



**Hewlett Packard**  
Enterprise

# **HPE Reference Architecture for Oracle 12c high availability with HPE Integrity Superdome X, 3PAR StoreServ 8450 All Flash and Serviceguard for Linux**

Scalability and high availability for Oracle Database

# Contents

Executive summary.....	3
Introduction.....	3
Solution overview.....	4
Optimizing for performance or recovery.....	5
Solution components.....	5
HPE Integrity Superdome X hardware overview.....	5
HPE 3PAR StoreServ 8450 All Flash overview and configuration.....	6
HPE Serviceguard for Linux.....	7
Oracle Data Guard.....	7
HPE Application Tuner Express (HPE-ATX).....	8
Best practices and configuration guidance for the solution.....	9
Network configuration.....	9
nPar configuration.....	9
SGLX configuration.....	11
SGLX installation and cluster creation.....	11
nPar configuration results.....	15
Performance testing.....	15
Recovery testing.....	16
Total Cost of Ownership Comparisons.....	17
Workload description.....	18
Summary.....	18
Implementing a proof-of-concept.....	18
Appendix A: Bill of materials.....	18
Appendix B: Linux kernel configuration.....	19
Appendix C: Oracle configuration files.....	20
HPE Serviceguard for Linux Oracle configuration.....	20
HPE Serviceguard for Linux with Oracle Data Guard configuration.....	21
Appendix D: HammerDB configuration files.....	22
Master database configuration file.....	22
Slave database configuration file.....	24
Appendix E: Listener configuration.....	26
HPE Serviceguard for Linux Oracle configuration.....	26
HPE Serviceguard for Linux with Oracle Data Guard configuration.....	26
Appendix F: HPE SGLX Enterprise Edition A.12.00.50 and code customizations.....	28
Resources and additional links.....	29

## Executive summary

Demands for scalable and highly available database implementations continue to escalate in today's IT organizations. Faster transaction processing speeds, capacity-based scaling, increased flexibility, high availability and business continuity are required to meet the needs of a 24/7 business.

An Oracle RAC solution is highly available and scalable, but it is expensive to deploy, requires Oracle RAC skills and is complex to manage. The solution described in this Reference Architecture provides an alternative for those customers who need a flexible environment, using Oracle 12c with an HPE Integrity Superdome X and nPar configuration that provides a high-performing and cost-effective alternative when failover availability is acceptable as per the defined service level agreements (SLA).

This document is intended to provide customers with a baseline level of performance demonstrating the expected database scalability associated with deploying Oracle 12c on HPE Integrity Superdome X and HPE 3PAR StoreServ All Flash storage and also provide guidelines for implementing highly available Oracle databases using HPE Serviceguard for Linux® and Oracle Data Guard. It also helps customers plan and architect a solution, to provide the appropriate level of performance and continue to meet availability SLAs that are key requirements for their business enterprises.

**Target audience:** This paper is designed for IT decision makers and IT professionals who use, program, manage, or administer Oracle databases that require scalable performance and high availability. Specifically, this information is intended for those who evaluate, recommend, or design new and existing IT high-performance architectures. Additionally, CIOs may be interested in this document as an aid to guide their organizations to determine when to implement highly available Oracle databases combined with HPE Serviceguard for Linux and the performance characteristics associated with that implementation.

**Document purpose:** The purpose of this document is to describe a Reference Architecture that organizations can utilize to plan for their deployment of Oracle 12c on HPE Integrity Superdome X and HPE 3PAR StoreServ All Flash storage and also plan for highly available Oracle Databases using HPE Serviceguard for Linux and Oracle Data Guard.

This Reference Architecture (RA) describes testing completed in September 2016.

## Introduction

As enterprises deploy more and more mission-critical workloads in their data centers, the need for highly available, high-performance solutions increases. As transactions processed using mission-critical data increase, data centers need to be able to quickly and easily scale to accommodate the growing need while maintaining availability, performance, and reliability. Enterprises need infrastructure solutions that maximize application uptime, standardize or consolidate hardware, and work with different availability designs in mind.

Traditional hardware requires extended downtime for maintenance, and can often be difficult to configure with necessary high availability for mission-critical workloads. In addition, many data centers have hardware from multiple vendors, which can create compatibility issues. Due, in part, to this piecemeal arrangement, planning for resource additions to traditional solutions can be difficult. Adding new resources can be time consuming, difficult, and can necessitate downtime. Previous IT models also make it difficult for administrators to tailor existing resources to best fit the needs of various workloads.

Built from the ground up to provide for ever-growing mission-critical requirements, the HPE Integrity Superdome X platform offers a modular solution that can grow with increasing demands. Each blade enclosure can scale from one to 8 blades, offering up to 16 CPU sockets and 24 TB of memory. With HPE nPar technology, IT can partition resources to create isolated environments for different applications. Each nPar has its own resources, operating system, applications, and electrical supply, providing increased protection from failures in other partitions. When maintenance is required, IT can take each nPar offline individually, allowing the rest of the enclosure to continue working. The Superdome X platform also offers five nines (99.999%) single-system availability by leveraging several reliability, availability, and serviceability (RAS) features.

The HPE 3PAR StoreServ 8450 All Flash storage array offers high-speed flash drives to boost application performance. A single system offers scaling storage that can grow with data and application needs. Automated provisioning and multi-tenant design introduce flexibility, while deduplication and HPE thin technologies save storage space.

In addition to hardware design considerations for a highly available Oracle solution, one must consider the software design options. At one end of the availability continuum is Oracle RAC, a relatively expensive option that adds complexities, but provides near-instant recovery options. With other less expensive and less complex options, recovery times are longer but might still be acceptable depending on an organization's SLA. One

example is HPE Serviceguard for Linux (SGLX), a software solution that brings high availability and disaster recovery capabilities. Used alone, SGLX is an economical option. Pairing SGLX with Oracle Data Guard (ODG) is an approach that further shortens the recovery time.

Every level of a data center - server, storage, network, OS, virtualization layer, virtual machine guest - must be monitored at all times. When SGLX detects a failure, it follows shutdown procedures, and moves applications to an available, appropriate resource - whether physical or virtual - to bring them back online. SGLX provides monitoring, failover, data loss prevention, and limited-to-no planned downtime. This Reference Architecture shows how SGLX works in the test environment during an unplanned failure.

## Solution overview

Traditional hardware stacks used for hosting applications can be difficult to scale up to meet growing demands. They consist of several, often proprietary, pieces that make adding hardware difficult and costly. In addition, guaranteeing maximum uptime for workloads can involve wasted resources, complicated setups, and inconsistent techniques. This Reference Architecture offers a blueprint for building a solution for Oracle Online Transaction Processing (OLTP) databases on the HPE Integrity Superdome X platform using HPE 3PAR StoreServ 8450 All Flash storage. This document also highlights business continuity solutions along with best practices for Oracle Database 12c with HPE Serviceguard for Linux and Oracle Data Guard tools and demonstrates the failover features of the HPE solution. This solution can be used for both OLTP and Online Analytical Processing (OLAP) databases.

In this RA, HPE nPar technology is utilized to build separate physical partitions. Each nPar acts as an independent server with its own resources. Each nPar can be adjusted to add or remove blades.

The Superdome X was divided into 2 nPars, each with 4 server blades, 288 logical processors (Hyper-Threading enabled), and 6 TB of memory. One partition housed an OLTP database driven by a HammerDB workload to generate performance and system utilization information and had SGLX installed for database availability.

---

### Note

Because Oracle licensing agreements prohibit the publication of absolute test results, the performance test results are normalized to those of the SGLX configuration without Oracle Data Guard.

---

The Reference Architecture consists of Hewlett Packard Enterprise hardware and software, as well as software from Red Hat® and Oracle, and consists of the following components:

- HPE Integrity Superdome X
  - 8x HPE BL920s Gen9 Intel® Xeon® E7-8890v3 2.5GHz 36-core blades
  - 12 TB of total memory, 1.5 TB per server blade
- 2x HPE 6125XLG Ethernet blade switch
- 4x Brocade 16Gb/28c PP+ embedded SAN switch
- HPE 3PAR StoreServ 8450 All Flash 4-node
  - 80x 480GB SSDs
- 1x HPE FlexFabric 5800 switch
- 1x HPE FlexFabric 5930 switch
- Red Hat Enterprise Linux v7.2
- HPE-ATX version 1.0.0-51.31
- Oracle Database 12c
- Oracle Data Guard 12c
- HPE Serviceguard for Linux Enterprise Edition A.12.00.50

---

**Note**

This Reference Architecture describes testing completed with HPE SGLX Enterprise Edition A.12.00.50 and code customizations as shown in [Appendix F](#) to enable HPE ATX for Oracle with SGLX. Since the original publication of this RA, the code customizations have been productized in HPE SGLX A.12.00.51.

---

**Optimizing for performance or recovery**

When designing your system architecture, technical choices must be made based on the goals of your SLA and your stakeholder input. Configurations requiring peak performance will have different parameters than those requiring fast recovery and failover times.

As such, this study presents results from two different test configurations. The first configuration demonstrates the performance characteristics of an 8-socket nPar configuration. The second test configuration presents the recovery characteristics of an 8-socket nPar configuration, measuring the time it takes to recover from a failure. In both of these configurations SGLX was used with Oracle Database 12c, with and without Oracle Data Guard. The performance-impact configuration varied from the recovery-impact configuration, with each configuration tailored to the goals of that test.

For the performance portion of the test, configuration options were selected to optimize database performance, which was measured in transactions per minute (TPM). The SGLX configuration without Oracle Data Guard was run first, then the configuration with SGLX and Oracle Data Guard was run. The TPM results were normalized to the SGLX configuration without Oracle Data Guard, and reported later in this document.

For the recovery portion of the test, configuration options were selected to optimize recovery times. The performance workload was repeated and a recovery event was triggered, while the HammerDB workload ran to simulate heavy database use. Oracle Data Guard was then introduced, and the recovery test was repeated.

**Solution components**

The following components make up the Reference Architecture solution.

**HPE Integrity Superdome X hardware overview**

The HPE Integrity Superdome X solution consists of the 18U-high HPE BladeSystem Superdome enclosure and HPE BL920s Gen9 server blades. Each enclosure can support up to 8 server blades that can be configured in a single hardware partition, or can be separated into smaller HPE nPars of one, 2, 3, 4, 6 or 8 server blades each. Each server blade is equipped with 2 Intel Xeon E7 v3 or v4 processors, up to 3 TB of DDR4 memory, 2 FlexLOM slots, and 3 Mezzanine PCIe Gen3 slots. With HPE nPar technology, systems can be customized with logically and electrically isolated partitions sized for any workload.

Several features of the Superdome X contribute to its performance and modular structure. At the most basic level is each processor and its associated local memory. With each processor socket equipped with an embedded memory controller, each CPU can access its local memory at extremely fast rates. Each pair of processors is connected via Intel QuickPath Interconnect (QPI) links. Hewlett Packard Enterprise fault tolerant Crossbar Fabric Modules (XMFs) facilitate communication among the server blades. The structure of these features provides the Superdome X with a non-uniform memory architecture (NUMA). All latencies between the processors and memory are relative to the positioning of the processor and the memory DIMM, where local memory has lower latencies than memory accessed across the interconnect fabric.

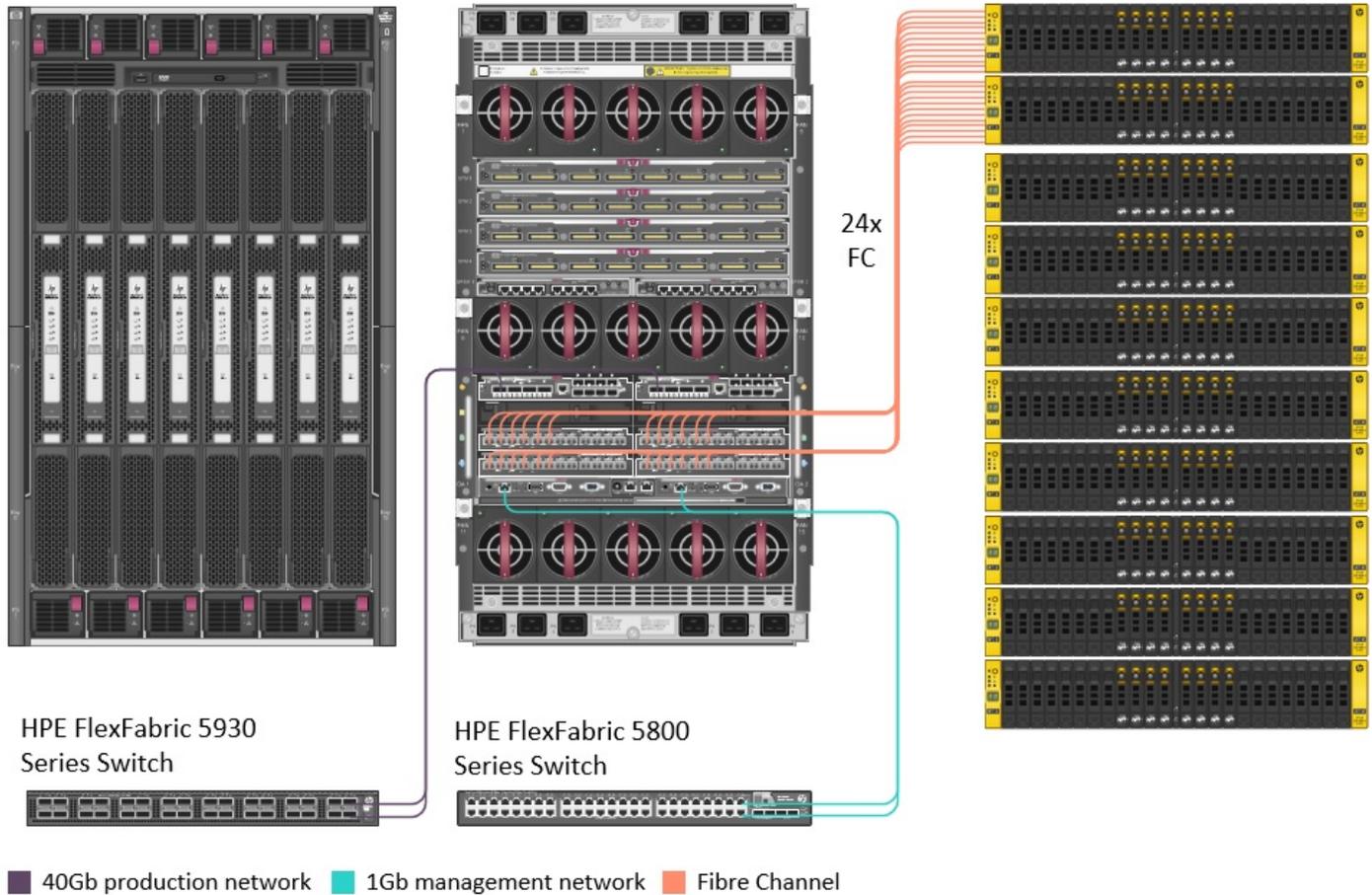
The following diagram shows the HPE Integrity Superdome X solution as configured for this Reference Architecture.

### HPE Integrity Superdome X

**Front:** 8x HPE BL920s Gen9 Server Blade

**Back:** 4x Brocade 16Gb SAN Switch for HPE BladeSystem c-Class and 2x HPE 6125XLG Ethernet Blade Switch

HPE 3PAR StoreServ 8450 all-flash 4-node Storage Base with 8x expansion trays



**Figure 1.** Logical solution diagram of HPE Integrity Superdome X with HPE 3PAR StoreServ 8450 All Flash storage array

**Note**

See the QuickSpecs links in the “Resources and additional links” section for more detailed information about hardware configurations and specifications.

**HPE 3PAR StoreServ 8450 All Flash overview and configuration**

The HPE 3PAR StoreServ 8450 All Flash storage array has been designed with the needs of mainstream midrange and enterprise deployments in mind. In addition to the 4-node configuration in this Reference Architecture, it is also available in a 2-node configuration. This storage system offers flash-optimized performance with the necessary resiliency, efficiency, and data mobility needed for diverse, mission-critical workloads. With such technologies as HPE 3PAR Gen5 Thin Express ASIC, the array can support mixed workloads while saving space by allowing thin

technologies to take place in silicon, including deduplication. Run transaction- and throughput-intensive workloads on the same storage resources without compromising the performance of either. Incorporate flash-optimized storage into the existing infrastructure without having to move to an entirely new architecture. Because SSD costs are dropping, Tier-1 data services, quad-node resiliency, and flexibility are available at midrange affordability. The HPE 3PAR StoreServ 8450 All Flash storage array can grow to meet the needs of businesses by adding enclosures to the environment.

## **HPE Serviceguard for Linux**

Increase the availability and uptime of your mission-critical applications and minimize the impact of unplanned outages with HPE Serviceguard for Linux. Providing both high availability (HA) and disaster recovery (DR) features, HPE Serviceguard for Linux packages applications and other services with their associated resources and monitors the entire package for any failure. Each package is monitored for faults related to hardware, software, OS, virtualization layer, virtual machine guests, network, and storage. When any failure is detected, SGLX shuts down the application to quickly and smartly relocate the application or service to another system with the necessary resources to bring it into production again. An HPE Serviceguard for Linux cluster allows for multiple node failures using redundant hardware to remove any single points of failure. This RA focuses on showing how quickly the database on the failed node returns to production after the failover event, as well as the performance of the databases before and after the failover event.

In addition, HPE Serviceguard for Linux uses Lock LUN and Quorum Server arbitration mechanisms to prevent data corruption and loss. The solutions in this Reference Architecture use the Lock LUN configuration. SGLX also minimizes planned downtimes using its Live Application Detach (LAD) and Rolling Upgrades features to perform maintenance on clusters and install upgrades for any OS and application without downtime. Take advantage of the Cluster Verification technology to find and fix cluster configuration issues before they advance and cause unplanned downtime.

## **SGLX-based HA and DR solution for Oracle Database 12c**

HPE Serviceguard for Linux provides automatic HA and DR protection for Oracle databases. This solution manages and completely automates the failover process of Oracle databases. Oracle is installed on each of the cluster nodes and the databases are configured such that the database data is located on shared storage which can be accessible from all the cluster nodes. The HPE Serviceguard Oracle toolkit allows SGLX to start, stop, monitor and recover the databases on the adoptive cluster nodes in case of failures. In this solution SGLX recovers the database by restarting the Oracle database on the adoptive cluster node.

## **Oracle Data Guard**

Oracle Data Guard enables Oracle databases to survive disasters and data corruption using its services that create, maintain, manage, and monitor the standby databases. ODG maintains these standby databases as consistent copies of the production database. If the production database is unavailable, in the case of a planned or unplanned outage, Oracle Data Guard can allow a standby database to take over the production role, thus reducing the downtime associated with the outage. ODG can be used with traditional backup, restoration, and cluster techniques to provide a high level of data protection and data availability. An Oracle Data Guard configuration consists of one production database, known as the primary database, and up to thirty standby databases. Each standby database is transactionally consistent with the primary (production) database, maintained by transmitting redo data from the primary database to the standby database.

There are three main types of standby databases: physical, logical, and snapshot. Physical databases are physically identical copies of the primary database on a block-for-block basis. The entire database, including schema and indexes, are identical. Oracle Data Guard uses Redo Apply technology to keep the standby database synchronized with the primary database. In addition, a physical standby database can be used for read-only access while maintaining synchronization, allowing applications to use standby databases for data protection and reporting.

Logical databases have the same logical information as the primary database, but the physical structure of the data can differ. Oracle Data Guard uses a SQL Apply feature to keep logical standby databases in sync with primary databases using SQL statements. Logical standby databases allow for upgrading Oracle Database software in a rolling fashion with next to no downtime.

Oracle Data Guard also has three protection modes: maximum availability, maximum performance, and maximum protection. The maximum availability protection mode is the highest level of data protection possible without compromising the availability of a primary database. In this mode, all redo data needed to recover the primary database is collected by a synchronized standby database before any transactions are committed. However, if the primary database is unable to write the redo data to a standby database, it will revert to maximum performance mode until it is able to start writing again. This will ensure that data will remain after a primary database failure so long as the standby database is intact.

The maximum performance protection mode offers the maximum amount of protection possible without affecting the performance of the primary database. This mode, which is the default mode, allows transactions to commit once the redo data is written to an online log, but before the data has been transferred to a standby database. Thus, the primary database isn't affected by slow redo writing to standby databases.

The maximum protection mode is similar to the maximum availability mode, but any time the primary database cannot write to the standby database, the primary database will be shut down. Due to the impact this mode could have on availability of the primary database, Oracle recommends any primary database set in this mode have at least two standby databases to help minimize downtime. This mode ensures no data loss on a primary database failure.

The databases used in this Reference Architecture used physical standby databases, and protection mode was set to maximum availability.

ODG replicates the data to one or more standby databases which can be used in case of failure of the primary database. Integrating ODG with SGLX has the following advantages:

- Provides high availability and disaster recovery with fully automatic and fast recovery for Oracle databases
- Monitoring for primary and secondary DB instances and the Oracle Data Guard processes
- Recovery Point Objective sensitive recovery
- Built-in monitoring capabilities to check system resources, such as network, volume groups, files systems, etc. and failover in the case of failure of any of these components
- Protect and recover the application stacks in addition to databases
- Failure and status notifications
- Advanced features to minimize downtime for planned maintenance

### **SGLX-based HA/DR solution for Oracle Database using Oracle Data Guard**

HPE Serviceguard for Linux along with Oracle Data Guard provides HA and DR protection for Oracle databases with much faster recovery times as compared to the solution with SGLX without Oracle Data Guard. HPE Serviceguard for Linux manages and completely automates the failover process of Oracle Data Guard databases. In this solution, primary and standby databases are configured with Oracle Data Guard. SGLX is utilized to start, stop, and monitor the databases and administer the replication between primary and standby databases. SGLX also performs automatic role management to recover from failures. In case of failures the solution automatically recovers the Oracle database by promotion of the standby database instance to primary. This mode of recovery is much faster compared to the restart-based solution as described above. The databases can be located on the same premises or in geographically dispersed data centers. SGLX performs recovery point objective (RPO) sensitive automatic role management to recover from failures.

### **HPE Application Tuner Express (HPE-ATX)**

HPE Application Tuner Express is a utility for Linux users to achieve maximum performance when running on larger 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 configuration policies:

- Round Robin Tree: Use a round robin distribution for processes. Include the root process/thread and all of its descendants.
- Round Robin Flat: Use a round robin distribution for processes. Include the root process/thread and only its direct descendants.
- Fill First Tree: Completely allocate all cores and threads within a physical processor then move on to the next processor. Include the root process/thread and all of its descendants.
- Fill First Flat: Completely allocate all cores and threads within a physical processor then move on to the next processor. Include the root process/thread and only its direct descendants.
- Pack: The root process/thread and all of its descendants will be launched on the same NUMA node/processor.

HPE-ATX is fully supported by Hewlett Packard Enterprise and can be downloaded from HPE Software Depot. The HPE-ATX tool was used only for the performance testing. HPE-ATX increases performance, but does not enhance the recovery process, so it was not used for the recovery testing.

## Best practices and configuration guidance for the solution

This Reference Architecture features the HPE Integrity Superdome X and the HPE 3PAR StoreServ 8450 All Flash storage array. The Superdome X used a boot-from-SAN architecture and all software and data were stored on the All Flash 3PAR array connected through Fibre Channel connections.

The HPE 3PAR StoreServ 8450 All Flash storage array used in this testing was configured as follows:

- 4-node
- 384 GB of cache
- 8x expansion shelves
- 80x 480GB SSD
- 24x 16Gb Fibre Channel ports

### Network configuration

#### Storage network

The HPE Superdome X comes equipped with four (4) Brocade 16Gb/28 SAN Switch Power Pack+ switches that communicate with the blades through fabric interconnects. The HPE 3PAR StoreServ 8450 All Flash storage array was connected to these switches using 24 Fibre cables. The transceivers used in the Superdome X were rated at 8Gb, so the Fibre Channel connections ran at 8Gb per second.

#### Client network

The Superdome X comes equipped with two (2) 6125XLG 10/40Gb switches in pass-through mode, one per interconnect module (ICM). Each switch has one (1) 40Gb port connected to an HPE FlexFabric 5930 switch. The client system is connected to the Superdome X via the FlexFabric 5930 switch.

For HPE Integrated Lights-Out (iLO) management, there are two (2) 1Gb ports connected to an HPE FlexFabric 5800 switch for management from the Superdome X, and two (2) 1Gb ports connected to the same HPE FlexFabric 5800 switch for storage management on the 3PAR array.

#### Network interface configuration

One physical network interface on the nPar was used for each Oracle pluggable database (PDB). Oracle listeners were placed on the NIC that was physically closest to the NUMA node it was serving.

---

### Note

When working with SGLX, be sure to limit NIC names to only 12 characters. For more information, see issue QXCR1001424075 in the “HPE Serviceguard for Linux 12.00.50 Release Notes” document. Links to all documents can be found in the Resources and additional links section at the end of this paper.

---

### nPar configuration

For this Reference Architecture, SGLX was used to create a highly available, redundant cluster for an 8-socket configuration. The first nPar was designated as the primary partition, while the second nPar was designated as the secondary partition. Hyper-Threading was enabled in each configuration.

### Storage configuration

The storage was configured according to Oracle best practices with two automatic storage management (ASM) disk groups: a data area and a fast recovery area. The data area contains the active database files, and any other files necessary for the level of redundancy being used. The fast recovery area contains all files related to recovery in the event of a failure, such as archives, file copies, and so on. For more information, see the Oracle Database High Availability Best Practices guide, [https://docs.oracle.com/database/121/HABPT/config\\_storage.htm#HABPT4790](https://docs.oracle.com/database/121/HABPT/config_storage.htm#HABPT4790).

For the OS, the configuration had one 500GB LUN per nPar, configured to boot from SAN. As there were two nPars in a cluster, one 0.25GB LUN was created as a lock LUN to facilitate communication between the server nodes. A quorum server could be used instead, if desired. The SGLX only LUNs were created using the 3PAR common provisioning groups (CPG) for RAID1 SSDs configuration option and formatted as fully provisioned. The SGLX with ODG was configured as thin provisioned RAID 5. Different RAID configurations and provisioning types were used due to the capacity of the array. Because the SGLX with ODG configuration required a standby database, additional log groups, and more physical LUNs, RAID5 and thin provisioning was used to conserve space.

### Operating system configuration and tuning

The nPar configuration ran Red Hat Enterprise Linux version 7.2 with Oracle-recommended settings for the operating system Linux kernel parameters, as listed in [Appendix B](#). For more information on Oracle recommended operating system parameters, reference the Oracle Database 12c installation guide for Linux; all resource links can be found in the Resources and additional links section at the end of this document.

### I/O tuning

The following OS settings were applied to the Linux SCSI devices (see also, "Optimizing Linux performance on HPE Integrity Superdome X"). Experimentation occurred, in particular, with the queue depth for each nPar. The queue length depends on the Superdome X and 3PAR HBA's queue depth, and the number of backing multipath volumes. You can run these commands after each reboot, or add them to the udev rules in `/etc/udev/rules.d` to make the changes persistent. For more information on these options, see "How to calculate suitable LUN queue depths: Host I/O queues and HPE 3PAR StoreServ Storage" in the "Resources and additional links" section below.

```
for i in /sys/block/*/queue/nomerges ; do echo 1 > $i; done
for i in /sys/block/sd*/device/queue_depth ; do echo 32 > $i; done
for i in /sys/block/sd*/queue/nr_requests ; do echo 1024 > $i; done
```

### Oracle configuration

The tests ran with an OLTP workload built by the HammerDB tool. For the 8-socket tests in the performance configuration and the recovery configuration, we used two Oracle Pluggable Databases (PDBs). Each PDB was sized to 2,500 warehouses. Each PDB had its own dedicated Oracle listener connected to a unique IP address.

### Resource limits

OS resource-usage limits were set for the Oracle user by creating the file `/etc/security/limits.d/99-grid-oracle.conf` with the following contents:

```
oracle - nproc 16384
oracle - nofile 65536
oracle - stack 32768
oracle - memlock 12058162960
```

### Oracle spfiles

We used different Oracle configuration options for the SGLX (without and with ODG) testing for optimum performance on each, but the options were similar. The parameters are in [Appendix C: Oracle configuration files](#)

**Oracle logs**

For the performance configuration, two (2) 200GB redo log groups were used for the database. For the recovery testing with SGLX and SGLX with ODG, seven (7) 10GB redo log groups were used for the primary database and eight (8) 10GB redo log groups were used for the standby database. In both cases, the logs were stored in the fast recovery area created on the storage. Different sizes and numbers of logs were used in order to optimize for each scenario. Larger logs are good for performance testing because they prevent the system from switching between logs, which can hinder performance. Having a larger number of smaller logs helps shorten recovery time.

**Oracle Data Guard**

Oracle Data Guard was configured using a physical standby database with maximum availability mode protection. The configuration used the Oracle Broker service, and the database used Flashback. The FAST\_START\_FAILOVER feature was disabled in the ODG Broker as required by SGLX for ODG.

**HammerDB configuration**

HammerDB was configured to use one instance of the 2,500 warehouses per PDB and tested with 144 virtual users per PDB in the performance configuration, and 72 virtual users in the recovery configuration. This study uses the TPM output from HammerDB to gauge performance.

Two PDBs were used. These databases used the AWR driver to gather Oracle AWR statistics. The second database was a slave database that used the default driver. To see an example of the configuration files, see Appendix D: HammerDB configuration files.

**SGLX configuration****SGLX configuration for heartbeat networks:**

- SGLX requires a minimum of one channel bonded network or two separate networks (one port each from two different network cards to avoid Single Points of Failure (SPOF)).
- It is recommended to have one dedicated network if possible, for increased resiliency, though heartbeats can be configured on all available networks too.

**SGLX cluster arbitration requirements (lock LUN or quorum server):**

Lock LUN – This is a disk based arbitration mechanism between cluster nodes. It requires a minimum of one 100 MB shared disk, accessible from both cluster nodes with redundant paths to the disk from each node.

Quorum server based arbitration can also be used. A quorum server is a standalone server outside of the cluster, with redundant network connectivity to arbitrate between cluster nodes.

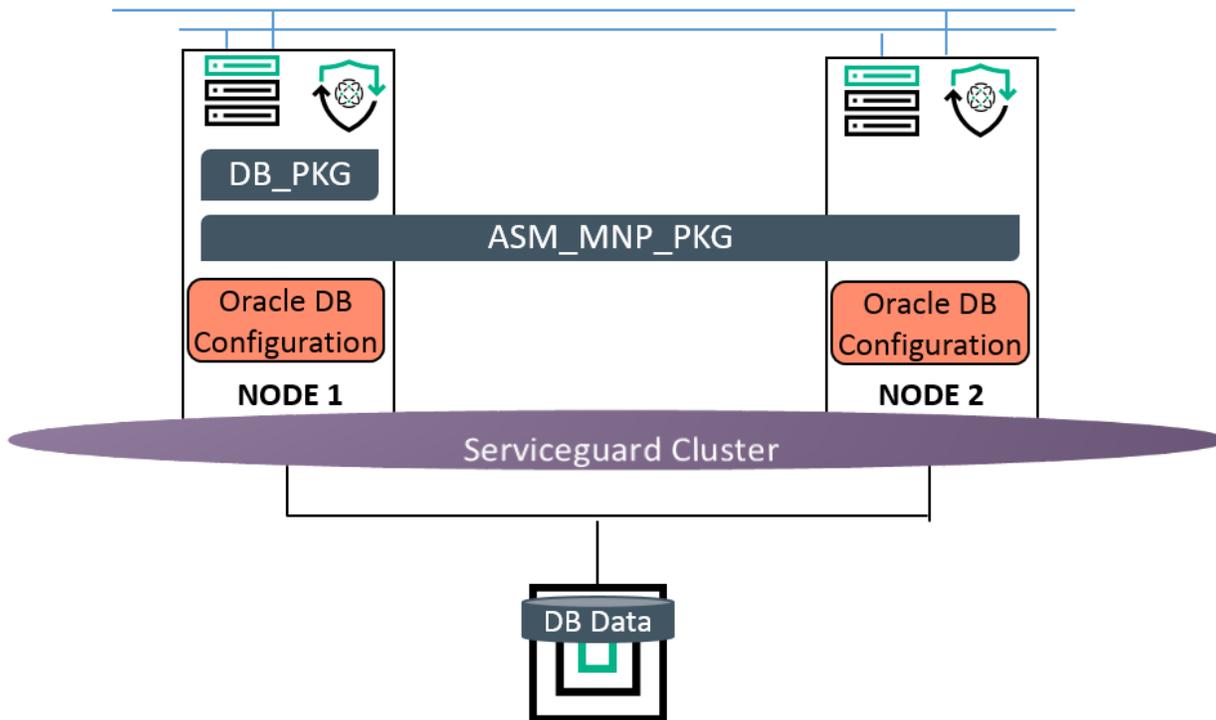
In this configuration, SGLX is configured with a bonded network as heartbeat and a lock LUN is configured as the cluster arbitration mechanism.

**SGLX installation and cluster creation**

- Install OS and Install SGLX using “cmeasyinstall”
- Once the installation is complete, create a SGLX cluster

**HA solution for Oracle databases (without Oracle Data Guard)**

- Install Oracle Grid ASM and database binaries local to each cluster node. While installing ensure that the ASM SID's are the same for both the nodes.
- Create ASM disk groups that are accessible from all cluster nodes (shared storage) to be used as database stores.
- Create SGLX packages that manage the ASM processes and database instance(s).
  - Oracle ASM processes are running on both nodes and the multi-node package named “ASM\_MNP\_PKG” manages them on the two nodes.
  - The Oracle DB is configured to run on both the nodes and will be active only on one node at any point of time. This database is managed by the failover package named “DB\_PKG” and is active on the nodes where this package is running.
- In this configuration, the following is the shared LUN configuration used for the Oracle baseline configuration:
  - Eight (8) 500GB LUNs were created and used for the data area disk group, and eight (8) 500GB LUNs were assigned for the fast recovery area disk group. The disk groups were set up as shared storage between the two nPars.
  - Two (2) 5GB LUNs were configured to hold the ASM configuration data for each of the two nPars.



**Figure 2.** Schematic representation of cluster and packages after the above steps have been performed for SGLX solution without ODG for Oracle DB.

**HA/DR solution configuration for Oracle database using Oracle Data Guard**

- Install Oracle Grid (ASM) and database binaries local to each cluster node.
- Create ASM disk groups for application data and fast recovery area that are local to each node.
- Deploy the solution using “cmoradgworkload” utility:
  - The Oracle ASM processes are running on both the nodes and are managed by the packages named “ASM\_MNP\_PKG” and “ASM1\_MNP\_PKG” on the respective nodes.

- The Oracle DB is configured to run on both the nodes with one nodes being the primary and the other the standby. This database is managed by the packages named “DB\_PKG” and “DB1\_PKG” on the respective nodes.
- The DB roles are managed by the package named “RM\_PKG”. This package will always run on the node which is the primary.
- Sixteen (16) 500GB LUNs were created and used for the two data area disk groups, and sixteen (16) 500GB LUNs were created and used for the two fast recovery areas. The ASM disk groups were not set up as shared storage. Instead, each nPar had its own ASM data and fast recovery area (8 LUNs for each).

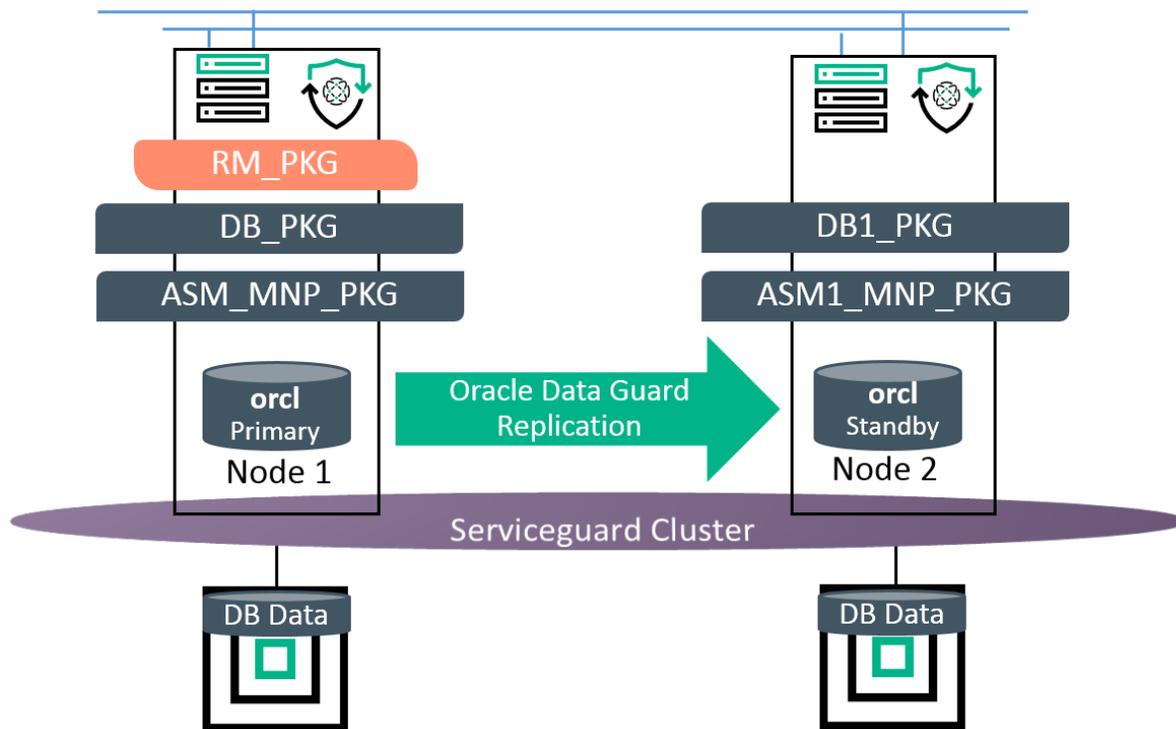


Figure 3. Schematic representation of cluster and packages after the above steps have been performed for SGLX solution with ODG for Oracle DB

### Executing the tests

For the performance configuration, the HammerDB workload ran a 3-minute ramp-up and a 5-minute test period for both of the SGLX configurations – with and without Oracle Data Guard. All results reported in this document are from the 5-minute test period. The entire HammerDB test cycle was executed first to gauge performance in TPM. The normalized TPM reported is only from the performance configuration.

For the recovery configuration, the HammerDB workload ran a 3-minute ramp-up and a 5-minute test period, with the altered parameters listed in [Appendix D](#) for recovery. The recovery scenario was run with SGLX, with and without Oracle Data Guard.

For the recovery event running SGLX without ODG, the primary nPar was powered off using the HPE Integrated Lights-Out (iLO) management console at 3 minutes into the HammerDB 5-minute test period. All system times were synced using NTP, and a script running on the primary nPar recorded every second to a log file. The time to recovery for both configurations (with and without Oracle Data Guard) is the time elapsed between (1) the last time recorded in the script log file, which represents the beginning of the failover event, and (2) the end time reported by SGLX, which represents the database returning to an up-and-running state.

### Oracle startup process

In the performance configuration, the HPE-ATX tool was used to start the Oracle listeners. We modified the halistener.mon file to invoke the HPE-ATX tool with the changes found in [Appendix E](#).

The HPE-ATX tool was used only for the performance testing. HPE-ATX increases performance, but does not enhance the recovery process, so it was not used for the recovery testing. For recovery testing the halistener.mon that ships with SGLX was used.

The Oracle listeners (listeners.ora) and TNS names files (tnsnames.ora) were updated with the details of additional listeners. For all testing, there was one additional listener per PDB; and, in the SGLX with ODG testing, a third listener was added for communication between the primary and secondary nodes for Oracle Data Guard communication. The contents of the listeners.ora and tnsnames.ora can be found in [Appendix E](#).

### Oracle database startup

In the performance configuration, the HPE-ATX tool was also used to start the Oracle database on the primary nPar. We modified the haoracle\_sql.sh with the changes found in [Appendix E](#).

As with the listener startup, the HPE-ATX tool was used only for the performance testing. For recovery testing the haoracle\_sql.sh that ships with SGLX was used

### Starting the HammerDB instance(s)

The following script was used to start HammerDB with the necessary configuration files. The script has a modifiable parameter to dictate the number of instances to start.

```
#!/bin/bash

# set the number of Hammerdb instances to run
num=2

# backup current config file
mv config.xml config.xml-$$ >& /dev/null

for i in $(eval echo p0{1..$num}); do
    echo $i
    ln -s config.xml-$i config.xml
    ls -l config.xml
    ./hammerdb.tcl &
    sleep 4
    rm config.xml
done
```

After starting the PDBs, several of their objects were tagged as “hot,” which resulted in some data copying among the NUMA nodes by running the following SQL file, hot-obj.sql via the following command:

```
sqlplus / @hot-obj.sql
```

The file hot-obj.sql contains the following:

```
alter session set container = p01;
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_RANDOM',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_RANDOM',NAMESPACE=>2);
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_OUTPUT',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_OUTPUT',NAMESPACE=>2);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'NEWORD',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'PAYMENT',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'DELIVERY',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'OSTAT',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'SLEV',NAMESPACE=>1);
```

```
alter session set container = p02;
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_RANDOM',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_RANDOM',NAMESPACE=>2);
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_OUTPUT',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_OUTPUT',NAMESPACE=>2);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'NEWORD',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'PAYMENT',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'DELIVERY',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'OSTAT',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'SLEV',NAMESPACE=>1);
```

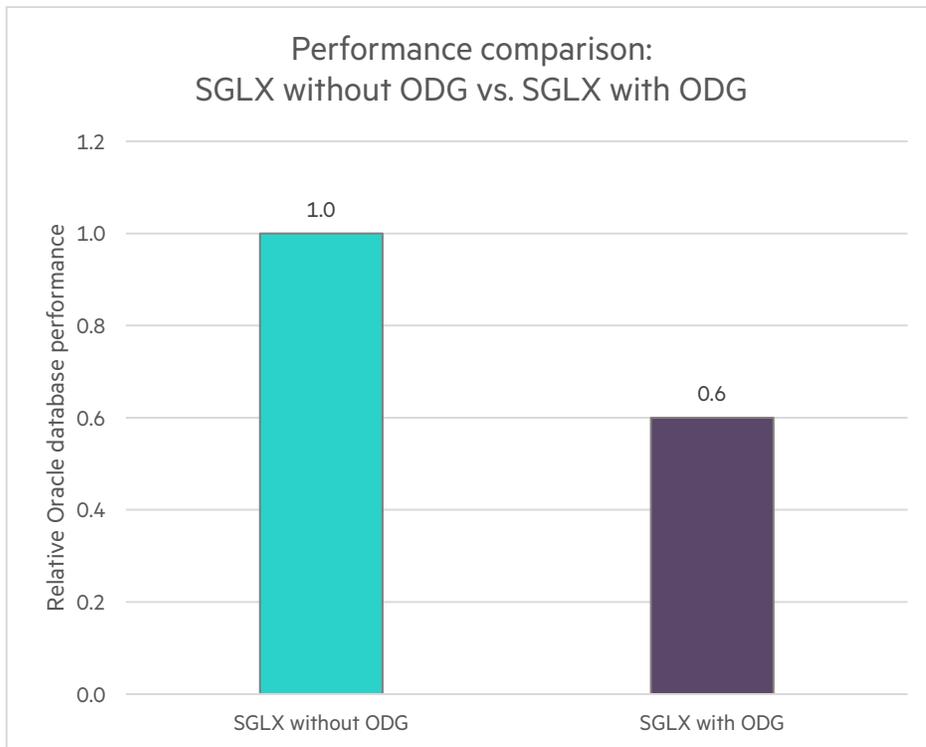
```
alter session set container = cdb$root;
exit;
```

## nPar configuration results

### Performance testing

As was mentioned earlier, during the performance phase of testing, configuration options were chosen to maximize database performance. Two performance configurations were tested. The first configuration used SGLX without Oracle Data Guard. The second configuration used SGLX with Oracle Data Guard.

The test was performed by running the HammerDB workload on the configuration first with SGLX without ODG and then a second time with SGLX and with ODG. As shown in Figure 4, a value of 1.0 was assigned to the number of transactions per minute performed by the SGLX without ODG configuration. When ODG was added, relative performance decreased by 40 percent due to the nature of the ODG replication overhead.



**Figure 4.** Relative performance of the 8-socket performance configuration before and after ODG and SGLX for ODG is installed

## Recovery testing

During the recovery phase of testing, configuration options were chosen to minimize recovery time. As with the performance testing, the recovery test was performed by running the HammerDB workload on the configuration first with SGLX without ODG and then a second time with SGLX with Oracle Data Guard.

As shown in Figure 5, total recovery time for the SGLX without ODG configuration was 13 minutes and 8 seconds. When ODG was added with SGLX, total recovery time decreased to 3 minutes and 22 seconds, a decrease of 74.4 percent. As the breakdown of the recovery process shows, this decrease can be traced to recovery time. While the detection phase of the process took 48 seconds on both configurations, recovery time took 12 minutes and 20 seconds on the SGLX configuration without ODG and 2 minutes and 34 seconds on the SGLX with the Oracle Data Guard configuration. This is due to the fact that the SGLX configuration had to recover all unapplied changes to the database, while the SGLX with ODG configuration had to recover less, as ODG applies changes in real time.

Note that the 48 seconds required to identify the problem and initiate a switchover could be further minimized by reducing the SGLX MEMBER\_TIMEOUT parameter. The default value for this parameter is 14 seconds, but the minimum value for this parameter is 3 seconds for a cluster with more than one heartbeat subnet. Setting lower values enables faster detection times making the cluster more sensitive.

Even that is an over simplification of the value that SGLX provides. While the ODG recovery time will be consistent regardless of the method used to initiate a recovery, had ODG had been deployed without the use of SGLX, the time to recognize and initiate a failover would be much longer and would include the following manual steps:

1. In case of failure, ensure graceful shutdown of the primary when possible
2. Identify a problem
3. Look up the commands required to make the backup database the new primary database
4. Enter and execute the commands
5. Check the consistency of the new primary database
6. Manually bring up the rest of the resources like IPs, listeners and the application stack

Additionally, you have the assurance SGLX will execute the same recovery commands in the same order, without error, every time. This is a level of consistency and rapidity that cannot be replicated by a human.

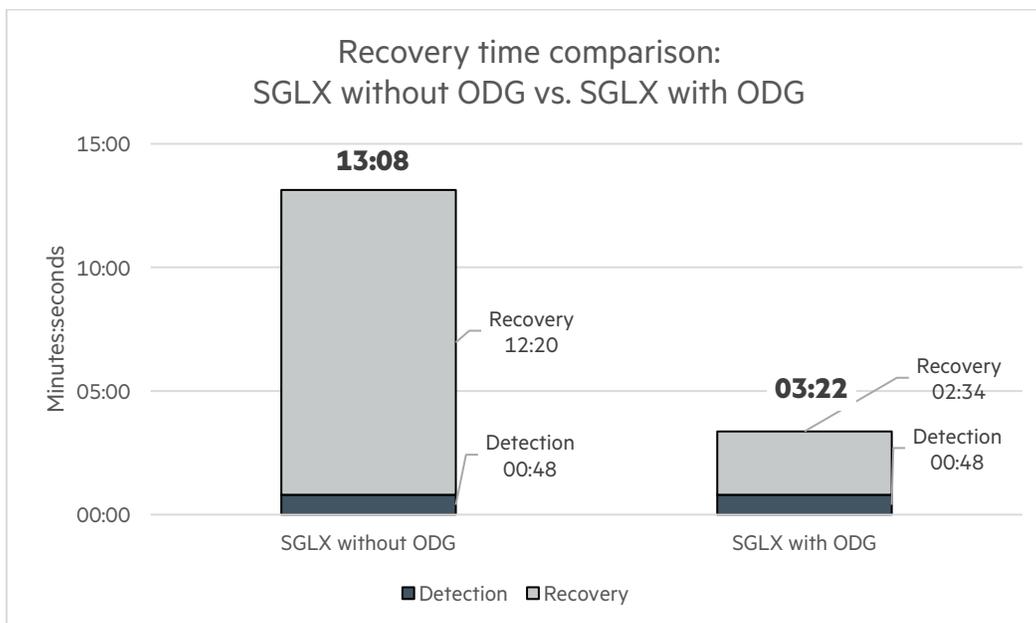


Figure 5. Time to recover from failover event

### Total Cost of Ownership Comparisons

Figure 6 and Figure 7 below show a total cost of ownership (TCO) comparison between an Oracle Exadata solution that is based on Oracle RAC and an HPE solution based on Superdome X, HPE 3PAR and HPE Serviceguard for Linux with Oracle Data Guard. The TCO comparison is based on an 8 socket configuration, which takes into account initial hardware and software cost, hardware and software support cost, implementation, infrastructure cost (space and power) and operational cost for a period of three years.<sup>1</sup> As the data shows, the Hewlett Packard Enterprise solution provides a 38% savings compared to the Oracle Exadata solution cost, implementation, infrastructure cost (space and power) and operational cost for a period of three years.

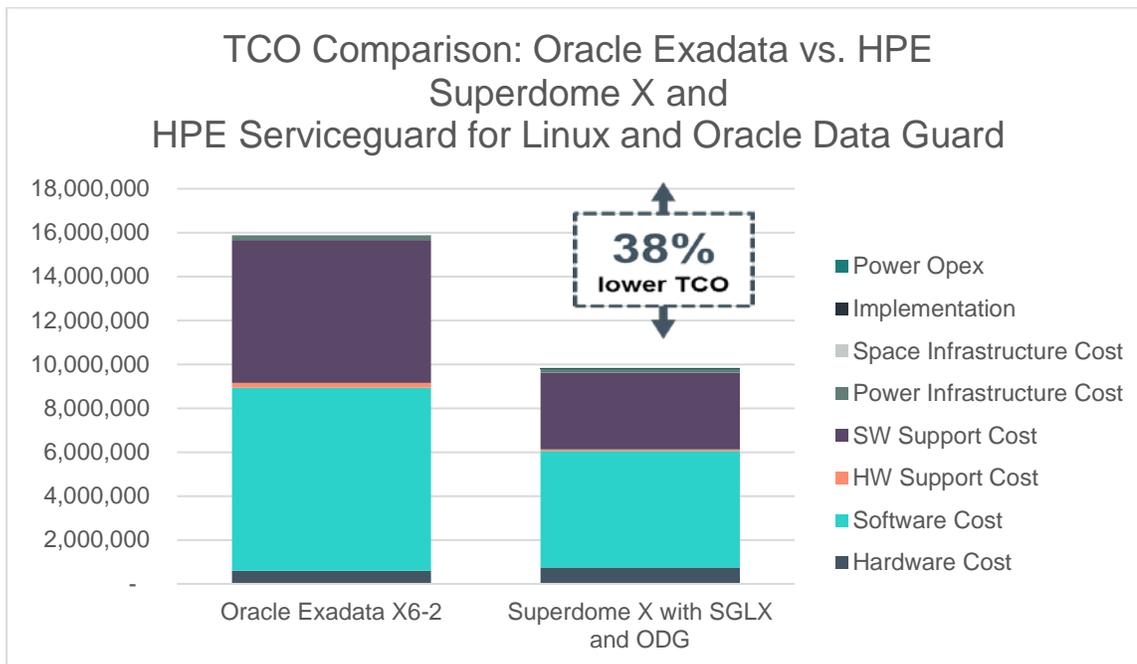


Figure 6. TCO comparison between Oracle Exadata solution and HPE Superdome X solution with SGLX and ODG

	Oracle Exadata X6-2	Superdome X with SGLX and ODG
■ Power Opex	23,885	20,827
■ Implementation	-	13,690
■ Space Infrastructure Cost	2,767	2,767
■ Power Infrastructure Cost	199,537	173,985
■ SW Support Cost	6,504,480	3,492,734
■ HW Support Cost	216,720	71,676
■ Software Cost	8,336,000	5,312,542
■ Hardware Cost	602,000	741,136

Figure 7. Component-wise cost break up used in TCO comparison between Oracle Exadata solution and HPE Superdome X solution with SGLX and ODG

<sup>1</sup> Compares an 8-socket HPE Superdome X server and HPE 3PAR StoreServ 8450 running RHEL, Oracle DB and HPE Serviceguard for Linux versus an Oracle Exadata X6-2 Half Rack running Exadata software, Oracle DB and Oracle RAC. Analysis based on list price. Power costs based on public data. Source: Based on Hewlett Packard Enterprise internal analysis results using publicly available competitive data, February 2017.

## Workload description

HammerDB is an open-source benchmark tool that can test the performance of many leading databases including Oracle Database 12c, which is used in this Reference Architecture. While the benchmark can simulate many types of workloads, the TPC-C-like workload for OLTP database simulation was used for this testing to represent mission-critical workloads that require maximum uptime. The configuration in this Reference Architecture uses HammerDB data to show the relative performance impact of installing Oracle Data Guard in your environment.

## Summary

Getting the most out of any server environment requires understanding the unique solution requirements so that an appropriate balance can be struck between performance and protection. A configuration that delivers the highest possible performance may not be the best option when SLAs require minimal downtime in the event of a failure. This testing presented two configuration options. The first option was designed to optimize performance, but the configuration choices required to achieve that high level of performance lengthened recovery time after a simulated failure. The second option was designed to deliver the shortest possible recovery time, but this came at the cost of performance. There is frequently a tradeoff between performance and recovery time, so it is important to weigh the relative importance of each in light of the overall goals and requirements for the workloads.

## 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](http://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](http://hpe.com/us/en/services/consulting.html).

Table 1. Bill of materials

Qty	Part Number	Description
<b>Rack and network infrastructure</b>		
1	H6J66A	HPE 42U 600x1075mm Advanced Shock Rack
2	H6J85A	HPE Rack Hardware Kit
1	BW906A	HPE 42U 1075mm Side Panel Kit
48	QK734A	HPE Premier Flex LC/LC OM4 2f 5m Cbl
1	JC105B	HPE 5800 48 port Switch
1	JH179A	HPE 5930 4-slot Switch
<b>HPE Integrity Superdome X</b>		
1	AT147A	HPE Superdome X Base Enclosure
8	AOR98A	HPE BL920s Gen9 2.5GHz 36c Svr Blade
96	H7B38A	HPE DDR4 128GB (4x32GB) Mem Module
8	BD505A	HPE iLO Adv incl 3yr TSU 1-Svr Lic
2	711307-B21	HPE 6125XLG Blade Switch
4	C8S47A	Brocade 16Gb/28c PP+ Embedded SAN Switch
8	700065-B21	HPE FlexFabric 20Gb 2P 630FLB Adptr
16	710608-B21	HPE QMH2672 16Gb FC HBA
48	AJ716B	HPE 8Gb Short Wave B-Series SFP+ 1 Pack
1	AT152A	HPE Superdome X Advanced Par LTU

Qty	Part Number	Description
<b>Storage</b>		
1	BW904A	HPE 42U 600X1075mm Enterprise Shock Rack
1	H6Z25A	HPE 3PAR StoreServ 8450 4N Stor Cent Base
16	K2Q95A	HPE 3PAR 8000 480GB SFF SSD
1	L7C17A	HPE 3PAR 8450 OS Suite Base LTU
80	L7C18A	HPE 3PAR 8450 OS Suite Drive LTU
8	H6Z26A	HPE 3PAR 8000 SFF(2.5in) SAS Drive Encl
64	K2Q95A	HPE 3PAR 8000 480GB SFF SSD
1	TK808A	HPE Rack Front Door Cover Kit
96	QK735A	HPE Premier Flex LC/LC OM4 2f 15m Cbl
8	QK734A	HPE Premier Flex LC/LC OM4 2f 5m Cbl
4	H5M58A	HPE Basic 4.9kVA/L6-30P/C13/NA/J PDU
1	BW906A	HPE 42U 1075mm Side Panel Kit
1	BD362A	HPE 3PAR StoreServ Mgmt/Core SW Media
1	TC472A	HPE Intelligent Inft Analyzer SW v2 LTU
1	HOJDOA1	HPE 3PAR Storserv Onl Imp QuickStart SVC
96	QK724A	HPE B-series 16Gb SFP+SW XCVR
<b>HPE Serviceguard for Linux</b>		
16	BB097ACE	HPE SGLX Enterprise edition x86 1y 24x7 PSL E-LTU

## Appendix B: Linux kernel configuration

The configuration for the Linux kernel as tested is shown below.

```
fs.file-max = 6815744
fs.aio-max-nr = 1048576

kernel.sem = 250 32000 100 128
kernel.shmmni = 4096

kernel.shmall = 1543503872
kernel.shmmax = 6322191859712

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 = 1048576

#
vm.hugetlb_shm_group = 54322
vm.nr_hugepages = 2713369

## parameters from HPE SDX RHEL report
kernel.numa_balancing = 0

## recommendations from red hat, 29 april 2016
vm.swappiness = 0
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
```

```
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
```

## Appendix C: Oracle configuration files

The Oracle pfiles were modified to include optimal settings for each configuration.

### HPE Serviceguard for Linux Oracle configuration

The following was the pfile file for the SGLX testing without ODG:

```
orclasm._data_transfer_cache_size=0
orclasm._db_cache_size=2979633561600
orclasm._java_pool_size=512m
orclasm._large_pool_size=42949672960
orclasm._oracle_base='/u02/app/oracle'#ORACLE_BASE set from environment
orclasm._pga_aggregate_target=4g
orclasm._shared_pool_size=60129542144
orclasm._streams_pool_size=0
*_ash_size=51539607552
*_bump_highwater_mark_count=10
*_disable_selftune_checkpointing=TRUE
*_disk_sector_size_override=TRUE
*_enable_NUMA_support=TRUE
*_fast_cursor_reexecute=TRUE
*_high_priority_processes='VKTM*|LG*'
*_in_memory_undo=TRUE
*_kgl_hot_object_copies=16
*_audit_file_dest='/u02/app/oracle/admin/orclasm/adump'
*_audit_trail='NONE'
*_compatible='12.1.0.2.0'
*_control_files='+DATA1/orclasm/control01.ctl','+FRA1/orclasm/control02.ctl'
*_db_block_size=8192
*_db_cache_size=5368709120000
*_db_create_file_dest='+DATA1'
*_db_create_online_log_dest_1='+FRA1'
*_db_domain=''
*_db_file_multiblock_read_count=32
*_db_flashback_retention_target=120
*_db_name='orclasm'
*_db_recovery_file_dest='+FRA1'
*_db_recovery_file_dest_size=2199023255552
*_db_unrecoverable_scn_tracking=FALSE
*_diagnostic_dest='/u02/app/oracle'
*_dispatchers='{PROTOCOL=TCP} (SERVICE=orclasmXDB)'
*_enable_pluggable_database=true
*_filesystemio_options='setall'
*_large_pool_size=85899345920
*_local_listener='LISTENER_ORCLASM','ORCLASM_0n1n4n5','ORCLASM_2n3n6n7'
*_log_checkpoint_timeout=0
*_log_checkpoints_to_alert=TRUE
*_open_cursors=10000
*_pga_aggregate_target=9663676416
*_processes=6000
*_recovery_parallelism=288
*_remote_login_passwordfile='EXCLUSIVE'
*_sga_max_size=5690831667200
*_shared_pool_size=107374182400
*_transactions_per_rollback_segment=1
*_undo_retention=1
```

```
*.undo_tablespace='UNDOTBS1'
*.use_large_pages='ONLY'
```

## HPE Serviceguard for Linux with Oracle Data Guard configuration

The following was the pfile file for the SGLX with ODG testing:

```
orcl...oracle_base='/u02/app/oracle'#ORACLE_BASE set from environment
*_ash_size=51539607552
*_bump_highwater_mark_count=10
*_disable_selftune_checkpointing=TRUE
*_disk_sector_size_override=TRUE
*_enable_NUMA_support=TRUE
*_fast_cursor_reexecute=TRUE
*_high_priority_processes='VKTML|LG*'
*_in_memory_undo=TRUE
*_kgl_hot_object_copies=16
*.archive_lag_target=0
*.audit_file_dest='/u02/app/oracle/admin/orcl/adump'
*.audit_trail='NONE'
*.compatible='12.1.0.2'
*.control_files='+DATA/orcl/control01.ctl','+DATA/orcl/control02.ctl'#Restore Controlfile
*.db_block_size=8192
*.db_cache_size=5368709120000
*.db_create_file_dest='+DATA'
*.db_create_online_log_dest_1='+FRA'
*.db_domain=''
*.db_file_multiblock_read_count=32
*.db_file_name_convert='+DATA/ORCL_STBY','+DATA/ORCL','+FRA/ORCL_STBY','+FRA/ORCL'
*.db_flashback_retention_target=120
*.db_name='orcl'
*.db_recovery_file_dest='+FRA'
*.db_recovery_file_dest_size=2199023255552
*.db_unique_name='orcl'
*.db_unrecoverable_scn_tracking=FALSE
*.dg_broker_start=TRUE
*.diagnostic_dest='/u02/app/oracle'
*.dispatchers='{PROTOCOL=TCP} (SERVICE=orclXDB)'
*.enable_pluggable_database=TRUE
*.fal_server='orcl_stby'
*.filesystemio_options='setall'
*.large_pool_size=85899345920
*.local_listener='LISTENER_ORCL','ORCL_0n1n4n5','ORCL_2n3n6n7','ODG_0'
*.log_archive_config='dg_config=(orcl_stby)'
*.log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST','valid_for=(ALL_LOGFILES, ALL_ROLES)'
*.log_archive_dest_2='service="orcl_stby"', 'SYNC NOAFFIRM delay=0 optional compression=disable max_failure=0
max_connections=1 reopen=300 db_unique_name="orcl_stby" net_timeout=30','valid_for=(online_logfile,all_roles)'
*.log_archive_dest_state_2='ENABLE'
orcl_stby.log_archive_format='%t-%s_%r.dbf'
orcl.log_archive_format='%t-%s_%r.dbf'
*.log_archive_max_processes=4
*.log_archive_min_succeed_dest=1
orcl_stby.log_archive_trace=0
orcl.log_archive_trace=0
*.log_checkpoint_timeout=0
*.log_checkpoints_to_alert=TRUE
*.log_file_name_convert='+DATA/ORCL_STBY','+DATA/ORCL','+FRA/ORCL_STBY','+FRA/ORCL'
*.open_cursors=10000
*.pga_aggregate_target=9663676416
*.processes=6000
```

```
*.recovery_parallelism=288
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_max_size=5690831667200
*.shared_pool_size=107374182400
*.standby_file_management='AUTO'
*.transactions_per_rollback_segment=1
*.undo_retention=1
*.undo_tablespace='UNDOTBS1'
*.use_large_pages='ONLY'
```

## Appendix D: HammerDB configuration files

Below are relevant sections of the HammerDB configuration files for the Master and Slave. The configuration files are from the performance testing and have 144 virtual users listed. Because 72 virtual users were used in the recovery testing, the virtual user setting was changed to 72 in the configuration files for the recovery scenario. Note that HammerDB by default includes configuration information for multiple database vendors, which are not included below.

### Master database configuration file

```
<?xml version="1.0" encoding="utf-8"?>
<hammerdb>
<look_and_feel>
<ttk_theme>
    <theme>classic</theme>
</ttk_theme>
<code_highlight>
    <highlight>>true</highlight>
</code_highlight>
</look_and_feel>
<benchmark>
    <rdbsms>Oracle</rdbsms>
    <bm>TPC-C</bm>
</benchmark>
<oracle>
    <service>
        <system_user>system</system_user>
        <system_password>Password1</system_password>
        <instance>p01</instance>
    </service>
    <tpcc>
        <schema>
            <count_ware>2500</count_ware>
            <num_threads>25</num_threads>
            <tpcc_user>tpcc</tpcc_user>
            <tpcc_pass>tpcc</tpcc_pass>
            <tpcc_def_tab>tpcc</tpcc_def_tab>
            <tpcc_ol_tab>tpcc_ol</tpcc_ol_tab>
            <tpcc_def_temp>temp</tpcc_def_temp>
            <plsql>0</plsql>
            <directory>
            </directory>
            <partition>true</partition>
            <tpcc_tt_compat>>false</tpcc_tt_compat>
        </schema>
    </tpcc>
</oracle>
```

```

        <total_iterations>100000000</total_iterations>
        <raiseerror>>false</raiseerror>
        <keyandthink>>false</keyandthink>
        <checkpoint>>false</checkpoint>
        <oradriver>awr</oradriver>
        <rampup>3</rampup>
        <duration>5</duration>
    </driver>
</tpcc>
<tpch>
    <schema>
        <scale_fact>1</scale_fact>
        <tpch_user>tpch</tpch_user>
        <tpch_pass>tpch</tpch_pass>
        <tpch_def_tab>tpchtab</tpch_def_tab>
        <tpch_def_temp>temp</tpch_def_temp>
        <num_tpch_threads>1</num_tpch_threads>
        <tpch_tt_compat>>false</tpch_tt_compat>
    </schema>
    <driver>
        <total_querysets>1</total_querysets>
        <raise_query_error>>false</raise_query_error>
        <verbose>>false</verbose>
        <degree_of_parallel>2</degree_of_parallel>
        <refresh_on>>false</refresh_on>
        <update_sets>1</update_sets>
        <trickle_refresh>1000</trickle_refresh>
        <refresh_verbose>>false</refresh_verbose>
    </driver>
</tpch>
<transaction_counter>
    <connect_string>system/Password1@p01</connect_string>
    <refresh_rate>10</refresh_rate>
    <rac>0</rac>
    <autorange>0</autorange>
</transaction_counter>
</oracle>
<virtual_user_options>
    <virtual_users>144</virtual_users>
    <user_delay>20</user_delay>
    <repeat_delay>20</repeat_delay>
    <iterations>1</iterations>
    <show_output>1</show_output>
    <log_to_temp>1</log_to_temp>
    <unique_log_name>1</unique_log_name>
    <no_log_buffer>0</no_log_buffer>
</virtual_user_options>
<mode>
    <autopilot_options>
        <apmode>disabled</apmode>
        <autopilot_duration>10</autopilot_duration>
        <autopilot_sequence>2 3 5 9 13 17 21 25</autopilot_sequence>
    </autopilot_options>
    <slave_options>

```

```

        <hostname>localhost</hostname>
        <id>0</id>
    </slave_options>
</mode>
<metrics>
    <metric_options>
        <agent_hostname>localhost</agent_hostname>
        <agent_id>0</agent_id>
    </metric_options>
</metrics>
</hammerdb>

```

### Slave database configuration file

```

<?xml version="1.0" encoding="utf-8"?>
<hammerdb>
<look_and_feel>
<ttk_theme>
    <theme>classic</theme>
</ttk_theme>
<code_highlight>
    <highlight>>true</highlight>
</code_highlight>
</look_and_feel>
<benchmark>
    <rdbms>Oracle</rdbms>
    <bm>TPC-C</bm>
</benchmark>
<oracle>
    <service>
        <system_user>system</system_user>
        <system_password>Password1</system_password>
        <instance>p02</instance>
    </service>
    <tpcc>
        <schema>
            <count_ware>2500</count_ware>
            <num_threads>25</num_threads>
            <tpcc_user>tpcc</tpcc_user>
            <tpcc_pass>tpcc</tpcc_pass>
            <tpcc_def_tab>tpcc</tpcc_def_tab>
            <tpcc_ol_tab>tpcc_ol</tpcc_ol_tab>
            <tpcc_def_temp>temp</tpcc_def_temp>
            <plsql>0</plsql>
            <directory>
            </directory>
            <partition>true</partition>
            <tpcc_tt_compat>false</tpcc_tt_compat>
        </schema>
    </tpcc>
    <driver>
        <total_iterations>100000000</total_iterations>
        <raiseerror>false</raiseerror>
        <keyandthink>false</keyandthink>
        <checkpoint>false</checkpoint>
        <oradriver>standard</oradriver>
    </driver>
</oracle>
</hammerdb>

```

```

        <rampup>3</rampup>
        <duration>5</duration>
    </driver>
</tpcc>
<tpch>
    <schema>
        <scale_fact>1</scale_fact>
        <tpch_user>tpch</tpch_user>
        <tpch_pass>tpch</tpch_pass>
        <tpch_def_tab>tpchtab</tpch_def_tab>
        <tpch_def_temp>temp</tpch_def_temp>
        <num_tpch_threads>1</num_tpch_threads>
        <tpch_tt_compat>>false</tpch_tt_compat>
    </schema>
    <driver>
        <total_querysets>1</total_querysets>
        <raise_query_error>>false</raise_query_error>
        <verbose>>false</verbose>
        <degree_of_parallel>2</degree_of_parallel>
        <refresh_on>>false</refresh_on>
        <update_sets>1</update_sets>
        <trickle_refresh>1000</trickle_refresh>
        <refresh_verbose>>false</refresh_verbose>
    </driver>
</tpch>
<transaction_counter>
    <connect_string>system/Password1@p02</connect_string>
    <refresh_rate>10</refresh_rate>
    <rac>0</rac>
    <autorange>0</autorange>
</transaction_counter>
</oracle>
<virtual_user_options>
    <virtual_users>144</virtual_users>
    <user_delay>20</user_delay>
    <repeat_delay>20</repeat_delay>
    <iterations>1</iterations>
    <show_output>0</show_output>
    <log_to_temp>0</log_to_temp>
    <unique_log_name>0</unique_log_name>
    <no_log_buffer>0</no_log_buffer>
</virtual_user_options>
<mode>
    <autopilot_options>
        <apmode>disabled</apmode>
        <autopilot_duration>10</autopilot_duration>
        <autopilot_sequence>2 3 5 9 13 17 21 25</autopilot_sequence>
    </autopilot_options>
    <slave_options>
        <hostname>localhost</hostname>
        <id>0</id>
    </slave_options>
</mode>
<metrics>

```

```

    <metric_options>
      <agent_hostname>localhost</agent_hostname>
      <agent_id>0</agent_id>
    </metric_options>
  </metrics>
</hammerdb>

```

## Appendix E: Listener configuration

The Oracle listeners and TNS names files were updated with the details of the additional listeners as follows for the testing.

### HPE Serviceguard for Linux Oracle configuration

**The following was the listeners.ora file for the SGLX testing without ODG:**

```

ORCLASM_0n1n4n5 =
  {ADDRESS = {PROTOCOL = TCP}{HOST = node1.localdomain}{PORT = 1611}}

ORCLASM_2n3n6n7 =
  {ADDRESS = {PROTOCOL = TCP}{HOST = node1.localdomain}{PORT = 1621}}

```

**The following was the tnsnames.ora file for the SGLX testing without ODG:**

```

ORCLASM =
  {DESCRIPTION =
    {ADDRESS = {PROTOCOL = TCP}{HOST = node1.localdomain}{PORT = 1539}}
    {CONNECT_DATA =
      {SERVER = DEDICATED}
      {SERVICE_NAME = orclasm}
    }
  }

LISTENER_ORCLASM =
  {ADDRESS = {PROTOCOL = TCP}{HOST = node1.localdomain}{PORT = 1539}}

ORCLASM_0n1n4n5 =
  {ADDRESS = {PROTOCOL = TCP}{HOST = 192.168.11.10}{PORT = 1611}}

ORCLASM_2n3n6n7 =
  {ADDRESS = {PROTOCOL = TCP}{HOST = 192.168.21.10}{PORT = 1621}}

```

### HPE Serviceguard for Linux with Oracle Data Guard configuration

**The following was the listeners.ora file for the SGLX with ODG testing:**

```

LISTENER =
  {DESCRIPTION_LIST =
    {DESCRIPTION =
      {ADDRESS = {PROTOCOL = TCP}{HOST = node2.localdomain}{PORT = 1521}}
      {ADDRESS = {PROTOCOL = IPC}{KEY = EXTPROC1521}}
      {UR=A}
    }
  }

ODG_0 =
  {DESCRIPTION_LIST =
    {DESCRIPTION =
      {ADDRESS = {PROTOCOL = TCP}{HOST = node2.localdomain}{PORT = 1600}}
      {SEND_BUF_SIZE=104857600}
    }
  }

```

```

        {RECV_BUF_SIZE=104857600}}
    {UR=A}
}
]
]

SID_LIST_LISTENER =
{SID_LIST =
  {SID_DESC =
    {SDU=65535}
    {GLOBAL_DBNAME = orcl_DGMGRL}
    {ORACLE_HOME = /u02/app/oracle/product/12.1.0/db_1}
    {SID_NAME = orcl}
    {UR=A}
  }
}
]

ORCL_0n1n4n5 =
{ADDRESS = {PROTOCOL = TCP}{HOST = node2.localdomain}{PORT = 1611}}

ORCL_2n3n6n7 =
{ADDRESS = {PROTOCOL = TCP}{HOST = node2.localdomain}{PORT = 1621}}

ADR_BASE_LISTENER = /u02/app/oracle

```

**The following was the tnsnames.ora file for the SGLX with ODG testing.**

```

LISTENER_ORCL =
{ADDRESS = {PROTOCOL = TCP}{HOST = node2.localdomain}{PORT = 1521}}

orcl =
{DESCRIPTION =
  {ADDRESS_LIST =
    {ADDRESS = {PROTOCOL = TCP}{HOST = orcl.localdomain}{PORT = 1600}}
  }
  {CONNECT_DATA =
    {SID = orcl}
  }
  {SDU=65535}
  {SEND_BUF_SIZE=104857600}
  {RECV_BUF_SIZE=104857600}
}

orcl_stby,ODG_0 =
{DESCRIPTION =
  {ADDRESS_LIST =
    {ADDRESS = {PROTOCOL = TCP}{HOST = orcl_stby.localdomain}{PORT = 1600}}
  }
  {CONNECT_DATA =
    {SID = orcl}
  }
  {SDU=65535}
  {SEND_BUF_SIZE=104857600}
  {RECV_BUF_SIZE=104857600}
}

```

```
ORCL_0n1n4n5 =
{ADDRESS = {PROTOCOL = TCP}{HOST = node2.localdomain}{PORT = 1611}}
```

```
ORCL_2n3n6n7 =
{ADDRESS = {PROTOCOL = TCP}{HOST = node2.localdomain}{PORT = 1621}}
```

## Appendix F: HPE SGLX Enterprise Edition A.12.00.50 and code customizations

This Reference Architecture describes testing completed with HPE SGLX Enterprise Edition A.12.00.50 and code customizations as shown below to enable HPE ATX for Oracle with SGLX. Since the original publication of this RA, the code customizations have been productized in HPE SGLX A.12.00.51.

### Oracle startup process

In the performance configuration, the HPE-ATX tool was used to start the Oracle listeners. We modified the halistener.mon file to invoke the HPE-ATX tool with the following changes:

```

${ORACLE_HOME}/bin/lsnrctl start ${LISTENER_NAME[$1]}
---
NODE_LIST=\`echo ${LISTENER_NAME[$1]} | awk -F_ '{print \2}' | sed 's/n/,/g'\`
echo "INSTANCE: ${LISTENER_NAME[$1]} NODE_LIST: \${NODE_LIST}" >
/tmp/lsnrctl_restart_${LISTENER_NAME[$1]}.txt
if [[ "\${NODE_LIST}" -ne "" ]]; then
hpe-atx -p ff_flat -t none -n \${NODE_LIST} -l
${ORACLE_HOME}/log/atx_lsnrctl_restart_${LISTENER_NAME[$1]}.log ${ORACLE_HOME}/bin/lsnrctl start
${LISTENER_NAME[$1]}
else
${ORACLE_HOME}/bin/lsnrctl start ${LISTENER_NAME[$1]}
fi

```

### Oracle database startup

In the performance configuration, the HPE-ATX tool was also used to start the Oracle database on the primary nPar. We modified the haoracle\_sql.sh with the following changes:

```

${ORACLE_HOME}/bin/lsnrctl start \${I}
---
NODE_LIST=\`echo \${I} | awk -F_ '{print \2}' | sed 's/n/,/g'\`
echo "INSTANCE: \${I} NODE_LIST: \${NODE_LIST}" > /tmp/lsnrctl_start_\${I}.txt
if [[ "\${NODE_LIST}" -ne "" ]]; then
hpe-atx -p ff_flat -t none -n \${NODE_LIST} -l ${ORACLE_HOME}/log/atx_lsnrctl_start_\${I}.log
${ORACLE_HOME}/bin/lsnrctl start \${I}
else
${ORACLE_HOME}/bin/lsnrctl start \${I}
fi
186c192
echo "startup \$START_MODE \${command};" | ${ORACLE_HOME}/bin/sqlplus '/ as sysdba'
---
hpe-atx -p rr_tree -l ${ORACLE_HOME}/log/atx_srvctl_start_db.log ${ORACLE_HOME}/bin/srvctl start
database -db ${ORACLE_SID_NAME} -o $START_MODE

```

As with the listener startup, the HPE-ATX tool was used only for the performance testing. For recovery testing the haoracle\_sql.sh that ships with SGLX was used.

## Resources and additional links

HPE Integrity Superdome X QuickSpecs, <http://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=c04383189>

HPE 3PAR StoreServ 8000 QuickSpecs, <http://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=c04607918>

HPE Serviceguard for Linux 12.00.50 Release Notes,

[http://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/doc/other/ha\\_cluster/pdfs/sg\\_120050\\_rn\\_201606.pdf](http://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/doc/other/ha_cluster/pdfs/sg_120050_rn_201606.pdf)

Managing HPE Serviceguard for Linux A.12.00.50,

[https://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/doc/other/ha\\_cluster/pdfs/m\\_sg\\_a120050\\_201607.pdf](https://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/doc/other/ha_cluster/pdfs/m_sg_a120050_201607.pdf)

HPE Serviceguard Toolkit for Oracle Data Guard on Linux User Guide A.12.00.50,

<http://h20564.www2.hpe.com/portal/site/hpsc/public/kb/docDisplay/?docId=c05207751>

HPE Serviceguard Toolkit for Oracle on Linux User Guide,

[http://h20566.www2.hpe.com/hpsc/doc/public/display?sp4ts.oid=376220&docLocale=en\\_US&docId=emr\\_na-c05176434](http://h20566.www2.hpe.com/hpsc/doc/public/display?sp4ts.oid=376220&docLocale=en_US&docId=emr_na-c05176434)

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

Oracle Database High Availability Best Practices, [https://docs.oracle.com/database/121/HABPT/config\\_storage.htm#HABPT4790](https://docs.oracle.com/database/121/HABPT/config_storage.htm#HABPT4790)

Oracle Database 12c installation guide for Linux, <https://docs.oracle.com/database/121/LADBI/toc.htm>

Optimizing Linux performance on HPE Integrity Superdome X, <http://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=4AA5-9698ENW>

How to calculate suitable LUN queue depths: Host I/O queues and HPE 3PAR StoreServ Storage,

<http://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=4AA4-5094ENW>

To help us improve our documents, please provide feedback at [hpe.com/contact/feedback](http://hpe.com/contact/feedback).



Sign up for updates



© Copyright 2017 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries. Intel, Xeon, and Intel Xeon, are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

4AA6-7931ENW, Rev. 2, March 2017