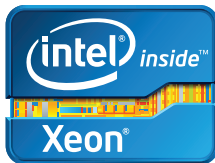


# Intel® Distribution for Apache Hadoop\* Software



## 1.0 Executive Summary

This document presents the reference architecture for Intel® Distribution for Apache Hadoop\* software on commodity cluster hardware using the Intel® Xeon® family of CPU-based motherboards and systems. The intended audiences for this document are customers and system architects looking for information on implementing Apache Hadoop clusters within their information technology environment for big data analytics.

First, the reference architecture introduces the high-level components that are included in the Intel® Distribution for Apache Hadoop\* software stack. Next, we discuss typical Apache Hadoop use cases and introduce the Intel Hadoop taxonomy and then talk about in detail, extensions and enhancements to the standard Apache Hadoop distribution that can be found only in the Intel® Distribution. Finally, the reference architecture describes the optimal configuration to deploy for a given Apache Hadoop solution with tips on benchmarking and performance tuning.

## Table of Contents

<b>1.0 Executive Summary</b> .....	1
<b>2.0 Intel® Distribution for Apache Hadoop* Software Overview</b> .....	2
2.1 Apache Hadoop.....	2
2.2 Intel Extensions.....	2
<b>3.0 Intel Hadoop Use Case Summary</b> ...3	
<b>4.0 Intel Hadoop Solution Taxonomy</b> ...3	
<b>5.0 Intel Hadoop Software Components</b> .....	4
5.1 Intel Manager .....	4
5.1.1 Enterprise vs Community Edition.....	4
5.1.2 Installation & Configuration .....	5
5.1.3 Health & Resource Monitoring.....	5
5.1.4 Hadoop Administration .....	6
5.2 Intel Hadoop Security Design .....	6
5.2.1 Implementing Secure Hadoop..	6
5.3 Real Time Transaction Support .....	7
5.4 Dynamic Replication Support.....	8
<b>6.0 Intel Hadoop Hardware Components</b> .....	8
6.1 Master Nodes.....	8
6.2 Data/Slave Nodes .....	8
6.3 Network Fabric .....	9
6.3.1 Access Switch or Top of Rack (ToR).....	9
6.3.2 Aggregation Switches.....	9
<b>7.0 Cluster Hardware Architectures</b> .....	9
7.1 Rack.....	9
7.2 Pod.....	9
7.3 Cluster .....	9
<b>8.0 Planning Considerations</b> .....	10
8.1 Data Sizing requirements .....	10
8.2 Bandwidth and Performance Requirements.....	10
<b>9.0 Performance, Tuning &amp; Benchmarking</b> .....	10
9.1 Optimization & Tuning .....	11
9.2 Configuring and Optimizing the Software Layer .....	11
9.3 Configuring and Optimizing the Hardware Layer .....	12
9.4 Benchmarking.....	13
<b>10.0 Conclusions</b> .....	13
<b>11.0 References &amp; Contacts</b> .....	14
<b>12.0 Abbreviations</b> .....	14

## 2.0 Intel® Distribution for Apache Hadoop\* software Overview

In this section we provide a quick introduction to Apache Hadoop and then talk about enhancements and additions that can only be found in the Intel® Distribution for Apache Hadoop.

### 2.1 Apache Hadoop

Apache Hadoop is an Apache project being built and used by a global community of contributors, using the Java\* programming language. Yahoo!, has been the largest contributor to this project, and uses Apache Hadoop extensively across its businesses. Other contributors and users include Facebook, LinkedIn, eHarmony, and eBay.

The core Apache Hadoop platform includes a distributed file system called Hadoop Distributed File System (HDFS), and a framework called Map/Reduce to execute jobs in parallel. Additionally, Apache Hadoop also includes Apache HBase\*, a No SQL like columnar data storage capability and Apache Hive\*, a query processing engine with a SQL-like syntax.

Intel has created a quality-controlled distribution of Apache Hadoop, referred to as Intel® Distribution for Apache Hadoop\* software in the rest of this document, based on the Apache Hadoop source base with feature enhancements and performance improvements that offers enterprise quality management software, deployment support, and consulting services.

### 2.2 Intel extensions

Intel, in partnership with other platform vendors, has developed a solution for big data that includes a feature enhanced controlled distribution of Apache Hadoop, with optimizations for better hardware performance, and services to streamline deployment and improve the end user experience.

The Intel® Distribution for Apache Hadoop\* software includes:

- The Intel® Manager for Apache Hadoop\* software to install, configure, monitor, and administer the Apache Hadoop cluster
- Enhancements to HBase and Hive for improved real-time query performance and end user experience

- Resource monitoring capability using Nagios\* and Ganglia\* in the Intel® Manager
- Superior security and performance through better integrated encryption and compression
- Packaged Apache Hadoop ecosystem that includes HBase, Hive, and Apache Pig\*, among other tools

This solution provides a foundational platform for Intel to offer additional solutions as the Apache Hadoop ecosystem evolves and expands.

Aside from the Apache Hadoop core technology (HDFS, MapReduce\*, etc.) Intel has designed additional capabilities to address specific customer needs for big data applications such as:

- Optimally installing and configuring the Apache Hadoop cluster
- Monitoring, reporting, and alerting of the hardware and software components
- Providing job-level metrics for analyzing specific workloads deployed in the cluster
- Infrastructure configuration automation
- Extensions to HBase and Hive to improve real-time transactional performance and features
- Enhancements to security and access control with better encryption and decryption capabilities

Intel® Distribution for Apache Hadoop\* software focuses on efficient integration of open source-based Apache Hadoop software distribution with commodity servers to deliver optimal solutions for a variety of use cases while minimizing total cost of ownership.

The supported operating environments for the Intel® Distribution for Apache Hadoop\* software are Red Hat Enterprise Linux\*, CentOS\*, and Oracle Linux\*. The recommended Java virtual machine (JVM) is the Oracle/Sun JVM. Refer to Table 6 below for more details. The hardware platforms, the operating system, and the Java virtual machine make up the foundation on which the Apache Hadoop software stack runs.

### 3.0 Intel Hadoop Use Case Summary

The table below outlines important use cases that a typical Intel Hadoop cluster can be used for.

Table 1. Intel Hadoop Solution Use Cases	
Use case	Description
Big data analytics	Ability to query in real time at the speed of thought on petabyte scale unstructured and semi structured data using HBase and Hive.
Data storage	Collect and store unstructured and semi-structured data in a secure, fault-resilient scalable data store that can be organized and sorted for indexing and analysis.
Batch processing of unstructured data	Ability to batch-process (index, analyze, etc.) tens to hundreds of petabytes of unstructured and semi-structured data.
Data archive	Medium-term (12-36 months) archival of data from EDW/DBMS to increase the length that data is retained or to meet data retention policies/compliance.
Integration with data warehouse	Extract, transfer and load data in and out of Hadoop into separate DBMS for advanced analytics.
Big data visualization	Capture, index and visualize unstructured and semi structured big data in real time
Search and predictive analytics	Crawl, extract, index and transform semi structured and unstructured data for search and predictive analytics

### 4.0 Intel Hadoop Solution Taxonomy

In Figure 1, the dark blue layer in the Intel Hadoop taxonomy is comprised of:

- The **Intel® Manager** for Apache Hadoop\* software, which is a web-based management console designed to install, configure, manage, monitor and administer the Intel Hadoop cluster. It uses Nagios and Ganglia to monitor resources and configure alerts in the cluster.

- The **Data Storage Framework (HDFS)** is the file system that Apache Hadoop uses to store data on the cluster nodes. HDFS is a distributed, scalable, and portable file system. Intel Hadoop includes compression and encryption for enhanced security and performance.

- The **Data Processing Framework (MapReduce)** is a massively-parallel compute framework inspired by Google's MapReduce papers. Intel Hadoop includes dynamic replication capabilities that wax and wane the number of replicas depending on workload characteristics.
- The **Real Time Query Processing Framework**, which includes **HBase**, a scalable distributed columnar data storage system for large tables, and **Hive** data warehouse infrastructure for ad-hoc query processing. Intel Hadoop includes extensions to support big tables across geographically distributed data centers, as well as feature additions to improve Hbase and Hive performance.

The components that constitute the Intel Hadoop solution taxonomy are described below:

- **HBase** is a columnar database management framework that uses the underlying HDFS framework to provide random and real time update to data. It has been designed and developed to provide the capability to host very large tables that can support billions of rows with millions of columns.
- **Hive** is the query engine framework for Hadoop that facilitates easy data summarization, ad-hoc queries, and the analysis of large datasets stored in HDFS and HBase.
- **Apache ZooKeeper\*** is a high-performance coordination service for distributed applications. It is used as a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.
- **Apache Sqoop\*** is a tool designed to efficiently transfer bulk data between Apache Hadoop and structured data stores such as relational databases. It can be used to import data from external data stores into Hadoop distributed files system or related systems like Hive and HBase. Conversely, Sqoop can be used to run map/reduce jobs that extract data from Apache Hadoop and export to external structured data stores and enterprise data warehouses.

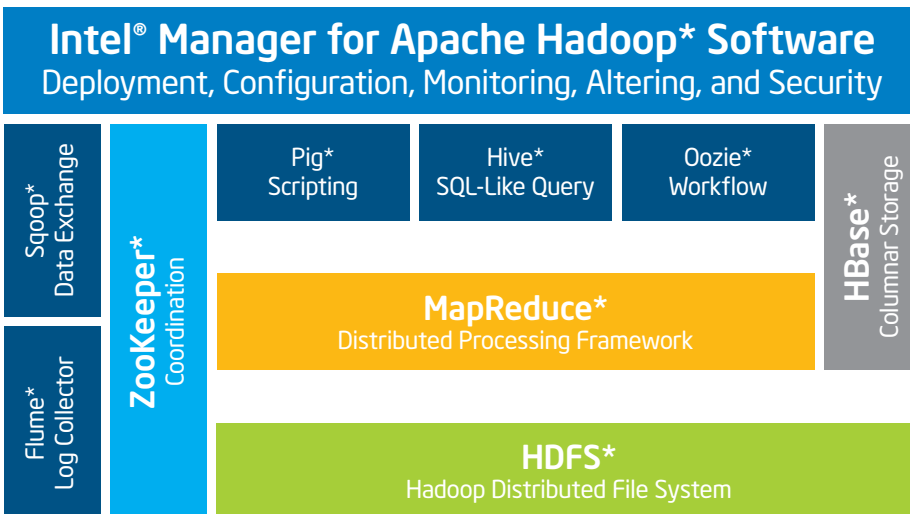


Figure 1. Intel Hadoop solution taxonomy

▪ **Apache Flume\*** is a distributed, reliable, and available system for efficiently collecting, aggregating, and moving large amounts of log data from many different sources to a centralized data store.

▪ **Pig** is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

▪ **Apache Mahout\*** is a scalable machine learning library that is intended for data mining on reasonably large data sets. The core algorithms for clustering, classification and batch based collaborative filtering are implemented on top of Hadoop using the Map/Reduce paradigm. Even though the core algorithms are written for Hadoop clusters, the libraries also run well on non-distributed (non Hadoop) single node systems as well.

▪ **Apache Oozie\*** is a scalable, reliable, and extensible system that includes a workflow scheduler system to manage Hadoop jobs. Oozie Workflow jobs are Directed Acyclical Graphs (DAGs) of actions. Oozie Coordinator jobs are recurrent Oozie Workflow jobs triggered by time (frequency) and data availability.

In addition to the above, Apache Hadoop has gained quick acceptance in the field of statistical analysis. R is a programming language with primitives and libraries for statistical analysis and visualization. R can be used with Hadoop streaming utility to directly run as mapper and reducer jobs or RMR the language that generates Map/Reduce code that can run on Hadoop can be used for statistical analysis.

## 5.0 Intel® Distribution for Apache Hadoop\* Software Enhancements

In this section we introduce in greater details the software components and extensions to Intel® Distribution for Apache Hadoop\* software. The components discussed in the sections below are enhancements to the Apache community distribution that will only be found in the Intel distribution.

### 5.1 Intel® Manager

The Intel® Manager for Apache Hadoop\* software streamlines configuration, management, security, administration, and resource monitoring of Apache Hadoop clusters. With this powerful, easy-to-use web based tool, IT can focus critical resources and expertise on driving business value from the Hadoop environment rather than worrying about the details of cluster management. Intel® Manager provides:

- Installation and configuration for Hadoop clusters
- Wizard-driven cluster management
- Proactive cluster health checks
- Monitoring and logging
- Secure authentication and authorization

An enterprise edition of the Intel® Manager is available in addition to the community standard edition.

Intel® Manager Enterprise is a subscription service that comprises of Intel support and a portfolio of software, including Intel® Management Services, which enables the end user to run the Apache Hadoop environments in a production environment, cost-effectively and with repeatable success.

By combining expert support with software components that deliver deep visibility into and across Apache Hadoop clusters, Intel® Manager Enterprise allows the user an efficient way to precisely provision and manage cluster resources.

It also allows an IT shop to apply familiar business metrics, such as measurable SLAs and chargebacks to their Apache Hadoop environment so it can be utilized optimally in the organization.

#### 5.1.1 Enterprise vs. Community Edition

The following table illustrates the difference between the Enterprise and Community editions of the Intel Manager.

**Table 2. Differences between Intel® Manager Community Edition and Intel® Manager Enterprise Edition**

Hadoop Manager Editions	Intel Hadoop Manager 2.0 Community Edition	Intel Hadoop Manager 2.0 Enterprise Edition
Maximum Number of Nodes Supported	Unlimited	Unlimited
Automated Installation and Deployment	√	√
Host-Level Monitoring	√	√
Secure Communication Between Server and Agents	√	√
Service Management	√	√
Manage HDFS, MapReduce, HBase, Hue, Oozie and Zookeeper	√	√
Automated Configuration	√	√
Audit Trail	√	√
Start/Stop/Restart Services	√	√
Add/Restart/Decommission Role Instances	√	√
Configuration Versioning and History	Not available in 2.0	√
Support for Kerberos	√	√
Service Monitoring	√	√
Proactive Health Checks	√	√
Status and Health Summary	√	√
Performance Monitoring	√	√
Intelligent Log Management	√	√
Events Management and Alerts	√	√
Activity Monitoring	√	√
Operational Reporting	Not available for Map/Reduce	√
Global Time Control	√	√
Support Integration	√	√

### 5.1.2 Installation and Configuration

Intel® Manager automatically installs, configures, and optimizes nodes across the Apache Hadoop framework and provides a web console to make configuration changes as needed.

Intel® Manager locates server nodes within a specific network and installs Apache Hadoop software components on selected server nodes. Using a wizard-based interface, the Hadoop administrator can deploy the Hadoop framework across all nodes, assign roles to the nodes, and optimize configuration settings for each role.

Installation using the Intel® Manager includes the following:

- Scanning server nodes on the specified network and installing Hadoop package components
- Defining a flexible network topology, rack settings, and an automatic replica placement scheme
- Assigning roles to nodes in the cluster
- Intelligently configuring Hadoop cluster nodes for optimal performance, using the node hardware configurations
- Adding and removing nodes from the live cluster as needed

Intel® Manager streamlines the configuration and distribution of changes throughout the cluster. Cluster configuration includes the following functionality:

- Providing easy configuration editing for an individual server node or the whole cluster with a user-friendly web interface
- Pushing updates to all nodes in the cluster
- Reducing the potential for configuration update failures with static checks for parameter interdependencies and other configuration errors
- Providing configuration recommendations for optimized performance

### 5.1.3 Health and Resource Monitoring

The Intel® Manager includes resource monitoring capability to automate the configuration of alerts and responses to unexpected events and failures within the Intel Hadoop cluster. The health and resource monitoring provides capabilities for three primary components of the cluster environment:

#### ▪ **Monitoring of cluster activities:**

The resource monitor component monitors the cluster, including hardware components, software components and Hadoop framework. The resource monitor will keep historical information regarding system usage, system availability, maintenance, and failure of events. The configurable dashboard tracks key processing, memory, network, and storage utilization metrics, including:

- A default view of the most important health aspects of a cluster: CPU, memory, network, storage JVM memory, logs, jobs, and more
  - Additional drill-down templates for deep insight into the health of all or selected parts of Hadoop cluster components, such as data nodes, Hadoop Distributed File System (HDFS\*), MapReduce jobs, and JVM
  - A grid of interactive graphs displaying historical data maintained by the Ganglia RRDTOOL storage database management subsystem
  - An overall system map showing the physical topology of one or more data centers organized as a grid containing one or more clusters
- **Alerts and events management:** The monitoring component is designed to be proactive in nature. The resource monitor component of Intel® Manager will alert system operations staff about occurrence of events that deviate from normal operations, if the administrator has designated them for notification. Standard

automated responses could include execution of custom scripts to send e-mail notification and/or take other corrective actions before the failure causes an outage that affects product workloads and users. Intel® Manager monitors a wide range of cluster events, including:

- High CPU usage
- Memory usage and swapping
- Network usage
- HDFS capacity
- HBase compaction storm
- Disk capacity
- Disk I/O utilization
- Frequent JVM garbage collection (GC) MapReduce job failure statistics

System administrators can define high-level events by combining multiple metrics and then trigger alerts with specified thresholds. Alerts are delivered by e-mail based on configuration and settings.

#### ▪ **Debugging of cluster runtime**

**operation:** The resource monitor provides the users and administrators of the Apache Hadoop environment with the necessary tools for debugging log files, tracking jobs and tasks, and monitoring and analyzing job performance characteristics. Intel® Manager provides a centralized view of logs on every node in the cluster. Administrators can:

- Easily read and search logs. To avoid creating overly large files, logs are accumulated and then split and rotated.
- Monitor master nodes via logged alerts for fatal events, errors, or warnings.
- Monitor the integrity of HDFS and HBase tables with reported errors.

The resource monitor is flexible enough to allow integration with existing operations management frameworks in the customer IT environment.

### 5.1.4 Hadoop Administration

The Intel® Manager includes an administration component that provides the following features:

- **Simple User Authentication** allows an administrator to create users and provide role based access to the Intel® Manager console to view and administer the Hadoop cluster.
- **LDAP and Active Directory** integration allows the Intel® Manager access to be integrated with an existing authentication infrastructure.

Intel Hadoop also provides fine grained encryption and decryption support, which is described under the security design component.

### 5.2 Intel Hadoop Security Design

Intel® Manager supports secure authentication and authorization using Kerberos and built-in access control rules, including:

- Authentication between users and the Apache Hadoop cluster to safeguard against malicious user impersonation.
- Authentication between Apache Hadoop cluster nodes to prevent group membership manipulation by users.
- Permission controls for specific HDFS files or directories.
- Authentication of users for Hive\* metastore access.
- Authentication between HBase and the secure HDFS

The security encryption is capable of integration with an existing Kerberos server installation. Sqoop and Pig support security with no additional configuration being required.

#### 5.2.1 Implementing Secure Hadoop

As outlined in the above section, Intel® Distribution for Apache Hadoop\* software provides the ability to deploy a Secure Hadoop solution in more than one level. Some of the features highlighted are:

- **HDFS Encryption/Decryption and Key Management:** Intel® Distribution provides a mechanism to encrypt and decrypt files stored in HDFS file system. The HDFS encryption feature prevents data from being leaked if an unauthorized user has access to the disk. Similarly, the files are protected as they are being accessed for reads/writes/appends.
- **Fine Grained Access Control:** Intel® Distribution provides fine grained access control features using Kerberos and Active Directory or LDAP. Using Intel Manager, the administrator of the Hadoop cluster will be able to better define the roles each user has and what privileges they have for accessing data that is on the cluster, or executing jobs on the cluster.

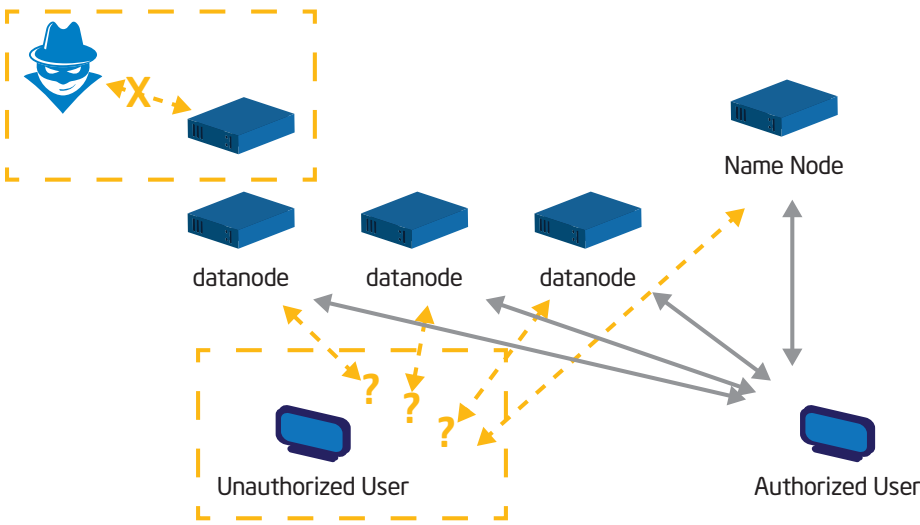


Figure 2. Encryption and decryption for HDFS

In the figure below, three possible scenarios are shown when a user tries to access a file that is encrypted.

1. When a user with a correct key tries to read/write a file the client access APIs will resolve the key privileges to access the file and passes the privilege to the MapReduce framework to provide access to the JVM processes to process the file.
2. In the second scenario, when a user with an incorrect key tries to access a file, the client access APIs prevent his process from proceeding further with the map and reduce jobs.
3. When unencrypted text or files are used then everything works smoothly.

### 5.3 Real-Time Transaction Support

As outlined in the Hadoop taxonomy section, HBase and Hive together provide the framework to develop transactional applications on the Hadoop platform by providing random and real time update capability.

Intel® Distribution includes a more robust and feature enhanced version of Apache HBase and Hive that is targeted for distributed transactional support. Specifically, the following are some of the specific Intel® Distribution feature enhancements:

- **Cross site big table support** provides the ability to create a single cross-site HBase table that spans multiple HBase clusters that are distributed geographically across many data centers. Even though such tables are physically stored across different clusters, they can be viewed as a single HBase table from a client application.
- **Customized table replication** allows the administrator to choose replication level by column for a given table. If certain columns in a table are expected to be accessed more frequently than others, those columns can be set to have a higher replication level.

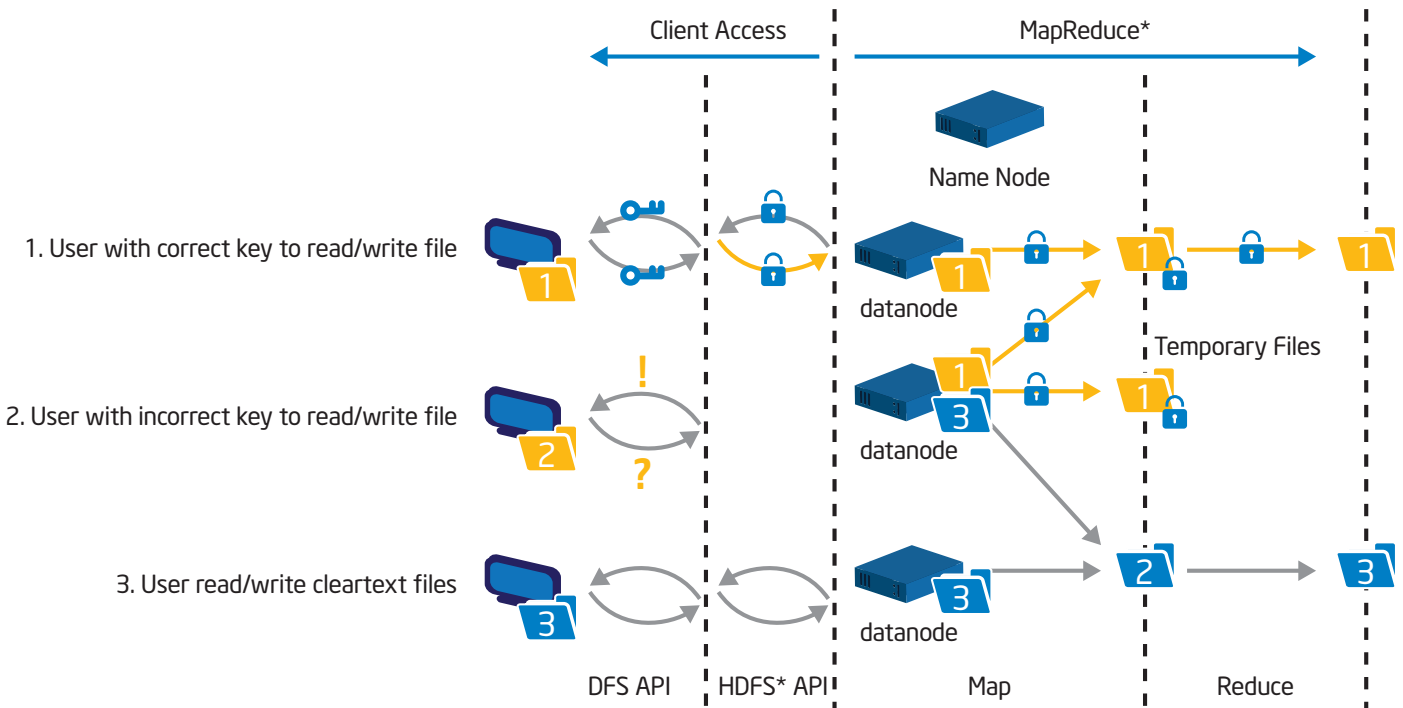


Figure 3. Fine grained access control

### 5.4 Dynamic Replication Support

The default file level replication support available in the HDFS framework is statically set at the time of file creation.

Intel® Distribution supports dynamic replication capability in the framework. When dynamic replication is turned on via Intel Manager, the cluster framework will increase the replication level of that file whenever it gets "hot." When the file is no longer "hot," the framework will reduce the replication level to default level.

A number of configurable and tunable parameters are provided to customize this feature via the Intel® Manager for a given installation.

### 6.0 Intel Hadoop Hardware Components

The recommended configuration for Intel Hadoop cluster hardware contains the following nodes:

- **Master Node**—One or more physical nodes (servers) that run the Hadoop Name node, Jobtracker and Secondary Name node components.
- **Slave Node**—Three or more nodes (servers) that run all the services required to store blocks of data on the local hard drives and execute processing tasks against that data.

- **Edge (Gateway) Node**—Provides the interface between the data and processing capacity available in the Hadoop cluster and client software that connect and use the services of Apache Hadoop.
- **Manager Node**—Runs the Intel® Manager and is used to administer the nodes of the Hadoop cluster.

#### 6.1 Master Nodes

Depending on the size of the Hadoop cluster, a master node potentially can run one or more of the three master node daemons, which are namenode, jobtracker and secondary namenode. The master node does not store any data and uses a lot of physical memory as it maintains critical data in memory for the functioning of Hadoop cluster.

**Table 3. A recommended configuration of Master node**

<b>CPU</b>	Two CPU sockets with six or eight cores Intel® Xeon® processor E5-2600 series @ 2.9 Ghz
<b>Memory</b>	48 GB ( 6X8 GB 1.35v 1333 MHz DIMMs) or 96 GB (6x16 GB 1.35v 1333 MHz DIMMs)
<b>Disk</b>	4 x 1 TB SATA drives in RAID 5 configuration
<b>Network</b>	1x dual-port 10 GbE NIC

We recommend a two-socket CPU with six to eight cores. For smaller clusters with six to twelve data nodes, we recommend 48 GB, but for larger clusters 96 GB memory is recommended. We recommend RAID drives for master nodes to eliminate single points of failures. For network one 10 Gb network interface card with dual port is recommended.

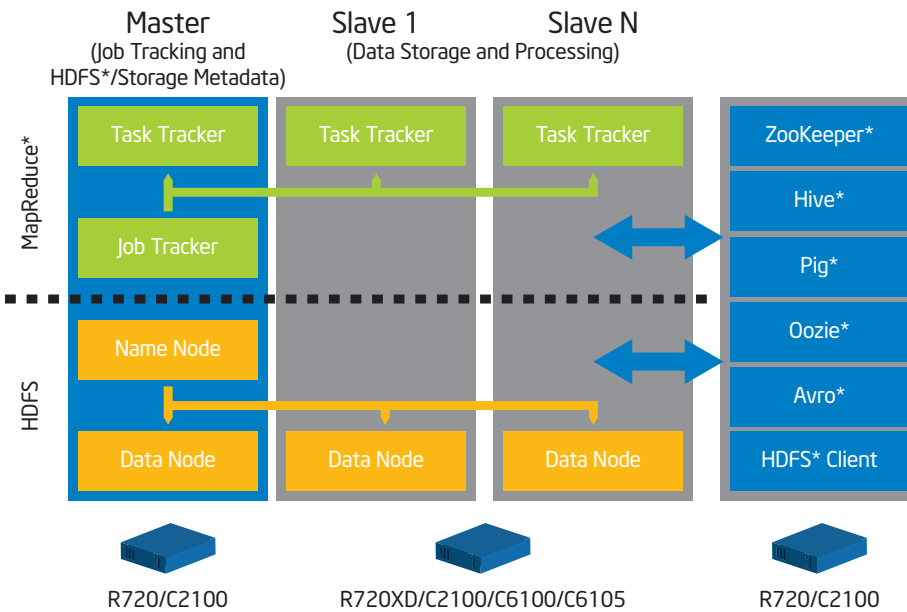
#### 6.2 Data/Slave Nodes

The data nodes in a Hadoop cluster store the data blocks and process information. Data nodes are rack mountable servers.

Typical commodity rack servers come in 1U or 2U rack heights. They are designed to deliver the most competitive feature set, best performance and best value. Most offer a large storage footprint, best-in-class I/O capabilities, and advanced management features. Some of the more recent classes of commodity server backplanes can accommodate up to 24 drives.

**Table 4: A recommended configuration of data node**

<b>CPU</b>	Two CPU sockets with six or eight cores Intel® Xeon® processor E5-2600 series @ 2.9 Ghz
<b>Memory</b>	48 GB ( 6X8 GB 1.35v 1333 MHz DIMMs) or 96 GB (6x16 GB 1.35v 1333 MHz DIMMs)
<b>Disk</b>	10-12, 1-3 TB SATA drives
<b>Network</b>	1x dual port 10 GbE NIC or 1x quad port 1 GbE NIC



**Figure 4.** Hardware components of Intel Hadoop cluster

We recommend two socket six or eight core Intel® Xeon® CPUs. For memory, we recommend 96 GB for higher end clusters and 48 GB for smaller clusters. We recommend 10-12 SATA disks without any RAID configuration for optimal performance. For network, we recommend gigabit ethernet cards for lower end clusters and ten gigabit ethernet cards for higher end clusters. We also recommend bonding of interfaces at the host end and the switch end for greater bandwidth.



### 6.3 Network Fabric

We recommend the usage of best-in-class gigabit (10 Gigabit) Ethernet switches as the top-of-rack connectivity to all Hadoop-related nodes. This reference architecture is used to support consistency in rapid deployments through the minimal differences in the network configuration.

From a reference architecture perspective, we suggest that at a minimum three distinct, separate VLANs be implemented for the network fabric:

- **Apache Hadoop Cluster Data LAN**— Connects the compute node NICs into the fabric used for sharing data and distributing work tasks among compute nodes.
- **Apache Hadoop Cluster Management LAN**—Connects all the iDRAC/BMCs in the cluster nodes.
- **Apache Hadoop Cluster Edge LAN**— Connects the cluster to the outside world.

#### 6.3.1 Access Switch or Top of Rack (ToR):

The servers connect to ToR switches. Typically there are two in each rack. The two ToR switches stack together in the same rack. This is useful in managing the two switches as a single unit and allowing the servers to connect into two different switches for redundancy. The ToR switches each have two expansion slots that can accept a two-port 10G module or a two-port stacking module. This architecture recommends one of each type in the two slots. The 10GbE module would be used to connect into the pod-interconnect switches, one port to each switch, forming a LAG. The stacking module would stack the switches together.

In multi-rack configurations, each rack is managed as a separate entity and ToR switches connect only to the pod-interconnect. The stacking ports are both connected to the switch in the same rack, while both 10GbE interfaces connect

to the pod-interconnect switches. In this situation the stacking bandwidth is doubled from stacking all six (6) switches across three racks together option and the failure domain is limited to a single rack rather than all three racks. The uplinks to the pod-interconnects would be a single LAG of four 10GbE ports, two from each switch. Each rack connects to the pod-interconnect independently, thereby scaling is easier.

#### 6.3.2 Aggregation Switches

The aggregation switches potentially scale Apache Hadoop deployments into hundreds of nodes in multi rack configurations. Apache Hadoop ToR switches connect to aggregate switches. The uplink would be on 10GbE interfaces from the ToR. The recommended architecture uses Virtual Link Trunking (VLT)\* between the two aggregation switches.

The stacks in each rack would divide their links between this pair of switches to achieve the powerful capability of active-active forwarding while using full bandwidth capability, in absence of any requirement for spanning tree. The aggregation switches also run layer-3 from the ToR as layer-2 alone is not an Apache Hadoop requirement. Therefore, for scaling to large deployments, layer-3 routing is a good option.

### 7.0 Cluster Hardware Architectures

Typical Hadoop clusters can be classified into three broad classes to accommodate for sizing as the Hadoop cluster grows. From smallest to largest, they are rack, pod, and cluster. Each has specific characteristics and sizing considerations documented in this reference architecture. The design goal for the Apache Hadoop environment is to enable the customer, to scale the environment by adding the additional capacity as needed, without the need to replace any existing components.

### 7.1 Rack

A rack is the smallest size designation for a Hadoop environment. A rack consists of all the necessary power, the network cabling, and the two Ethernet switches necessary to support 16 to 20 data nodes. These nodes should utilize their own power connectivity and space within the data center, separate from other racks, and be treated as a fault zone.

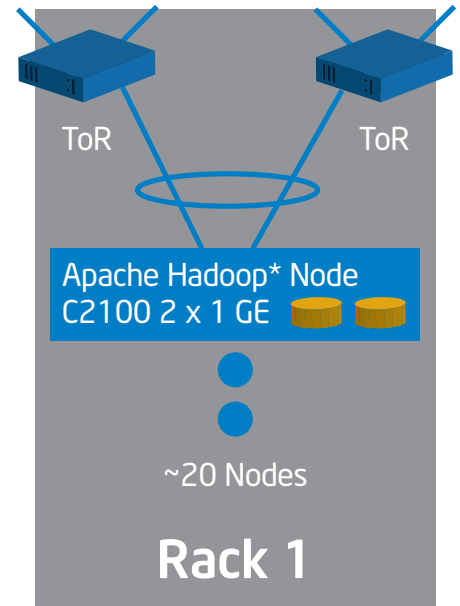


Figure 5. A single Intel Apache Hadoop cluster rack

#### 7.2 Pod

A pod consists of the administration and operation infrastructure to support three racks. Each rack will contain its own top of rack switches and the top of rack switches from each rack is connected with a higher bandwidth network interconnect. A pod can consist of up to 20 data nodes.

#### 7.3 Cluster

A cluster is a set of two to twelve pods. It is a set of Hadoop nodes that share the same Name Node and management tools for operating the Hadoop environment.

## 8.0 Planning Considerations

Planning and sizing for the right Apache Hadoop cluster depends on the following factors:

- **Storage capacity:** Amount of total storage, or the growth per year in the amount of data that will be stored, will determine the cluster size to start with. The amount of disk storage per node is limited by the number and size of disks that can be included in a node. Block replication factor, will multiply the amount of total storage required, by the replication factor level, as a 90 TB total raw capacity cluster can in reality only store 30 TB or less as Apache Hadoop, by default uses a replication level of three.
- **Workload resource consumption:** CPU and memory utilization of the workloads that will run on the cluster, will also determine the size and capacity of the cluster. Compute intensive workloads

may require appropriate number of CPU cores free (not doing disk and network I/O) with the right amount of free physical memory. If the cluster usage is anticipated to be as a storage cluster, then the stress on CPU and memory will be low. If large amounts of data is expected to be transferred between nodes during reduce phase, network fabric can become a bottleneck.

- **Transactional type:** The software stack chosen during installation and configuration will depend on the transactional type of the application workload. Real time applications that require ad-hoc query and update capabilities will require HBase and/or Hive over core Hadoop infrastructure. Data mining and analytics capabilities will require other components of the ecosystem to be configured and made available.

## 8.1 Data sizing requirements

Sizing existing data that needs to be stored and estimating the rate of influx of new data that needs to be captured and persisted on an ongoing basis is quite important. In addition to sizing the data from an application domain perspective, Apache Hadoop replication of data should also be factored into the sizing calculation.

Additionally, if dynamic replication feature of Intel® Distribution is enabled or if the customized table replication feature of HBase in Intel® Distribution is enabled, additional data requirements for these features should also be accounted for in the data sizing exercise.

## 8.2 Bandwidth and Performance Requirements

Real time applications with millisecond response time requirements and continuous analytics workloads that run on large data sets on Apache Hadoop will require high bandwidth between the nodes of the cluster.

Also, a cost over performance benefit evaluation between using a gigabit versus 10 Gb Ethernet for the network fabric needs to be performed to choose the right network fabric.

Data ingestion and extraction, in and out of a Hadoop cluster will be critical to most application. Networking interface bonding can provide much needed bandwidth relieve.

## 9.0 Performance Considerations

The Apache Hadoop system is composed of a number of components that are integrated in layers. The performance of any Apache Hadoop system is based on optimal tuning of each layer.

Because there are so many variables within several interdependent parts in each layer, tuning and optimizing a Hadoop system can be a challenge and may take some time. Also, each layer may be controlled by different IT teams, including developers and infrastructure teams, making the coordination of tuning efforts imperative.

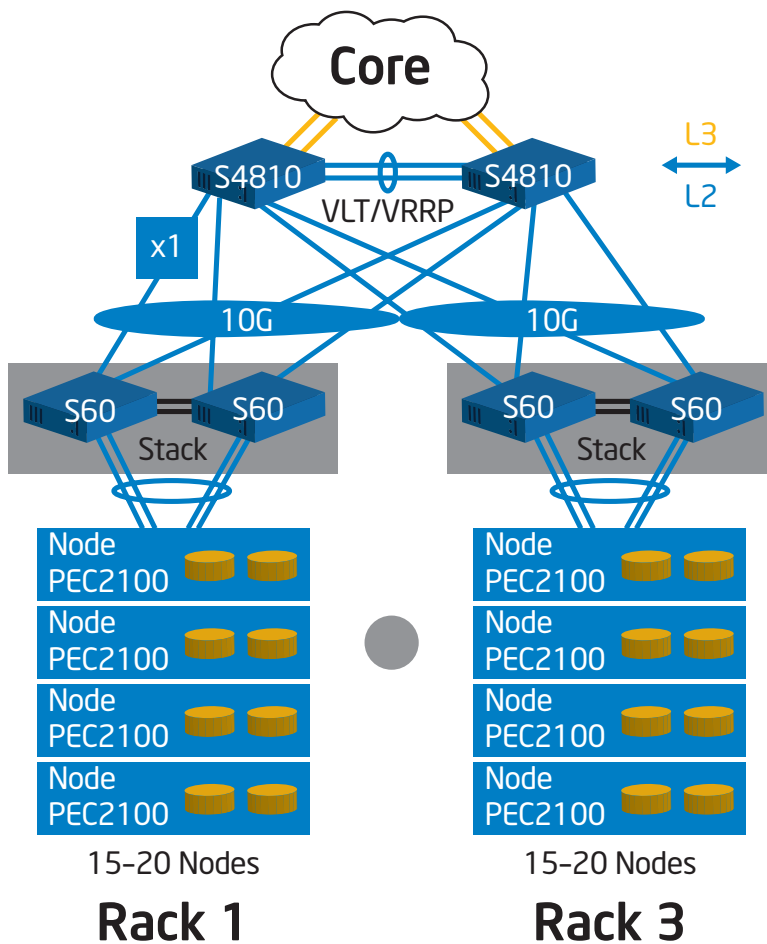


Figure 6. A multi-rack Intel Apache Hadoop cluster

Looking top down, the first tunable layer is the application layer. Any user applications developed for the Apache Hadoop framework will have variables that can be set to provide the best performance for that application.

The next layer is the actual Hadoop framework, which includes its two main components, MapReduce and the HDFS. Settings can be customized using the Intel® Manager.

The third layer is the software layer, which includes the JVM as well as the C and C++ programs, where parameters can be set accordingly.

The fourth layer is the operating system, and the fifth layer is the hardware resources. Selection of hardware, such as CPU, memory, type of network interface card (NIC), and number and type of hard drives can greatly affect the performance of the Apache Hadoop system.

## 9.1 Optimization and Tuning

This section is relevant to application users and developers. The Apache Hadoop framework solves the big data problem by managing tens to hundreds of petabytes of data. Processing large amounts of data involves reading and writing activities within the Hadoop system. These activities are very resource intensive, so it is imperative to finely tune these activities as much as possible for best performance.

**Reduce disk and network I/O:** Disk and network I/O activities can be reduced by tuning the memory buffer threshold and by using compression, which reduces the bandwidth and CPU needed for processing.

### MapReduce—Memory Buffer

- *io.sort.factor*—This represents the number of input stream files to be merged at once during map or reduce tasks. The value should be sufficiently large (e.g., 100) rather than the default value of 10.
- *io.sort.mb*—This is the total size of the result and its metadata buffer and is associated with map tasks. The default value of 100 can be set higher, according to the HDFS block size, to 200 or 300.

- *io.sort.record.percent*—This is the percentage of the total metadata buffer size. The key-value pair combines the account information, which is fixed at 16 bytes, with the actual record, and is represented as a percentage ratio. This ratio should be adjusted based on the size of the key-value pair of the particular job, including the number of records and record size. Larger records should have a smaller account information ratio and smaller records should have a larger account information ratio.
- *mapred.job.shuffle.input.buffer.percent*—Increase the shuffle buffer size to store large and numerous map output in memory, while reserving part of the memory for the framework.
- *mapred.inmem.merge.threshold*—Avoid spill frequency by setting the value high enough, or set to zero (0) to disable it.
- *mapred.job.shuffle.merge.percent*—Adjust this value to balance between the spill frequency and the probability of the copier thread getting blocked.
- *mapred.job.reduce.input.buffer.percent*—Specify a relatively high value rather than the default (0), which can decrease disk I/O operations. Reserve enough memory space for the real reduce phase.

### MapReduce—Compression

- *mapred.output.compress* or *mapred.compress.map.output*—Compression can be enabled (or disabled) to compress both the intermediate and final output data sets on HDFS system. These settings should be set to true.
- *mapred.output.compression.codec* or *mapred.map.output.compression.codec*—The codec can be configured using any one of a variety of compression types such as zlib\*, LZ0\*, and gzip\*. Benchmark results indicate that LZ0 and Snappy\* are the most well-balanced and efficient codecs. Benchmarking tests compared using no compression to using zlib and LZ0 codecs. Although zlib has a higher compression ratio than LZ0 (meaning it saves more I/O bandwidth), it still takes longer to complete. Tests also found

that zlib overwhelmed the CPUs. Tests showed that LZ0 functioned well across hardware resources, showing a 45 percent performance gain over zlib.

- *mapred.output.compression.type*—Each block in the HDFS system contains several records. Therefore, block-level compression is always better than record-level compression.

### HDFS System—Block Size and Handlers

- *dfs.block.size*—By default, the minimum block file size is 64 MB. To increase performance and decrease the mapping time, this number should be increased to 128 MB or 256 MB. An increase from 128 MB to 256 MB reduced running time by 7 percent.

### MapReduce—Load Balancing

- *mapred.reduce.tasks*—Generally, the number of reduce tasks should be smaller than map tasks in an Apache Hadoop job.
- *mapred.reduce.slowstart.completed.maps*—Depending on the job, this value can be increased or decreased to set the delay for the reduce tasks, which leaves resources available for other tasks. Use a higher value for larger data sets to increase the delay and smaller values for smaller data sets to decrease the delay. For example, if the variable is set to 50 percent, then the reduce tasks will start after 50 percent of the mapped tasks have finished.
- *mapred.reduce.parallel.copies*: This tracks the number of copies read that can be executed concurrently. The default setting is 5. To speed up the sort workload, specify a value between 16 and 25. Tests show that there is no benefit to setting the number much higher than this.

## 9.2 Configuring and Optimizing the Software Layer

This section is most relevant to the IT infrastructure team. Selecting and configuring the operating system and application software have major implications for performance and stability.

**Configure Java settings:** The Apache Hadoop framework and many of its applications run on Java. It is extremely important that the JVM runs as optimally as possible.

### Garbage Collection/Java Virtual Machine

- Use “server” mode to appropriately manage resources for garbage collection (GC), especially for Apache Hadoop processes such as JobTracker and NameNode.
- Enable the following GC options to support the Apache Hadoop framework:
  - Use *parallel GC algorithm*
  - `XX:ParallelGCThreads=8`
  - `XX:+UseConcMarkSweepGC`
- Set the parameter `java.net.preferIPv4Stack` to True to reduce overhead.

### Configure Hadoop framework settings

Settings for the HDFS system and MapReduce also need to be configured for optimal performance.

#### HDFS System—Block Size and Handlers

- `dfs.datanode.max.xcievers`—The default maximum number of threads that can connect to a data node simultaneously is 256. If this value is not high enough, you will receive an I/O exception error. To prevent this error, increase the `xreceiver` number to a higher number, such as 2,048, before starting the data node services.

#### MapReduce—Load Balancing

- `mapred.tasktracker.[map/reduce].tasks.maximum`—This setting determines the right number of task slots on each TaskTracker. The maximum number of map and reduce slots will be set in the range of  $(\text{cores\_per\_node})/2$  to  $2 \times (\text{cores\_per\_node})$ . For example, an Intel® Xeon processor 5500 with eight cores, dual sockets, and hyper threading (HT) turned on would require 16 map slots and eight reduce slots. For an Intel® Xeon® processor 5600 with 12 cores and dual sockets, the variable should be set to 24 map slots and eight reduce slots.

### Optimize Linux\* operating system installation

The subsystem in the Linux operating system can be configured to improve I/O performance for the Apache Hadoop system.

#### File System

- Use ext4 as the local file system on slave nodes. Although ext2, ext3, and XFS can be used, benchmarking studies show significant performance improvements with ext4. Gains were over 15 percent when using ext4 over the default, ext3, and even greater with other file system types.
- When using the ext4 file system, disable the recording of the file system access time, using `noatime` and `nodiratime` options, to improve performance as much as 10 percent.
- When using the ext4 file system, use the ordered or write back journal mode for increased performance. Using journal mode in other file system types will actually decrease performance and should be disabled.
- When using the ext4 file system, increase the inode size from 256 K to 4 MB (`-T largefile4`) to improve performance by as much as 11 percent.
- Increase the read-ahead buffer size to improve the performance of sequential reads of large files. For example, increasing the size from 256 sectors to 2,048 sectors can save about 8 percent in running time.

#### Operating System

- Increase the Linux open-file-descriptor limit using `/etc/security/limits.conf`. The default value of 1,024 is too low for the Apache Hadoop daemon and may result in I/O exceptions in the JVM layer. Increase this value to approximately 64,000.
- If using kernel 2.6.28, increase the `epoll` file descriptor limit using `/etc/sysctl.conf`. The default value of 128 is too low for the Apache Hadoop daemon. Increase it to approximately 4,096.

- Increase `nrproc` (number of processes) in `/etc/security/limit.conf`. The default value is 90, which is too small to run the Apache Hadoop daemon and may result in I/O exception errors like “java.lang.OutOfMemoryError: unable to create new native thread.” For the Red Hat\* Enterprise Linux 6 operating system, edit `90-*.conf` under folder `etc/security/limits.d`, and set the hard and soft `nproc` to unlimited.

### 9.3 Configuring and Optimizing the Hardware Layer

This section is most relevant to the IT infrastructure team. Determining the configuration of the servers in the cluster is critically important for providing the highest performance and reliability of these machines. Because workloads may be bound by I/O, memory, or processor resources, system-level hardware also may need to be adjusted on a case-by-case basis.

#### Enable hardware settings

Optimizing hardware is often a balance between cost, capacity, and performance.

#### Processor

- Enabling hyper-threading will benefit CPU-intensive workloads and does not impact I/O-intensive workloads. For example, benchmarking tests show that HT can run as much as 25 percent more tasks per minute.

#### Network

- Enable channel bonding to resolve network-bound and I/O-intensive workloads. The channel bonding of two NIC ports will double the bandwidth and can improve the sort workload running time by 30 percent
- Multiple-RX/TX-queue supported. Try to bind the queue to a different core, which can spread out the network interrupts onto different cores.

**Table 5: HiBench, The Details**  
**Intel's HiBench suite looks at 10 workloads in four categories**

Category	Workload	Introduction
Micro benchmarks	Sort	<ul style="list-style-type: none"> <li>This workload sorts its input data, which is generated using the Apache Hadoop* RandomTextWriter example.</li> <li>Representative of real-world MapReduce* jobs that transform data from one format to another.</li> </ul>
	WordCount	<ul style="list-style-type: none"> <li>This workload counts the occurrence of each word in the input data, which is generated using Apache Hadoop RandomTextWriter.</li> <li>Representative of real-world MapReduce jobs that extract a small amount of interesting data from a large data set.</li> </ul>
	TeraSort	<ul style="list-style-type: none"> <li>A standard benchmark for large-size data sorting that is generated by the TeraGen program.</li> </ul>
	Enhanced DFSIO	<ul style="list-style-type: none"> <li>Tests HDFS* system throughput of a Hadoop cluster.</li> <li>Computes the aggregated bandwidth by sampling the number of bytes read or written at fixed time intervals in each map task.</li> </ul>
Web search	Nutch Indexing	<ul style="list-style-type: none"> <li>This workload tests the indexing subsystem in Nutch*, a popular Apache* open-source search engine. The crawler subsystem in Nutch is used to crawl an in-house Wikipedia* search engine. The crawler subsystem in Nutch is used to crawl an in-house Wikipedia* as workload input.</li> <li>This large-scale indexing system is one of the most significant uses of MapReduce (for example, in Google* and Facebook* platforms).</li> </ul>
	Page Rank	<ul style="list-style-type: none"> <li>This workload is an open-source implementation of the page-rank algorithm, a link-analysis algorithm used widely in web search engines.</li> </ul>
Machine learning	K-Means Clustering	<ul style="list-style-type: none"> <li>Typical application area of MapReduce for large-scale data mining and machine learning (for example, in Google and Facebook platforms).</li> <li>K-Means is a well-known clustering algorithm.</li> </ul>
	Bayesian Classification	<ul style="list-style-type: none"> <li>Typical application area of MapReduce for large-scale data mining and machine learning (for example, in Google and Facebook platforms).</li> <li>This workload tests the naive Bayesian (a well-known classification algorithm for knowledge discovery and data mining) trainer in the Mahout* open-source machine-learning library from Apache.</li> </ul>
Analytical query	Hive* Join	<ul style="list-style-type: none"> <li>This workload models complex analytic queries of structured (relational) tables by computing the sum of each group over a single read-only table.</li> </ul>
	Hive Aggregation	<ul style="list-style-type: none"> <li>This workload models complex analytic queries of structured (relational) tables by computing both the average and sum for each group by joining two different tables.</li> </ul>

### Hard Drive Settings

- Run in Advanced Host Controller Interface (AHCI) mode with Native Command Queuing (NCQ) enabled to improve the I/O performance of multiple, simultaneous read/write activities.
- For better I/O performance, enable the hard drive's write cache.

### 9.4 Benchmarking

Benchmarking is the quantitative foundation for measuring the success of any computer system. Intel developed the HiBench suite as a comprehensive set of benchmarks for the Apache Hadoop framework. The individual measures represent important Apache Hadoop applications across a range of tasks. HiBench includes synthetic micro benchmarks as well as real-world Apache Hadoop applications representative of a wider range of large-scale data analytics (for example, search indexing and machine learning).

Not all these benchmarks may be relevant for your organization. The following will help you understand what each benchmark measures, so you can map the relevant ones to your own Apache Hadoop environment. HiBench 2.1 is now available as open source under Apache\* License 2.0 at <https://github.com/hibench/HiBench-2.1>.

### 10.0 Conclusions

The Intel® Distribution for Apache Hadoop\* software lowers the barrier to adoption for organizations looking to use Apache Hadoop in production. With a customer-centered approach, Intel® Distribution will allow creation of rapidly deployable and highly optimized end-to-end Hadoop solutions running on commodity hardware. Continuing with Intel® Distribution, Intel will architect and develop the software components to address Hadoop deployment requirements at an enterprise level.

As Apache Hadoop becomes the de facto platform for business-critical applications, the data that is stored in Apache Hadoop becomes crucial for ensuring business continuity. Intel Hadoop provides the right mix of Apache Hadoop components with enhancements to the Apache Hadoop ecosystem, to not only provide a distributed linearly scalable data warehousing platform but also includes sufficient features and tools to provide near real-time ad-hoc query capabilities using Hbase and Hive.

Most of the popular enterprise-grade big data applications for extract, transform, load (ETL) processing and real time continuous analytics of large unstructured data over Hadoop have been tested to work out of the box on the Intel® Distribution for Apache Hadoop\* software.

Lastly, the open, integrated approach to enterprise-wide systems management enables the end user to build a comprehensive big data solution based on open standards integrated with industry-leading partners.

## 11.0 References & Contacts

Intel Big Data Resources:  
[www.intel.com/bigdata](http://www.intel.com/bigdata)

Download Intel® Distribution for Apache Hadoop\* software at [hadoop.intel.com](http://hadoop.intel.com)

Contact: +1-855-229-5580 or email  
[ASIPcustomer@intel.com](mailto:ASIPcustomer@intel.com)

## 12.0 Abbreviations

**Table 6: Abbreviations**

Abbreviation	Definition	Abbreviation	Definition
DBMS	Database management system	BMC	Baseboard management controller
EDW	Enterprise data warehouse	AHCI	Advanced Host Controller Interface
HDFS	Hadoop File System	NCQ	Native Command Queuing
NIC	Network interface card	GBE	Gigabit Ethernet
OS	Operating system	LZO	Lempel-Ziv-Oberhumer, a lossless data compression algorithm
ToR	Top-of-rack switch/router	LAN	Local Area Network

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.


The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order. Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's Web site at [www.intel.com](http://www.intel.com).

Copyright © 2013 Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, and the Intel Xeon badge are trademarks of Intel Corporation in the U.S. and other countries.

\*Other names and brands may be claimed as the property of others.

Printed in USA

0213/TC/PRW/PDF

 Please Recycle

328693-001US

