WHITE PAPER–NOVEMBER 2022

Virtualizing High Performance Computing (HPC) on VMware vSphere 7 Using Intel® Select Solution for HPC

vmware®

Overview

Introduction

Enterprise R&D organizations employ High Performance Computing (HPC) to develop innovative products, shorten design times, and lower overall costs. HPC aggregates computing power across multiple systems, leveraging specialized parallel computing software to solve highly complex problems in the required timeframe. Investment in HPC powers new inventions and accelerates innovations across industries—from drug discovery, aerospace, and climate research, to manufacturing and finance.

This document aims to provide enterprises with a reference architecture to deploy HPC workloads on the VMware vSphere® platform. We partner with Intel to demonstrate the deployment and performance of Intel® Select Solution for HPC on vSphere. We present a validated architecture and demonstrate how it can be efficiently deployed on the vSphere platform to leverage the benefits on VMware virtualized infrastructure. Enterprises can leverage this reference architecture to deploy HPC applications and workloads on vSphere platforms to replace or augment their existing HPC environments and help accelerate the time to actionable insights.

Note: This is a co-branding solution paper between VMware and Intel.

Business Challenges and Values

Enterprise IT companies leverage VMware to streamline and operationalize most of their IT workloads. These companies have traditionally relied on specialized HPC teams to host and run their HPC workloads. Advancements in VMware vSphere make it easy to leverage and virtualize the same optimized infrastructure components (compute accelerators, RDMA network, and low-latency interconnects) needed to meet the computational and scaling needs of HPC workloads. Existing VMware customers can leverage vSphere for HPC workloads and get the advantages of using consistent infrastructure for running their HPC and AI workloads.

R&D workloads such as modeling and simulation, virtual prototyping, machine learning, and analytics have characteristics that are different from the needs of traditional IT development teams. This paper demonstrates a validated and automated approach to deploying an HPC cluster on vSphere to meet the performance requirements of these computation-intensive and parallel workloads. The content can be leveraged by IT infrastructure teams to deploy infrastructure for their HPC customers. This approach provides the following advantages:

- HPC clusters can be spun up quickly leveraging the same management and operation interface used by VI administrators, enabling R&D teams to innovate and outpace the competition.
- The solution is compliant with industry standards and best practices for HPC by leveraging open source and community driven HPC software stacks.
- The extensive compatibility matrix for vSphere allows the use of hardware and software components important for HPC to be used as needed for the workload.

Audience

This tutorial is intended for IT administrators and product evaluators who are familiar with VMware vSphere ,vSAN ,and VMware vCenter Server. Familiarity with networking and storage in a virtual environment is assumed. Knowledge of other technologies, such as OpenHPC is also helpful.

Technology Overview

The HPC cluster on vSphere 7 (referred to as vHPC in the rest of this document) leverages VMware vSphere and vSAN to provide a performant virtualized infrastructure for deployment of HPC clusters efficiently using the vHPC-toolkit. vHPC leverages OpenHPC as it is widely used in the HPC community and is part of the Intel Select Solution for HPC. Intel oneAPI Base & HPC Toolkit provides optimized libraries and components used for the development and execution of HPC applications.

- VMware vSphere
- VMware vSAN
- vHPC Toolkit
- OpenHPC
- Intel oneAPI Base & HPC Toolkit

VMware vSphere

VMware vSphere is VMware's virtualization platform, which transforms data centers into aggregated computing infrastructures that include CPU, storage, and networking resources. vSphere manages these infrastructures as a unified operating environment and provides operators with the tools to administer the data centers that participate in that environment. The two core components of vSphere are ESXi™ and vCenter Server®. ESXi is the hypervisor platform used to create and run virtualized workloads. vCenter Server is the management plane for the hosts and workloads running on the ESXi hosts.

VMware vSAN

VMware vSAN is the industry-leading software powering VMware's software-defined storage and HCI solution. vSAN helps customers evolve their data center without risk, control IT costs, and scale to tomorrow's business needs. vSAN, native to the market-leading hypervisor, delivers flash-optimized, secure storage for all your critical vSphere workloads, and is built on industry-standard x86

servers and components that help lower TCO in comparison to traditional storage. It provides the agility to scale IT easily and offers the industry's first native HCI encryption.

vSAN simplifies Day 1 and Day 2 operations, and customers can quickly deploy and extend cloud infrastructure and minimize maintenance disruptions. vSAN helps modernize Hyperconverged Infrastructure (HCI) by providing administrators a unified storage control plane for both block and file protocols and provides significant enhancements that make it a great solution for traditional virtual machines as well as cloud-native applications. vSAN helps reduce the complexity of monitoring and maintaining infrastructure and enables administrators to rapidly provision a file share in a single workflow for Kubernetes-orchestrated cloud-native applications.

vHPC Toolkit

vHPC toolkit is a flexible, extensible, and easy-to-use toolkit developed by VMware OCTO, that allows users to deploy and manage virtualized clusters for HPC and machine learning (ML) environments. It leverages vSphere APIs to help vSphere perform common vSphere tasks related to a vHPC cluster, such as VM cloning, setting Latency Sensitivity, sizing vCPUs and memory, creating SRIOV distributed virtual switch (DVS) network, using assignable hardware (AH) accelerators (RDMA interconnects, GPU and FPGA).

OpenHPC

OpenHPC repo contains HPC libs and software (like Slurm, compilers, MPI, and others). It integrates a multitude of components that are commonly used in HPC systems and are freely available for open-source distribution. Packages provided by OpenHPC have been pre-built with HPC integration in mind intending to provide re-usable building blocks for the HPC community. The community includes representation from a variety of sources including software vendors, equipment manufacturers, research institutions, supercomputing sites, and others.

Intel oneAPI Base & HPC Toolkit

Intel oneAPI Base Toolkit is an open, cross-architecture programming model that frees developers to use a single code base across multiple architectures. The result is accelerated compute without vendor lock-in. The Intel oneAPI HPC Toolkit (HPC Kit) delivers what developers need to build, analyze, optimize, and scale HPC applications with the latest techniques in vectorization, multithreading, multi-node parallelization, and memory optimization. This toolkit is an add-on to the Intel oneAPI Base Toolkit, which is required for full functionality. It also includes access to the Intel Distribution for Python programming language, the Intel oneAPI DPC++/C++ Compiler, powerful data-centric libraries, and advanced analysis tools.

Solution Configuration

This section describes the hardware configurations, software components, and HPC specific configurations that were adopted to create a performant vHPC platform. In this section we will cover:

- Architecture diagram and network design
- Hardware resources for HPC workload cluster
- Software resources
- vSAN configuration
- Environment settings

Architecture Diagram and Network Design

Figure 1 shows the architecture diagram of a typical vHPC system, and we will cover three critical categories for HPC infrastructure: compute, storage, and network.

For computation, a typical vHPC cluster includes one FrontEnd VM that manages the vHPC cluster consisting of multiple compute VMs that run the workloads. The size and number of compute VMs vary depending on the requirements of workloads or organization business needs. In our design, each ESXi host is deployed with one Compute VM that can utilize all the available CPU and memory resources on the host. Alternatively, multiple smaller VMs can be deployed on a single host based on the desired use cases or application requirements. We recommend that the FrontEnd VM to be hosted in a separate vSphere cluster with other management components. In our design, it is hosted in the management cluster which is running on another vSphere cluster.

For storage services, vSAN datastore is used to host all virtual machines. Network File System (NFS) is used to provide shared storage service for the FrontEnd and Compute VMs in the vHPC cluster. NFS can be designed by two options: a hyperconverged solution where the NFS file share is provided by the vSAN file service, and a more traditional approach using an external NFS storage solution.

For network, we configured a VMware vSphere Distributed Switch™ (VDS) for management networks with four Distributed PortGroups (PGs) configured with different VLANs:

- ESXi Management PortGroup for vSphere management traffic
- vSAN PortGroup for vSAN traffic
- vMotion PortGroup for vMotion traffic

- VM Management PortGroup for vHPC cluster management traffic

Remote Direct Memory Access (RDMA) is widely used for HPC message passing and can be used in a virtual environment as well. To support HPC applications that leverage parallel computing and are communication intensive, we recommend using a high performance RDMA interconnect for workload network. The use of *RDMA in vSphere* allows for accessing memory data from one host to another. This greatly improves throughput and performance while lowering latency.

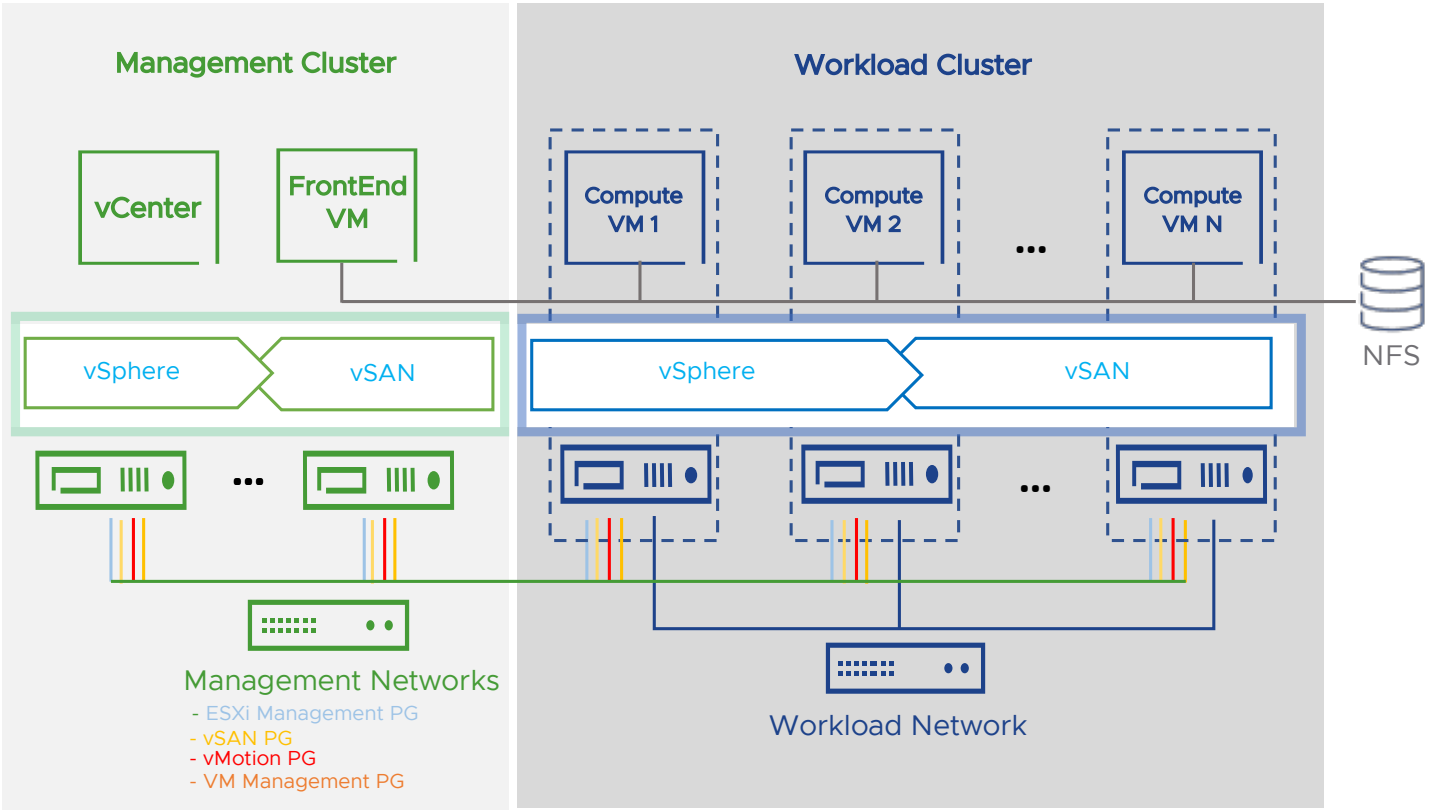


Figure 1. Architecture Diagram for vHPC

Figure 2 and Figure 3 demonstrate two configuration options in vSphere for the Workload Network (RDMA interconnect) between the compute VMs:

- Assign the network adapter via DirectPath I/O
- Assign the network adapter via SR-IOV Passthrough

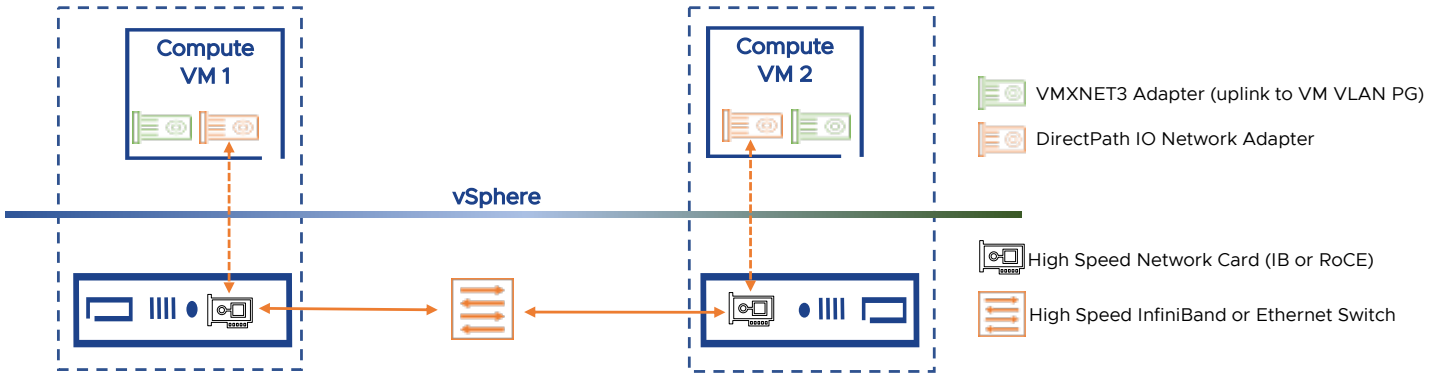


Figure 2. Assign the Network Adapter via DirectPath I/O

Configuring the RDMA interconnect via DirectPath I/O is more straightforward as no distributed virtual switch creation is required. It also achieves near bare-metal performance. However, the hardware must be used by a single VM exclusively.

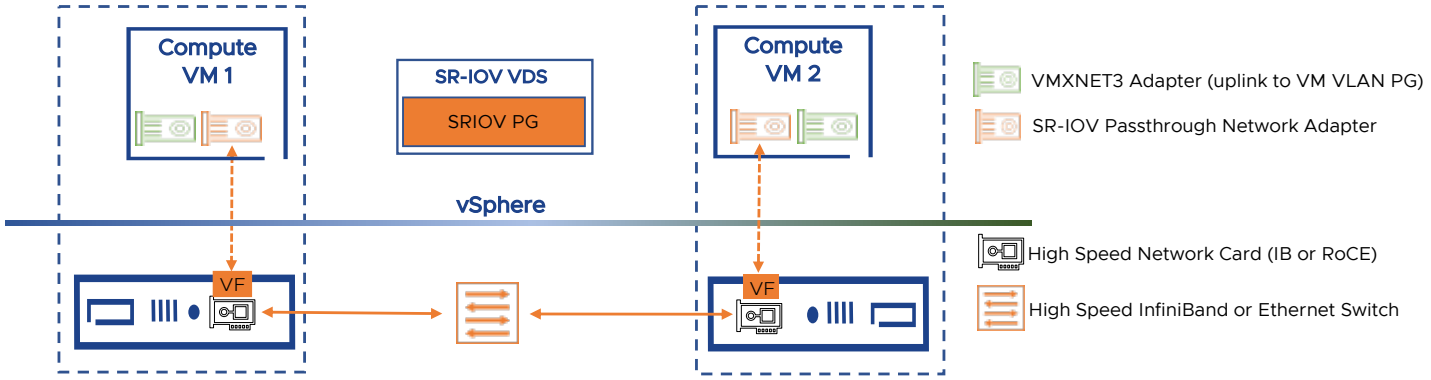


Figure 3. Assign the Network Adapter via SR-IOV Passthrough

Configuring the RDMA interconnect via SR-IOV involves creating a new SR-IOV VDS (shown in Figure 3) and enabling SR-IOV in BIOS settings and network adapter settings (firmware and driver). This layer of indirection may introduce very subtle overhead. However, it enables the underlying hardware Virtual Functions to be shared among multiple VMs. This configuration is required when running more than one compute VMs per host.

Hardware Resources for HPC Workload Cluster

Figure 4 shows the hardware topology of HPC workload cluster. We used four **Dell PowerEdge R740XD vSAN ready nodes** for the workload cluster. The workload network uses **Mellanox SB7800 100 Gb InfiniBand switch,** and the MTU is set to 4096. Updating the OS image of the InfiniBand switch to the latest version is also required. No additional switch settings are required for InfiniBand. If there is a preference to use Ethernet, RDMA over Converged Ethernet (RoCE) can also be leveraged to meet the low latency and high bandwidth requirements of HPC applications. For RoCE, we recommend an MTU of 9000, and enabling Priority Flow control (PFC) and Explicit Congestion Notification (ECN) on the Ethernet switch configured for RoCE. The management network uses a separate pair of **Dell PowerSwitch S5296F** top of rack (ToR) switches, which provides redundancy and are connected by a virtual link trunking interconnect (VLTi). Table 1 lists the hardware resources of each server.

Table 1: Hardware Resources

PROPERTY	SPECIFICATION
CPU	2 x Intel® X®(R) Gold 6248R CPU @ 3.00GHz 24 Cores
RAM	384 GB (24 * 16GB RDIM)
RDMA NIC	100 GbE NVIDIA Mellanox ConnectX-5 VPI Dual Ports MT27800
Management NIC	25 GbE NVIDIA Mellanox ConnectX-4 Lx Dual Ports MT27710
Storage adapter	1 x Dell HBA330 Adapter
Disks	2 x TOSHIBA Disk 372.61 GB as vSAN Cache Disk 2 x WDC Disk 1.75 TB as vSAN Capacity Disk

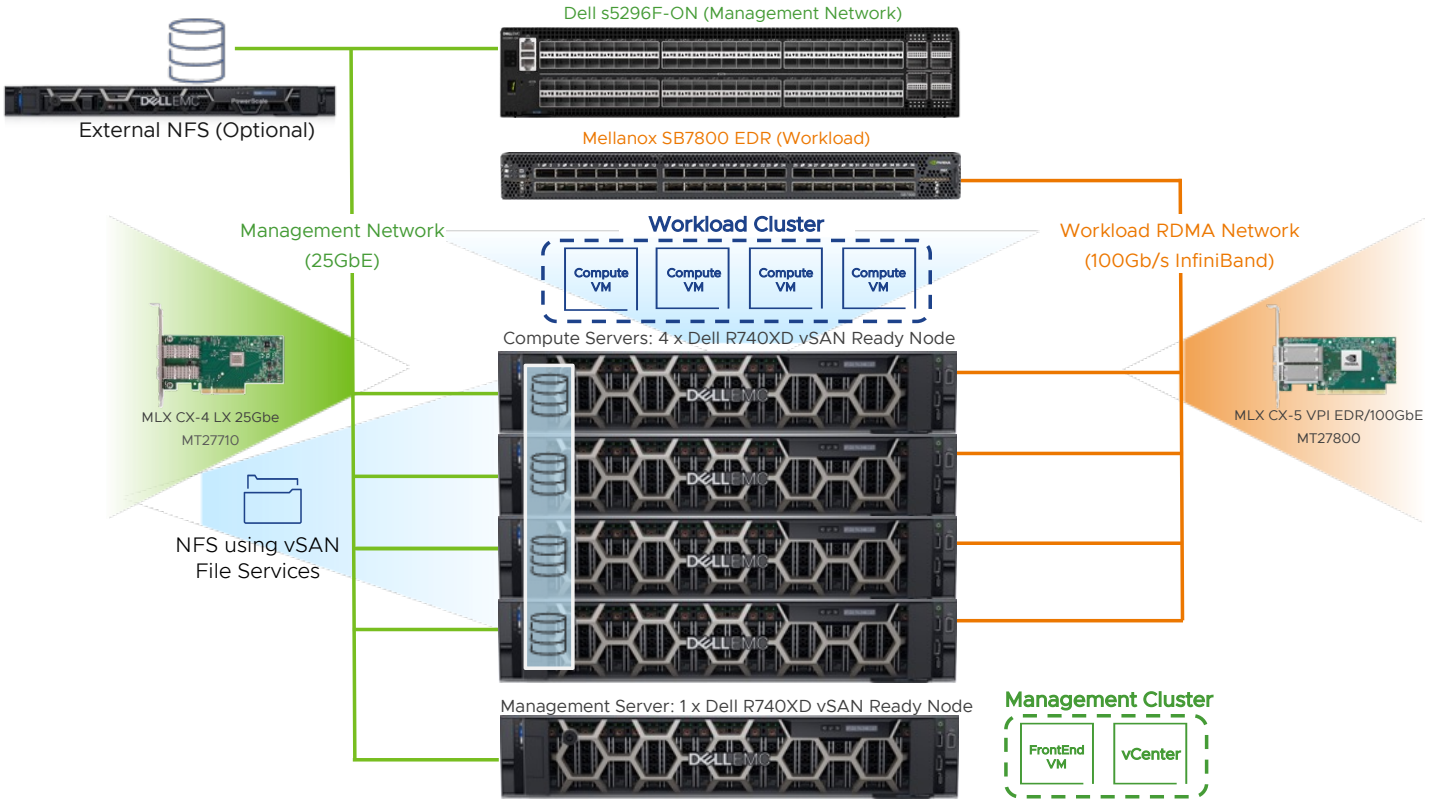


Figure 4. Hardware Topology of HPC Workload Cluster

Software Resources

Table 2 shows the software resources used in this solution.

Table 2: Software Resources

SOFTWARE	VERSION	PURPOSE
VMware vSphere	7.0.2 (18455184)	Bring the benefits of the cloud to on-premises workloads. vSphere supercharges performance, enhances operational efficiency, and accelerates innovation.
Red Hat Enterprise Linux (RHEL)	8.1	Operating System for the FrontEnd VM and Compute VM.
xCAT	2.16.2	An open-source tool for automating deployment, scaling, and management of bare metal servers and virtual machines.
OpenHPC	2.4	A collaborative, community effort that initiated from a desire to aggregate common ingredients required to deploy and manage High Performance Computing (HPC) Linux clusters including provisioning tools, resource management, I/O clients, development tools, and a variety of scientific libraries.
Intel oneAPI Base & HPC Toolkit	2022.1.2	A core set of tools and libraries for developing high-performance, data-centric applications across diverse architectures.
Intel Cluster Checker (CLCK)	2021 Update 6 (build 20220318)	Intel Cluster Checker provides tools to collect data from the cluster, analysis of the collected data, and provides a clear report of the analysis. It helps to quickly identify issues and improve utilization of resources.

OpenMPI	4.1.2	An open-source Message Passing Interface.
GCC	9.3.0	GNU Compiler Collection
Unified Communication X (UCX)	1.12.0	An open-source, production-grade communication framework for data-centric and high-performance applications.
Spack	0.17.1	Spack is a package management tool designed to support multiple versions and configurations of software on a wide variety of platforms and environments.
OSU MicroBenchmark	5.7.1	Measures the performance of an MPI implementation.
Mellanox Firmware tools	MFT & NMST 4.18.1	A set of firmware management tools used to generate a standard or customized NDIVIA firmware image Querying for firmware information and burn a firmware image.
Native Mellanox (NMLX) Driver	4.21.71.101	Mellanox native ESXi drivers enable industry-leading performance and efficiency as non-virtualized environments using hardware offloads such as RDMA over Converged Ethernet (RoCE) on VMware vSphere.
OpenFabrics Enterprise Distribution (OFED)	5.4-3.0.3.0	An open-source software for RDMA and kernel bypass applications.

vSAN Configuration

vSAN was configured per the following configurations:

Two vSAN disk groups were configured per host. For each disk group, one 372.6 GB TOSHIBA flash disk was used for the cache tier, and one 1.75 TB WDC flash disk was used for the capacity tier. The total capacity of the vSAN datastore was 14 TB. Multiple disk groups are preferred to get performance advantages and reduce failure domains within a host. Refer to *VMware vSAN Design Guide* for more information about vSAN cluster design considerations.

vSAN data Deduplication and Compression (DD&C) services were disabled in our case as it costs additional CPU cycles and potentially impact vSAN performance. Users can choose to enable DD&C to improve the space efficiency depending on the workloads. Refer to *vSAN Space Efficiency Technologies* for more details.

vHPC Compute VMs were deployed on vSAN datastore with the vSAN default storage policy (RAID 1 FTT=1) as shown in Figure 5 and Figure 6. RAID1 FTT=1 (Mirroring) uses more disk space but provides better performance in general. RAID 5 FTT=1 (Erasure Coding) uses less disk space, but the performance could be reduced in some cases. Depending on the cluster size and business needs, other RAID levels and FTT can be configured. Refer to *vSAN policies* for more information.

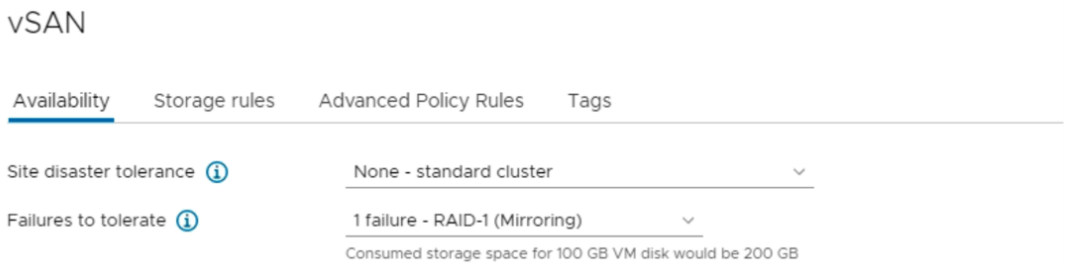


Figure 5. vSAN Storage Policy Availability Settings

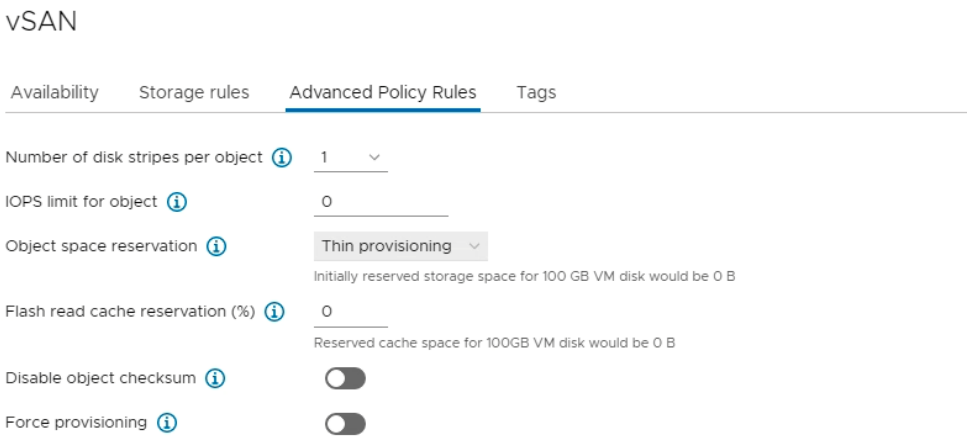


Figure 6. vSAN Storage Policy Advanced Settings

Configure NFS File Share using vSAN File Service

vSAN File Service is a layer that sits on top of vSAN to provide file shares. It currently supports SMB, NFSv3, and NFSv4.1 file shares. vSAN File Service comprises of vSAN Distributed File System (vDFS) and a Storage Services Platform (SSP). vDFS provides the underlying scalable filesystem by aggregating vSAN objects. SSP provides resilient file server end points and a control plane for deployment, management, and monitoring. File shares are integrated into the existing vSAN Storage Policy Based Management, and on a per-share basis. vSAN file service brings in the capability to host the file shares directly on the vSAN cluster.

Enabling vSAN File Services and creating NFS file shares for vHPC take minimal efforts because vSAN is enabled in the environment. We mount the vSAN NFS file share on the FrontEnd and all the Compute VMs to share installed packages, store application data, and host directories for users.

Follow the instructions to Configure vSAN File Services and Create a File Share.

Environment Settings

Recommended BOIS, vSphere and VM Settings

Table 3 presents a summary table of recommended settings to configure the BOIS and vSphere. Also update the BIOS of your server to the latest version.

Table 3: Summary of Recommended Settings for MPI Workload

LEVEL	SETTINGS	VALUE
BIOS	Power management profile	Performance Per Watt (OS)
	Hyperthreading (HT)	Enable
	Turbo	Enable
	Virtualization	Enable
	SR-IOV	Enable
	Node interleaving	Disabled
	Sub-NUMA Clustering (SNC)	Enable
	LLC Prefetch	Disabled.
	I/O Snoop HoldOff Response	2K Cycles
	Above 4G mapping/encoding	Enable (optional if high-end GPU is used)
vSphere	Power management	High performance
	High speed NIC	Choose DirectPath IO or SR-IOV based on the requirement
	MTU of VDS for RDMA workload	4096 for IB and 9000 for RoCE

VM	CPU Reservation	Calculate the total frequency (MHz) of all reserved cores
	CPU Limit	Unlimited
	Memory Reservation	All Locked
	Memory Limit	Unlimited
	Latency Sensitivity	High

At BIOS level, setting Performance Per Watt (OS) profile is important as it will let vSphere control processors' P-states and C-states. P-states (aka Power Regulator) are voltage frequency states and usually the biggest impact on unexpected CPU's performance, for example workloads may not run on a normal frequency when the default dynamic power policy (e.g., DAPC) is used. C-states can turn off certain components of CPU to save power when thread is idling. When C-states is enabled, vSphere can make certain cores idle so that other cores will run faster by using turbo boost. We can check whether the two states are presented to vSphere by checking whether "ACPI P-states" and "ACPI C-states" are shown in the power management technology panel of vSphere Client. Allowing vSphere to control these two settings is the fundamental to achieve expected high performance in the later steps. Other recommended BIOS settings are in the Table 2.

At vSphere level, we choose High performance as the active policy of power management when Performance Per Watt (OS) is enabled in BIOS. For RDMA workloads, we can use either DirectPath IO, or SR-IOV with MTU set for the VDS.

At VM level, setting Latency Sensitivity (LS) to high is the key step to ensure achieving high throughput and low latency for the MPI workloads. It will set a profile that changes CPU, memory, and network for the VM. It requires CPU and memory reservations. We give an example on how to calculate the CPU and Memory reservations in the *Best Practices* section. Refer to *Extreme Performance Series: Performance Best Practice* for more details about how LS=high is implemented. Memory reservation is required when attaching SR-IOV adapter to a VM. Setting CPU limit and memory limit as "unlimited" is also necessary, otherwise users may see unexpected behaviors.

Prepare a VM template

The operating system of RHEL 8.1 is installed on both Compute VMs and FrontEnd VM. Preparing a minimal RHEL 8.1 VM template can simplify the deployment. Once the minimal RHEL VM template is ready, we can clone it and change the settings to create Compute VM template and FrontEnd VM template.

According to the above recommended settings, Table 4 shows the settings for the Compute VM template. When configuring the VM with full CPU reservation, we want to maximize the vCPUs for the Compute VM. Even though the host has a total of 48 cores, we can only set a maximum of 44 vCPUs, as 4 cores are reserved for vSphere services. And we cover how to determine the vCPU and memory reservation settings for the VM in the *Best Practices* section.

A VMXNet3 adapter is configured per VM for the management traffic between FrontEnd VM and Compute VMs, while an SR-IOV Passthrough adapter is configured on each Compute VM for the high-speed RDMA interconnection among Compute VMs. The settings of Compute VM can be adjusted accordingly based on the needs of workloads.

Table 4: Compute VM Settings Design Choice

Property	Specification
CPU	44 vCPU (131560 MHz Reserved)
Cores per Socket	22
CPU Limit	Unlimited
Memory	320GB (All locked)
Memory Limit	Unlimited
Latency Sensitivity	High

Hard Disks	1 x 256GB on vSAN Datastore
SCSI Controller	VMware Paravirtual
Network Adapter 1	VMXNet3
Network Adapter 2	SR-IOV passthrough

FrontEnd VM is a lightweight role. As we tested, the FrontEnd VM with 2 cores, 8GB memory, and 256G hard disk can be fully functional. Configure the FrontEnd VM properly based on the number of concurrent login users. Table 5shows the settings of the FrontEnd VM template in our environment.

Table 5: FrontEnd VM Settings

Property	Specification
CPU	2
Memory	8GB
Hard Disks	1 x 256GB
SCSI Controller	VMware Paravirtual
Network Adapter	VMXNet3

Additional Configurations for the Compute VM Template:

- Set proper network MTU for the Compute VM template: 4096 for InfiniBand and 9000 for RoCE.
- Install the latest Mellanox OFED.
  Since the Mellanox RDMA adapter is used in this design, the latest version of Mellanox OFED needs to be installed.
- Install packages on the vSAN file share and mount the file share to the VM template.
  Packages can be installed on the NFS file share, and the file share can be mounted by editing the /etc/fstab file. The example below shows the editing in /etc/fstab to mount the Intel oneAPI Base Toolkit installation path to the template.

  ```
  ${vSAN_NFS_Primary_IP}:/vsanfs/oneapi    /opt/intel/oneapi     nfs     rw 0 0
  ```

- Enable passwordless ssh for a non-root admin user.
  The command below copies the local ssh key of a non-root admin user (for example, vmware) to the VM template itself, thus passwordless ssh can be enabled for this user in the virtual cluster.

  ```
  ssh-copy-id vmware@localhost
  ```

Refer to *Running HPC and Machine Learning Workloads on VMware vSphere* for more details about BIOS, ESXi, VM and guest OS settings and best practices.

vHPC Deployment

In this section, we present two deployment approaches of the vHPC cluster in our design:

- Option 1 - Manually install the HPC cluster by following the deployment steps in the OpenHPC deployment guide
- Option 2 - Automatically install the HPC cluster by using the vHPC-Toolkit

Option 1 - Manual Deployment Procedure

Figure 7 presents the manual deployment procedures of vHPC. It can be divided into three stages:

1. Provision VDS and VMs.
2. Use xCAT to configure FrontEnd VM and Compute VMs, then install SLURM on them.
3. Install packages on NFS based on the needs of workloads.

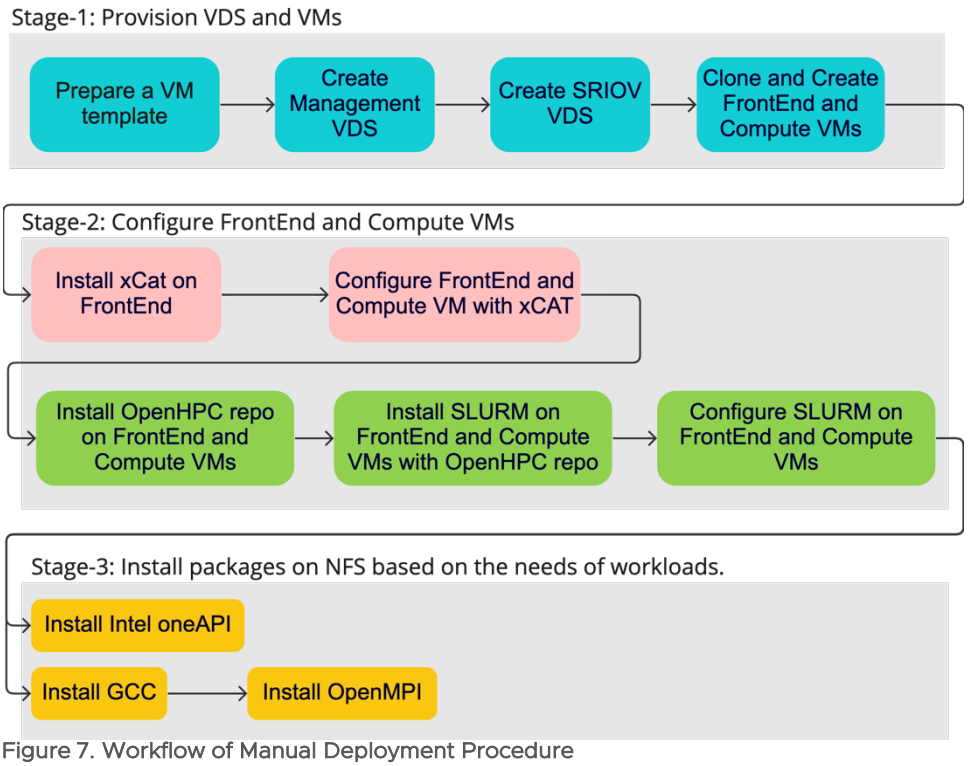


Figure 7. Workflow of Manual Deployment Procedure

Stage 1: Create vSphere Distributed Switch (VDS) and Clone VMs

In this step, we first create two VDS; for example, Management VDS and SR-IOV VDS as designed in the Configuration section. To do this manually with vSphere Client, refer to Create a vSphere Distributed Switch.

Secondly, we clone the VM template to one FrontEnd VM and four Compute VMs. The FrontEnd VM will use fewer resources as mentioned in our design, thus we set it to 2 vCPUs, 8 GB memory and assign the VM Management PortGroup to the VMXNet3 network adapter.

Stage 2: Configure FrontEnd and Compute VMs

We use variable substitution (${variable}) in the following command-line examples to provide an easy replacement to fit your environment. A summary of the required and optional variables used throughout this procedure is presented below. Refer to the OpenHPC Install Guide for more command details.

```
# Variables used in xCAT
${frontend_name}                          # Hostname for FrontEnd VM
${compute_group_name}                     # Group name of all Compute VMs
${c_name[0]}, ${c_name[1]}, ...           # Hostnames for Compute VMs
${num_computes}                           # number of Compute VMs
${c_ip[0]}, ${c_ip[1]}, ...               # Compute VMs's IP address
# Variable used in SLURM
${num_sockets}                            # number of sockets in a Compute VM
${cores_per_socket}                       # cores per socket in a Compute VM
${threads_per_core}                       # threads per core in a Compute VM
```

Install xCat on FrontEnd VM and Configure vHPC cluster

xCat is the tool to manage HPC clusters; for example, sync files using *xdcp* command and execute commands across all the machines in the cluster by *xdsh* command. In this way, all the Compute VMs can be configured in parallel. For readers who will use xCAT for the first time, we provide the following commands for you to start up, for the full documentation and additional settings, refer to the xCAT stateful documentation.

We first use the commands below to install xCAT and its dependent packages on the FrontEnd VM.

```
## Enable xCat's Public repo
wget -P /etc/yum.repos.d https://xcat.org/files/xcat/repos/yum/latest/xcat-core/xcat-core.repo
## Enable the repo of xCat's dependent package for local use
wget -P /etc/yum.repos.d https://xcat.org/files/xcat/repos/yum/xcat-dep/rh8/x86_64/xcat-dep.repo
yum -y install xCAT
## enable xCAT tools for use in current shell
. /etc/profile.d/xcat.sh
```

Next, we add all the Compute VMs into the xCAT database.

```
for ((i=0; i<${num_computes}; i++)) ; do
    mkdef -t node ${c_name[$i]} groups=$compute_group_name,all ip=${c_ip[$i]} arch=x86_64
done
```

Then we can configure the Compute VMs remotely in parallel such as sync DNS or hosts files, disable the firewall on FrontEnd and Compute VMs.

```
# Sync hosts file
xdcp $compute_group_name /etc/hosts /etc/
# Sync DNS files to all Compute VMs
xdcp $compute_group_name /etc/resolv.conf /etc/resolv.conf
# Disable Firewall on all compute VMs
xdsh $compute_group_name systemctl disable firewalld.service
```

Install OpenHPC Repo

OpenHPC repo contains common tools and libraries required to deploy and manage HPC Linux clusters including provisioning tools, resource management, I/O clients, development tools, and a variety of scientific libraries. We use it to download verified software packages and speed up our provision.

- Install OpenHPC Repo on FrontEnd VM.
  `dnf -y install http://repos.openhpc.community/OpenHPC/2/CentOS_8/x86_64/ohpc-release-2-1.el8.x86_64.rpm`

- Install OpenHPC Repo on all the Compute VMs using xCAT from FrontEnd VM.
  `xdsh $compute_group_name dnf -y install http://repos.openhpc.community/OpenHPC/2/CentOS_8/x86_64/ohpc-release-2-1.el8.x86_64.rpm`

Install Slurm and Start Slurm Service

Slurm is an open-source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters. It provides a way of scheduling HPC jobs and allocating cluster resources to those jobs. Slurm needs to be installed on FrontEnd VM and Compute VMs.

Below are the commands to install Slurm from openHPC repo and configure Slurm:

- Install Slurm-server on FrontEnd VM.

  ```
  dnf -y install ohpc-slurm-server
  ```

- Edit the /etc/slurm/slurm.conf file on FrontEnd VM.

```
# Manually edit slurm.conf
# 1. Find the lines below and change them as follows:
ClusterName=${ClusterName}
ControlMachine=${frontend_name}
NodeName=${c_name} Sockets=${num_sockets} CoresPerSocket=${cores_per_socket}
ThreadsPerCore=${cores_per_socket} State=UNKNOWN
PartitionName=${PartitionName} Nodes=${c_name} Default=YES MaxTime=24:00:00 State=UP
Oversubscribe=EXCLUSIVE
ReturnToService=2
SelectType=select/linear
# 2. Comment the following lines
SelectTypeParameters=CR_Core     # Comment this line!
JobCompType=jobcomp/none         # Comment this line! This is OpenHPC typo
```

- Install Slurm-client on Compute VMs.

```
xdsh $compute_group_name dnf -y install ohpc-slurm-client
```

- Sync slurm.conf and munge.key between FrontEnd VM and Compute VMs

```
xdcp $compute_group_name /etc/slurm/slurm.conf /etc/slurm/slurm.conf
xdcp $compute_group_name /etc/munge/munge.key /etc/munge/munge.key
```

- Start Slurm and Munge services.

```
# Start munge and slurm controller on FrontEnd VM
systemctl enable munge
systemctl enable slurmctld
systemctl start munge
systemctl start slurmctld
# Start slurm clients on Compute VMs
xdsh $compute_group_name systemctl enable munge
xdsh $compute_group_name systemctl enable slurmd
xdsh $compute_group_name systemctl start munge
xdsh $compute_group_name systemctl start slurmd
# Update compute node's state as Idle on FrontEnd VM
scontrol update NodeName=[${c_name}] State=Idle
```

Stage 3: Install Packages on NFS based on the Workload Needs

Additional packages such as Intel oneAPI Base Toolkit, GCC compiler, and OpenMPI can be installed depending on the requirements of the workloads. The packages should be installed on the NFS file share so that all the Compute VMs can access. Refer to the following links to install the selected versions:

- Install Intel OneAPI Base and HPC Toolkit
- Install GCC
- Install OpenMPI

Option 2 - vHPC-toolkit for Declarative Install of HPC Cluster on vSphere

vHPC toolkit is a flexible, extensible, and easy-to-use open-source toolkit. It allows users to deploy and manage virtualized clusters for HPC and Machine Learning (ML) environments. Instead of typing commands in the terminal or manually configuring the environment through vSphere UI, vHPC toolkit can speed up the cluster provision by automating the whole process using vSphere python API. The toolkit also provides great extensibility. Adding additional tasks in a cluster is as easy as adding new entries in the definition file.

vHPC toolkit performs the following tasks in the predefined sequential order:

1. Create SVS/DVS and network port groups
2. Clone VMs
3. Customize CPU/memory of VMs
4. Perform other VM customization, such as CPU/memory shares, reservation, latency sensitivity
5. Add network adapters to VMs
6. Configure networking of a VM, such as IP address, DNS, gateway, and others

7. Add devices to VMs, such as DirectPath IO devices, SR-IOV adapters, and others
8. Power on VMs
9. Connect to *Open VM Tools* and return IP addresses of VM to users
10. Run post provision script

Same type of tasks across multiple VMs or multiple SVS/DVS are executed in parallel.

Figure 8 presents the workflow to create a vHPC cluster using the vHPC-Toolkit. We can divide the installation steps broadly into three stages.

1. Create VM templates for Compute and FrontEnd VMs before running the vHPC-toolkit.
2. Set up the vHPC cluster by using vHPC-toolkit:
   a) Create a key-value definition file to describe VMs, networks, and the post provision scripts for the vHPC cluster.
   b) Execute the vHPC command to provision the cluster.
3. Install packages on NFS based on the workload needs after running the vHPC-toolkit.

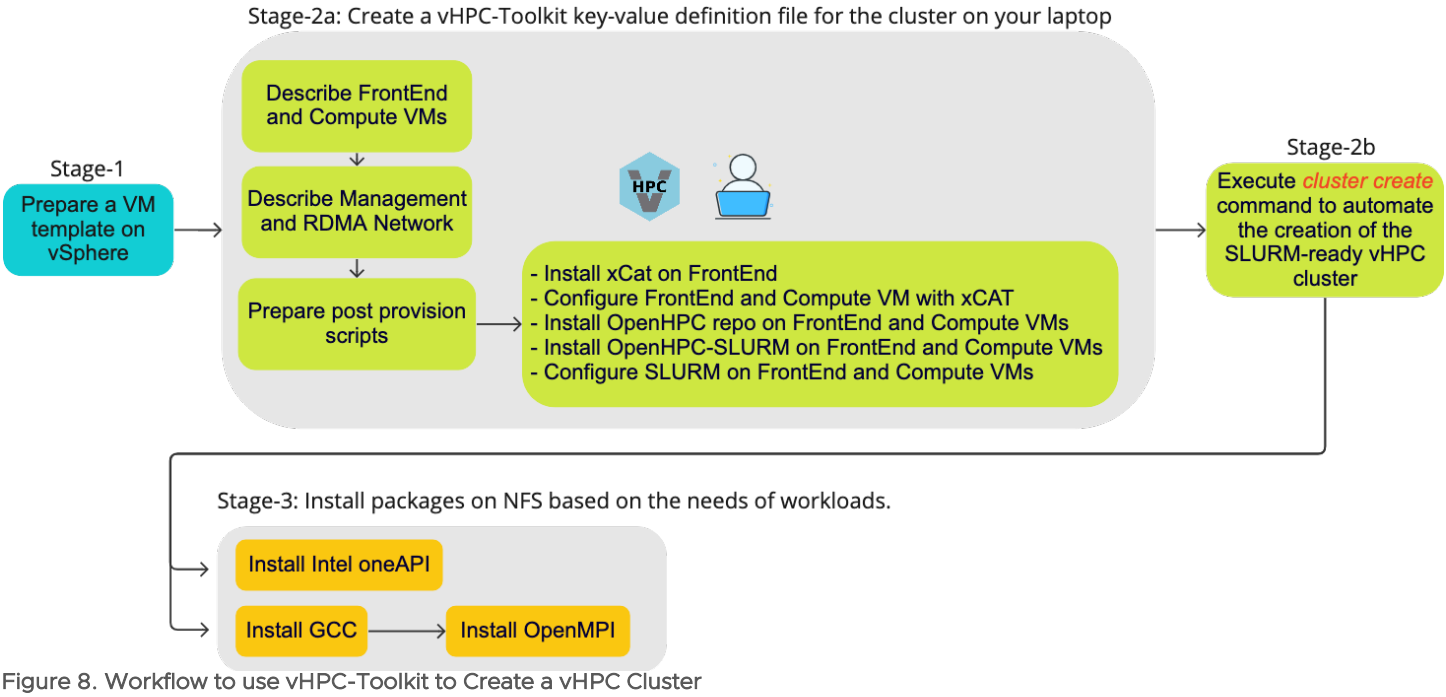


Figure 8. Workflow to use vHPC-Toolkit to Create a vHPC Cluster

An Example of a vHPC-toolkit cluster definition file used in Stage-2

rhel-hpc-demo-cluster.conf gives an example of a cluster definition file. The file structure is as follows:

```
# Key value Macros
[FRONTEND]
Key: Value
...
[FRONTEND-POST]
script: /path/to/rhel-hpc-demo-frontend.sh
...
[COMPUTE]
Key: Value
...
[Management-NETWORK]
Key: Value
...
[RDMA-NETWORK]
Key: Value
...
[Other user-defined macros]
...

##### Execution-step1. Create VDS
[_DVS_]
{RDMA_VDS_name}: Key: Value ...
{Management_VDS_name}: Key: Value ...
...
##### Execution-step2. Create VMs
[_VMS_]
{FrontEnd_VM_name}: FRONTEND Management-NETWORK FRONTEND-POST
{Compute_VM_name}: COMPUTE Management-NETWORK RDMA-NETWORK
```

- vHPC-Toolkit will firstly run the [_DVS_] section then the [_VMS_] section, and can append the user-defined macros declared before the two sections.
- The [_DVS_] section shows that the two VDS for management and RDMA will be created. [_VMS_] section describes 1 FrontEnd VM and 4 Compute VMs to be created.
- The [FRONTEND] macro specifies the VM settings of FrontEnd VM, and the [COMPUTE] macro specifies the Compute VMs. Besides, both the FrontEnd VM and Compute VM connect to the "management" network using Management-Network macro. Additionally, the 4 Compute VMs will also connect to the RDMA interconnect (SR-IOV) as described in the [RDMA-NETWORK] macro.
- The [POST-FRONTEND] macro links to a post-provision script "rhel-hpc-demo-frontend.sh".In this script, we use the same command mentioned above to install xCAT, configure FrontEnd and Compute VMs, and set up Slurm on the cluster.

Based on the cluster definition described in the cluster definition file, we can simply run the *cluster* command below to automate the creation of the Slurm-ready vHPC cluster.

```
./vhpc_toolkit cluster --create --file ../examples/cluster-scripts/rhel-hpc-demo-cluster.conf
```

Full documentation of vHPC-Toolkit is available in the GitHub page. Administrators are recommended to watch our VMUG tutorial video to learn the basic and advanced usages of vHPC-Toolkit to speed up their vHPC deployment and management.

vHPC Cluster Validation and Performance Testing

In this section[1], we demonstrate how to verify the deployment, functionalities, and performance of vHPC on vSphere 7 using the Intel Select Solution for HPC tools. The validation covers the following scenarios:

- Check Cluster Health using Intel Cluster Checker
- vHPC vs Bare-mental HPC Performance Analysis
- Workload Benchmark Performance

[1] **Disclaimer:** Performance varies by use, configuration and other factors. Learn more at *https://www.intel.com/PerformanceIndex*. Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for configuration details. No product or component can be absolutely secure. Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy. Your costs and results may vary. Intel technologies may require enabled hardware, software, or service activation.

Cluster Health Check using Intel Cluster Checker

Intel Cluster Checker (CLCK) verifies the configuration and performance of Linux*-based clusters and checks the cluster's compliance with the Intel Select Solution for Simulation and Modeling. After loading the Intel oneAPI by using the command source /opt/intel/oneapi/setvars.sh, we can launch the following four tests. The first three tests can be launched on more than 4 nodes to ensure the uniformity of all nodes in the cluster. Since the last simulation and modeling check takes longer time than the first three tests, in order not to interfere the production of the whole cluster, CLCK is designed to run on exactly 4 nodes.

- **Check the Basic Health of the Cluster**

  Firstly, we launch the basic health check to examine the functionalities and functions of each node. It also checks the uniformity of CPU, ethernet, InfiniBand and so on across all the nodes in the cluster.

  ```
  clck -f nodelist -F health_base --db health_base.db -o health_base.log

  ----------------------------------------------------------------
  4 nodes tested:         compute-[1-4]
  4 nodes with no issues: compute-[1-4]
  0 nodes with issues:
  ----------------------------------------------------------------
  FUNCTIONALITY
  No issues detected.
  HARDWARE UNIFORMITY
  No issues detected.
  PERFORMANCE
  No issues detected.
  SOFTWARE UNIFORMITY
  No issues detected.
  ```

  The above shows the desired output for a healthy cluster without any issues detected. If issues are detected, troubleshoot with the warning logs.

- **Check the Extended Health of the Cluster**

  The command below adds a variety of performance related checks in addition to the basic health check; for example, internode connectivity, uniformity of file system and memory, functionalities of MPI, peal and python, sgemm and stream performance test. It is optional but highly recommended. Similarly, "No issues detected" is the desired output for a healthy cluster.

  ```
  clck -f nodelist -F health_extended_user --db health_extended_user.db -o health_extended_user.log
  ```

- **Check the Compliance with Intel Specifications**

  Next, we check the compliance with Intel HPC platform specifications. This step will examine whether the cluster meets the requirement of intel_hpc_platform_compat-hpc-cluster-2.0, which includes checking the core-intel-runtime related libraries. "No issues detected" is the desired result.

  ```
  clck -f nodelist -F intel_hpc_platform_compat-hpc-cluster-2.0 --db platformspec.db -o
  platformspec.log
  ```

- **Check the Simulation and Modeling Compliance**

  For this step, we launch the simulation and modeling check. The commands below execute various checks including the benchmarks (DGEMM, HPCG_SINGLE, HPCG_CLUSTER, HPL, IMB_PINGPONG, STREAM).

  For 2nd Generation Intel® Xeon® Scalable Processors, we use the following command.

  ```
  clck -f nodelist -F select_solutions_sim_mod_user_plus_2018.0 --db
  select_solutions_sim_mod_user_plus_2018.0.db -o select_solutions_sim_mod_user_plus_2018.0.log
  ```

  For 3rd Generation Intel® Xeon® Scalable Processors, use the following command.

```
clck -f nodelist -F select_solutions_sim_mod_user_plus_2021.0 --db
select_solutions_sim_mod_user_plus_2021.0.db -o select_solutions_sim_mod_user_plus_2021.0.log
```

"No issues detected" is desired again, and the performance of five micro-benchmark results will show up.

vHPC Performance Validation using Intel Cluster Checker (CLCK) Benchmarks

In this section, we use Intel Cluster Checker Benchmarks to compare the performance results between the virtualized HPC and the bare-metal HPC and demonstrate that there is no significant overhead introduced by virtualizing the HPC cluster. The same set of compute servers are used for both virtualized HPC and bare-mental HPC configurations. However, as part of the best practices for vHPC, we are only able to configure the compute VM with 44 cores when applying CPU reservation in the VM settings. Although we are reducing the CPU cores available to the HPC application with full CPU reservation in the VM, tests have shown that it provides better performance compared to the alternative—full size VM without CPU reservation.

We present results for the following 3 configurations:

- Bare-metal HPC with an external NFS appliance (Dell EMC F200) for application data store
- vHPC: Virtualized HPC with an external NFS appliance for application data
- vHPC with vSAN: Virtualized HPC with NFS File Share on vSAN

Intel CLCK uses DGEMM to benchmark the computing capabilities of a processor. By default, CLCK run DGEMM with 55% of available memory as the matrix input size and 45 iterations. Figure 9 shows that the DGEMM performance in either Virtual or Virtual vSAN has the same performance as Bare Metal. STREAM is a simple synthetic benchmark program that measures sustainable memory bandwidth (in MB/s) and the corresponding computation rate for simple vector calculations. STREAM results in Figure 9 show that the memory performance on a single node in vHPC is nearly the same as Bare Metal.

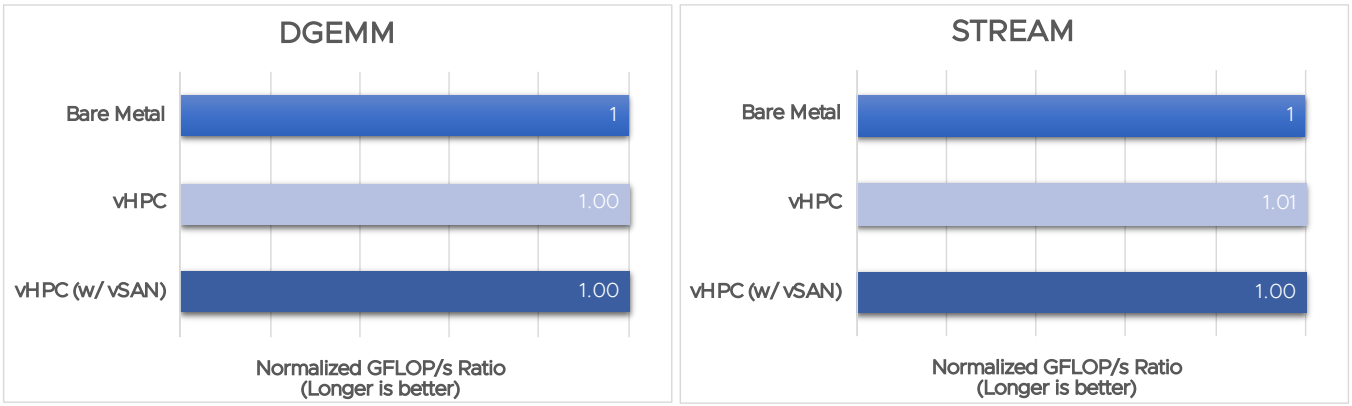


Figure 9. DGEMM & STREAM Performance for Bare Metal, Virtual, and Virtual with vSAN

The next set of tests IMB_PINGPONG measures the network performance. This benchmark measures the passing of a single message from the initiating host to his peer and then waiting for an answer. It collects the pairwise bandwidth and latency measurements. Higher bandwidth and lower latency are considered as better performance. IMB_PINGPONG results in Figure 10 demonstrate a 7~10% delta for the InfiniBand network performance between vHPC and Bare Metal, and this is attributed to having fewer cores per Compute Node in the vHPC environment.

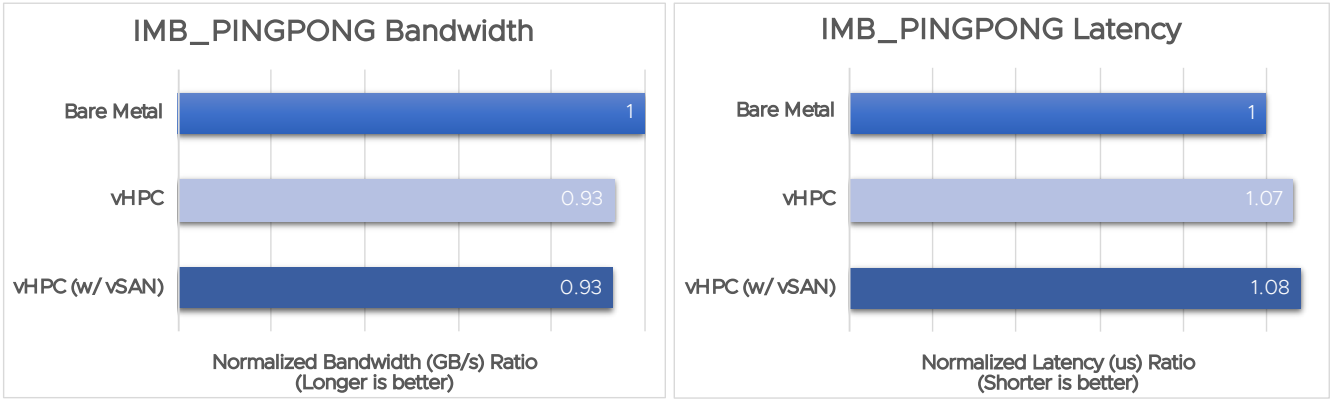


Figure 10. IMB_PINGPONG Bandwidth & Latency Performance for Bare Metal, Virtual, and Virtual with vSAN

The HPCG benchmark is intended to model the data access patterns of real-world applications such as "sparse matrix" calculations, thus testing the effect of limitations of the memory subsystem and internal interconnect of the supercomputer on its computing performance. HPCG test is performed in two scenarios – as an individual job on each of the four nodes (hpcg_single) and as a single job performed across all four nodes (hpcg_cluster).

HPCG results in Figure 11 show that both single node and cluster performance in vHPC configuration is nearly the same as Bare Metal.

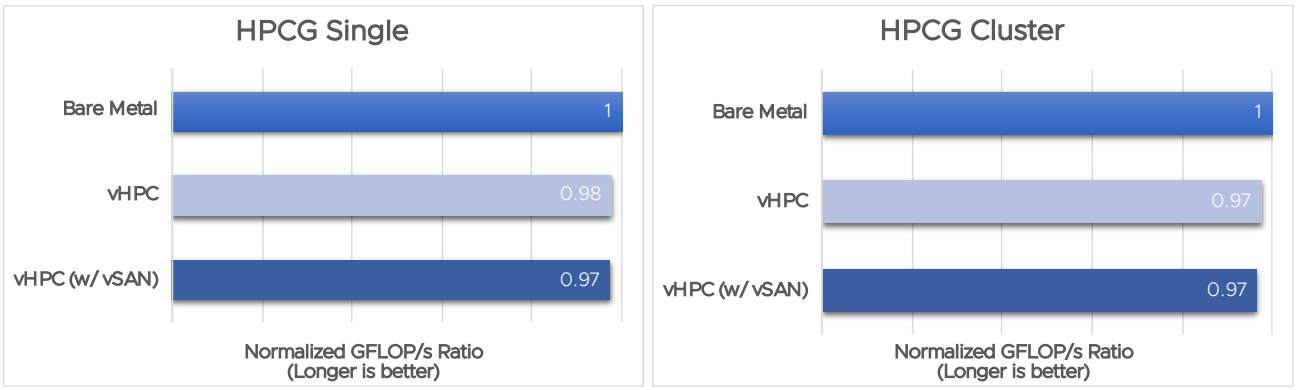


Figure 11. HPCG Single Node & Cluster Performance for Bare Metal, Virtual, and Virtual with vSAN

The last test in the Intel CLCK benchmark suite is High Performance Linpack (HPL). HPL is a software package that solves a (random) dense linear system in double precision arithmetic (64 bits) on distributed-memory computers. Figure 12 shows that the HPL performance in vHPC is 6% lower compared to Bare Metal, and this is again due to the difference in CPU cores between the vHPC and bare metal compute instances.

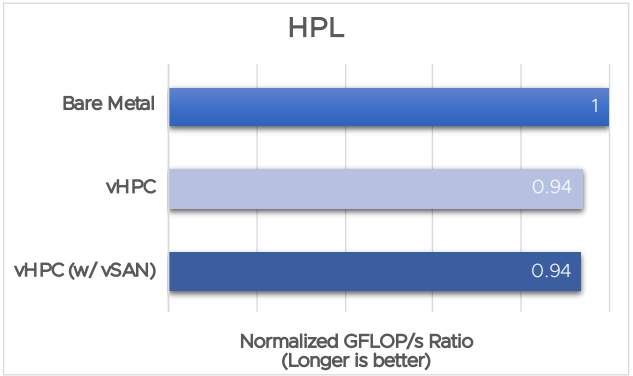


Figure 12. HPL Performance for Bare Metal, Virtual, and Virtual with vSAN

vHPC Workload Performance Results

After validating the cluster capability with Intel CLCK, we run real world HPC applications to further evaluate the performance efficiency of vHPC cluster. These tests will stress CPU, networking and in some cases, the storage resources of the HPC cluster.

HPC performance on virtualized infrastructure is primarily determined by two factors: hardware virtualization support and virtual infrastructure performance capability. With advances in both VMware vSphere as well as hardware virtualization support on, applications running on a single VM can generally run at close to full speed in the VMware virtualized environment. MPI applications by nature are more challenging, requiring sustained and intensive communication between nodes, making them sensitive to interconnect performance. Continued performance optimization efforts at VMware for improving network stack performance, we see near bare metal performance running these challenging HPC workloads as demonstrated by the low overhead observed in our tests.

Table 6 shows the HPC applications and benchmark input dataset that were used for real world HPC application testing. Each set of experiments were run 5 times and we measure the average throughput across the 5 runs. To demonstrate the strong scaling efficiencies, we ran the tests on one, two and four compute nodes. Relative performance is shown for all applications and the performance on a single Bare Metal node is used as the baseline.

Table 6: Workload and Benchmark Details

Application	Vertical Domain	Benchmark Dataset	Version
GROMACS	Life Sciences – Molecular Dynamics	HECBioSim 3M Atoms	2020.5
LAMMPS	Molecular Dynamics – Fluid and materials	*EAM Metallic Solid Benchmark*	20210310
Weather Research and Forecasting (WRF)	Weather and Environment	*Conus 12KM*	3.9.1.1
OpenFOAM	Manufacturing – Computational Fluid Dynamics (CFD)	*Motorbike 6M cell mesh*	9

GROMACS

GROMACS is a molecular dynamics application often used for simulating the Newtonian equations of motions for system with millions of biochemical molecules like proteins, lipids, and nucleic acids that have a lot of complicated bonded interactions. We test GROMACS on the 3 million atom system - a pair of hEGFR tetramers of *1IVO* and *1NQL*. Figure 12a shows relative performance for Bare Metal and vHPC leveraging different NFS options,

Figure 13a shows that the GROMACS application performance delta between either Virtual and the Bare Metal is 11.0% on 4 nodes, which is expected due to the difference in number of MPI ranks used between virtual (176 Ranks) and bare metal (192 Ranks) in the simulation.

LAMMPS

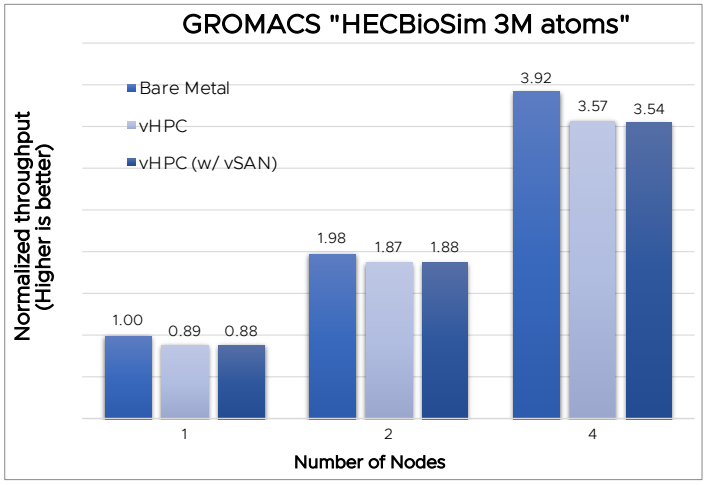
LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) is a classical molecular dynamics simulator which focuses on materials modeling for solid-state materials (metals, semiconductors), soft matter (biomolecules, polymers) and coarse-grained or mesoscopic systems. We tested LAMMPS using the copper metallic solid with Embedded Atom Method (EAM) on 1M atoms dataset. Figure 13b shows again that the LAMMPS application performance delta between Virtual and the Bare Metal is 7.1% on 4 nodes.

WRF

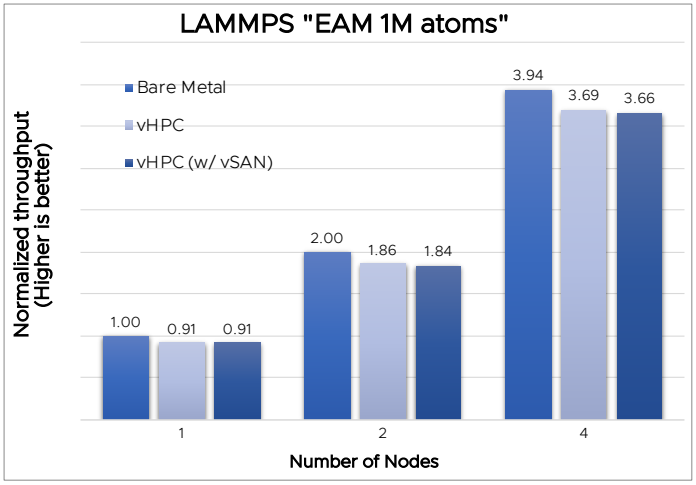
Weather Research and Forecasting (WRF) is a numerical weather prediction system designed for both atmospheric research and operational forecasting applications. We use the 12KM Conus dataset which is a 48-hour, 12km resolution case over the Continental U.S. (CONUS) domain October 24, 2001 that uses the Eulerian Mass (EM) dynamics. Figure 13c shows WRF performance for Bare Metal and Virtual configurations. We measure the throughput by the mean time per step and observe a performance delta of 5.4% on 4 nodes.

OpenFOAM

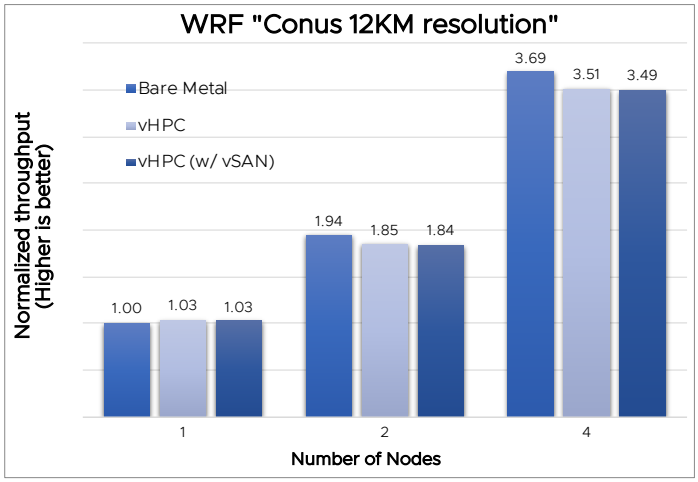
OpenFOAM is an open-source computational fluid dynamics (CFD) software. CFD applications represents the largest user community of HPC infrastructures in engineering and manufacturing industries. We use the commonly used Motorbike model and run the simulation using 6 million atoms dataset to calculate the steady flow around a motorcycle and rider. Figure 12d charts the performance for bare metal and virtual clusters and we observed a difference of 6.4% when using 4 nodes. Figure 13d only shows a single virtual configuration - vHPC with external NFS, and does not include data for vHPC with NFS service on vSAN. Running OpenFOAM generates a very large number of parallel file transactions that was observed to overload the vSAN file service. Therefore, we recommend using an external NFS service with adequate resources for I/O intensive applications like OpenFOAM.



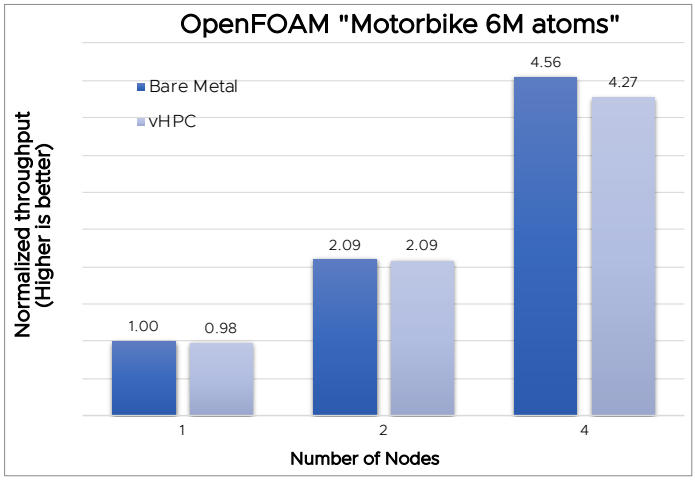
(a)



(b)



(c)



(d)

Figure 13. WRF, LAMMPS, GROMACS & OpenFOAM Performance on Bare Metal, Virtual, and Virtual with vSAN

Best Practice

vSAN Cluster Design

Multiple disk groups are preferred to get performance advantages and reduce failure domains within a host. Refer to *VMware vSAN Design Guide* for more information about vSAN cluster design considerations.

Recommendation: Use multiple disk groups for each host.

vSAN Data Deduplication and Compression

DD&C services cost extra CPU cycles and were disabled in our case to potentially get better performance. Users can choose to enable DD&C to improve the space efficiency depending on the workloads.

Recommendation: Refer to *vSAN Space Efficiency Technologies* for more details.

vSAN Storage Policy

Storage Policy Based Management (SPBM) allows you to align storage with the application demands of the virtual machines. The Number of Failures to Tolerate capability addresses the key customer and design requirement of availability. With FTT, availability is provided by maintaining replica copies of data, to mitigate the risk of a host or component failure resulting in lost connectivity to data or potential data loss. RAID1 FTT=1 (Mirroring) uses more disk space but provides better performance in general. RAID 5 FTT=1 (Erasure Coding) uses less disk space, but the performance is reduced in some cases. Depending on the cluster size and business needs, other RAID levels and FTT can be configured.

Recommendation: Refer to *vSAN policies* for more information.

vSAN Encryption

vSAN can perform data at rest encryption. Data is encrypted after all other processing, such as deduplication, is performed. Data at rest encryption protects data on storage devices in case a device is removed from the cluster. Use encryption as per your company's Information Security requirements.

Recommendation: Enable Encryption as required per your InfoSec.

Configurations for BIOS, ESXI, VM and Guest OS

The white paper "Running HPC and Machine Learning Workloads on VMware vSphere " provides best practices to configure BIOS, ESXI, VM and Guest OS for running HPC workloads on VMware vSphere.

The max number of vCPUs that can be reserved by a VM is calculated as floor(#physical cores available in host - #cores reserved by ESXi).

To determine the number of cores reserved by ESXi, users can run the command *sched-stats -t groups | less -s* on ESXi when no VMs are powered on.

- In the top row, the returned *resvMHz* column indicates the total CPU resources reserved for ESXi services. The same value, divided by the system's MHz-per-core, indicates the number of reserved cores.
- For example, we execute the above *sched-stats* command on ESXi and notice that the *resvMHz* column shows 10386 MHz on the row of host. Then we check that the CPU frequency of each core shows 2.99 GHz on vSphere Client, thus 10386 MHz / (2.99 * 1000 MHz / core) = 3.5 cores is used for vSphere services. Since our host has 48 physical cores, we use the above formula is floor(48-3.5)=44 cores, which is the maximum number of cores to be fully reserved by a user VM.
- We configure the VM to reserve 2.99 * 1000 MHz / cores * 44 cores = 131560 MHz. This step can be simplified by using vhpc-toolkit

To determine the max memory available to set to a user VM, you can run the following command:

```
[root@esxi:~] echo $(( $(vsish -e get /sched/groups/4/stats/capacity | sed -n 's/^   mem-
unreserved:\([0-9]\+\) KB$/\1/p') / 1024 / 1024)) GB
```

The number 4 in the command is the user pool for any supported and shipping build. You can also make sure the user pool's id is 4 by this command `vsish -e set /sched/groupPathNameToID /host/user`

Recommendation: Follow the instructions in the white paper to configure the BIOS Settings, ESXI Settings, and VM and Guest OS settings.

RDMA Interconnect Configurations

Configuring the RDMA interconnect NIC via DirectPath IO is more straightforward, and it also achieves near bare-metal performance. This option is preferred when the entire NIC is actively used by a single VM exclusively. Configuring the RDMA interconnect via SR-IOV involves creating a new SR-IOV VDS. This layer of indirection may introduce very subtle overhead. And it enables the underlying hardware to be shared among multiple VMs.

Recommendation: Configure RDMA interconnect by using DirectPath IO when the entire hardware is used by a single VM exclusively. Otherwise, the SR-IOV RDMA interconnect should be used. The MTU of VDS needs to enable Jumbo Frame, thus set to 9000.

Hardware Resource Considerations

We present a four-node virtual cluster that uses RDMA, vSAN and external NFS to meet the requirements of Intel Select Solutions. The cluster can be extended to a larger scale cluster based on the workloads needs of HPC users. Hardware (e.g., CPU, memory, network,

storage) can also be customized depending on the application needs. For general purpose HPC clusters that run various distributed workloads or the HPC clusters run high MPI communication workloads, high speed interconnect (IB or RoCE) is recommended. This paper presented a starting point for virtualized HPC clusters and the hardware specifications were chosen to accommodate larger production scale workloads running on cluster with higher number of compute nodes, For smaller clusters as demonstrated in the paper of 4 compute nodes, 25 Gbps or even ethernet may provide enough resources to meet performance requirements.

Summary

High Performance Computing (HPC) workloads are forecasted to be one of the fastest-growing workload types. With VMware, you can capture the benefits of virtualization for HPC workloads while delivering performance that is comparable to Bare Metal. Our approach to virtualizing HPC adds a level of flexibility, operational efficiency, agility, and security that cannot be achieved in bare-metal environments—enabling faster time to insights and discovery. Together with Intel Select Solution for HPC, enterprises can have confidence working on a validated HPC system on vSphere 7.

In this solution, we showcase the architecture design of vHPC with Intel Select Solution on vSphere 7 including the hardware and software components, network design, the deployment and validation process, and best practices. Extensive benchmarks and workloads are tested to evaluate the impact of running HPC applications in a virtualized environment, including leveraging vSAN for NFS file services. The performance results demonstrate that the vHPC solution is able to provide near bare-metal performance, while providing a greater level of flexibility, efficiency, agility, and security.

References

- InfiniBand SR-IOV Setup and Performance Study on vSphere 7.x
- RoCE SR-IOV Setup and Performance Study on vSphere 7.x
- InfiniBand and RoCE DirectPath IO Setup and Performance Study on vSphere 7.x
- Intel Cluster Checker
- OpenHPC Install Guide
- xCAT stateful documentation
- Running HPC and Machine Learning Workloads on VMware vSphere
- vHPC-Toolkit Full Documentation
- vHPC-Toolkit Tutorial Video in VMware User Group (VMUG) Meeting
- VMware vSphere
- VMware vSAN
- VMware vCenter Server
- Configure vSAN File Services
- Create a File Share
- Create a vSphere Distributed Switch
- Install Intel oneAPI Base & HPC Toolkit
- Install GCC
- Install OpenMPI
- VMworld Barcelona 2018: VIN2677BE – Extreme Performance Series: Performance Best Practices

About the Authors

Yuankun Fu, Member of Technical Staff and Aaron Chu Sr. Solution Engineer, wrote the original content of this solution RA.

Yuankun Fu has 11 years of HPC experience and has a Ph.D. degree in Computer Science from Purdue University. He currently works as a Member of Technical Staff in HPC/ML group of VMware OCTO and focuses on HPC/ML application performance on the VMware platform. He works on a wide variety of HPC projects, from creating technical guides and performance best practices to root-causing performance challenges when running highly technical workloads on customer platforms.

Aaron Chu, Sr. Solution Engineer in the Workload Application Solutions team of the Cloud Infrastructure Business Group.

The following members also contributed to the paper:

- **Ramesh Radhakrishnan**, Technical Director, VMware

Ramesh Radhakrishnan leads the High-Performance Computing team, working to bring the value of VMware technologies to HPC. Previously, he was a Senior Distinguished Engineer at Dell Technologies, where he began his career in the Enterprise Server and Storage business group. He has a Ph.D. in Computer Science and Engineering from the University of Texas at Austin.

- **Chen Wei**, Director in the Workload Application Solutions team, VMware
- **Catherine Xu**, Group Manager in the Workload Application Solutions team, VMware
- **Uppuluri Lokendra,** Cloud Solutions Architect, Intel

Lokendra Uppuluri is a solutions architect in Data center and AI group at Intel. He is primarily focused on AI workloads with a particular passion for the emergent area of HPC/AI convergence. He has 13+ years of experience architecting and building cloud native, performant, scalable enterprise applications in AI. His domain expertise extends to Healthcare, Financial, Robotics verticals. He is an invited speaker for various industry events where he shares his passion for architecting applications for emergent technologies.

- **Agustin Malanco Leyva**, Emerging Workloads Solutions Architect, VMware

Agustin Malanco is currently working as an Emerging Workloads Solutions Architect for the Office of the CTO. He has worked for VMware since 2011 across many different roles, ranging from pre-sales, multiple solution architecture positions, and as a Technical Product Manager. Agustin has 13+ years of professional experience and a proven track record across multiple VMware products and data center technologies. He is currently one of six individuals in the world to hold three VCDX (#141) certifications and serves as a VCDX Panelist

- **Patryk Wolsza,** Cloud Solutions Architect, Intel

Patryk Wolsza has more than a decade of experience in designing, building, and tuning cloud architectures to store different types of workloads from on-premise through hyperscalers to edge. Currently he is working in the R&D department of Data Center and AI Group to shape the way how clouds are used by Enterprise. He delivered multiple publications with partners about workloads in virtualized datacenters explaining how to properly set them up, tune, and benchmark them. Since 2019 he holds the title of VMware vExpert.

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