WHITE PAPER — AUGUST 2018

ACCELERATING ORACLE PERFORMANCE USING VSPHERE PERSISTENT MEMORY (PMEM)

Reference Architecture

vmware®

Table of Contents

Table of Contents, continued

Table of Contents, continued

Executive Summary

Business Case

Customers have successfully run their business-critical Oracle workloads with high-performance demands on VMware vSphere® for many years.

Deploying IO-intensive Oracle workloads requires fast storage performance with low latency and resiliency from database failures. Latency, which is a measurement of response time, directly impacts a technology's ability to deliver faster performance for business-critical applications.

There has been a disruptive paradigm shift in data storage called Persistent Memory (PMEM) that resides between DRAM and disk storage in the data storage hierarchy. The technology enables byte-addressable updates and does not lose data if power is lost. Instead of having nonvolatile storage at the bottom with the largest capacity but the slowest performance, nonvolatile storage is now very close to DRAM in terms of performance.

PMEM is a byte-addressable form of computer memory that has the following characteristics:

- DRAM-like latency and bandwidth
- Regular load/store CPU instructions
- Paged/mapped by operating system just like DRAM
- Data is persistent across reboots

PMEM falls between the two ends of the spectrum, DRAM and Flash Storage. DRAM is expensive and volatile, and Flash is cheaper, slower, and persistent.

VMware vSphere 6.7 brings a lot of great new features and innovations—especially vSphere Persistent Memory (PMEM) which aids business-critical Oracle workloads, offering both enhanced performance and faster recovery.

Solution Overview

With the release of vSphere 6.7, PMEM or NVDIMMs are now supported and can be used for the host and or a VM. Now applications, whether modified to use NVDIMMs or legacy VMs, can take advantage of PMEM on VMware vSphere.

When NVDIMM modules are installed in supported hardware along vSphere 6.7, a PMEM datastore is automatically created on the host. That datastore is managed by the Virtual Center and DRS, no action is required to manage.

This paper examines the performance of Oracle databases using VMware PMEM in different modes for redo log-enhanced performance, accelerating flash cache performance and a possibility of reducing Oracle licenses.

Key Results

The following highlights validate performance of Oracle databases using VMware PMEM in different modes:

- Improved performance of Oracle Redo Log using vPMEMDisk-backed vmdks/vPMEM disks in DAX mode
- Accelerating performance using Oracle Smart Flash Cache
- Potential reduction in Oracle Licensing

Introduction

Purpose

This reference architecture validates the ability of vSphere 6.7 PMEM functionality to provide enhanced performance of Oracle databases using VMware PMEM in different modes for redo log performance, accelerating flash cache performance and a possibility of reducing Oracle licenses.

Scope

This reference architecture covers the following use cases for Oracle workloads with D vSphere 6.7 PMEM functionality:

- Improved performance of Oracle Redo Log using vPMEMDisk-backed vmdks/vPMEM disks in DAX mode
- Accelerating Performance using Oracle Smart Flash Cache
- Potential reduction in Oracle Licensing

Audience

This reference architecture is intended for Oracle Database Administrators and Virtualization and Storage Architects involved in planning, architecting, and administering a workload intensive Oracle environment on VMware SDCC platform.

Terminology

This paper includes the following terminology.

TERM	DEFINITION
Oracle Single Instance	Oracle Single-Instance database consists of a set of memory structures, background processes, and physical database files that serve the database users.
Oracle Automatic Storage Management (Oracle ASM)	Oracle ASM is a volume manager and a file system for Oracle database files that support Oracle Single-Instance database and Oracle RAC configurations.

Table 1. Terminology

Technology Overview

Overview

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

- VMware vSphere
- VMware vSAN™
- VMware SDDC
- VMware Cloud on AWS
- Micron Technology
- vSphere 6.7 Persistent Memory Modes
- Oracle Database 18c
- Oracle Database Architecture
- Oracle Multitenant Architecture
- Oracle Automatic Storage Management
- Oracle ASMLIB and ASMFD
- Linux Device Persistence and udev Rules
- Oracle Smart Flash Cache
- Oracle Automatic Workload Repository

VMware vSphere

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

vSphere 6.7 is the next-generation infrastructure for next-generation applications and focuses on simplifying management at scale, securing both infrastructure and workloads, being the universal platform for applications, and providing a seamless hybrid cloud experience. Features such as Enhanced Linked Mode with embedded Platform Services Controllers bring simplicity back to vCenter Server architecture. Support for TPM 2.0 and Virtualization Based Security provides organizations with a secure platform for both infrastructure and workloads. The addition of support for RDMA over Converged Ethernet v2 (RoCE v2), huge pages, suspend/resume for vGPU workloads, Persistent Memory, and native 4k disks shows that the hypervisor is not a commodity and that vSphere 6.7 enables more functionality and better performance for more applications.

More information about VMware vSphere new features can be found here.

VMware vSAN

VMware vSAN powers industry-leading Hyper-Converged Infrastructure solutions with a vSphere-native, high-performance architecture and helps organizations evolve their data center without risk, control IT costs, and scale to address tomorrow's business needs.

vSAN 6.7 delivers a new HCI experience architected for the hybrid cloud with operational efficiencies that reduce time to value through a new, intuitive user interface, and provides consistent application performance and availability through advanced self healing and proactive support insights. Seamless integration with VMware's complete software-defined data center (SDDC) stack and leading hybrid cloud offerings makes it the most complete platform for virtual machines—whether running business-critical databases, virtual desktops or next-generation applications.

More information about VMware vSAN 6.7 can be found here.

VMware SDDC

The mobile cloud era is changing line-of-business (LOB) expectations of IT. For IT organizations to securely deliver the anticipated improvements in service quality and speed, a Software-Defined Data Center (SDDC) approach is required. The VMware approach to the SDDC delivers a unified platform that supports any application and provides flexible control. The VMware architecture for the SDDC empowers companies to run hybrid clouds and to leverage unique capabilities to deliver key outcomes that enable efficiency, agility, and security. Enterprises using VMware technology have three ways to establish an SDDC and transition at their own pace: build their own using reference architectures, use a converged infrastructure, or use a hyper-converged infrastructure wherein the full SDDC is delivered already implemented on the customer's hardware of choice.

More information on VMware SDDC can be found here.

VMware Cloud on AWS

VMware Cloud on AWS is an on-demand service that enables customers to run applications across vSphere-based cloud environments with access to a broad range of AWS services. Powered by VMware Cloud Foundation™, this service integrates vSphere, vSAN, and NSX® along with VMware vCenter management, and is optimized to run on dedicated, elastic, bare-metal AWS infrastructure.

With VMware Hybrid Cloud Extension, customers can easily and rapidly perform large-scale, bi-directional migrations between on-premises and VMware Cloud on AWS environments.

With the same architecture and operational experience on-premises and in the cloud, IT teams can now quickly derive instant business value from use of the AWS and VMware hybrid cloud experience.

Micron Technology

Persistent Memory (PMEM) is a new product family that gives system architects an unprecedented choice for balancing system performance and total cost of ownership. Persistent Memory bridges the gap between DRAM and storage, allowing greater flexibility in data management by providing nonvolatile, low-latency memory closer to the processor. Because it resides on the DRAM bus, Persistent Memory can provide ultra-fast DRAM-like access to critical data. Combining the data reliability of traditional storage with ultra-low latency and high bandwidth opens up opportunities to optimize systems and manage data in a whole new way.

NVDIMMs are a nonvolatile Persistent Memory solution that combine NAND flash, DRAM and an optional power source into a single memory subsystem. They deliver DRAM-like latencies and can back up the data they handle, providing the ability to restore quickly if power is interrupted. NVDIMMs operate in the DRAM memory slots of servers to handle critical data at DRAM speeds. In the event of a power fail or system crash, an onboard controller safely transfers data stored in DRAM to the onboard nonvolatile memory, thereby preserving data that would otherwise be lost. When the system stability is restored, the controller transfers the data from the NAND back to the DRAM, allowing the application to efficiently pick up where it left off.

The backup power source for Micron's NVDIMMs can be provided either by a tethered AgigA Tech® PowerGEM® ultracapacitor or by routing a persistent supply through the motherboard to the DRAM 12V pins. NVDIMMs provide performance and data security advantages for a wide range of enterprise-class server and storage applications.

Micron NVDIMM are designed for applications that are sensitive to down time and require high performance to enable frequent access to large data sets. Micron NVDIMMs combine the speed of DRAM, the persistent storage of NAND and an optional power source into a single memory subsystem that delivers increased system performance and reliability. NVDIMMs are ideal for big data analytics, storage appliances, RAID cache, in-memory databases, and online transaction processing.

vSphere 6.7 Persistent Memory

Persistent Memory (PMEM) is a type of non-volatile DRAM (NVDIMM) that has the speed of DRAM but retains contents through power cycles. It is a new layer that sits between NAND flash and DRAM, providing faster performance. It's also non-volatile unlink DRAM.

This brings incredible performance possibilities as NVDIMMs are equal to or near the speed of DRAM—almost 100 times faster than SSDs. By moving the data closer to where the analysis is done, the CPU can access the data as if it was in RAM with DRAM-like latencies.

With vSphere Persistent Memory (PMEM), customers using supported hardware servers, can get the benefits of ultra-high-speed storage at a price point closer to DRAM-like speeds at flash-like prices.

The following diagram shows the convergence of memory and storage.

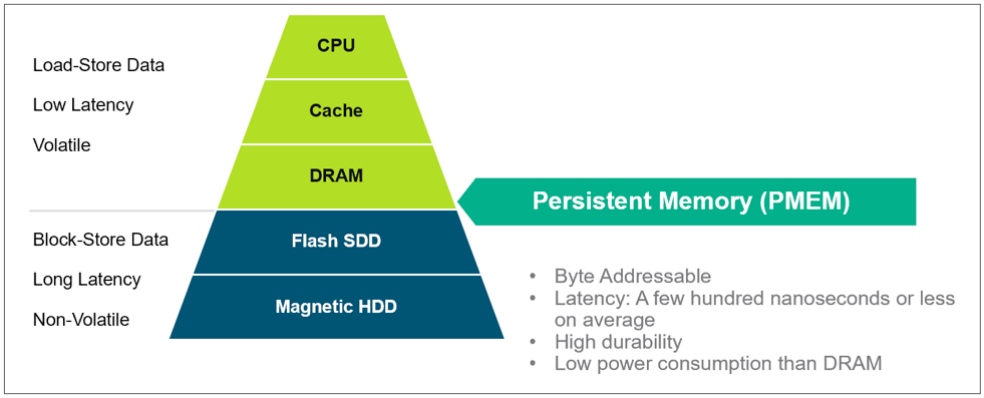

Figure 1. Convergence of Memory and Storage

Technology at the top of the pyramid (comprised of DRAM and the CPU cache and registers) have the shortest latency (best performance) but this comes at a higher cost relative to the items at the bottom of the pyramid. All of these components are accessed directly by the application—also known as load/storage access.

Technology at the bottom of the pyramid, represented by Magnetic media (HDDs and tape) and NAND flash (represented by SSDs and PCIe Workload Accelerators), have longer latency and lower costs relative to the technology at the top of the pyramid. These technology components have block access, meaning data is typically communicated in blocks of data and the applications are not accessed directly.

vSphere 6.7 supports two modes of accessing Persistent Memory:

- vPMEMDisk
  - o Presents NVDIMM capacity as a host local datastore
  - o Requires no guest operating system changes to leverage this technology
- vPMEM
  - o Exposes NVDIMM capacity to the virtual machine through a new virtual NVDIMM device
  - o Guest operating systems (GOS) use it directly as a block device or in DAX mode
  - o Within the GOS, we can configure NVDIMMs four distinct modes, each with their own advantages and disadvantages
    - » Raw: Presents as /dev/pmemN a block device
      - * Default mode configuration when first installing NVDIMMs into a system
      - * Supports filesystems with or without DAX (ext4, xfs)
      - * A raw pmem namespace does not support sector atomicity by default (See “sector” mode instead.)
    - » Sector: Presents as /dev/pmemNs a block device with sector atomicity
      - * Supports filesystem DAX (ext4, xfs) (recommended over raw mode)
      - * Requires storing extra “struct page” entries on regular system memory or Persistent Memory

    * map=mem: Regular system memory. Intended for small Persistent Memory capacities.
    * map=dev: Persistent Memory. Intended for large Persistent Memory capacities.
  » DAX: Presents as /dev/daxN.M a character device supporting DAX

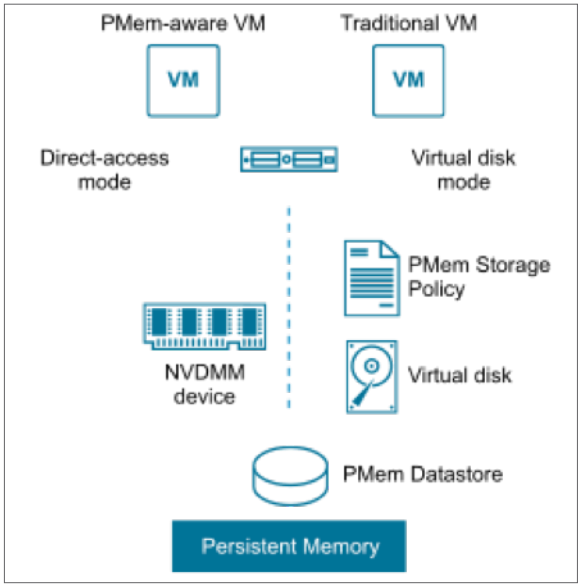

Figure 2. Traditional VM and PMEM-aware VM

When NVDIMM modules are installed in supported hardware and with vSphere 6.7, a PMEM datastore is automatically created on the host. That datastore is managed by the Virtual Center and DRS, no action is required to manage.

Considerations when using vPMEMDisk and vPMEM mode at the time of writing this paper:

No.	vPMEMDisk MODE	vPMEM MODE
1	Supported by all hardware versions, so no specific VM-compatible level needed	Virtural NVDIMM requires hardware version 14 or higher
2	Legacy guest OS may use the storage, no additional guest operating system support needed.	The OS must also support the use of PMEM, for example, Windows Server 2016 and Enterprise RedHat 7.4 or later.
3	No application-level support needed.	Application-level support needed to use vPMEM in some modes e.g., memory mode with DAX option
4	Hot add of vmdk on vPMEMDisk-backed datastore is currently not supported (trying to do so results in an error "Hot plug of device 'x' is not supported for this virtual machine").	Hot add of virtual NVDIMM to a VM is not supported (trying to do so results in an error "Devices of this type cannot be added while this virtual machine is powererd on").
5	Can use migrate workflows, and can be moved to or from a regular datastore	It always uses Persistent Memory and cannot be moved to a regular datastore.
6	Can be moved between hosts (Live vMotion possible without hardware NVDIMMs)	They can be migrated to another host (Live vMotion) only if that host also contains PMEM.
7	Snapshots are currently unsupported.	Snapshots are currently unsupported.

Table 2. Modes and Considerations

To use NVDIMMs as non-volatile memory a supported OS is needed. NVDIMMs are currently enabled in these distro releases:

- SLES 12 SP2, SP3
- RHEL 7.3, 7.4
- Fedora 24, 25, 26

More information on vSphere 6.7 Persistent Memory Modes can be found here and here.

Oracle Database 18c

Oracle Database 18c, the latest generation of the world's most popular database, provides businesses of all sizes with access to the world's fastest, most scalable and reliable database technology for secure and cost-effective deployment of transactional and analytical workloads in the cloud, on-premises, and hybrid cloud configurations.

Oracle Database 18c brings new functionality and improvements to features, including:

- Multitenant Architecture for massive cost savings and agility
- In-Memory Column Store for massive performance gains for real-time analytics
- Native Database Sharding for high availability of massive web applications
- Many more critical capabilities for enhanced database performance, availability, security, analytics, and application development

More information on Oracle Release 18c can be found here.

ORACLE ASMFD HELPS PREVENT CORRUPTION IN ORACLE ASM DISKS AND FILES WITHIN THE DISK GROUP

- Oracle ASM Filter Driver (Oracle ASMFD) rejects write I/O requests that are not issued by Oracle software. This write filter helps to prevent users with administrative privileges from inadvertently overwriting Oracle ASM disks, thus preventing corruption in Oracle ASM disks and files within the disk group. For disk partitions, the area protected is the area on the disk managed by Oracle ASMFD, assuming the partition table is left untouched by the user.
- Oracle ASMFD simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.
- More information on Oracle ASMFD can be found here.

Oracle Database Architecture

An Oracle database server consists of a database and at least one database instance in case of a single instance database. In case of Real Application Cluster, an Oracle database will have more than one instance accessing the database.

- A Database is a set of files, located on disk, that store data. These files can exist independently of a database instance.
- An instance is a set of memory structures that manage database files. The instance consists of a shared memory area, called the system global area (SGA), and a set of background processes. An instance can exist independently of database files.

The physical database structures that comprises a database include:

- Data files: Every Oracle database has one or more physical data files, which contain all the database data. The data of logical database structures, such as tables and indexes, is physically stored in the data files.
- Control files: Every Oracle database has a control file. A control file contains metadata specifying the physical structure of the database, including the database name and the names and locations of the database files.
- Online redo log files: Every Oracle Database has an online redo log, which is a set of two or more online redo log files. An online redo log is made up of redo entries (also called redo log records), which record all changes made to data.
- Many other files including parameter files, archived redo files, backup files, and networking files are important to any oracle database operations

More information about Oracle Database Architecture can be found here.

Oracle Multitenant Architecture

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

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

All Oracle databases before Oracle Database 12c were non-CDBs.

More information about Oracle Multitenant Architecture can be found here.

Oracle Automatic Storage Management

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

Oracle ASM is Oracle's recommended storage management solution that provides an alternative to conventional volume managers, file systems, and raw devices.

Oracle ASM uses disk groups to store data files. An Oracle ASM disk group is a collection of disks that Oracle ASM manages as a unit. You can add or remove disks from a disk group while a database continues to access files from the disk group. See Overview of Oracle Automatic Storage Management for more information.

Oracle ASMLIB and ASMFD

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

The Oracle Automatic Storage Management library driver simplifies the configuration and management of block disk devices by eliminating the need to rebind block disk devices used with Oracle Automatic Storage Management (Oracle ASM) each time the system is restarted.

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

More information on Oracle ASMLib can be found here.

Linux Device Persistence and udev Rules

Device names in Linux are not guaranteed to be persistent across reboots and so a device, for instance /dev/sdb, can be renamed as /dev/sdc on next reboot.

For device persistence, Linux udev rules can be used which guarantees device persistence across reboot.

More information on configuring device persistence for Oracle storage can be found here.

Oracle Smart Flash Cache

Oracle Database 11g Release 2 introduced a new database feature: Database Smart Flash Cache. This feature allows customers to increase the effective size of the Oracle database buffer cache without adding more main memory to the system.

For transaction-based workloads, Oracle database blocks are normally loaded into a dedicated shared memory area in main memory called the System Global Area (SGA). Database Smart Flash Cache allows the database buffer cache to be expanded beyond the SGA in main memory to a second level cache on flash memory.

Online Transaction Processing (OLTP) environments benefit from this technology. It is appropriate for new Oracle Database deployments with IO-intensive workloads as well as existing environments with main memory constraints and IO-bound applications. Benefits can include both increased transaction throughput and improved application response times. It is also supported in Real Application Cluster (RAC) environments; Database Smart Flash Cache can be applied to individual RAC nodes as required.

At the time of writing this paper, Database Smart Flash Cache is only supported on databases running on the Solaris or Oracle Linux operating systems.

More information on Oracle Smart Flash Cache can be found here.

Oracle Automatic Workload Repository

Oracle Database automatically persists the cumulative and delta values for most of the statistics at all levels (except the session level) in the Automatic Workload Repository (AWR). This process is repeated on a regular time period and the results are captured in an AWR snapshot. The delta values captured by the snapshot represent the changes for each statistic over the time period.

AWR collects, processes, and maintains performance statistics for problem detection and self-tuning purposes. This gathered data is stored both in memory and in the database and is displayed in both reports and views.

The statistics collected and processed by AWR include:

- Object statistics that determine both access and usage statistics of database segments
- Time model statistics based on time usage for activities, displayed in the V$SYS_TIME_MODEL and V$SESS_TIME_MODEL views
- Some of the system and session statistics collected in the V$SYSSTAT and V$SESSTAT views
- SQL statements that are producing the highest load on the system, based on criteria such as elapsed time and CPU time
- Active Session History (ASH) statistics, representing the history of recent sessions activity

More information on Oracle AWR can be found here.

Solution Configuration

This section introduces the resources and configurations for the solution, including:

- Architecture diagram
- Hardware resources
- Persistent Memory configuration
- Software resources
- Network configuration
- VM and Oracle configuration

Architecture Diagram

As shown in Figure 1, the key designs for testing the Oracle workload performance using vSphere 6.7 Persistent Memory were:

- A 3-node vSphere 6.7 Cluster
- Each ESXi server had two sockets, 24 cores each with Hyperthreading and 384GB of RAM
- Each ESXi server had 12 16GB NVDIMMs (Model MTA18ASF2G72XF1Z-2G6V21AB)
- All ESXi servers had access to an All Flash Storage for block storage
- Two identical VMs were created for the use cases
- Each VM had 12 vCPUs and 64GB memory
- Oracle Linux 7.4/Red Hat Linux 7.4 operating system was used for database VMs
- Oracle database version was 12.2.0.1.0 with Oracle SGA set to 32GB and PGA set to 12GB

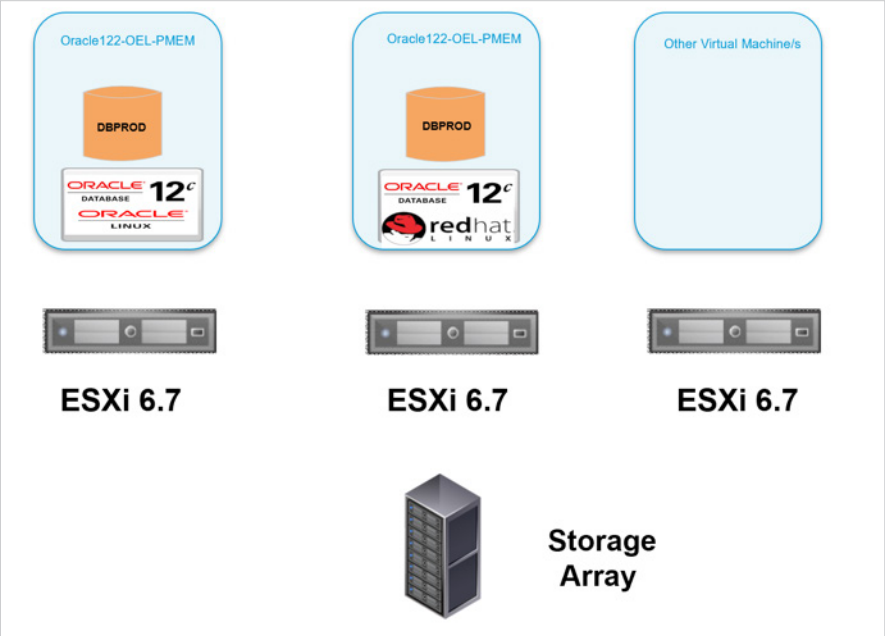

Figure 3. 3-node vSphere Cluster with Micron NVDIMM

Hardware Resources

Table 3 shows the hardware resources used in this solution.

DESCRIPTION	SPECIFICATION
Server	3 x ESXi Server
Server Model	Dell Inc. PowerEdge R740
CPU	2 sockets with 24 cores each, Intel® Xeon® Platinum 8168 CPU at 2.70GHz with hyper-threading enabled
RAM	384GB Samsung DDR4 RDIMMS
Persistent Memory	12 x 16GB Micron DDR4 NVDIMMS
Storage controller	1 x 12G SAS Modular RAID Controller
Disks	All Flash Array datastores with Emulex LightPulse LPe32000 PCIe HBA
Network	2 x Intel® Ethernet Controller X710 for 10GbE SFP+

Table 3. Hardware Resources

Figure 4 shows us the summary of one of the ESXi servers in the vSphere 6.7 cluster.

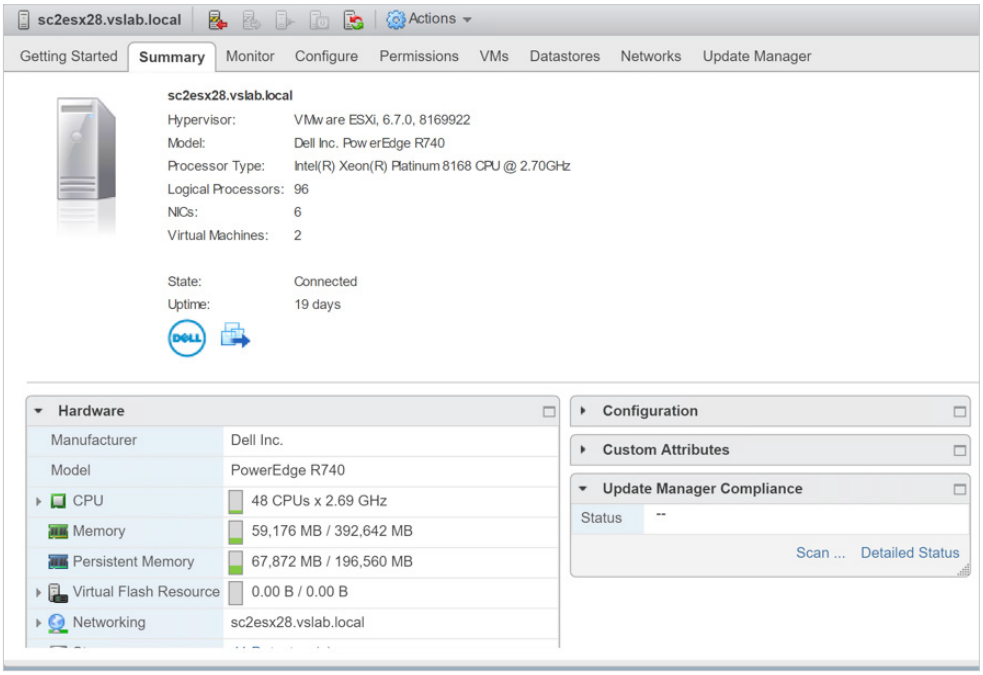

Figure 4. ESXi Servers Summary

Persistent Memory Configuration

PMEM datastore is not visible when logging into the Virtual Center via the web client. With vSphere 6.7, only one PMEM local datastore is allowed per host.

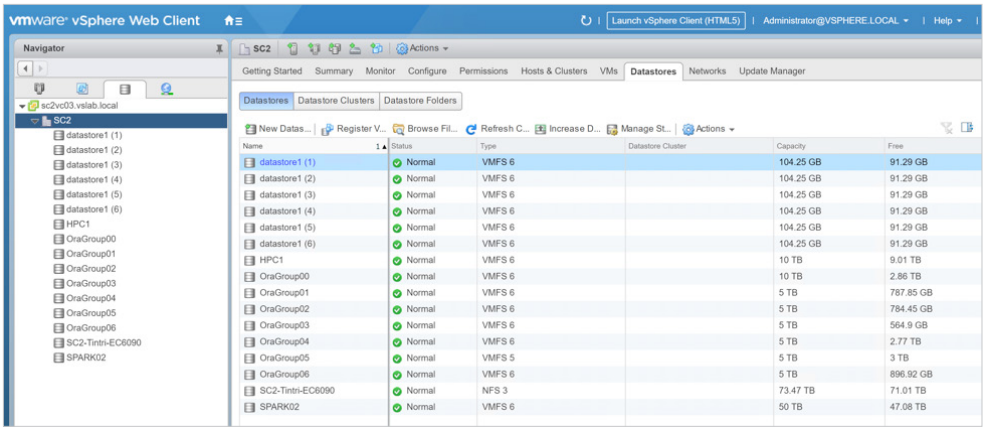

Figure 5. Datastores

Details about Persistent Memory can be seen by directly logging to the ESXi server using the root credentials. Clicking on the 'Persistent Memory' tab and then clicking on the 'Modules' option shows the NVDIMMS modules.

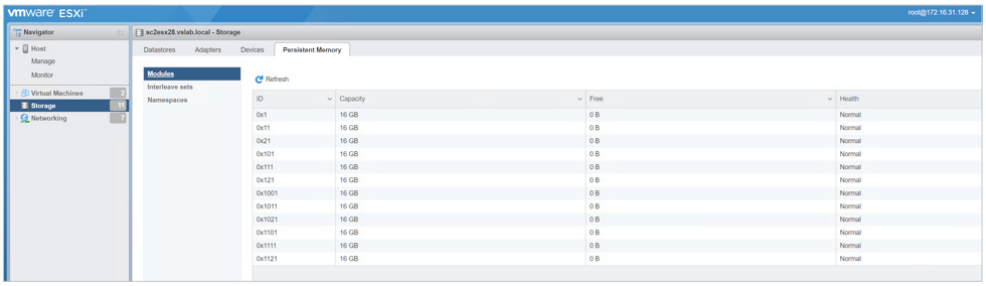

Figure 6. NVDIMMS Modules

ESXi reads namespaces and combines multiple namespaces into one logical volume by writing GPT headers. ESXi uses VMFS-L as a file system format. Each namespace must be marked as 'In Use' for ESXi to create a logical volume.

Clicking on the 'Namespace' option shows us the NVDIMMS namespaces.

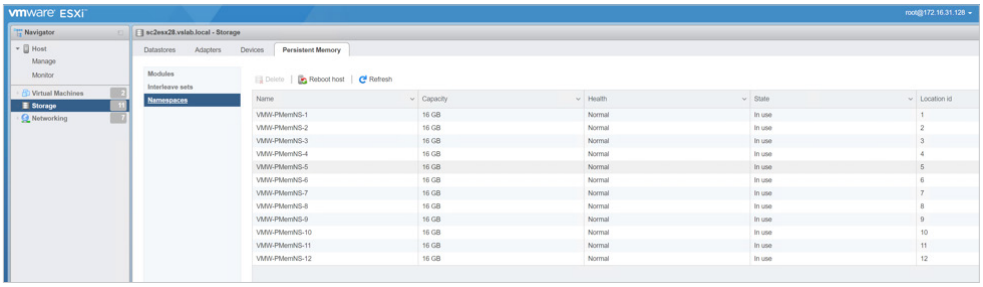

Figure 7. NVDIMMS Namespaces

Clicking on the 'Interleave sets' option shows us the NVDIMMS Interleave sets.

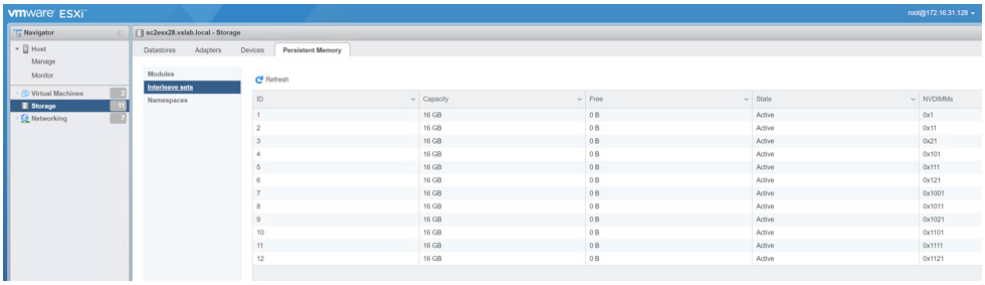

Figure 8. NVDIMMS Interleave Sets

Clicking on the 'Datastores' option shows us the PMEM Datastore of type PMEM mounted on the ESXi server.

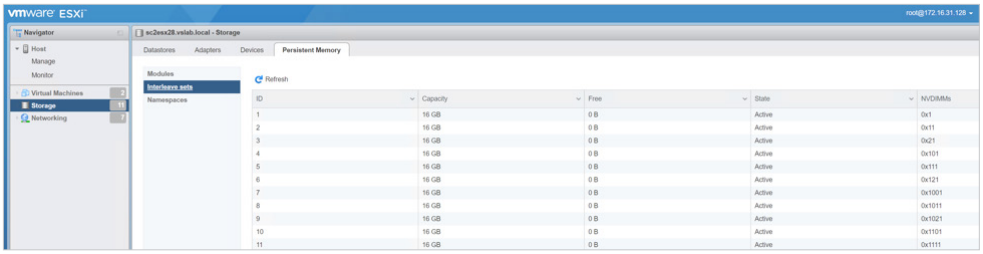

Figure 9. PMEM Datastore

More details about the PMEM datastore can be found by clicking on it.

Figure 10. PMEM Datastore Details

Also, we can monitor PMEM datastore stats using ESXCLI.

Software Resources

Table 4 shows the software resources used in this solution.

SOFTWARE	VERSION	PURPOSE
VMware vCenter Server and ESXi	6.7	ESXi cluster to host virtual machines, All Flash Array provides the datastores. VMware vCenter Server provides a centralized platform for managing VMware vSphere environments.
Oracle Linux	7.4	Oracle database server nodes
Oracle Database 12c	12.2.0.1.0	Oracle database
Oracle Workload Generator for OLTP	SLOB 2.4.2.1	To generate Oracle workload

Table 4. Software Resources

Network Configuration

A VMware vSphere Distributed Switch (VDS) acts as a single virtual switch across all associated hosts in the data cluster. This setup allows virtual machines to maintain a consistent network configuration as they migrate across multiple hosts. The vSphere Distributed Switch uses two 10GbE adapters per host as shown in Figure 11.

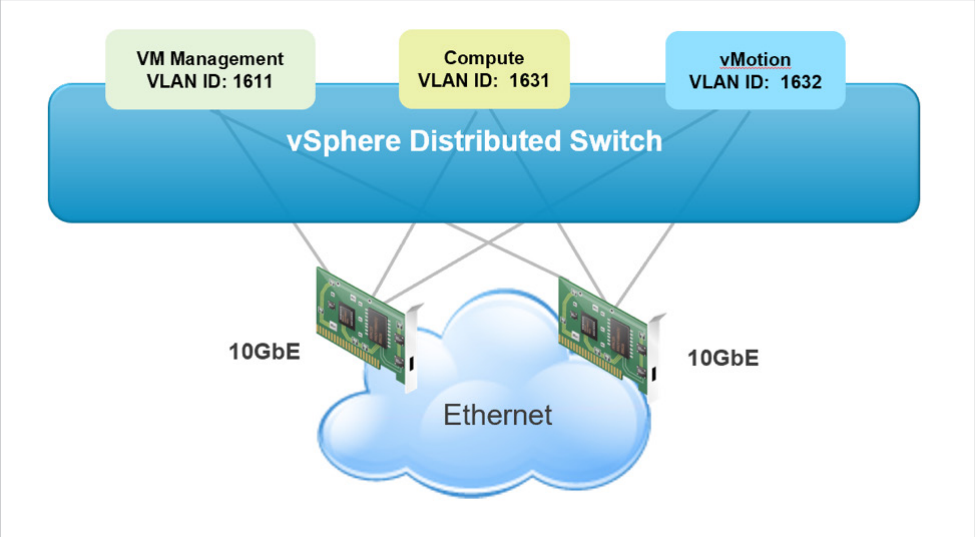

Figure 11. vSphere Distributed Switch Port Group Configuration in All-Flash vSAN

A port group defines properties regarding security, traffic shaping, and NIC teaming. Jumbo frames (MTU=9000 bytes) was enabled on the vSphere vMotion interface and the default port group setting was used.

Figure 11 shows the distributed switch port groups created for different functions and the respective active and standby uplinks to balance traffic across the available uplinks. Three port groups were created:

- VM Management port group for VMs
- Compute port group for kernel port used by Compute traffic
- vSphere vMotion port group for kernel port used by vSphere vMotion traffic

VM and Oracle Configuration

Two VMs were created for the test cases with 12 vCPU and 64 GB memory.

- VM 'Oracle122-OEL-PMEM' was created with Oracle Linux 7.4 operating system.
  - This VM was used for all vPMEMDisk use cases.
  - OEL 7.4 was not compatible with vPMEM mode at the time of writing this paper.
- VM 'Oracle122-RHE-PMEM' was created with Red Hat 7.4 operating system.
  - This VM was used for all vPMEM use cases.

Oracle configuration was kept similar on both VMs:

- Oracle 12.2.0.1.0 Grid Infrastructure and RDBMS binaries were installed on both VMs.
- A single instance database 'DBPROD' was created on both VMs.
- All database-related vmdks were set to Eager Zero thick in Independent Persistent mode to ensure maximum performance with no snapshot capability.
- All database-related vmdks were partitioned using Linux utilities with proper alignment offset and labelled with Oracle ASMLib or Linux udev for device persistence.
- Oracle ASM 'DATA_DG' and 'REDO_DG' disk group were created with external redundancy and configured with default allocation unit (AU) size of 1M.
- ASM 'DATA_DG' and 'REDO_DG' disks were presented on different PVSCSI controllers for performance and queue depth purposes.
- The complete list of Oracle initialization parameters can be found in the Appendix.
- All best practices for Oracle on VMware SDDC was followed as per the 'Oracle Databases on VMware—Best Practices Guide' which can be found here.

VM 'Oracle122-OEL-PMEM'

The steps to add a PVSCSI controller-backed vPMEMDisk vmdk or NVME controller-backed vPMEMDisk vmdk is the same as adding a regular vmdk on a traditional storage based datastore:

- Power off the VM.
- Right-click on the VM and Click 'Edit Settings.'
- Choose add new 'SCSI controller' option and select controller type.
- Choose add a 'New Hard Disk' vmdk.
- Select 'VM storage policy' as 'Host-local PMem Default Storage Policy.'

• Choose rest of the vmdk characteristics.

• Click 'OK.'

In addition to the regular PVSCSI attached vmdks assigned to the VM, as shown below in the table, the VM was also allocated two VMDK's backed by vPMEMDisk datastore.

• vmdk on SCSI (3:0) using PVSCSI controller

• vmdk on NVME (0:0) using NVME controller

Table 5 provides VM 'Oracle122-OEL-PMEM' disk layout and ASM disk group configuration.

NAME	SCSI TYPE	SCSI ID (CONTROLLER, LUN)	SIZE (GB)	TYPE	SOURCE	HARD DISK	DEVICE
Operating System (OS) /	Paravirtual	SCSI (0:0)	50	ext4 Filesystem	All Flash SAN	1	/dev/sda1
Oracle binary disk /u01	Paravirtual	SCSI (0:1)	50	ext4 Filesystem	All Flash SAN	2	/de/sdb1
Database data disk 1	Paravirtual	SCSI (1:0)	2048	DATA_DG (ASM)	All Flash SAN	3	/dev/sdd1
Redo disk 1	Paravirtual	SCSI (2:0)	32	REDO_DG (ASM)	All Flash SAN	4	/dev/sdc1
Redo disk 2	Paravirtual	SCSI (3:0)	32	REDO_SCSI_ PMEM_DG (ASM)	vPMEMDisk	5	/dev/sde1
Redo disk 3	NVME	NVME (0:0)	32	REDO_NVME_ PMEM_DG (ASM)	vPMEMDisk	6	/dev/nvme0n1p1
Redo disk 4	Paravirtual	SCSI (3:1)	32	/redolog ext44 filesystem	vPMEMDisk	7	/dev/sde1
Redo disk 5	Paravirtual	SCSI (3:2)	32	/redolog_pmem ext4 filesystem	vPMEMDisk	8	/dev/sdf1

Table 5. Oracle Database VM 'Oracle122-OEL-PMEM' Disk Layout

Details of PVSCSI controller-backed vPMEMDisk vmdk are shown as below:

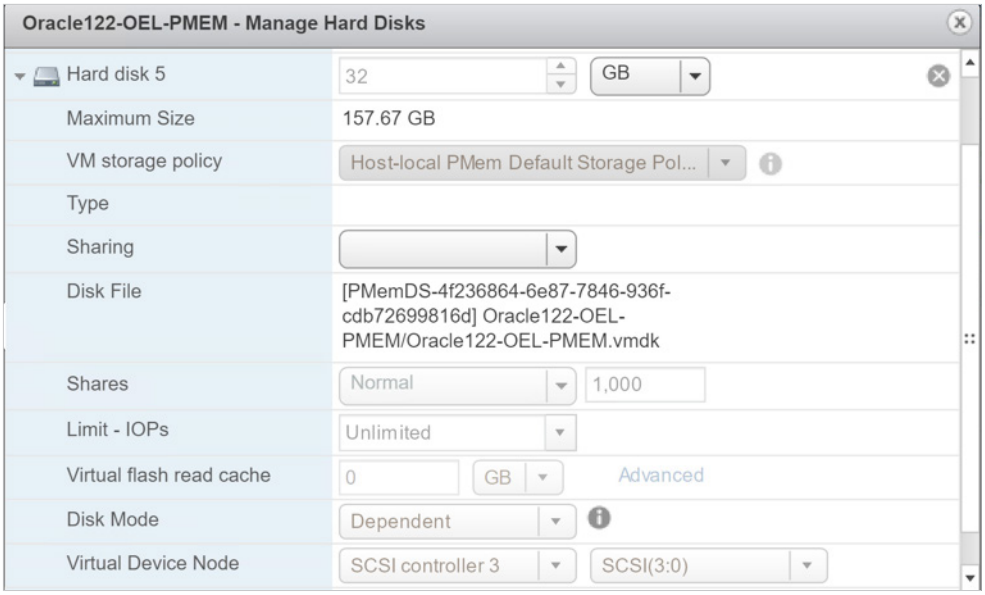

Figure 12. PVSCSI Controller-backed vPMEMDisk vmdk

Details of NVME controller-backed vPMEMDisk vmdk are shown as below:

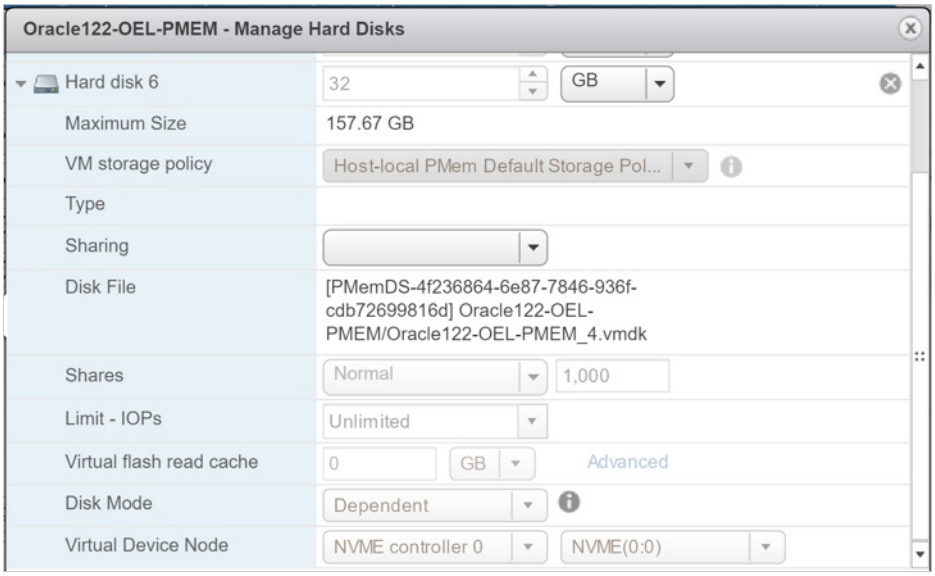

Figure 13. NVME Controller-backed vPMEMDisk vmdk

Listing of Linux disk devices which shows Oracle ASM disks:

```
[root@oracle122-oel-pmem ~]# blkid
/dev/sda1: UUID="a30550f2-4f1b-4566-ab77-2e45d5549802" TYPE="xfs"
/dev/sda2: UUID="AfEcND-dKce-mY8Y-lHnT-LX02-VMYY-UoEfmr" TYPE="LVM2_member"
/dev/sdc1: LABEL="REDO_DISK01" TYPE="oracleasm"
/dev/sdb1: UUID="qdLbqS-hC3v-CNLf-SSus-pcSL-RwkS-sJXUgh" TYPE="LVM2_member"
/dev/sdd1: LABEL="DATA_DISK01" TYPE="oracleasm"
/dev/mapper/ol-root: UUID="763d05b9-12c8-4554-9129-682c74adbe0a" TYPE="xfs"
/dev/mapper/ol-swap: UUID="261bfd9d-51a2-4a12-8ab4-5948a95fe5af" TYPE="swap"
/dev/mapper/vg2_oracle-LogVol_u01: UUID="2313fca8-fe16-46ff-afa1-9c541aa45c72" TYPE=
/dev/sde1: UUID="juSS20-22oT-NNZv-hVWF-nzgf-naDh-kT949Q" TYPE="LVM2_member"
/dev/mapper/vg2_redolog-LogVol_redolog: UUID="5ce90ceb-6010-4e0c-824b-cc18fa138d16"
```

Figure 14. Listing of Linux Disk Devices

Listing of ASM Disk Groups:

```
State    Type    Rebal  Sector  Logical_Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
MOUNTED  EXTERN  N         512             512   4096  4194304   2097148   680552                0          680552              0
MOUNTED  EXTERN  N         512             512   4096  1048576     32767    24451                0           24451              0
MOUNTED  EXTERN  N         512             512   4096  1048576     32767    32714                0           32714              0
MOUNTED  EXTERN  N         512             512   4096  1048576     32767    30643                0           30643              0
ASMCMD>
```

Figure 15. Listing of ASM Disk Groups

VM 'Oracle122-RHEL-PMEM'

The steps to add a NVDIMM are shown below:

- Power off the VM.
- Right-click on the VM and Click 'Edit Settings.'
- Choose add new 'NVDIMM.'
- A new NVDIMM controller and device is added.
- Assign NVDIMM size.
- Click 'OK.'

In addition to the regular PVSCSI attached vmdks assigned to the VM, as shown below in the table, the VM was also allocated three NVDIMMs.

- NVDIMM1 using vPMEM raw mode for ASM redo disk group using udev
- NVDIMM2 using vPMEM memory mode for ASM redo disk group using udev
- NVDIMM3 using vPMEM memory mode for ext4 dax mounted filesystem for redo logs

Table 6 provides VM 'Oracle122-RHEL-PMEM' disk layout and disk group configuration.

NAME	SCSI TYPE	SCSI ID (CONTROLLER, LUN)	SIZE (GB)	TYPE	SOURCE	HARD DISK	DEVICE
Operating System (OS) /	Paravirtual	SCSI (0:0)	50	ext4 Filesystem	All Flash SAN	1	/dev/sda1
Oracle binary disk /u01	Paravirtual	SCSI (0:1)	50	ext4 Filesystem	All Flash SAN	2	/de/sdb1
Database data disk 1	Paravirtual	SCSI (1:0)	2048	DATA_DG (ASM)	All Flash SAN	3	/dev/sdd1
Redo disk 1	Paravirtual	SCSI (2:0)	32	REDO_DG (ASM)	All Flash SAN	4	/dev/sde1
Redo disk 2	Paravirtual	SCSI (3:0)	32	/redolog (ext4 Filesystem)	All Flash SAN	5	/dev/sdc1
Redo disk 3	NVDIMM1	Not Applicable	32	REDO_RAW_DG (ASM)	vPMEM	6	/dev/pmem0p1
Redo disk 4	NVDIMM2	Not Applicable	32	REDO_ MEMORY_DG (ASM)	vPMEM	7	/dev/pmem1p1
Redo disk 5	NVDIMM3	Not Applicable	32	/redolog_pmem (ext4 dax Filesystem)	vPMEM	8	/dev/pmem2p1

Table 6. Oracle Database VM 'Oracle122-RHEL-PMEM' Disk Layout

Details of NVDIMM1 in raw mode vmdk can be found in the figure below.

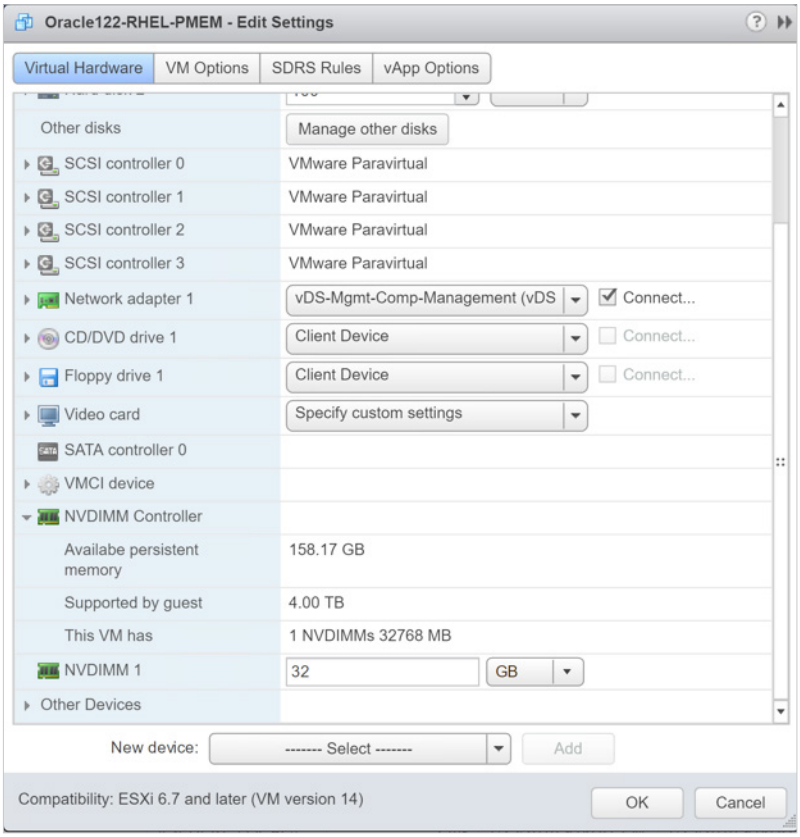

Figure 16. Characteristics of NVDIMM1 vPMEM vmdk

Details of NVDIMM2 for memory mode vmdk can be found in the figure below.

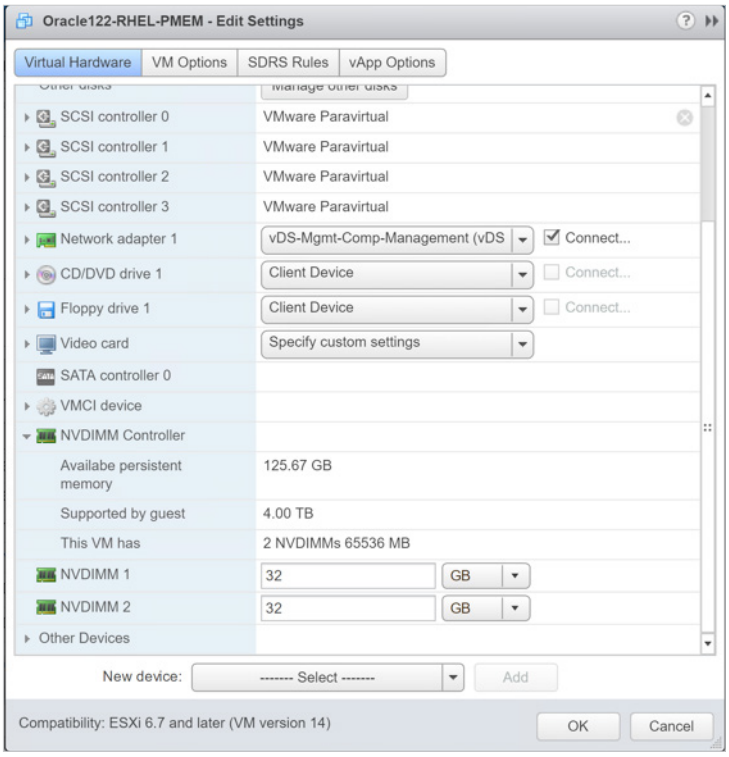

Figure 17. Characteristics of NVDIMM2 vPMEM vmdk

Details of NVDIMM3 for memory DAX mode vmdk can be found in the figure below.

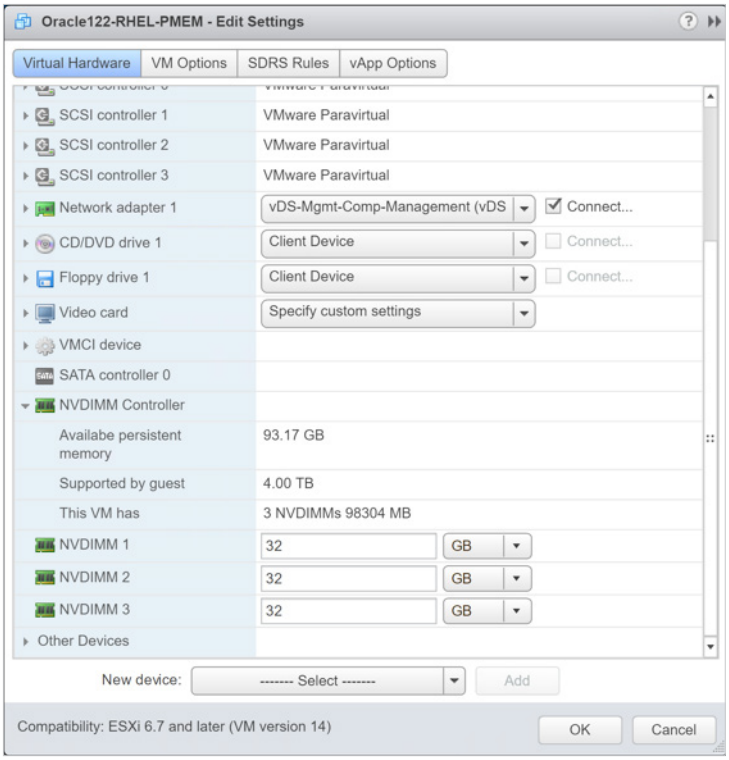

Figure 18. Characteristics of NVDIMM3 vPMEM vmdk

Listing of vPMEM namespaces:

```
    "uuid":"9d588e2f-a076-4aef-be4d-7a6b441bfdf4",
    "blockdev":"pmem2",
    "numa_node":0
  },
  {
    "dev":"namespace1.0",
    "mode":"memory",
    "size":33820770304,
    "uuid":"d039b24f-57b1-4693-8007-ee22fc2e3b06",
    "blockdev":"pmem1",
    "numa_node":0
  },
  {
    "dev":"namespace0.0",
    "mode":"raw",
    "size":34359738368,
    "blockdev":"pmem0",
    "numa_node":0
  }
]
[root@oracle122-rhel ~]#
```

Figure 19. Listing of vPMEM namespaces

Partition the /dev/pmemX devices using appropriate alignment offset:

- /dev/pmem0 in PMEM raw mode for ASM
- /dev/pmem1 in PMEM memory mode for ASM
- /dev/pmem2 in PMEM memory mode for file system DAX mode

Listing of Linux disk devices:

```
/dev/sdb1: UUID="dBR5Ev-use9-R0AL-hdXy-lJEo-zHdg-JORNGi" TYPE="LVM2_member"
/dev/sdd1: LABEL="DATA_DISK01" TYPE="oracleasm"
/dev/sde1: LABEL="REDO_DISK01" TYPE="oracleasm"
/dev/mapper/rhel-root: UUID="11075f4b-94b4-494b-9b8e-46eee1c93309" TYPE="xfs"
/dev/mapper/rhel-swap: UUID="36d90b7c-0f5b-4795-bac9-37d7b3de7309" TYPE="swap"
/dev/pmem0: PTTYPE="dos"
/dev/pmem0p1: TYPE="oracleasm"
/dev/pmem1: PTTYPE="dos"
/dev/pmem1p1: TYPE="oracleasm"
/dev/pmem2: PTTYPE="dos"
/dev/pmem2p1: UUID="715ab974-6120-49b4-9876-4356c01cced4" TYPE="ext4"
/dev/mapper/rhel-home: UUID="c3160371-c818-45fa-92f1-635dc78ed848" TYPE="xfs"
/dev/mapper/vg2_oracle-LogVol_u01: UUID="43c166df-6784-4d32-b557-2591c8fb4781"
/dev/mapper/vg2_redolog-LogVol_redolog: UUID="1167218e-344b-4bee-ae34-1eef516a
[root@oracle122-rhel ~]#
```

Figure 20. Listing of Linux Disk Devices

Listing of disk partitions:

```
  ├─rhel-root                   253:0    0 35.6G  0 lvm  /
  ├─rhel-swap                   253:1    0    6G  0 lvm  [
  └─rhel-home                   253:2    0 17.4G  0 lvm  /
sdb                               8:16   0  100G  0 disk
└─sdb1                            8:17   0  100G  0 part
  └─vg2_oracle-LogVol_u01       253:3    0  100G  0 lvm  /
sdc                               8:32   0   32G  0 disk
└─sdc1                            8:33   0   32G  0 part
  └─vg2_redolog-LogVol_redolog  253:4    0   32G  0 lvm  /
sdd                               8:48   0    2T  0 disk
└─sdd1                            8:49   0    2T  0 part
sde                               8:64   0   32G  0 disk
└─sde1                            8:65   0   32G  0 part
sr0                              11:0    1 1024M  0 rom
pmem0                           259:0    0   32G  0 disk
└─pmem0p1                       259:1    0   32G  0 part
pmem1                           259:2    0 31.5G  0 disk
└─pmem1p1                       259:3    0 31.5G  0 part
pmem2                           259:4    0 31.5G  0 disk
└─pmem2p1                       259:5    0 31.5G  0 part /
[root@oracle122-rhel ~]#
```

Figure 21. Listing of Disk Partitions udev Rules for ASM Disks

udev rules for ASM disks:

Figure 22. udev Rules for ASM Disks

Listing of ASM Disk Groups:

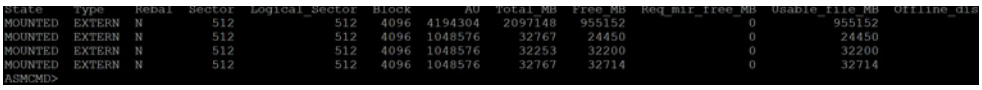

Figure 23. Listing of ASM Disk Groups

Contents of /etc/fstab showing ext filesystem (both dax and non-dax) for redo log:

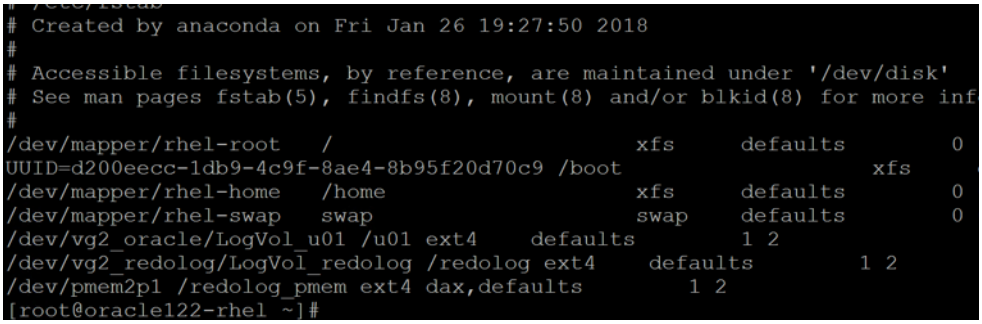

Figure 24. /etc/fstab Contents

Mount options showing /redolog_pmem in DAX mode:

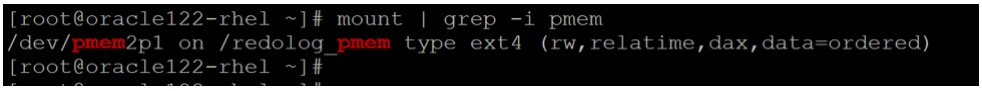

Figure 25. ext4 filesystem in DAX Mode

Solution Validation

The proposed solution is designed and deployed using two Oracle single instance databases, one on OEL 7.4 for vPMEMDisk use case and one on RHEL 7.4 for vPMEM use case, on a 3-node vSphere Cluster with Micron NVDIMMS.

The scenarios tested are summarized below:

SCENARIOS	PMEM MODE	USE CASE	STORAGE
Improved Performance of Oracle Redo Log	vPMEMDisk	Redo logs on Oracle ASM	Traditional storage
		Redo logs on Oracle ASM	vPMEMDisk datastore with PVSCSI Controller
		Redo logs on Oracle ASM	vPMEMDisk datastore with NVME Controller
		Redo logs on File system	Traditional storage
		Redo logs on File system	vPMEMDisk datastore with PVSCSI Controller
Improved Performance of Oracle Redo Log	vPMEM	Redo logs on Oracle ASM	Traditional storage
		Redo logs on Oracle ASM	vPMEM with raw mode
		Redo logs on Oracle ASM	vPMEM with memory mode
		Redo logs on File system	Traditional storage
		Redo logs on File system	vPMEM with memory mode with DAX option
Accelerating Performance Using Oracle Database Smart Flash Cache	vPMEMDisk	Oracle Flash Cache on ASM	No caching
		Oracle Flash Cache on ASM	Caching using ASM on vPMEMDisk datastore with PVSCSI Controller
		Oracle Flash Cache on File system	Caching using ext4 file system on vPMEMDisk datastore with PVSCSI Controller
Potential Reduction in Oracle Licensing	vPMEMDisk	Redo logs on Oracle ASM	Traditional storage
		Redo logs on Oracle ASM	vPMEMDisk datastore with NVME Controller

Table 7. Persistent Memory Test Cases

In this section, we present the test methodologies and processes used in this reference architecture.

Solution Test Overview

This solution primarily uses SLOB TPCC like workload generator to generate heavy batch processing workload on the Oracle database.

A large Oracle database was created in a virtual machine against which SLOB tool was deployed to load database schemas. Subsequently, SLOB batch processing workload

was generated on the database.

During this workload generation, Oracle AWR, and Linux SAR reports were used to compare the performance and validate the testing use cases.

The Oracle database was restarted after every test case to ensure no blocks or SQLs cached in the SGA.

Test and Performance Data Collection Tools

Test Tools and Configuration

Oracle OLTP Workload

SLOB is an Oracle workload generator designed to stress test storage I/O capability, specifically for Oracle database using OLTP workload. SLOB is not a traditional transactional benchmark tool. It is used to validate performance of the storage subsystem without application contention.

SLOB and Database Configuration

- Database VM (12 vCPU and 64GB memory)
- Database VM with a 2,048GB SLOB schema
- Workload is purely a 100 percent write to mimic a heavy IO database batch processing workload (SLOB parameter UPDATE_PCT was set to 100).
- Number of users set to 1 with 0 think time to hit each database with maximum requests concurrently to generate extremely intensive batch workload.
- SLOB parameter SCALE for the workload was set to 1024GB with Oracle SGA set to 32GB.
- SLOB parameter REDO_STRESS for the workload was set to HEAVY.
- SLOB parameter RUN_TIME was set to 30 minutes.
- Detailed SLOB configuration is included in Appendix A SLOB Configuration.

Key Metrics Data Collection Tools

We used the following monitoring tools in this solution:

- Oracle AWR reports with Automatic Database Diagnostic Monitor (ADDM): Automatic Workload Repository (AWR) collects, processes, and maintains performance statistics for problem detection and self-tuning purposes for Oracle database. This tool can generate report for analyzing Oracle performance.

  The Automatic Database Diagnostic Monitor (ADDM) analyzes data in AWR to identify potential performance bottlenecks. For each of the identified issues, it locates the root cause and provides recommendations for correcting the problem.

  More information on Oracle AWR can be found here.

• Linux SAR (system Activity Report):
Linux sar helps collects and evaluate a variety of information regarding system activity. With performance problems, sar also permits retroactive analysis of the load values for various sub-systems (CPUs, memory, disks, interrupts, network interfaces and so forth).

More information on Linux SAR can be found here.

Improved Performance of Oracle Redo Log

We look at below different test cases where we use heavy write intensive batch processing SLOB workload against an Oracle database with different sets of Redo log groups using the two Persistent Memory modes.

• vPMEMDisk mode

• vPMEM mode

vPMEMDisk Mode

SCENARIOS	PMEM MODE	USE CASE	STORAGE
Improved Performance of Oracle Redo Log	vPMEMDisk	Redo logs on Oracle ASM	Traditional storage
		Redo logs on Oracle ASM	vPMEMDisk datastore with PVSCSI Controller
		Redo logs on Oracle ASM	vPMEMDisk datastore with NVME Controller
		Redo logs on File system	Traditional storage
		Redo logs on File system	vPMEMDisk datastore with PVSCSI Controller

Table 8. Oracle Redo Log Test Cases — vPMEMDisk Mode

Test Case 1: Overview

VM 'Oracle122-OEL-PMEM' has a single instance database 'DBPROD' running. A 100 percent write-intensive database batch processing workload was run against the database.

Test Case 1

• Sixteen database redo log groups had already been created in the REDO_DG ASM disk group on traditional storage at the time of database creation.

• Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the workload, being very batch intensive, was waiting on Checkpoint process to complete, and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.

• Run SLOB against the database 'DBPROD' for the 30-minute run time.

- Generate the AWR report and SAR output for the run time.
- This test is the baseline for all subsequent comparisons.

Test Cases 2 and 3: Overview

Using the same VM 'Oracle122-OEL-PMEM' with the single instance database 'DBPROD' running, Test Cases 2 and 3 were performed:

Test Case 2

- Add sixteen new database redo log groups to the REDO_SCSI_PMEM_DG ASM disk group which consists of vPMEMDisk-backed vmdk using PVSCSI Controller.
- Drop existing database redo log groups on the REDO_DG ASM disk group on traditional storage.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report and SAR output for the run time.

Test Case 3

- Add sixteen new database redo log groups to the REDO_NVME_PMEM_DG ASM disk group which consists of vPMEMDisk backed vmdk using NVME Controller.
- Drop existing database redo log groups on the REDO_SCSI_PMEM_DG ASM disk group.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report and SAR output for the run time.

Test Cases 1, 2, and 3: Summary

The above three test runs show:

- Reduction in wait times for critical database events (e.g., 'log file switch completion,' 'log file parallel write,' etc.)
- Increase in the amount of work by the workload
- Lessens the impact of log file switches
  - Even though log file switches occurred multiple times during the tests, there was no impact on performance due to the fast write time to the vPMEMDisk vmdk.

Summary performance metrics for all three uses cases:

TEST RUN METRICS	BASELINE 8:00-8:30AM	vPMEMDISK WITH PVSCSI CONTROLLER 9:30-10:00AM	vPMEMDISK WITH NVME CONTROLLER 10:15-10:45AM
Executes (SQL)	576	587.7	604.8
Transactions	558.6	567.5	583.1
log file switch (checkpoint incomplete) — Ave Wait	18.40ms	16.76ms	14.58ms
log file switch completion	18.95ms	17.42ms	20.19ms
log file switch (private strand flush incomplete) — Avg Wait	22.22ms	17.64ms	17.36ms
log file parallet write — Avg Wait	511.49ms	76.04us	75.10us

Table 9. Summary Performance Metrics for All Three Use Cases

Test Cases 4 and 5: Overview

VM 'Oracle122-OEL-PMEM' has a single instance database 'DBPROD' running. A 100 percent write-intensive database batch processing workload was run against the database.

Test Case 4

- Add sixteen new database redo log groups to the /redolog ext4 filesystem backed by traditional storage using PVSCSI Controller.
- Drop old redo log file on ASM disk groups.
- Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the workload, being very batch intensive, was waiting on Checkpoint process to complete and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report and SAR output for the run time.
- This test is the baseline for all subsequent comparisons.

Test Case 5

- Add sixteen new database redo log groups to the /redolog_pmem ext4 filesystem backed by vPMEMDisk storage using PVSCSI Controller.
- Drop old redo log file on the /redolog ext4 filesystem.
- Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the workload, being very batch intensive, was waiting on Checkpoint process to complete and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report and SAR output for the run time.

Test Cases 4 and 5: Summary

The above two test runs show:

- Reduction in wait times for critical database events (e.g., 'log file switch [private strand flush incomplete]', 'log file parallel write,' etc.
- Increase in the amount of work by the workload
- Lessens the impact of log file switches
  - Even though log file switches occurred multiple times during the tests, there was no impact on performance due to the fast write time to the vPMEMDisk vmdk.

Analysis of the test results can be found at Appendix C.

vPMEM Mode

SCENARIOS	PMEM MODE	USE CASE	STORAGE
Improved Performance of Oracle Redo Log	vPMEM	Redo logs on Oracle ASM	Traditional storage
		Redo logs on Oracle ASM	vPMEM with raw mode
		Redo logs on Oracle ASM	vPMEM with memory mode
		Redo logs on File system	Traditional storage
		Redo logs on File system	vPMEM with memory mode with DAX option

Table 10. Oracle Redo Log Test Cases — vPMEM Mode

Test Case 1: Overview

VM 'Oracle122-RHEL-PMEM' has a single instance database 'DBPROD' running. A 100 percent write-intensive database batch processing workload was run against the database.

Test Case 1

- Sixteen database redo log groups already been created in the REDO_DG ASM disk group on traditional storage at the time of database creation.
- Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the workload, being very batch intensive, was waiting on Checkpoint process to complete and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report and SAR output for the run time.
- This test is the baseline for all subsequent comparisons.

Test Cases 2 and 3: Overview

Using the same VM 'Oracle122-RHEL-PMEM' with the single instance database 'DBPROD' running, Test Cases 2 and 3 were performed:

Test Case 2

- Add sixteen new database redo log groups to the REDO_RAW_DG ASM disk group backed by vPMEM in raw mode.
- Drop existing database redo log groups on the REDO_DG ASM disk group on traditional storage.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report and SAR output for the run time.

Test Case 3

- Add sixteen new database redo log groups to the REDO_MEMORY_DG ASM disk group backed by vPMEM in memory mode.
- Drop existing database redo log groups on the REDO_RAW_DG ASM disk group.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report and SAR output for the run time.

Test Cases 1, 2, and 3: Summary

The above three test runs show:

- Reduction in wait times for critical database events (e.g., 'log file switch completion,' 'log file parallel write,' etc.)
- Increase in the amount of work by the workload
- Lessens the impact of log file switches
  - Even though log file switches occurred multiple times during the tests, there was no impact on performance due to the fast write time to the vPMEM vmdk.

Summary performance metrics for all three uses cases:

TEST RUN METRICS	BASELINE 7:13-7:44AM	vPMEM RAW MODE 11:22-11:53AM	vPMEM MEMORY MODE 12:22-12:53PM
Executes (SQL)	662.2	676.8	664.4
Transactions	640.4	656.8	643.2
log file switch completion	21.88ms	18.87ms	18.29ms
log file switch (private strand flush incomplete) — Avg Wait	26.62ms	27.23ms	21.98ms
log file parallet write — Avg Wait	617.73us	0	0

Table 11. Summary Performance Metrics for All Three Use Cases

Test Cases 4 and 5: Overview

Using the same VM 'Oracle122-RHEL-PMEM' with the single instance database 'DBPROD' running, Test Cases 4 and 5 were conducted.

Test Case 4

- Add sixteen new database redo log groups to the /redolog ext4 file system on traditional storage without DAX option.
- Drop existing database redo log groups on the REDO_DG ASM disk group on traditional storage.
- Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the workload, being very batch intensive, was waiting on Checkpoint process to complete and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.'
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report and SAR output for the run time.
- This test is the baseline for all subsequent comparisons.

Test Case 5

- Add sixteen new database redo log groups to the /redolog_pmem ext4 filesystem backed by vPMEM with DAX mode option.
- Drop existing database redo log groups on the /redolog ext4 filesystem on traditional storage.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report and SAR output for the run time.

Test Cases 4 and 5: Summary

The above two test runs show:

- Reduction in wait times for critical database events (e.g., 'log file switch completion,' 'log file switch [private strand flush incomplete],' etc.
- Increase in the amount of work by the workload
- Lessens the impact of log file switches
  - Even though log file switches occurred multiple times during the tests, there was no impact on performance due to the fast write time to the vPMEM vmdk.

Summary performance metrics for both use cases:

TEST RUN METRICS	EXT4 FILESYSTEM 3:28-3:58PM	EXT4 FILESYSTEM IN DAX MODE 4:16-4:46PM
Executes (SQL)	646.8	680.6
Transactions	625.8	660.9
log file switch completion — Avg Wait	22.90ms	18.89ms
log file switch (private strand flush incomplete) — Avg Wait	23.87ms	20.76ms

Table 12. Summary Performance Metrics for Test Cases 4 and 5

Analysis of the test results can be found at Appendix C.

Accelerating Performance Using Oracle Smart Flash Cache

We look at below different scenarios where we use heavy write-intensive batch processing SLOB workload against an Oracle database with Oracle Smart Flash Cache configured using the vPMEMDisk mode.

Note, vPMEM mode cannot be used as at the time of writing this paper:

- Database Smart Flash Cache is only supported on databases running on the Solaris or Oracle Linux operating systems.
- vPMEM mode is not supported on Oracle Enterprise Linux.

SCENARIO	PMEM MODE	USE CASE	STORAGE
Accelerating Performance Using Oracle Database Smart Flash Cache	vPMEMDisk	Oracle Flash Cache on ASM	No caching
		Oracle Flash Cache on ASM	Caching using ASM on vPMEMDisk datastore with PVSCSI Controller
		Oracle Flash Cache on file system	Caching using ext4 file system on vPMEMDisk datastore with PVSCSI Controller

Table 13. Oracle Database Smart Flash Cache Test Cases

Test Cases 1, 2, and 3: Overview

VM 'Oracle122-OEL-PMEM' has a single instance database 'DBPROD' running. A 100 percent write-intensive database batch processing workload was run against the database.

Test Case 1

- Oracle Smart Flash Cache is turned off (default).
- Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the

workload, being very batch intensive, was waiting on Checkpoint process to complete and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.

- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report for the run time.
- This test is the baseline for all subsequent comparisons.

Test Case 2

- Oracle Smart Flash Cache is turned on using ASM disk group on vPMEMDisk datastore with PVSCSI Controller.
- Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the workload, being very batch intensive, was waiting on Checkpoint process to complete and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.
- Initialization parameter 'db_flash_cache_file' was set to '+FLASH_DG/flashfile,' 'db_flash_cache_size' was set to 62G.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report for the run time.

Test Case 3

- Oracle Smart Flash Cache is turned on using ext4 filesystem on vPMEMDisk datastore with PVSCSI Controller.
- Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the workload, being very batch intensive, was waiting on Checkpoint process to complete and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.
- Initialization parameter 'db_flash_cache_file' was set to '/flashcache/flashfile', 'db_flash_cache_size' was set to 62G.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report for the run time.

Test Cases 1, 2, and 3: Summary

The above three test runs show:

- Faster read times for single block reads from Smart Flash cache
- Increase in the amount of work by the workload

Analysis of the test results can be found at Appendix C.

Potential Reduction in Oracle Licensing

We look at below different scenarios where we use heavy write-intensive batch processing SLOB workload against an Oracle database configured with different sets

of Redo log groups using the two Persistent Memory modes:

- vPMEMDisk mode
- vPMEM mode

The intention of this test is to see if we can run the same workload with no reduction in performance with reduced number of vCPUs.

Doing this across the board for all Oracle workloads will in turn potentially lead to:

- Reduced number of physical server cores needed to run the same workload
  - o This in turn will lead to a potential reduction in the number of Oracle Enterprise Edition core licenses needed to run the same workload with possible revenue savings.

SCENARIOS	PMEM MODE	USE CASE	STORAGE
Potential Reduction in Oracle Licensing	vPMEMDisk	Redo logs on Oracle ASM	Traditional storage
		Redo logs on Oracle ASM	vPMEMDisk datastore with NVME Controller

Table 14. Reduction in Oracle Licensing Test Cases

The test results were found to be similar in the vPMEM case as well.

Test Cases 1 and 2: Overview

VM 'Oracle122-OEL-PMEM' has a single instance database 'DBPROD' running. A 100 percent write-intensive database batch processing workload was run against the database.

Test Case 1

- Sixteen database redo log groups already been created in the REDO_DG ASM disk group on traditional storage at the time of database creation.
- Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the workload, being very batch intensive, was waiting on Checkpoint process to complete and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.
- The number of vCPUs of the VM was 12.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report for the run time.
- This test is the baseline for all subsequent comparisons.

Test Case 2

- Add 16 new database redo log groups to the REDO_NVME_DG ASM disk group

backed by vPMEMDisk with NVME Controller.

- Drop existing database redo log groups on the REDO_DG ASM disk group on traditional storage.
- Initialization parameter 'db_writer_processes' was set at '3' as the initial run of the workload, being very batch intensive, was waiting on Checkpoint process to complete and the intention of the test is to demonstrate the reduced wait time on 'log file switch' event.
- The number of vCPUs of the VM was reduced to 9.
- Run SLOB against the database 'DBPROD' for the 30-minute run time.
- Generate the AWR report for the run time.
- This test is the baseline for all subsequent comparisons.

Test Cases 1 and 2: Summary

The above two test runs show placing redo log files on vPMEMDisk-backed vmdk helps with reducing vCPUs with no effect to the workload performance. Doing this across the board for all Oracle workloads will in turn potentially lead to:

- Reduced number of physical server cores needed to run the same workload.
- A potential reduction in the number of Oracle Enterprise Edition core licenses. needed to run the same workload with possible revenue savings

Analysis of the test results can be found at Appendix C.

Conclusion

The above test cases using vPMEMdisk and vPMEM mode indicates a reduction in wait times for critical database events (e.g.,'log file switch completion,' 'log file switch [private strand flush incomplete]) and at the same time an increase in the amount of work done by the workload.

The table below shows a summary of test cases and results:

SCENARIOS	PMEM MODE	USE CASE	STORAGE	RESULTS SUMMARY
Improved Performance of Oracle Redo Log	vPMEMDisk	Redo logs on Oracle ASM	• Traditional storage • vPMEMDisk datastore with PVSCSI Controller • vPMEMDisk datastore with NVME Controller	• Reduction in wait times for critical database events (e.g., 'log file switch completion,' 'log file parallel write,' etc.) • Increase in the amount of work by the workload • Lessens the impact of log file switches even though log file switches occurred multiple times during the tests; there was no impact on performance due to the fast write time to the vPMEMDisk vmdk
		Redo logs on File System	• Traditional storage • vPMEMDisk datastore with PVSCSI Controller	• Reduction in wait times for critical database events (e.g., 'log file switch (private strand flush incomplete),' 'log file parallel write,' etc. • Increase in the amount of work by the workload • Lessens the impact of log file switches even though log file switches occurred multiple times during the tests; there was no impact on performance due to the fast write time to the vPMEMDisk vmdk
Improved Performance of Oracle Redo Log	vPMEM	Redo logs on Oracle ASM	• Traditional storage • vPMEM with raw mode • vPMEM with memory mode	• Reduction in wait times for critical database events (e.g., 'log file switch completion,' 'log file parallel write,' etc.) • Increase in the amount of work by the workload • Lessens the impact of log file switches even though log file switches occurred multiple times during the tests; there was no impact on performance due to the fast write time to the vPMEM vmdk

SCENARIOS	PMEM MODE	USE CASE	STORAGE	RESULTS SUMMARY
		Redo logs on File system	• Traditional storage • vPMEM with memory mode with DAX option	• Reduction in wait times for critical database events (e.g., 'log file switch completion,' 'log file switch [private strand flush incomplete], etc.) • Increase in the amount of work by the workload • Lessens the impact of log file switches even though log file switches occurred multiple times during the tests; there was no impact on performance due to the fast write time to the vPMEM vmdk
Accelerating Performance Using Oracle Database Smart Flash Cache	vPMEMDisk	Oracle Flash Cache on ASM	• No caching • Caching using ASM on vPMEMDisk datastore with PVSCSI Controller • Caching using ext4 file system on vPMEMDisk datastore with PVSCSI Controller	• Faster read times for single block reads from Smart Flash cache • Increase in the amount of work by the workload
Potential Reduction in Oracle Licensing		Redo logs on Oracle ASM	• Traditional storage • vPMEMDisk datastore with NVME Controller	• Reduced number of physical server cores needed to run the same workload • In turn will lead to a potential reduction in the number of Oracle Enterprise Edition core licenses needed to run the same workload with possible revenue savings

Table 15. Persistent Memory — Summary of Test cases and Results

Deploying IO-intensive Oracle workloads requires fast storage performance with low latency and resiliency from database failures. Latency, which is a measurement of response time, directly impacts a technology's ability to deliver faster performance for business-critical applications.

Persistent Memory (PMEM) technology enables byte-addressable updates and prevents data loss during power interuptions. Instead of having nonvolatile storage at the bottom with the largest capacity but the slowest performance, nonvolatile storage is now very close to DRAM in terms of performance.

PMEM is a byte-addressable form of computer memory that has the following characteristics:

• DRAM-like latency and bandwidth

• Regular load/store CPU instructions

• Paged/mapped by operating system just like DRAM

• Data is persistent across reboots

VMware vSphere 6.7 brings a lot of great new features and innovations—especially vSphere Persistent Memory (PMEM) which aids business-critical Oracle workloads, offering both enhanced performance and faster recovery.

Using the capabilities of Micron PMEM and vSphere Persistent Memory capability, we were able to showcase several use cases like improved performance of Oracle Redo Log, accelerating performance using Oracle Smart Flash Cache, and potential reduction in Oracle licensing costs.

Appendix A SLOB Configuration

SLOB configuration files

SLOB 2.4.0 slob.conf

UPDATE_PCT=100

SCAN_PCT=0

RUN_TIME=1800

WORK_LOOP=0

SCALE=1024G

SCAN_TABLE_SZ=1M

WORK_UNIT=64

REDO_STRESS=HEAVY

LOAD_PARALLEL_DEGREE=20

THREADS_PER_SCHEMA=1000

DATABASE_STATISTICS_TYPE=awr # Permitted values: [statspack|awr]

*#### Settings for SQL*Net connectivity:*

Uncomment the following if needed:

ADMIN_SQLNET_SERVICE=dbprod_pmem_oel_pdb1

SQLNET_SERVICE_BASE=dbprod_pmem_oel_pdb1

#SQLNET_SERVICE_MAX="if needed, replace with a non-zero integer"

#

Note: Admin connections to the instance are, by default, made as SYSTEM

```
#       with the default password of "manager". If you wish to use another
#       privileged account (as would be the cause with most DBaaS), then
#       change DBA_PRIV_USER and SYSDBA_PASSWD accordingly.
#### Uncomment the following if needed:
DBA_PRIV_USER=sys
SYSDBA_PASSWD=vmware123

#### The EXTERNAL_SCRIPT parameter is used by the external script calling feature of runit.sh.
#### Please see SLOB Documentation at https://kevinclosson.net/slob for more information

EXTERNAL_SCRIPT=''

##########################
#### Advanced settings:
#### The following are Hot Spot related parameters.
#### By default Hot Spot functionality is disabled (DO_HOTSPOT=FALSE).

DO_HOTSPOT=FALSE
HOTSPOT_MB=8
HOTSPOT_OFFSET_MB=16
HOTSPOT_FREQUENCY=3

#### The following controls operations on Hot Schema
#### Default Value: 0. Default setting disables Hot Schema

HOT_SCHEMA_FREQUENCY=0

#### The following parameters control think time between SLOB
#### operations (SQL Executions).
#### Setting the frequency to 0 disables think time.
```

THINK_TM_FREQUENCY=0

THINK_TM_MIN=.1

THINK_TM_MAX=.5

#########################

export UPDATE_PCT RUN_TIME WORK_LOOP SCALE WORK_UNIT LOAD_ PARALLEL_DEGREE REDO_STRESS

export DO_HOTSPOT HOTSPOT_MB HOTSPOT_OFFSET_MB HOTSPOT_FREQUENCY HOT_SCHEMA_FREQUENCY THINK_TM_FREQUENCY THINK_TM_MIN THINK_TM_ MAX

We used the following command to start the SLOB workload with 24 users:

"/u01/software/SLOB/SLOB/runit.sh -s 1 -t 1000"

Appendix B Oracle Initialization Parameter Configuration

Oracle Initialization Parameters

**.audit_file_dest='/u01/admin/DBPROD6/adump'*

**.audit_sys_operations=TRUE*

**.audit_trail='db'*

**.awr_pdb_autoflush_enabled=TRUE*

**.compatible='12.2.0.0.0'*

**.control_files='+DATA_DG/control01.ctl','+DATA_DG/control02.ctl','+DATA_DG/ control03.ctl'*

**.db_block_size=8192*

**.db_cache_advice='ON'*

**.db_create_file_dest='+DATA_DG'*

**.db_domain=''*

**.db_name='DBPROD6'*

**.db_recovery_file_dest='+DATA_DG'*

**.db_recovery_file_dest_size=200G*

**.db_writer_processes=3*

**.diagnostic_dest='/u01/admin/DBPROD6'*

**.enable_pluggable_database=true*

**.instance_name='DBPROD6'*

**.instance_number=1*

**.log_archive_format='%t_%s_%r.dbf'*

**.open_cursors=1000*

**.parallel_max_servers=100*

**.pga_aggregate_limit=12G*

**.pga_aggregate_target=6G*

**.processes=3000*

**.remote_login_passwordfile='exclusive'*

**.resource_manager_plan=''*

**.result_cache_max_result=10*

**.result_cache_max_size=3178496*

**.sga_max_size=32G*

**.sga_target=32G*

**.statistics_level='ALL'*

**.thread=1*

**.timed_os_statistics=0*

**.timed_statistics=TRUE*

**.undo_tablespace='UNDOTBS01'*

**.use_large_pages='only'*

Appendix C Oracle AWR Analysis

Improved Performance of Oracle Redo Log

vPMEMDisk Mode

Summary of Test Cases

- Test case 1: Redo logs on ASM using traditional storage
- Test case 2: Redo logs on ASM using vPMEMDisk datastore with PVSCSI Controller
- Test case 3: Redo logs on ASM using vPMEMDisk datastore with NVME Controller
- Test case 4: Redo logs on ext4 filesystem using traditional storage
- Test case 5: Redo logs on ext4 filesystem using vPMEMDisk

Test Cases 1, 2, and 3

The AWR report and SAR report for the above run times were analyzed and below results were observed. AWR metrics were analyzed for Test Case 1 which was the baseline metrics.

- 'log file switch completion' was 18.95ms
- 'log file switch (checkpoint incomplete)' was 18.40ms

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Avg Wait	% DB time	Wait Class
db file sequential read	61,887,878	1.7M	28.04ms	96.3	User I/O
enq: TX - row lock contention	24,398	40.8K	1671.22ms	2.3	Application
DB CPU		4030.3		.2	
db file scattered read	171,753	3427.6	19.96ms	.2	User I/O
log file switch completion	81,459	1543.4	18.95ms	.1	Configuration
read by other session	39,523	1167.3	29.53ms	.1	User I/O
library cache: mutex X	5,746	823.2	143.27ms	.0	Concurrency
log file switch (private strand flush incomplete)	12,616	280.3	22.22ms	.0	Configuration
library cache load lock	2,370	120.5	50.84ms	.0	Concurrency
cursor: pin S wait on X	1,064	94.8	89.09ms	.0	Concurrency

Figure 26. Top 10 Foreground Events for Test Case 1

Foreground Wait Events

- s - second, ms - millisecond, us - microsecond, ns - nanosecond
- Only events with Total Wait Time (s) >= .001 are shown
- ordered by wait time desc, waits desc (idle events last)
- %Timeouts: value of 0 indicates value was < .5%. Value of null is truly 0

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait	Waits /txn	% DB time
db file sequential read	61,887,878		1,735,554	28.04ms	60.02	96.33
enq: TX - row lock contention	24,398		40,775	1671.22ms	0.02	2.26
db file scattered read	171,753		3,428	19.96ms	0.17	0.19
log file switch completion	81,459		1,543	18.95ms	0.08	0.09
read by other session	39,523		1,167	29.53ms	0.04	0.06
library cache: mutex X	5,746		823	143.27ms	0.01	0.05
log file switch (private strand flush incomplete)	12,616		280	22.22ms	0.01	0.02
library cache load lock	2,370		120	50.84ms	0.00	0.01
cursor: pin S wait on X	1,064		95	89.09ms	0.00	0.01
row cache mutex	1,542		20	12.83ms	0.00	0.00
log file switch (checkpoint incomplete)	965		18	18.40ms	0.00	0.00

Figure 27. Foreground Events for Test Case 1

8:00 - 8:30am Baseline

08:00:01 AM	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
08:10:01 AM	nvme0n1	0.53	1.63	2.65	8.00	0.00	0.05	0.05	0.00
08:10:01 AM	sdc	1971.11	22472.59	45375.26	34.42	0.88	0.45	0.12	22.77
08:10:01 AM	sde	0.53	1.63	2.65	8.00	0.00	0.02	0.02	0.00
08:20:01 AM	nvme0n1	0.37	0.29	2.65	8.00	0.00	0.05	0.05	0.00
08:20:01 AM	sdc	2010.32	23298.06	47095.31	35.02	0.90	0.45	0.13	25.30
08:20:01 AM	sde	0.37	0.29	2.65	8.00	0.00	0.05	0.05	0.00
08:30:02 AM	nvme0n1	0.37	0.29	2.65	8.00	0.00	0.06	0.06	0.00
08:30:02 AM	sdc	1988.35	23098.54	46432.67	34.97	0.89	0.45	0.13	25.03
08:30:02 AM	sde	0.37	0.29	2.65	8.00	0.00	0.06	0.06	0.00

Figure 28. Sar Disk Output for Test Case 1

AWR metrics comparison was made between the run times of Test cases 1, 2, and 3.

Comparing Test Cases 1 and 2:

- Reduced wait times for below database event
  - o 'log file switch completion' reduced from 18.95ms to 17.42ms
  - o 'log file switch (private strand flush incomplete)' reduced from 22.22ms to 17.64ms
  - o 'log file parallel write' reduced from 511.49us to 76.04us
- Amount of work done increased
  - o Number of 'Executes (SQL) per second' increased from 576 to 587.7
  - o Number of 'Transactions per second' increased from 558.6 to 567.5

Top Timed Events

- Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
db file sequential read	User I/O	61,903,626	1,735,820.89	28.04ms	96.34	db file sequential read	User I/O	63,504,401	1,751,902.73	27.59ms	97.28
enq: TX - row lock contention	Application	24,398	40,774.53	1671.22ms	2.26	enq: TX - row lock contention	Application	15,331	25,026.05	1632.38ms	1.39
CPU time			4,030.31		0.22	CPU time			4,116.52		0.23
db file scattered read	User I/O	171,938	3,433.34	19.97ms	0.19	db file scattered read	User I/O	172,027	3,407.01	19.81ms	0.19
log file switch completion	Configuration	81,466	1,543.62	18.95ms	0.09	log file switch completion	Configuration	82,132	1,430.99	17.42ms	0.08
read by other session	User I/O	39,523	1,167.30	29.53ms	0.06	db file parallel write	System I/O	1,667,775	890.39	533.88us	0.05
db file parallel write	System I/O	1,628,249	895.15	549.76us	0.05	read by other session	User I/O	24,661	732.68	29.71ms	0.04
library cache: mutex X	Concurrency	5,746	823.23	143.27ms	0.05	control file sequential read	System I/O	11,418	289.66	25.37ms	0.02
log file parallel write	System I/O	900,475	460.59	511.49us	0.03	log file switch (private strand flush incomplete)	Configuration	13,696	241.62	17.64ms	0.01
control file sequential read	System I/O	10,976	280.47	25.55ms	0.02	library cache: mutex X	Concurrency	4,042	238.43	58.99ms	0.01
log file switch (private strand flush incomplete)	Configuration	12,616	280.32	22.22ms	0.02	log file parallel write	System I/O	938,462	71.36	76.04us	0.00

Figure 29. AWR Metrics Comparison Between Test Cases 1 and 2

9:30 - 10:00am vPMEMDisk with PVSCSI Controller

09:30:01 AM	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
09:40:01 AM	nvme0n1	0.38	0.36	2.65	8.00	0.00	0.06	0.06	0.00
09:40:01 AM	sdc	0.38	0.36	2.65	8.00	0.00	0.40	0.40	0.02
09:40:01 AM	sde	1859.38	21021.23	42623.04	34.23	0.10	0.05	0.02	3.37
09:50:01 AM	nvme0n1	0.36	0.23	2.65	8.00	0.00	0.06	0.06	0.00
09:50:01 AM	sdc	0.36	0.23	2.65	8.00	0.00	0.34	0.34	0.01
09:50:01 AM	sde	2103.90	23746.06	48090.97	34.14	0.11	0.05	0.02	3.73
10:00:01 AM	nvme0n1	0.37	0.29	2.65	8.00	0.00	0.05	0.05	0.00
10:00:01 AM	sdc	0.37	0.29	2.65	8.00	0.00	0.40	0.40	0.01
10:00:01 AM	sde	2076.89	23887.94	47156.91	34.21	0.11	0.05	0.02	3.79

Figure 30. Sar Disk Output for Test Case 3

Comparing Test Cases 2 and 3

- Reduced wait times for below database event
  - o 'log file switch completion' number of waits and total time reduced from (82,132 waits,1,430.99 seconds) to (44,899 waits, 906.44 seconds)
  - o 'log file switch (private strand flush incomplete)' reduced from 17.64ms to 17.36ms
  - o 'log file parallel write' reduced from 76.04us to 75.10us
- Amount of work done increased
  - o Number of 'Executes (SQL) per second' increased from 587.7 to 604.8
  - o Number of 'Transactions per second' increased from 567.5 to 583.1

Top Timed Events

• Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
db file sequential read	User I/O	63,504,401	1,751,902.73	27.59ms	97.28	db file sequential read	User I/O	65,146,878	1,748,858.33	26.84ms	97.09
enq: TX - row lock contention	Application	15,331	25,026.05	1632.38ms	1.39	enq: TX - row lock contention	Application	16,888	26,752.52	1584.11ms	1.49
CPU time			4,116.52		0.23	CPU time			4,197.57		0.23
db file scattered read	User I/O	172,027	3,407.01	19.81ms	0.19	db file scattered read	User I/O	172,082	3,295.56	19.15ms	0.18
log file switch completion	Configuration	82,132	1,430.99	17.42ms	0.08	library cache: mutex X	Concurrency	8,666	1,532.47	176.84ms	0.09
db file parallel write	System I/O	1,667,775	890.39	533.88us	0.05	log file switch completion	Configuration	44,899	906.44	20.19ms	0.05
read by other session	User I/O	24,661	732.68	29.71ms	0.04	db file parallel write	System I/O	1,668,200	868.00	520.32us	0.05
control file sequential read	System I/O	11,418	289.66	25.37ms	0.02	log file switch (private strand flush incomplete)	Configuration	49,811	864.72	17.36ms	0.05
log file switch (private strand flush incomplete)	Configuration	13,696	241.62	17.64ms	0.01	read by other session	User I/O	26,667	763.48	28.63ms	0.04
library cache: mutex X	Concurrency	4,042	238.43	58.99ms	0.01	control file sequential read	System I/O	10,953	274.44	25.06ms	0.02

Figure 31. AWR Metrics Comparison Between Test Cases 2 and 3

```
10:15 - 10:45am vPMEMDisk with NVME Controller

10:20:01 AM       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svctm     %util
10:30:01 AM   nvme0n1   2133.51  24735.64  50010.75     35.03      0.11      0.05      0.02      4.17
10:30:01 AM       sdc      0.37      0.36      2.64      8.00      0.00      0.34      0.34      0.01
10:30:01 AM       sde      0.37      0.36      2.64      8.00      0.00      0.01      0.01      0.00
10:40:01 AM   nvme0n1   2115.35  24403.27  48974.78     34.69      0.11      0.05      0.02      4.15
10:40:01 AM       sdc      0.36      0.23      2.65      8.00      0.00      0.34      0.34      0.01
10:40:01 AM       sde      0.36      0.23      2.65      8.00      0.00      0.03      0.03      0.00
```

Figure 32. Sar Disk Output for Test Case 3

Test Cases 4 and 5

The AWR report and SAR report for the above run times were analyzed and below results were observed. AWR metrics comparison was made between the run times of Test Cases 4 and 5.

Comparing Test Cases 4 and 5

- Reduced wait times for below database event
  - o 'log file parallel write' reduced from 1.96ms to 255.29us
  - o 'log file switch (private strand flush incomplete)' reduced from 23.22ms to 17.31ms
  - o 'log file switch completion' remained almost the same from 20.26ms to 20.34ms
- Amount of work done increased

- Number of 'Executes (SQL) per second' increased from 580.9 to 609.8
- Number of 'Transactions per second' increased from 559.5 to 587.6

Top Timed Events

• Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
db file sequential read	User I/O	62,082,230	1,734,786.80	27.94ms	96.28	db file sequential read	User I/O	65,434,480	1,741,476.50	26.61ms	96.68
enq: TX - row lock contention	Application	25,455	40,775.66	1601.87ms	2.26	enq: TX - row lock contention	Application	21,355	33,753.07	1580.57ms	1.87
CPU time			4,087.01		0.23	CPU time			4,281.32		0.24
db file scattered read	User I/O	171,724	3,634.10	21.16ms	0.20	db file scattered read	User I/O	171,859	3,281.45	19.09ms	0.18
log file parallel write	System I/O	882,652	1,731.16	1.96ms	0.10	library cache: mutex X	Concurrency	8,546	1,950.92	228.28ms	0.11
log file switch completion	Configuration	77,802	1,576.16	20.26ms	0.09	log file switch completion	Configuration	49,519	1,007.34	20.34ms	0.06
read by other session	User I/O	39,643	1,212.90	30.60ms	0.07	read by other session	User I/O	33,946	964.41	28.41ms	0.05
library cache: mutex X	Concurrency	7,618	1,015.73	133.33ms	0.06	db file parallel write	System I/O	1,681,876	871.12	517.95us	0.05
db file parallel write	System I/O	1,626,139	889.79	547.18us	0.05	log file switch (private strand flush incomplete)	Configuration	45,698	791.02	17.31ms	0.04
log file switch (private strand flush incomplete)	Configuration	15,335	356.10	23.22ms	0.02	control file sequential read	System I/O	11,875	290.68	24.48ms	0.02
control file sequential read	System I/O	10,657	270.25	25.36ms	0.01	log file parallel write	System I/O	968,312	247.20	255.29us	0.01

Figure 33. AWR Metrics Comparison Between Test Cases 4 and 5

vPMEM Mode

Summary of Test cases

- Test Case 1: ASM Redo logs using traditional storage
- Test Case 2: ASM Redo logs using vPMEM with raw mode
- Test Case 3: ASM Redo logs using vPMEM with memory mode
- Test Case 4: Filesystem Redo logs using traditional storage
- Test Case 5: Filesystem Redo logs using vPMEM memory dax mode

Test Cases 1 , 2, and 3

The AWR report and SAR report for the above run times were analyzed and below results were observed. AWR metrics were analyzed for Test Case 1 which were the baseline metrics.

- 'log file switch completion' was 21.88ms
- 'log file switch (private strand flush incomplete)' was 26.62ms
- 'log file parallel write' was 617.73us

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Avg Wait	% DB time	Wait Class
free buffer waits	1.7E+08	1.2M	7.07ms	67.2	Configuration
db file sequential read	72,153,131	490.6K	6.80ms	27.2	User I/O
enq: TX - row lock contention	18,585	26.6K	1432.01ms	1.5	Application
latch free	3,151,088	14.8K	4.68ms	.8	Other
DB CPU		9157.3		.5	
write complete waits	10,948	5381.6	491.56ms	.3	Configuration
db file scattered read	174,709	2856.9	16.35ms	.2	User I/O
library cache: mutex X	3,981	1958.8	492.05ms	.1	Concurrency
log file switch completion	54,942	1202.2	21.88ms	.1	Configuration
log file switch (private strand flush incomplete)	15,855	422	26.62ms	.0	Configuration

Figure 34. Top 10 Foreground Events for Test Case 1

AWR metrics comparison was made between the run times of Test Cases 1, 2, and 3.

Comparing Test Cases 1 and 2:

- Reduced wait times for below database event
  - o 'log file switch completion' reduced from 21.88ms to 18.87ms
  - o 'log file parallel write' completely reduced from 617.73us to 0
  - o 'log file switch (private strand flush incomplete)' slightly increased from 26.62ms to 27.23ms
- Amount of work done increased
  - o Number of 'Executes (SQL) per second' increased from 662.2 to 676.8
  - o Number of 'Transactions per second' increased from 640.4 to 656.8

Top Timed Events

• Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
free buffer waits	Configuration	171,332,486	1,211,605.10	7.07ms	67.21	free buffer waits	Configuration	171,594,070	1,212,757.44	7.07ms	67.33
db file sequential read	User I/O	72,172,772	490,742.04	6.80ms	27.22	db file sequential read	User I/O	73,716,443	486,445.69	6.60ms	27.01
enq: TX - row lock contention	Application	18,585	26,613.91	1432.01ms	1.48	enq: TX - row lock contention	Application	21,613	30,164.73	1395.68ms	1.67
latch free	Other	3,428,676	15,579.99	4.54ms	0.86	latch free	Other	3,467,502	15,700.30	4.53ms	0.87
CPU time			9,157.26		0.51	CPU time			9,192.73		0.51
write complete waits	Configuration	10,953	5,385.82	491.72ms	0.30	write complete waits	Configuration	11,100	5,351.38	482.11ms	0.30
db file scattered read	User I/O	175,510	2,863.13	16.31ms	0.16	db file scattered read	User I/O	175,044	2,795.14	15.97ms	0.16
library cache: mutex X	Concurrency	3,981	1,958.83	492.05ms	0.11	log file switch completion	Configuration	55,040	1,038.42	18.87ms	0.06
log file switch completion	Configuration	54,943	1,202.27	21.88ms	0.07	library cache: mutex X	Concurrency	5,268	430.70	81.76ms	0.02
log file parallel write	System I/O	934,909	577.52	617.73us	0.03	log file switch (private strand flush incomplete)	Configuration	9,260	252.19	27.23ms	0.01
log file switch (private strand flush incomplete)	Configuration	15,856	422.02	26.62ms	0.02	-					

Figure 35. AWR Metrics Comparison Between Test Cases 1 and 2

Comparing Test Cases 2 and 3

- Reduced wait times for below database event
  - o 'log file switch completion' reduced from 18.87ms to 18.29ms
  - o 'log file parallel write' completely reduced to 0
  - o 'log file switch (private strand flush incomplete)' reduced from 27.23ms to 21.98ms
- Amount of work done increased
  - o Number of 'Executes (SQL) per second' slightly reduced from 676.8 to 664.4
  - o Number of 'Transactions per second' slightly reduced from 656.8 to 643.2

Top Timed Events

• Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
free buffer waits	Configuration	171,594,070	1,212,757.44	7.07ms	67.33	free buffer waits	Configuration	170,704,363	1,219,690.25	7.15ms	67.62
db file sequential read	User I/O	73,716,443	486,445.69	6.60ms	27.01	db file sequential read	User I/O	72,860,956	490,793.84	6.74ms	27.21
enq: TX - row lock contention	Application	21,613	30,164.73	1395.68ms	1.67	enq: TX - row lock contention	Application	12,798	17,952.88	1402.79ms	1.00
latch free	Other	3,467,502	15,700.30	4.53ms	0.87	latch free	Other	3,617,447	16,835.16	4.65ms	0.93
CPU time			9,192.73		0.51	CPU time			9,143.84		0.51
write complete waits	Configuration	11,100	5,351.38	482.11ms	0.30	write complete waits	Configuration	10,907	5,359.83	491.41ms	0.30
db file scattered read	User I/O	175,044	2,795.14	15.97ms	0.16	library cache: mutex X	Concurrency	5,369	2,986.49	556.25ms	0.17
log file switch completion	Configuration	55,040	1,038.42	18.87ms	0.06	db file scattered read	User I/O	175,283	2,786.48	15.90ms	0.15
library cache: mutex X	Concurrency	5,268	430.70	81.76ms	0.02	log file switch completion	Configuration	41,379	756.91	18.29ms	0.04
log file switch (private strand flush incomplete)	Configuration	9,260	252.19	27.23ms	0.01	log file switch (private strand flush incomplete)	Configuration	17,893	393.25	21.98ms	0.02

Figure 36. AWR Metrics Comparison Between Test Cases 2 and 3

Test Cases 4 and 5

The AWR report and SAR report for the above run times were analyzed and below results were observed. AWR metrics were analyzed for Test case 4 which were the baseline metrics.

- 'log file switch completion' was 22.90ms
- 'log file switch (private strand flush incomplete)' was 23.87ms

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Avg Wait	% DB time	Wait Class
free buffer waits	1.7E+08	1.2M	7.26ms	66.9	Configuration
db file sequential read	70,622,618	500.5K	7.09ms	27.8	User I/O
enq: TX - row lock contention	15,866	23.2K	1460.41ms	1.3	Application
latch free	3,144,429	14.5K	4.61ms	.8	Other
DB CPU		8935.1		.5	
write complete waits	10,401	5131.3	493.35ms	.3	Configuration
db file scattered read	174,656	2880.9	16.49ms	.2	User I/O
library cache: mutex X	5,036	1722.4	342.01ms	.1	Concurrency
log file switch (private strand flush incomplete)	41,713	995.5	23.87ms	.1	Configuration
log file switch completion	18,929	433.4	22.90ms	.0	Configuration

Figure 37. Top 10 Foreground Events for Test Case 4

```
3:28-3:58 pm ext4 filesystem

[root@oracle122-rhel sa]# sar -f sa25 -dp  | more
Linux 3.10.0-862.3.3.el7.x86_64 (oracle122-rhel.vslab.local)    07/25/2018      _x86_64_        (12 CPU)
....
03:20:01 PM       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svctm     %util
03:30:01 PM       sdc    256.94      0.20  35990.36    140.07      0.54      2.10      0.27      6.92
03:40:01 PM       sdc   1878.24      0.00  68568.37     36.51      0.78      0.42      0.31     58.40
03:50:01 PM       sdc   1832.29      0.00  65923.28     35.98      0.77      0.42      0.31     57.63
04:00:01 PM       sdc   1679.51      0.00  59997.96     35.72      0.71      0.42      0.32     53.12
```

Figure 38. Sar Disk Output for Test Case 4

The AWR report and SAR report for the above run times were analyzed and below results were observed. AWR metrics comparison was made between the run times of Test Cases 4 and 5.

Comparing Test Cases 4 and 5

- Reduced wait times for below database event
  - o 'log file switch completion' reduced from 22.90ms to 18.89ms
  - o 'log file switch (private strand flush incomplete)' reduced from 23.87ms to 20.76ms
- Amount of work done increased
  - o Number of 'Executes (SQL) per second' increased from 646.8 to 680.6
  - o Number of 'Transactions per second' increased from 625.8 to 660.9

Top Timed Events

- Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
free buffer waits	Configuration	166,218,602	1,206,518.14	7.26ms	66.93	free buffer waits	Configuration	169,150,568	1,205,315.68	7.13ms	66.91
db file sequential read	User I/O	70,627,617	500,510.12	7.09ms	27.77	db file sequential read	User I/O	73,927,484	483,242.91	6.54ms	26.83
enq: TX - row lock contention	Application	15,866	23,170.92	1460.41ms	1.29	enq: TX - row lock contention	Application	27,382	37,053.97	1353.22ms	2.06
latch free	Other	3,409,610	15,285.01	4.48ms	0.85	latch free	Other	3,536,781	16,225.30	4.59ms	0.90
CPU time			8,935.06		0.50	CPU time			9,254.25		0.51
write complete waits	Configuration	10,402	5,132.07	493.37ms	0.28	write complete waits	Configuration	11,266	5,528.04	490.68ms	0.31
db file scattered read	User I/O	175,074	2,884.17	16.47ms	0.16	db file scattered read	User I/O	175,495	2,742.07	15.62ms	0.15
log file parallel write	System I/O	856,349	2,024.76	2.36ms	0.11	log file switch completion	Configuration	42,633	805.18	18.89ms	0.04
library cache: mutex X	Concurrency	5,036	1,722.37	342.01ms	0.10	library cache: mutex X	Concurrency	4,049	750.28	185.30ms	0.04
log file switch (private strand flush incomplete)	Configuration	41,713	995.51	23.87ms	0.06	log file switch (private strand flush incomplete)	Configuration	17,462	362.57	20.76ms	0.02
-log file switch completion	Configuration	18,929	433.43	22.90ms	0.02	-log file parallel write	System I/O	1,025,193	29.02	28.30us	0.00

Figure 39. AWR Metrics Comparison Between Test Cases 4 and 5

Accelerating Performance Using Oracle Smart Flash Cache

vPMEMDisk Mode

Summary of Test Cases

- Test Case 1: No caching
- Test Case 2: Flash cache using ASM on vPMEMDisk datastore with PVSCSI Controller
- Test Case 3: Flash cache using ext4 filesystem on vPMEMDisk datastore with PVSCSI Controller

Test Cases 1, 2, and 3

The AWR report for the above run times were analyzed and below results were observed. AWR metrics comparison was made between the run times of Test Cases 1 and 2. AWR metrics comparison was made between the run times of Test Cases 1 and 3.

Comparing Test Cases 1 and 2

- Faster read times for single block reads from Smart Flash cache
  - o "db flash cache single block physical read" event has 3,608,241 waits with Average time of 80.25us
  - o Good 'flash cache hit ratio' for the amount of cache allocated as reads were also coming from the flash cache, not just the disks
- Amount of work done increased
  - o Number of 'Executes (SQL) per second' increased from 583.4 to 611.8
  - o Number of 'Transactions per second' increased from 559.9 to 589.8
  - o Logical read (blocks) increased from 39,186.3 to 41,272.3
  - o '%Idle time' and '%IO Wait Time' reduced, '%User Time' & '%System Time' increased, indicating more work done

Host Configuration Comparison

	1st	2nd	Diff	%Diff
Number of CPUs:	12	12	0	0.0
Number of CPU Cores:	12	12	0	0.0
Number of CPU Sockets:	12	12	0	0.0
Physical Memory:	64152.8M	64152.8M	0M	0.0
Load at Start Snapshot:	.59	.65	.06	10.2
Load at End Snapshot:	771.66	836.09	64.43	8.3
%User Time:	16.27	21.17	4.89	30.1
%System Time:	6.07	7.13	1.06	17.5
%Idle Time:	75.62	67.94	-7.67	-10.2
%IO Wait Time:	73.21	65.56	-7.65	-10.4

Figure 40. Host Configuration Comparison Between Test Cases 1 and 2

Top Timed Events

- Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
db file sequential read	User I/O	62,341,841	1,741,207.95	27.93ms	96.62	db file sequential read	User I/O	62,895,828	1,760,548.06	27.99ms	97.76
enq: TX - row lock contention	Application	21,425	35,541.20	1658.87ms	1.97	enq: TX - row lock contention	Application	12,696	19,814.47	1560.69ms	1.10
CPU time			4,063.39		0.23	CPU time			4,436.18		0.25
db file scattered read	User I/O	171,213	3,430.05	20.03ms	0.19	db file scattered read	User I/O	160,972	3,209.75	19.94ms	0.18
log file switch completion	Configuration	75,851	1,445.29	19.05ms	0.08	log file switch (private strand flush incomplete)	Configuration	77,637	1,523.30	19.62ms	0.08
library cache: mutex X	Concurrency	6,433	1,345.04	209.08ms	0.07	read by other session	User I/O	18,059	549.30	30.42ms	0.03
read by other session	User I/O	34,371	1,036.54	30.16ms	0.06	log file parallel write	System I/O	960,787	498.74	519.09us	0.03
db file parallel write	System I/O	1,627,062	889.23	546.53us	0.05	log file switch completion	Configuration	18,674	382.57	20.49ms	0.02
log file parallel write	System I/O	929,988	474.82	510.57us	0.03	db flash cache single block physical read	User I/O	3,608,241	289.58	80.25us	0.02
log file switch (private strand flush incomplete)	Configuration	18,466	383.12	20.75ms	0.02	control file sequential read	System I/O	11,508	289.18	25.13ms	0.02
control file sequential read	System I/O	11,504	290.28	25.23ms	0.02	library cache: mutex X	Concurrency	4,386	194.98	44.46ms	0.01
-						db file parallel write	System I/O	6,049,163	169.06	27.95us	0.01

Figure 41. AWR Metrics Comparison Between Test Cases 1 and 2

```
SQL> SELECT * FROM v$flashfilestat;

FLASHFILE# NAME                                                 BYTES    ENABLED SINGLEBLKRDS SINGLEBLKRDTIM_MICRO     CON_ID
---------- -------------------------------------------------- ---------- ---------- ------------ -------------------- ----------
         1 +FLASH_DG/flashfile                                6.6572E+10          1      3609579               289610797          0

SQL>
```

Figure 42. Flash Cache Statistics

Comparing Test Cases 1 and 3:

- Faster read times for single block reads from Smart Flash cache
  - "db flash cache single block physical read" event has Average time of 73.64us
  - Good 'flash cache hit ratio' for the amount of cache allocated as reads were also coming from the flash cache, not just the disks
- Amount of work done increased
  - Number of 'Executes (SQL) per second' increased from 583.4 to 612.1
  - Number of 'Transactions per second' increased from 559.9 to 590.6

- Logical read (blocks) increased from 39,186.3 to 41,581.8
- '%Idle time' and '%IO Wait Time' reduced, '%User Time' & '%System Time' increased, indicates more work done

Top Timed Events

- Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
db file sequential read	User I/O	62,341,841	1,741,207.95	27.93ms	96.62	db file sequential read	User I/O	63,766,375	1,732,225.98	27.17ms	96.18
enq: TX - row lock contention	Application	21,425	35,541.20	1658.87ms	1.97	enq: TX - row lock contention	Application	29,084	46,227.95	1589.46ms	2.57
CPU time			4,063.39		0.23	CPU time			4,357.04		0.24
db file scattered read	User I/O	171,213	3,430.05	20.03ms	0.19	db file scattered read	User I/O	160,921	3,416.98	21.23ms	0.19
log file switch completion	Configuration	75,851	1,445.29	19.05ms	0.08	db flash cache write	User I/O	1,181,661	1,548.13	1.31ms	0.09
library cache: mutex X	Concurrency	6,433	1,345.04	209.08ms	0.07	log file switch (private strand flush incomplete)	Configuration	66,741	1,273.41	19.08ms	0.07
read by other session	User I/O	34,371	1,036.54	30.16ms	0.06	read by other session	User I/O	37,517	1,095.03	29.19ms	0.06
db file parallel write	System I/O	1,627,062	889.23	546.53us	0.05	log file switch completion	Configuration	28,072	558.49	19.89ms	0.03
log file parallel write	System I/O	929,988	474.82	510.57us	0.03	log file parallel write	System I/O	959,898	498.49	519.32us	0.03
log file switch (private strand flush incomplete)	Configuration	18,466	383.12	20.75ms	0.02	library cache: mutex X	Concurrency	3,302	368.93	111.73ms	0.02
-						-db file parallel write	System I/O	5,570	0.98	176.47us	0.00

Figure 43. Timed Events Comparison Between Test Cases 1 and 3

Wait Events

- Ordered by absolute value of 'Diff' column of '% of DB time' descending (idle events last)

		% of DB time			# Waits/sec (Elapsed Time)			Total Wait Time (sec)			Avg Wait Time		
Event	Wait Class	1st	2nd	Diff	1st	2nd	%Diff	1st	2nd	%Diff	1st	2nd	%Diff
enq: TX - row lock contention	Application	1.97	2.57	0.59	11.60	15.75	35.78	35,541.20	46,227.95	30.07	1658.87ms	1589.46ms	-4.18
db file sequential read	User I/O	96.62	96.18	-0.44	33,746.96	34,527.42	2.31	1,741,207.95	1,732,225.98	-0.52	27.93ms	27.17ms	-2.72
db flash cache write	User I/O	0.00	0.09	0.09	0.00	639.83	100.00	0.00	1,548.13	100.00	.00ns	1.31ms	100.00
library cache: mutex X	Concurrency	0.07	0.02	-0.05	3.48	1.79	-48.56	1,345.04	368.93	-72.57	209.08ms	111.73ms	-46.56
log file switch (private strand flush incomplete)	Configuration	0.02	0.07	0.05	10.00	36.14	261.40	383.12	1,273.41	232.38	20.75ms	19.08ms	-8.05
db file parallel write	System I/O	0.05	0.00	-0.05	880.76	3.02	-99.66	889.23	0.98	-99.89	546.53us	176.47us	-67.27
log file switch completion	Configuration	0.08	0.03	-0.05	41.06	15.20	-62.98	1,445.29	558.49	-61.36	19.05ms	19.89ms	4.41
db flash cache single block physical read	User I/O	0.00	0.01	0.01	0.00	965.95	100.00	0.00	131.36	100.00	.00ns	73.64us	100.00

Figure 44. Waits Events Comparison Between Test Cases 1 and 3

```
SQL> SELECT * FROM v$flashfilestat;

FLASHFILE# NAME                                                  BYTES    ENABLED SINGLEBLKRDS SINGLEBLKRDTIM_MICRO     CON_ID
---------- -------------------------------------------------- ---------- ---------- ------------ -------------------- ----------
         1 /flashcache/flashfile                              6.6572E+10          1      1785777             131461248          0

SQL>
```

Figure 45. Flash Cache Statistics

Potential Reduction in Oracle Licensing

vPMEMDisk Mode

Summary of Test Cases

- Test Case 1: Redo log group on traditional storage
- Test Case 2: Redo log group on vPMEMDisk datastore with NVME Controller

Test Cases 1 and 2

The AWR report for the above run times were analyzed and below results were observed. AWR metrics comparison was made between the run times of Test Cases 1 and 2.

Comparing Test Case 1 and 2

- Reduction in overall database events wait times
- Reduction in 'IO Wait Time' and 'Idle Time'
- Increase in 'System Time' and 'User Time' (indicative of more work done)
- Amount of work done increased
  - Number of 'Executes (SQL) per second' increased from 597.4 to 614.6
  - Number of 'Transactions per second' increased from 577.7 to 588.5

Top Timed Events

- Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
db file sequential read	User I/O	63,536,194	1,721,969.76	27.10ms	95.59	db file sequential read	User I/O	65,331,905	1,735,644.21	26.57ms	96.31
enq: TX - row lock contention	Application	32,677	52,992.92	1621.72ms	2.94	enq: TX - row lock contention	Application	29,414	46,022.00	1564.63ms	2.55
CPU time			4,125.98		0.23	CPU time			4,092.70		0.23
db file scattered read	User I/O	173,792	3,513.07	20.21ms	0.20	db file scattered read	User I/O	174,346	3,248.92	18.63ms	0.18
read by other session	User I/O	53,185	1,539.06	28.94ms	0.09	log file switch (private strand flush incomplete)	Configuration	73,591	1,542.35	20.96ms	0.09
library cache: mutex X	Concurrency	6,377	1,316.06	206.38ms	0.07	read by other session	User I/O	46,565	1,317.14	28.29ms	0.07
log file switch (private strand flush incomplete)	Configuration	47,169	1,021.82	21.66ms	0.06	library cache: mutex X	Concurrency	9,277	1,198.15	129.15ms	0.07
log file switch completion	Configuration	47,124	955.04	20.27ms	0.05	db file parallel write	System I/O	1,369,072	747.63	546.09us	0.04
db file parallel write	System I/O	1,657,741	881.47	531.73us	0.05	log file switch completion	Configuration	21,286	494.79	23.24ms	0.03
log file parallel write	System I/O	952,756	485.44	509.51us	0.03	control file sequential read	System I/O	11,911	289.63	24.32ms	0.02
control file sequential read	System I/O	10,801	265.77	24.61ms	0.01	log file parallel write	System I/O	928,378	89.63	96.55us	0.00

Figure 46. Timed Events for Test Cases 1 and 2

Load Profile

	1st per sec	2nd per sec	%Diff	1st per txn	2nd per txn	%Diff
DB time:	975.8	974.8	-0.1	1.7	1.7	-1.8
CPU time:	2.2	2.2	-0.9	0.0	0.0	0.0
Background CPU time:	0.5	0.5	-3.8	0.0	0.0	0.0
Redo size (bytes):	11,729,725.3	11,945,920.7	1.8	20,304.4	20,300.0	-0.0
Logical read (blocks):	40,377.7	41,154.1	1.9	69.9	69.9	0.1
Block changes:	74,352.5	75,754.6	1.9	128.7	128.7	0.0
Physical read (blocks):	35,307.1	36,114.6	2.3	61.1	61.4	0.4
Physical write (blocks):	34,467.9	35,204.5	2.1	59.7	59.8	0.3
Read IO requests:	34,918.4	35,720.4	2.3	60.4	60.7	0.4
Write IO requests:	33,077.4	33,744.5	2.0	57.3	57.3	0.1
Read IO (MB):	275.8	282.1	2.3	0.5	0.5	0.0
Write IO (MB):	269.3	275.0	2.1	0.5	0.5	0.0
IM scan rows:	0.0	0.0	0.0	0.0	0.0	0.0
Session Logical Read IM:	0.0	0.0	0.0	0.0	0.0	0.0
User calls:	15.8	16.0	1.5	0.0	0.0	0.0
Parses (SQL):	14.5	16.6	14.6	0.0	0.0	0.0
Hard parses (SQL):	0.3	0.7	106.1	0.0	0.0	0.0
SQL Work Area (MB):	1.5	1.6	8.8	0.0	0.0	8.8
Logons:	0.8	0.9	8.6	0.0	0.0	0.0
Executes (SQL):	597.4	614.6	2.9	1.0	1.0	1.0
Transactions:	577.7	588.5	1.9			

Figure 48. Load Profile for Test Cases 1 and 2

Figure 49. CPU Listing for Test Case 1

Figure 50. CPU Listing for Test Case 2

Reference

White Paper

For additional information, see the following white papers:

- Oracle Databases on VMware Best Practices Guide
- vSphere Persistent Memory

Product Documentation

For additional information, see the following product documentation:

- Oracle 12c Database Online Documentation
- vSphere Persistent Memory

Other Documentation

For additional information, see the following document:

- SLOB Resources

Acknowledgements

Author: Sudhir Balasubramanian, Staff Solution Architect, works in the Cloud Platform Business Unit (CPBU). Sudhir specializes in the virtualization of Oracle business-critical applications. Sudhir has more than 20 years' experience in IT infrastructure and database technology, working as the Principal Oracle DBA and Architect for large enterprises focusing on Oracle, EMC storage, and Unix/Linux technologies. Sudhir holds a master's in computer science from San Diego State University and is one of the authors of "Virtualize Oracle Business Critical Databases," a comprehensive authority for Oracle DBAs on the subject of Oracle and Linux on vSphere. Sudhir is a VMware vExpert, Ex-Member of the CTO Ambassador Program, and an Oracle ACE.

Thanks to the following for their reviews and inputs:

- Don Sullivan: Product Line Marketing Manager for Business-Critical Applications
- Charu Chaubal: Director, Cloud Platform Technical Marketing

Thanks to the following for their infrastructure assistance:

- Mohan Potheri: Sr. Solutions Architect, Technical Marketing
- Micron Technology

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