virtual Disks Hot Add and Hot Remove

One can hot add and hot remove a virtual hard disk to an existing VM without shutting the VM down.

Oracle workloads can take advantage of virtual disk hot add and hot remove online for storage reconfiguration or capacity planning.

For further information, review:

- *On Demand Scaling up / Scaling down Storage resources for Oracle production workloads – Hot Add and Hot Remove non-clustered Disks*
- *"RAC" n "RAC" all night – Oracle RAC on vSphere 6.x – Add shared VMDK online without downtime for Oracle RAC ASM / OCFS2*
- *To be "RDM for Oracle RAC", or not to be, that is the question [Add Shared RDM in Physical/Virtual Compatibility mode for Oracle RAC]*

Virtual Disks Hot Extend

Beginning with RHEL 7, Oracle workloads can take advantage of virtual disks hot extend to allow resizing of disk partitions online without any downtime.

For further information, review:

- *On Demand hot extend non-clustered Oracle Disks online without downtime – Hot Extend Disks*
- *On Demand hot extend clustered VMDKs online without downtime – Hot Extend RAC clustered disks*

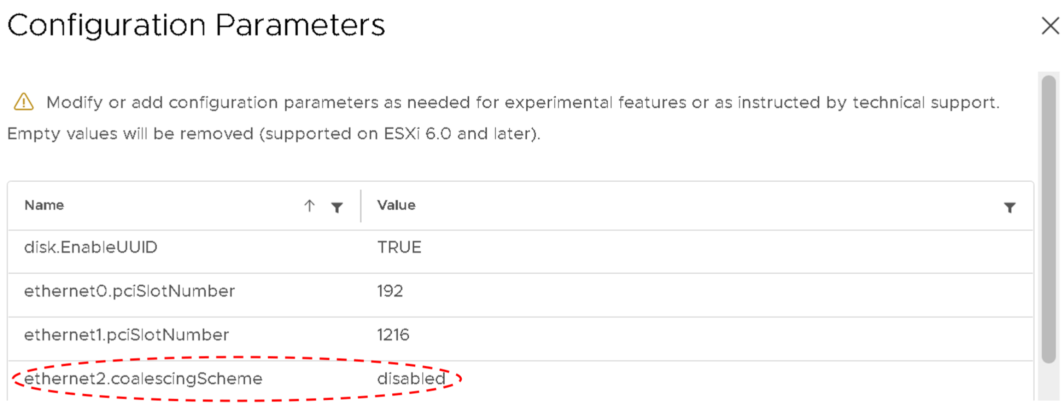
VM Network Best Practices

Interrupt Coalescing

Virtual network interrupt coalescing can reduce the number of interrupts, thus potentially decreasing CPU utilization. Depending on the workload, this can increase network latency by anywhere from a few hundred microseconds to a few milliseconds.

Recommendation: As a best practice, consider disabling virtual interrupt coalescing for Oracle RAC interconnect private NIC by setting ethernetX.coalescingScheme=disabled where ethernetX is the vmnic of the Oracle RAC interconnect adapter(s).

For example, in VM **Oracle1912-OEL83**, to disable interrupt coalescing for network adapter two, set **ethernet2.coalescingScheme=disabled** in the VM advanced configuration screen.



Configuration Parameters

⚠ Modify or add configuration parameters as needed for experimental features or as instructed by technical support. Empty values will be removed (supported on ESXi 6.0 and later).

Name	Value
disk.EnableUUID	TRUE
ethernet0.pciSlotNumber	192
ethernet1.pciSlotNumber	1216
ethernet2.coalescingScheme	disabled

FIGURE 9. VM Oracle1912-OEL83 Configuration Parameters

For further information, review:

- *Low throughput for UDP workloads on Windows virtual machines (2040065)*

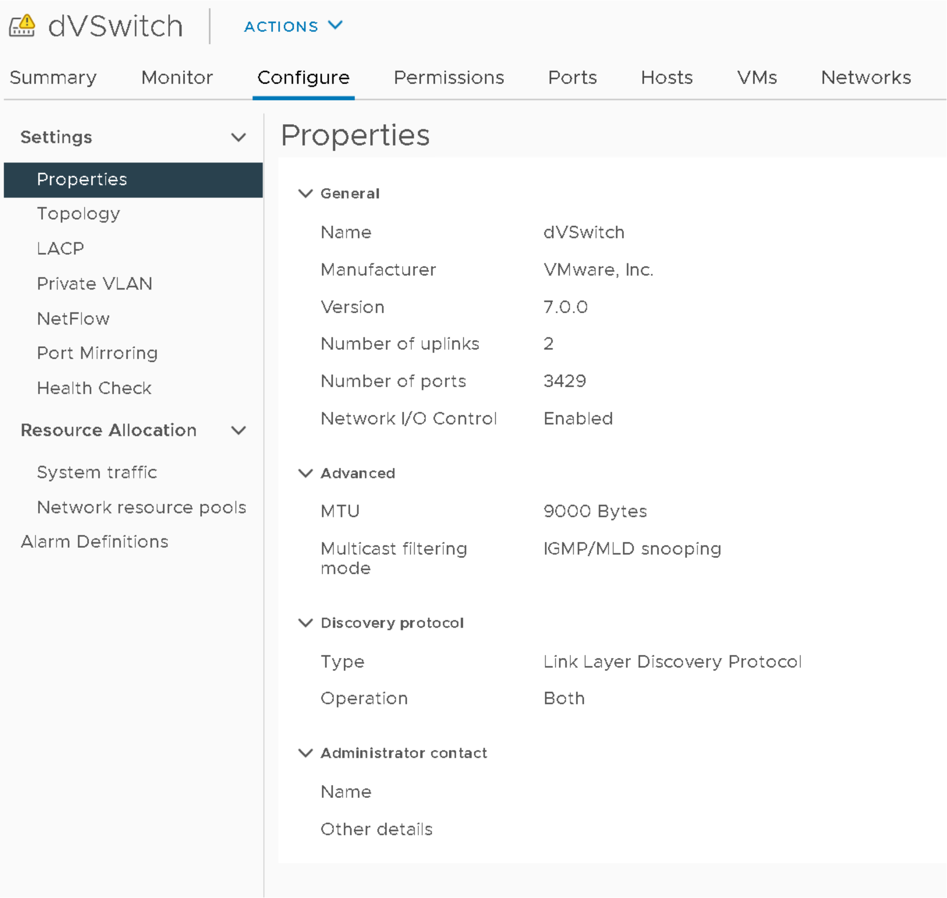
Jumbo Frames and Virtual NIC MTU size

A jumbo frame MTU (i.e., maximum transmission unit) can be increased from the default 1500 bytes to 9000 bytes. The larger payload provides for increased efficiency and higher throughput.

Oracle, by default, uses an 8KB block size to store data in the database. Increasing the TCP/IP payload to 9000+ bytes allows an entire Oracle block to be transmitted as one jumbo frame instead of dividing it up into six TCP/IP packets.

Only enhanced VMXNET3 adapters can be used with jumbo frames.

For example, the distributed switch **dVSwitch** has MTU set to 9000 bytes (jumbo frames).



Properties

General

Name	dVSwitch
Manufacturer	VMware, Inc.
Version	7.0.0
Number of uplinks	2
Number of ports	3429
Network I/O Control	Enabled

Advanced

MTU	9000 Bytes
Multicast filtering mode	IGMP/MLD snooping

Discovery protocol

Type	Link Layer Discovery Protocol
Operation	Both

Administrator contact

Name	
Other details	

FIGURE 10. dVSwitch Properties

Recommendation: As a best practice, consider enabling jumbo frames on the virtual switches in which you have configured, for example, Oracle RAC private interconnect, Oracle cache fusion traffic, Oracle backup traffic, or Oracle data loading traffic.

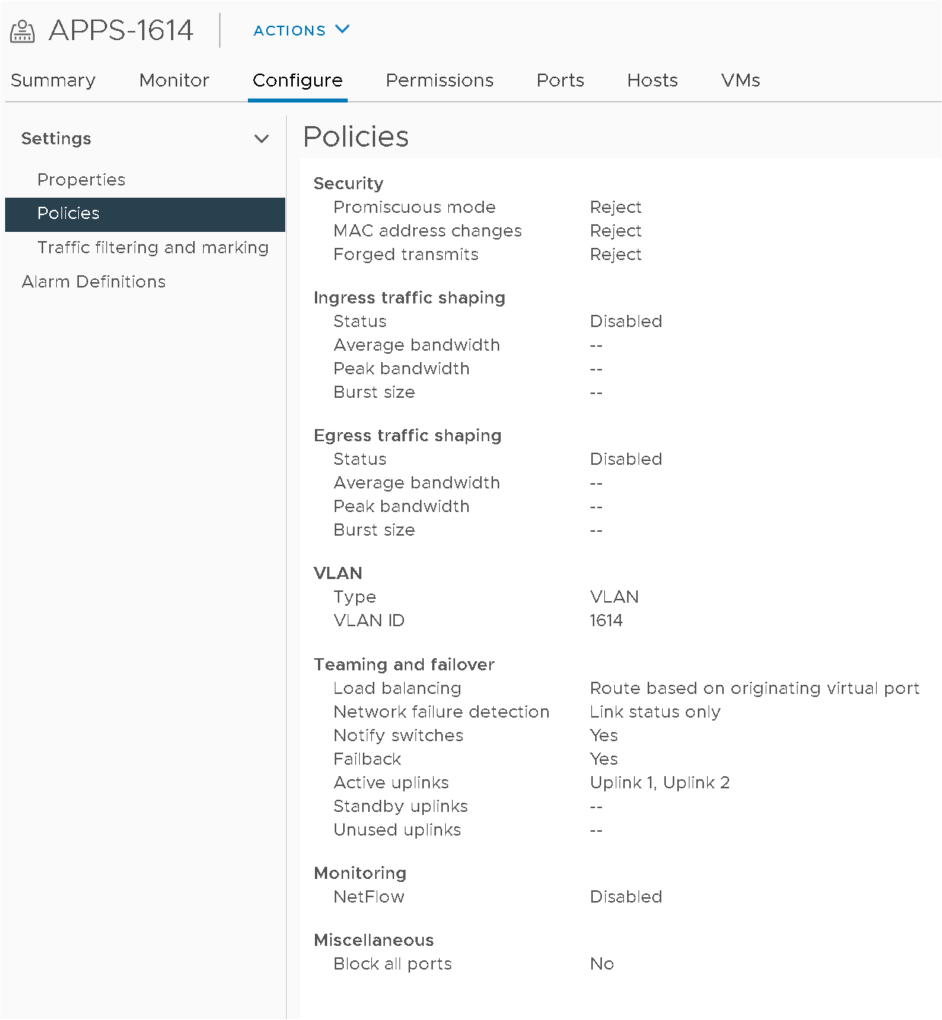
For further information, review:

- *Enabling Jumbo Frames on virtual distributed switches (1038827)*
- *Virtual network adapters that support jumbo frames (1015556)*
- *Using Jumbo Frames on VMware NSX for Oracle Workloads*

Load Balancing on VMware vSphere Standard Switch and vSphere Distributed Switch

One of the advantages of using the **Route Based on Originating Virtual Port** as the default for load balancing is low resource consumption because, in most cases, the virtual switch calculates uplinks for VMs only once.

For example, on the distributed switch **dVSwitch**, the port group **APPS-1614** has load balancing set to **Route Based** on **Originating Virtual Port**.



Policies

Security	
Promiscuous mode	Reject
MAC address changes	Reject
Forged transmits	Reject
Ingress traffic shaping	
Status	Disabled
Average bandwidth	--
Peak bandwidth	--
Burst size	--
Egress traffic shaping	
Status	Disabled
Average bandwidth	--
Peak bandwidth	--
Burst size	--
VLAN	
Type	VLAN
VLAN ID	1614
Teaming and failover	
Load balancing	Route based on originating virtual port
Network failure detection	Link status only
Notify switches	Yes
Failback	Yes
Active uplinks	Uplink 1, Uplink 2
Standby uplinks	--
Unused uplinks	--
Monitoring	
NetFlow	Disabled
Miscellaneous	
Block all ports	No

FIGURE 11. Port Group APPS-1614 Configuration

Recommendation: As a best practice, consider using Route Based on Originating Virtual Port as the default load balancing method on vSphere Standard Switch and vSphere Distributed Switch for Oracle RAC port group.

For further information, review:

- *Route Based on Originating Virtual Port.*

VM Maintenance Best Practices

Disabling Time Synchronization

Recommendation: As a best practice, consider using a native time synchronization service – such as NTPD or Chrony in Linux-based operating systems or W32Time in Microsoft Windows – and disable time synchronization on a VM.

For further information, review:

- *Disabling Time Synchronization (1189)*

VMware Tools

VMware Tools™ is a set of services and components that enables several features in various VMware products for better management and seamless user interactions with guest operating systems.

Recommendation: As a best practice, consider open-vm-tools for Oracle workloads.

For further information, review:

- *Open VM Tools*
- *Understanding the Three Types of VM Tools*

In-Guest Best Practices

CPU Best Practices

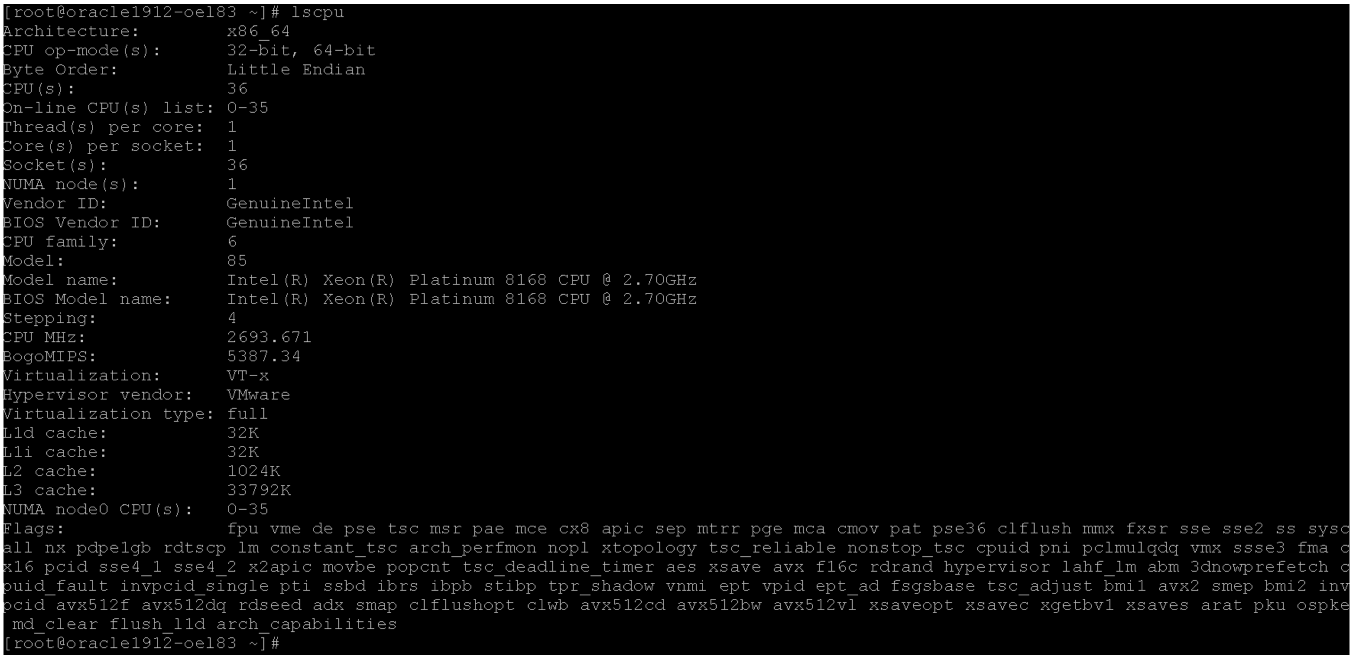
OS NUMA

To disable NUMA at an OS level, add the key-value pair **numa=off** to the kernel command line in boot loader configuration.

In RHEL 7, grub config must be rebuilt for changes to take effect. Rebooting the system will disable NUMA.

In this example, on VM **Oracle1912-OEL83**, setting **numa=off** in the grub, rebuilding the grub, and rebooting, we see NUMA is disabled at an OS level.

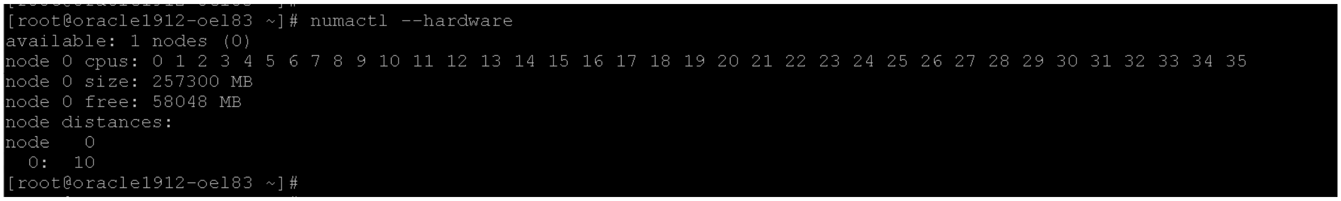
In the guest OS, we see 36 sockets, one NUMA node with cores per socket set to one, and NUMA disabled.



```
[root@oracle1912-oel83 ~]# lscpu
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              36
On-line CPU(s) list: 0-35
Thread(s) per core:  1
Core(s) per socket:  1
Socket(s):           36
NUMA node(s):        1
Vendor ID:           GenuineIntel
BIOS Vendor ID:      GenuineIntel
CPU family:          6
Model:               85
Model name:          Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
BIOS Model name:     Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
Stepping:            4
CPU MHz:             2693.671
BogoMIPS:            5387.34
Virtualization:      VT-x
Hypervisor vendor:   VMware
Virtualization type: full
L1d cache:           32K
L1i cache:           32K
L2 cache:            1024K
L3 cache:            33792K
NUMA node0 CPU(s):   0-35
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss sysc
all nx pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology tsc_reliable nonstop_tsc cpuid pni pclmulqdq vmx ssse3 fma c
x16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch c
puid_fault invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnmi ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 inv
pcid avx512f avx512dq rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves arat pku ospke
 md_clear flush_l1d arch_capabilities
[root@oracle1912-oel83 ~]#
```

FIGURE 12. VM Oracle1912-OEL83 Configuration Details

Numactl command output shows 36 CPUs with one NUMA node.



```
[root@oracle1912-oel83 ~]# numactl --hardware
available: 1 nodes (0)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
node 0 size: 257300 MB
node 0 free: 58048 MB
node distances:
node   0
  0:  10
[root@oracle1912-oel83 ~]#
```

FIGURE 13. Numactl Command Output

The RHEL 8 boot loader, grand unified boot loader (GRUB2), differs from the GRUB2 in RHEL 7.

Recommendation: As a best practice, consider keeping NUMA enabled in ESXi hardware BIOS and at the guest operating system level, which should also be the default settings for NUMA support within most servers and guest operating systems.

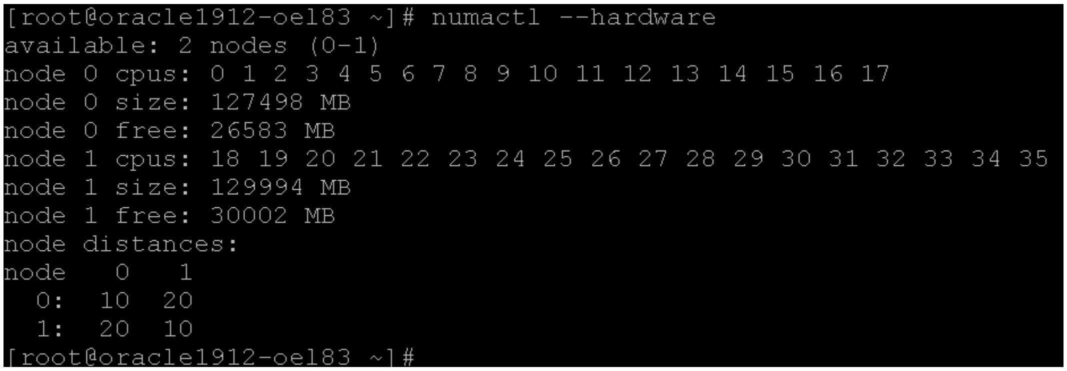
For further information, review:

- RedHat *How to disable NUMA in RHEL system?*
- RedHat *How can I visualize my system's NUMA topology in Red Hat Enterprise Linux?*
- RedHat *Working with GRUB 2*
- RedHat *numstat*

numactl Linux utility

Linux numactl lets administrators run a process with a specified scheduling or memory placement policy. Numactl can also set a persistent policy for shared memory segments or files and set the processor affinity and memory affinity of a process.

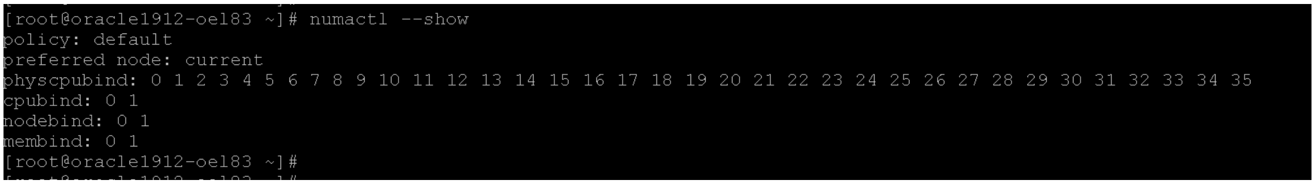
In this example, on VM **Oracle1912-OEL83**, the command **numactl –hardware** displays an inventory of available nodes on the system, including the relative distances between nodes.



```
[root@oracle1912-oel83 ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
node 0 size: 127498 MB
node 0 free: 26583 MB
node 1 cpus: 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
node 1 size: 129994 MB
node 1 free: 30002 MB
node distances:
node   0   1
  0:  10  20
  1:  20  10
[root@oracle1912-oel83 ~]#
```

FIGURE 14. System Hardware Inventory

The command **numactl –show** displays NUMA policy settings of the current process.



```
[root@oracle1912-oel83 ~]# numactl --show
policy: default
preferred node: current
physcpubind: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
cpubind: 0 1
nodebind: 0 1
membind: 0 1
[root@oracle1912-oel83 ~]#
```

FIGURE 15. NUMA Policy Settings

Recommendation: As a best practice, consider installing the numactl utility to accurately see the vNUMA configuration on the OS.

For further information, review:

- RedHat *numactl*
- RedHat *How to determine if NUMA configuration is enabled or disabled?*

Automatic NUMA Balancing

RHEL 7 introduced *Automatic NUMA balancing* to improve the performance of applications running on NUMA aware systems and automatically activate when booted on hardware with NUMA properties.

An application will generally perform best when the threads of its processes are accessing memory on the same NUMA node as those on which the threads are scheduled. Automatic NUMA balancing moves tasks (which can be threads or processes) closer to the memory they are accessing. It also moves application data to memory closer to the tasks that reference it. This is all done automatically by the kernel when automatic NUMA balancing is active.

Automatic NUMA balancing is enabled by default in Red Hat Enterprise Linux 7 and will automatically activate when booted on hardware with NUMA properties.

Automatic NUMA balancing is enabled when both of the following conditions are met:

```
# numactl --hardware shows multiple nodes

# cat /proc/sys/kernel/numa_balancing shows 1
```

To disable automatic NUMA balancing for the period of the session (i.e., not persistent as it is e-enabled at the next reboot), run the below command:

```
echo 0 > /proc/sys/kernel/numa_balancing
```

To disable automatic NUMA balancing and make it persistent across reboots, set the below key-value pair in the in the /etc/sysctl.conf (or in the appropriate sysctl configuration file under /etc/sysctl.d). A value of 1 means the feature is enabled, a value of **0** means disabled:

```
kernel.numa_balancing=0
```

For example, the system below has NUMA balancing disabled:

```
(root) # sysctl -e kernel.numa_balancing
kernel.numa_balancing = 0
```

Recommendation: As a best practice, consider turning off Automatic NUMA balancing for Oracle workloads as outlined in the Oracle Metalink below.

For further information, review:

- *Oracle Linux: RHCK 7 Automatic NUMA Balancing Induces High IO Wait (Doc ID 2749259.1)*
- Oracle *Requirements for Installing Oracle Database 19c on OL7 or RHEL7 64-bit (x86-64) (Doc ID 2551169.1)*
- Oracle *Requirements for Installing Oracle Database/Client 19c on OL8 or RHEL8 64-bit (x86-64) (Doc ID 2668780.1)*
- RedHat *Automatic NUMA balancing*

Memory Best Practices

HugeTLB Pages (Static and Traditional HugePages)

Physical memory is managed in chunks called pages. On most architectures supported by RHEL 7, the default size of a memory page is 4KB. However, specific applications can benefit from using larger page sizes.

RHEL and OEL enable the use of larger page sizes for applications working with big data sets. Using larger page sizes can improve the performance of such applications.

In addition to the usual 4KB memory pages, ESXi also provides 2MB memory pages (commonly referred to as *large pages*). ESXi assigns these 2MB machine memory pages to guest operating systems whenever possible; it does this even if the guest operating system doesn't request them (though the full benefit of large pages comes only when the guest operating system and applications use them as well).

Using very large page sizes can improve OS system performance by reducing the amount of system resources required to access page table entries.

For Oracle databases, using hugepages reduces the operating system maintenance of page states, and increases translation lookaside buffer (TLB) hit ratio.

HugePages provides the following advantages:

- Increased performance through increased TLB hits
- Pages are locked in memory and never swapped out, which provides RAM for shared memory structures such as SGA
- Contiguous pages are preallocated and cannot be used for anything else but for System V shared memory (e.g., SGA)
- Less bookkeeping work for the kernel for that part of virtual memory because of larger page sizes

Recommendation: As a best practice, consider using traditional hugepages for Oracle workloads.

For further information, review:

- RedHat *How can I configure hugepages in Red Hat Enterprise Linux?*
- RedHat *How do I check for hugepages usage and what is using it?*
- RedHat *Configuring huge pages*

Transparent HugePages

Beginning with RHEL 6, OEL 6, SLES 11, and UEK 2 kernels, transparent hugepages are implemented and enabled (default) to improve memory management. Transparent hugepages are similar to the hugepages available in previous Linux releases.

The main difference is that transparent hugepages are set up dynamically at run time by the khugepaged thread in kernel, while regular hugepages had to be preallocated at the bootup time.

Oracle recommends disabling transparent hugepages because it causes performance issues with Oracle databases and causes node reboots in RAC. In addition, transparent hugepages may cause problems even in a single-instance database environment, with unexpected performance problems or delays. As such, Oracle recommends disabling transparent hugepages on all database servers running Oracle.

To check if transparent hugepages are enabled or not, run the below command:

```
[root@oracle1912-oel83 ~]# cat /sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]
[root@oracle1912-oel83 ~]#
```

For RHEL kernel, the file path is different from above:

```
# cat /sys/kernel/mm/redhat_transparent_hugepage/enabled
```

One method for disabling transparent hugepages boot time is to edit the grub file and add the bellow key-value pair:

```
transparent_hugepage=never
```

The grub config must be rebuilt for changes to take effect. Rebooting the system will disable transparent hugepages.

Recommendation: As a best practice, consider disabling transparent hugepages because of the various performance issues outlined above by Oracle.

For further information, review:

- Oracle *ALERT: Disable Transparent HugePages on SLES11, RHEL6, RHEL7, OL6, OL7, and UEK2 and above (Doc ID 1557478.1)*
- RedHat *How to use, monitor, and disable transparent hugepages in RHEL6 and 7?*

Linux 1GB HugePages

RHEL 6.7 and later systems support 2MB and 1GB hugepages, which can be allocated at boot or at runtime.

Beginning with RHEL 7.1, there are two ways of reserving hugepages: at boot time and at run time. Reserving at boot time increases the possibility of success because the memory has not yet been significantly fragmented. However, on NUMA machines, the number of pages is automatically split among NUMA nodes. The run-time method allows you to reserve hugepages per NUMA node. If the run-time reservation is done as early as possible in the boot process, the probability of memory fragmentation is lower.

The vSphere 6.7 release introduces memory mappings for 1GB page sizes.

Per the explanation below, Oracle is not currently compatible with 1GB hugepages. It's recommended to use 2MB static hugepages until the issue is fixed.

As explained in Oracle Metalink unpublished Bug 17271305, the issue is in Oracle code. Ideally the fixed SGA would be in 2MB hugepages (it is small enough to fit into a 2MB huge page) but would require additional changes to Oracle code since the Linux shm*() interface only supports one hugepage size (whatever is set to default, either 2MB or 1GB).

For further information, review:

- Oracle *Startup Error ORA-27123: Unable To Attach To Shared Memory Segment Linux-x86_64 Error: 22: Invalid argument (Doc ID 1607545.1)*
- RedHat *Configuring huge pages*
- *Accelerating Oracle workloads with vSphere 6.7 Guest 2M & GB Huge Pages – An Investigation*

Tuned Profile tuned-profiles-oracle

Tuned is a service that monitors the system and optimizes the performance under certain workloads. The core of Tuned are **profiles** which tune the system for different use cases.

The profiles provided with **Tuned** are divided into the following categories:

- Power-saving profiles
- Performance-boosting profiles

The performance-boosting profiles include those that focus on the following aspects:

- Low latency for storage and network
- High throughput for storage and network
- Virtual machine performance
- Virtualization host performance

For example, on VM **Oracle1912-OEL83**, the following command lists the various Tuned profiles:

```
[root@oracle1912-oel83 tuned]# tuned-adm profile
```

Available profiles:

```
- accelerator-performance     - Throughput performance based tuning with disabled higher latency STOP states
- balanced                    - General non-specialized tuned profile
- desktop                     - Optimize for the desktop use-case
- hpc-compute                 - Optimize for HPC compute workloads
- intel-sst                   - Configure for Intel Speed Select Base Frequency
- latency-performance         - Optimize for deterministic performance at the cost of increased power consumption
- network-latency             - Optimize for deterministic performance at the cost of increased power consumption,
                                focused on low latency network performance
- network-throughput          - Optimize for streaming network throughput, generally only necessary on older CPUs
                                or 40G+ networks
- optimize-serial-console     - Optimize for serial console use.
- oracle                      - Optimize for Oracle RDBMS
- powersave                   - Optimize for low power consumption
- throughput-performance      - Broadly applicable tuning that provides excellent performance across a variety
                                of common server workloads
- virtual-guest               - Optimize for running inside a virtual guest
- virtual-host                - Optimize for running KVM guests
```

Current active profile: virtual-guest

```
[root@oracle1912-oel83 tuned]#
[root@oracle1912-oel83 tuned]# tuned-adm list | grep oracle
- oracle     - Optimize for Oracle RDBMS
[root@oracle1912-oel83 tuned]#
```

The command below also shows the current active profile

```
[root@oracle1912-oel83 tuned]# tuned-adm active
Current active profile: virtual-guest
[root@oracle1912-oel83 tuned]#
```

One can enable the tuned oracle profile via the command below

```
[root@oracle1912-oel83 tuned]# tuned-adm profile oracle
```

Verify the currently active profile:

```
[root@oracle1912-oel83 tuned]# tuned-adm active
Current active profile: oracle
[root@oracle1912-oel83 tuned]#
```

The Tuned oracle profile package **tuned-profiles-oracle** can be installed via the yum install command:

```
[root@oracle1912-oel83 tuned]# yum install tuned-profiles-oracle
```

During the OS installation, the best profile for the system is selected automatically. On VMware, the default profile is **virtual-guest**.

- virtual-guest – A profile designed for RHEL 8 VMs and VMware guests **based on the throughput-performance profile** that, among other tasks, decreases virtual memory swappiness and increases disk readahead values. It does not disable disk barriers. It inherits the throughput-performance profile and changes the energy_performance_preference and scaling_governor attributes to the performance profile.
- Oracle – **A profile optimized for Oracle database** loads based on throughput-performance profile. It additionally disables transparent huge pages and modifies other performance-related kernel parameters. This profile is provided by the tuned-profiles-oracle package.

Details of the **tuned-profiles-oracle** settings is shown as below

```
[root@oracle1912-oel83 tuned]# cat /usr/lib/tuned/oracle/tuned.conf
#
# tuned configuration
#

[main]
summary=Optimize for Oracle RDBMS
include=throughput-performance

[sysctl]
vm.swappiness = 10
vm.dirty_background_ratio = 3
vm.dirty_ratio = 40
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
kernel.shmmax = 4398046511104
kernel.shmall = 1073741824
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 6815744
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 9000 65499
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.panic_on_oops = 1
kernel.numa_balancing = 0

[vm]
transparent_hugepages=never
[root@oracle1912-oel83 tuned]#
```

Details of the virtual-guest settings is shown as below

```
[root@oracle1912-oel83 tuned]# cat /usr/lib/tuned/virtual-guest/tuned.conf
#
# tuned configuration
#

[main]
summary=Optimize for running inside a virtual guest
include=throughput-performance
```

```
[sysctl]
# If a workload mostly uses anonymous memory and it hits this limit, the entire
# working set is buffered for I/O, and any more write buffering would require
# swapping, so it's time to throttle writes until I/O can catch up.  Workloads
# that mostly use file mappings may be able to use even higher values.
#
# The generator of dirty data starts writeback at this percentage (system default
# is 20%)
vm.dirty_ratio = 30

# Filesystem I/O is usually much more efficient than swapping, so try to keep
# swapping low.  It's usually safe to go even lower than this on systems with
# server-grade storage.
vm.swappiness = 30
[root@oracle1912-oel83 tuned]#
```

Recommendation: As a best practice, consider setting the Tuned profile to tuned-profiles-oracle for performance reasons.

For further information, review:

- Oracle *Tuned Tools Benefit Can Be Used In Oracle RAC Environment (Doc ID 2096977.1)*
- RedHat *Getting started with Tuned*

Memory Swapping

The kernel parameter vm.swappiness is used to define how aggressive the kernel will swap memory pages. Higher values will increase aggressiveness, lower values decrease the amount of swap. A value of **0** instructs the kernel not to initiate swap until the amount of free and file-backed pages is less than the high-water mark in a zone.

Learn more about *vm.swappiness* in the *official kernel memory documentation*.

Swap usage can be lowered by enabling hugepages, but this will only apply mostly to Oracle Database cases as enabling hugepages won't stop system or application layer from swapping.

The confusion about vm.swappiness comes from the fact that in older Red Hat kernels, a value of 0 for vm.swappiness resulted in the minimal amount of swapping to avoid an out-of-memory (OOM) condition. In newer kernels (as of RHEL kernel 2.6.32-303), a value of 0 will completely disable swap, but a value of 1 will provide the minimal amount of swapping to avoid an OOM condition.

Oracle general recommendation is to set vm.swappiness to a value of 1. There is not much practical difference between a value of 0 and a value of 1 on older kernels, but 1 is the safe setting to avoid an OOM condition on newer kernels (assuming that some swap space is available, of course).

For example, to set vm.swappiness to 1 and make it persistent across reboots:

```
echo "vm.swappiness=1" >> /etc/sysctl.conf
sysctl -p
```

For example, the **tuned-profiles**-oracle package on RedHat, which is a profile optimized for Oracle database loads based on throughput-performance profile along with additionally disabling transparent hugepages and modifying other performance-related kernel parameters sets the vm.swappiness value to **10**.

```
[root@oracle1912-oel83 tuned]# cat /usr/lib/tuned/oracle/tuned.conf
#
# tuned configuration
#

[main]
summary=Optimize for Oracle RDBMS
include=throughput-performance

[sysctl]
vm.swappiness = 10
vm.dirty_background_ratio = 3
vm.dirty_ratio = 40
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
kernel.shmmax = 4398046511104
kernel.shmall = 1073741824
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 6815744
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 9000 65499
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.panic_on_oops = 1
kernel.numa_balancing = 0

[vm]
transparent_hugepages=never
[root@oracle1912-oel83 tuned]#
```

Recommendation: As a best practice, consider setting vm.swappiness to a value of 1 (the *swappiness* value can be between 0 and 100).

For further information, review:

- Oracle *vm.swappiness - 0 or 1? (Doc ID 1917687.1)*
- RedHat *How swappiness sysctl works RHEL7 and under which condition system will choose to swap*
- RedHat *Is there a tuned profile available for Oracle RDBMS?*
- Redhat *Getting started with Tuned*

Linux kdump and kernel.panic_on_oops

Kdump is an advanced crash-dumping mechanism. When enabled, the system is booted from the context of another kernel. This second kernel reserves a small amount of memory, and its only purpose is to capture the core dump image in case the system crashes. Since being able to analyze the core dump helps significantly in determining the exact cause of the system failure, it is strongly recommended to have this feature enabled.

Recommendation: As a best practice, consider setting up kdump on Linus OS.

The kernel parameter kernel.panic_on_oops controls the kernel's behavior when an oops or BUG is encountered.

- 0: try to continue operation
- 1: panic immediately. If the **panic** sysctl is also non-zero, then the machine will be rebooted.

An example of the kernel parameter kernel.panic_on_oops setting is shown below:

```
[root@oracle1912-oel83 sysctl.d]# cat /etc/sysctl.d/99-sysctl.conf | grep kernel.panic_on_oops
# oracle-database-preinstall-19c setting for kernel.panic_on_oops is 1 per Orabug 19212317
kernel.panic_on_oops = 1
```

Recommendation: As a best practice, consider setting up kernel.panic_on_oops on Linux OS. The Oracle Database Preinstallation RPM will ensure the above setting.

For further information, review:

- Oracle *Linux OS Service 'kdump' (Doc ID 550097.1)*
- Oracle *How to Configure "kdump" on Oracle Linux 7 (Doc ID 2049919.1)*
- Oracle *Configure Crash Dump on RedHat - Kdump recovery service (Doc ID 2003139.1)*
- Oracle *Process To Capture Dumps Manually When Linux OS Hangs on Baremetal and Virtual Env (VMware Guest / Oracle VM) (Doc ID 2091397.1)*
- Oracle *How to Obtain Kernel Freeze or Hang Information Using Hardware NMI (Doc ID 549779.1)*
- RedHat *Has the kernel parameter 'kernel.panic_on_oops' always default been set to 1 in Red Hat Enterprise Linux?*
- RedHat *What panic parameters are available for use to panic a system when it is hanging or sluggish?*
- Linux *Documentation for /proc/sys/kernel/**

Linux Out-of-Memory (OOM)

The *OOM Killer or Out of Memory Killer* is a process that the Linux kernel employs when the system is critically low on memory. This situation occurs because processes on the server are consuming a large amount of memory and the system requires more memory for its own processes and for allocation to other processes.

Recommendation: As a best practice, right size Oracle memory requirements and avoid memory over-commitment.

For further information, review:

- Oracle *How to Configure the Linux Out-of-Memory Killer*
- RedHat *How to troubleshoot Out of memory (OOM) killer in Red Hat Enterprise Linux?*
- RedHat *How to disable the Out of memory or oom-killer?*

Storage Best Practices

I/O Scheduling

The disk scheduler is responsible for ordering the I/O requests submitted to a storage device. RHEL 7 and OEL 7 traditional, single-queue schedulers have been deprecated.

Linux 2.6 kernel had 4 IO schedulers:

- Completely Fair Queuing (CFQ)
- NOOP (noop)
- Anticipatory (anticipatory)
- Deadline (deadline)

In RHEL 8, block devices support only multi-queue scheduling. This enables the block layer performance to scale well with fast solid-state drives (SSDs) and multi-core systems.

RHEL 8 and OEL 8 support the following multi-queue disk schedulers:

- none
- mq-deadline
- bfq
- kyber

For example, for VM **Oracle1912-OEL83**, the Guest I/O scheduler is set to **none** as shown below. The parameter **elevator** is set to a value of **none**.

```
[root@oracle1912-oel83 ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=”$(sed ‘s, release .*$,,g’ /etc/system-release)”
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT=”console”
GRUB_CMDLINE_LINUX=”crashkernel=auto resume=/dev/mapper/ol-swap rd.lvm.lv=ol/root
rd.lvm.lv=ol/swap rhgb quiet net.ifnames=0 biosdevname=0 elevator=none
transparent_hugepage=never vmw_pvscsi.cmd_per_lun=254 vmw_pvscsi.ring_pages=32”
GRUB_DISABLE_RECOVERY=”true”
GRUB_ENABLE_BLSCFG=true
[root@oracle1912-oel83 ~]#
```

For example, for device sdc, the IO scheduler is set to **none** (active scheduler is listed in square brackets ([ ]).

```
[root@oracle1912-oel83 ~]# cat /sys/block/sdc/queue/scheduler
[none] mq-deadline kyber bfq
[root@oracle1912-oel83 ~]#
```

Recommendation from RedHat:

If the hypervisor is known to do its own I/O scheduling, then guests often benefit greatly from the noop I/O scheduler. It allows the host or hypervisor to optimize the I/O requests and prioritize based on its view on the I/O from one or multiple guests. The noop scheduler can still combine small requests from the guest OS into larger requests before handing the I/O to the hypervisor, however noop follows the idea to spend as few CPU cycles as possible in the guest for I/O scheduling. The host or hypervisor will have an overview of the requests of all guests and have a separate strategy for handling I/O.

Recommendation: As a best practice, consider using noop (RHEL 7 and earlier) or none (RHEL 8 and later) for guest IO scheduling, as testing has shown that, generally, noop/none (deadline, in some cases) performs better for virtualized Linux guests. ESX uses an asynchronous intelligent I/O scheduler, and, for this reason, virtual guests should see improved performance by allowing ESX to handle I/O scheduling. Test your workload with your choice of I/O schedulers.

For further information, review:

- RedHat *What is the suggested I/O scheduler to improve disk performance when using RHEL with virtualization?*
- RedHat *How to use the Noop or None IO Schedulers*
- Red *RHEL 7 I/O Schedulers*
- Red *RHEL 8 Setting the disk scheduler*
- *Tuning options for disk I/O performance in Linux 2.6 kernel-based virtual machines (2011861)*

Partition Alignment

Most system and database administrations prefer to create partitions, as this clearly informs the system administrator that the device is in use.

RHEL 5, OEL 5 and earlier versions did not align partitions by default. The default was a 63-block partition offset, or 63 x 512 bytes = 31.5KB, causing block-sized writes to span multiple on-disk segments.

RHEL 6, OEL 6 and later versions establish 1MB partition alignment by default. In general, offsets should start on multiples of two, equal to or greater than 64KB.

The best practice is to assign an alignment offset to the in-guest disk partitions during partitioning to align the writes and reads during I/O processing. Partition alignment prevents performance I/O degradation caused by I/O crossing track boundaries and storage potentially needing to write two sectors or blocks on disk for every filesystem block.

It's recommended to seek out advice on partition alignment offset from your storage vendor. For example, Dell PowerStore uses a 128KB track size; a 0 offset (if no partitions are created) or a 1MB offset are perfectly aligned. For RHEL 5 and OEL5, it's recommended that you align the partition offset to 1MB. Oracle recommends using an offset of 1MB (first sector = 2048 when sectors are 512 bytes) as stated in Oracle best practice and starter kits.

Larger offsets can be used as well, especially for larger LUNs. For example, if partitions are aligned at 16MB (first sector = 32768, when sectors are 512 bytes), I/O will be aligned for stripe widths up to 16MB.

Oracle ASMLIB requires that the disks be partitioned, and Oracle recommends that you create a single whole-disk partition on each disk to use. Oracle ASMFD and Linux udev do not require partition headers.

Partitioning can be done using fdisk or parted or diskpart depending on the operating system. However, only **parted** can create partitions larger than 2TB.

The following steps show how to use the fdisk command:

- Create a single primary partition on the device.
- Use **x** to enter fdisk **expert** mode.
- Use **b** to change the partition **beginning** offset.
- Enter 2,048 for a 1MB offset (2,048 x 512 bytes blocks).
- Optionally, use **p** to **print** the partition table layout
- Use **w** to **write** the partition table.

For example, for device sdc, the partitioning alignment is set to 1MB default (2048 OS blocks)

```
[root@oracle1912-oel83 ~]# fdisk -lu /dev/sdc
Disk /dev/sdc: 500 GiB, 536870912000 bytes, 1048576000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xe612a59a

Device      Boot Start    End          Sectors       Size    Id    Type
/dev/sdc1   2048          1048575999   1048573952    500G    83    Linux

[root@oracle1912-oel83 ~]#
```

Recommendation: As a best practice, align the partition at least a 1MB (2048 blocks) boundary to avoid performance I/O degradation caused by I/O crossing track boundaries. VMware recommends seeking storage vendor advice on partition alignment offset.

For further information, review:

- *Oracle Linux: How To Align Partitions on Large Hard Disk Drives (HDDs) for Optimal Performance (Doc ID 1523947.1)*
- Oracle *Configuring Disk Devices to Use Oracle ASMLIB*
- RedHat *How can I properly re-align filesystems on un-aligned partitions or LVM devices?*
- RedHat *Partition and File System Tools*

Multipathing

Multipathing software is critical for any database deployment for a few reasons:

- **Load-balancing and performance** – Multipathing software allows processing read-and-write I/Os to the device across multiple storage paths, spreading the load across the multiple paths to load balance and avoid single path.
- **Path-failover** – The multipath software creates a pseudo device name or alias for use by the application. I/Os to the pseudo device are serviced by all the different paths. If a path stops working, the multipath software automatically directs I/Os to the remaining paths. If a path comes back, the software will resume servicing I/Os to that path automatically. Meanwhile, the application continues to use the pseudo device, regardless of which paths are active.
- **Multipath pseudo devices work well with Oracle ASM** – Oracle ASM requires a single presentation of each storage device and will not list a device as an ASM candidate if it finds multiple paths to that device. By using the multipath pseudo name (or user-friendly alias), ASM recognizes a single presentation, and the multipath software takes care of spreading the I/Os across the different paths of that pseudo device.

ESXi supports and implements storage multipathing via a special VMkernel layer called pluggable storage architecture (PSA). The native multipathing plug-in (NMP) is the VMkernel multipathing module that ESXi provides by default. The NMP associates physical paths with a specific storage device and provides a default path-selection algorithm based on the array type. The NMP is extensible and manages additional submodules, called path selection policies (PSPs) and storage array type policies (SATPs). PSPs and SATPs can be provided by VMware or by a third party.

Recommendation: As a best practice, remember that there are no advantages to implementing multipathing in a guest operating system as it does not increase performance unless the storage vendor software requires it.

For further information, review:

- Oracle *About Using Oracle ASM with Multipath Disks*
- Redhat Setting up DM Multipath
- Pluggable Storage Architecture and Path Management

Enable SCSI_ID for devices

By default, running the Linux command scsi_id against any operating system disks will not yield the SCSI ID of the disk. This is the expected behavior beginning with VMware ESX 4.0 and later.

In order for the SCSI inquiry command to return the SCSI ID of the disk, power off the VM, enable the parameter disk.EnableUUID in the .vmx configuration file as below, and restart the VM.

```
disk.EnableUUID = “TRUE”
```

After setting the above parameter and restarting the VM, the SCSI inquiry command now returns the SCSI ID of the device.

In RHEL 7 and later, the scsi_id command is located in /usr/lib/udev/scsi_id. In RHEL 6 and earlier, the command is located in /sbin/scsi_id.

```
[root@oracle1912-oel83 /]# /usr/lib/udev/scsi_id -gud /dev/sdc
36000c29a44b6e7c20c2f023b83cb7887
[root@oracle1912-oel83 /]#
```

Recommendation: As a best practice, enable the parameter disk.EnableUUID in the VM .vmx file.

For further information, review:

- RedHat *Why is scsi_id not returning any output in a VM on VMware ESX 4.0 and above*
- RedHat *How to check disk.EnableUUID parameter from VM in vSphere for Openshift Container Platform*
- *sg_inq command takes a long time to run on ESX 4.0 and later (1029157)*

Maximum I/O Size for Block Devices

The maximum allowed size of a Linux I/O request in KB is determined by two parameters:

max_sectors_kb and max_hw_sectors_kb in sysfs.

The Linux sysfs parameters description for the above two parameters are as follows:

- hw_sector_size (RO) – This is the hardware sector size of the device in bytes
- max_hw_sectors_kb (RO) – This is the maximum number of kilobytes supported in a single data transfer
- max_sectors_kb (RW) – This is the maximum number of kilobytes that the block layer will allow for a filesystem request. It must be smaller than or equal to the maximum size allowed by the hardware.

For example, VM Oracle1912-OEL83 running OEL 8.3 OS with Oracle 19.12 RDBMS and Grid Infrastructure displays the following values for the above parameters:

```
[root@oracle1912-oel83 queue]# cat /etc/redhat-release
RHELrelease 8.4 (Ootpa)

[root@oracle1912-oel83 queue]# cat hw_sector_size
512

[root@oracle1912-oel83 queue]# cat max_sectors_kb
1280

[root@oracle1912-oel83 queue]# cat max_hw_sectors_kb
32767
[root@oracle1912-oel83 queue]#
```

The parameter max_sectors_kb restricts the largest I/O size that the OS will issue to a block device and cannot exceed the maximum value set by max_hw_sectors_kb.

The Linux maximum I/O size max_sectors_kb can be made persistent and set in the rc.local or using UDEV rules.

Recommendation: As a best practice, refer to storage vendor documentation to ensure the maximum allowed OS I/O size matches storage vendor I/O size expectations.

For further information, review:

- Linux *sysfs documentation*
- Linux Generic block device tuning parameters

Network Best Practices

Enable Jumbo Frames for Oracle RAC Private Interconnect

A jumbo frame MTU can be increased from the default 1500 bytes to 9000 bytes. The larger payload provides for increased efficiency and higher throughput.

Oracle, by default, uses an 8KB block size to store data in the database. Increasing the TCP/IP payload to 9000+ bytes allow an entire Oracle block to be transmitted as one jumbo frame instead of dividing it up into six TCP/IP packets.

On VMware NSX, private network MTU can be changed from 1500 to 8922 (9000 – 50 bytes of VXLAN overhead – 28 bytes of ICMP/ping = 8922 bytes). The network interface name and subnet remain the same.

1. Shutdown Oracle Clusterware stack on all nodes.
2. Make the required network change of MTU size at the OS network layer, ensure private network is available with the desired MTU size (in this case 8922 bytes) and ping with the desired MTU size works on all cluster nodes.
3. Restart Oracle Clusterware stack on all nodes.

Recommendation: As a best practice, consider making the changes referenced above to the Oracle RAC private interconnect network adapter for enhanced performance of Oracle RAC private interconnect traffic.

For further information, review:

- Oracle *Recommendation for the Real Application Cluster Interconnect and Jumbo Frames (Doc ID 341788.1)*
- Oracle *How to Modify Private Network Information in Oracle Clusterware (Doc ID 283684.1)*
- *Using Jumbo Frames on VMware NSX for Oracle Workloads*

Receive Side Scaling (RSS) and Multiqueue Support

Receive side scaling (RSS) is a network driver technology that enables the efficient distribution of network receive processing across multiple CPUs in multiprocessor systems.

Receive side scaling (RSS) and multiqueue support are included in the VMXNET3 Linux device driver. The VMXNET3 device always supported multiple queues, however the Linux driver previously used one Rx and one Tx queue.

For the VMXNET3 driver included with the Linux operating system, multiqueue support was introduced in Linux kernel version 2.6.37 and maintained in subsequent versions.

Multiqueue is a technique designed to enhance networking performance by allowing the Tx and Rx queues to scale with the number of CPUs in multi-processor systems. Many physical network adapters support multiqueue. When enabled on a NIC, Linux VMs running on vSphere 5 can take advantage of it.

In some Linux distributions, this driver technology is referred to as receive packet steering (RPS), which is the software version of hardware-based RSS.

For example, for VM **Oracle1912-OEL83**, see the RSS indirection table:

```
oracle@oracle1912-oel83:ora19c:/home/oracle> ethtool -i eth0
driver: vmxnet3
version: 1.4.17.0-k-NAPI
firmware-version:
expansion-rom-version:
bus-info: 0000:0b:00.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
oracle@oracle1912-oel83:ora19c:/home/oracle>

oracle@oracle1912-oel83:ora19c:/home/oracle> ethtool -x eth0
RX flow hash indirection table for eth0 with 8 RX ring(s):
0:      0    1    2    3    4    5    6    7
8:      0    1    2    3    4    5    6    7
16:     0    1    2    3    4    5    6    7
24:     0    1    2    3    4    5    6    7
RSS hash key:
Operation not supported
RSS hash function:
    toeplitz: on
    xor: off
    crc32: off
oracle@oracle1912-oel83:ora19c:/home/oracle>
```

Recommendation: As a best practice, consider enabling RSS to increase workload performance.

For further information, review:

- RedHat *What is Receive Side Scaling (RSS), and how do I configure it in RHEL?*
- RSS and multiqueue support in Linux driver for VMXNET3 (2020567)

TCP Segmentation Offload

Use TCP segmentation offload (TSO) in VMkernel network adapters and VMs to improve the network performance of workloads that have severe latency requirements.

By default, TSO is enabled in the VMXNET 3 VM adapters.

For example, VM **Oracle1912-OEL83** ethernet adapter **eth0** has the following offload characteristics:

```
[root@oracle1912-oel83 ~]# ethtool -k eth0
Features for eth0:
rx-checksumming: on
tx-checksumming: on
        tx-checksum-ipv4: off [fixed]
        tx-checksum-ip-generic: on
        tx-checksum-ipv6: off [fixed]
        tx-checksum-fcoe-crc: off [fixed]
        tx-checksum-sctp: off [fixed]
scatter-gather: on
        tx-scatter-gather: on
        tx-scatter-gather-fraglist: off [fixed]
tcp-segmentation-offload: on
        tx-tcp-segmentation: on
        tx-tcp-ecn-segmentation: off [fixed]
        tx-tcp-mangleid-segmentation: off
        tx-tcp6-segmentation: on
generic-segmentation-offload: on
generic-receive-offload: on
large-receive-offload: on
rx-vlan-offload: on
tx-vlan-offload: on
ntuple-filters: off [fixed]
receive-hashing: on
highdma: on
rx-vlan-filter: on [fixed]
vlan-challenged: off [fixed]
tx-lockless: off [fixed]
netns-local: off [fixed]
tx-gso-robust: off [fixed]
tx-fcoe-segmentation: off [fixed]
tx-gre-segmentation: off [fixed]
tx-gre-csum-segmentation: off [fixed]
tx-ipxip4-segmentation: off [fixed]
tx-ipxip6-segmentation: off [fixed]
tx-udp_tnl-segmentation: off [fixed]
tx-udp_tnl-csum-segmentation: off [fixed]
tx-gso-partial: off [fixed]
tx-sctp-segmentation: off [fixed]
tx-esp-segmentation: off [fixed]
tx-udp-segmentation: off [fixed]
fcoe-mtu: off [fixed]
tx-nocache-copy: off
loopback: off [fixed]
rx-fcs: off [fixed]
rx-all: off [fixed]
tx-vlan-stag-hw-insert: off [fixed]
rx-vlan-stag-hw-parse: off [fixed]
rx-vlan-stag-filter: off [fixed]
l2-fwd-offload: off [fixed]
hw-tc-offload: off [fixed]
esp-hw-offload: off [fixed]
esp-tx-csum-hw-offload: off [fixed]
rx-udp_tunnel-port-offload: off [fixed]
tls-hw-tx-offload: off [fixed]
tls-hw-rx-offload: off [fixed]
rx-gro-hw: off [fixed]
tls-hw-record: off [fixed]
rx-gro-list: off [fixed]
[root@oracle1912-oel83 ~]#
```

Recommendation: As a best practice, ensure TCP segmentation offload (TSO) is enabled in the guest OS to improve network performance of workloads with severe latency requirements.

For further information, review:

- *Understanding TCP Segmentation Offload (TSO) and Large Receive Offload (LRO) in a VMware environment (2055140)*
- *TCP Segmentation Offload*

Large Receive Offload

Use large receive offload (LRO) to reduce the CPU overhead for processing packets that arrive from the network at a high rate.

By default, LRO is enabled in the VMXNET3 VM adapters.

For example, VM **Oracle1912-OEL83** ethernet adapter **eth0** has the following offload characteristics:

```
[root@oracle1912-oel83 ~]# ethtool -k eth0
Features for eth0:
rx-checksumming: on
tx-checksumming: on
    tx-checksum-ipv4: off [fixed]
    tx-checksum-ip-generic: on
    tx-checksum-ipv6: off [fixed]
    tx-checksum-fcoe-crc: off [fixed]
    tx-checksum-sctp: off [fixed]
scatter-gather: on
    tx-scatter-gather: on
    tx-scatter-gather-fraglist: off [fixed]
tcp-segmentation-offload: on
    tx-tcp-segmentation: on
    tx-tcp-ecn-segmentation: off [fixed]
    tx-tcp-mangleid-segmentation: off
    tx-tcp6-segmentation: on
generic-segmentation-offload: on
generic-receive-offload: on
large-receive-offload: on
rx-vlan-offload: on
tx-vlan-offload: on
ntuple-filters: off [fixed]
receive-hashing: on
highdma: on
rx-vlan-filter: on [fixed]
vlan-challenged: off [fixed]
tx-lockless: off [fixed]
netns-local: off [fixed]
tx-gso-robust: off [fixed]
tx-fcoe-segmentation: off [fixed]
tx-gre-segmentation: off [fixed]
tx-gre-csum-segmentation: off [fixed]
tx-ipxip4-segmentation: off [fixed]
tx-ipxip6-segmentation: off [fixed]
tx-udp_tnl-segmentation: off [fixed]
tx-udp_tnl-csum-segmentation: off [fixed]
tx-gso-partial: off [fixed]
tx-sctp-segmentation: off [fixed]
```

```
tx-esp-segmentation: off [fixed]
tx-udp-segmentation: off [fixed]
fcoe-mtu: off [fixed]
tx-nocache-copy: off
loopback: off [fixed]
rx-fcs: off [fixed]
rx-all: off [fixed]
tx-vlan-stag-hw-insert: off [fixed]
rx-vlan-stag-hw-parse: off [fixed]
rx-vlan-stag-filter: off [fixed]
l2-fwd-offload: off [fixed]
hw-tc-offload: off [fixed]
esp-hw-offload: off [fixed]
esp-tx-csum-hw-offload: off [fixed]
rx-udp_tunnel-port-offload: off [fixed]
tls-hw-tx-offload: off [fixed]
tls-hw-rx-offload: off [fixed]
rx-gro-hw: off [fixed]
tls-hw-record: off [fixed]
rx-gro-list: off [fixed]
[root@oracle1912-oel83 ~]#
```

Recommendation: As a best practice, consider using large receive offload (LRO) to reduce the CPU overhead for processing packets that arrive from the network at a high rate.

For further information, review:

- RedHat *How do I enable or disable LRO (Large Receive Offload) or GRO (Generic Receive Offload)?*
- *Understanding TCP Segmentation Offload (TSO) and Large Receive Offload (LRO) in a VMware environment (2055140)*
- *Poor TCP performance might occur in Linux virtual machines with LRO enabled (1027511)*
- *Large Receive Offload*

Oracle Best Practices

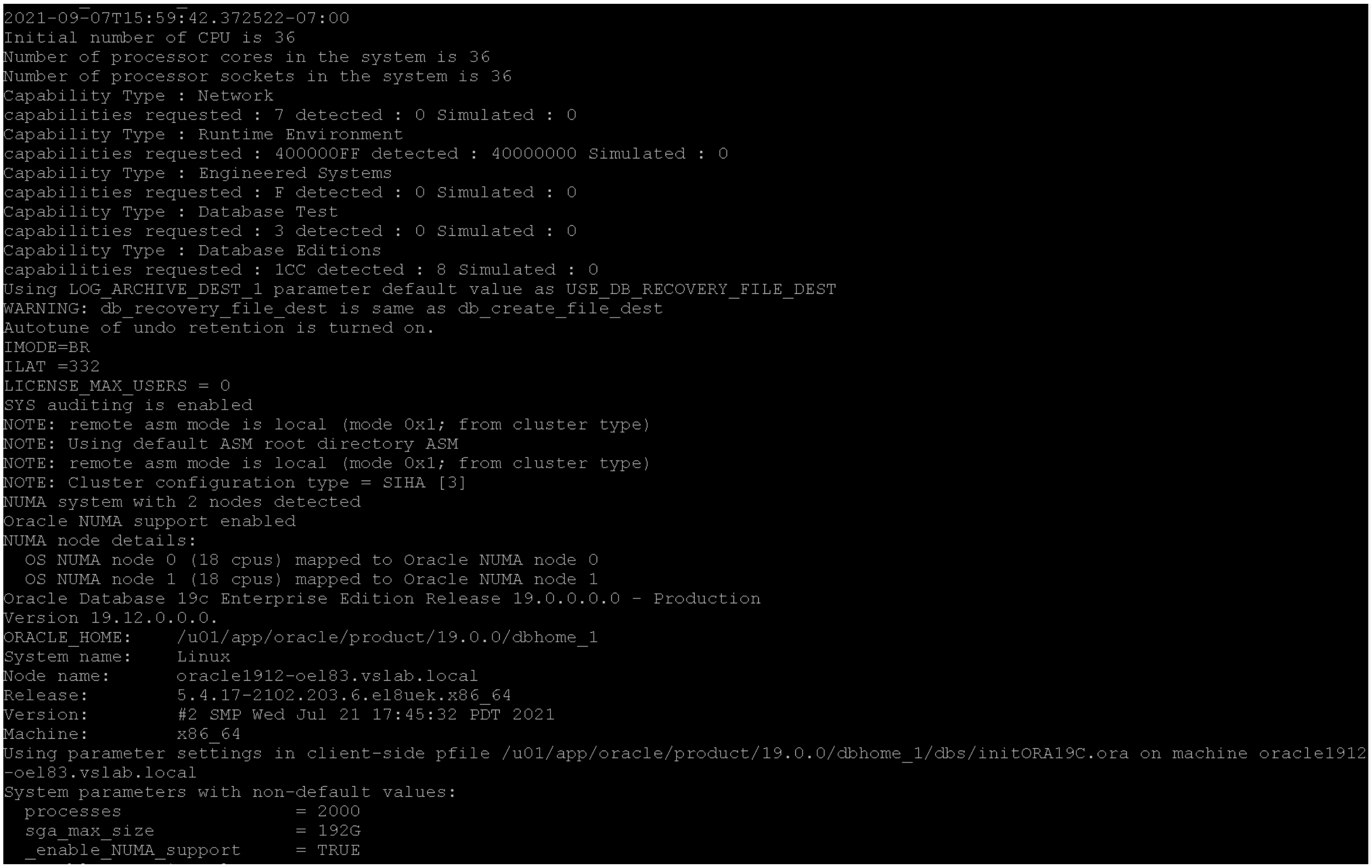
CPU Best Practices

Oracle NUMA

Oracle NUMA support was enabled by default in Oracle 10g. In Oracle 11g and later, Oracle NUMA support is disabled by default.

The two key NUMA initialization parameters in 10g and 11g are _enable_numa_optimization (10g) and _enable_numa_support (11g).

Setting _enable_numa_support='TRUE' in the initiation parameter results in the database starting up as NUMA aware, as shown in the alert log below:



```
2021-09-07T15:59:42.372522-07:00
Initial number of CPU is 36
Number of processor cores in the system is 36
Number of processor sockets in the system is 36
Capability Type : Network
capabilities requested : 7 detected : 0 Simulated : 0
Capability Type : Runtime Environment
capabilities requested : 400000FF detected : 40000000 Simulated : 0
Capability Type : Engineered Systems
capabilities requested : F detected : 0 Simulated : 0
Capability Type : Database Test
capabilities requested : 3 detected : 0 Simulated : 0
Capability Type : Database Editions
capabilities requested : 1CC detected : 8 Simulated : 0
Using LOG_ARCHIVE_DEST_1 parameter default value as USE_DB_RECOVERY_FILE_DEST
WARNING: db_recovery_file_dest is same as db_create_file_dest
Autotune of undo retention is turned on.
IMODE=BR
ILAT =332
LICENSE_MAX_USERS = 0
SYS auditing is enabled
NOTE: remote asm mode is local (mode 0x1; from cluster type)
NOTE: Using default ASM root directory ASM
NOTE: remote asm mode is local (mode 0x1; from cluster type)
NOTE: Cluster configuration type = SIHA [3]
NUMA system with 2 nodes detected
Oracle NUMA support enabled
NUMA node details:
  OS NUMA node 0 (18 cpus) mapped to Oracle NUMA node 0
  OS NUMA node 1 (18 cpus) mapped to Oracle NUMA node 1
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.12.0.0.0.
ORACLE_HOME:    /u01/app/oracle/product/19.0.0/dbhome_1
System name:    Linux
Node name:      oracle1912-oel83.vslab.local
Release:        5.4.17-2102.203.6.el8uek.x86_64
Version:        #2 SMP Wed Jul 21 17:45:32 PDT 2021
Machine:        x86_64
Using parameter settings in client-side pfile /u01/app/oracle/product/19.0.0/dbhome_1/dbs/initORA19C.ora on machine oracle1912
-oel83.vslab.local
System parameters with non-default values:
  processes                = 2000
  sga_max_size             = 192G
  _enable_NUMA_support     = TRUE
```

FIGURE 16. Oracle Database Alert Log

The decision to NUMA or not NUMA for Oracle workloads depends on several factors, including:

- Whether the workload requires more vCPUs than the number of physical cores in a NUMA node
- Whether the workload is memory latency sensitive or memory bandwidth intensive

Recommendation: As a best practice, follow Oracle's recommendation and test it sufficiently in a test environment, apply all the bug fixes or patch sets required for the Oracle database version, review all known NUMA issues in the Oracle database available for download before deciding to use it with production system. In some circumstances, enabling Oracle NUMA support might improve performance.

VMware recommends you do NOT change parameters in the initialization parameter file that begin with an "_" because these are hidden parameters and should only be changed when receiving specific instructions from Oracle support.

For further information, review:

- Oracle *Enable Oracle NUMA support with Oracle Server Version 11gR2 (Doc ID 864633.1)*
- *Oracle NUMA Usage Recommendation (Doc ID 759565.1)*
- *To NUMA or not to NUMA – Oracle workloads and NUMA*

Oracle CPU_COUNT & CPU_MIN_COUNT

Oracle initialization parameter **CPU_COUNT** specifies the total number of available CPU threads on CPUs with multiple CPU threads. Various components of Oracle Database are configured based on the number of CPUs, such as the optimizer, parallel query, and resource manager.

If CPU_COUNT is set to **0** (its default setting), then Oracle Database continuously monitors the number of CPUs reported by the operating system and uses the current count.

If CPU_COUNT is set to a value other than **0**, then Oracle Database will use this count rather than the actual number of CPUs, thus disabling dynamic CPU reconfiguration. In this case, Oracle cannot take advantage of the increase in the number of CPU to the OS.

When resource manager is managing CPU (RESOURCE_MANAGER_PLAN is set), then the database's CPU utilization is limited to CPU_COUNT CPU threads. This feature is called *instance caging*. If resource manager is enabled at the CDB level, then the PDB's CPU utilization is limited to the PDB's CPU_COUNT.

CPU_MIN_COUNT specifies the minimum number of CPUs required by a pluggable database (PDB) at any given time. For multi-threaded CPUs, this number corresponds to CPU threads, not CPU cores.

You can set this parameter at the CDB level, and for each individual PDB. This enables you to control each PDB's minimum share of CPU utilization within a CDB. If the sum of the CPU_MIN_COUNT values across all open PDBs in a CDB is equal to the value of CPU_MIN_COUNT for the CDB, then the CDB instance is considered full. If the sum exceeds the value of CPU_MIN_COUNT for the CDB, then the CDB instance is over-provisioned. Oracle does not prevent you from over-provisioning a CDB.

Resource manager is enabled at the CDB level by setting the RESOURCE_MANAGER_PLAN at the root level to the name of a CDB resource plan. If the CDB resource plan has no configured CPU directives, that is, the SHARES and UTILIZATION_LIMIT directives are unset, then resource manager uses the CPU_COUNT and CPU_MIN_COUNT settings for the PDB to manage CPU utilization.

Recommendation: As a best practice, unset the parameter CPU_COUNT in the initialization parameter file to take advantage of the CPU hot add feature unless otherwise necessary.

For further information, review:

- Oracle *CPU_COUNT* & *CPU_MIN_COUNT*
- Oracle *Dynamic CPU Scaling in Oracle Database 19c*

Memory Best Practices

Oracle and Static and Traditional HugePages

Oracle automatic memory management (AMM) and OS hugepages are not compatible. When you use AMM, the entire SGA memory is allocated by creating files under /dev/shm. When Oracle Database allocates SGA with AMM, hugepages are not reserved.

To use hugepages on, disable AMM.

Once the number of hugepages for Oracle SGA is calculated, the following parameter needs to be set in the /etc/sysctl.conf or an appropriate sysctl configuration file in the /etc/sysctl.d directory:

> vm.nr_hugepages = X where X = number of hugepages for Oracle SGA calculated as per the Metalink note above.

For example, for an SGA of 192GB, the hugepages pool size is set to 98306:

```
oracle@oracle1912-oel83:ora19c:/etc/sysctl.d> grep -i hugepages 99-sysctl.conf
vm.nr_hugepages=98306 #192G
oracle@oracle1912-oel83:ora19c:/etc/sysctl.d>
```

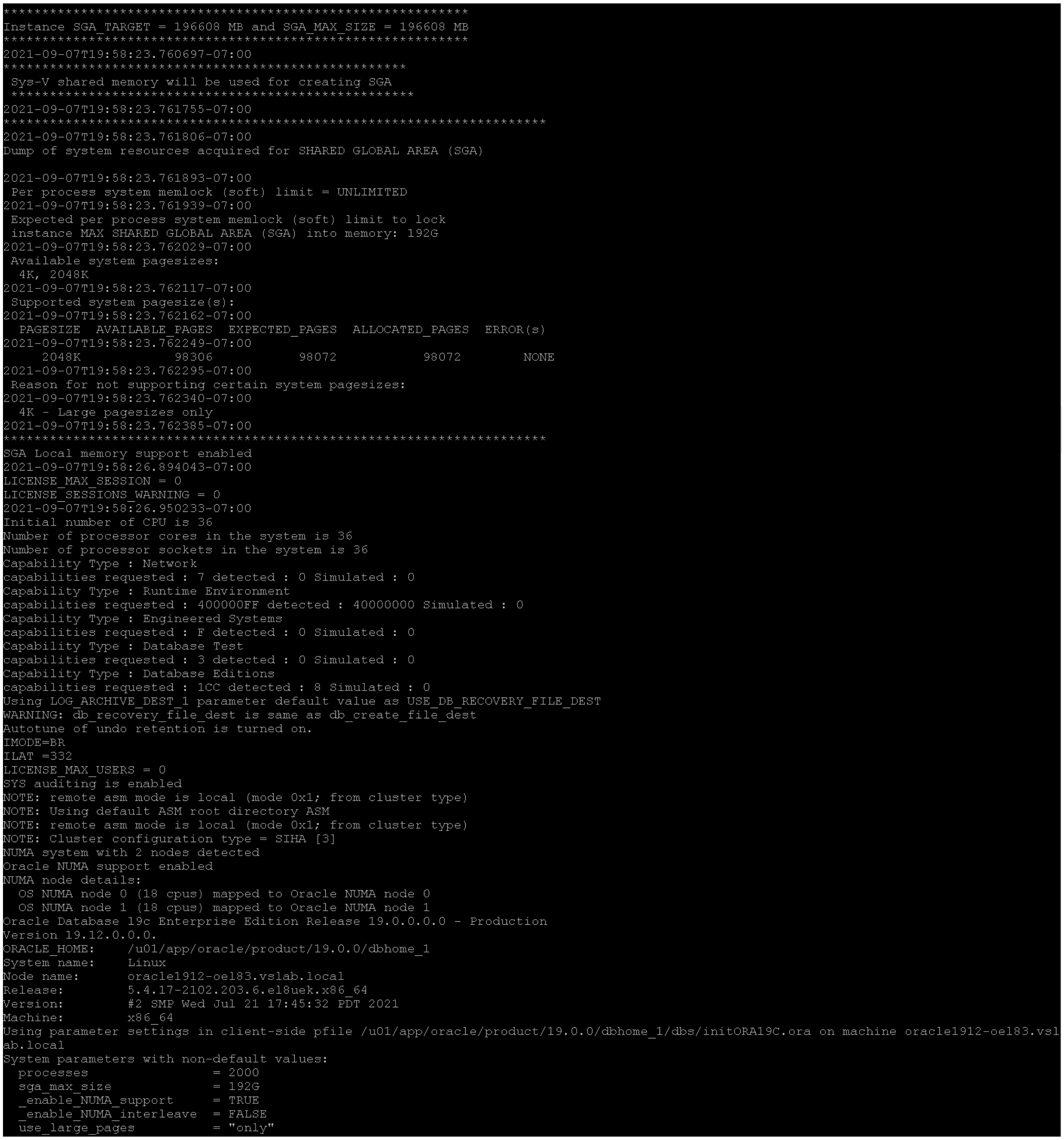
The Oracle database can take advantage of Linux hugepages by setting the initialization parameter USE_LARGE_PAGES.

USE_LARGE_PAGES is used to manage the database's use of large pages for SGA memory with values (TRUE, FALSE, AUTO, ONLY) with TRUE as the default value.

The description of the various values is as shown below:

- TRUE – Specifies that the instance can use large pages if large pages are configured on the system
- FALSE – Specifies that the instance will not use large pages
- AUTO – Specifies that, during startup, the instance will calculate and request the number of large pages it requires. If the operating system cannot fulfill this request, then the instance will start with a combination of large and regular pages.
- ONLY – Specifies that the instance will fail to start if large pages cannot be used for the entire SGA memory. Oracle recommends this setting for consistent performance.

Setting USE_LARGE_PAGES=ONLY in the initiation parameter results in the startup of the database using the 2M hugepages as shown in the alert log below:



```
************************************************************
Instance SGA_TARGET = 196608 MB and SGA_MAX_SIZE = 196608 MB
************************************************************
2021-09-07T19:58:23.760697-07:00
****************************************************
 Sys-V shared memory will be used for creating SGA
 ****************************************************
2021-09-07T19:58:23.761755-07:00
**********************************************************************
2021-09-07T19:58:23.761806-07:00
Dump of system resources acquired for SHARED GLOBAL AREA (SGA)

2021-09-07T19:58:23.761893-07:00
 Per process system memlock (soft) limit = UNLIMITED
2021-09-07T19:58:23.761939-07:00
 Expected per process system memlock (soft) limit to lock
 instance MAX SHARED GLOBAL AREA (SGA) into memory: 192G
2021-09-07T19:58:23.762029-07:00
 Available system pagesizes:
  4K, 2048K
2021-09-07T19:58:23.762117-07:00
 Supported system pagesize(s):
2021-09-07T19:58:23.762162-07:00
  PAGESIZE  AVAILABLE_PAGES  EXPECTED_PAGES  ALLOCATED_PAGES  ERROR(s)
2021-09-07T19:58:23.762249-07:00
     2048K            98306           98072            98072        NONE
2021-09-07T19:58:23.762295-07:00
 Reason for not supporting certain system pagesizes:
2021-09-07T19:58:23.762340-07:00
  4K - Large pagesizes only
2021-09-07T19:58:23.762385-07:00
**********************************************************************
SGA Local memory support enabled
2021-09-07T19:58:26.894043-07:00
LICENSE_MAX_SESSION = 0
LICENSE_SESSIONS_WARNING = 0
2021-09-07T19:58:26.950233-07:00
Initial number of CPU is 36
Number of processor cores in the system is 36
Number of processor sockets in the system is 36
Capability Type : Network
capabilities requested : 7 detected : 0 Simulated : 0
Capability Type : Runtime Environment
capabilities requested : 400000FF detected : 40000000 Simulated : 0
Capability Type : Engineered Systems
capabilities requested : F detected : 0 Simulated : 0
Capability Type : Database Test
capabilities requested : 3 detected : 0 Simulated : 0
Capability Type : Database Editions
capabilities requested : 1CC detected : 8 Simulated : 0
Using LOG_ARCHIVE_DEST_1 parameter default value as USE_DB_RECOVERY_FILE_DEST
WARNING: db_recovery_file_dest is same as db_create_file_dest
Autotune of undo retention is turned on.
IMODE=BR
ILAT =332
LICENSE_MAX_USERS = 0
SYS auditing is enabled
NOTE: remote asm mode is local (mode 0x1; from cluster type)
NOTE: Using default ASM root directory ASM
NOTE: remote asm mode is local (mode 0x1; from cluster type)
NOTE: Cluster configuration type = SIHA [3]
NUMA system with 2 nodes detected
Oracle NUMA support enabled
NUMA node details:
  OS NUMA node 0 (18 cpus) mapped to Oracle NUMA node 0
  OS NUMA node 1 (18 cpus) mapped to Oracle NUMA node 1
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.12.0.0.0.
ORACLE_HOME:    /u01/app/oracle/product/19.0.0/dbhome_1
System name:    Linux
Node name:      oracle1912-oel83.vslab.local
Release:        5.4.17-2102.203.6.el8uek.x86_64
Version:        #2 SMP Wed Jul 21 17:45:32 PDT 2021
Machine:        x86_64
Using parameter settings in client-side pfile /u01/app/oracle/product/19.0.0/dbhome_1/dbs/initORA19C.ora on machine oracle1912-oel83.vsl
ab.local
System parameters with non-default values:
  processes                = 2000
  sga_max_size             = 192G
  _enable_NUMA_support     = TRUE
  _enable_NUMA_interleave  = FALSE
  use_large_pages          = "only"
```

FIGURE 17. Oracle Database Alert Log

Recommendation: As a best practice, consider using Oracle automatic shared memory management (ASMM) and traditional hugepages for performance reasons. It's recommended to set the initialization parameter USE_LARGE_PAGES to ONLY for consistent performance as well.

For further information, review:

- Oracle *HugePages on Linux: What It Is… and What It Is Not… (Doc ID 361323.1)*
- *Oracle Linux: Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration (Doc ID 401749.1)*
- Oracle *USE_LARGE_PAGES*
- Oracle *Using HugePages on Linux*
- *Accelerating Oracle workloads with vSphere 6.7 Guest 2M & GB Huge Pages – An Investigation*

PRE_PAGE_SGA and LOCK_SGA Initialization Parameters

In systems with Linux hugepages configured, the shared segments cannot be locked or unlocked, and the lock flag never gets set by the kernel. Hugepages are universally regarded as pinned.

On Linux, the Oracle database initialization parameters PRE_PAGE_SGA, and LOCK_SGA can be set to TRUE to pin the SGA pages into memory. But hugepages already makes the SGA pages non-swappable, so those parameters are not needed to have SGA pages unswappable.

For further information, review:

- Oracle *LOCK_SGA Not Working (Doc ID 1274096.1)*

GIMR and HugePages Memory

Oracle Grid Infrastructure installations include the grid infrastructure management repository (GIMR). When hugepages is configured on cluster member nodes, the GIMR system global area (SGA) is installed into hugepages memory. The GIMR SGA occupies up to 1GB of hugepages memory. Oracle Grid Infrastructure starts up before Oracle Database is installed on the cluster.

Recommendation: As a best practice, consider allocating memory to hugepages large enough for all databases intended to run SGA on the cluster and to accommodate the SGA for the grid infrastructure management repository.

For further information, review:

- Oracle *About the Grid Infrastructure Management Repository*
- Oracle *Overview of HugePages*

Storage Best Practices

Oracle Optimized Flexible Architecture (OFA)

Oracle Optimal Flexible Architecture (OFA) rules help to organize database software and configure databases to allow multiple databases of different versions, owned by different users, to coexist.

In earlier Oracle Database releases, the OFA rules provided optimal system performance by isolating fragmentation and minimizing contention. In current releases, OFA rules provide consistency in database management and support, and simplify expanding or adding databases or adding additional hardware.

By default, Oracle Universal Installer places Oracle Database components in directory locations and with permissions in compliance with OFA rules. Oracle recommends that you configure all Oracle components in accordance with OFA guidelines.

Recommendation: As a best practice, Oracle recommends that you accept the OFA default. Following OFA rules is especially of value if the database is large, or if you plan to have multiple databases.

For further information, review:

- Oracle *About the Optimal Flexible Architecture Standard*

Oracle Database Layout Considerations

Generally, databases have different disk access patterns as shown in the following table:

OPERATION	RANDOM/SEQUENTIAL	READ/WRITE
OLTP – Transaction Log	Sequential	Write
OLTP – Data	Random	Read/Write
Bulk Insert	Sequential	Write
Read Ahead (DSS, Index Scans)	Sequential	Read
Backup	Sequential	Read

TABLE 2. Typical Oracle Disk Access Patterns

The following general guidelines can help to achieve best performance:

- Oracle Grid Infrastructure and RDBMS binaries can be installed on the OS VMDK as long as the OS VMDK has adequate space for the operating system files OR are installed on a dedicated VMDK
- It's recommended to place Oracle data, redo logs, archivelogs and backups files into separate PVSCSI controllers as the IIO access patterns are different for each type. Oracle accesses data and transaction log files with very different I/O patterns. While data file access is mostly random, transaction log file access is sequential only.
- VMware highly recommends using multiple paravirtual SCSI (PVSCSI) controllers with multiple VMDKs per PVSCSI controller for Oracle workloads. The use of multiple paravirtual SCSI controllers and multiple VMDKs per PVSCSI controller allows for the execution of several parallel I/O operations inside the guest operating system. It also results in load balancing among the PVSCSI controllers.
- Manage performance of VMFS. The aggregate IOPS demands of all VMs on the VMFS should not exceed the IOPS capability the underlying physical disks.
- Use VMware vSphere® Storage DRS™ for automatic load balancing between datastores to provide space and avoid I/O bottlenecks as per pre-defined rules. Consider scheduling invocation of vSphere Storage DRS for off-peak hours to avoid performance penalties while moving a VM.

For mission-critical databases, it is common practice in physical environments to spread the database over multiple LUNs to maximize I/O performance (e.g., placing log and data files in separate LUNs). Follow similar guidelines when virtualized.

An example layout is shown in the following figure:

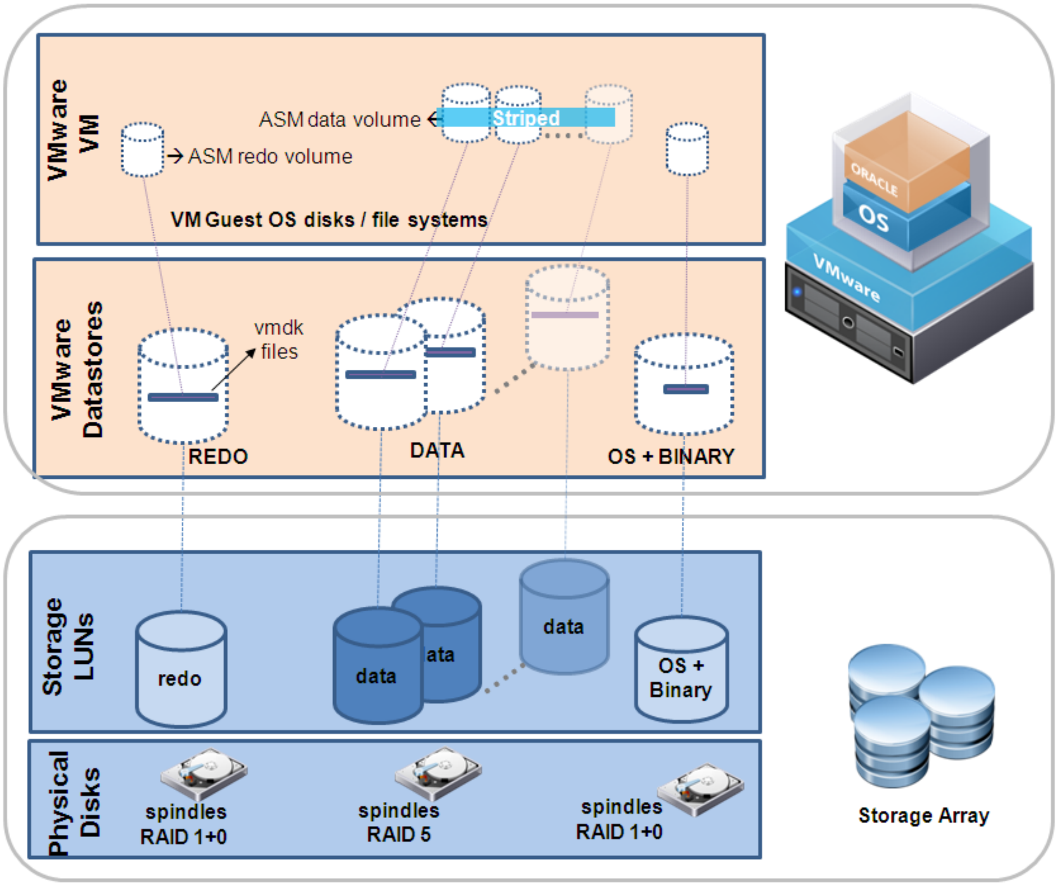


FIGURE 18. Example Storage Layout of Oracle OLTP Database on VMware

The above figure represents an example storage design for a virtualized Oracle OLTP database. The design is based on the following principles:

- At a minimum, an optimized architecture requires collaboration among the database, VMware, and storage administrators.
- Follow storage vendor best practices for database layout on their arrays (as is done in the physical world).

Note that this figure illustrates only an example. Actual configurations for customer deployments can differ.

An example of a two-node Oracle RAC VMDK layout using Oracle ASM is shown below:

- All ASM disks presented on different PVSCSI controllers for performance and queue depth purposes
- Oracle ASM disk groups **DATA_DG** and **FRA_DG** created with external redundancy
- Oracle ASM disk group **REDO_DG** created with external redundancy
- Oracle ASM disk group **CRS_DG** houses the CRS & VOTE disks and set to high redundancy
- Oracle ASM disk group **GIMR_DG** created as part of the grid infrastructure install and houses the grid infrastructure management repository and is set to external redundancy

- Oracle ASM disk group **GRID_DG** created as part of the grid infrastructure install is used for Grid Infrastructure (GI), which is a required component when using Oracle ASM or RAC. Even in single instance (non-clustered), it's recommended to create this ASM disk group so that database data is not mixed with GI management components. It is set to external or normal redundancy.

An example of an Oracle RAC VMDK layout is as shown below:

NAME	SCSI TYPE	SCSI ID (CONTROLLER, LUN)	SIZE (GB)	TYPE	DISK NAME
Operating System (OS) /	Paravirtual	SCSI (0:0)	100	Filesystem	/dev/sda1
Oracle binary disk /u01	Paravirtual	SCSI (0:1)	100	Filesystem	/dev/sdb1
GRID disk 1	Paravirtual	SCSI (0:2)	100	GRID_DG (ASM)	GRID_DISK01
FRA disk 1	Paravirtual	SCSI (0:3)	500	FRA_DG (ASM)	FRA_DISK01
GIMR disk 1	Paravirtual	SCSI (0:4)	250	GIMR_DG (ASM)	GIMR_DISK01
DATA disk 1	Paravirtual	SCSI (1:0)	1024	DATA_DG (ASM)	DATA_DISK01
DATA disk 2	Paravirtual	SCSI (1:1)	1024	DATA_DG (ASM)	DATA_DISK02
DATA disk 1	Paravirtual	SCSI (2:0)	1024	DATA_DG (ASM)	DATA_DISK03
DATA disk 2	Paravirtual	SCSI (2:1)	1024	DATA_DG (ASM)	DATA_DISK04
CRS & VOTE disk 1	Paravirtual	SCSI (3:0)	5	CRS_DG (ASM)	CRS_DISK01
CRS & VOTE disk 1	Paravirtual	SCSI (3:1)	5	CRS_DG (ASM)	CRS_DISK02
CRS & VOTE disk 1	Paravirtual	SCSI (3:2)	5	CRS_DG (ASM)	CRS_DISK03
CRS & VOTE disk 1	Paravirtual	SCSI (3:3)	5	CRS_DG (ASM)	CRS_DISK04
CRS & VOTE disk 1	Paravirtual	SCSI (3:4)	5	CRS_DG (ASM)	CRS_DISK05
REDO disk 1	Paravirtual	SCSI (3:5)	50	REDO_DG (ASM)	REDO_DISK01

TABLE 3. Oracle RAC VMDK Layout

An example of a single-instance Oracle database VMDK layout using Oracle ASM is shown below:

NAME	SCSI TYPE	SCSI ID (CONTROLLER, LUN)	SIZE (GB)	TYPE	DISK NAME
Operating System (OS) /	Paravirtual	SCSI (0:0)	100	Filesystem	/dev/sda1
Oracle binary disk /u01	Paravirtual	SCSI (0:1)	100	Filesystem	/dev/sdb1
GRID disk 1	Paravirtual	SCSI (0:2)	100	GRID_DG (ASM)	GRID_DISK01

FRA disk 1	Paravirtual	SCSI (0:3)	500	FRA_DG (ASM)	FRA_DISK01
DATA disk 1	Paravirtual	SCSI (1:0)	1024	DATA_DG (ASM)	DATA_DISK01
DATA disk 2	Paravirtual	SCSI (1:1)	1024	DATA_DG (ASM)	DATA_DISK02
DATA disk 1	Paravirtual	SCSI (2:0)	1024	DATA_DG (ASM)	DATA_DISK03
DATA disk 2	Paravirtual	SCSI (2:1)	1024	DATA_DG (ASM)	DATA_DISK04
REDO disk 1	Paravirtual	SCSI (3:0)	50	REDO_DG (ASM)	REDO_DISK01

TABLE 4. Single-Instance Oracle Databased VMDK Layout

The Oracle RAC and single-instance database VMDK layouts reflect the principles regarding throughput, load balancing and fairness using:

- Multiple PVSCSI adapters
- Multiple VMDKs per PVSCSI controller
- PVSCSI controllers and VMDKs with increased queue depth

An example of deploying an Oracle RAC on the VMware vSphere platform is shown and explained in *Oracle VMware Hybrid Cloud High Availability Guide*.

Recommendation: As a best practice, increase PVSCSI and VMDK queue depth to the maximum in order to push larger I/O bandwidth, unless the underlying storage vendor recommends otherwise. Use multiple PVSCSI adapters with VMDKs spread across PVSCSI controllers for load balancing and fairness.

For further information, review:

- Oracle *I/O Configuration*
- *PVSCSI Controllers and Queue Depth – Accelerating performance for Oracle Workloads*
- *PVSCSI Controllers and Queue Depth – ASM SAME and Oracle Workloads*

Oracle ASM Considerations

Recommendation: As a best practice, when creating ASM disk groups, consider the following Oracle recommendations:

- Create ASM disk groups with equal disk types and geometries. An ASM disk group is essentially a grid of disks. Group performance is limited by its slowest member.
- Oracle ASM data distribution policy is capacity-based. Ensure that Oracle ASM disks in a disk group have the same capacity to maintain balance.
- A database can span multiple ASM disk groups but ensure that each ASM disk group is mounted and used by one database exclusively. This enables independent optimization of storage-based operations for each individual database. It's recommendation to avoid mixing production and non-production workloads into the same disk groups.
- Create multiple ASM disk groups based on I/O characteristics. Oracle recommends creating a separate disk group, for example:
  - +DATA – ASM disk group used for the database data, such as data files, control files, undo tablespace, and system tablespace
  - +REDO – ASM disk group used for the database redo logs

  - +TEMP (Optional) – When TEMP is very active and large, one could separate TEMP to its own ASM disk group
  - +FRA – ASM disk group used for database archive logs and flashback logs (if used). Note that flashback logs might consume much larger capacity than archive logs.
  - +GRID – ASM disk group used for grid infrastructure (GI), which is a required component when using Oracle ASM or RAC. Even in single-instance (non-clustered) deployments, it's recommended to create the ASM disk group so that database data is not mixed with GI management components.
  - +GIMR – ASM disk group grid infrastructure management repository (GIMR) for Oracle RAC
  - +CRS – ASM disk group for Oracle Cluster Registry (OCR) and voting files for Oracle RAC
- Consider configuring the ASM disk groups with external redundancy if the storage can provide protection through hardware mirrors or copies of the primary LUN. When using external redundancy, disk failures are transparent to the database and consume no additional database CPU cycles because of offload to storage processors.
- A disk partition can be the entire disk drive or a section of a disk drive. However, the Oracle ASM disk cannot be in a partition that includes the partition table because the partition table would be overwritten.
- Use the Oracle ASM Filter Driver feature to provide consistent device naming and permission persistency
- Using multiple ASM disks allows for multiple I/O queues in the OS that can then assist with performance. However, too many disks may prove cumbersome to manage. Operational efficiency and performance are key considerations. When choosing how many disks to use in a new ASM disk group, one option is to determine the expected data size after the first 12 months and then factor in the expected rate of growth. Because all disks in a disk group should be of uniform size, a smaller initial disk size will provide for more granular growth. A larger disk size will result in a fewer number of disks initially and will allow for larger capacity growth without demanding an excessive number of new disks later. It's generally recommended to use a minimum of four ASM disks per ASM disk group, though this may vary by use case.
- Initialization parameters FILESYSTEMIO_OPTIONS and DISK_ASYNCH_IO are not needed when using Oracle ASM, as FILESYSTEMIO_OPTIONS controls the behavior of I/O options given by the operating system but has no effect when ASM is used.
  - FILESYSTEMIO_OPTIONS enables asynchronous I/O (AIO) when the value is set to SETALL/ASYNC. However, ASM bypasses the filesystem layer in this case because ASM I/O is entirely controlled by the DISK_ASYNCH_IO parameter. AIO needs to be enabled or disabled by setting disk_asynch_io to parameter values TRUE/FALSE. It's recommended to still set *filesystemio_options* to SETALL since some database files, notably RMAN backup sets, may still be generated on file systems not created from ASM volumes.
  - The DISK_ASYNCH_IO parameter defaults to TRUE. The *disk_asynch_io* parameter enables asynchronous I/O to the storage subsystem. Without this option enabled, all I/O will be synchronous. Since Oracle ASM bypasses the traditional file system, the *disk_asynch_io* parameter is the only option for controlling synchronous or asynchronous I/O to database files in ASM. Async I/O offers significant performance benefits to most Oracle databases.

For further information, review:

- Oracle *ASM Inherently Performs Asynchronous I/O Regardless of filesystemio_options Parameter (Doc ID 751463.1)*
- Oracle *How to Check if Asynchronous I/O is Working On Linux (Doc ID 237299.1)*
- Oracle *Recommendations for Storage Preparation*

Oracle ASM Allocation Unit (AU) Considerations

Every Oracle ASM disk is divided into allocation units (AU). An allocation unit is the fundamental unit of allocation within a disk group. A file extent consists of one or more allocation units. An Oracle ASM file consists of one or more file extents.

When you create a disk group, you can set the Oracle ASM allocation unit size with the AU_SIZE disk group attribute. The values can be 1, 2, 4, 8, 16, 32, or 64MB, depending on the specific disk group compatibility level.

Larger AU sizes typically provide performance advantages for data warehouse applications that use large sequential reads and very large databases (VLDBs).

Beginning with Oracle 11g, Oracle recommends 4MB for the AU size for a disk group. 8MB AUs might also be a good choice. Some benefits of using 4MB AU and 8MB AU are:

- Increased I/O throughput of the I/O subsystem
- Reduced SGA size to manage the extent maps in the database instance

- Faster datafile initialization
- Increased file size limits
- Reduced database open time
- 8MB AU may deliver improved performance if the database is large, as there are fewer AUs to manage

The objective in setting AU size is to define it small enough that the AUs do not become too hot and large enough for optimal sequential reads while reducing the number of AUs to manage.

Recommendation: As a best practice, consider an AU size of between 1MB and 4MB for mostly OLTP workloads and an 8MB or larger AU size for OLAP workloads. For very large databases, consider an AU size of 8MB or 16MB or an even larger AU size for disk groups in which the majority of the data will be stored.

For further information, review:

- Oracle *About Oracle ASM Allocation Units*

Oracle ASM SAME and Striping Considerations

The stripe and mirror everything (SAME) methodology has been recommended by Oracle for many years as a means to optimize high availability, performance, and manageability. Oracle ASM simplifies storage management through the principle of SAME. Oracle ASM striping has two primary purposes:

- To balance loads across all the disks in a disk group in order to maximize I/O bandwidth
- To reduce I/O latency

Coarse-grained striping provides load balancing for disk groups while fine-grained striping reduces latency for certain file types by spreading the load more widely.

To stripe data, Oracle ASM separates files into stripes and spreads data evenly across all the disks in a disk group. By default, ASM uses an Allocation Unit (AU) size of 1MB (4MB starting release 12.2) and stripes the data across the whole disk group by using the AU as its stripe-depth.

The fine-grained stripe size always equals 128KB in any configuration; this provides lower I/O latency for small I/O operations. The coarse-grained stripe size is always equal to the AU size (not the data extent size).

For storage arrays where the track size is 128KB, its recommended to go with fine-grain striping as the striping method for workloads with primarily sequential writes. Because sequential writes tend to be large I/Os, breaking them in to 128 KB stripes improves latency and allows the storage array to handle them more efficiently.

Recommendation: As a best practice, consider using ASM fine-grain striping for the redo logs. This method is especially useful for in-memory databases (i.e., where transactions are fast and redo write load can be heavy) or for batch data loads (i.e., as again, the redo logs write load is heavy). ASM fine-grain striping is also recommended for control files and flashback logs. Coarse-grained striping can be used for all other files.

For further information, review:

- Oracle *Stripe and Mirror Everything (SAME)*
- Oracle *Understanding Oracle ASM Striping*

Oracle ASM 4k Blocksize

Oracle Database and Grid Infrastructure have supported 4K devices since version 11g release 2. Linux OS requires kernel version 2.6.32 or later for 4K device support. Beginning with Oracle release 12.1, ASM only supports 512e drives.

By **default**, database online redo log files have a block size of 512 bytes. I/O requests sent to the redo log files are in increments of the redo block size. This is the blocking factor Oracle uses within redo log files.

Beginning with Oracle Database version 11.2, this can be changed with the BLOCKSIZE clause to values of 512, 1024 or 4096. For example:

```
SQL> ALTER DATABASE ADD LOGFILE GROUP 5 SIZE 100M BLOCKSIZE 4096;
```

For a 512e emulation device, the logical sector size is 512 bytes, and the physical sector size is 4096 bytes (4K). Hence, the above command will fail unless the sector size check is overridden by setting the underscore parameter _disk_sector_size_override to true in the database instance (do not set this parameter in the ASM instance) using the following command:

```
ALTER SYSTEM SET "_DISK_SECTOR_SIZE_OVERRIDE"="TRUE";
```

At this point, online redo logs can be created with the desired 4096 byte block size.

Note: For a 512e emulation device, significant performance degradation will occur when a redo log write is not aligned with the beginning of the 4K physical sector. 4KB sector drives mainly affect the redo log files (online/standby/archived).

The following are considerations for ASM disk group block size based on database components:

- Log files (online/standby/archived redo log files)
  - It's recommended to create 4KB log block size logs on 4KB native mode disks.
  - For 4KB emulation-mode disks, redo block size could be 512bytes or 4096 bytes (4KB is the preferred block size).
  - SQL provides redo log block size:
    - Select lebsz from x$kccle
- Datafiles
  - With 4KB sector disks, it's recommended to create 4KB or larger database block size (db_block_size) datafiles on 4KB emulation-mode disks/
  - On 4KB native-mode disks, it's recommended to create 4KB block or larger datafiles.
  - SQL provides database block size:
    - Select value from v$parameter where name = 'db_block_size';
- Controlfile
  - The control file block is 16KB, so the 4KB sector size does not affect control files
  - SQL provides control file block size:
    - Select cfbsz from x$kcccf
- SPFILE
  - As of this writing, SPFILE continues to have issues with 4K block size ASM disk groups. It's recommended to use a 512 sector size disk group for SPFILE or start the instance using pfile.
- OCR/Voting Disks
  - The block size for both Oracle Cluster Registry (OCR) and Cluster Synchronization Services (CSS) voting files is 512 bytes. I/O operations to these file objects are therefore sized as a multiple of 512 bytes.

vSphere 6.7 released 4kn local storage support. vSphere 6.5/6.7/7.x and vSAN 6.5/6.7/7.x and later support 512e drives as direct attached drives.

Currently, vSphere and vSAN will expose 512n (both logical and physical sector sizes are 512 bytes) to the guest OS.

For further information, review:

- Oracle *FAQ: Flash Storage with ASM (Doc ID 1626228.1)*
- Oracle *Using 4k Redo Logs on Flash, 4k-Disk and SSD-based Storage (Doc ID 1681266.1)*
- Oracle *Supporting ASM on 4K/4096 Sector Size (SECTOR_SIZE) Disks (Doc ID 1630790.1)*
- Oracle *Bug 16870214 DB startup fails with ORA-17510 if spfile is in 4k sector size diskgroup*
- *FAQ: Support statement for 512e and 4K Native drives for VMware vSphere and vSAN (2091600)*
- *Device Sector Formats*
- *Support for 4Kn HDDs*

Oracle ASM Maximum I/O Size

The hidden Oracle parameter *_asm_maxio* governs the size of the largest I/O that ASM will make to the storage subsystem. The parameter defaults to 1MB on ASM 10g, 11g and 12c.

The SQL statement gives the value of _asm_maxio parameter value on 19c:

```
SQL> select x.ksppinm name, y.ksppstvl value from x$ksppi x, x$ksppcv y where (x.indx = y.indx) and
(x.ksppinm like '%_asm_maxio%');

NAME                        VALUE
--------------------------------------------
_asm_maxio                  1048576
```

Although the parameter may be increased, it does not actually allow I/O to exceed 1MB as this value appears to be a hard limit. Setting maximum I/O to a value less than 1MB will likely increase latency for scanning operations.

The allocation unit does **not** directly govern the size of I/O to the storage subsystem. That is controlled by the extent size or the maximum ASM I/O size parameter.

The allocation unit size **does** control the size of the metadata held in ASM and, for larger databases, there is a performance advantage to using a larger AU size. A 1GB data file using the default allocation unit would require 1024 1MB extents. Setting the AU to 4MB would reduce the extent count to 256.

Oracle internal hard-coded parameter SSTIOMAX is an internal parameter or constant used by Oracle, which limits the maximum amount of data transfer in a single I/O of a read-or-write operation. This parameter is fixed and cannot be tuned or changed.

For further information, review:

- Oracle *About Oracle ASM Allocation Units*
- Oracle *'SSTIOMAX AND DB_FILE_MULTIBLOCK_READ_COUNT IN ORACLE 7 AND 8 (Doc ID 131530.1)*
- Oracle *What Is The Value Of SSTIOMAX And DB_FILE_MULTIBLOCK_READ_COUNT In Oracle 8i, 9i,10g, 11g? (Doc ID 291239.1)*

Oracle ASM Rebalance and Compact Considerations

When ASM rebalances the disk group, there is a compact phase at the end of the rebalance. The compact phase moves the data to the higher performing tracks of the spinning disks.

Beginning with Oracle 12c, it is possible to disable the compact phase, which is enabled (TRUE) by default, on individual ASM disk groups by setting the hidden disk group attribute **_rebalance_compact** to **FALSE**.

```
SQL > ALTER DISKGROUP DATADG SET ATTRIBUTE '_rebalance_compact'='FALSE'';
```

For Oracle pre-12c releases, this phase can only be disabled on the ASM-instance level which affects all ASM disk groups. Since all applications are different and have their own data usage patterns, it is recommended to test with this feature disabled before implementing the change in production.

Recommendation: As a best practice, consider disabling the ASM compact phase, as there may be no benefit to compacting the data on VMware.

For further information, review:

- Oracle *What is compact Phase and how it can be disabled (Doc ID 1902001.1)*

Oracle ASM Thin Provisioning and Space Reclamation

Oracle ASM disk groups have certain attributes, one of them referred to as THIN_PROVISIONED. This attribute enables or disables the functionality to discard unused storage space after a disk group rebalance is completed. The attribute value can be **true** to enable or **false** to disable the functionality. The default value is **false**.

Storage vendor products that support thin provisioning have the capability to reuse the discarded storage space for a more efficient overall physical storage utilization.

Given such support from a storage vendor, when the COMPACT phase of a rebalance operation has completed, Oracle ASM informs the storage which space is no longer used and available for repurposing.

For example –

```
    SQL> alter diskgroup DATA_DG rebalance with balance compact wait;
```

During ASM storage reclamation, ASM first performs a manual rebalance, which defrags (compacts) the ASM disk group by moving ASM extents to gaps created by the deleted data. When the ASM disk group is compacted, its high-water mark (HWM) is updated based on its new allocated capacity. Then, ASM sends SCSI unmap commands to the storage system to reclaim the space above the new HWM.

If a rebalance is not running, you can issue a rebalance with the ALTER DISKGROUP ... REBALANCE SQL statement.

The `THIN_PROVISIONED` attribute is supported only with Oracle ASM Filter Driver (Oracle ASMFD) in Oracle Grid Infrastructure 12.2 and later releases of Linux.

Recommendation: As a best practice, consider using ASM disk groups with the THIN_PROVISIONED attribute for storage space savings.

For further information, review:

- Oracle *THIN_PROVISIONED*
- Oracle *Manually Rebalancing Disk Groups*
- *Reclaiming dead space from Oracle databases on VMware Hybrid Platform*

Oracle ASM Disks Online Resizing

RHEL 7 and later allow resizing disk partitions online without any downtime. As to older style partitions, this feature has been added in the current release of RHEL 7 with a feature request (RFE has been filed to add support for online resizing of disk partitions to RHEL 7 in private RHBZ#853105; with this feature, it's possible to resize the disk partitions online in RHEL 7).

Oracle ASM on RHEL 6, OEL 6 and earlier may use ASM add and drop disks with the rebalance option to resize the ASM disks online without downtime.

By resizing disks, we can grow an ASM disk group without having to add new ASM disks, beginning with RHEL 7 and OEL 7, using Oracle ASMLIB or ASMFD or Linux udev. To speed up this operation, one can rebalance the power limit to the operation to minimize the impact of a rebalance operation on the production system. The higher the rebalance power limit the more resources will be used to rebalance the disk allocation—and the faster the rebalance operation will complete.

By way of example, the high-level steps involved follow below:

- Extend the existing ASM disk from old size to new size online without shutting down database and ASM instances using VMware web client.
- Rescan the OS to see the new disk size. The OS sees the increased size.
- ASM and the database are both still online.
- Delete and re-create the partition table to see the new disk size. ASM still shows the old disk size.
- Run the ASM command 'alter diskgroup <> resize all;' to resize the disk group. The new size is written to the Oracle ASM disk header.
- ASM now reflects the new disk size.

Recommendation: As a best practice, consider using the Oracle ASM disk online resizing feature to avoid downtime.

For further information, review:

- Oracle *Manually Rebalancing Disk Groups*
- Redhat *Does RHEL 7 support online resize of disk partitions?*
- *On Demand hot extend non-clustered Oracle Disks online without downtime – Hot Extend Disks*

Oracle ASM Intelligent Data Placement and Physical Disk Placement

In the past, Oracle DBAs seeking maximum performance sought to place critical files, usually the redo logs, on the outer rim of physical disks to achieve maximum performance.

The introduction of zone bit recorded (ZBR) disks meant that there was a greater volume of data on the outer edges of the physical spinning disk. With the higher relative speed of the head passing over the disk, the transfer rate was up to 50 percent higher on the outer edge.

Since version 11.2.0.3, ASM offers intelligent data placement (IDP) that seeks to place files on the fastest outer parts of disks.

For Oracle workloads on VMware, such complexities are not relevant and will yield little to no performance improvement. The underlying geometry of any spinning disks is hidden from the ESXi host.

The intelligent data placement (IDP) feature has been **deprecated** since Oracle ASM 12c Release 2 (12.2) and support is planned for removal in a future release.

For further information, review:

- Oracle *IDP.BOUNDARY and IDP.TYPE*

Network Best Practices

Oracle Net, a component of Oracle Net Services, enables a network session from a client application to an Oracle Database server. When a network session is established, Oracle Net acts as the data courier for both the client application and the database. It is responsible for establishing and maintaining the connection between the client application and database, as well as exchanging messages between them. Oracle Net can perform these jobs because it is located on each computer in the network.

The Oracle*Net performance tuning is the same on the VMware platform as it is on a physical architecture. Settings would have to be chosen with consideration for VM network physical constraints.

For further information, review:

- *Oracle Net Performance Tuning (Doc ID 67983.1)*
- *Setting SEND_BUF_SIZE and RECV_BUF_SIZE (Doc ID 260984.1)*
- *About Oracle Net Services*
- *Understanding Performance*
- *Optimizing Performance*
- *Assessing and Tuning Network Performance for Data Guard and RMAN (Doc ID 2064368.1)*

Summary

The best practices and guidelines discussed in this document are summarized in this section.

SECTION	SUB-SECTION	RECOMMENDATION FOR BEST PRACTICES	REFERENCES
VM CPU	NUMA	Consider keeping NUMA enabled in server hardware BIOS and at the guest operating system level.	• *Setting the number of cores per CPU in a virtual machine (1010184)* • *Setting corespersocket can affect guest OS topologies (81383)* • *Virtual Machine vCPU and vNUMA Rightsizing – Guidelines* • *Does corespersocket Affect Performance?* • *What is PreferHT and When to Use It* • *Virtual Machines with preallocated memory might be placed NUMA remote at random power-ons (76362)*
	CPU Hot Plug and Hot Add	Consider CPU hot plug when VM NUMA optimization is not needed.	• RedHat *Hot adding CPU/RAM/NIC is supported for VMware guest?* • RedHat *Is it possible to "hotplug" a CPU on a running Red Hat Enterprise Linux?* • *On Demand Scaling up resources for Oracle production workloads – Hot Add CPU and Hot Add Memory*
	CPU Hot Remove	vSphere does not currently support CPU hot remove in its web client. CPU hot remove is an in-guest capability in RHEL 7.1 and later.	• RedHat *How to hot add/remove virtual or real CPUs in RHEL7?* • RedHat *CPU/Memory "Hot-Add" and "Hot-Remove" Support in RHEL version 7* • *Hot Remove CPU and Memory for Oracle production workloads*
VM Memory	Memory Reservation	Consider setting memory reservations equal to the sum of the size of the Oracle SGA, the Oracle PGA, the Oracle background processes stack space, and the operating system used memory in production environments.	• *Oracle Initialization Parameters* • *vSphere Resource Management*
	Memory Hot Plug and Hot Add	Rely on rightsizing rather than memory hot plug and consider memory hot plug for applications which tend to have unpredictable memory requirements.	• Oracle *Initialization Parameters* • RedHat *Hot adding CPU/RAM/NIC is supported for VMware guest?* • *On Demand Scaling up resources for Oracle production workloads – Hot Add CPU and Hot Add Memory*

	Memory Hot Remove	vSphere does not currently support memory hot remove in its web client. Memory hot remove is an in-guest capability currently unsupported by VMware.	• *Accelerating Oracle Performance using vSphere Persistent Memory (PMEM).* • *Accelerating Oracle Performance using vSphere Persistent Memory (PMEM) - Reference Architecture* • *Oracle and vSphere Persistent Memory (PMEM) – vPMEM v/s vPMEMDisk* • *Oracle and vSphere Persistent Memory (PMEM) – Oracle Instance Recovery – An Investigation* • *SAP HANA with Intel Optane Persistent Memory on VMware vSphere*
	Persistent Memory	Consider using persistent memory to accelerate Oracle workloads.	• *Accelerating Oracle Performance using vSphere Persistent Memory (PMEM).* • *Accelerating Oracle Performance using vSphere Persistent Memory (PMEM) - Reference Architecture* • *Oracle and vSphere Persistent Memory (PMEM) – vPMEM v/s vPMEMDisk* • *Oracle and vSphere Persistent Memory (PMEM) – Oracle Instance Recovery – An Investigation* • *SAP HANA with Intel Optane Persistent Memory on VMware vSphere*
VM Storage	General	Consider the use of VMDKs for all Oracle workloads unless there is a specific use case calling for RDMs.	• *Oracle on vSphere – Summary of Storage options*
	Multiple PVSCSI Controllers and PVSCSI/VMDK Queue Depth	As a best practice, increase the PVSCSI and VMDK queue depth to the maximum in order to push larger I/O bandwidth, unless the underlying storage vendor recommends otherwise. Use multiple PVSCSI adapters with VMDKs spread across the PVSCSI controllers for load balancing and fairness.	• *Queues, Queues and more Queues*
	Virtual Disks Hot Add and Hot Remove (Informational)	Oracle workloads can take advantage of hot add and hot remove of virtual hard disks online for storage reconfiguration or capacity planning.	• *On Demand Scaling up / Scaling down Storage resources for Oracle production workloads – Hot Add and Hot Remove non-clustered Disks* • *"RAC" n "RAC" all night – Oracle RAC on vSphere 6.x – Add shared VMDK online without downtime for Oracle RAC ASM / OCFS2* • *To be "RDM for Oracle RAC", or not to be, that is the question [Add Shared RDM in Physical/Virtual Compatibility mode for Oracle RAC]*
	Virtual Disks Hot Extend (Informational)	Oracle workloads can take advantage of hot extend of virtual disks beginning with RHEL 7 and later to allow resizing of disk partitions online without any downtime.	• *On Demand hot extend non-clustered Oracle Disks online without downtime – Hot Extend Disks* • *On Demand hot extend clustered VMDKs online without downtime – Hot Extend RAC clustered disks*

VM Network	Interrupt Coalescing	Consider disabling virtual interrupt coalescing for Oracle RAC interconnect private NIC by setting ethernetX.coalescingScheme=disabled where ethernetX is the vmnic of the Oracle RAC interconnect adapter(s).	• *Low throughput for UDP workloads on Windows virtual machines (2040065)*
	Jumbo Frames and Virtual NIC MTU size	Consider enabling jumbo frames on the virtual switches in which you have configured, for example, Oracle RAC private interconnect, Oracle cache fusion traffic, Oracle backup traffic, or Oracle data loading traffic.	• *Enabling Jumbo Frames on virtual distributed switches (1038827)* • *Virtual network adapters that support jumbo frames (1015556)* • *Using Jumbo Frames on VMware NSX for Oracle Workloads*
	Load Balancing on vSphere Standard Switch and Distributed Switch	Consider the use of Route Based on Originating Virtual Port as the default load balancing method on vSphere Standard Switch and vSphere Distributed Switch for Oracle RAC port groups.	• *Route Based on Originating Virtual Port*
VM Maintenance	Disabling Time Synchronization	Consider using a native time synchronization service – such as NTPD or Chrony in Linux based operating systems or W32Time in Microsoft Windows – and disable time synchronization on the VM.	• *Disabling Time Synchronization (1189)*
	VMware Tools	Consider open-vm-tools for Oracle workloads.	• *Open VM Tools* • *Understanding the Three Types of VM Tools*
In-Guest CPU	OS NUMA	Consider keeping NUMA enabled in ESXi hardware BIOS and at the guest operating system level, which should also be the default settings for NUMA support within most servers and guest operating systems.	• RedHat *How to disable NUMA in RHEL system?* • RedHat *How can I visualize my system's NUMA topology in Red Hat Enterprise Linux?* • RedHat *Working with GRUB 2* • RedHat *numstat*
	numactl Linux utility	Consider installing the numactl utility on the OS to accurately see the vNUMA configuration on the OS.	• RedHat *numactl* • RedHat *How to determine if NUMA configuration is enabled or disabled?*
	Automatic NUMA balancing	Consider turning off automatic NUMA balancing for Oracle workloads as outlined in the Oracle Metalink referenced.	• *Oracle Linux: RHCK 7 Automatic NUMA Balancing Induces High IO Wait (Doc ID 2749259.1)* • Oracle *Requirements for Installing Oracle Database 19c on OL7 or RHEL7 64-bit (x86-64) (Doc ID 2551169.1)* • Oracle *Requirements for Installing Oracle Database/Client 19c on OL8 or RHEL8 64-bit (x86-64) (Doc ID 2668780.1)* • RedHat *Automatic NUMA balancing*

In-Guest Memory	HugeTLB Pages (Static/Traditional HugePage)	Consider using traditional hugepages for Oracle workloads.	• RedHat *How can I configure hugepages in Red Hat Enterprise Linux?* • RedHat *How do I check for hugepages usage and what is using it?* • RedHat *Configuring huge pages*
	Transparent Huge Pages	Consider disabling transparent hugepages because of the various performance issues outlined above by Oracle.	• Oracle *ALERT: Disable Transparent HugePages on SLES11, RHEL6, RHEL7, OL6, OL7, and UEK2 and above (Doc ID 1557478.1)* • RedHat *How to use, monitor, and disable transparent hugepages in RHEL6 and 7?*
	Linux 1GB Huge Pages	Oracle is not currently compatible with 1GB hugepages per the explanation referenced. It's recommended to use 2MB static hugepages until the issue is fixed.	• Oracle *Startup Error ORA-27123: Unable To Attach To Shared Memory Segment Linux-x86_64 Error: 22: Invalid argument (Doc ID 1607545.1)* • RedHat *Configuring huge pages* • *Accelerating Oracle workloads with vSphere 6.7 Guest 2M & GB Huge Pages – An Investigation*
	Tuned Profile tuned-profiles-oracle	Consider setting the Tuned profile to tuned-profiles-oracle for performance reasons.	• Oracle *Tuned Tools Benefit Can Be Used In Oracle RAC Environment (Doc ID 2096977.1)* • RedHat *Getting started with Tuned*
	Memory Swapping	Consider setting vm.swappiness to a value of 1 (the 'swappiness' value can be between 0 and 100).	• Oracle *vm.swappiness - 0 or 1? (Doc ID 1917687.1)* • RedHat *How swappiness sysctl works RHEL7 and under which condition system will choose to swap* • RedHat *Is there a tuned profile available for Oracle RDBMS?* • RedHat *Getting started with Tuned*
	Linux kdump and kernel.panic_on_oops	Consider setting up kdump on Linux OS.	• Oracle *Linux OS Service 'kdump' (Doc ID 550097.1)* • Oracle *How to Configure "kdump" on Oracle Linux 7 (Doc ID 2049919.1)* • Oracle *Configure Crash Dump on RedHat - Kdump recovery service (Doc ID 2003139.1)* • Oracle *Process To Capture Dumps Manually When Linux OS Hangs on Baremetal and Virtual Env (VMware Guest / Oracle VM) (Doc ID 2091397.1)* • Oracle *How to Obtain Kernel Freeze or Hang Information Using Hardware NMI (Doc ID 549779.1)* • RedHat *Has the kernel parameter 'kernel.panic_on_oops' always default been set to 1 in Red Hat Enterprise Linux?* • RedHat *What panic parameters are available for use to panic a system when it is hanging or sluggish?* • Linux *Documentation for /proc/sys/kernel/**
	Linux Out of Memory (OOM)	Right size Oracle memory requirements and avoid memory over-commitment.	• Oracle *How to Configure the Linux Out-of-Memory Killer* • RedHat *How to troubleshoot Out of memory (OOM) killer in Red Hat Enterprise Linux?* • RedHat *How to disable the Out of Memory or oom-killer?*

In-Guest Storage	I/O Scheduling	Consider using noop (RHEL 7 and earlier) or none (RHEL 8 and later) for guest I/O scheduling, as testing has shown that, generally, noop/none (deadline in some cases) performs better for virtualized Linux guests. ESX uses an asynchronous intelligent I/O scheduler and, for this reason, virtual guests should see improved performance by allowing ESX to handle I/O scheduling. Test your workload with the choice of I/O schedulers.	• RedHat *What is the suggested I/O scheduler to improve disk performance when using RHEL with virtualization?* • RedHat *How to use the Noop or None IO Schedulers* • Red *RHEL 7 I/O Schedulers* • Red *RHEL 8 Setting the disk scheduler* • *Tuning options for disk I/O performance in Linux 2.6 kernel-based virtual machines (2011861)*
	Partition Alignment	Align the partition at least 1MB (2048 blocks) boundary to avoid performance I/O degradation caused by I/O crossing track boundaries. It's recommended to seek storage vendor advice for partition alignment offset.	• *Oracle Linux: How To Align Partitions on Large Hard Disk Drives (HDDs) for Optimal Performance (Doc ID 1523947.1)* • Oracle *Configuring Disk Devices to Use Oracle ASMLIB* • RedHat *How can I properly re-align filesystems on un-aligned partitions or LVM devices?* • RedHat *Partition and File System Tools*
	Multipathing	There are no advantages in implementing multipathing in a guest operating system as it does not increase performance unless the storage vendor software requires it.	• Oracle *About Using Oracle ASM with Multipath Disks* • Redhat *Setting up DM Multipath* • *Pluggable Storage Architecture and Path Management*
	Enable SCSI_ID for devices	Enable the parameter disk. EnableUUID in the VM .vmx file.	• RedHat *Why is scsi_id not returning any output in a VM on VMware ESX 4.0 and above* • RedHat *How to check disk.EnableUUID parameter from VM in vSphere for Openshift Container Platform* • *sg_inq command takes a long time to run on ESX 4.0 and later (1029157)*
	Maximum I/O size for block devices	Refer to storage vendor documentation to ensure the maximum allowed OS I/O size matches storage vendor expectations.	• Linux *sysfs documentation* • Linux *Generic block device tuning parameters*
In-Guest Network	Enable Jumbo Frames for Oracle RAC Private Interconnect	Consider making the previously referenced change to the Oracle RAC private interconnect network adapter for enhanced performance of Oracle RAC private interconnect traffic.	• Oracle *Recommendation for the Real Application Cluster Interconnect and Jumbo Frames (Doc ID 341788.1)* • Oracle *How to Modify Private Network Information in Oracle Clusterware (Doc ID 283684.1)* • *Using Jumbo Frames on VMware NSX for Oracle Workloads*
	Receive Side Scaling (RSS) and Multiqueue Support	Consider enabling RSS to increase workload performance.	• RedHat *What is Receive Side Scaling (RSS), and how do I configure it in RHEL?* • *RSS and multiqueue support in Linux driver for VMXNET3 (2020567)*

	TCP Segmentation Offload	Ensure TCP segmentation offload (TSO) is enabled in the guest OS to improve network performance of workloads with severe latency requirements.	• *Understanding TCP Segmentation Offload (TSO) and Large Receive Offload (LRO) in a VMware environment (2055140)* • *TCP Segmentation Offload*
	Large Receive Offload	Consider using large receive offload (LRO) to reduce CPU overhead for processing packets that arrive from the network at a high rate.	• RedHat *How do I enable or disable LRO (Large Receive Offload) or GRO (Generic Receive Offload)?* • *Understanding TCP Segmentation Offload (TSO) and Large Receive Offload (LRO) in a VMware environment (2055140)* • *Poor TCP performance might occur in Linux virtual machines with LRO enabled (1027511)* • *Large Receive Offload*
Oracle CPU	Oracle NUMA	Follow Oracle's recommendation and test it sufficiently in a test environment, apply all the bug fixes or patch sets required for the Oracle database version, review all known NUMA issues in the Oracle database available for download before deciding to use it with production system. In some circumstances, enabling Oracle NUMA support might improve performance.	• Oracle *Enable Oracle NUMA support with Oracle Server Version 11gR2 (Doc ID 864633.1)* • *Oracle NUMA Usage Recommendation (Doc ID 759565.1)* • *To NUMA or not to NUMA – Oracle workloads and NUMA*
	Oracle CPU_COUNT & CPU_MIN_COUNT	Unset the parameter CPU_COUNT in the initialization parameter file to take advantage of the CPU Hot Add feature unless otherwise.	• Oracle *CPU_COUNT* & *CPU_MIN_COUNT* • Oracle *Dynamic CPU Scaling in Oracle Database 19c*
Oracle Memory	Oracle and Static/Traditional Huge Pages	Consider using Oracle automatic shared memory management (ASMM) and traditional hugepages for performance reasons. It's recommended to set the initialization parameter USE_LARGE_PAGES to ONLY for consistent performance as well.	• Oracle *HugePages on Linux: What It Is… and What It Is Not… (Doc ID 361323.1)* • *Oracle Linux: Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration (Doc ID 401749.1)* • Oracle *USE_LARGE_PAGES* • Oracle *Using HugePages on Linux* • *Accelerating Oracle workloads with vSphere 6.7 Guest 2M & GB Huge Pages – An Investigation*
	PRE_PAGE_SGA and LOCK_SGA Initialization Parameters	On Linux, the Oracle Database initialization parameters PRE_PAGE_SGA, and LOCK_SGA can be set to TRUE to pin the SGA pages into memory. But hugepages already makes the SGA pages non-swappable, so these parameters are not needed to have SGA pages unswappable.	• Oracle *LOCK_SGA Not Working (Doc ID 1274096.1)*

	GIMR and HugePages Memory	Consider allocating memory to hugepages large enough for all databases planned to run SGA on the cluster and to accommodate the SGA for the Grid Infrastructure management repository.	• Oracle *About the Grid Infrastructure Management Repository* • Oracle *Overview of HugePages*
Oracle Storage	Oracle Optimized Flexible Architecture (OFA)	Oracle recommends that you accept the OFA default. Following OFA rules is especially of value if the database is large or if you plan to have multiple databases.	• Oracle *About the Optimal Flexible Architecture Standard*
	Oracle Database Layout Considerations	Increase the PVSCSI and VMDK queue depth to the maximum in order to push larger I/O bandwidth, unless the underlying storage vendor recommends otherwise. Use multiple PVSCSI adapters with VMDKs spread across the PVSCSI controllers for load balancing and fairness.	• Oracle *I/O Configuration* • *PVSCSI Controllers and Queue Depth – Accelerating performance for Oracle Workloads* • *PVSCSI Controllers and Queue Depth – ASM SAME and Oracle Workloads*
	Oracle ASM Considerations	Refer to this section details for Oracle recommendations when creating ASM disk groups.	• Oracle *ASM Inherently Performs Asynchronous I/O Regardless of filesystemio_options Parameter (Doc ID 751463.1)* • Oracle *How to Check if Asynchronous I/O is Working On Linux (Doc ID 237299.1)* • Oracle *Recommendations for Storage Preparation*
	Oracle ASM Allocation Unit (AU) Considerations	Consider an AU size of between 1MB and 4MB for mostly OLTP workloads, and an 8MB or larger AU size for OLAP workloads. For very large databases, consider an AU size of 8MB or 16MB or an even larger AU size for disk groups where the majority of the data will be stored.	• Oracle *About Oracle ASM Allocation Units*
	Oracle ASM SAME and Striping considerations	Consider using ASM fine-grain striping for redo logs. This method is especially useful for in-memory databases (i.e., where transactions are fast and redo write load can be heavy) or for batch data loads (i.e., as again, the redo logs write load is heavy). ASM fine-grain striping is also recommended for control files and flashback logs. Coarse-grained striping can be used for all other files.	• Oracle *Stripe and Mirror Everything (SAME)* • Oracle *Understanding Oracle ASM Striping*

	Oracle ASM 4k blocksize	Currently, vSphere and vSAN will expose 512n (both logical and physical sector size is 512 bytes) to the guest OS.	• Oracle *FAQ: Flash Storage with ASM (Doc ID 1626228.1)* • Oracle *Using 4k Redo Logs on Flash, 4k-Disk and SSD-based Storage (Doc ID 1681266.1)* • Oracle *Supporting ASM on 4K/4096 Sector Size (SECTOR_SIZE) Disks (Doc ID 1630790.1)* • Oracle *Bug 16870214 DB startup fails with ORA-17510 if spfile is in 4k sector size diskgroup* • *FAQ: Support statement for 512e and 4K Native drives for VMware vSphere and vSAN (2091600)* • *Device Sector Formats* • *Support for 4Kn HDDs*
	Oracle ASM Maximum IO size	Informational only	• Oracle *'SSTIOMAX AND DB_FILE_MULTIBLOCK_READ_COUNT IN ORACLE 7 AND 8 (Doc ID 131530.1)* • Oracle *What Is The Value Of SSTIOMAX And DB_FILE_MULTIBLOCK_READ_COUNT In Oracle 8i, 9i,10g, 11g? (Doc ID 291239.1)*
	Oracle ASM Rebalance and Compact considerations	Consider disabling the ASM compact phase as there may be no benefit to compacting the data on VMware.	• Oracle *What is compact Phase and how it can be disabled (Doc ID 1902001.1)*
	Oracle ASM Thin Provisioning and Space Reclamation	Consider using ASM disk groups with the THIN_PROVISIONED attribute for storage space savings.	• Oracle *THIN_PROVISIONED* • Oracle *Manually Rebalancing Disk Groups* • *Reclaiming dead space from Oracle databases on VMware Hybrid Platform*
	Oracle ASM Disks Online Resizing	Consider using the Oracle ASM disk online resizing feature to avoid downtime.	• Oracle *Manually Rebalancing Disk Groups* • Redhat *Does RHEL 7 support online resize of disk partitions?* • *On Demand hot extend non-clustered Oracle Disks online without downtime – Hot Extend Disks*
	Oracle ASM intelligent data placement and physical disk placement	Informational only	• Oracle *IDP.BOUNDARY and IDP.TYPE*
Oracle Network		Oracle*Net Performance Tuning is the same on the VMware Platform as it is on a physical architecture. Settings should be chosen with consideration for VM network physical constraints.	• *Oracle Net Performance Tuning (Doc ID 67983.1)* • *Setting SEND_BUF_SIZE and RECV_BUF_SIZE (Doc ID 260984.1)* • *About Oracle Net Services* • *Understanding Performance* • *Optimizing Performance* • *Assessing and Tuning Network Performance for Data Guard and RMAN (Doc ID 2064368.1)*

Conclusion

This document provides best practice guidelines for designing and implementing Oracle single-instances and Oracle RAC in VMs to run on vSphere. The recommendations are not specific to a particular hardware set nor to the size and scope of a particular Oracle implementation. The examples and considerations in this document provide guidance only and do not represent strict design requirements, as varying application requirements can result in many valid configuration possibilities.

The successful deployment of Oracle on VMware hybrid multi-clouds is not significantly different from deploying Oracle on physical servers. DBAs can fully leverage their current skillsets while also delivering the benefits associated with virtualization.

This guide assumes that the reader is conversant with basic administration of a typical VMware vSphere-based infrastructure. References to standard VMware vSphere administration documents are provided as appropriate.

Resources

- *Oracle VMware Hybrid Cloud High Availability Guide*
- *Oracle Business Continuity and Disaster Recovery on VMware Hybrid Multi-Clouds REFERENCE ARCHITECTURE*
- *Virtualizing Oracle Workloads with VMware vSphere Virtual Volumes on VMware Hybrid Cloud*
- *Accelerating Oracle Performance using vSphere Persistent Memory (PMEM) – Reference Architecture*
- *Oracle Monster Virtual Machine Performance on vSphere 6.5*
- *Virtualizing Performance-Critical Database Applications in VMware vSphere 6.0*
- *Oracle 12c and vSphere 6.0 Monster VM*
- *Accelerating Virtualized Oracle 12c performance with vSphere 5.5 Advanced Features Flash Read Cache and vMotion*
- *Oracle RAC and vSphere 5.1 vMotion*
- *Performance Study of Oracle RAC on VMware vSphere 5.0*
- *Oracle 12c RAC Performance on vSphere 5 and EMC*
- *Setting Multi Writer Flag for Oracle RAC on vSphere without any downtime*
- *New method of enabling Multiwriter VMDK flag in vSphere 6.0 Update 1 (UI + API)*
- *About VMware vSphere Flash Read Cache and vFlash Read Cache Deprecation Announced*
- *VXLAN Series – Different Components – Part 1 (Broadcast/Multicast Explained)*
- *Oracle Database on VMware vSAN 6.7*
- *Oracle Database 12c on VMware Virtual SAN 6.2 All-Flash*
- *Oracle Database 12c on VMware VSAN — Day 2 Operations and Management'*
- *Oracle Documentation*
- *Oracle Enterprise Linux Documentation*
- *Oracle Linux Yum Server*
- *Oracle Linux and Unbreakable Enterprise Kernel (UEK) Releases*
- *RedHat Documentation*
- *VMware Documentation*
- *VMware Compatibility Guide*
- *VMware Configuration Maximums*
- *VMware Application Blog*
- *VMware Performance Team blog*
- *VMware vSphere Blog*
- *Oracle on VMware Hybrid Clouds Videos & Demos*

Acknowledgements

Author: Sudhir Balasubramanian, Senior Staff Solution Architect & Global Oracle Practice Lead works in the Cloud Platform CIBG. Sudhir specializes in the virtualization of Oracle business-critical applications. Sudhir has more than 26 years' experience in IT infrastructure and database, working as the Principal Oracle DBA and Architect for large enterprises focusing on Oracle, EMC storage, and Unix/Linux technologies. Sudhir holds a Master's degree in Computer Science from San Diego State University. Sudhir is the Lead Author of the *Virtualize Oracle Business Critical Databases* book, which is a comprehensive authority for Oracle DBAs on the subject of Oracle and Linux on vSphere. Sudhir is a VMware vExpert, Alumni Member of the VMware CTO Ambassador Program and an Oracle ACE.

Thanks to the following people for their contributions:

- **Oleg Ulyanov** – Solutions Architect, Microsoft Applications

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 vmware.com