**Frequently asked questions for Hadoop**

## Q1. What is Hadoop?

[Hadoop](http://hadoop.apache.org/core/) is a distributed computing platform written in Java. It incorporates features similar to those of the [Google File System](http://en.wikipedia.org/wiki/Google_File_System) and of [MapReduce](http://en.wikipedia.org/wiki/MapReduce).

## Q2. What platforms and Java versions does Hadoop run on?

1. Java 1.6.x or higher
2. Linux and Windows are the supported operating systems, but BSD, Mac OS/X, and [OpenSolaris](http://wiki.apache.org/hadoop/OpenSolaris) are known to work. (Windows requires the installation of [Cygwin](http://www.cygwin.com/)).

## Q3 .How well does Hadoop scale?

Hadoop has been demonstrated on clusters of up to 4000 nodes. Sort performance on 900 nodes is good (sorting 9TB of data on 900 nodes takes around 1.8 hours) and [improving](http://wiki.apache.org/hadoop/FAQ?action=AttachFile&do=view&target=sort900-20080115.png) using these non-default configuration values:

* dfs.block.size = 134217728
* dfs.namenode.handler.count = 40
* mapred.reduce.parallel.copies = 20
* mapred.child.java.opts = -Xmx512m
* fs.inmemory.size.mb = 200
* io.sort.factor = 100
* io.sort.mb = 200
* io.file.buffer.size = 131072

Sort performances on 1400 nodes and 2000 nodes are pretty good too - sorting 14TB of data on a 1400-node cluster takes 2.2 hours; sorting 20TB on a 2000-node cluster takes 2.5 hours. The updates to the above configuration being:

* mapred.job.tracker.handler.count = 60
* mapred.reduce.parallel.copies = 50
* tasktracker.http.threads = 50
* mapred.child.java.opts = -Xmx1024m

## Q4. What kind of hardware scales best for Hadoop?

The short answer is dual processor/dual core machines with 4-8GB of RAM using ECC memory, depending upon workflow needs. Machines should be moderately high-end commodity machines to be most cost-effective and typically cost 1/2 - 2/3 the cost of normal production application servers but are not desktop-class machines. This cost tends to be $2-5K.

## Q5. I have a new node I want to add to a running Hadoop cluster; how do I start services on just one node?

This also applies to the case where a machine has crashed and rebooted, etc, and you need to get it to rejoin the cluster. You do not need to shutdown and/or restart the entire cluster in this case.

First, add the new node's DNS name to the conf/slaves file on the master node.

Then log in to the new slave node and execute:

$ cd path/to/hadoop  
$ bin/hadoop-daemon.sh start datanode  
$ bin/hadoop-daemon.sh start tasktracker

If you are using the dfs.include/mapred.include functionality, you will need to additionally add the node to the dfs.include/mapred.include file, then issue hadoop dfsadmin -refreshNodes and hadoop mradmin -refreshNodes so that the [NameNode](http://wiki.apache.org/hadoop/NameNode) and [JobTracker](http://wiki.apache.org/hadoop/JobTracker) know of the additional node that has been added.

## Q6. Is there an easy way to see the status and health of a cluster?

There are web-based interfaces to both the [JobTracker](http://wiki.apache.org/hadoop/JobTracker) ([MapReduce](http://wiki.apache.org/hadoop/MapReduce) master) and [NameNode](http://wiki.apache.org/hadoop/NameNode) (HDFS master) which display status pages about the state of the entire system. By default, these are located at<http://job.tracker.addr:50030/> and<http://name.node.addr:50070/>.

The [JobTracker](http://wiki.apache.org/hadoop/JobTracker) status page will display the state of all nodes, as well as the job queue and status about all currently running jobs and tasks. The [NameNode](http://wiki.apache.org/hadoop/NameNode) status page will display the state of all nodes and the amount of free space, and provides the ability to browse the DFS via the web.

You can also see some basic HDFS cluster health data by running:

$ bin/hadoop dfsadmin -report

## Q7. How much network bandwidth might I need between racks in a medium size (40-80 node) Hadoop cluster?

The true answer depends on the types of jobs you're running. As a back of the envelope calculation one might figure something like this:

60 nodes total on 2 racks = 30 nodes per rack Each node might process about 100MB/sec of data In the case of a sort job where the intermediate data is the same size as the input data, that means each node needs to shuffle 100MB/sec of data In aggregate, each rack is then producing about 3GB/sec of data However, given even reducer spread across the racks, each rack will need to send 1.5GB/sec to reducers running on the other rack. Since the connection is full duplex, that means you need 1.5GB/sec of bisection bandwidth for this theoretical job. So that's 12Gbps.

However, the above calculations are probably somewhat of an upper bound. A large number of jobs have significant data reduction during the map phase, either by some kind of filtering/selection going on in the Mapper itself, or by good usage of Combiners. Additionally, intermediate data compression can cut the intermediate data transfer by a significant factor. Lastly, although your disks can probably provide 100MB sustained throughput, it's rare to see a MR job which can sustain disk speed IO through the entire pipeline. So, I'd say my estimate is at least a factor of 2 too high.

So, the simple answer is that 4-6Gbps is most likely just fine for most practical jobs. If you want to be extra safe, many inexpensive switches can operate in a "stacked" configuration where the bandwidth between them is essentially backplane speed. That should scale you to 96 nodes with plenty of headroom. Many inexpensive gigabit switches also have one or two 10GigE ports which can be used effectively to connect to each other or to a 10GE core.

## Q8. I am seeing connection refused in the logs. How do I troubleshoot this?

You get a ConnectionRefused Exception when there is a machine at the address specified, but there is no program listening on the specific TCP port the client is using -and there is no firewall in the way silently dropping TCP connection requests.

Unless there is a configuration error at either end, a common cause for this is the Hadoop service isn't running

## Q9. Why is the 'hadoop.tmp.dir' config default user.name dependent?

We need a directory that a user can write and also not to interfere with other users. If we didn't include the username, then different users would share the same tmp directory. This can cause authorization problems, if folks' default umask doesn't permit write by others. It can also result in folks stomping on each other, when they're, e.g., playing with HDFS and re-format their filesystem.

## Q10. Does Hadoop require SSH?

Hadoop provided scripts (e.g., start-mapred.sh and start-dfs.sh) use ssh in order to start and stop the various daemons and some other utilities. The Hadoop framework in itself does not **require** ssh. Daemons (e.g. [TaskTracker](http://wiki.apache.org/hadoop/TaskTracker) and [DataNode](http://wiki.apache.org/hadoop/DataNode)) can also be started manually on each node without the script's help.

## Q11. What does "NFS: Cannot create lock on (some dir)" mean?

This actually is not a problem with Hadoop, but represents a problem with the setup of the environment it is operating.

Usually, this error means that the NFS server to which the process is writing does not support file system locks. NFS prior to v4 requires a locking service daemon to run (typically rpc.lockd) in order to provide this functionality. NFSv4 has file system locks built into the protocol.

In some (rarer) instances, it might represent a problem with certain Linux kernels that did not implement the flock() system call properly.

It is highly recommended that the only NFS connection in a Hadoop setup be the place where the [NameNode](http://wiki.apache.org/hadoop/NameNode) writes a secondary or tertiary copy of the fsimage and edits log. All other users of NFS are not recommended for optimal performance.

# 