



Cloudera Enterprise Reference Architecture on EMC DSSD D5 Storage Appliance



Important Notice

© 2010-2016 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, Cloudera Impala, Impala, and any other product or service names or slogans contained in this document, except as otherwise disclaimed, are trademarks of Cloudera and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Cloudera or the applicable trademark holder.

Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation. All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Cloudera.

Cloudera may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Cloudera, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property.

The information in this document is subject to change without notice. Cloudera shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Cloudera, Inc.

1001 Page Mill Road, Building 2

Palo Alto, CA 94304-1008

info@cloudera.com

US: 1-888-789-1488

Intl: 1-650-843-0595

www.cloudera.com

Release Information

Version: 5.8

Date: July 15, 2016

Table of Contents

[Executive Summary](#)

[DSSD D5 Appliance Overview](#)

[Audience and Scope](#)

[Glossary of Terms](#)

[DSSD D5 Storage Appliance for HDFS and Bare-Metal Nodes as Compute Nodes](#)

[Software Compatibility](#)

[Network Architecture](#)

[Physical Cluster Topology](#)

[Physical Cluster Component List](#)

[Logical Cluster Topology](#)

[Physical DSSD Client/D5 Topology](#)

[Cluster Management](#)

[Setting up the Cluster](#)

[Before You Start](#)

[Setting Up the Cluster Using Cloudera Manager](#)

[Upgrade and Downgrade](#)

[Security](#)

[Access Control to Data Stored on the DSSD D5 Appliance](#)

[Security Implications with Short Circuit Reads \(SCR\)](#)

[DSSD Specific Tuning Requirements](#)

[CPU](#)

[Identify CPUs and NUMA Nodes](#)

[Determine the NUMA Node Attached to the vpci Driver](#)

[Select a CPU Identifier to Assign to the DSSD DataNode](#)

[Short Circuit Reads \(SCR\)](#)

[HBase and Impala](#)

[General Platform Tuning Recommendations](#)

[CPU](#)

[CPU BIOS Settings](#)

[CPUfreq Governor](#)

[Memory](#)

[Minimize Anonymous Page Faults](#)

[Disable Transparent Hugepage Compaction and Defragmentation](#)

[Network](#)

[Verify NIC Advanced Features](#)

[NIC Ring Buffer Configurations](#)

[Storage](#)

[Disk/FS Mount Options](#)

[FS Creation Options](#)

[Application Tuning Recommendations](#)

[HBase](#)

[HDFS Parameters](#)

[HBase Parameters](#)

[References](#)

Executive Summary

This document is a high-level design and best-practices guide for deploying Cloudera Enterprise on a cluster backed by the [EMC® DSSD™ D5™](#) storage appliance.

DSSD D5 Appliance Overview

DSSD D5 is a ground-up design of an all-flash storage appliance. It provides ultra-dense, high-performance, highly available, and very low latency shared flash storage. DSSD D5 can connect redundantly to up to 48 compute nodes through PCIe Gen3 client cards. Each client can directly access the DSSD D5 pool of flash memory as if it were local to the client CPU. The result is extremely low latency, high IOPS, and high bandwidth that is superior to direct server-attached flash. DSSD D5 also offers the data-sharing capabilities and operational efficiencies available only in fabric-attached storage (all-flash arrays).

Audience and Scope

This guide is for IT architects who are responsible for the design and deployment of high-performance infrastructure backed by DSSD D5 in the data center, Hadoop administrators and architects who are data center architects, engineers and others who collaborate with data center specialists.

This document describes Cloudera recommendations on the following topics:

- DSSD D5 hardware configuration considerations and best practices
- Cluster hardware/platform considerations
- Data network considerations
- Cluster performance tuning guidelines
- Application (HBase) tuning guidelines

Glossary of Terms

Term	Description
Apache HBase	The high-performance, distributed data store built for Apache Hadoop.
Apache Hive	Data warehouse infrastructure that provides easy, familiar batch processing for Apache Hadoop.
Apache Oozie	Workflow scheduler system to manage Apache Hadoop jobs.
Apache Spark	Open standard for flexible in-memory data processing for batch, real-time, and advanced analytics. Through the One Platform initiative, Cloudera is committed to helping the community adopt Spark as a replacement for MapReduce in the Hadoop ecosystem as the default data execution engine for analytic workloads.

CM	Control Module, a DSSD D5 appliance component. Not to be mistaken with commonly used abbreviation for Cloudera Manager
CDH	The 100% open source Hadoop distribution from Cloudera. It includes the leading Hadoop ecosystem components to store, process, discover, model, and serve unlimited data. CDH is based entirely on open standards for long-term architecture.
Cloudera Manager	End-to-end management solution for CDH, Impala, and Cloudera Search. Cloudera Manager enables easy and effective provisioning, monitoring, and management of Hadoop clusters and CDH installations.
Cloudera Navigator	Complete data governance solution for Hadoop, offering data discovery, continuous optimization, audit, lineage, metadata management, and policy enforcement. As part of Cloudera Enterprise, Cloudera Navigator enables high-performance agile analytics, continuous data architecture optimization, and regulatory compliance.
Cloudera Search (Solr)	A fully integrated search tool, powered by Apache Solr, that makes Hadoop accessible through integrated full-text search.
DSSD D5	DSSD high-performance storage appliance.
DHP	DSSD Hadoop API plug-in.
DataNode	Worker nodes of the cluster to which the HDFS data is written. Also refers to the process that provides the DataNode functionality.
Flood (libflood)	Software that runs on both the client and the D5 appliance, providing the DMA engine that moves application I/O requests from the client user space directly to the flash modules (FMs). This is orchestrated by Flood using NVMe™ virtual PCIe ports, mapped to the physical PCIe ports. Client administrators can create and manage objects using the Flood client CLI or through a browser user interface (BUI).
FM	Flash module, a DSSD D5 appliance component.
HDD	Hard disk drive.
HDFS	Hadoop Distributed File System.
High Availability	Configuration that addresses availability issues in a cluster. In a standard configuration, the NameNode is a single point of failure (SPOF). Each cluster has a single NameNode, and if that machine or process became unavailable, the cluster as a whole is unavailable until the NameNode is either restarted or brought up on a new host. The secondary NameNode does not provide failover capability.

	High availability enables running two NameNodes in the same cluster: the active NameNode and the standby NameNode. The standby NameNode allows a fast failover to a new NameNode in case of machine crash or planned maintenance.
HUE	Hadoop User Experience; the open source web GUI that lets you easily interact with Apache Hadoop.
Impala	Currently an Apache Incubator project, the open source, analytic MPP database for Apache Hadoop that provides the fastest time-to-insight.
Intermediate storage	Also called temporary or spill storage or scratch space; storage space used by MapReduce, Spark, and Impala to store intermediate or temporary data when the data cannot fit in memory.
ISL	Inter-switch link.
JBOD	Just a bunch of disks. In contrast to disks configured through software or hardware with redundancy mechanisms for data protection.
Job History Server	A component of YARN; a process that archives job metrics and metadata.
Jumbo frames	Ethernet frames with more than 1500 bytes of payload.
LRO	Large receive offload; a technique that improves throughput of network connections. LRO coalesces multiple incoming packets from a single stream into a large receive buffer before passing them up the networking stack.
Master node	A server node that provides management and monitoring services for the cluster. No computational or storage services are provided by a master node.
Multihoming	In this document, refers to having multiple data networks for the cluster.
NameNode	The metadata master of HDFS essential for the integrity and proper functioning of the distributed filesystem. Also refers to the process that provides the NameNode functionality.
NIC	Network interface card.
NodeManager (NM)	A component of YARN; a process that starts application processes and manages resources on the DataNodes.
NUMA	Nonuniform memory access. Addresses memory access latency in multi-socket servers, where memory that is remote to a core (that is, local to another socket) needs to be

	accessed. This is typical of SMP (symmetric multiprocessing) systems, and there are several strategies to optimize applications and operating systems.
OOB management or BMC network	Out-of-band management and Baseboard Management Controller networks; dedicated network for managing server hardware.
OS	Operating system.
PCIe	PCI Express (Peripheral Component Interconnect Express).
PDU	Power distribution unit.
QJM QJN	<p>Quorum Journal Manager. Provides a fencing mechanism for high availability in a Hadoop cluster. This service is used to distribute HDFS edit logs to multiple hosts (at least three are required) from the active NameNode. The standby NameNode reads the edits from the JournalNodes and constantly applies them to its own namespace. In case of a failover, the standby NameNode applies all of the edits from the JournalNodes before promoting itself to the active state.</p> <p>Quorum JournalNodes. Nodes on which the journal services are installed.</p>
RAID	Redundant array of independent disks. A data storage virtualization technology that combines multiple physical disk drive components into a single logical unit for the purposes of data redundancy, performance improvement, or both.
ResourceManager (RM)	A component of YARN; a process that manages compute resources and schedules compute jobs for the Hadoop cluster.
ToR	Top of rack.
TSO	TCP segmentation offload.
Worker node	A server node that provides computational and storage services for the cluster. Usually no management or monitoring services are provided by a worker node.
YARN	Provides open source resource management for Hadoop, so you can move beyond batch processing and open your data to a diverse set of workloads, including interactive SQL, advanced modeling, and real-time streaming.
ZooKeeper	A centralized service for maintaining configuration information, naming, and providing distributed synchronization and group services.

DSSD D5 Storage Appliance for HDFS and Bare-Metal Nodes as Compute Nodes

In this architecture, the DSSD storage appliance provides the storage backend for HDFS DataNode. DSSD provides a Hadoop plugin that replaces the HDFS DataNode shipped in CDH. The bare-metal nodes provide the compute resources needed. Local storage for the cluster nodes is still required to host the OS and provide intermediate storage.

Software Compatibility

This document is intended for:

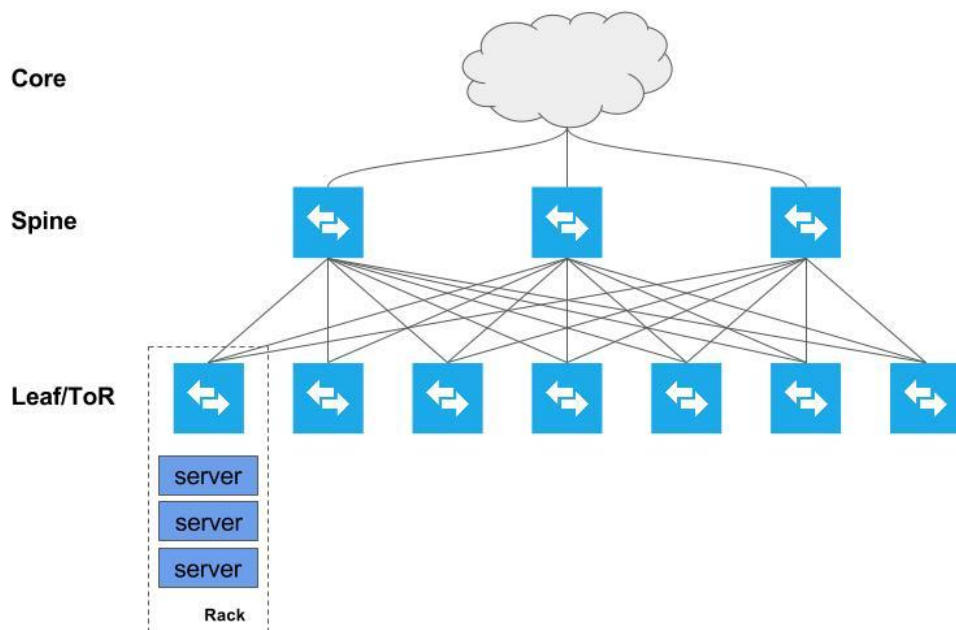
- Cloudera release 5.8 (including Cloudera Manager and CDH) and beyond.
- DSSD Hadoop Plugin release 1.2 and beyond.
- DSSD firmware release 201602.3.0 and beyond.

Warning:

The Cloudera and DSSD software/firmware releases mentioned above are **not** backward compatible. Upgrade from earlier releases is not supported.

Network Architecture

A spine/leaf model is recommended for the cluster data network architecture, as shown in the following diagram. This model provides the best balance between performance and redundancy.



Note:

DSSD D5 uses a direct PCIe link to transfer data with servers in the cluster and does not use the cluster data network.

The following table describes the requirements for the network links.

Logical Network	Connection	Description
Cluster Data Network	Bonded 10 Gbps Ethernet, jumbo frame enabled	Dedicated network for cluster internal communication. Cloudera Manager uses this network to manage server nodes in the cluster.
OOB management and/or BMC network (Optional)	1 Gbps Ethernet	For server management, vendor specific.

Important:

- Cloudera does not support multihoming for the Cluster Data Network.
- Do not use OOB management and BMC network for the Cluster Data Network.

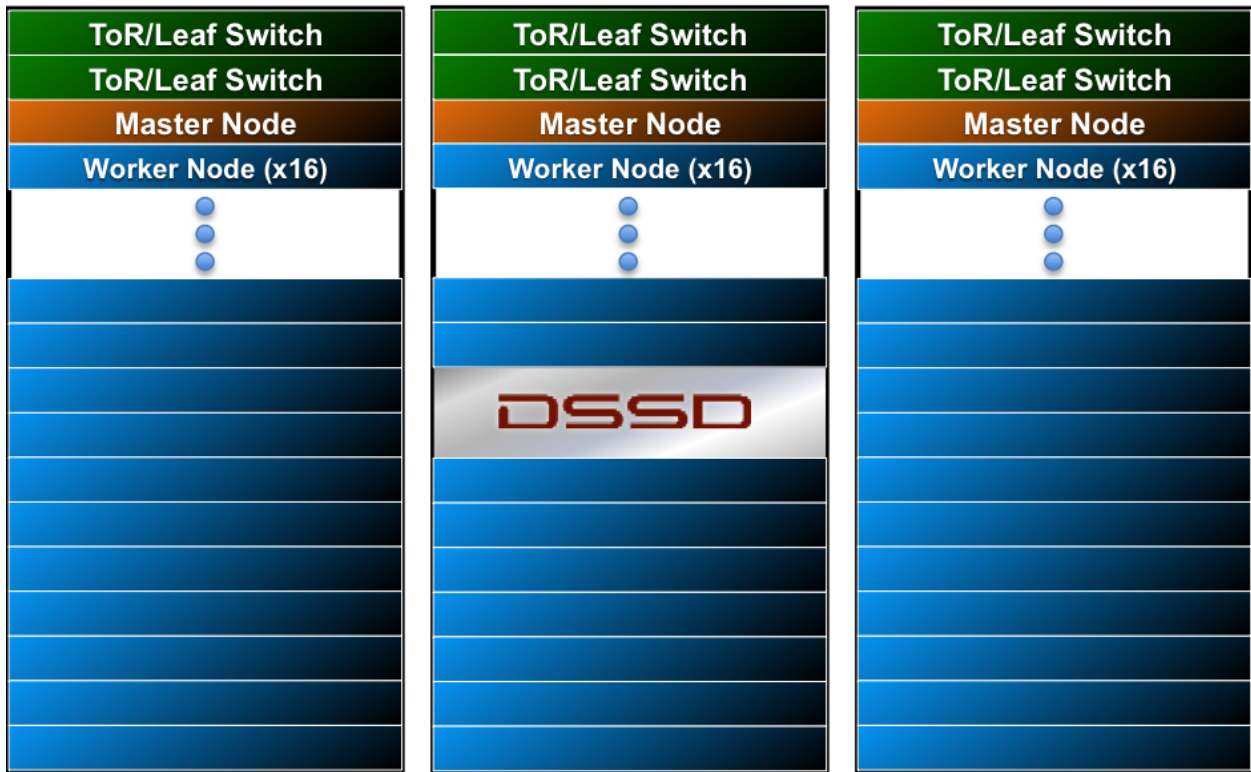
Physical Cluster Topology

The DSSD D5 appliance connects to servers through a PCIe link. A client card must be installed in each server that needs to access the DSSD D5. Each client card provides two ports for redundant connection to the DSSD D5. The following diagram shows the physical connection of a client card and a DSSD D5 appliance. See the [DSSD Installation & Service Guide](#) for more details.

**Important:**

The DSSD client card must be installed in specific PCIe slots on the compute node. See the [DSSD certified hardware list](#) for compatible servers.

The following diagram shows an example design of DSSD-backed Hadoop cluster with standard 42 RU racks. The diagram assumes 1 RU ToR/Leaf switches, 2 RU servers for master and worker nodes, and a DSSD D5 appliance with a form factor of 5 RU. 1 RU server can be used for worker nodes for higher rack density.



In order to scale both storage capacity as well as compute capacity, add more DSSD D5 appliances and/or worker nodes. Follow the physical layout diagram shown above as a guideline for resource augmentation.

Important:

This architecture supports multiple DSSD D5 appliances. Each DSSD D5 appliance can support up to 48 worker nodes. A worker node can only be connected to a single DSSD D5 appliance.

Physical Cluster Component List

Component	Configuration	Description	Quantity
Physical servers	2-socket, 6-10 physical cores per socket > 2 GHz; must be certified by DSSD.	Hosts that house the various NodeManager, compute instances, and DSSD client software.	Minimum 3 master + 48 worker (51 nodes)
NICs	Dual-port 10 Gbps Ethernet NICs. The connector type depends on the network design; could be SFP+ or Twinax.	Provide the data network services.	1 (dual port) per server. 2 NICs with one port each can be used for resiliency against NIC failures.

Internal HDDs	Standard OS sizes (300 GB—1 TB) drives. Can be larger, but not required.	Ensure continuity of service on server resets.	2 per physical server configured as a RAID 1 volume (mirrored).
Ethernet ToR/leaf switches	Minimally 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 2 per rack, for redundancy.
Ethernet spine switches	Minimally 10 Gbps switches with sufficient port density to accommodate incoming ISL bandwidth and ensure required throughput over the spine (for inter-rack traffic).	Same considerations as for ToR switches.	Depends on the number of racks.
DSSD client card	Must be installed on specific PCIe slot in the server.	Provide data link to the DSSD D5 appliance.	1 per worker node.

Note:

The worker nodes that run DataNode services **must** have access to the DSSD D5 appliance.

Logical Cluster Topology

For YARN NodeManager instances, data protection at the HDFS level is not required because the physical nodes are running only the compute part of the cluster.

The minimum requirements to build out the cluster are:

- 3 master nodes
- 5 worker nodes

This document assumes the cluster has 3 master modes and 48 worker nodes.

The following table identifies service roles for different node types.

	Master Node	Master Node	Master Node	Worker Nodes
ZooKeeper	ZooKeeper	ZooKeeper	ZooKeeper	
HDFS	NN, QJN	NN, QJN	QJN	DSSD-DN
YARN	ResourceManager	ResourceManager	Job History Server	NodeManager
Hive			MetaStore, WebHCat, HiveServer2	

Management Services	Cloudera Manager Agent	Cloudera Manager Agent	Cloudera Manager, Cloudera Manager Agent, Oozie Server, Misc. Management Services	Cloudera Manager Agent
Cloudera Navigator			Navigator Services, Key Management Services	
HUE			HUE	
HBase	HMaster	HMaster	HMaster	RegionServer
Impala			StateStore, Catalog Server	Impala Daemon (impalad)
Spark				Runs on YARN
Solr				Search
DSSD D5 Software Components				DSSD client software

Important:

- ZooKeeper must be assigned a dedicated spindle because it is sensitive to disk latency.
- DSSD client software must be installed to access the DSSD D5 appliance.

The following table provides size recommendations for the physical nodes.

Component	Configuration	Description	Quantity
Master Nodes	2-socket with 6-10 physical cores per socket > 2 GHz; minimally 128 GB RAM; 8-10 disks.	These nodes house the Cloudera Master services and serve as the gateway/edge device that connects the rest of the customer network to the Cloudera cluster.	3
Worker Nodes	2-socket with 6-10 or more physical cores per socket > 2 GHz; minimally 256 GB RAM 2 x OS disks, 2 x SATA or SAS drives or 2 x SSDs, 1 x DSSD client card.	These nodes house the DHP HDFS DataNodes and YARN node managers and any additional required services.	48

Note:

A higher CPU core count is recommended for worker nodes because DSSD client software requires dedicated CPU cores for high-performance I/O.

The following table provides recommendations for storage allocation.

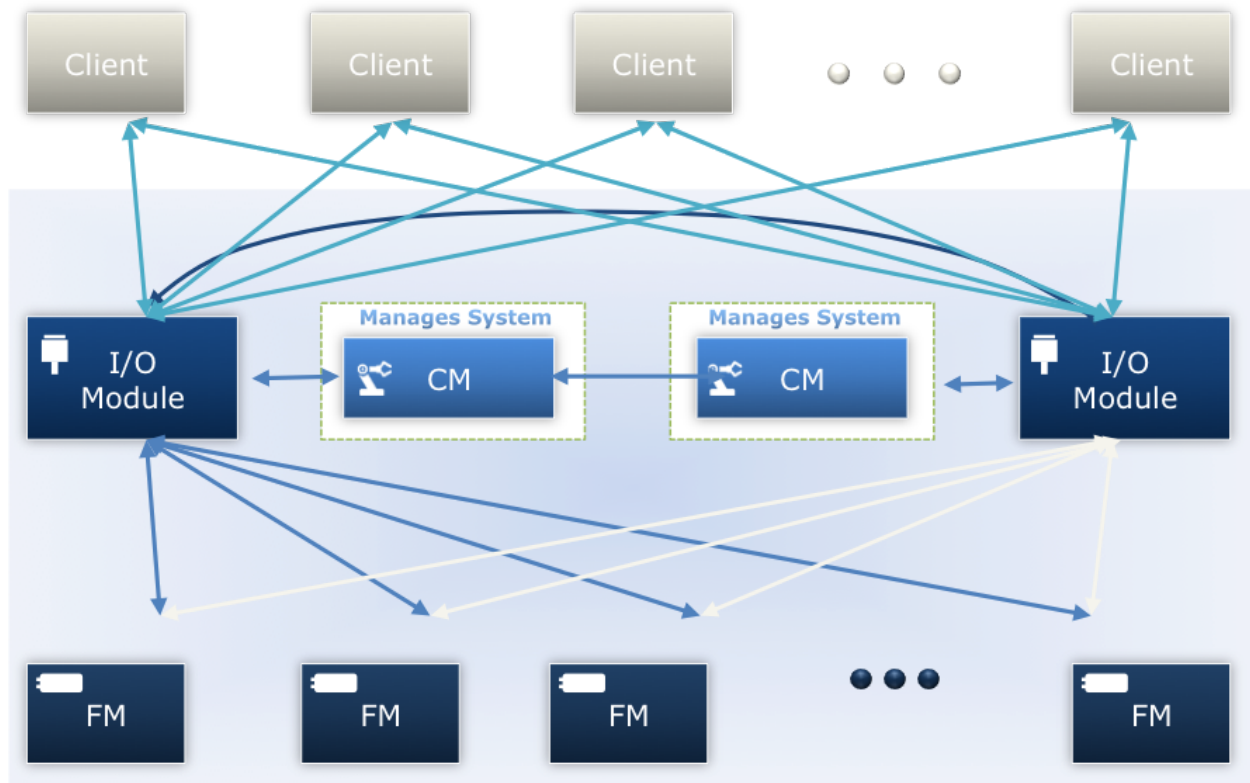
Node	Disk Layout	Notes
Master	<ul style="list-style-type: none"> • 2 x 500 GB OS (RAID 1) • Swap partition <= 2 GB • 4 x 500 GB RAID 10 (database) • 1 x 500 GB RAID 0 (ZooKeeper) 	<p>Avoid fracturing the filesystem layout into multiple smaller filesystems. Instead, keep a separate “/” and “/var”.</p> <p>Avoid sharing the ZooKeeper disk with any other services.</p>
Worker	<ul style="list-style-type: none"> • 2 x 500 GB OS (RAID 1) • Approximately 20% of total HDFS storage needs to be provisioned as intermediate storage across these nodes. 	<p>Avoid fracturing the filesystem layout into multiple smaller filesystems. Instead, keep a separate “/” and “/var”.</p> <p>More or faster local spindles will speed up the intermediate shuffle stage of MapReduce.</p>

Important:

- The maximum usable capacity provided by a single DSSD D5 appliance is 100 TB.
- The default HDFS block size for a CDH cluster powered by DSSD D5 is 512 MB (optimized for HBase).
- A single DSSD D5 appliance supports up to 6.9 million HDFS blocks. Cloudera Manager by default configures 4 GB for the NameNode heap size based on the 512 MB default block size and the 100 TB capacity limit of DSSD D5. The NameNode heap size can be increased. Cloudera recommends configuring 1 GB heap space per 1 million HDFS blocks.

Physical DSSD Client/D5 Topology

The following diagram highlights the logical connection between a DSSD D5 appliance and DSSD D5 clients (worker nodes), as well as internal data path for the D5. See the [DSSD documentation](#) for details about D5 architecture.



Cluster Management

Setting up the Cluster

Cloudera Manager automates the process of setting up the cluster. See the [Cloudera CDH 5.8 installation guide](#) for details.

Important:

The DSSD D5 appliance is configured and managed through its own management console. Cloudera Manager does not provide management of the DSSD D5 appliance.

The following sections highlights key steps for setting up a DSSD D5 powered Hadoop cluster using Cloudera Manager.

Before You Start

Before installing Cloudera Manager, you must complete the following tasks, using tools and documentation provided for the DSSD appliance:

1. Install and rack the DSSD Storage Appliance.
2. Install the DSSD PCIe cards in the DataNode hosts.
3. Connect the DataNode hosts to the DSSD appliance.
4. Install and configure the DSSD drivers.
5. Install and configure the DSSD client software.

6. Create a volume on the DSSD Storage Appliance for the DataNodes.
7. Determine the NUMA node attached to the DSSD `vpci` driver on worker nodes.
8. Review [DSSD documentation](#) regarding configuration requirements for multiple DSSD D5 appliances.
9. Review [DSSD documentation](#) regarding configuration requirements for HDFS to support DSSD D5 appliance failover.

Setting Up the Cluster Using Cloudera Manager

After completing the steps above, you can install Cloudera Manager and set up the cluster. Detailed steps of the installation are described in the [Cloudera CDH 5.8 installation guide](#). Key steps for cluster installation using Cloudera Manager are highlighted below:

1. Enable **DSSD Mode** before proceeding with the cluster setup wizard.
2. Specify the name of volume created for the DataNodes.
3. Specify the maximum HDFS capacity provided by each DSSD D5 in the cluster.
4. Specify a parcel directory or repository for DSSD Hadoop Plugin parcels.

Important:

- The DSSD Hadoop Plugins can only be installed as parcels.
- By default, Cloudera Manager allocates the entire capacity of the DSSD D5 appliance (100 TB) for the DataNodes.
- All worker nodes connected to the same DSSD D5 appliance must be assigned with the same rack ID, even if they are located in different physical racks.

Upgrade and Downgrade

Upgrade and downgrade of the DSSD Client software (Flood) and DSSD D5 appliance are not supported by Cloudera Manager. See the [DSSD Administrator's Guide](#) for details.

Warning:

- Upgrade from releases before Cloudera 5.8 and DHP 1.2 is not supported.
- All cluster services must be stopped to upgrade the D5 appliance.

Security

Access Control to Data Stored on the DSSD D5 Appliance

Currently DSSD does not support fine-grained access control to data stored on the DSSD D5 appliance. If a client server has access to the DSSD D5 appliance, it will have access to **all** data stored on the appliance. See the *DSSD Administrator's Guide* for more details.

To control access to data stored on the DSSD D5 appliance, Cloudera Manager assigns the DSSD `vpci` device (installed in `/dev/dssd/` by default) to the same group as `hdfs` user, thereby limiting access to the DSSD D5 appliance.

Note:

HDFS data-at-rest encryption is transparent to the storage backend and is supported for this architecture.

Security Implications with Short Circuit Reads (SCR)

With CDH and Cloudera Manager, the DSSD-backed cluster can achieve the same level of security as regular disk- or SSD-backed clusters. However, because DSSD D5 does not yet support fine-grained access control of data, when SCR is enabled, the `hbase` and `impala` users (for the HBase and Impala services) must be in the same Linux user group as the `hdfs` user. This allows the `hbase` and `impala` users access to the DSSD D5 appliance using SCR. Keep in mind that both users will have access to all data stored on the DSSD D5 appliance.

Important:

Administrators should be aware of the SCR security limitation when deciding if SCR should be enabled for HBase or Impala.

DSSD Specific Tuning Requirements

For optimal performance, DSSD has specific tuning requirements for nodes with access to the DSSD D5 appliance. This section highlights how to configure some of these key parameters. See the [DSSD Hadoop Plugin Installation Guide](#) for more details.

CPU

Identify CPUs and NUMA Nodes

The DSSD client software includes a device driver for the PCIe ports used by the DSSD software.

Important:

Please make sure NUMA mode is enabled (NUMA mode is usually enabled by default) in the Worker nodes' BIOS.

For performance reasons, in multi-socket machines you must determine the nonuniform memory access (NUMA) node to which the device driver is attached. The DSSD Hadoop Plugin will load its instance of `libflood`, and this process must run on the same NUMA node as the device driver to avoid performance degradation.

Cloudera and DSSD recommend performing this step prior to setting up the cluster using Cloudera Manager.

Identify which CPU identifiers are associated with each NUMA node:

```
$ numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
node 0 size: 65439 MB
node 0 free: 62991 MB
node 1 cpus: 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
node 1 size: 65536 MB
node 1 free: 62486 MB
node distances:
node  0  1
  0:  10  21
  1:  21  10
```

The example output displays even-numbered CPUs on NUMA node 0 and odd-numbered CPUs on NUMA node 1.

Note:

Not all systems assign CPUs to NUMA nodes in this pattern.

Determine the NUMA Node Attached to the `vpci` Driver

Determine which NUMA node the `vpci` driver is attached to:

```
# grep . /sys/class/vpci_rp/vpci*/device/numa*  
/sys/class/vpci_rp/vpci6:0/device/numa_node:0  
/sys/class/vpci_rp/vpci7:0/device/numa_node:0
```

The example output shows that the `vpci` driver is attached to NUMA node 0.

Select a CPU Identifier to Assign to the DSSD DataNode

The final step is to select a CPU identifier to assign to the DSSD DataNode. The best practice is to select identifiers with high numbers because low CPU IDs tend to be used, for instance, by the Linux scheduler. In the previous example, CPU identifiers 28 or 30 are appropriate selections.

Note:

The Hadoop Plugin package for Cloudera and for Apache Hadoop provides a script called `detect_cpu_id` that automates the process of selecting a CPU core or cores to assign to the DataNode.

The following example of `detect_cpu_id` script selects two CPU IDs. The environment variable `$DSSD_DATANODE_PREFIX` is defined as the base pathname where the Cloudera Manager parcel for the DSSD DataNode is installed.

```
# $DSSD_DATANODE_PREFIX/bin/detect_cpu_id 2  
vpci driver attached to numa node 0  
numa node 0 has the following cpu ids: 0 2 4 6 8 10 12 14 16 18 20 22  
24 26 28 30  
Selected cpu-ids: [30,28]
```

You can then use the selected CPU IDs when setting up the cluster using Cloudera Manager. Or, if the cluster is already set up, you can use Cloudera Manager to update the values. In both cases, under HDFS Configuration, update the **Libflood CPU ID** configuration parameter, which is the Identifier of the CPU cores `libflood` will utilize.

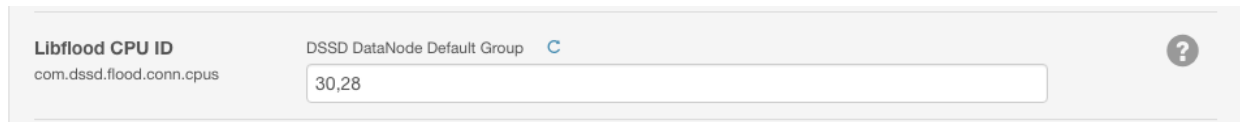
Important:

The default value for **Libflood CPU ID** is “all”, which indicates that the DSSD DataNode and `libflood` can utilize as many CPU cores as available on the server to perform I/O operations against the DSSD D5 appliance. Using this default value may have detrimental performance effects. DSSD recommends dedicating specific CPU IDs for the **Libflood CPU ID** configuration parameter.

See the [DSSD Hadoop Plugin Installation Guide](#) for more details.

The configuration string is: `com.dssd.flood.conn.cpus`

The following example shows the **Libflood CPU ID** configuration set to the two (comma-separated) CPU IDs obtained by the `detect_cpu_id` script:



The screenshot shows a configuration interface for 'Libflood CPU ID'. The configuration string 'com.dssd.flood.conn.cpus' is displayed. The value field contains '30,28'. The group is 'DSSD DataNode Default Group' with a blue 'C' icon. A help icon (?) is in the top right corner.

Short Circuit Reads (SCR)

Short Circuit Reads (SCR) allows the HDFS client to bypass the DataNode process to directly access data stored on the DSSD D5 appliance. For the same reason mentioned earlier, CPU IDs must be configured to prevent performance degradation.

HBase and Impala

SCR for HBase and Impala can be enabled on their corresponding configuration pages in Cloudera Manager. In addition, you should also update the **Libflood Short-Circuit Read CPU ID** configuration parameter, which specifies which CPU cores `libflood` utilizes to perform short-circuit reads.

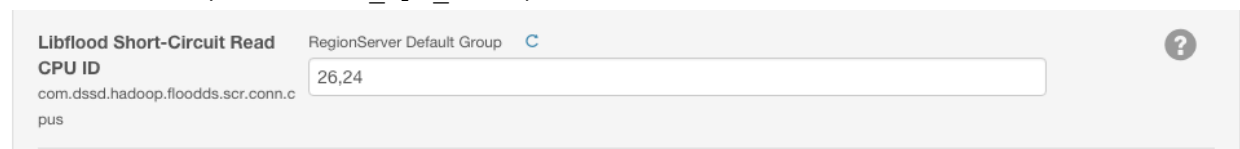
Important:

Like the **Libflood CPU ID** configuration parameter, the default value for **Libflood Short-Circuit Read CPU ID** is “all”, which indicates that the DSSD SCR plugin and `libflood` can utilize as many CPU cores as available on the server to perform I/O operations against the DSSD D5 appliance. Using this default value may have detrimental performance effects. DSSD recommends dedicating specific CPU IDs for the **Libflood Short-Circuit Read CPU ID** configuration parameter.

See the [DSSD Hadoop Plugin Installation Guide](#) for more details.

The configuration string is: `com.dssd.hadoop.floodds.scr.conn.cpus`

The following example shows the **Libflood Short-Circuit CPU ID** configuration set to the two (comma-separated) CPU IDs obtained by the `detect_cpu_id` script:



The screenshot shows a configuration interface for 'Libflood Short-Circuit Read CPU ID'. The configuration string 'com.dssd.hadoop.floodds.scr.conn.cpus' is displayed. The value field contains '26,24'. The group is 'RegionServer Default Group' with a blue 'C' icon. A help icon (?) is in the top right corner.

Important:

To avoid potential performance degradation, **Libflood Short-Circuit Read CPU ID** should use different CPU IDs than the ones used for **Libflood CPU ID**.

General Platform Tuning Recommendations

Cloudera Manager automates much of the platform tuning. This section highlights a few general recommendations.

Note:

These are general recommendations and should be applied only after sufficient testing.

CPU

CPU BIOS Settings

In your compute nodes BIOS, set CPU to Performance mode.

CPUfreq Governor

The following CPUfreq governor types are available in RHEL 6 & 7.

Governor Type	Description
cpufreq_performance	Forces the CPU to use the highest possible clock frequency. Intended for heavy workloads, this is best for interactive workloads.
cpufreq_powersave	Forces the CPU to stay at the lowest clock frequency possible.
cpufreq_ondemand	Allows CPU frequency to scale to maximum under heavy load, but drop down to the lowest frequency under light or no load. This is the ideal governor and, after appropriate testing, can be used to reduce power consumption under low load/idle conditions.
cpufreq_userspace	Allows userspace programs to set the frequency. This is used in conjunction with the cpuspeed daemon.
cpufreq_conservative	Similar to the cpufreq_ondemand, but switches frequencies more gradually.

Find the appropriate kernel modules available on the system, and then use `modprobe` to add the required driver:

```
# modprobe cpufreq_performance
```

After a governor is loaded into the kernel, enable it:

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

Available drivers are in the `/lib/modules/<kernelversion>/kernel/arch/<architecture>/kernel/cpu/cpufreq/` directory:

```
# cd /lib/modules/2.6.32-358.14.1.el6.centos.plus.x86_64/kernel/arch/x86/k
ernel/cpu/cpufreq
# ls
acpi-cpufreq.ko  mperf.ko  p4-clockmod.ko  pcc-cpufreq.ko
pownow-k8.ko    speedstep-lib.ko
```

If the required `cpufreq` drivers are not available, get them from `/lib/modules/<kernelversion>/kernel/drivers/cpufreq`:

```
# cd /lib/modules/2.6.32-358.14.1.el6.centos.plus.x86_64/kernel/drivers/cp
ufreq
# ls
cpufreq_conservative.ko  cpufreq_ondemand.ko  cpufreq_powersave.ko
```

```
cpufreq_stats.ko  freq_table.ko
```

Note:

Use the `uname -r` command to see the kernel version.

The `cpupower` utility is provided by the `cpupowerutils` package. If you have not installed it, you can set the tunables in `/sys/devices/system/cpu/<cpu id>/cpufreq/` by:

```
# echo performance > /sys/devices/system/cpu/<cpu id>/cpufreq/scaling_governor
```

Memory

Minimize Anonymous Page Faults

Minimize anonymous page faults, thereby freeing memory from page cache before “swapping” application pages.

To minimize anonymous page faults:

1. Edit `/etc/sysctl.conf` to add following line:

```
vm.swappiness=1
```

2. Run the following command:

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

Disable Transparent Hugepage Compaction and Defragmentation

Add the following commands to `/etc/rc.local` to ensure that transparent hugepage compaction and defragmentation remain disabled across reboots:

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

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

In RHEL 7.x the directories have changed to:

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

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

For RHEL 7.x, the above listed configurations can be managed using [tuned daemon](#).

Network

Add the following parameters to `/etc/sysctl.conf`:

- Disable TCP timestamps to improve CPU utilization (optional and depends 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 maximum 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

Determine which features are available with your NIC by using `ethtool`:

```
$ sudo ethtool -k
Features for eth0:
rx-checksumming: on
tx-checksumming: off
scatter-gather: off
tcp-segmentation-offload: off
udp-fragmentation-offload: off
generic-segmentation-offload: off
generic-receive-offload: on
large-receive-offload: off
rx-vlan-offload: on
tx-vlan-offload: on
ntuple-filters: off
receive-hashing: off
```

Modern NICs, particularly high-performance NICs, have various offload capabilities. Cloudera recommends enabling them.

In particular, tcp-segmentation-offload (TSO), scatter-gather (SG), and generic-segmentation-offload (GSO) should be enabled if not enabled by default:

```
$ sudo ethtool -K eth0 tso on sg on gso on
```

NIC Ring Buffer Configurations

Check existing ring buffer sizes:

```
$ ethtool -g eth0
Ring parameters for eth0:
Pre-set maximums:
RX:          4096
RX Mini: 0
RX Jumbo: 0
TX:          4096
Current hardware settings:
RX:          256
RX Mini: 0
RX Jumbo: 0
TX:          256
```

After checking the preset maximum values and the current hardware settings, use the following commands to resize the ring buffers:

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

Note:

The ring buffer sizes depend to some degree on network topology and might need to be tuned, depending on the nature of the workload. For 10 Gbps NICs, consider setting the RX and TX buffers to the preset maximums shown by `ethtool`.

Storage

Disk/FS Mount Options

Disable “atime” from the intermediate storage disks by using the `noatime` option when mounting the FS.

In the `/etc/fstab` file, ensure that the appropriate filesystems have the `noatime` mount option specified:

```
LABEL=ROOT /          ext4      noatime    0 0
```

FS Creation Options

FS creation for intermediate storage disks:

- Enable journal mode.
- Reduce superuser block reservation from 5% to 1% for root, using the `-m 1` option.
- Use the `sparse_super`, `dir_index`, and `extent` options to minimize the number of superblock backups and use B-tree indexes for directory trees and extent-based allocations.

```
#mkfs -t ext4 -m 1 -O
sparse_super,dir_index,extent,has_journal /dev/sdb1
```

Application Tuning Recommendations

All applications in CDH are supported in the DSSD D5-backed cluster. Generally, the default configuration provided by Cloudera Manager enables good application performance. Some applications, such as HBase, may require specific tuning to achieve best performance on the DSSD D5 appliance. This section highlights the tuning recommendations for these applications.

Note:

These are general recommendations and should only be applied after sufficient testing.

HBase

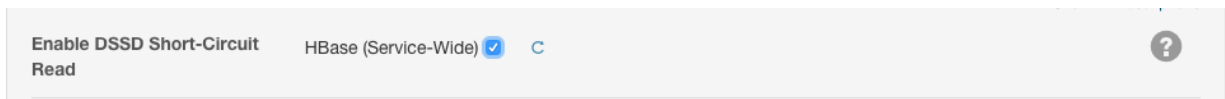
Cloudera Manager automatically configures the DSSD DataNode for general usage. For HBase, Cloudera and DSSD recommend the following tuning parameters.

HDFS Parameters

- **DataNode Handler Count:** The number of server threads for the DataNode.
Configuration string: `dfs.datanode.handler.count`
Suggested Value: 60
- **Java Heap Size of DataNode in Bytes:** Maximum size in bytes for the Java Process heap memory. Passed to Java `-Xmx`.
Suggested value: 2 GiB
- **Filesystem Trash Interval:** Number of minutes between trash checkpoints. Also controls the number of minutes after which a trash checkpoint directory is deleted. To disable the trash feature, enter 0.
Configuration string: `fs.trash.interval`
Suggested value: 0 (disabled)

HBase Parameters

- **Enable DSSD Short-Circuit Read:** This allows HDFS client roles of this service that are colocated with DSSD DataNodes to read DSSD volumes directly, instead of indirectly through the DSSD DataNode.



- **HBase RegionServer Handler Count:** Number of RPC server instances spun up on RegionServers.
Configuration string: `hbase.regionserver.handler.count`
Suggested value: 120
- **HStore Blocking Store Files:** If the number of HStoreFiles in any one HStore exceeds this number, updates are blocked for this HRegion until a compaction is completed, or until the value specified for `hbase.hstore.blockingWaitTime` has been exceeded.
Configuration string: `hbase.hstore.blockingStoreFiles`
Suggested value: 50
- **RegionServer Small Compactions Thread Count:** Number of threads for completing small compactions.
Configuration string: `hbase.regionserver.thread.compaction.small`
Suggested value: 1

- **HBase Memstore Block Multiplier:** Blocks writes if the size of the memstore increases to the value of `hbase.hregion.block.memstore` multiplied by the value of `hbase.hregion.flush.size` bytes. This setting prevents runaway memstore during spikes in update traffic. Without an upper bound, memstore fills such that when it flushes, it takes a long time to compact or split, or an "out of memory" error occurs.
Configuration string: `hbase.hregion.memstore.block.multiplier`
Suggested value: 4
- **Per-RegionServer Number of WAL Pipelines:**
Configuration string: `hbase.wal.regiongrouping.numgroups`
Suggested value: 10
- **WAL Provider:** The implementation used by the RegionServer for the write-ahead log.
Configuration string: `hbase.wal.provider`
Suggested value: Multiple HDFS WAL

References

1. [Cloudera Documentation \(Cloudera Enterprise 5.8.x\)](#)
2. [EMC DSSD Documentation](#)