# Dell PowerScale and Cloudera CDP Private Cloud Base Reference Architecture

## Cloudera Data Platform Private Cloud Base 7.1.6 and PowerScale OneFS 9.2

January 2022

H18864.1

White Paper

Abstract

This document is a high-level design, performance, and best-practices guide. It details deploying Cloudera Data Platform Private Cloud Base 7.1.6 on bare-metal infrastructure with the Dell PowerScale scale-out NAS solution as a shared storage back end.

Dell Technologies

## Copyright



**Cloudera disclaimer**

# Contents

# Executive summary

## Overview

Dell Technologies works closely with Cloudera to identify the needs of Apache Hadoop customers and to validate hardware- and software-based solutions. These efforts empower organizations with deeper insights, and help them employ enhanced, data-driven decision making by using the right infrastructure for the right data.

Dell PowerScale and Cloudera combine to create a powerful yet simple, highly efficient, and massively scalable storage platform with integrated support for Hadoop analytics. PowerScale is the first and only scale-out NAS platform to incorporate native support for the HDFS layer. Unstructured data on PowerScale and native HDFS layer allow you to quickly implement an in-place data analytics approach. This ability helps avoid unnecessary capital expenditures, increased operational costs, and time-consuming replication of  big data to a separate infrastructure.

This document describes the high-level design, performance results, and best practices for deploying Cloudera Data Platform Private Cloud Base on bare-metal infrastructure with PowerScale scale-out NAS as a shared-storage back end.

## Audience and scope

This guide is for IT architects who are responsible for the design and deployment of infrastructure and a shared-storage platform in the data center. It is also written for Hadoop administrators, data-center architects, and related specialists.

This document describes recommendations from Dell Technologies and Cloudera regarding the following topics:

- Storage array considerations
- Data network considerations
- Hardware platform considerations

## Revisions

| Date | Description |
|---|---|
| July 2021 | Initial release |
| January 2022 | Template update |

**Note**: This document may contain language from third-party content that is not under Dell Technologies' control and is not consistent with current guidelines for Dell Technologies' own content. When such third-party content is updated by the relevant third parties, this document will be revised accordingly.

## We value your feedback

Dell Technologies and the authors of this document welcome your feedback on this document. Contact the Dell Technologies team by email.

**Author:** Kirankumar Bhusanurmath, Analytics Solutions Architect, Dell Technologies

**Note**: For links to other documentation for this topic, see the PowerScale Info Hub.

### Terminology (optional)

The following table provides definitions for some of the terms that are used in this document.

**Table 1. Terminology**

| Term | Definition |
|---|---|
| NameNode | A separate server that holds metadata for every file that is stored on the DataNodes. On an PowerScale OneFS cluster, every node in the cluster acts as a NameNode. |
| DataNode | The worker node of the cluster to which the HDFS data is written. On an PowerScale OneFS cluster, every node in the cluster acts as a DataNode |
| HDFS | Hadoop Distributed File System. In an PowerScale OneFS cluster with Hadoop deployment, OneFS serves as the file system for Hadoop compute clients. |
| NodeManager | Process that starts application processes and manages resources on the DataNodes. |
| RM | ResourceManager. The resource management component of YARN. This initiates application startup and controls scheduling on the DataNodes of the cluster (one instance per cluster). |
| ToR | Top of rack. |
| ZK | ZooKeeper. A centralized service for maintaining configuration information, naming, and providing distributed synchronization and group services. |

# PowerScale distributed storage array for HDFS and bare-metal nodes as compute nodes

### Introduction

In this model, PowerScale replaces the HDFS that is shipped in Cloudera Data Platform Private Cloud Base.

In this architecture, PowerScale acts as the HDFS storage layer, and the bare-metal nodes only provide the compute resources required. Considerations for a storage component are not required, but you must ensure a reasonable oversubscription ratio between PowerScale switches and the compute node switches.

The following graphic depicts a cluster deployed across several racks (Rack1, Rack 2, and up to Rack n).

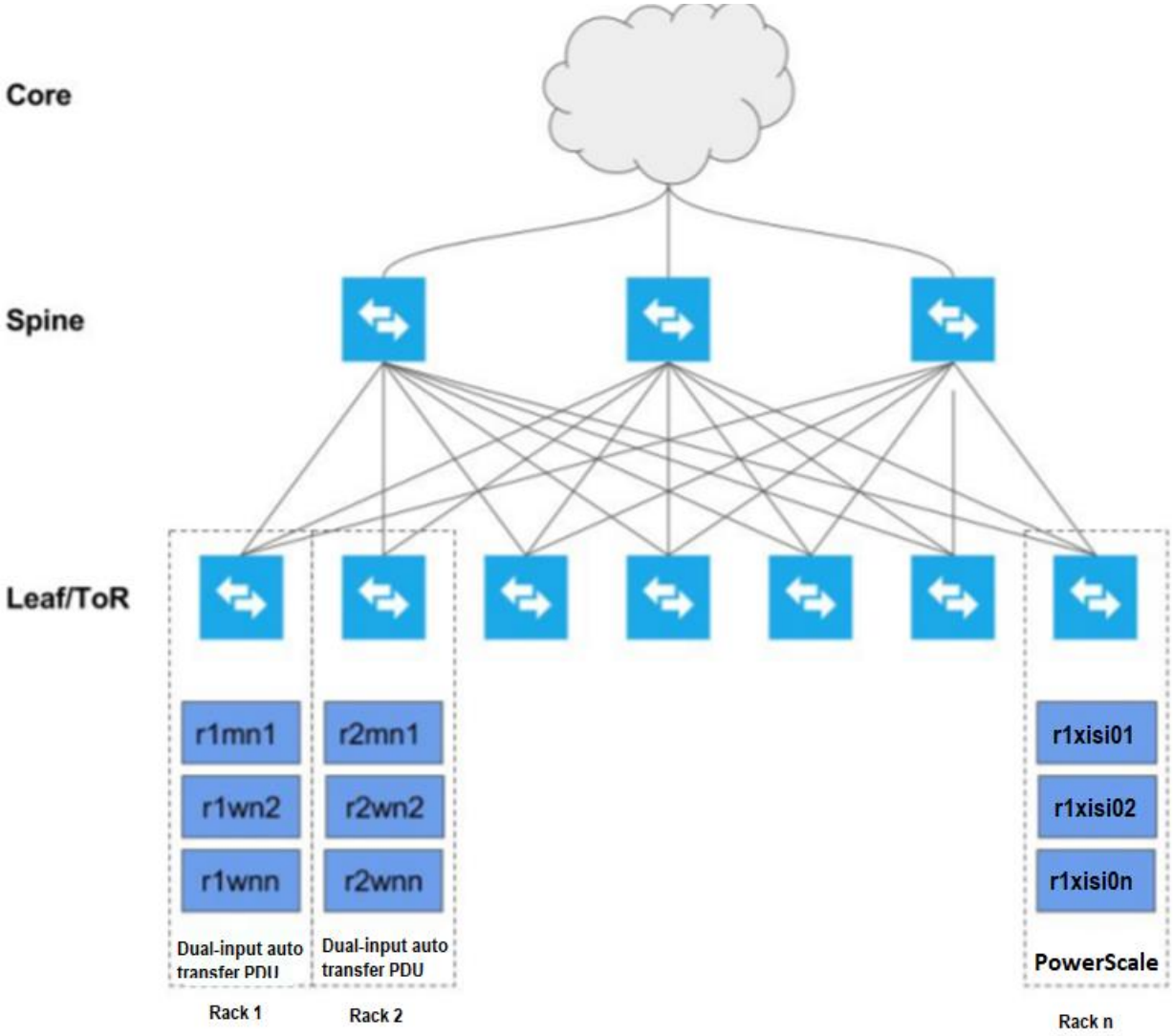


**Figure 1. Physical cluster topology**

Each host is networked to two top-of-rack (TOR) switches, which are connected to spine switches, which are connected to the enterprise network. This deployment model allows each host maximum throughput and minimum latency, while encouraging scalability. The specifics of the network topology are described in the following subsections.

## Physical cluster component list

The following table describes the cluster components, configurations, and quantity.

**Table 2. Physical cluster component list**

| Component | Configuration | Description | Quantity |
|---|---|---|---|
| Physical servers | Dell PowerEdge R740 servers<br>2 x Intel Xeon Gold 6148 CPU @ 2.4Ghz 20 core<br>768 GB RAM | Hosts that house the various NodeManager and compute instances. | Minimum 3 master + 3 worker + 1 utility + 1 edge (8 nodes) |
| NICs | 10 Gbps Ethernet NICs (minimum required) | Provide the data network services | At least one per server, although two NICs can be bonded for additional throughput. |
| Internal disk drives | 2 SSD 480 GB (RAID) | These ensure continuous service on server resets. | Two per physical server configured as a RAID 1 volume (mirrored). |
| Ethernet ToR or leaf switches | Minimum of 10 Gbps switches with sufficient port density to accommodate the compute cluster. These require enough ports to create a realistic spine-leaf topology providing ISL bandwidth above a 1:4 oversubscription ratio (preferably 1:1). | Although most enterprises have mature data network practices, consider building a dedicated data network for the Hadoop cluster. | At least two per rack. |
| Ethernet spine switches | Minimum of 40 Gbps switches with sufficient port density to accommodate incoming ISL links and ensure required throughput over the spine (for intertraffic). | Same considerations as for ToR switches. | Depends on the number of racks. |

## Logical cluster topology

The following node architecture depicts the CDP Private Cloud Base high-level architecture.

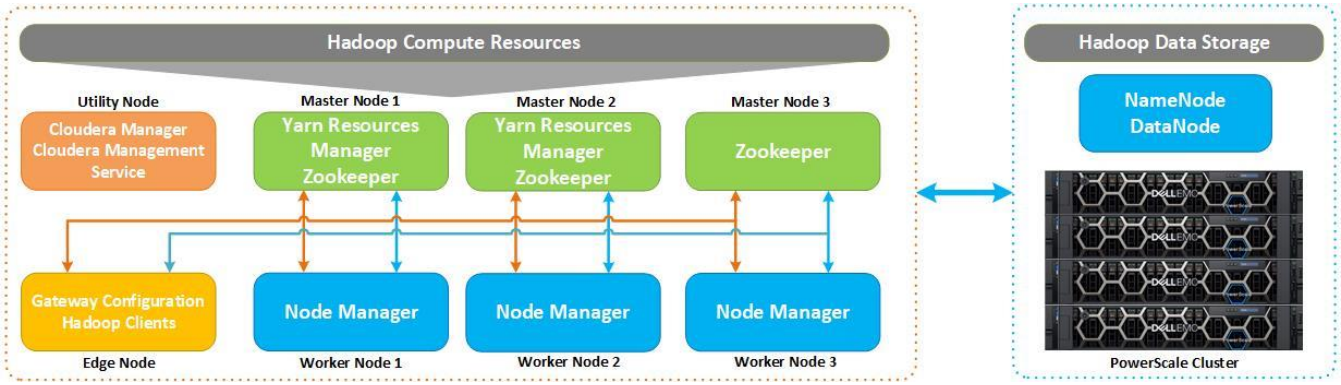


**Figure 2. Node architecture**

The cluster environment consists of multiple software services running on multiple physical server nodes.

The implementation divides the server nodes into several roles, and each node has a configuration that is optimized for its role in the cluster. The physical server configurations are divided into two broad classes:

- **Worker nodes:** Worker nodes complete most of the Hadoop processing.
- **Master nodes:** Master nodes support services needed for the cluster operation.

The high-performance network fabric connects the cluster nodes together and separates the core data network from management functions.

The minimum supported configuration is eight cluster nodes, which include three master nodes, one utility node, one edge node, and three worker nodes. Starting with a ten-node cluster with five worker nodes is a common practice. The nodes have the roles that are described in cluster node roles.

**Note**: All nodes roles listed below are required.

**Table 3. Cluster node and roles**

| Node role | Hardware configuration |
|---|---|
| Master nodes | Infrastructure |
| Utility nodes | Infrastructure |
| Edge nodes | Infrastructure |
| Worker nodes | Worker |

The following node definitions define the various nodes.

**Table 4. Node definitions**

| Role | Definition |
|---|---|
| Master nodes | Runs all the daemons that are required to manage the cluster storage and compute services |
| Worker nodes | Runs all the services that are required to store blocks of data on the local hard drives and run processing tasks against that data |
| Utility nodes | Runs Cloudera Manager and the Cloudera Management Services |
| Edge nodes | Contains all client-facing configurations and services, including gateway configurations |

## Role assignment recommendation

CDP Private Cloud Base nodes and roles describe the recommended host role assignments for a medium-sized high availability deployment.

**Table 5. CDP Private Cloud Base nodes and roles**

| Role | Definition |
|---|---|
| Master Node1 | YARN ResourceManager, ZooKeeper, JobHistory Server, Spark History Server, Kudu master |
| Master Node2 | YARN ResourceManager, ZooKeeper, Kudu master |
| Master Node3 | ZooKeeper, Kudu master (All require an odd number of masters for high availability.) |
| Worker nodes | NodeManager, Impalad, Kudu tablet server |
| Utility nodes | Cloudera Manager, Cloudera Management Service, Hive Metastore, Impala Catalog Server, Impala StateStore, Oozie, ZooKeeper (Requires a dedicated disk), Apache Atlas, Apache Ranger |
| Edge nodes | Hue, HiveServer2, Gateway configuration |

These recommendations for role assignments are intended as a starting point. Depending on the cluster size and the services that are used, the role assignments may differ. See Runtime Cluster Hosts and Role Assignments in the CDP Private Cloud Base documentation for more details.

## Sizing recommendation for physical nodes

The following table provides size recommendations for the physical nodes.

**Table 6. Physical nodes size recommendations**

| Component | Configuration | Description | Quantity |
|---|---|---|---|
| Master nodes:<br>Two-socket with 6–10 cores/socket > 2 GHz;<br>Minimum 128 GB RAM; 8–10 disks | 2U 2-socket nodes with at least 256 GB RAM | These nodes house the Master services and serve as the gateway or edge device that connects the rest of the customer's network to the Hadoop cluster. | Three (for scaling up to 100 cluster nodes). |
| Compute instances:<br>Two-socket with 6–10 cores/socket > 2 GHz;<br>Minimum 256 GB RAM<br>2 x OS disks, 8 SATA or SAS drives or 2x SSDs | At least 8 SATA or SAS drives, or 2 SSD drives for intermediate storage. | These nodes house the YARN node managers, and other required services. | Results in the field show that 1:2 ratio of PowerScale nodes to compute nodes is reasonable for most use cases. If there are heavy Impala workloads, use s 1:1.5 ratio. If PowerScale has 5 nodes, have 8 compute nodes. |
| Utility or edge nodes:<br>Two-socket with 6 to 10 cores/socket > 2 GHz;<br>Minimum 128 GB RAM; 8 to 10 disks | 2U 2-socket nodes with at least 256 GB RAM | These nodes house the management services and serve as the gateway or edge device that connects the rest of the customer's network to the Hadoop cluster. | 1 each for the cluster. |

## Sizing recommendation for storage allocation

The following table provides recommendations for storage allocation.

**Table 7. Storage size recommendation**

| Node and role | Disk layout | Description |
|---|---|---|
| **Management or master** | 2 x 500 GB OS (RAID 1)<br>Swap partition <= 2 GB<br>4 x 500 GB RAID 10 (database)<br>1 x 500 GB RAID 0 - ZooKeeper | Avoid fracturing the file system layout into multiple smaller file systems. Instead, keep a separate **'/'** and **/var**. |
| **Compute nodes** | 2 x 500 GB OS (RAID 1)<br>Approximately 20% of total DFS storage (in this case, PowerScale storage) must be provisioned as intermediate storage on these nodes. The storage can be direct-attached SAS or SATA drives, or a pair of SSD drives of sufficient capacity. Distribute the 20% of capacity evenly across all the NodeManager nodes, with its own mount-point and file system. | Avoid fracturing the file system layout into multiple smaller file systems. Instead, keep a separate **'/'** and **/var**.<br>For example, for 10 TB of total storage in PowerScale, 2 TB is needed for intermediate storage.<br>Having more or faster local spindles will speed up the intermediate shuffle stage of MapReduce. |

## Supportability and compatibility matrix

The following table provides PowerScale support for Cloudera Data Platform Private Cloud Base 7.1.6.

**Table 8. Supportability and compatibility matrix**

| CDP Private Cloud Base Components | Versions | Supportability |
|---|---|---|
| Apache YARN, Apache MapReduce | 3.1.1.7.1.6.0-297 | Yes |
| Apache ZooKeeper | 3.5.5.7.1.6.0-297 | Yes |
| Apache HBase | 2.2.3.7.1.6.0-297 | Yes |
| Apache Atlas, Apache Ranger | 2.1.0.7.1.6.0-297 | Yes |
| Apache Hive | 3.1.3000.7.1.6.0-297 | Yes |
| Apache Tez | 0.9.1.7.1.6.0-297 | Yes |
| Apache Spark | 2.4.5.7.1.6.0-297 | Yes |
| Hive on Tez | 1.0.0 | Yes |
| Apache Impala | 3.4.0.7.1.6.0-297 | Yes |
| Apache Sqoop | 1.4.7.7.1.6.0-297 | Yes |
| Apache Knox | 1.3.0.7.1.6.0-297 | Yes |
| Apache Oozie | 5.1.0.7.1.6.0-297 | Yes |

| CDP Private Cloud Base Components | Versions | Supportability |
|---|---|---|
| Hue | 4.5.0 | Yes |
| Apache Phoenix | 5.1.0.7.1.6.0-297 | Yes |
| Apache Zeppelin | 0.8.2.7.1.6.0-297 | Yes |
| Apache Livy | 0.6.0.7.1.6.0-297 | Yes |
| Apache Kafka | 2.5.0.7.1.6.0-297 | Yes |
| Apache Solr | 8.4.1.7.1.6.0-297 | Yes |
| Cruise Control | 2.0.100 | Yes |
| Schema Registry | 0.10.0.7.1.6.0 | Yes |
| Stream Replication Manager | 1.0.0 | Yes |
| Stream Messaging Manager | 2.1.0 | Yes |
| Data Analytics Studio | 1.4.2 | Yes |

For version-specific compatibility, see the Hadoop distributions and products supported by OneFS web page.

See the Cloudera Quality Assurance Test Suite (QATS) certification summary report for service-specific features supported on OneFS.

PowerScale: Cloudera CDP Private Cloud Base / QATS Execution Summary Report

# Performance testing and platform positioning

## DFSIO testing with Isilon F800

This section describes a series of industry-standard benchmark tests that analyze the performance throughput and reliability of the Dell Isilon F800 node.

### Test environment setup

The performance lab environment was configured with eight Isilon F800 nodes, and eight PowerEdge R740 servers with the following specifications (per server):

- 768 GB RAM
- 2 x Intel Xeon Gold 6148 CPU @ 2.4 GHz 20 Core
- Intel 2P X710/2 I350 rNDC
- MLNX 40 Gb 2P ConnectX3Pro adapter
- Two SSD 480 GB (RAID 1)
- CentOS Linux release 7.5.1804

The back-end network between the compute nodes and Isilon system is 40 Gbps with Jumbo Frames set (MTU-9162) for the NICs and the switch ports.

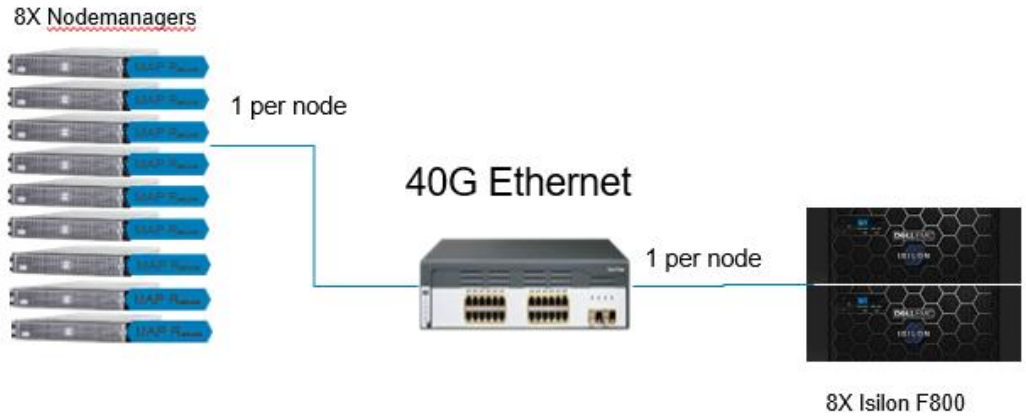


**Figure 3. Performance environment architecture diagram**

## Testing with DFSIO

This performance testing uses the DFSIO utility that comes with Cloudera enterprise. While it is not a perfect benchmark, it is useful in providing relative results within the environment. The goal was to perform a test that would keep as many cores in the NodeManagers running as continuously as possible. Since DFSIO creates a task for each file, this test was simple to deploy. With the compute side flooded, this scenario shows the effect on Isilon as an HDFS datastore.

Example DFSIO command:

```
yarn jar /opt/cloudera/parcels/CDH/lib/
/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-
client-
```

## Hadoop cluster configurations

The major configuration applied is the size of each mapper and reducer. Using some simple YARN math, we used 2 vCores of each container, and since each server had 8 vCores, this would result in 40 containers per server. The allocated memory was 8 GB per container, which used 320 GB of the total 768 GB for each run.

### Performance results

As shown below, the F800 system provides good throughput results that reflect 18 GB/s for read and 16 GB/s for writes.

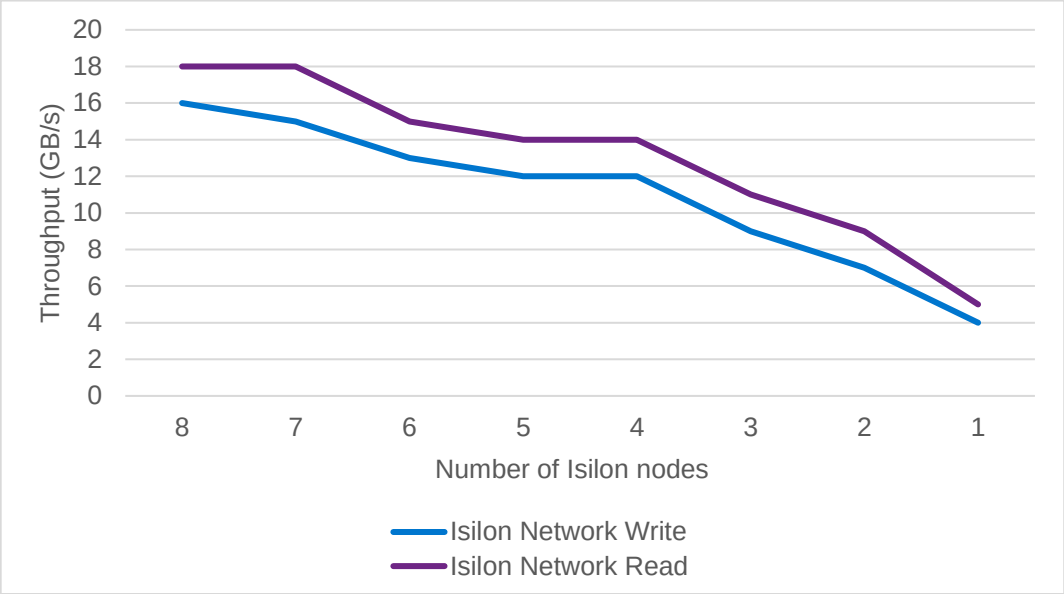


**Figure 4. Isilon throughput results**

For more detailed results, see the DFSIO testing with Isilon F800 blog post.

## PowerScale model for Hadoop use cases

The following table shows the recommended PowerScale/Isilon models for Hadoop use cases.

**Table 9. PowerScale/Isilon model for Hadoop use cases**

| Model | Use cases |
|---|---|
| F900<br>F800 | • Multi-PB high-density Hadoop clusters<br>• Multitenant, no compromise<br>• High-performance data discovery and visualization |
| F600<br>F200 | • Flash Tier for existing clusters<br>• Small edge clusters |
| H5600 | • Capacity centric<br>• Migration target for existing clusters<br>• Data discovery and visualization |
| H500 | • Starter pack: Lower storage initial deployment, lower performance<br>• Hadoop: MapReduce, Hive, Spark<br>• Hadoop Tiered Storage Solution |
| A200<br>A2000 | Archive for traditional Hadoop cluster. Only as a cold tier. |

# Platform tuning recommendations

## Introduction

This section provides tuning recommendations for the compute nodes. These recommendations are generic and should be applied only after sufficient testing. For detailed information, see Cloudera enterprise reference architecture for bare metal deployments.

For PowerScale performance tuning and best practices, see the OneFS best practice guide. For PowerScale best practices for Hadoop data storage, see Hadoop data storage best practice guide.

## CPU

### CPU BIOS settings

In the BIOS of the compute nodes, set the CPU to Performance mode for best performance.

### CPUfreq governor

The following CPUfreq governor types are available in Red Hat Enterprise Linux 7. Check other operating-system-specific governors if you are not using CentOS or Red Hat Enterprise Linux 7.

| Governor Type | Description |
|---|---|
| **cpufreq_performance** | Forces the CPU to use the highest possible clock frequency. It is meant for heavy workloads. This is a best fit for interactive workloads. |
| **Cpufreq_powersave** | Forces the CPU to stay at the lowest clock frequency possible. |
| **Cpufreq_ondemand** | This allows CPU frequency to scale to maximum under heavy load, but drops down to lowest frequency under light or no load. This is the ideal governor and subject to appropriate testing, and can be used (as it will reduce power consumption under low load or idle conditions) |
| **Cpufreq_userspace** | This allows user space programs to set the frequency. This is used with the cpuspeed daemon. |
| **Cpufreq_conservative** | Similar to the cpufreq_ondemand but it switches frequencies more gradually |

Find the appropriate kernel modules for available on the system, and use modprobe to add the driver needed.

```
# modprobe cpufreq_performance
```

After a particular governor is loaded into the kernel, enabled it using the following command.

```
# cpupower frequency-set –governor cpufreq_performance
```

## Memory

### Minimize anonymous page faults

Minimize anonymous page faults by setting **vm.swappiness = 1**, which frees them from the page cache before swapping application pages (this reduces the OOM-killer invocation).

Edit the **/etc/sysctl.conf** file in your editor of choice, and add following line.

```
vm.swappiness=1
```

Then, run the following:

```
# sysctl –p
# sysctl –a|grep "vm.swappiness"
```

### Disable transparent huge-page compaction

```
echo "never" >
/sys/kernel/mm/redhat_transparent_hugepage/enabled
```

### Disable transparent huge-page defragmentation

```
echo "never" > /sys/kernel/mm/redhat_transparent_hugepage/defrag
```

Add these commands to **/etc/rc.local** to ensure that transparent huge page compaction and defragmentation remain disabled across reboots.

## Network

The back-end network between compute nodes and PowerScale should be 40 Gbps with Jumbo Frames set (MTU=9162) for the NICs and the switch ports.

### Compute nodes network tuning

Add the following parameters in **/etc/sysctl.conf**.

Disable TCP timestamps to improve CPU utilization (this is an optional parameter and will depend on your NIC vendor).

```
net.ipv4.tcp_timestamps=0
```

Enable TCP sacks to improve throughput.

```
net.ipv4.tcp_sack=1
```

Increase the maximum length of processor input queues.

```
net.core.netdev_max_backlog=250000
```

Increase the TCP max and default buffer sizes using setsockopt().

```
net.core.rmem_max=4194304
net.core.wmem_max=4194304
net.core.rmem_default=4194304
net.core_wmem_default=4194304
net.core.optmem_max=4194304
```

Increase memory thresholds to prevent packet dropping.

```
net.ipv4.tcp_rmem=4096 87380 4194304
net.ipv4.tcp_wmem=4096 65536 4194304
```

Set the socket buffer to be divided evenly between TCP window size and application buffer.

```
net.ipv4.tcp_adv_win_scale=1
```

### Verify NIC advanced features

Verify which features are available with your NIC using **ethtool**.

```
$ sudo ethtool -k <ethX>
```

There are various offload capabilities that modern NICs (and especially high-performance NICs) have, and it is always recommended to enable them.

A few features such as tcp-segmentation-offload (TSO), scatter-gather (SG), and generic-segmentation-offload (GSO) are good features to enable (if not enabled by default).

### NIC ring buffer configurations

Check existing ring buffer sizes.

```
$ ethtool -g <ethX>
```

After checking the preset maximum values and the current hardware settings, the following command can be used to resize the ring buffers:

```
# ethtool -G <interface> rx <newsize>
```

Or

```
# ethtool -G <interface> tx <newsize>
```

**Note**: The ring buffer sizes depend on the network topology to a certain degree and might need to be tuned depending on the nature of the workload. For 10 G NICs, setting the RX and TX buffers to maximum is reasonable. This setting should be tuned based on the network architecture and type of traffic.

## Storage

### Disk and FS mount options

Disable **atime** from the data disks (and root fs) using the **noatime** option during mounting of the FS.

In the /etc/fstab file, ensure that the appropriate file system has the noatime mount option specified.

```
LABEL=ROOT /          xfs     noatime         0 0
```

Create separate mount points for separate disk drives, and provision all of them for the scratch space. For example, **yarn.nodemanager.local-dirs** is a comma-separated list of local-directories that you can configure to be used for copying files during localization. The

idea behind allowing multiple directories is to use multiple disks for localization. This practice helps both failover (one or a few disks going bad does not affect all containers) and load balancing (no single disk is bottlenecked with writes). Individual directories should be configured, if possible, on different local disks.

### FS creation options

For FS creation, enable journal mode, reduce superuser block reservation from 5% to 1% for root (using the -m1 option), and use the **sparse_super,dir_index,extent** options (minimize number of super block backups and use b-tree indexes for directory trees, extent-based allocations).

```
# mkfs.xfs –t xfs –m1 –O sparse_super,dir_index,extent,has_journal
/dev/sdb1
```

## Tuning OneFS for HDFS operations

### OneFS TCP tuning

The default TCP stack of OneFS requires tuning for Hadoop and 40 GbE connectivity. The tuning must be done within the CLI directly on PowerScale. A tcptune.sh script is available at GitHub.

Run sh ./tcptune.sh Max to make the changes. An example script run is shown below:

Before changes:

```
isilon# sh ./tcptune.sh Max
Tuning TCP stack to Max
TCP sysctls before...
kern.ipc.maxsockbuf=2097152
net.inet.tcp.sendbuf_max=2097152
net.inet.tcp.recvbuf_max=2097152
net.inet.tcp.sendbuf_inc=8192
net.inet.tcp.recvbuf_inc=16384
net.inet.tcp.sendspace=131072
net.inet.tcp.recvspace=131072
efs.bam.coalescer.insert_hwm=209715200
efs.bam.coalescer.insert_lwm=178257920
```

After changes:

```
Apply tuning...
Value set successfully
Value set successfully
Value set successfully
Value set successfully
Value set successfully
Value set successfully
Value set successfully
Value set successfully
TCP sysctls after...
kern.ipc.maxsockbuf=104857600
net.inet.tcp.sendbuf_max=52428800
net.inet.tcp.recvbuf_max=52428800
```

```
net.inet.tcp.sendbuf_inc=16384
net.inet.tcp.recvbuf_inc=32768
net.inet.tcp.sendspace=26214400
net.inet.tcp.recvspace=26214400
efs.bam.coalescer.insert_hwm=209715200
efs.bam.coalescer.insert_lwm=178257920
net.inet.tcp.mssdflt=8948
```

### Block sizes

On a PowerScale cluster, the default HDFS block size is 128 MB, which optimizes performance for most use cases. Aligning HDFS client block size with OneFS HDFS block size lets PowerScale nodes read and write in large blocks, which can decrease drive-seek operations and increase performance for MapReduce jobs.

### HDFS connection and limits

A four-node PowerScale cluster would support 1,600 parallel HDFS connections in a minute. That is 1600 YARN containers before tasks begin to fail due to timeout.

If you consider a compute machine with dual 24-core processors, that would be 48 cores with hyperthreading enabled, and the total cores will be 48 * 2 = 96. Considering one core for each container, there will be 96 containers a Physical compute server.

From the above details, we can determine Fan-in ration as 1600 / 96 ~= 16 servers or 4:1 compute server to PowerScale node ratio.

### HDFS Statistics for tuning

Run the **isi statistics** command to obtain statistics for client connections, the file system, and protocols. For HDFS protocol statistics, run **isi statistics pstat –protocol=hdfs.**

By analyzing the columns titled NetIn and NetOut, you can determine whether HDFS connections are predominantly reading or writing data. Looking at the distribution if input and output across all the nodes shows whether Hadoop is using all the nodes for a MapReduce job.

### Hadoop scratch space

Local high-speed disk SSD is preferred for scratch space. A local direct-attached disk will always be faster than a NFS-mounted file system for a scratch disk. This represents a standard analytics workflow: The shared data—remote/HDFS; the scratch/shuffle space—should be local to node for better performance. If sufficient local disk is not available, use an NFS mount to provide extra disk space to a disk lite client through NFS from the same Isilon cluster. However, it will always have performance implications, since you are placing this data across the wire on the same shared resource that HDFS is using. This setting would allow modification to the location of the mapred shuffle space if needed.

### Storage pools: NodePools for different datasets

A difficulty arises if you are analyzing two different datasets, one with large files and another with small files. In such a case, you might be able to cost-effectively store and optimally analyze both types by using storage pools. Storage pools let you group different files by attributes and then store them in different pools of storage: Small files can be routed by a OneFS policy to SSDs, for instance, and large files can be routed to X-Series

nodes. Then, you can use OneFS SmartConnect zones to associate a set of compute clients with each dataset to optimize the performance of the MapReduce jobs that analyze each set. For more information, see the section Align datasets with storage pools. You could, for example, direct one compute cluster to S-series DataNodes and another compute cluster to X-series DataNodes with Isilon virtual racks. For more information, see the section on rack awareness.

## Data protection

OneFS takes a more efficient approach to data protection than HDFS. The HDFS protocol, by default, replicates a block of data three times to protect it and to make it highly available (3X Mirroring). Instead of replicating the data, OneFS stripes the data across the cluster over its internal InfiniBand network and protects the data with forward error correction codes.

**Best practice:** Use a OneFS protection policy that meets the requirements and suggested level for the Isilon cluster configuration. If more nodes are added, the protection policy may need to be reevaluated.

## Data access patterns

For most if not all workflows the data access pattern is best represented by the **streaming** setting. This will increase sequential-read performance, OneFS will stripe data across more drives and prefetches data well before data requests. For large quantities of small files (<1 MB), concurrent access may be better because the impact of prefetch will be much less.

The following graph shows the performance of H5600 for streaming and concurrent data access.

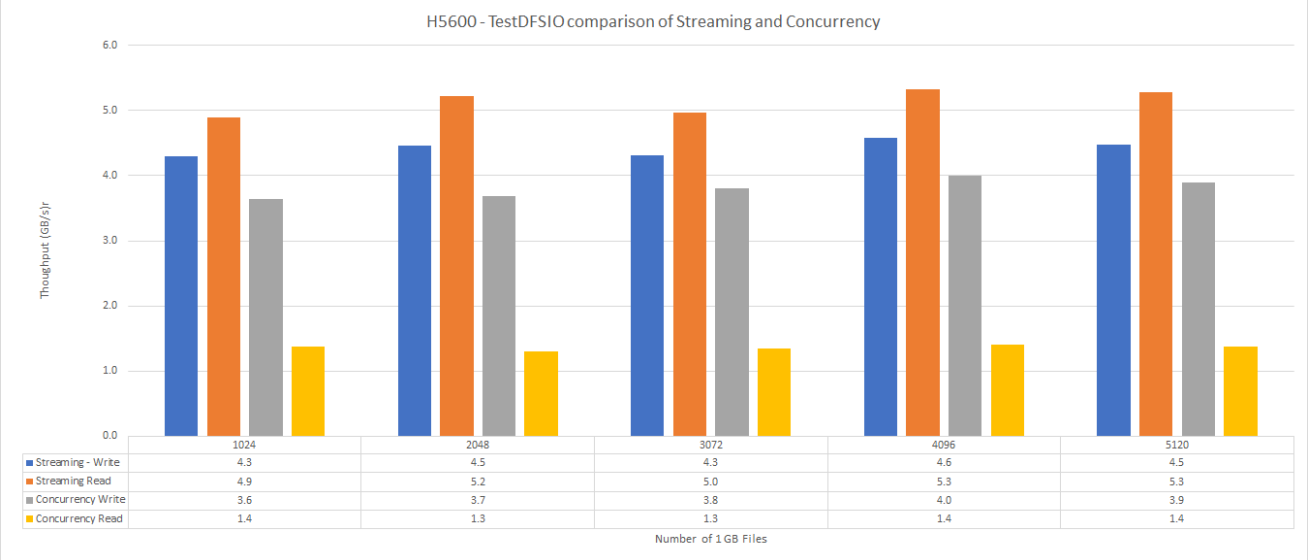


**Figure 5. F800 streaming and concurrency**

The H5600 streaming and concurrency measured here is the aggregate throughput of a DFSIO job with different counts of 1 GB files.

# Technical support and resources

**Technical support**

Dell.com/support is focused on meeting customer needs with proven services and support.

**Related resources**

- Cloudera CDP Private Cloud Base 7.1.6 on PowerScale Install guide
- Cloudera QATS CDP on PowerScale Execution summary report
- OneFS 8.2.0 HDFS Reference Guide
- OneFS 8.2.0 Web Admin Guide
- Using CDH with Isilon Storage
- Using Hadoop with OneFS info hub
- Isilon best practice guide for Hadoop data storage