HPE Reference Architecture for Oracle RAC 12c best practices and scaling on HPE Synergy Gen10 platforms and HPE 3PAR StoreServ 9450 Storage

Reference Architecture

Contents

Executive summary

IT departments are under constant pressure to reduce costs without sacrificing quality or functionality. When implementing and operating huge data repositories with Oracle databases, the amount of compute and storage resources required is significant. One common approach to reducing costs is to standardize on a specific hardware and software stack. This reduces the complexity of the environment and reduces total operating expenses. However, standardizing on a platform that does not offer a full feature set and the latest technology and security defeats the purpose. HPE Synergy and HPE 3PAR storage provide the benefit of standardizing on systems that offer the latest technology and security with Gen10 platforms and provide the ability to scale up and out as needed.

In today's business, the demands of rapid Oracle database implementations continue to escalate. Faster transaction processing speeds, capacity-based scaling, increased flexibility, high availability and business continuity are required to meet the needs of the 24/7 business. In the day-to-day management of Oracle database environments, administrators need to be able to quickly deploy new servers, easily update existing systems, and upgrade processing capabilities for scale-out performance. With traditional infrastructure, these activities are disruptive and time consuming.

This Reference Architecture demonstrates how HPE Synergy Gen10 platforms and HPE 3PAR StoreServ All Flash storage provide an ideal environment for deploying Oracle Real Application Clusters (RAC). HPE Synergy offers fluid resource pools which can be customized for specific database needs. Oracle resources can be deployed rapidly through the software-defined intelligence embedded in the HPE Synergy Composer and HPE Synergy Image Streamer. An administrator can utilize HPE Synergy Image Streamer to develop a deployment plan to install and configure both the operating system and application software. A server profile defined in the HPE Synergy Composer can use that deployment plan to configure a new server in a matter of minutes, compared to hours or days utilizing traditional infrastructure.

This paper also shows that throughput can be scaled linearly as Oracle RAC nodes are added to the cluster, as well as demonstrating that additional processing capacity can be quickly and easily configured by migrating from a dual-processor HPE Synergy 480 Gen10 to a quad-processor HPE Synergy 660 Gen10 Compute Module.

Target audience: This Hewlett Packard Enterprise white paper is designed for IT professionals who use, program, manage, or administer large databases that require high availability and high performance. This information is intended for those who evaluate, recommend, or design new IT high performance architectures. Specifically, organizations deploying Oracle RAC in an HPE Synergy environment can use this paper to guide their implementation plans.

Document purpose: The purpose of this document is to describe a Reference Architecture, highlighting the usage of HPE Synergy and HPE 3PAR StoreServ storage to deploy Oracle RAC solutions. Best practices for configuring the environment, deployment plans for automating the OS and application installation, and relative performance of one to four Oracle RAC nodes are provided.

Solution overview

The key components of this solution are HPE Synergy Composer, HPE Synergy Image Streamer, HPE Synergy Gen10 Compute Modules, HPE 3PAR StoreServ 9450 all-flash storage, HPE Application Tuner Express, and Oracle Real Application Clusters.

HPE Synergy Composer

HPE Synergy Composer provides the enterprise-level management to compose and deploy system resources to match your application needs. This management appliance uses software-defined intelligence with embedded HPE OneView to aggregate Compute, Storage and Fabric resources in a manner that scales to your application needs, instead of being restricted to the fixed ratios of traditional resource offerings.

HPE Synergy Image Streamer

HPE Synergy Image Streamer is a new approach to deployment and updates for composable infrastructure. This management appliance works with HPE Synergy Composer for fast software-defined control over physical compute modules with operating system and application provisioning. HPE Synergy Image Streamer enables true stateless computing combined with the capability for image lifecycle management. This management appliance rapidly deploys and updates infrastructure.

HPE Synergy Image Streamer adds a powerful dimension to 'infrastructure as code'—the ability to manage physical servers like virtual machines. In traditional environments, deploying an OS and applications or hypervisor is time consuming because it requires building or copying the software image onto individual servers, possibly requiring multiple reboot cycles. In HPE Synergy, the tight integration of HPE Synergy Image Streamer with HPE Synergy Composer enhances server profiles with images and personalities for true stateless operation.

HPE Synergy Composer, powered by HPE OneView, captures the physical state of the server in the server profile. HPE Synergy Image Streamer enhances this server profile (and its desired configuration) by capturing your golden image as the 'deployed software state' in the form of bootable image volumes. These enhanced server profiles and bootable OS plus application images are software structures ('infrastructure as code')—no compute module hardware is required for these operations. The bootable images are stored on redundant HPE Synergy Image Streamer appliances, and they are available for fast implementation onto multiple compute modules at any time. This enables bare-metal compute modules to boot directly into a running OS with applications and multiple compute modules to be quickly updated.

Figure 1 shows how HPE Synergy Composer and HPE Synergy Image Streamer manage a compute module via a server profile.

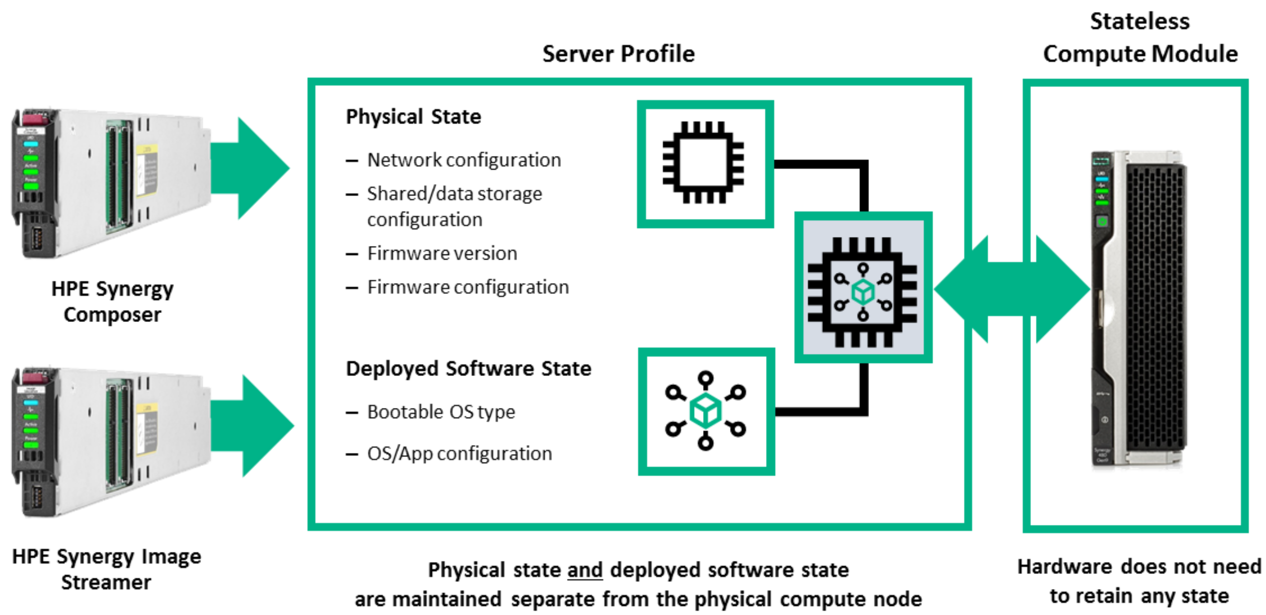

Figure 1. HPE Synergy Composer and HPE Synergy Image Streamer managing compute module with a server profile

Figure 2 shows the HPE Synergy Image Streamer Dashboard, which displays the resources available to create and modify OS images.

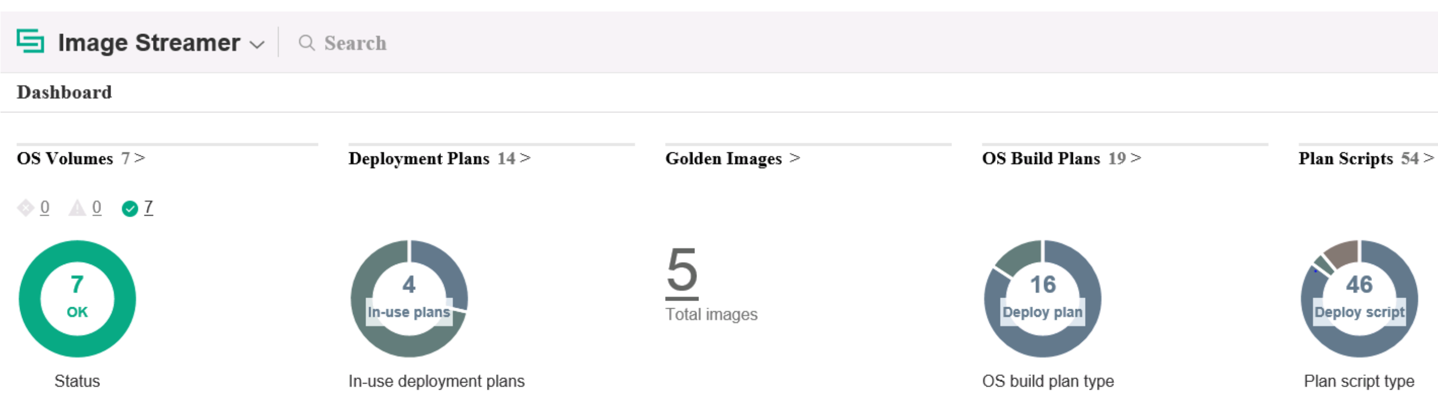

Figure 2. HPE Synergy Image Streamer Dashboard

HPE Synergy Image Streamer supports a variety of operations for flexibility in how you handle your images. For example, you can capture golden images for your use, import golden images from another location, or modify some of your 'known good' images for re-use. This flexibility allows you to easily establish your desired images for use. A variety of images can be used on HPE Synergy Image Streamer. Reference implementations provide artifacts for recent versions of VMware® ESXi (5.0, 6.0 and later), and application images based on Red Hat® Enterprise Linux® (RHEL 6.7 and later, 7.2 and later) and SUSE Linux® (SLES 12 SP1 and later) using ext3 and ext4 file systems. The sample artifacts available for RHEL may be found at https://github.com/HewlettPackard/image-streamer-rhel. The set of Reference Architectures based on Image Streamer, with their associated artifacts are available at https://github.com/HewlettPackard/image-streamer-reference-architectures. You can also enable your own specific images and image types using the tools provided with HPE Synergy Image Streamer.

HPE Synergy Image Streamer building blocks

HPE Synergy Image Streamer uses the following components for capture and deployment of images:

- **Plan script:** A guestfish[1] script used by OS build plans to personalize OS volumes based upon the values of custom attributes.
- **OS build plan:** A set of plan scripts used to modify the configuration of an OS volume during the deployment or capture process.
- **Golden image:** A generic format of an application and operating system image that can be customized for multiple deployments.
- **Deployment plan:** A combination of an OS build plan and golden image that is used by a server profile for the deployment of a server.

HPE Synergy Gen10 Compute Modules

HPE Synergy Gen10 Compute Modules deliver more choice for performance, capacity, efficiency, and flexibility to power most workloads with support for the full range of Intel® Xeon® Scalable Family processors in a 1S, 2S or 4S form factor. They provide a 25% performance increase over prior generations. The compute modules are designed to create a pool of flexible compute capacity within a composable infrastructure. In addition, Gen10 iLO 5 and silicon root of trust are designed to meet challenges such as attacks on firmware, ensuring effective security protections.

The HPE Synergy 480 Gen10 Compute Module comes in a two-socket, half-height form factor to support demanding workloads. Powered by the latest Intel Xeon Scalable processors, HPE Synergy 480 Gen10 utilizes HPE DDR4 SmartMemory supporting up to 3TB. It has flexible storage controller options with up to two SFF drives (4 uFF drives) and/or up to two internal M.2 drives, and three I/O mezzanine slots. It is designed to create a pool of flexible compute capacity within a composable infrastructure, the HPE Synergy 480 Gen10 Compute Module is the ideal platform for general-purpose enterprise workload performance now and in the future.

The HPE Synergy 660 Gen10 Compute Module comes in a four-socket, full-height form factor to support demanding workloads and virtualization density. Powered by 2 or 4 Intel Xeon Scalable family processors, it has 48 slots for HPE DDR4 SmartMemory supporting up to 6TB (with 128 GB DIMMs). It has flexible storage controller options with up to four SFF drives (8 uFF drives) and/or up to four internal M.2 drives, and six (6) I/O mezzanine slots. This makes the HPE Synergy 660 Gen10 Compute Module the ideal platform for virtualization density, high availability, and scale-up enterprise workloads.

Figure 3. HPE Synergy 480 Gen10 and HPE Synergy 660 Gen10 Compute Modules

[1] For more information about the guestfish scripting language, see http://libguestfs.org/guestfish.1.html

HPE 3PAR StoreServ 9450 all-flash storage

HPE 3PAR StoreServ 9450 all-flash storage is an enterprise-class array that helps consolidate primary storage workloads without compromising performance, scalability, data services or resiliency. This newest 3PAR model is built for all-flash consolidation, delivering the performance, simplicity, and agility needed to support a hybrid IT environment. It can scale up to 6000 TiB of raw capacity, and is capable of over 2 million IOPS at sub-millisecond latency. These capabilities are complemented by enterprise-class, Tier-1 features, and functionality. HPE 3PAR StoreServ All Flash is designed for 99.9999% availability with full hardware redundancy, supporting availability objectives for the most demanding environments. Enhanced storage capabilities provide continuous data access and the HPE 3PAR Priority Optimization software offers fine-grained QoS controls to ensure predictable service levels for all applications without physical partitioning of resources.

Figure 4. HPE 3PAR StoreServ 9450 storage

HPE Infosight for 3PAR

In addition, HPE 3PAR customers can now benefit from HPE InfoSight for 3PAR. HPE Infosight is an industry-leading predictive analytics platform that brings software-defined intelligence to the data center with the ability to predict and prevent infrastructure problems before they happen. The first release of HPE Infosight for 3PAR provides the following capabilities:

- **Cross-stack analytics.** For 3PAR customers running the latest release of the 3PAR operating system[2], HPE InfoSight offers the ability to resolve performance problems and pinpoint the root cause of issues between the storage and host virtual machines (VMs). It also provides visibility to locate "noisy neighbor" VMs.
- **Global visibility.** Through a new cloud portal that combines HPE Infosight and HPE StoreFront Remote, all current 3PAR customers with systems that are remotely connected will see detailed performance trending, capacity predictions, health checks and best practice information across all of their 3PAR arrays.
- **Foundation to enable predictive support.** Analytics and automation infrastructure are now in place that in the future will be used to detect anomalies, predict complex problems, and route cases directly to Level 3 support.

[2] Requires 3PAR OS version 3.3.1 GA or later and Service Processor version 5.0.3

HPE 3PAR benefits for Oracle

Oracle customers can use HPE 3PAR StoreServ Storage to address their most significant challenges including:

Performance – HPE 3PAR delivers high throughput and low latencies in multi-tenant, mixed-workload Oracle environments. With industry leading performance and sub-millisecond latencies, HPE 3PAR provides high transactions-per-second (TPS) and minimal wait times for Oracle OLTP workloads, using features such as Priority Optimization and Adaptive Flash Cache[3].

Efficiency and Data Reduction – In many Oracle environments, overprovisioning the primary database has become a matter of survival. As Oracle databases grow, every added core creates additional license and support fees. HPE 3PAR helps reduce Oracle sprawl and simplifies instance consolidation, driving higher TPS and providing a reduced storage footprint. What's more, HPE 3PAR Adaptive Data Reduction technologies, such as compression, can boost storage efficiency while helping enterprises bypass costly Oracle compression license fees.

High Availability and Data Protection – Many Oracle environments face challenges with a growing primary database with an increasing number of applications writing to that database. As databases get larger, backup windows, recovery point objectives, and recovery time objectives become harder to meet. HPE 3PAR offers a broad set of solutions that drive high availability for Oracle, including snapshots and Remote Copy. For direct snapshot copies to HPE StoreOnce Systems as the target backup appliance, HPE 3PAR includes Recovery Manager Central (HPE 3PAR RMC-O) software at no additional cost, delivering fast backup and recovery for Oracle data. The HPE StoreOnce Catalyst Plug-in for Oracle is also offered free of charge and is tightly integrated with Oracle Recovery Manager (RMAN), giving the Oracle database administrator complete visibility and control for backup and recovery tasks. For more information about this solution, see HPE Reference Architecture for Comprehensive Oracle Backup, Restore and Disaster Recovery using HPE RMC and HPE StoreOnce. In addition, HPE 3PAR Peer Persistence can be deployed with Oracle RAC to provide customers with a highly available stretched cluster.

Taken together, these features help Oracle database and storage administrators manage even the most demanding Oracle environments, delivering the performance, data protection, efficiency, and high availability needed to keep critical applications and business processes up and running.

For more details about the HPE 3PAR features that benefit Oracle environments, see Best Practices for Oracle Database on HPE 3PAR StoreServ Storage.

HPE Application Tuner Express (HPE-ATX)

HPE Application Tuner Express is a utility for Linux® users to achieve maximum performance when running on multi-socket servers. Using this tool, you can align application execution with the data in memory resulting in increased performance. The tool does not require any changes to your applications, but runs alongside them. Because many x86 applications today were designed for older systems, HPE-ATX was designed to take advantage of the resources of newer servers to run workloads more efficiently. HPE-ATX offers the following launch policies to control the distribution of an application's processes and threads in a NUMA environment:

- Round Robin: Each time a process (or thread) is created, it will be launched on the next NUMA node in the list of available nodes. This ensures even distribution across all of the nodes.
- Fill First: Each time a process (or thread) is created, it will be launched on the same NUMA node until the number of processes (or threads) matches the number of CPUs in that node. Once that node is filled, future processes will be launched on the next NUMA node.
- Pack: All processes (or threads) will be launched on the same NUMA node.
- None: No launch policy is defined. Any child process or sibling thread that is created will inherit any NUMA affinity constraints from its creator.

HPE-ATX is fully supported by Hewlett Packard Enterprise and can be downloaded from HPE Software Depot.

Oracle Real Application Clusters

Oracle Database with the Oracle Real Application Clusters (RAC) option allows multiple database instances running on different servers to access the same physical database stored on shared storage. The database spans multiple systems, but appears as a single unified database to the application. This provides a scalable computing environment, where capacity can be increased by adding more nodes to the cluster. While all servers in the cluster must run the same OS and the same version of Oracle, they need not have the same capacity, which allows adding servers with more processing power and memory when more performance is required. This architecture also provides high availability, as RAC instances running on multiple nodes provides protection from a node failure.

[3] Adaptive Flash Cache utilizes SSDs as a cache for slower storage devices, and therefore is not needed in the all-flash storage arrays such as the HPE 3PAR StoreServ 9450.

The Oracle RAC architecture is well-suited for HPE Synergy environments, as the HPE Synergy composable infrastructure provides an efficient mechanism for quickly deploying Oracle RAC nodes, as well as adding more HPE Synergy compute modules to the cluster, or scaling up to higher capacity compute modules when more performance is needed.

Solution components

The HPE Synergy components used in this solution included three HPE Synergy 12000 Frames, with two HPE Synergy Composers, two HPE Synergy Image Streamers, five HPE Synergy compute modules, two Virtual Connect SE 40 Gb F8 Modules for Synergy (Master Modules in Figure 5), four HPE Synergy 20Gb Interconnect Link Modules (Satellite Modules in Figure 5), and two HPE Virtual Connect SE 16Gb Fibre Channel Modules for Synergy. For this solution, HPE 3PAR StoreServ 9450 all-flash storage was used for the Oracle database table spaces, indexes and logs, plus the voting disk for Oracle RAC. Shared storage is required for Oracle RAC implementations, and the all-flash performance, as well as mission-critical resiliency of the HPE 3PAR StoreServ 9450 make it ideal for Oracle RAC environments. Figure 5 shows a logical layout of the three-frame HPE Synergy solution with HPE 3PAR StoreServ 9450 Storage.

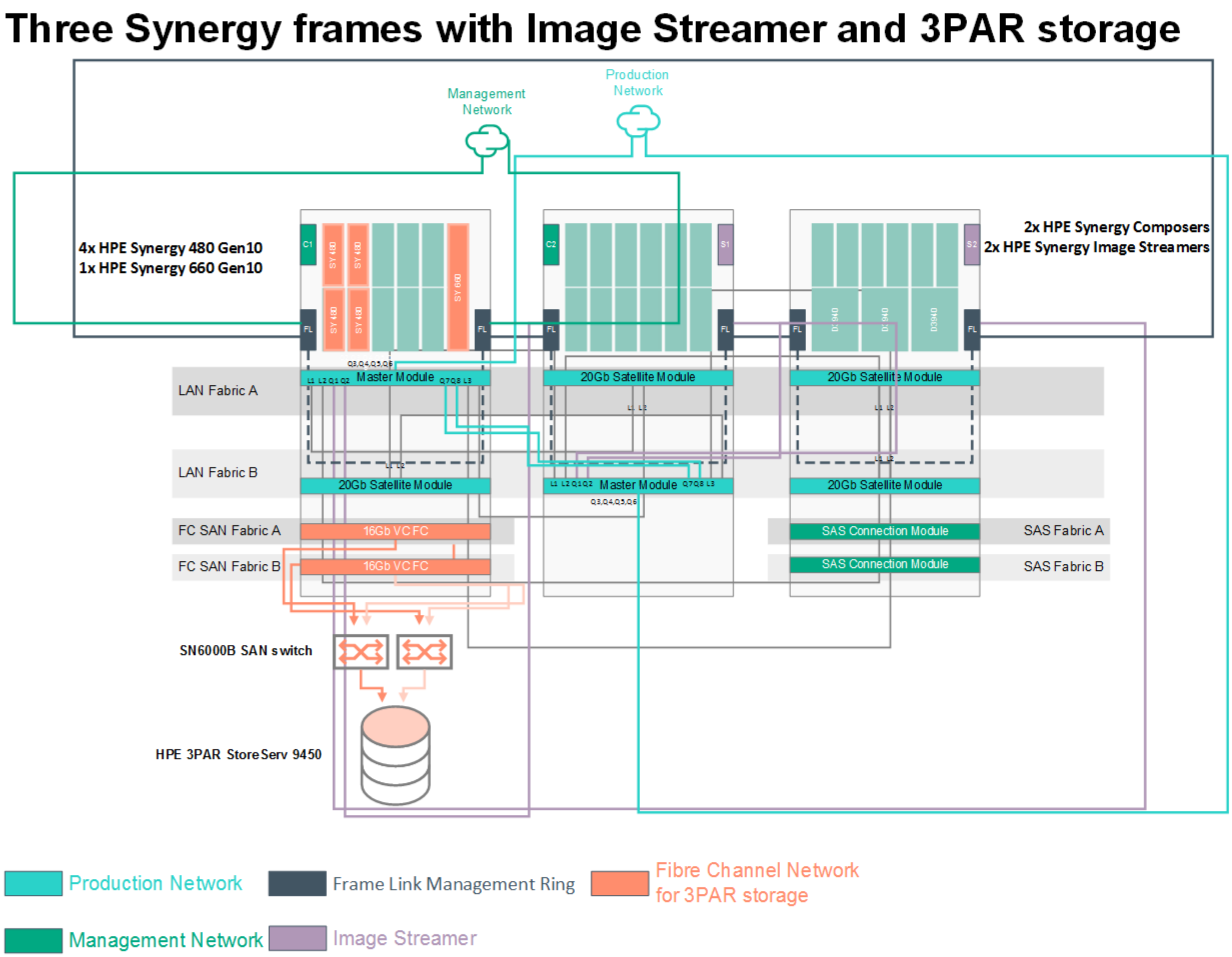

Figure 5. Logical diagram of three-frame HPE Synergy solution with HPE 3PAR StoreServ 9450 Storage

The primary focus of this Reference Architecture is to highlight the benefits of the HPE Synergy platform for Oracle RAC in terms of ease of deployment, manageability, and scalability. Although the RAC cluster nodes were contained in a single Synergy frame, the HPE Synergy plus HPE 3PAR solution for Oracle RAC as tested exceeds 99.999% availability[4]. For environments requiring an even higher level of availability, the configuration shown in Figure 6 is recommended. Note that in this environment, the HPE Synergy compute modules are spread across multiple frames to avoid a single point of failure. In both Figures 5 and 6, there is redundancy in the HPE Synergy Fabric, with two Virtual Connect SE 40Gb F8 Modules for Synergy (labeled Master Module, spread across two frames) and two HPE Virtual Connect SE 16Gb Fibre Channel Modules for Synergy in the frames containing the compute modules for Oracle RAC. In addition, the environment includes two HPE Synergy Composers and two HPE Synergy Image Streamers.

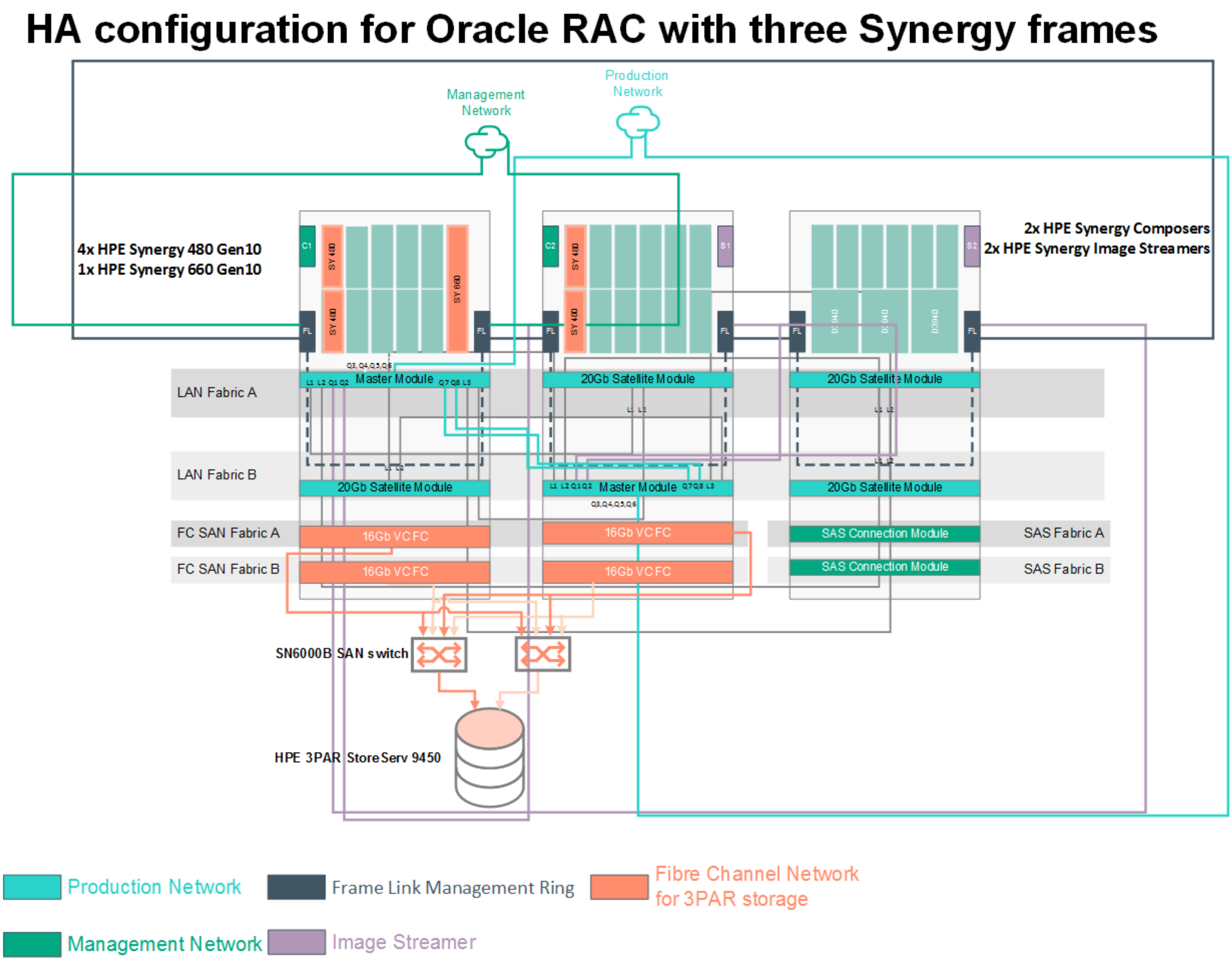

Figure 6. Logical diagram of HA configuration for Oracle RAC with three Synergy frames

[4] The information and data provided herein may vary subject to a broad range of factors and do not extend nor create warranties expressed or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. The only warranties provided by Hewlett Packard Enterprise are set forth in the HPE limited warranty statement.

Hardware

A three-frame HPE Synergy environment was configured to provide redundancy for the HPE Synergy Composer and HPE Synergy Image Streamer. The second and third frames had interconnects and compute modules that are not listed here because they were utilized for different applications.

The three HPE Synergy 12000 Frames had the following components:

- Two HPE Synergy Composers
- Two HPE Synergy Image Streamers
- Four HPE Synergy 480 Gen10 Compute Modules
- One HPE Synergy 660 Gen10 Compute Modules
- Two HPE Virtual Connect SE 40Gb F8 Modules for Synergy
- Four HPE Synergy 20Gb Interconnect Link Modules
- Two HPE Virtual Connect SE 16Gb Fibre Channel Modules for Synergy

The HPE Synergy compute modules were configured as follows:

Four HPE Synergy 480 Gen10 Compute Modules with:

- 2 x Intel Xeon-Platinum 8180 28-core processors with a clock speed of 2.5GHz
- 512GB of memory
- 1 x HPE Synergy 3820C 10/20Gb Converged Network Adapter
- 2 x HPE Synergy 3830C 16G FC HBA

One HPE Synergy 660 Gen10 Compute Module with:

- 4 x Intel Xeon-Platinum 8180 28-core processors with a clock speed of 2.5GHz
- 512GB of memory
- 1 x HPE Synergy 3820C 10/20Gb Converged Network Adapter
- 4 x HPE Synergy 3830C 16G FC HBA

The HPE 3PAR StoreServ 9450 was configured as follows:

- 4-node
- 896GiB of cache
- 8 X drive enclosures
- 80 X 400GB SAS SFF SSDs
- 8 X 4-port 16Gb Fibre Channel ports

Software

Red Hat Enterprise Linux version 7.4 plus the patches to address the Meltdown and Spectre speculative execution security issues (3.10.0-693.21.1 kernel)

Application software

- Oracle Database 12c R2 (12.2.0.1.0) Enterprise Edition (includes Oracle 12c Real Application Clusters)
- Oracle Database 12c R2 (12.2.0.1.0) Grid Infrastructure

Best practices and configuration guidance for Oracle RAC solution on HPE Synergy

BIOS settings for HPE Synergy compute modules

The HPE OneView server profile was used to manage the BIOS settings for all of the HPE Synergy compute modules. The Transactional Application Processing workload profile was configured for this testing. This profile is intended to be used for business processing environments, such as online transaction processing (OLTP) applications. It configures various BIOS settings, including setting the Power Regulator Mode to "static high performance". For more information about Workload Profiles, see UEFI Workload-based Performance and Tuning Guide for HPE ProLiant Gen10 Servers and HPE Synergy.

Linux kernel boot options

The golden image used to deploy the HPE Synergy compute modules had the boot options transparent_hugepage=never and intel_idle.max_cstate=1 added to the kernel boot command line. Transparent huge pages are enabled by default in RHEL, and Oracle recommends disabling this setting to avoid memory allocation delays at runtime. Note that while dynamically-allocated transparent huge pages were disabled, statically-allocated huge pages were configured via the vm.nr_hugepages kernel parameter as described in the Linux kernel settings section below.

Internal HPE benchmarking efforts have demonstrated optimal performance with the option intel_idle.max_cstate=1, which allows Cstate 1 transitions and encourages TurboBoost functionality.

Note

The Intel driver ignores the C-state settings in the BIOS, so it is necessary to add this option to the kernel boot command line.

Linux kernel settings

The RHEL kernel parameters were set in the golden image according to Oracle's recommendations. For the complete list, see Appendix B. Here are a few settings of particular note:

- RHEL automatic NUMA balancing was disabled in the Linux kernel (kernel.numa_balancing=0).
- The number of huge pages (vm.nr_hugepages) was set large enough to hold the Oracle SGA (note that this setting was configurable in an Image Streamer plan script to allow for varying SGA and memory sizes).

HPE 3PAR StoreServ volumes

HPE OneView was used to create shared volumes on the HPE 3PAR StoreServ 9450 for the Oracle RAC database. Volumes with the following attributes were used:

- 8 x 512GB RAID 1 volumes for the database, tablespaces, indexes and undo tablespace. These were configured as the Oracle ASM disk group DATA.
- 8 x 128GB RAID 5 volumes for the first redo log disk group. These were configured as the Oracle ASM disk group REDOa.
- 8 x 128GB RAID 5 volumes for the second redo log disk group. These were configured as the Oracle ASM disk group REDOb.
- 1 x 16GB RAID 5 volume for the RAC voting disk. This was used for the Oracle ASM disk group VOTING.
- 1 x 16GB RAID 5 volume as a database recovery file destination. This was used for the Oracle ASM disk group FRA.
- 1 x 40GB RAID 5 volume for the Grid Infrastructure Management Repository. This was used for the Oracle ASM disk group MGMT.

It should be noted that initial tests comparing RAID 1 and RAID 5 volumes for the redo logs showed no difference in performance, so RAID 5 was used for the final tests.

Figure 7 shows the disk layout that was used for the Oracle RAC configuration.

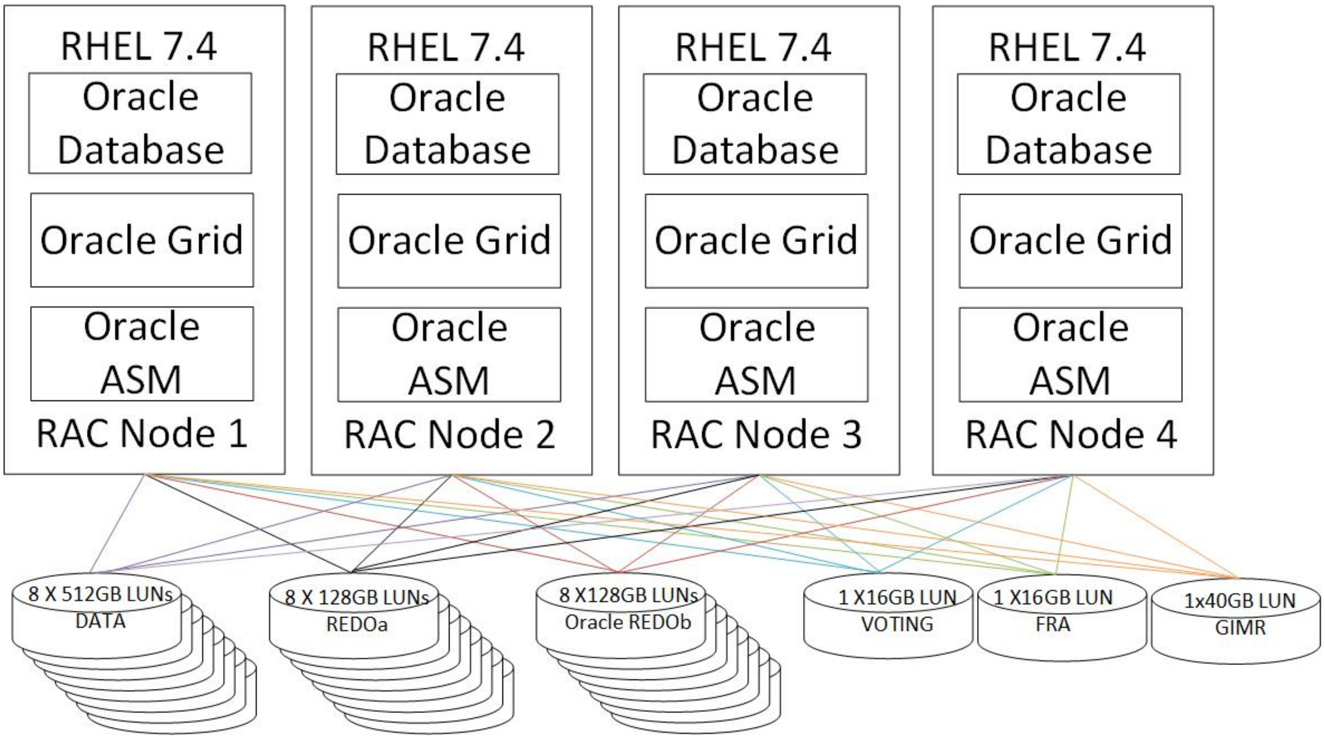

Figure 7. Disk layout for Oracle RAC nodes

Oracle configuration

The following Oracle configuration was used for this testing. For specific Oracle database parameter settings, see Appendix D.

- Two 300GB redo logs per RAC node
- An undo tablespace of 400GB per RAC node
- A temp tablespace of 150GB
- Oracle was configured to only use huge pages (use_large_pages='ONLY')
- HPE-ATX was used to evenly distribute the Oracle processes across all nodes in the server (see Appendix E for ATX configuration script)

Steps to create Image Streamer deployment plan and deploy compute module

The high-level steps for creating 1) a deployment plan, and 2) a server profile for an Oracle database environment are listed below. The details for implementing each of these steps are provided in the following sections.

Steps to develop an HPE Synergy Image Streamer deployment plan

1. Create golden image.
2. Create plan scripts to customize golden image.
3. Create OS build plan using plan scripts.
4. Create deployment plan using OS build plan and golden image.

Steps to deploy a compute module using the HPE Synergy Image Streamer deployment plan

1. Create required networks.
2. Create 3PAR volumes required for Oracle data, logs and voting disk groups.
3. Create server profile utilizing networks, storage and OS deployment plan.

Develop HPE Synergy Image Streamer deployment plan

Step 1: Create golden image

The following steps were required to create a RHEL 7 golden image for deploying an Oracle RAC node.

Note

The information provided here is for configuring a two-node RAC environment. Additional RAC nodes may be configured by following the steps provided in HPE Reference Architecture for deploying Oracle 12c with HPE Synergy Image Streamer.

Note

An OS deployment plan must be in place to create an empty OS volume for step one below. For instructions on obtaining the deployment plan and more details on installing RHEL, see the document HPE Synergy Image Streamer – Capture RHEL 7.3

---

1. Create an HPE OneView server profile with an OS deployment plan that creates an empty OS volume of size 40 GB and assign the profile to a compute module.
2. Install RHEL 7.4 on the empty volume, adding ip=ibft as a kernel boot parameter.
3. Create /etc/resolv.conf with address of the DNS server containing all hostnames and IP addresses required for the RAC installation
4. Configure NTP.
5. Edit the file /etc/sysconfig/network and add the entry “NOZEROCONF=yes”.
6. Disable the avahi-daemon service.
7. Set the kernel parameters required for an Oracle database installation.
8. Install all RHEL packages required for an Oracle database installation.
9. Create the oracle and grid user accounts and groups.
10. Create the directory /u01/app on the root partition. This will be the location of the Grid Infrastructure and Oracle RAC installation.
11. Set the limits required by oracle and grid users in /etc/security/limits.d/oracle-limits.conf.
12. Obtain and install the packages for Oracle ASM libraries (oracleasmlib-2.0.12-1.el7.x86_64.rpm and oracleasm-support-2.1.8-3.el7.x86_64.rpm).
13. Configure the oracleasm service.
14. Install and configure sshpass from https://sourceforge.net/projects/sshpass. This is used by the sshkeydist.sh script to provide the passwords required when configuring ssh keys.
15. Unpack the zip files for the Grid Infrastructure kit in /u01/app/12.2.0/grid and the Oracle database kit in /home/oracle.
16. Install the cvuqdisk package from the rpm directory in the Grid Infrastructure kit.
17. Remove the grid files from /u01/app/12.2.0/grid, but leave the zip file in /home/grid (the plan scripts will unpack it on the first RAC node and then the RAC installation copies the files to the second node).
18. Create the response files grid-RAC.rsp in /home/grid and db-RAC.rsp in /home/oracle (see Appendix H).

Note

Some of the settings listed above will be modified each time that the golden image is used to deploy a new OS volume via Image Streamer. These settings include the hostname, domain, IP addresses, some kernel parameters, and some account limits.

After the OS has been customized according to the steps listed above, the Image Streamer "Create Golden Image" interface is used to create an image named "RHEL 7.4 Oracle RAC before install", as shown in Figure 8. Do the following to create the golden image:

1. Shut down the OS.
2. Find the OS volume number in the HPE OneView server profile created in step one above. It is listed under the OS Deployment section of the profile.
3. On the Image Streamer Golden Image screen, select "Create golden image" and specify a name ("RHEL 7.4 Oracle RAC before install"), description, OS volume, and Capture OS build plan. Note that the Capture OS build plan, "HPE – Foundation 1.0 – capture OS Volume as is" is chosen in this case.

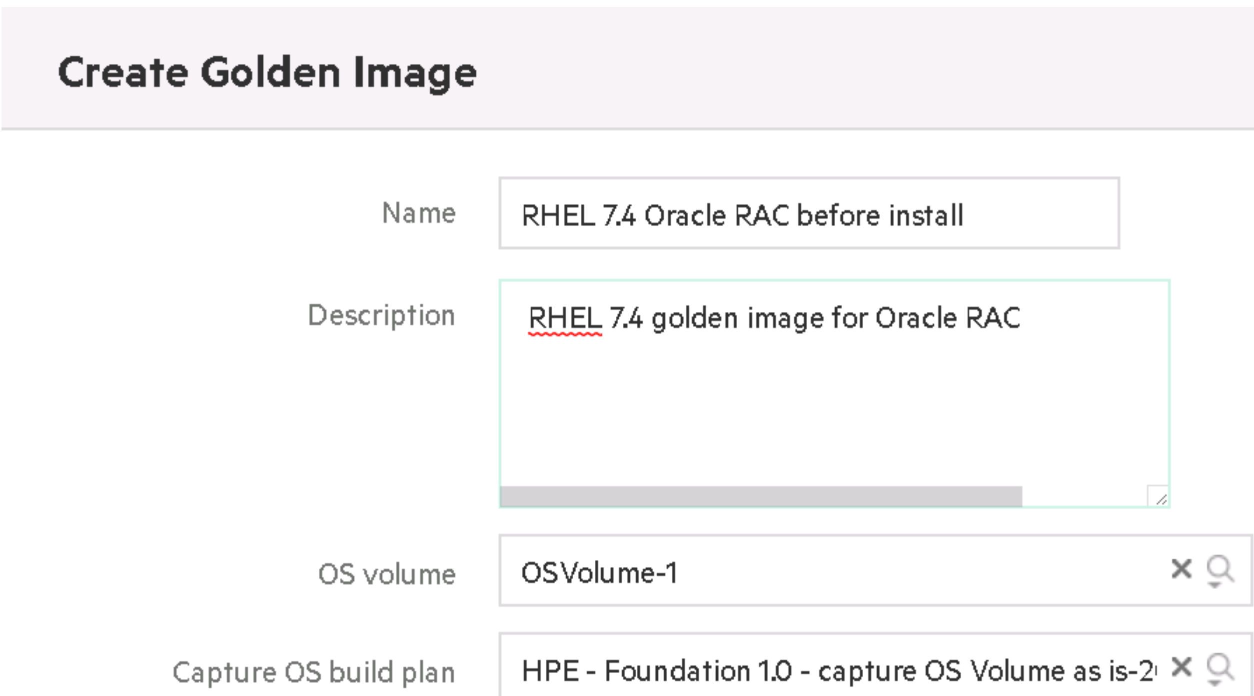

Figure 8. Create golden image

Steps 2 and 3: Create HPE Synergy Image Streamer plan scripts and OS build plan

Table 1 lists the HPE Synergy Image Streamer plan scripts that were developed to create an OS build plan named "RHEL 7 Deploy RAC Node". An Oracle RAC installation requires that ssh keys be exchanged between all servers in the cluster before the RAC software can be installed. Due to this requirement, two compute modules must be configured using the deployment plan created with this OS build plan. To prevent timing issues, one compute module must be powered on and finish booting prior to powering on the second compute module. The scripts to exchange the ssh keys and install Oracle RAC will be run from the second compute module when it is powered on for the first time. The contents of the plan scripts are included in Appendix I.

Note

All of the plan scripts are run with no interaction required by the user.

Table 1. Plan scripts for deploying Oracle RAC node

Plan script name	Purpose
RHEL7 – RAC mount	Mount root filesystem
RHEL7 – RAC hostname configuration	Set hostname and create /etc/hosts with all IPs for RAC environment
RHEL7 – RAC update resolv.conf	Set search domain in resolv.conf
RHEL7 – RAC public and private network configuration	Configure network devices for public network and RAC private network
RHEL7 - RAC update kernel params	Configure kernel parameters required for Oracle
RHEL7 – RAC update limits	Configure user limits for oracle account
RHEL7 – RAC create udev rules and multipath.conf	Create udev rules and multipath.conf for Oracle RAC volumes
RHEL7 – RAC create firstboot service	Create service to install Oracle RAC
RHEL7 - RAC create firstboot.sh script	Create shell script used by service
RHEL7 – RAC create sshkeydist.sh	Create script to configure passwordless login with ssh keys
RHEL7 – RAC create install_RAC.sh	Create script to install RAC database
RHEL7 – RACunmount	Unmount root file system
HPE - Foundation 1.0 – attempt or fail deployment – v1.00	Allow failing deployment for debugging purposes

Figure 9 shows the set of steps to create a golden image and customize it with Image Streamer plan scripts to configure an Oracle RAC environment. The components needed in the golden image are shown in the green boxes. The steps automated by the Image Streamer plan scripts to customize the golden image are in the blue boxes. Finally, the orange boxes list the scripts that run automatically at first boot to configure Oracle RAC.

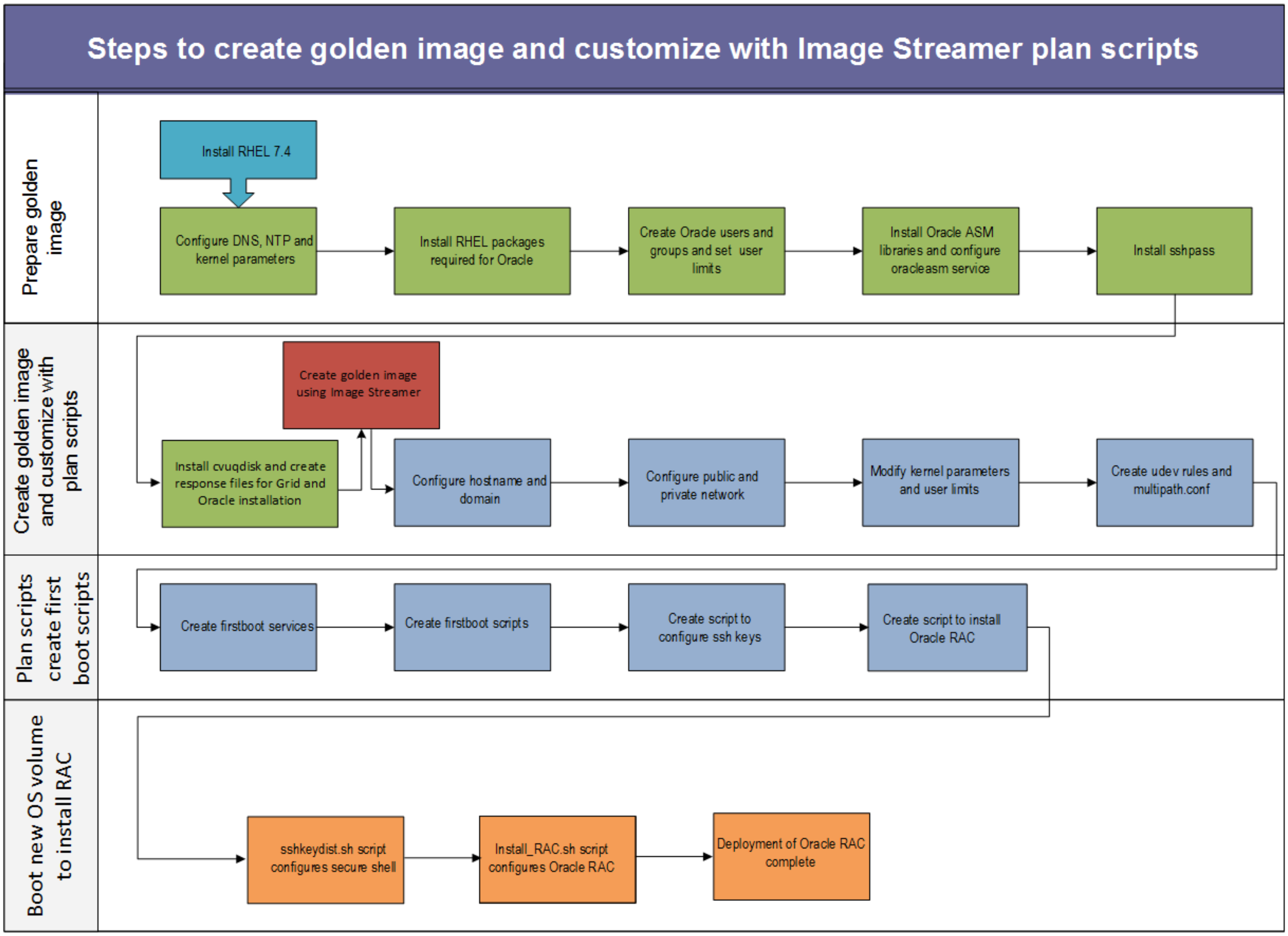

Figure 9. Steps to create golden image and customize with Image Streamer plan scripts

Step 4: Create HPE Synergy Image Streamer deployment plan

An Image Streamer deployment plan is created by specifying an OS build plan and a golden image, using the "Create Deployment Plan" dialog box shown in Figure 10. Note that the name of the OS build plan has not yet been filled in, because once it is entered, the list of custom attributes will be displayed, making it difficult to view the entire dialog box. The list of custom attributes and their default settings for the deployment plan is included in Appendix J.

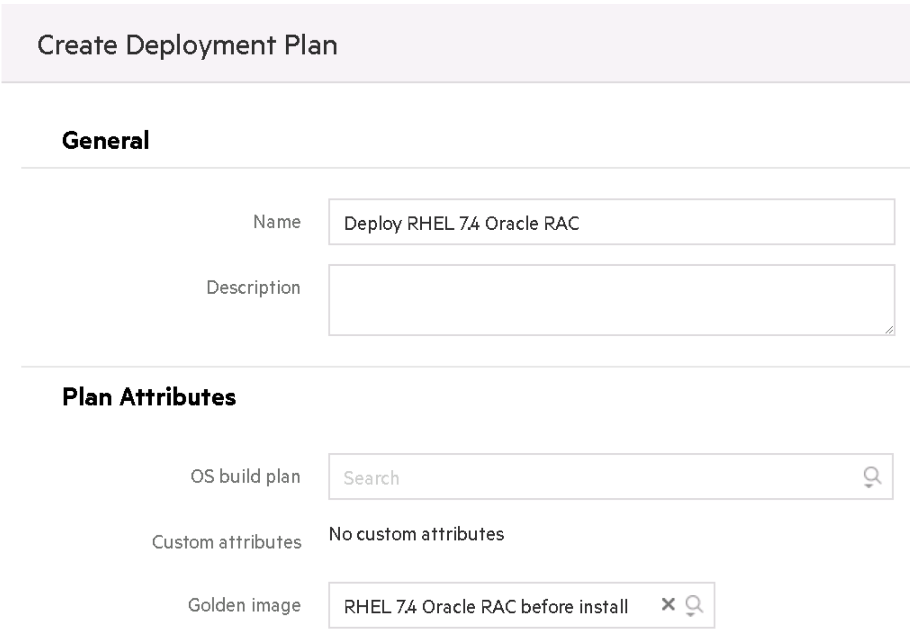

Figure 10. Create deployment plan

Deploy a compute module using HPE Synergy Image Streamer deployment plan

This section describes the steps required for deploying an HPE Synergy compute module with an Image Streamer deployment plan. Prior to deploying a compute module, the HPE Synergy environment must be configured with the networks and storage volumes required for an Oracle RAC implementation.

Step 1: Create required networks

The following networks are required in the HPE Synergy environment, for usage in the server profiles that deploy Oracle RAC:

- Public network
- Private network for RAC interconnect traffic (heartbeat)
- Image Streamer deployment network

Table 2 summarizes the configuration for each network. All networks were created with a preferred bandwidth of 2.5 Gb/second and a maximum bandwidth of 20 Gb/second. Note that the private network for the RAC heartbeat is not assigned a VLAN or uplink set, since communications are only needed between the HPE Synergy compute modules.

Table 2. Networks for Oracle RAC deployments

Description	Network name	Type	VLAN	Subnet ID	Uplink set
Public network	Network	Ethernet	210	10.210.40.0	Network
Private network for RAC	RAC	Ethernet	None	192.168.0.0	None
Image Streamer deployment network	I3S_Deployment	Ethernet	100	10.100.0.0	I3S Deployment

Figure 11 shows the Connections section of the server profile for an Oracle RAC node.

Note

The Image Streamer deployment network is automatically added to the server profile when an OS deployment plan is selected, and it must be on ports 3:1a and 3:2a. The iSCSI boot configuration is also automatically added to the profile.

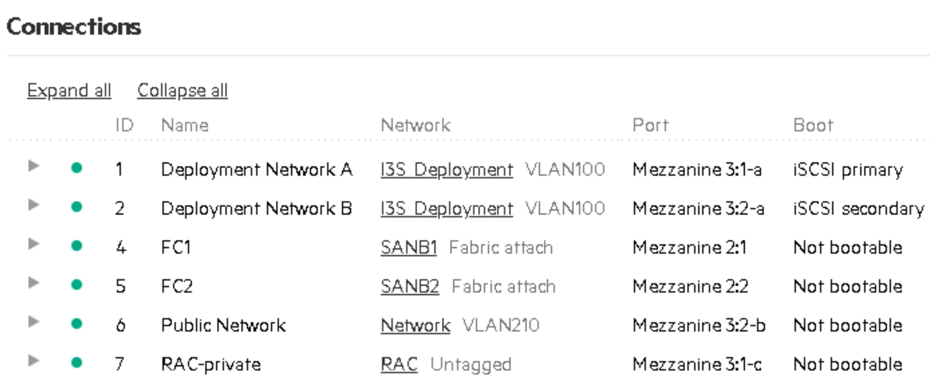

Figure 11. Network and SAN connections in server profile

Step 2: Storage configuration

The HPE OneView Create Volume dialog box was used to configure 3PAR storage for the Oracle data, logs and RAC voting disks. For the RAC configuration, all volumes were created with the sharing attribute set to Shared, as shown in Figure 12.

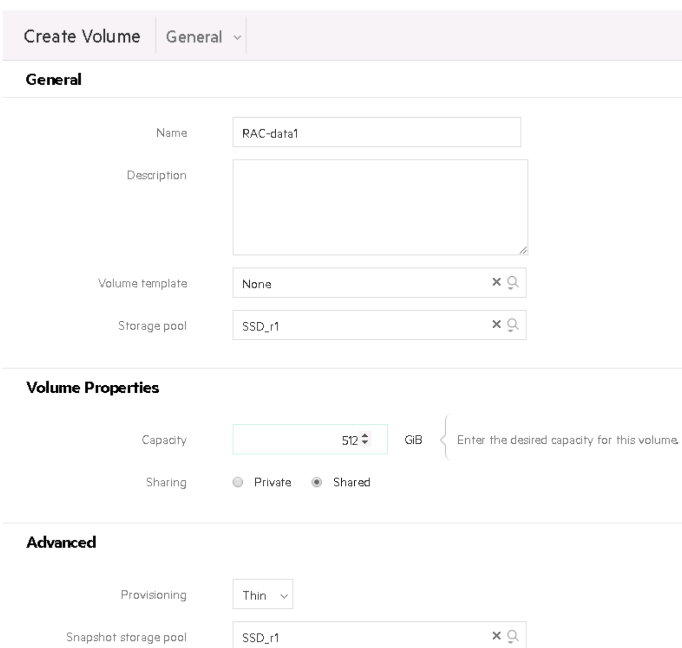

Figure 12. Create shared volume for Oracle RAC nodes

When creating the server profile for the new RAC node, all that was needed was to add the existing volumes to the server profile. Figure 13 shows the SAN Storage portion of the server profile for the new RAC node.

SAN Storage Expand all Collapse all

Name	LUN	Pool	Size	Sharing	Enabled Paths	Boot
RAC-data1	7	SSD_r1	512.00 GiB	Shared	2	n/a
RAC-data2	24	SSD_r1	512.00 GiB	Shared	2	n/a
RAC-data3	4	SSD_r1	512.00 GiB	Shared	2	n/a
RAC-data4	26	SSD_r1	512.00 GiB	Shared	2	n/a
RAC-data5	6	SSD_r1	512.00 GiB	Shared	2	n/a
RAC-data6	12	SSD_r1	512.00 GiB	Shared	2	n/a
RAC-data7	3	SSD_r1	512.00 GiB	Shared	2	n/a
RAC-data8	5	SSD_r1	512.00 GiB	Shared	2	n/a
RAC-redo1a	23	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo2a	19	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo3a	13	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo4a	20	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo5a	21	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo6a	18	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo7a	2	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo8a	10	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo1b	11	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo2b	16	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo3b	25	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo4b	9	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo5b	1	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo6b	8	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo7b	17	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-redo8b	14	SSD_r5	128.00 GiB	Shared	2	n/a
RAC-OCR	22	SSD_r5	16.00 GiB	Shared	2	n/a
RAC-voting	15	SSD_r5	16.00 GiB	Shared	2	n/a
RAC-GIMR	27	SSD_r5	40.00 GiB	Shared	2	n/a

Figure 13. SAN storage connections in server profile

Step 3: Create a server profile

After the required networks and storage volumes have been configured, a server profile can be created that utilizes these components along with an OS deployment plan that will configure a node for an Oracle RAC cluster.

Note

Deploying a RAC node requires the presence of a DNS server for the environment that includes the hostnames and all IP addresses required for all of the nodes in the cluster.

Figure 14 shows the Create Server Profile dialog box, with the OS Deployment section specifying the OS deployment plan "Deploy RHEL 7.4 Oracle RAC". At this point, the user may also modify the deployment settings listed for the deployment plan. This includes settings such as hostnames, IP addresses, kernel parameters and passwords. For Oracle RAC nodes, it is necessary to provide IP addresses for the public network, the private network for RAC heartbeat, the VIP addresses, and the scan addresses. The full set of attributes that can be customized for this deployment plan are listed in Appendix J. When the server profile is created, the plan scripts specified in the OS build plan are used to customize the new OS volume. After the profile creation has completed, the compute module can be powered on, and the steps required to install Oracle RAC are run by the firstboot services.

Note

After the creation of a server profile for the first compute module in the RAC cluster, power the compute module on and wait for the OS to start up before powering on the second compute module for the cluster. The first compute module must have network connectivity before the firstboot services and scripts run on the second compute module to configure the ssh keys and install Oracle RAC for both of the nodes in the cluster. The second compute module to be booted will be the primary cluster node, as the RAC installation will run from that system.

Note

The creation of a server profile also allows the user to specify firmware updates, boot settings, and BIOS settings to be applied to the server. These steps are not shown here because the focus is on OS deployment.

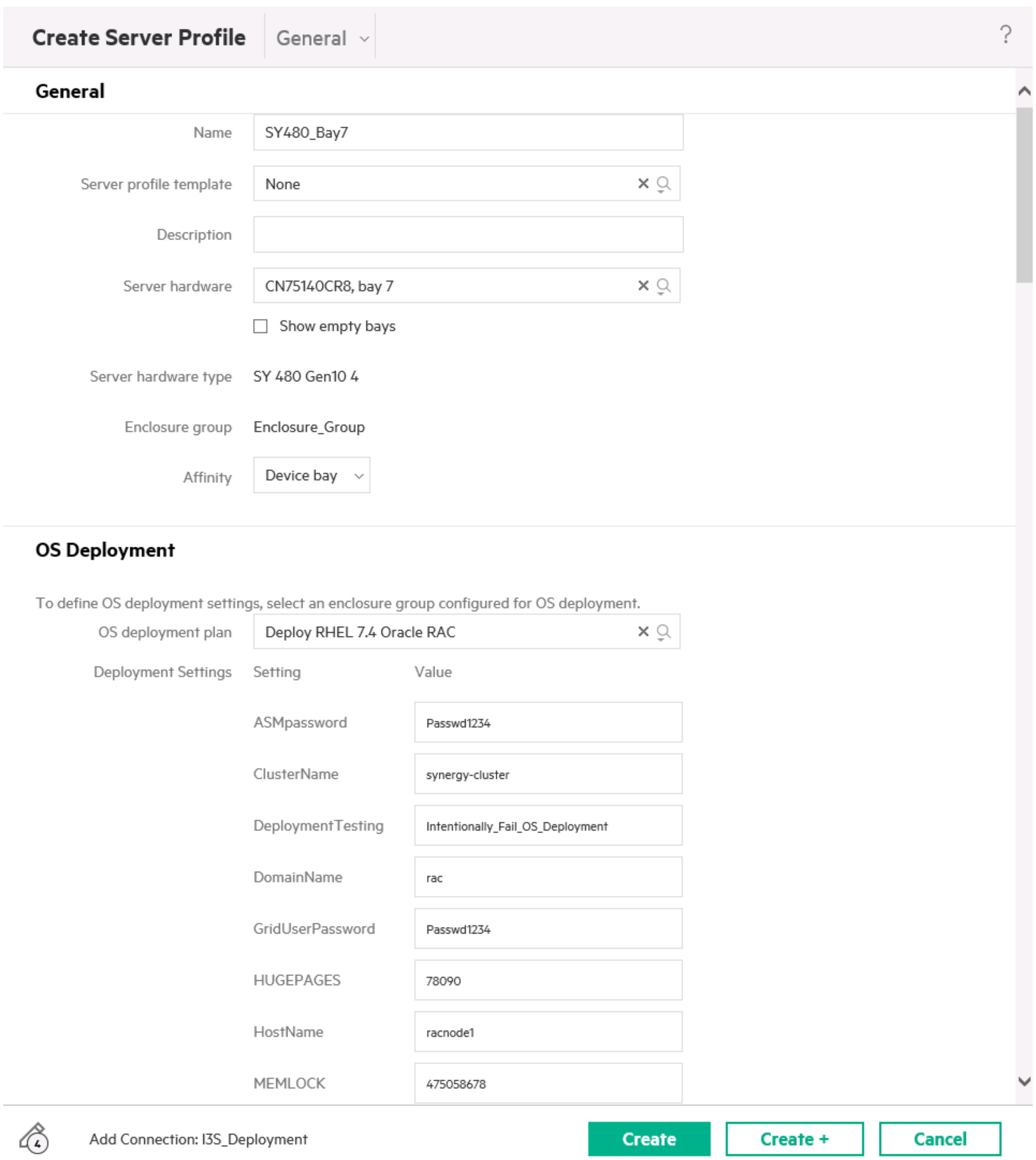

Figure 14. Create server profile with deployment plan

Capacity and sizing

Two types of performance metrics were gathered for the this Reference Architecture: (1) the time required to deploy an HPE Synergy compute module with a server profile and deployment plan, and (2) the scalability of Oracle single instance database and Oracle RAC nodes in an HPE Synergy environment.

Deploying a compute module with HPE OneView server profile and HPE Synergy Image Streamer deployment plan

Using HPE OneView server profiles with HPE Synergy Image Streamer deployment plans allows users to very quickly deploy a server for an Oracle RAC environment. Figure 15 shows the Activity section of a server profile for deploying a new RAC node. The total time to complete the deployment was 2 minutes and 4 seconds. The time required to customize the OS volume (i.e., apply the plan scripts) was a mere 19 seconds. Applying the profile to the server, including the boot mode settings and server settings took 1 minute and 26 seconds.

Note

The creation of a server profile also allows the user to specify firmware updates, boot settings, and BIOS settings to be applied to the compute module. These steps can add a significant and highly variable amount of time to the deployment. They are not included here because the focus is on OS deployment using Image Streamer. Firmware updates may be conducted when the compute module is added to the environment, avoiding the need to do this at OS deployment time. The iSCSI boot configuration for the HPE Synergy Image Streamer OS volume is set automatically in the server profile and is included in the timings shown here.

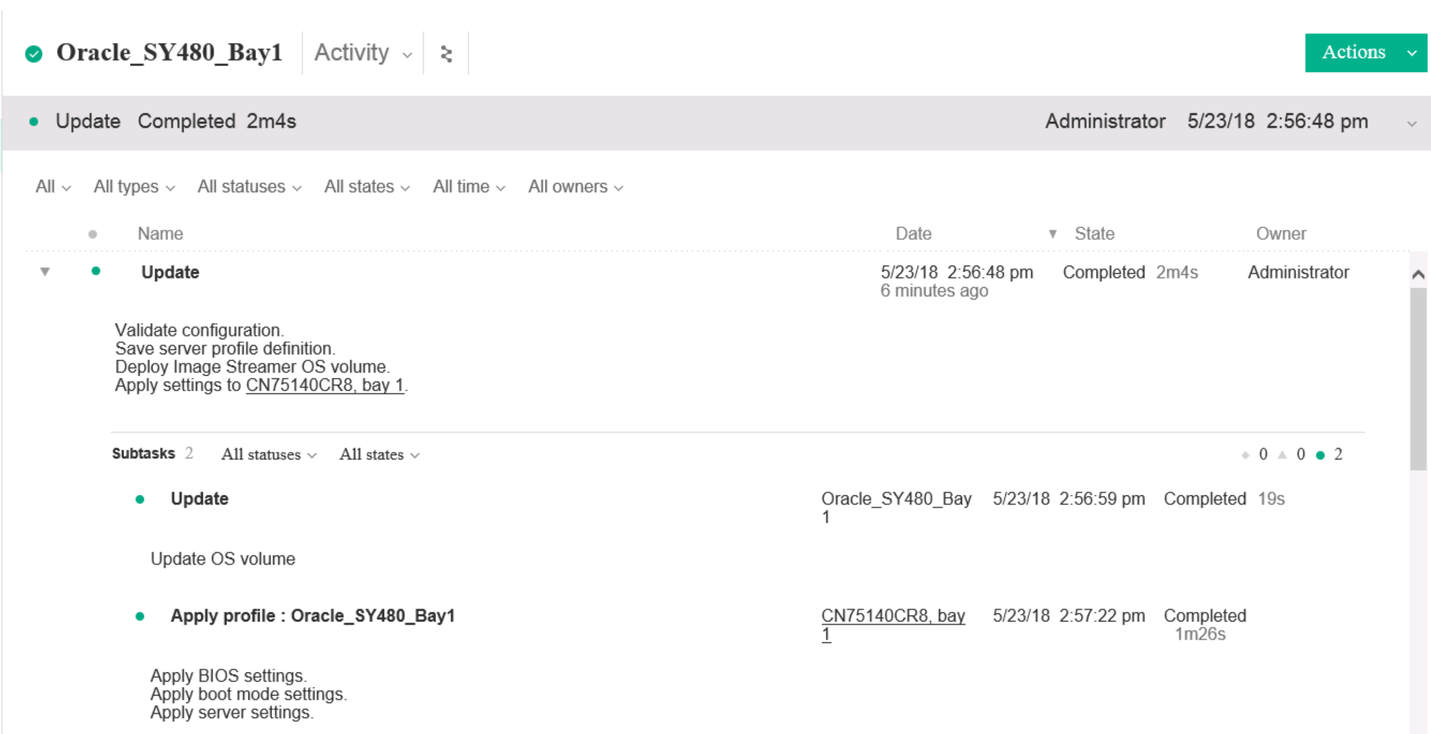

Figure 15. Activity log for deploying Oracle RAC node

Workload description for Oracle RAC scalability tests

The Oracle workload was tested using HammerDB, an open-source tool. The tool implements an OLTP-type workload (60 percent read and 40 percent write) with small I/O sizes of a random nature. The transaction results have been normalized and are used to compare test configurations. Other metrics measured during the workload come from the operating system and/or standard Oracle Automatic Workload Repository (AWR) statistics reports.

The OLTP test, performed on a 1TB database, was both moderately CPU and highly I/O intensive. The environment was tuned for maximum user transactions. After the database was tuned, the transactions per minute (TPM) were recorded for a varying number of Oracle connections. Because customer workloads vary so much in characteristics, the measurement was made with a focus on maximum transactions.

Oracle Enterprise Database version 12.2.0.1 was used in this test configuration.

Analysis and recommendations for Oracle RAC scalability

Testing was conducted with both an Oracle single-instance database, as well as Oracle RAC, in order to compare the performance and scalability of RAC to a single instance. One to four RAC nodes were tested. All of these runs were on HPE Synergy 480 Gen10 Compute Modules with identical hardware configurations. In addition, single instance performance was measured on an HPE Synergy 660 Gen10 Compute Module to demonstrate the additional throughput that can be achieved when more compute capacity is required.

Figure 16 shows relative throughput (measured in transactions per minute) for Oracle RAC with one to four nodes as compared to a single-instance Oracle database. All results are normalized to the single-instance data for each Oracle connection count. Note that RAC One Node provided slightly lower throughput than the single-instance database, due to the overhead required for implementing RAC. When adding RAC nodes, throughput increased linearly, with two nodes providing up to 1.78 times, three nodes offering up to 2.34 times, and four nodes achieving up to 2.91 times the number of transactions as the single-instance database.

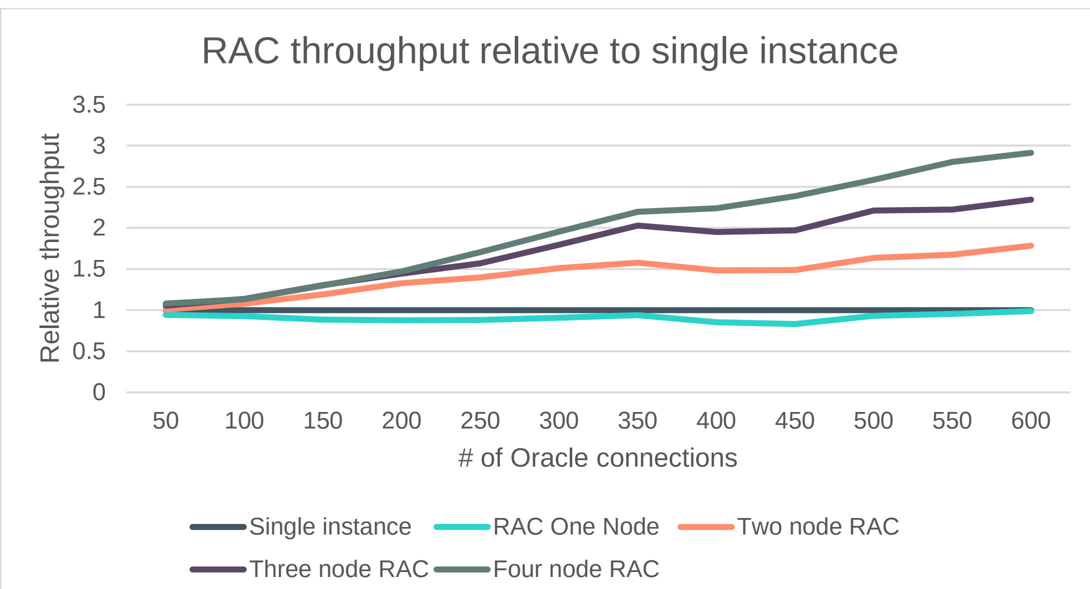

Figure 16. Oracle RAC throughput relative to Oracle single-instance database

Figure 17 shows relative throughput for two, three and four nodes as compared to One Node RAC. All results are normalized to the One Node data for each connection count. Throughput scaled linearly as RAC nodes were added, with two nodes providing up to 1.81 times, three nodes offering up to 2.37 times, and four nodes achieving up to 2.95 times the throughput of One Node RAC.

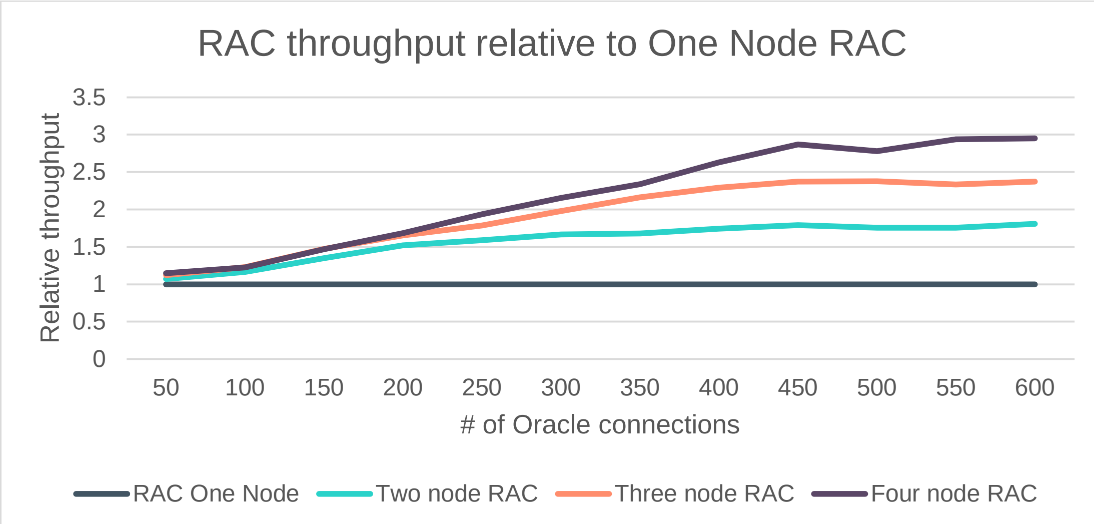

Figure 17. Oracle RAC throughput relative to One Node RAC

Figure 18 shows that CPU utilization for each configuration as the number of Oracle connections increased. With a single instance database, utilization reached 89%. As RAC nodes were added, the utilization decreased because the load is shared across multiple compute modules and more time is spent coordinating the activities between the nodes in the cluster.

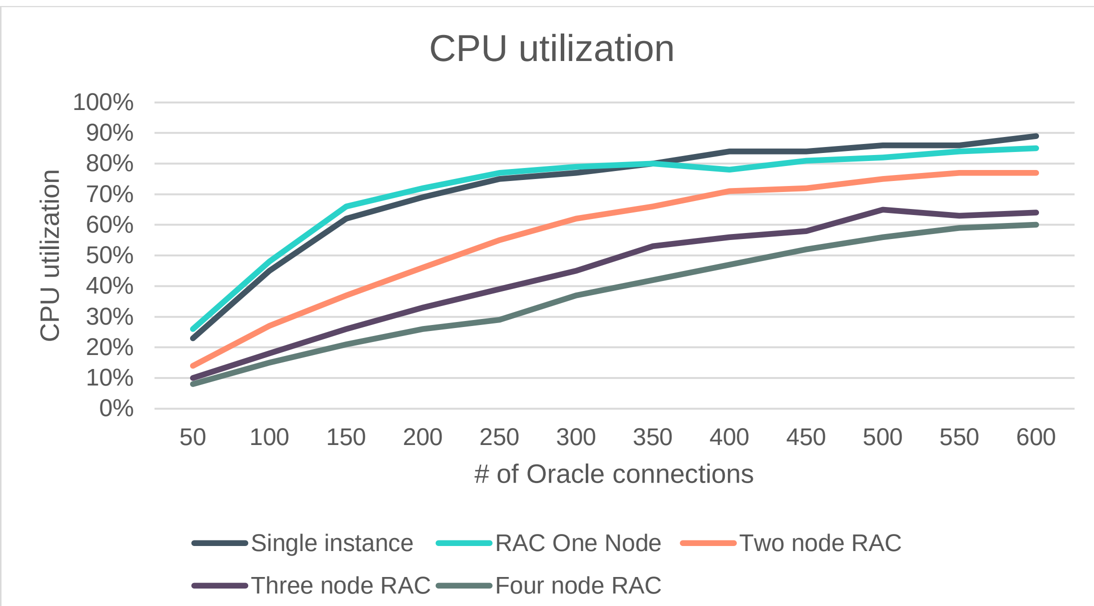

Figure 18. CPU utilization for all configurations

Figure 19 shows the transaction latency for each configuration at each connection count. Note that the latency decreases as nodes are added, reflecting the ability of the solution to scale

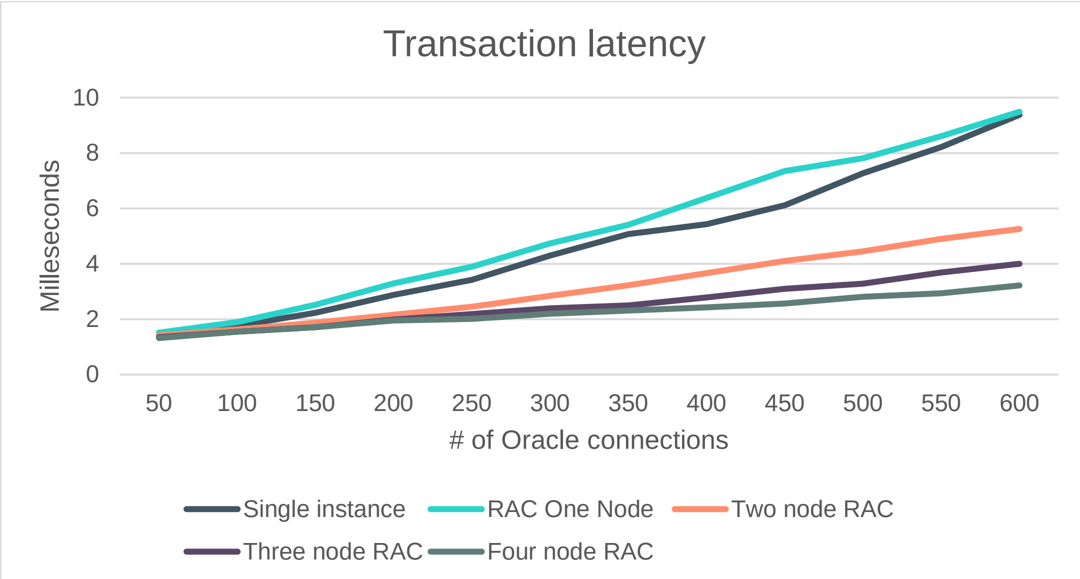

Figure 19. Transaction latencies for all configurations at various connection counts

One of the benefits of the HPE Synergy solution is the ability to easily scale up to a server with more processing power when more capacity is required. The final test demonstrates how Oracle throughput can be increased by moving from a dual-socket HPE Synergy 480 Gen10 to a quad-socket HPE Synergy 660 Gen10 when the CPUs on the HPE Synergy 480 are fully utilized and more capacity is required. For this test, the number of cores per processor in the HPE Synergy 480 and the HPE Synergy 660 was reduced to four, in order to demonstrate a scenario where processing power was the bottleneck. Figure 20 shows that Oracle throughput can be increased by a factor of 1.9 by moving the workload from the HPE Synergy 480 to the HPE Synergy 660 Compute Module.

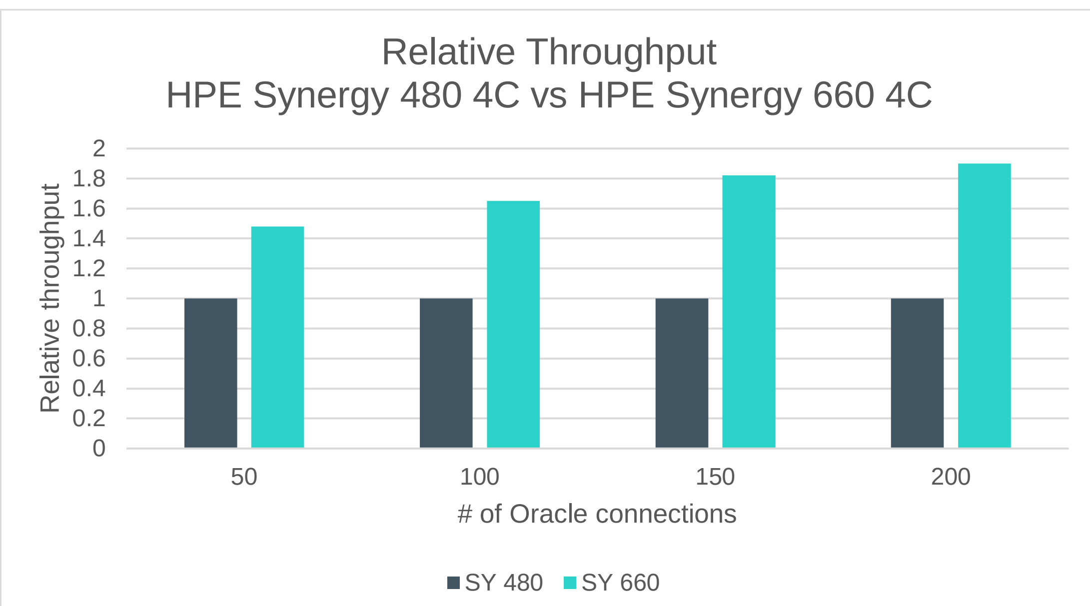

Figure 20. Relative throughput for HPE Synergy 480 Gen10 compared to HPE Synergy 660 Gen10

Summary

This Reference Architecture demonstrates how HPE Synergy enables Oracle database administrators to accelerate Oracle RAC 12c deployment and easily update their environments. HPE Synergy Composer and HPE Synergy Image Streamer can be utilized to create deployment plans to install and configure Oracle RAC clusters. The fluid resource pools and software-defined intelligence of HPE Synergy allow administrators to rapidly compose any configuration required, reducing deployment time from hours or days down to minutes. More specifically, this Reference Architecture shows the following benefits of utilizing HPE Synergy for Oracle RAC solutions.

- HPE Synergy Composer with embedded HPE OneView seamlessly manages the entire environment, including configuration of network resources required for Oracle RAC on the compute modules, creation and management of HPE 3PAR StoreServ volumes for the Oracle RAC nodes, and deploying the OS and application software on the compute modules.
- Testing shows that HPE Synergy Composer plus HPE Synergy Image Streamer allows administrators to configure a new system for Oracle in less than three minutes, which is a significant reduction as compared to traditional deployment times of hours or days.
- Oracle RAC scales linearly as HPE Synergy compute modules are added to the cluster.
- If HPE Synergy 480 Gen10 Compute Modules run out of processing capacity, the environment can easily be scaled up to HPE Synergy 660 Gen10 Compute Modules to double Oracle throughput. This is much easier to configure in an HPE Synergy environment due to the HPE OneView server profiles and ease of moving resources from one compute module to another.

This Reference Architecture describes solution testing performed in April and May 2018.

Implementing a proof-of-concept

As a matter of best practice for all deployments, HPE recommends implementing a proof-of-concept using a test environment that matches as closely as possible the planned production environment. In this way, appropriate performance and scalability characterizations can be obtained. For help with a proof-of-concept, contact an HPE Services representative (hpe.com/us/en/services/consulting.html) or your HPE partner.

Appendix A: Bill of materials

Note

Part numbers are at time of testing and subject to change. The bill of materials does not include complete support options or other rack and power requirements. If you have questions regarding ordering, please consult with your HPE Reseller or HPE Sales Representative for more details. hpe.com/us/en/services/consulting.html

Table 3. Bill of materials for HPE Synergy solution with HPE 3PAR StoreServ 9450 all flash array

Qty	Part Number	Description
		Rack and Network Infrastructure
1	BW908A	HPE 642 1200mm Shock Intelligent Rack
1	BW909A	HPE 42U 1200mm Side Panel Kit
1	BW930A	HPE Air Flow Optimization Kit
3	BW928A	HPE 10pk 1U Universal Filler Panel
		HPE Synergy Frame Components
3	797740-B21	HPE Synergy 12000 Configure-to-order Frame with 1x Frame Link Module 10x Fans
3	804942-B21	HPE Synergy Frame Link Module
3	798096-B21	HPE 6X 2650W AC Titanium Hot Plug FIO Power Supply Kit
2	804353-B21	HPE Synergy Composer
2	804937-B21	HPE Synergy Image Streamer
2	794502-B21	HPE Virtual Connect SE 40Gb F8 Module for HPE Synergy
2	779227-B21	HPE Virtual Connect SE 16Gb Fibre Channel Module for Synergy

Qty	Part Number	Description
4	779218-B21	HPE Synergy 20Gb Interconnect Link Module
		HPE Synergy Compute Module Components
4	871942-B21	HPE Synergy 480 Gen10 Configure-to-order Premium Compute Module
4	872119-L21	HPE Synergy 480/660 Gen10 Intel Xeon-Platinum 8180 FIO Processor Kit
4	872119-B21	HPE Synergy 480/660 Gen10 Intel Xeon-Platinum 8180 Processor Kit
64	815100-B21	HPE 32GB (1x16GB) Dual Rank x4 DDR4-2666 RDIMMs
8	777264-B21	HPE 340GB SATA RI M.2 2280 SSD
4	873165-B21	HPE SY 480 Gen10 M.2 FIO Adptr Brd Kit
4	777430-B21	HPE Synergy 3820C 10/20Gb Converged Network Adapter
8	777452-B21	HPE Synergy 3830C 16G FC HBA
1	871931-B21	HPE Synergy 660 Gen10 Configure-to-order Premium Backplane Compute Module
2	872119-L21	HPE Synergy 480/660 Gen10 Intel Xeon-Platinum 8180 FIO Processor Kit
2	872119-B21	HPE Synergy 480/660 Gen10 Intel® Xeon-Platinum 8180 Processor Kit
16	815100-B21	HPE 32GB (1x16GB) Dual Rank x4 DDR4-2666 RDIMMs
1	777430-B21	HPE Synergy 3820C 10/20Gb Converged Network Adapter
4	777452-B21	HPE Synergy 3830C 16G FC HBA
		Storage
1	BW904A	HPE 42U 600X1075mm Enterprise Shock Rack
1	Q0E92A	HPE 3PAR StoreServ 9450 2N+SW Storage Base
2	Q7F41A	HPE 3PAR 9450+SW Storage Node
4	Q0E96A	HPE 3PAR 9000 4pt 12Gb SAS HBA
8	Q0E97A	HPE 3PAR 9000 4-pt 16Gb FC Adapter
80	Q0F40A	HPE 3PAR 9000 400GB+SW SFF SSD
2	QK753B	HPE SN6000B 16Gb 48/24 FC Switch
48	QK724A	HPE B-series 16Gb SFP+SW XCVR
48	QK735A	HPE Premier Flex LC/LC OM4 2f 15m Cbl
8	QK734A	HPE Premier Flex LC/LC OC4 2f 5m Cbl
1	P9M30A	HPE 3PAR Direct Connect Cabling Option
1	Q1H95A	HPE 3PAR 1U Rack Accessories Kit
16	716197-B21	HPE Ext 2.0m MiniSAS HD to MiniSAS HD Cbl
8	Q0E95A	HPE 3PAR 9000 2U SFF SAS Drive Encl
1	Q0F86A	HPE 3PAR StoreServ RPS Service Processor
1	TK808A	HPE Rack Front Door Cover Kit
4	P9Q39A	HPE G2 Basic Mdlr 4.9kVA/C19 NA/JP PDU
1	BW932A	HPE 600mm Rack Stabilizer Kit
1	BW906A	HPE 42U 1075mm Side Panel Kit
1	L7F19A	HPE 3PAR All-inclusive Single-system Latest Media

Appendix B: Linux kernel settings

The RHEL kernel parameters that were set in the /etc/sysctl.conf file in the golden image are listed below. The plan script “RHEL7 – RAC update kernel params” edits this file to modify the settings for kernel.shmmax and vm.nr_hugepages, which are dependent upon the memory configuration of a server and the desired Oracle SGA setting.

```
kernel.sem = 250 32000 100 128
kernel.shmall = 4294967295
kernel.shmmax = 270255800320
fs.file-max = 6815744
kernel.shmmni = 4096
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048586
vm.nr_hugepages = 78090
vm.hugetlb_shm_group = 507
kernel.numa_balancing = 0
net.ipv4.tcp_rmem = 4096 87380 134217728
net.ipv4.tcp_wmem = 4096 65536 134217728
net.core.netdev_max_backlog = 300000
```

Appendix C: User account limits

The /etc/security/limits.d/oracle-limits.conf file included the following settings for the oracle and grid user accounts. This file is edited by the plan script “RHEL7 – RAC update limits” to modify the setting for memlock, which is dependent upon the memory configuration of a server:

```
oracle soft   nofile    1024
oracle   hard   nofile    65536
oracle   soft   nproc    16384
oracle   hard   nproc    16384
oracle   soft   stack    10240
oracle   hard   stack    32768
# set memlock to 90% of memory
oracle   hard   memlock   475058678
oracle   soft   memlock   475058678
grid     soft   nofile    1024
grid     hard   nofile    65536
grid     soft   nproc    16384
grid     hard   nproc    16384
grid     soft   stack    10240
grid     hard   stack    32768
# set memlock to 90% of memory
grid    hard   memlock   475058678
grid    soft   memlock   475058678
```

Appendix D: Oracle initialization parameters

The Oracle initialization parameters used for the four-node RAC tests are listed below.

```
ora4801.__data_transfer_cache_size=0
ora4804.__data_transfer_cache_size=0
ora4802.__data_transfer_cache_size=0
ora4803.__data_transfer_cache_size=0
ora4801.__db_cache_size=138512695296
ora4804.__db_cache_size=138512695296
ora4802.__db_cache_size=138512695296
ora4803.__db_cache_size=138512695296
ora4801.__inmemory_ext_roarea=0
ora4804.__inmemory_ext_roarea=0
ora4802.__inmemory_ext_roarea=0
ora4803.__inmemory_ext_roarea=0
ora4801.__inmemory_ext_rwarea=0
ora4804.__inmemory_ext_rwarea=0
ora4802.__inmemory_ext_rwarea=0
ora4803.__inmemory_ext_rwarea=0
ora4801.__java_pool_size=3758096384
ora4804.__java_pool_size=3758096384
ora4802.__java_pool_size=3758096384
ora4803.__java_pool_size=3758096384
ora4801.__large_pool_size=536870912
ora4804.__large_pool_size=536870912
ora4802.__large_pool_size=536870912
ora4803.__large_pool_size=536870912
ora4801.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
ora4802.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
ora4803.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
ora4804.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
ora4801.__pga_aggregate_target=54223962112
ora4804.__pga_aggregate_target=54223962112
ora4802.__pga_aggregate_target=54223962112
ora4803.__pga_aggregate_target=54223962112
ora4801.__sga_target=162671886336
ora4804.__sga_target=162671886336
ora4802.__sga_target=162671886336
ora4803.__sga_target=162671886336
ora4801.__shared_io_pool_size=536870912
ora4804.__shared_io_pool_size=536870912
ora4802.__shared_io_pool_size=536870912
ora4803.__shared_io_pool_size=536870912
ora4801.__shared_pool_size=18253611008
ora4804.__shared_pool_size=18253611008
ora4802.__shared_pool_size=18253611008
ora4803.__shared_pool_size=18253611008
ora4801.__streams_pool_size=0
ora4804.__streams_pool_size=0
ora4802.__streams_pool_size=0
ora4803.__streams_pool_size=0
*._enable_NUMA_support=true
*._fast_cursor_reexecute=true
```

```
*._high_priority_processes='VKTM*|LG*'
*.audit_file_dest='/u01/app/oracle/admin/ora480/adump'
*.audit_trail='db'
*.cluster_database=true
*.compatible='12.2.0'
*.control_files='+DATA/ORA480/CONTROLFILE/current.261.973968531'
*.db_block_size=8192
*.db_create_file_dest='+DATA'
*.db_name='ora480'
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=ora480XDB)'
family:dw_helper.instance_mode='read-only'
ora4802.instance_number=2
ora4801.instance_number=1
ora4803.instance_number=3
ora4804.instance_number=4
*.local_listener='-oraagent-dummy-'
*.lock_sga=TRUE
*.nls_language='AMERICAN'
*.nls_territory='AMERICA'
*.open_cursors=3000
*.pga_aggregate_target=51560m
*.processes=3000
*.remote_login_passwordfile='exclusive'
*.sga_target=154680m
ora4802.thread=2
ora4801.thread=1
ora4803.thread=3
ora4804.thread=4
ora4802.undo_tablespace='UNDOTBS2'
ora4801.undo_tablespace='UNDOTBS1'
*.undo_tablespace='UNDOTBS2'
ora4803.undo_tablespace='UNDOTBS3'
ora4804.undo_tablespace='UNDOTBS4'
*.use_large_pages='ONLY'
```

The Oracle initialization parameters used for the single instance tests are listed below.

```
*.audit_file_dest='/u01/app/oracle/admin/or480/adump'
*.audit_trail='db'
*.compatible='12.2.0'
*.control_files='+DATA/OR480/CONTROLFILE/current.261.974742693'
*.db_block_size=8192
*.db_create_file_dest='+DATA'
*.db_name='or480'
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=or480XDB)'
*.local_listener='LISTENER_OR480'
*.nls_language='AMERICAN'
*.nls_territory='AMERICA'
*.open_cursors=3000
*.pga_aggregate_target=51547m
*.processes=3000
*.remote_login_passwordfile='EXCLUSIVE'
```

```
*.sga_target=155136m
*.undo_tablespace='UNDOTBS1'
_high_priority_processes='VKTM*|LG*'
lock_sga=TRUE
use_large_pages='ONLY'
result_cache_max_size=794304K
_enable_NUMA_support=true
_fast_cursor_reexecute=true
```

Appendix E: HPE-ATX configuration script

The following script was used to start the Oracle listener processes under ATX. The Round Robin policy was used to evenly distribute the processes across the NUMA nodes in the system.

```
#!/bin/bash
lsnrctl stop
hpe-atx -p rr_flat -l listener.log lsnrctl start
```

Appendix F: Multipath configuration

RHEL 7.4 has the required 3PAR multipath settings built in, so the /etc/multipath.conf file created by the Image Streamer plan script was just used to change the polling interval from 5 (default setting) to 10 seconds, and to configure aliases for each of the 3PAR volumes for Oracle RAC.

```
## Use user friendly names, instead of using WWIDs as names.
defaults {
       polling_interval 10
       user_friendly_names yes
       find_multipaths yes
multipaths {
        multipath {
                wwid 360002ac0000000000000690001f77c
                alias data1
        }
        multipath {
                wwid 360002ac00000000000006a0001f77c
                alias data2
        }
        multipath {
                wwid 360002ac00000000000006b0001f77c
                alias data3
        }
        multipath {
                wwid 360002ac00000000000006c0001f77c
                alias data4
        }
        multipath {
                wwid 360002ac00000000000006d0001f77c
                alias data5
        }
        multipath {
                wwid 360002ac00000000000006e0001f77c
                alias data6
        }
```

```
multipath {
        wwid 360002ac0000000000000006f0001f77c
        alias data7
}
multipath {
        wwid 360002ac000000000000000700001f77c
        alias data8
}
multipath {
        wwid 360002ac000000000000000710001f77c
        alias redo1a
}
multipath {
        wwid 360002ac000000000000000720001f77c
        alias redo2a
}
multipath {
        wwid 360002ac000000000000000730001f77c
        alias redo3a
}
multipath {
        wwid 360002ac000000000000000740001f77c
        alias redo4a
}
multipath {
        wwid 360002ac000000000000000750001f77c
        alias redo5a
}
multipath {
        wwid 360002ac000000000000000760001f77c
        alias redo6a
}
multipath {
        wwid 360002ac000000000000000770001f77c
        alias redo7a
}
multipath {
        wwid 360002ac000000000000000780001f77c
        alias redo8a
}
multipath {
        wwid 360002ac000000000000000790001f77c
        alias redo1b
}
multipath {
        wwid 360002ac0000000000000007a0001f77c
        alias redo2b
}
multipath {
        wwid 360002ac0000000000000007b0001f77c
        alias redo3b
}
```

```
        multipath {
                wwid 360002ac0000000000000007c0001f77c
                alias redo4b
        }
        multipath {
                wwid 360002ac0000000000000007d0001f77c
                alias redo5b
        }
        multipath {
                wwid 360002ac0000000000000007e0001f77c
                alias redo6b
        }
        multipath {
                wwid 360002ac0000000000000007f0001f77c
                alias redo7b
        }
        multipath {
                wwid 360002ac000000000000000800001f77c
                alias redo8b
        }
        multipath {
                wwid 360002ac000000000000000450001f77c
                alias voting
        }
        multipath {
                wwid 360002ac000000000000000460001f77c
                alias fra
        }
        multipath {
                wwid 360002ac000000000000000670001f77c
                alias gimr
        }
}
```

Appendix G: udev rules files

A udev rules file /etc/udev/rules.d/10-3par.rules was created for the golden image to set attributes for the 3PAR volumes used for the Oracle database:

```
ACTION=="add|change", KERNEL=="dm-*", PROGRAM="/bin/bash -c 'cat
/sys/block/$name/slaves/*/device/vendor | grep 3PARdata'", ATTR{queue/rotational}="0",
ATTR{queue/scheduler}="noop", ATTR{queue/rq_affinity}="2", ATTR{queue/nomerges}="1",
ATTR{queue/nr_requests}="128"
```

A udev rules file /etc/udev/rules.d/12-dm-permission.rules was created by the Image Streamer plan script to set the required ownership of the Oracle LUNs:

```
ENV{DM_NAME}=="data1", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="data2", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="data3", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="data4", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="data5", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="data6", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="data7", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
```

```
ENV{DM_NAME}=="data8", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo1a", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo2a", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo3a", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo4a", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo5a", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo6a", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo7a", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo8a", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo1b", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo2b", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo3b", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo4b", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo5b", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo6b", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo7b", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="redo8b", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="voting", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="fra", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
ENV{DM_NAME}=="gimr", OWNER:="grid", GROUP:="asmadmin", MODE:="660"
```

Appendix H: Oracle response files

The Oracle Grid Infrastructure and Database products each provide sample response files for silent installations of the products. These files were modified as follows and included in the "RHEL 7.4 Oracle RAC before install" golden image used by the "RHEL 7.4 Oracle RAC" deployment plan.

The file /home/grid/grid-RAC.rsp included the following entries. The scanName, clusterName, clusterNodes, and password settings are modified by the plan script "RHEL7 – RAC create install_RAC.sh".

```
oracle.install.responseFileVersion=/oracle/install/rspfmt_crsinstall_response_schema_v12.2.0
INVENTORY_LOCATION=/u01/app/oraInventory
oracle.install.option=CRS_CONFIG
ORACLE_BASE=/u01/app/grid
oracle.install.asm.OSDBA=asmdba
oracle.install.asm.OSOPER=asmoper
oracle.install.asm.OSASM=asmadmin
oracle.install.crs.config.gpnp.scanName=synergy-rac
oracle.install.crs.config.gpnp.scanPort=1521
oracle.install.crs.config.ClusterConfiguration=STANDALONE
oracle.install.crs.config.configureAsExtendedCluster=false
oracle.install.crs.config.clusterName=synergy-cluster
oracle.install.crs.config.gpnp.configureGNS=false
oracle.install.crs.config.autoConfigureClusterNodeVIP=false
oracle.install.crs.config.clusterNodes=racnode1.rac:racnode1-vip.rac:HUB,racnode2:racnode2-vip:HUB
oracle.install.crs.config.networkInterfaceList=ibft0:10.100.0.0:3,ibft1:10.100.0.0:3,ens3f3:10.210.40.
0:1,ens3f4:192.168.0.0:5,virbr0:192.168.122.0:3
oracle.install.asm.configureGIMRDataDG=true
oracle.install.crs.config.useIPMI=false
oracle.install.asm.storageOption=ASM
oracle.install.asmOnNAS.configureGIMRDataDG=false
oracle.install.asm.SYSASMPassword=Password1234
oracle.install.asm.diskGroup.name=VOTING
oracle.install.asm.diskGroup.redundancy=EXTERNAL
oracle.install.asm.diskGroup.AUSize=4
```

```
oracle.install.asm.diskGroup.disksWithFailureGroupNames=/dev/mapper/voting,
oracle.install.asm.diskGroup.disks=/dev/mapper/voting
oracle.install.asm.diskGroup.diskDiscoveryString=/dev/mapper
oracle.install.asm.monitorPassword=Password1234
oracle.install.asm.gimrDG.name=MGMT
oracle.install.asm.gimrDG.redundancy=EXTERNAL
oracle.install.asm.gimrDG.AUSize=4
oracle.install.asm.gimrDG.disksWithFailureGroupNames=/dev/mapper/gimr,
oracle.install.asm.gimrDG.disks=/dev/mapper/gimr
oracle.install.asm.configureAFD=false
oracle.install.crs.configureRHPS=false
oracle.install.crs.config.ignoreDownNodes=false
oracle.install.config.managementOption=NONE
oracle.install.config.omsPort=0
oracle.install.crs.rootconfig.executeRootScript=false
```

The file /home/oracle/db-RAC.rsp included the following entries. The CLUSTER_NODES setting is modified by the plan script "RHEL7 – RAC create install_RAC.sh".

```
oracle.install.responseFileVersion=/oracle/install/rspfmt_dbinstall_response_schema_v12.2.0
oracle.install.option=INSTALL_DB_SWONLY
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1
ORACLE_BASE=/u01/app/oracle
oracle.install.db.InstallEdition=EE
oracle.install.db.OSDBA_GROUP=dba
oracle.install.db.OSOPER_GROUP=oper
oracle.install.db.OSBACKUPDBA_GROUP=dba
oracle.install.db.OSDGDBA_GROUP=dba
oracle.install.db.OSKMDBA_GROUP=dba
oracle.install.db.OSRACDBA_GROUP=dba
oracle.install.db.CLUSTER_NODES=racnode1,racnode2
oracle.install.db.isRACOneInstall=false
oracle.install.db.rac.serverpoolCardinality=0
oracle.install.db.config.starterdb.type=GENERAL_PURPOSE
oracle.install.db.ConfigureAsContainerDB=false
oracle.install.db.config.starterdb.memoryOption=false
oracle.install.db.config.starterdb.installExampleSchemas=false
oracle.install.db.config.starterdb.managementOption=DEFAULT
oracle.install.db.config.starterdb.omsPort=0
oracle.install.db.config.starterdb.enableRecovery=false
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
DECLINE_SECURITY_UPDATES=true
```

Appendix I: Plan scripts for Oracle RAC installation

Note

The set of Reference Architectures based on Image Streamer, with their associated artifacts are available at https://github.com/HewlettPackard/image-streamer-reference-architectures.

RHEL7 – RAC mount

```
# List the Volume Group and lvols:
vgs
lvs
list-filesystems
# Mount the root partition (Assume that the partition type is LVM type)
-mount /dev/rhel/root /
#Create temporary directory
mkdir-p /tmp/ImageStreamer
```

RHEL7 – RAC hostname configuration

```
# Set hostname
upload -<<EOF /etc/hostname
@HostName@.@DomainName@
EOF
# Add /etc/hosts entry for hostname
upload -<<END /tmp/ImageStreamer/hosts_configure
#! /bin/bash
cp /etc/hosts /tmp/ImageStreamer/hosts
sed ’/127.0.0.1/a @PubNetwork_IP@ @HostName@.@DomainName@ @HostName@\n@OtherhostPubNet_IP@
@OtherHostname@.@DomainName@ @OtherHostname@\n@PrvNetwork_IP@ @PrvNetworkName@.@DomainName@
@PrvNetworkName@\n@OtherhostPrvNetwork_IP@ @OtherhostPrvNetName@.@DomainName@ @OtherhostPrvNetName@\n@Vip_IP@
@Vip@.@DomainName@ @Vip@\n@OtherhostVip_IP@ @OtherhostVip@.@DomainName@ @OtherhostVip@\n@Scan2_IP@
@Scan2@.@DomainName@ @Scan2@\n@Scan3_IP@ @Scan3@.@DomainName@ @Scan3@\n’ </tmp/ImageStreamer/hosts >/etc/hosts
cp /etc/sysconfig/network /tmp/ImageStreamer/network
sed ’/\HOSTNAME=/s/HOSTNAME=.*/HOSTNAME=@HostName@/g’ </tmp/ImageStreamer/network >/etc/sysconfig/network
exit 0
END
chmod 755 /tmp/ImageStreamer/hosts_configure
command /tmp/ImageStreamer/hosts_configure
echo ”-------- new /etc/hostname----------”
cat /etc/hostname
echo ”-------- new /etc/hosts ----------”
cat /etc/hosts
echo ”-------- new /etc/sysconfig/network  ----------”
cat /etc/sysconfig/network
```

RHEL7 – RAC update resolv.conf

```
upload -<<EOF /tmp/ImageStreamer/update_resolv
#!/bin/bash
ex -c ’1,$s/search *.*/search @DomainName@/’  -c wq /etc/resolv.conf
echo ”-------------new resolv.conf-----------------”
cat /etc/resolv.conf
EOF
chmod 755 /tmp/ImageStreamer/update_resolv
command /tmp/ImageStreamer/update_resolv
```

RHEL7 – RAC public and private network configuration

```
upload -<<END /tmp/ImageStreamer/pub_prv_net_configure
#! /bin/bash
```

```
cat <<EOF >/etc/sysconfig/network-scripts/ifcfg-ens3f3
DEVICE=ens3f3
BOOTPROTO=none
HWADDR=@PubNetworkHWAddr@
IPADDR=@PubNetwork_IP@
NETMASK=@PubNetwork_netmask@
GATEWAY=@PubNetwork_gateway@
ONBOOT=yes
EOF
cat <<EOF >/etc/sysconfig/network-scripts/ifcfg-ens3f4
DEVICE=ens3f4
BOOTPROTO=none
HWADDR=@PrvNetworkHWAddr@
IPADDR=@PrvNetwork_IP@
NETMASK=@PrvNetwork_netmask@
ONBOOT=yes
EOF
END
chmod 755 /tmp/ImageStreamer/pub_prv_net_configure
command /tmp/ImageStreamer/pub_prv_net_configure
echo "-------- new /etc/sysconfig/network-scripts/ifcfg-ens#f#----------"
cat /etc/sysconfig/network-scripts/ifcfg-ens3f3
cat /etc/sysconfig/network-scripts/ifcfg-ens3f4
```

RHEL7 – RAC update kernel params

```
upload -<<EOF /tmp/ImageStreamer/update_kernel_params
#!/bin/bash
ex -c '1,$s/vm.nr_hugepages *= *[0-9]*/vm.nr_hugepages = @HUGEPAGES@/g' -c '1,$s/kernel.shmmax *= *[0-
9]*/kernel.shmmax = @SHMMAX@/g' -c wq /etc/sysctl.conf
EOF
chmod 755 /tmp/ImageStreamer/update_kernel_params
command /tmp/ImageStreamer/update_kernel_params
```

RHEL7 – RAC update limits

```
upload -<<EOF /tmp/ImageStreamer/update_limits
#!/bin/bash
ex -c '1,$s/memlock *[0-9][0-9]*/memlock @MEMLOCK@/g' -c wq /etc/security/limits.d/oracle-limits.conf
EOF
chmod 755 /tmp/ImageStreamer/update_limits
command /tmp/ImageStreamer/update_limits
```

RHEL7 – RAC create udev rules and multipath.conf

```
upload -<<END /tmp/ImageStreamer/udevrules_permission_and_multipath-configuration
#! /bin/bash
cat <<CONF1>/etc/multipath.conf
# Added by Imagestreamer
defaults {
        polling_interval 10
        user_friendly_names yes
        find_multipaths yes
}
blacklist {
}
multipaths {
CONF1
echo "# Following entries added by ImageStreamer" > /etc/udev/rules.d/12-oracle-permissions.rules
IFS=$';'
# we expect input something like this "wwid1,alias1;wwid2,alias2;"
for i in `echo "@WWIDsAndAliases@"`
```

```
do
lunwwid=$(echo $i | cut -d"," -f1)
lunalias=$(echo $i | cut -d"," -f2)
echo ENV\{DM_NAME\}==\"$lunalias\", OWNER:=\"grid\", GROUP:=\"asmadmin\", MODE:=\"660\"  >> /etc/udev/rules.d/12-oracle-permissions.rules
                              cat <<CONF2>>/etc/multipath.conf
                                      multipath {
                wwid $lunwwid
                alias $lunalias
        }
CONF2
done
echo "}"  >>/etc/multipath.conf
exit 0
END
chmod 755 /tmp/ImageStreamer/udevrules_permission_and_multipath-configuration
command /tmp/ImageStreamer/udevrules_permission_and_multipath-configuration
echo "-------- new /etc/multipath.conf ----------"
cat /etc/multipath.conf
echo "-------- new /etc/udev/rules.d/12-oracle-permissions.rules ----------"
cat /etc/udev/rules.d/12-oracle-permissions.rules
```

RHEL7 – RAC create firstboot service

```
upload -<<EOF /etc/systemd/system/firstboot.service
[Unit]
Description= Configure ssh keys and install Oracle RAC
After=network.target multi-user.target multipathd.service
[Service]
Type=idle
TimeoutStartSec=0
User=root
ExecStart=/root/bootstrap/firstboot.sh
[Install]
WantedBy=multi-user.target
EOF
chmod 0664 /etc/systemd/system/firstboot.service
ln-s /etc/systemd/system/firstboot.service /etc/systemd/system/multi-user.target.wants/firstboot.service
echo "------ created firstboot.service---------------"
cat /etc/systemd/system/firstboot.service
```

RHEL7 – RAC create firstboot.sh script

```
mkdir /root/bootstrap
chmod 0744 /root/bootstrap
upload -<<EOF /root/bootstrap/firstboot.sh
#!/bin/bash
if [ ! -f  /root/bootstrap/.firstboot ]; then
        echo "First time boot, .firstboot not found"
        # wait for network to start
        sleep 30
        # only run if both nodes are up
        ping -c1 @OtherHostname@
        if [ $? -eq 0 ]; then
            if [ -f /root/bootstrap/sshkeydist.sh ]; then
                /root/bootstrap/sshkeydist.sh
                if [ $? -eq 0 ]; then
                    if [ -f /root/bootstrap/install_RAC.sh ]; then
                       /root/bootstrap/install_RAC.sh
                       if [ $? -eq 0 ]; then
                           # create .firstboot on both hosts
```

```
                        touch /root/bootstrap/.firstboot
                        ssh @OtherHostname@ 'touch /root/bootstrap/.firstboot'
                    else
                         echo "RAC install failed"
                     fi
                 else
                    echo "could not find /root/bootstrap/install_RAC.sh"
                 fi
             else
                 echo "ssh key config failed"
             fi
         else
             echo "could not find /root/bootstrap/sshkeydist.sh"
         fi
     else
         echo "no network connectivity to @OtherHostname@"
     fi
else
     echo "firstboot check complete"
fi
EOF
chmod 0754 /root/bootstrap/firstboot.sh
echo "--------------- created firstboot.sh script ---------------"
cat /root/bootstrap/firstboot.sh
```

RHEL7 – RAC create sshkeydist.sh

```
upload -<<END /root/bootstrap/sshkeydist.sh
        #! /bin/bash
        # Setting up passwdlesslogin for Oracle
        su oracle -c 'mkdir ~/.ssh; \
        chmod 700 ~/.ssh; \
        ssh-keygen -b 1024 -f /home/oracle/.ssh/id_rsa -N "";\
        sshpass -p @OracleUserPassword@ ssh-copy-id -f -o StrictHostKeyChecking=no oracle@@@OtherHostname@; \
        sshpass -p @OracleUserPassword@ ssh-copy-id  -f -o StrictHostKeyChecking=no oracle@@@HostName@; \
        sleep 3; ssh oracle@@@OtherHostname@ "(ssh-keygen -N \"\" -b 1024 -f /home/oracle/.ssh/id_rsa ; \
         sshpass -p @OracleUserPassword@ ssh-copy-id  -f -o StrictHostKeyChecking=no oracle@@@OtherHostname@; \
        sshpass -p @OracleUserPassword@ ssh-copy-id  -f -o StrictHostKeyChecking=no oracle@@@HostName@;)"'
        # Setting up passwdlesslogin for Grid
        su grid -c 'mkdir ~/.ssh; \
        chmod 700 ~/.ssh; \
        ssh-keygen -b 1024 -f /home/grid/.ssh/id_rsa -N "";\
        sshpass -p @GridUserPassword@ ssh-copy-id  -f -o StrictHostKeyChecking=no grid@@@OtherHostname@; \
         sshpass -p @GridUserPassword@ ssh-copy-id  -f -o StrictHostKeyChecking=no grid@@@HostName@; \
        sleep 3; ssh grid@@@OtherHostname@ "(ssh-keygen -N \"\" -b 1024 -f /home/grid/.ssh/id_rsa; \
         sshpass -p @GridUserPassword@ ssh-copy-id  -f -o StrictHostKeyChecking=no grid@@@OtherHostname@; \
        sshpass -p @GridUserPassword@ ssh-copy-id -f -o StrictHostKeyChecking=no grid@@@HostName@;)"'
        # Setting up passwdlesslogin for root
        mkdir /root/.ssh;
        chmod 700 /root/.ssh;
        ssh-keygen -b 1024 -f /root/.ssh/id_rsa -N "";
        sshpass -p @RootUserPassword@ ssh-copy-id  -f -o StrictHostKeyChecking=no root@@@OtherHostname@;
        sleep 5; ssh-add; ssh root@@@OtherHostname@ "(ssh-keygen -N \"\" -b 1024 -f /root/.ssh/id_rsa ; \
        sshpass -p @RootUserPassword@ ssh-copy-id  -f -o StrictHostKeyChecking=no root@@@HostName@;)"
END
chmod 0754 /root/bootstrap/sshkeydist.sh
echo "==== cat /root/bootstrap/sshkeydist.sh ===="
cat /root/bootstrap/sshkeydist.sh
```

RHEL7 – RAC create install_RAC.sh

```
upload -<<EOF /root/bootstrap/install_RAC.sh
#!/bin/bash
# install Oracle RAC
# First install grid
su - grid -c 'unzip /home/grid/linuxx64_12201_grid_home.zip -d /u01/app/12.2.0/grid'
su - grid -c 'rm -f /home/grid/linuxx64_12201_grid_home.zip'
# replace scanName, clusterName, clusterNodes, and passwords in response file
su - grid -c "sed -i 's/scanName=.*/scanName=@Scan1@/' /home/grid/grid-RAC.rsp"
su - grid -c "sed -i 's/clusterName=.*/clusterName=@ClusterName@/' /home/grid/grid-RAC.rsp"
su - grid -c "sed -i
's/clusterNodes=.*/clusterNodes=@HostName@.@DomainName@:@Vip@.@DomainName@:HUB,@OtherHostname@:@OtherhostVip@:HUB
/' /home/grid/grid-RAC.rsp"
su - grid -c "sed -i 's/SYSASMPassword=.*/SYSASMPassword=@ASMpassword@/' /home/grid/grid-RAC.rsp"
su - grid -c "sed -i 's/monitorPassword=.*/monitorPassword=@MonitorPassword@/' /home/grid/grid-RAC.rsp"
su - grid -c '/u01/app/12.2.0/grid/gridSetup.sh  -waitForCompletion -skipPrereqs -silent -responseFile
/home/grid/grid-RAC.rsp'
/u01/app/oraInventory/orainstRoot.sh
/u01/app/12.2.0/grid/root.sh
ssh @OtherHostname@ '/u01/app/oraInventory/orainstRoot.sh'
ssh @OtherHostname@ '/u01/app/12.2.0/grid/root.sh'
su - grid -c '/u01/app/12.2.0/grid/gridSetup.sh -executeConfigTools -silent -responseFile /home/grid/grid-
RAC.rsp'
# Next install Oracle DB sw
chmod 775 /u01/app
ssh @OtherHostname@ 'chmod 775 /u01/app'
# replace cluster node names in response file
su - oracle -c "sed -i 's/CLUSTER_NODES=.*/CLUSTER_NODES=@HostName@,@OtherHostname@/' /home/oracle/db-RAC.rsp"
# install Oracle DB sw
su - oracle -c '/home/oracle/database/runInstaller -waitForCompletion -skipPrereqs -silent -responsefile
/home/oracle/db-RAC.rsp'
/u01/app/oracle/product/12.2.0/dbhome_1/root.sh
ssh @OtherHostname@ '/u01/app/oracle/product/12.2.0/dbhome_1/root.sh'
EOF
chmod 0754 /root/bootstrap/install_RAC.sh
echo "-------created install_RAC.sh----------"
cat /root/bootstrap/install_RAC.sh
```

RHEL7 – RAC unmount

```
Remove the temporary directory created
rm-rf /tmp/ImageStreamer
#unmount the root partition
unmount /
```

HPE – Foundation 1.0 – attempt or fail deployment

```
# This plan script may be used to test failure of OS deployment.
echo "Attempt or Intentionally Fail OS Deployment as Requested."
@DeploymentTesting:Intentionally_Fail_OS_Deployment@
```

Appendix J: Custom attributes for deployment plan

Table 4. Custom attributes for Oracle RAC deployment plan

Custom attribute name	Default value	Comment
ASMpassword	Passwd1234	Password for SYSASMPassword parameter for Grid silent installation response file
ClusterName	synergy-cluster	RAC cluster name in Grid silent installation response file
DeploymentTesting	Intentionally_Fail_OS_Deployment	Used to cause deployment to fail for debugging purposes
DomainName	rac	Domain name for Oracle RAC hosts
GridUserPassword	Passwd1234	Password for grid user account
HostName	racnode1	Host name for RAC node
HUGEPAGES	78090	Setting for vm.nr_hugepages kernel parameter in /etc/sysctl.conf
MEMLOCK	475058678	Setting for memlock parameter in /etc/security/limits.d/oracle-limits.conf
MonitorPassword	Passwd1234	Password for monitorPassword parameter in Grid silent installation response file
OracleUserPassword	Passwd1234	Password for Oracle user account
OtherHostname	racnode2	Hostname for other node in cluster
OtherhostPrvNetName	racnode2-prv	Name for private network on other node in cluster
OtherhostPrvNetwork_IP	192.168.0.155	IP address for private network on other node in cluster
OtherhostPubNet_IP	10.210.42.155	IP address for public network for other node in cluster
OtherhostVip	racnode2-vip	Virtual IP name for other node in cluster
OtherhostVip_IP	10.210.42.140	Virtual IP address for other node in cluster
PrvNetwork_IP	192.168.0.154	IP address for private network
PrvNetwork_netmask	255.255.255.0	Network mask for private network
PrvNetworkHWAddr	0a:10:4c:d0:00:97	MAC address for network device for private network (from server profile)
PrvNetworkName	racnode1-prv	Name for private network
PubNetwork_gateway	10.210.40.1	IP address for gateway for public network
PubNetwork_IP	10.210.42.154	IP address for public network
PubNetwork_netwmask	255.255.248.0	Network mask for public network
PubNetworkHWAddr	0a:10:4c:d0:00:56	MAC address for network device for public network (from server profile)
RootUserPassword	Passwd1234	Password for root user account
Scan1	synergy-rac	Cluster scan name #1
Scan2	synergy-scan2	Cluster scan name #2
Scan2_IP	10.21.0.42.144	IP address for cluster scan name #2
Scan3	synergy-scan3	Cluster scan name #3
Scan3_IP	10.210.42.145	IP address for cluster scan name #3
SHMMAX	270255800320	Setting for shmmax kernel parameter in /etc/sysctl.conf
Vip	racnode1-vip	Virtual IP name
Vip_IP	10.210.42.139	Virtual IP address

Custom attribute name	Default value	Comment
WWIDsAndAliases	360002ac000000000000000230001f77c,data1; 360002ac000000000000000240001f77c,data2; 360002ac000000000000000250001f77c,data3; 360002ac000000000000000260001f77c,data4; 360002ac000000000000000270001f77c,data5; 360002ac000000000000000280001f77c,data6; 360002ac000000000000000290001f77c,data7; 360002ac0000000000000002a0001f77c,data8; 360002ac000000000000000470001f77c,redo1a; 360002ac000000000000000480001f77c,redo2a; 360002ac000000000000000490001f77c,redo3a; 360002ac0000000000000004a0001f77c,redo4a; 360002ac0000000000000004b0001f77c,redo5a; 360002ac0000000000000004c0001f77c,redo6a; 360002ac0000000000000004d0001f77c,redo7a; 360002ac0000000000000004e0001f77c,redo8a; 360002ac0000000000000004f0001f77c,redo1b; 360002ac000000000000000500001f77c,redo2b; 360002ac000000000000000510001f77c,redo3b; 360002ac000000000000000520001f77c,redo4b; 360002ac000000000000000530001f77c,redo5b; 360002ac000000000000000540001f77c,redo6b; 360002ac000000000000000550001f77c,redo7b; 360002ac000000000000000560001f77c,redo8b; 360002ac000000000000000450001f77c,voting; 360002ac000000000000000460001f77c,fra; 360002ac0000000000000001f0001f77c,backup; 360002ac000000000000000670001f77c,gimr;	String containing list of WWIDs and aliases for all 3PAR volumes for inclusion in /etc/multipath.conf file

Resources and additional links

HPE Reference Architectures, hpe.com/info/ra

HPE Synergy, hpe.com/info/synergy

HPE 3PAR StoreServ Storage, hpe.com/info/3par

HPE Storage for Oracle Databases, hpe.com/storage/oracle

Oracle Database 12c Release 2, Oracle Real Application Clusters, oracle.com/technetwork/products/clustering/rac-wp-12c-1896129.pdf

HPE 3PAR StoreServ Storage: A reference and best practices guide, https://h20195.www2.hpe.com/v2/GetPDF.aspx/4AA4-4524ENW.pdf

Best Practices for Oracle Database on HPE 3PAR StoreServ Storage, https://support.hpe.com/hpsc/doc/public/display?docId=emr_na-a00038978en_us

HPE Application Tuner Express, https://h20392.www2.hpe.com/portal/swdepot/displayProductInfo.do?productNumber=HPE-ATX

UEFI Workload-based Performance and Tuning Guide for HPE ProLiant Gen10 Servers and HPE Synergy, http://h20566.www2.hpe.com/hpsc/doc/public/display?docId=a00016408en_us

HPE Synergy Image Streamer – Capture RHEL 7.3, https://github.com/HewlettPackard/image-streamer-rhel/blob/v3.1/docs/rhel7.3/HPE%20Synergy%20Image%20Streamer%20RHEL%207.3%20Capture.pdf

Sample RHEL artifact bundles, https://github.com/HewlettPackard/image-streamer-rhel

HPE Synergy Image Streamer Reference Architectures, https://github.com/HewlettPackard/image-streamer-reference-architectures

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

a00049200enw, August 2018, Rev. 1