

HDP Getting Started Guide

1.0 (Nov 30, 2012)

HDP Getting Started Guide

1.0 (2012-11-30)

Copyright © 2012 Hortonworks, Inc. All rights reserved.

Hortonworks Data Platform (HDP) is a 100% open source data management platform based on Apache Hadoop. It allows you to load, store, process and manage data in virtually any format and at any scale. As the foundation for the next generation enterprise data architecture, HDP includes all of the necessary components to begin uncovering business insights from the quickly growing streams of data flowing into and throughout your business – TODO - verify

Table of Contents

1. Hardware Recommendations For Apache Hadoop	1
1.1. Overview	1
1.2. Typical Hadoop Cluster	1
1.3. Typical Workloads Patterns For Hadoop	2
1.4. Balanced Workload Deployments	3
1.5. Server Node Hardware Recommendations	5
1.5.1. Hardware selection for slave nodes (DataNodes, TaskTrackers, and RegionServers)	5
1.5.2. Hardware selection for master nodes (NameNode, JobTracker, HBase Master)	8
1.6. Hardware selection for HBase	9
1.7. Other Issues	10
1.8. Conclusion	11

List of Tables

1.1. For small clusters (5-50 nodes):	11
1.2. For medium to large clusters (100s to 1000s nodes):	11

1. Hardware Recommendations For Apache Hadoop

1.1. Overview

Hadoop and HBase workloads tend to vary a lot and it takes experience to correctly anticipate the amounts of storage, processing power, and inter-node communication that will be required for different kinds of jobs.

This document provides insights on choosing the appropriate hardware components for an optimal balance between performance and both initial as well as the recurring costs (see: [Conclusion](#) for a brief summary of the hardware sizing recommendations).

Hadoop is a software framework that supports large-scale distributed data analysis on commodity servers. Hortonworks is a major contributor to open source initiatives (Apache Hadoop, HDFS, Pig, Hive, HBase, Zookeeper) and has extensive experience managing production level Hadoop clusters. Hortonworks recommends following the design principles that drive large, hyper-scale deployments. For a Hadoop or HBase cluster, it is critical to accurately predict the size, type, frequency, and latency of analysis jobs to be run. When starting with Hadoop or HBase, begin small and gain experience by measuring actual workloads during a pilot project. This way you can easily scale the pilot environment without making any significant changes to the existing servers, software, deployment strategies, and network connectivity.

1.2. Typical Hadoop Cluster

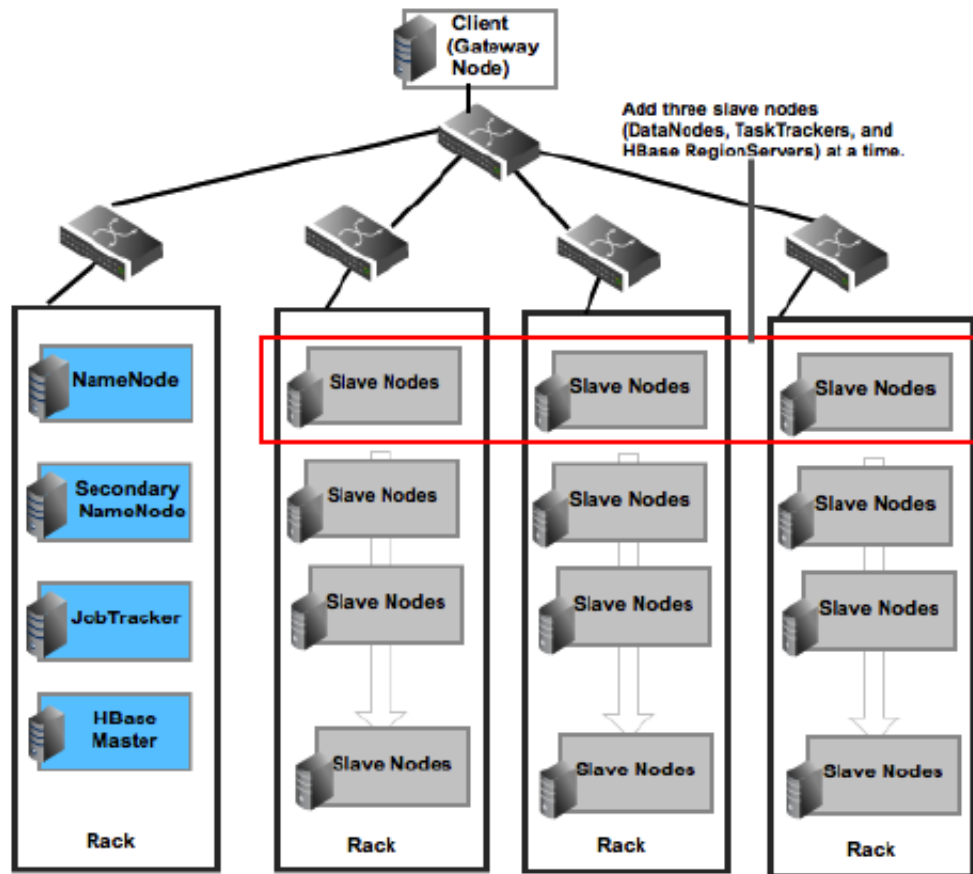
Hadoop and HBase clusters have two types of machines: **masters** (the HDFS NameNode, the MapReduce JobTracker, and the HBase Master) and **slaves** (the HDFS DataNodes, the MapReduce TaskTrackers, and the HBase RegionServers). The DataNodes, TaskTrackers, and HBase RegionServers are co-located or co-deployed for optimal data locality. In addition, HBase requires the use of a separate component (ZooKeeper) to manage the HBase cluster.

Hortonworks recommends separating master and slave nodes because of the following reasons:

- Task workloads on the slave nodes should be isolated from the masters.
- Slaves nodes are frequently decommissioned for maintainance.

For evaluation purpose, you can also choose to deploy Hadoop using single-node installation (all the masters and the slave processes reside on the same machine). Setting up a small cluster (of two nodes) is a very straightforward task - one node acts as both NameNode/JobTracker and the other node acts as DataNode and TaskTracker. Clusters of three or more machines typically use a dedicated NameNode/JobTracker and all the other nodes act as the slave nodes. Typically, medium to large Hadoop cluster consists of a two or three-level architecture built with rack-mounted servers. Each rack of servers is

interconnected using a 1 Gigabit Ethernet (GbE) switch. Each rack-level switch is connected to a cluster-level switch (which is typically a larger port-density 10GbE switch). These cluster-level switches may also interconnect with other cluster-level switches or even uplink to another level of switching infrastructure.



NOTE: DataNodes, TaskTrackers, and RegionServers are typically co-deployed.

1.3. Typical Workloads Patterns For Hadoop

Disk space, I/O Bandwidth (required by Hadoop), and computational power (required for the MapReduce processes) are the most important parameters for accurate hardware sizing. Additionally, if you are installing HBase, you also need to analyze your application and its memory requirements, because HBase is a memory intensive component. Based on the typical use cases for Hadoop, the following workload patterns are commonly observed in production environments:

Balanced Workload

If your jobs are distributed equally across the various job types (CPU bound, Disk I/O bound, or Network I/O bound), your cluster has balanced workload pattern. This is a good default configuration for unknown or evolving workloads.

Compute Intensive

These workloads are CPU bound and are characterized by the need of large number of CPUs and large amounts of memory to store in-process data. (This usage pattern is typical for natural language processing or HPCC workloads.)

I/O Intensive

Typical MapReduce job (like sorting) requires very little compute power but relies more on the I/O bound capacity of the cluster (for example if you have lot of cold data). Hadoop clusters utilized for such workloads are typically I/O intensive. For this type of workload, we recommend investing in more disks per box.

Unknown or evolving workload patterns

Most teams looking to build a Hadoop cluster are often unaware of their workload patterns. Also, the first jobs submitted to Hadoop are very different than the actual jobs in the production environments. For these reasons, Hortonworks recommends that you either use the Balanced workload configuration or invest in a pilot Hadoop cluster and plan to evolve as you analyze the workload patterns in your environment.

1.4. Balanced Workload Deployments

When a team is just starting with Hadoop or HBase, begin small and gain experience by measuring actual workloads during a pilot project. We recommend starting with a relatively small pilot cluster, provisioned for a “**balanced**” workload.

For pilot deployments, you can start with 1U/machine and use the following recommendations:

Two quad core CPUs | 12 GB to 24 GB memory | Four to six disk drives of 2 terabyte (TB) capacity.

The minimum requirement for network is 1GigE all-to-all and can be easily achieved by connecting all of your nodes to a Gigabit Ethernet switch. In order to use the spare socket for adding more CPUs in future, you can also consider using either a six or an eight core CPU.

For small to medium HBase clusters, provide each Zookeeper server around 1GB of RAM and if possible its own disk.

Jumpstart - Hadoop Cluster

One way to quickly deploy Hadoop cluster, is to opt for “cloud trials” or use virtual infrastructure. Hortonworks makes the distribution available through Hortonworks Data Platform (HDP). HDP can be easily installed in public and private clouds using Whirr, Microsoft Azure, and Amazon Web Services. For more details, contact the [Hortonworks Support Team](#).

However, note that cloud services and virtual infrastructures are not architected for Hadoop. Hadoop and HBase deployments in this case, might experience poor performance due to virtualization and suboptimal I/O architecture.

Tracking resource usage for pilot deployments

Hortonworks recommends that you monitor your pilot cluster using Ganglia, Nagios, or other performance monitoring framework that may be in use in your data center. You can also use the following guidelines to monitor your Hadoop and HBase clusters:

- Measure resource usage for CPU, RAM, Disk I/O operation per second (IOPS), and network packets sent and received. Run the actual kinds of query or analysis jobs that are of interest to your team.
- Ensure that you your data sub-set scaled to the size of your pilot cluster.
- Analyze the monitoring data for resource saturation. Based on this analysis, you can categorize your jobs as CPU bound, Disk I/O bound, or Network I/O bound.



Note

Most Java applications expand RAM usage to the maximum allowed. However, such jobs should not be analyzed as memory bound unless swapping happens or if the JVM experiences full-memory garbage collection events. (Full-memory garbage collection events are typically occur when the node appears to cease all useful work for several minutes at a time.)

- Optionally, customize your job parameters or hardware or network configurations to balance resource usage. If your jobs fall in the various workload patterns equally, you may also choose to only manipulate the job parameters and keep the hardware choices "balanced".
- For HBase cluster, you should also analyze Zookeeper, because the network and memory problems for HBase are often detected first in Zookeeper.

Using Hortonworks Data Platform (HDP) Monitoring Dashboard

You can also use the HDP Monitoring Dashboard for monitoring key metrics and alerts of your Hadoop clusters. HDP Monitoring Dashboard provides out of the box integration with Ganglia and Nagios. For more details, see: Hortonworks Data Platform.

Challenges - Tuning job characteristics to resource usage

Relating job characteristics to resource requirements is tricky for a variety of reasons we can only touch on briefly here. The method in which a job is coded or the job data is represented can have large impact on resource balance. For example, resource cost can be shifted between disk IOPS and CPU by the choice of compression scheme or parsing format; or per-node CPU and disk activity can be traded for inter-node bandwidth by the implementation of the Map/Reduce strategy.

Furthermore, Amdahl's Law shows how resource requirements can change in grossly non-linear ways with changing demands: a change that might be expected to reduce computation cost by 50% may instead cause a 10% change or a 90% change in net performance.

Reusing pilot machines

With a pilot cluster in place, start analyzing workloads patterns to identify CPU and I/O bottlenecks. It is common to have heterogeneous Hadoop clusters, especially as they evolve in size. Starting with a set of machines that are not perfect for your workload will

not impact your reuse capability, because these machines can be reused in the production clusters.



Tip

To achieve a positive return on investment (ROI), ensure that the machines in your pilot clusters are less than 10% of your production cluster.

1.5. Server Node Hardware Recommendations

The following recommendations provide insights into the best practices for selecting the number of nodes, storage options per node (number of disks, size of disks, MTBF, and the replication cost of disk failures), compute power per node (sockets, cores, clock speed), RAM per node, and network capability (number, speed of ports).

- [Hardware selection for slave nodes \(DataNodes, TaskTrackers, and RegionServers\)](#)
- [Hardware selection for master nodes \(NameNode, JobTracker, HBase Master\)](#)

While the hardware considerations in this section are generally applicable to all the servers in the Hadoop and HBase cluster, the focus here is on the slave nodes (DataNodes, TaskTrackers, and RegionServers). Slave nodes represent the majority of the infrastructure. (This section provides a general guideline for balanced workloads on slave nodes in production environments.)



Note

Hadoop cluster nodes do not require many features typically found in an enterprise data center server.

1.5.1. Hardware selection for slave nodes (DataNodes, TaskTrackers, and RegionServers)

The following recommendations are based on the Hortonworkers' experience gleaned in production data centers:

Choosing a server platform

Typically, dual-socket servers are optimal for Hadoop deployments. For medium to large clusters, using these servers is a best choice over the entry-level servers, because of the load-balancing and parallelization capabilities. In terms of density, it is advisable to select server hardware that fits into a low number of rack units. Typically, 1U or 2U servers are used in 19" racks or cabinets.

Storage options

For general-purpose Hadoop applications, we recommend using relatively large number of hard drives (typically eight to twelve SATA LFF drives) per server. At the time of this publication, typical capacity in production environments is around 2 TB per drive. Based on our experience, highly I/O intensive environments have started using 12 x 2 TB SATA drives. The optimal balance between cost and performance is generally achieved with 7,200 RPM

SATA drives. If your current or predicted storage is experiencing a growth rate you should also consider using 3 TB disks.

SFF disks are being adopted in some configurations for better disk bandwidth. However, we recommend that you monitor your cluster for any potential disk failures because more disks will increase the rate of disk failures. If you do have large number of disks per server, you can use two disk controllers, so that the I/O load can be shared across multiple cores. Hortonworks strongly recommends using either SATA or SAS interconnects only.

Once you have set-up an HDFS cluster using a low-cost reliable storage option, you will observe that the old data stays on the cluster indefinitely and the storage demands will grow quickly. With 12-drive systems, you typically get 24 TB or 36 TB per node. Using this storage capacity in a node is only practical with Hadoop release 1.0.0 or later (because the failures are handled gracefully allowing machines to continue serving from their remaining disks).

It is important to note that Hadoop is storage intensive and seek efficient, but does not require fast and expensive hard drives. If your workload pattern is not I/O intensive, it is safe to add only four or six disks per node. Note that power costs are proportional to the number of disks and not to terabytes. We therefore recommend that you add disks for storage and not for seeks.



Note

RAID vs. JBOD: Using RAID on Hadoop slave machines is not recommended, because Hadoop orchestrates data redundancy across all the slave nodes. However, it is strongly recommended to use RAID for Hadoop master machines (especially the NameNode server).

As a final consideration, we strongly recommend purchasing disk drives with good MTBF numbers, because the slave nodes in Hadoop suffer routine probabilistic failures.

Your slave nodes do not need expensive support contracts that offer services like replacement of disks within two hours or less. Hadoop is designed to adapt to slave node disk failure and therefore you should treat maintenance activity for the slave nodes as an ongoing task rather than an emergency.

It is good to be able to swap out disks without taking the server out of the rack, though switching them off (briefly) is an inexpensive operation in a Hadoop cluster. Using SSD disks for master nodes can increase your costs for bulk storage at present. As the costs for these disks decrease, it could present opportunities in future.

Memory sizing

In a Hadoop cluster, it is critical to provide sufficient memory to keep the processors busy without swapping and without incurring excessive costs for non-standard motherboards. Depending on the number of cores, your slave nodes typically require 24 GB to 48 GB of RAM for Hadoop applications. For large clusters, this amount of memory sufficiently provides extra RAM (approximately 4 GB) for the Hadoop framework and for your query and analysis processes (HBase and/or Map/Reduce).

To detect and correct random transient errors introduced due to thermodynamic effects and cosmic rays, we strongly recommend using error correcting code (ECC) memory. Error-

correcting RAM allows you to trust the quality of your computations. Some parts (chip-kill/chip spare) have been shown to offer better protection than traditional designs, as they show less recurrence of bit errors. (See, [DRAM Errors in the Wild: A Large-Scale Field Study, Schroeder et al, 2009.](#))

If you want to retain the option of adding more memory to your servers in future, ensure there is space to do this alongside the initial memory modules. It is expensive to replace all the memory modules.

Memory provisioning

Memory can also be provisioned at commodity prices on low-end server motherboards. It is typical to over-provision memory. The unused RAM will be consumed either by your Hadoop applications (typically when you run more processes in parallel) or by the infrastructure (used for caching disk data to improve performance).

Processors

Although it is important to understand your workload pattern, for most systems we recommend using medium clock speed processors with less than two sockets. For most workloads, the extra performance per node is not cost-effective. For large clusters, use at least two quad core CPU for the slave machines.

Power consideration

Power is a major concern when designing Hadoop clusters. Instead of purchasing the biggest and fastest nodes, it is important to analyze the power utilization for the existing hardware. We observed huge savings in pricing and power by avoiding fastest CPUs, redundant power supplies, etc.

Nowadays, vendors are building machines for cloud data centers that are designed to reduce cost, power, and are light-weight. Supermicro, Dell, and HP all have such product lines for cloud providers. So, if you are buying in large volume, we recommend evaluating the stripped-down "cloud servers".

For slave nodes, a single power supply unit (PSU) is sufficient, but for master servers use redundant PSUs. Server designs that share PSUs across adjacent servers can offer increased reliability without increased cost.

Some co-location sites bill based on the maximum-possible power budget and not the actual budget. In such a location the benefits of the power saving features of the latest CPUs are not realized completely. We therefore recommend checking the power billing options of the site in advance.



Note

Power consumption of the cluster: Electricity and cooling account for 33.33% to 50% of the equipment total life cycle cost in the modern data centers.

Network

This is the most challenging parameter to estimate because Hadoop workloads vary a lot. The key is buying enough network capacity at reasonable cost so that all nodes in the

cluster can communicate with each other at reasonable speeds. Large clusters typically use dual 1 GB links for all nodes in each 20-node rack and 2*10 GB interconnect links per rack going up to a pair of central switches.

A good network design will consider the possibility of unacceptable congestion at critical points in the network under realistic loads. Generally accepted oversubscription ratios are around 4:1 at the server access layer and 2:1 between the access layer and the aggregation layer or core. Lower oversubscription ratios can be considered if higher performance is required. Additionally, we also recommend having 1 GE oversubscription between racks.

It is critical to have dedicated switches for the cluster instead of trying to allocate a VC in existing switches -the load of a Hadoop cluster would impact the rest of the users of the switch. It is also equally critical to work with the networking team to ensure that the switches suit both Hadoop and their monitoring tools.

Design the networking so as to retain the option of adding more racks of Hadoop/HBase servers. Getting the networking wrong can be expensive to fix. The quoted bandwidth of a switch is analogous to the miles per gallon ratings of an automobile -you are unlikely to replicate it. "Deep buffering" is preferable to low-latency in switches. Enabling Jumbo Frames across the cluster improves bandwidth through better checksums and possibly also provide packet integrity.



Note

Network strategy for your Hadoop clusters Analyze the ratio of network-to-computer cost. Ensure that the network cost is always around 20% of your total cost. Network costs should include your complete network, core switches, rack switches, any network cards needed, etc. Keep in mind that Hadoop grew up with commodity!

1.5.2. Hardware selection for master nodes (NameNode, JobTracker, HBase Master)

The master nodes, being unique, have significantly different storage and memory requirements than the slave nodes.

Storage options

We recommend using dual NameNode servers - one primary and one secondary. Both NameNode servers should have highly reliable storage for their namespace storage and edit-log journaling. Typically, hardware RAID and/or reliable network storage are justifiable options.

The master servers should have at least four redundant storage volumes, some local and some networked, but each can be relatively small (typically 1TB).



Note

The RAID disks on the master nodes are where support contracts are needed. We recommend including an on-site disk replacement option in your support contract so that a failed RAID disk can be replaced faster.

Multiple vendors sell NAS software. It is important to check their specifications before you invest in any NAS software.

Storage options for JobTracker servers

JobTracker servers do not need the RAID storage because they save their persistent state to HDFS and the JobTracker server can actually be run on a slave node with a bit of extra RAM. However, using the same hardware specification as the NameNode server provides a plan for migrating the NameNode to the same server as the JobTracker in the case of the NameNode failure and a copy of the NameNode's state can be saved to the network storage.

Memory sizing

The amount of memory required for the master nodes depends on the number of file system objects (files and block replicas) to be created and tracked by the NameNode. 64 GB of RAM supports approximately 100 million files. Some sites are now experimenting with 128GB of RAM, for even larger namespaces.

Processors

The NameNodes and their clients are very "chatty". We therefore recommend providing 16 or even 24 CPU cores to handle messaging traffic for the master nodes.

Network

Providing multiple network ports and 10 GB bandwidth to the switch is also acceptable (if the switch can handle it).

1.6. Hardware selection for HBase

HBase uses different types of caches to fill up the memory, and as a general rule, the more memory HBase has, the better it can cache read requests. Each slave node in an HBase cluster (RegionServer) maintains a number of regions (regions are the chunks of the data in memory). For large clusters, it is important to ensure that the HBase Master and NameNode run on separate server machines. Note that in large scale deployments, the Zookeeper nodes are not co-deployed with the Hadoop/HBase slave nodes.

Choosing storage options

In a distributed setup, HBase stores its data in Hadoop DataNodes. To get the maximum read/write locality, HBase RegionServers and DataNodes are co-deployed on the same machines. Therefore, all recommendations for the DataNode/TaskTracker hardware setup are also applicable to the RegionServers. Depending on whether your HBase applications are read/write or processing oriented, you must balance the number of disks with the number of CPU cores available. Typically, you should have at least one core per disk.

Memory sizing

HBase Master nodes(s) are not as compute intensive as a typical RegionServer or the NameNode server. Therefore a more modest memory setting can be chosen for the HBase master. RegionServer memory requirements depend heavily on the workload characteristics

of your HBase cluster. Although over provisioning for memory benefits all the workload patterns, with very large heap sizes Java's stop-the-world GC pauses may cause problems.

In addition, when running HBase cluster with Hadoop core, you must ensure that you over-provision the memory for Hadoop MapReduce by at least 1 GB to 2 GB per task on top of the HBase memory.

1.7. Other Issues

Weight

The storage density of the latest generation of servers means that the weight of the racks needs to be taken into account. You should verify that the weight of a rack is not more than the capacity of the datacenter's floor.

Scalability

It is easy to scale a Hadoop cluster by adding new servers or whole server racks to the cluster and increasing the memory in the master nodes to deal with the increased load. This will generate a lot of "rebalancing traffic" at first, but will deliver extra storage and computation. The master nodes do matter, we therefore recommend that you pay the premiums for those machines.

Use the following guidelines to scale your existing Hadoop cluster:

- Ensure there is potential free space in the data centre near to the Hadoop cluster. This space should be able to accommodate the power budget for more racks.
- Plan the network to cope with more servers
- It might be possible to add more disks and RAM to the existing servers -and extra CPUs if the servers have spare sockets. This can expand an existing cluster without adding more racks or network changes.
- To perform this hardware upgrade in a live cluster can take considerable time and effort and so we recommend that you should plan the expansion one server at a time.
- CPU parts do not remain on the vendors price list forever. If you do plan to add a second CPU, consult with your reseller on when they will cut the price of CPUs that your existing parts and buy these parts when available. This typically takes at least 18 months time period.
- You are likely to need more memory in the master servers.

Support contracts

The concept to consider here is "care for the master nodes, keep an eye on the slave nodes". You do not need traditional enterprise-class support contracts for the majority of the nodes in the cluster, as their failures are more of a statistics issue than a crisis. The money saved in support can go into more slave nodes.

Commissioning

Hortonworks plans to cover the best practices commissioning a Hadoop cluster in a future document. For now, note that the “smoke tests” that come with the Hadoop cluster are a good initial test, followed by Terasort. Some of the major server vendors offer in factory commissioning of Hadoop clusters for an extra fee. This can have a direct benefit in ensuring that the cluster is working before you receive and pay for it. There is an indirect benefit in that if the terasort performance is lower on-site than in-factory, it is possible to conclude that the network is the likely culprit and so it is possible to track down the problem faster.

1.8. Conclusion

Achieving optimal results from a Hadoop implementation begins with choosing the correct hardware and software stacks. The effort involved in the planning stages can pay off dramatically in terms of the performance and the total cost of ownership (TCO) associated with the environment. Additionally, the following composite system stack recommendations can help benefit organizations in the planning stages:

Table 1.1. For small clusters (5-50 nodes):

Machine Type	Workload Pattern/ Cluster Type	Storage	Processor (# of Cores)	Memory (GB)	Network
Slaves	Balanced workload	Four to six 2 TB disks	One Quad	24	1 GB Ethernet all-to-all
	HBase cluster	Six 2 TB disks	Dual Quad	48	
Masters	Balanced and/or HBase cluster	Four to six 2 TB disks	Dual Quad	24	

Table 1.2. For medium to large clusters (100s to 1000s nodes):

Machine Type	Workload Pattern/ Cluster Type	Storage	Processor (# of Cores)	Memory (GB)	Network
Slaves	Balanced workload	Four to six 1 TB disks	Dual Quad	24	Dual 1 GB links for all nodes in a 20 node rack and 2 x 10 GB interconnect links per rack going to a pair of central switches.
	Compute intensive workload	Four to six 1 TB or 2 TB disks	Dual Hexa Quad	24-48	
	I/O intensive workload	Twelve 1 TB disks	Dual Quad	24-48	
	HBase clusters	Twelve 1 TB disks	Dual Hexa Quad	48-96	
Masters	All workload patterns/HBase clusters	Four to six 2 TB disks	Dual Quad	Depends on number of file system objects to be created by NameNode.	

For Further Reading

- [Best Practices for Selecting Apache Hadoop Hardware](#) (Hortonworks blog)
- [Amdahl's Law](#) (Wikipedia)
- “Installation Requirements, Hardware”, [HBase, The Definitive Guide](#) by Lars George, O'Reilly 2011, Chapter 2 (page 34 ff.)

- [Hadoop Network and Compute Architecture Considerations](#) by Jacob Rapp, Cisco (Hadoop World 2011 presentation)
- [Hadoop network design challenge](#) (Brad Hedlund.com)
- [Scott Carey's email](#) on smaller hardware for smaller clusters (email to general@hadoop.apache.org, Wed, 10 Aug 2011 17:24:25 GMT)
- [Failure Trends in a Large Disk Drive Population](#) – Google Research Paper
- HBase production deployments:
 - [StumbleUpon's HBase cluster](#)
 - [EBay's HBase cluster](#)