Hama Installation Guide v0.5

Introduction

HAMA is a distributed computing framework based on BSP (Bulk Synchronous Parallel) [1] computing technique for massive scientific computations (e.g., matrix, graph, network, ..., etc) designed to run on massive data-sets stored in Hadoop Distributed File System(HDFS). It is currently an incubator project by the Apache Software Foundation.

Apache Hama leverages BSP computing techniques to speed up iteration loops during the iterative process that requires several passes of messages before the final processed output is available. It provides an easy and flexible programming model, as compared with traditional models of Message Passing [2].It is compatible with any distributed storage (e.g., HDFS, HBase, …, etc), so you can use the Hama BSP on your existing Hadoop clusters. Finding shortest paths, value of PI etc. are some of the problems tackled by Hama today.(See “Random Communication Benchmark” results at http://wiki.apache.org/hama/Benchmarks).

This document explains the installation steps for Hama. You can deploy Hama in the following ways:

- Local Mode
- Pseudo-Distributed Mode
- Distributed Mode
    - Pure Hama cluster
    - Hama on Hadoop-YARN.
    - Hama on cloud. (AWS or Rackspace)

If you are a beginner, local mode should be a good starting point. With some configuration changes, you can convert multiple local-mode configured Hama machines into a Hama cluster. Even if you are interested in distributed mode of Hama, it is recommended that you read the local mode instructions first and then go to distributed mode installation instructions. The document also explains the different ways to have Hama in a cluster or cloud.

HDFS Installation (Single node)

Apache Hama runs its algorithms on Hadoop Distributed File-System(HDFS). You can develop Hama jobs that uses Hadoop I/O library to read and write records to the file-system. HDFS is also vital for features like checkpointing and fault tolerance for Hama.

(The instructions below are picked up from Michael G. Noll's set of instructions for Hadoop installation on his homepage here - http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/)
Special thanks to him for permitting us to do so. He has also done a good job in explaining certain failure scenarios. So do refer to the link or other resources if any topic is not covered. You are most welcome to ask for help on our user mailing list - **hama-user@incubator.apache.org**

Note: Instructions provided were carried out on Ubuntu 11.10

Prerequisites

Java 6 Installation.

Download Java SE 6 from the following link. http://www.oracle.com/technetwork/java/javase/downloads/index.html

Download and install using the command.

```
$:~/Downloads$  ./jdk-6u31-linux-i586.bin
```

After extraction, locate the java parent directory in the current location and move it to preferred location.

```
$: ~/Downloads/jre1.6.0_31$sudo mv  jdk1.6.0_31  /opt/java-6-sun
$: ~/Downloads/jre1.6.0_31$ /opt/java-6-sun/bin/java -version
java version "1.6.0_31"
Java(TM) SE Runtime Environment (build 1.6.0_31-b04)
Java HotSpot(TM) Client VM (build 20.6-b01, mixed mode, sharing)
```

It is upto you on where to move this directory to. We would be assuming that the directory jre1.6.0_31 is moved to "/opt" , for the remaining part of the document, just as an example. On Ubuntu, one can install java using sudo apt-get command. Please refer to Michael's blog for installation. Update the $PATH and $JAVA_HOME variable based on this installation detail. The document would revisit this shortly. Verify the version of your Java with the command above to verify that you are using Sun or Oracle distribution of Java.

Configuring a Hadoop system user

It is a good practice to create a dedicated user for running hadoop. Follow the steps below to create a user ‘hduser’ in group ‘hadoop’.

```
user@virtualbox:~$  sudo addgroup hadoop
user@virtualbox:~$  sudo adduser --ingroup hadoop hduser
```

Configuring SSH

Hadoop uses SSH for communication within nodes and subsystems. The following steps would be enough to setup a single node cluster.
- Install ssh server and client.

```
user@virtualbox:~$  sudo apt-get install openssh-server openssh-client
```

- Change user to hduser

```
user@virtualbox:~$  sudo su - hduser
```

- Generate a SSH key for hduser

```
hduser@virtualbox:~$  ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.ssh/id_rsa.
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
The key fingerprint is:
c9:b4:74:04:aa:9b:81:12:29:95:00:5e:00:31:5a:9f hduser@hadoop-VirtualBox
The key's randomart image is:
+--[ RSA 2048]----+
|...work of art... |
+-----------------+
```

- Add the generated SSH key to enable access to the local machine

```
hduser@virtualbox:~$  cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

- Get your SSH key fingerprint known to the localhost, to avoid any more questions for future ssh connections to localhost.

```
hduser@virtualbox:~$  ssh localhost
```

```
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is cb:9b:b3:93:dd:4b:2e:2d:a0:bb:9c:bd:9b:04:d0:31.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
```

Disabling IPV6

This is not a required step. If you don't want to leave an unused IPV6 port in all systems do the following:

```
Add the following lines to:

/etc/sysctl.conf

# Disable IPV6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

Add to /etc/modprobe.d/blacklist.conf
# Disable IPV6
blacklist ipv6

Add to /etc/modprobe.d/blacklist-rare-network.conf
# Disable IPV6
alias net-pf-10 off
```

[Source: http://pastebin.com/qDr0cVAQ]

Save all the files and reboot the system.

Hadoop installation

In this section, we would be installing a single node hadoop system.

Download and install

Download hadoop-1.0.0.tar.gz from Apache Mirrors for Hadoop. As mentioned before, the document assumes "/opt/" as the parent directory for installing the binaries. Execute following commands to install hadoop.

- Untar the downloaded tar gz file in the location.

```
user@virtualbox:~$  sudo cd /opt
user@virtualbox:~$  sudo tar xzvf hadoop-1.0.0.tar.gz
```

- It is upto you to create a soft link named 'hadoop' or move the untarred directory as hadoop

```
user@virtualbox:~$  sudo ln -s hadoop-1.0.0  hadoop
OR
user@virtualbox:~$  sudo mv hadoop-1.0.0  hadoop
```

- Change the ownership of the untarred files to hduser

```
user@virtualbox:~$  sudo chown -R hduser:hadoop hadoop-1.0.0
user@virtualbox:~$  sudo chown -R hduser:hadoop hadoop
```

Note: If you have downloaded the hadoop-1.0.0-bin.tar.gz, the paths for some of the files mentioned in the next steps are different.

Update hduser environment

Make the following changes in the $HOME/.bashrc file or any other file you use to source your environment.

```
# Set Hadoop-related environment variables
export HADOOP_PREFIX=/opt/hadoop

# Set JAVA_HOME (we will also configure JAVA_HOME directly for Hadoop later on)
export JAVA_HOME=/opt/java-6-sun

# Some convenient aliases and functions for running Hadoop-related commands
unalias fs &> /dev/null
alias fs="hadoop fs"
unalias hls &> /dev/null
alias hls="fs -ls"

export PATH=$PATH:$HADOOP_PREFIX/bin:$JAVA_HOME/bin
```

Configure hadoop

For the following instructions we assume that your current directory is /opt/hadoop.

- Modify etc/hadoop/hadoop-env.sh (of conf/hadoop-env.sh) to update the JAVA_HOME variable.

```
# The java implementation to use.  Required.
export JAVA_HOME=/opt/java-6-sun
```

- Create directories for HDFS
    - Create a directory for the actual HDFS storage
    - Make sure hduser has the appropriate read and write permissions for files in the directory. In the following commands, we create an empty directory to keep the HDFS storage data in /opt/hdfs_data. You are free to use any directory, provided you don't work on it outside hadoop related operations. /home/user/Desktop would be a bad choice. The files stored in this

directory would be written by Hadoop filesystem.

```
user@virtualbox:~$ sudo mkdir -p /opt/hdfs_data
user@virtualbox:~$ sudo chown -R hduser:hadoop /opt/hdfs_data
```

- Create a directory for the temporary storage required for HDFS. This directory facilitates temporary storage for HDFS. We choose the name /opt/hdfs_tmp in this case.You are free to use any directory, provided you don't work on it outside hadoop related operations. /home/user/ Desktop would be a bad choice.

```
user@virtualbox:~$ sudo mkdir /opt/hdfs_tmp
user@virtualbox:~$ sudo chown -R hduser:hadoop /opt/hdfs_tmp
```

- Update /etc/hadoop/hdfs-site.xml (of conf/hdfs-site.xml)

Add the following lines between configuration property in hdfs_site.xml. We specify the data directory and the replication factor in this file. Please paste the text below in between <configuration> …. </configuration> tag.

```
<property>
 <name>dfs.replication</name>
 <value>1</value>
 <description>Default block replication.
 The actual number of replications can be specified when the file is created.
 The default is used if replication is not specified in create time.
 </description>
</property>
<property>
 <name>dfs.name.dir</name>
 <value>/opt/hdfs_data/</value>
 <description>Default block replication.
 The actual number of replications can be specified when the file is created.
 The default is used if replication is not specified in create time.
 </description>
</property>
```

Update etc/hadoop/core-site.xml (or conf/core-site.xml)

Add the following lines for configuration property in core_site.xml. We specify the temporary directory and the default file system name in the file. Please paste the text below in between <configuration> …. </configuration> tag.

```
<property>
 <name>hadoop.tmp.dir</name>
 <value>/opt/hdfs_tmp/</value>
 <description>A base for other temporary directories.</description>
</property>

<property>
```

```
 <name>fs.default.name</name>
 <value>hdfs://localhost:54310</value>
 <description>The name of the default file system.  A URI whose
 scheme and authority determine the FileSystem implementation.  The
 uri's scheme determines the config property (fs.SCHEME.impl) naming
 the FileSystem implementation class.  The uri's authority is used to
 determine the host, port, etc. for a filesystem.</description>
</property>
```

Deploy HDFS installation

We now have the configuration required to start HDFS service. We would be first formatting the HDFS. This is a step required only for the first time when you bring HDFS up. Please note that using the following command would remove data stored in HDFS if any. It is assumed that hadoop command is in the $PATH.

```
user@hadoop-VirtualBox:/opt/hadoop$ sudo su - hduser
hduser@hadoop-VirtualBox:~$ which hadoop
/opt/hadoop/bin/hadoop
hduser@hadoop-VirtualBox:~$ hadoop namenode -format

12/04/23 20:31:46 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = hadoop-VirtualBox/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 1.0.0
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/
branch  -1.0 -r 1214675; compiled by 'hortonfo' on Thu Dec 15 16:41:31 UTC 2011
************************************************************/
Re-format filesystem in /opt/hdfs_data ? (Y or N) Y
12/04/23 20:31:50 INFO util.GSet: VM type       = 32-bit
12/04/23 20:31:50 INFO util.GSet: 2% max memory = 19.33375 MB
12/04/23 20:31:50 INFO util.GSet: capacity      = 2^22 = 4194304 entries
12/04/23 20:31:50 INFO util.GSet: recommended=4194304, actual=4194304
12/04/23 20:31:51 INFO namenode.FSNamesystem: fsOwner=hduser
12/04/23 20:31:51 INFO namenode.FSNamesystem: supergroup=supergroup
12/04/23 20:31:51 INFO namenode.FSNamesystem: isPermissionEnabled=true
12/04/23 20:31:51 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
12/04/23 20:31:51 INFO namenode.FSNamesystem: isAccessTokenEnabled=false
accessKeyUpd  ateInterval=0 min(s), accessTokenLifetime=0 min(s)
12/04/23 20:31:51 INFO namenode.NameNode: Caching file names occuring more than 10 ti
mes
12/04/23 20:31:52 INFO common.Storage: Image file of size 112 saved in 0 seconds.
12/04/23 20:31:52 INFO common.Storage: Storage directory /opt/hdfs_data has been succ
```

```
essfully formatted.
12/04/23 20:31:52 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at hadoop-VirtualBox/127.0.1.1
************************************************************/
```

- Now run the following command to start HDFS.

```
hduser@virtualbox:~$/opt/hadoop/bin/start-dfs.sh
starting namenode, logging to /opt/hadoop-1.0.0/libexec/../logs/hadoop-hduser-
namenode-hadoop-VirtualBox.out
localhost: starting datanode, logging to /opt/hadoop-1.0.0/libexec/../logs/hadoop-hduser-
datanode-hadoop-VirtualBox.out
localhost: starting secondarynamenode, logging to /opt/hadoop-1.0.0/libexec/../logs/
hadoop-hduser-secondarynamenode-hadoop-VirtualBox.out
```

- Issue ‘jps’ command and verify if you have the following commands running.

```
8260 NameNode
8486 DataNode
8733 SecondaryNameNode
8777 Jps
```

- To stop HDFS run the following command.
As you can see the commands shuts down Namenode, Datanode and Secondary Namenode.

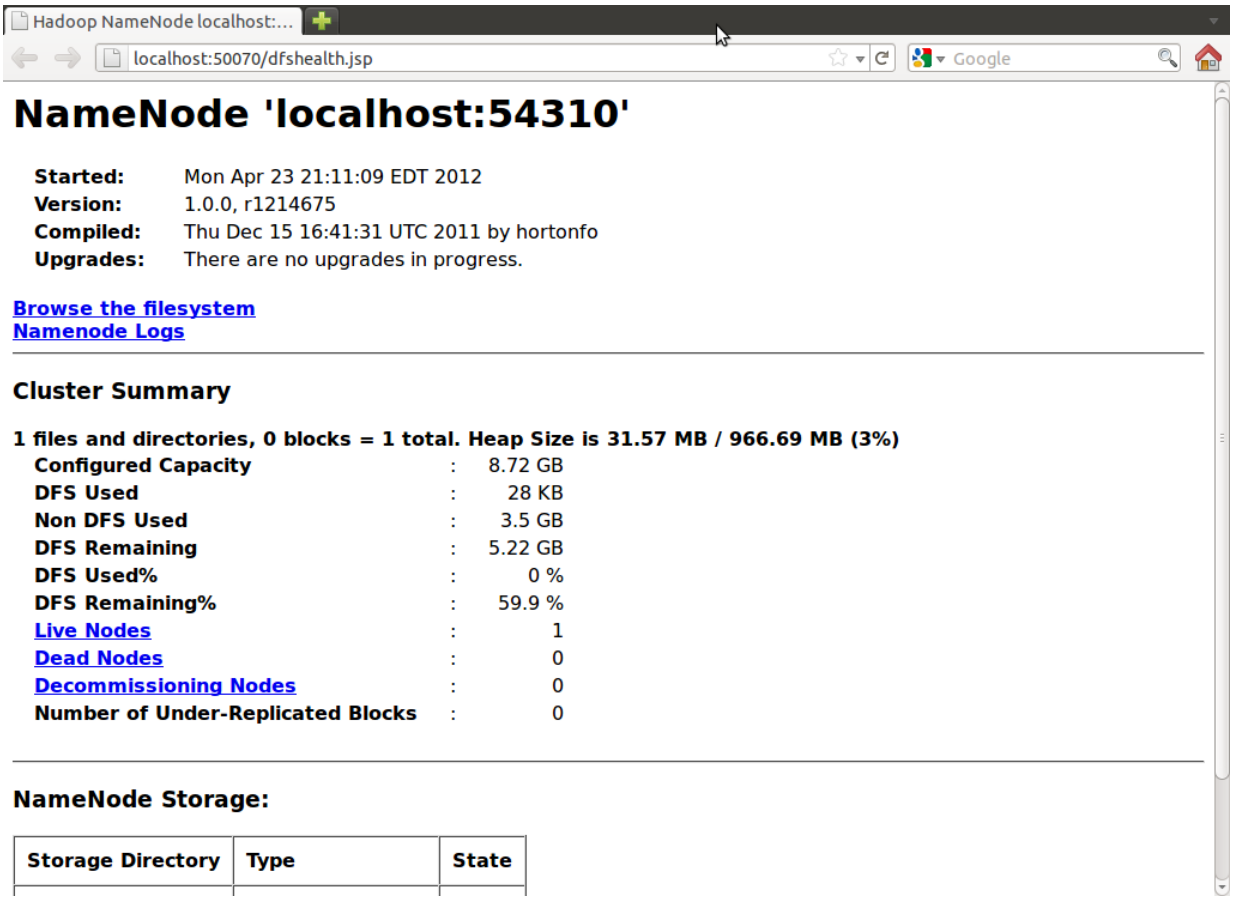
```
hduser@virtualbox:~$/opt/hadoop/bin/stop-dfs.sh
stopping namenode
localhost: stopping datanode
localhost: stopping secondarynamenode
```

Restart the HDFS service.
To check the status of HDFS, issue the following command:

```
hduser@virtualbox:~$  hadoop dfsadmin -report
Configured Capacity: 9361321984 (8.72 GB)
Present Capacity: 5605801984 (5.22 GB)
DFS Remaining: 5605773312 (5.22 GB)
DFS Used: 28672 (28 KB)
DFS Used%: 0%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
```

```
Missing blocks: 0

-------------------------------------------------
Datanodes available: 1 (1 total, 0 dead)

Name: 127.0.0.1:50010
Decommission Status : Normal
Configured Capacity: 9361321984 (8.72 GB)
DFS Used: 28672 (28 KB)
Non DFS Used: 3755520000 (3.5 GB)
DFS Remaining: 5605773312(5.22 GB)
DFS Used%: 0%
DFS Remaining%: 59.88%
Last contact: Mon Apr 23 21:20:07 EDT 2012
```

You can check the status on Web UI at the following address:**http://localhost:50070/**



You can refer to other sources for configuring hadoop to run map-reduce jobs. It is out of scope of this document. In the next section, we discuss how we can install and run Apache Hama BSP jobs over HDFS.

Hama Installation (Local/Pseudo-Distributed mode)

Now that we have installed hadoop file system on the machine, let us get into Apache Hama installation. It is similar to how we installed hadoop.

Download and install

- Download the tar-ball hama-dist-0.5.0-incubating.tar.gz.

```
user@virtualbox:~$ sudo tar xzvf hama-dist-0.5.0-incubating.tar.gz
```

- Create soft link or move the release directory as hama.

```
user@virtualbox:~$ sudo ln -s hama-0.5.0-incubating hama
OR
user@virtualbox:~$ sudo mv hama-0.5.0-incubating hama
```

Update user environment

- Change ownership permissions of all the files in hama directory.

```
user@virtualbox:~$ sudo chown -R hduser:hadoop /opt/hama
user@virtualbox:~$ sudo chown -R hduser:hadoop /opt/hama-0.5.0-incubating
```

- Add the following entries in the .bashrc file

```
export HAMA_HOME=/opt/hama
export PATH=$PATH:$HADOOP_PREFIX/bin:$HAMA_HOME/bin
```

So now the final version of bashrc file is as followed:

```
# Set Hadoop-related environment variables
export HADOOP_PREFIX=/opt/hadoop
export HAMA_HOME=/opt/hama
# Set JAVA_HOME (we will also configure JAVA_HOME directly for Hadoop later on)
export JAVA_HOME=/opt/java-6-sun

# Some convenient aliases and functions for running Hadoop-related commands
unalias fs &> /dev/null
alias fs="hadoop fs"
unalias hls &> /dev/null
```

```
alias hls="fs -ls"

export PATH=$PATH:$HADOOP_PREFIX/bin:$JAVA_HOME/bin:$HAMA_HOME/bin
```

Change Hama Configuration

For the following steps, it is assumed that your current directory is /opt

- Update conf/hama-env.sh. Set the value of JAVA_HOME variable with /opt/java-6-sun

```
export JAVA_HOME=/opt/java-6-sun
```

- Update hama-site.xml. It is easy to succumb to temptation of modifying the values in conf/hama-default.xml. However, the expected practice is to change the values in hama-site.xml. Add the following lines in between <configuration>.. </configuration>.

```
<property>
   <name>bsp.master.address</name>
   <value>localhost</value>
   <description>The address of the bsp master server. Either the
   literal string "local" or a host[:port] (where host is a name or
   IP address) for distributed mode.
   </description>
 </property>
 <property>
   <name>bsp.system.dir</name>
   <value>/tmp/hama-hduser/bsp/system</value>
   <description>The shared directory where BSP stores control files.
   </description>
 </property>
 <property>
   <name>bsp.local.dir</name>
   <value>/tmp/hama-hduser/bsp/local</value>
   <description>local directory for temporal store.</description>
 </property>
<property>
   <name>hama.tmp.dir</name>
   <value>/tmp/hama-hduser</value>
   <description>Temporary directory on the local filesystem.</description>
 </property>
 <property>
   <name>fs.default.name</name>
   <value>hdfs://localhost:54310</value>
   <description>
     The name of the default file system. Either the literal string
     "local" or a host:port for HDFS.
   </description>
```

```
 </property>
```

Deploy Hama

- Start the hama processes by issuing running the start-bspd.sh script. It starts the BSP Master and GroomServer process.

```
hduser@virtualbox:~$/opt/hadoop/bin/start-bspd.sh
starting bspmaster, logging to /opt/hama/bin/../logs/hama-hduser-bspmaster-hadoop-VirtualBox.out
localhost: starting groom, logging to /opt/hama/bin/../logs/hama-hduser-groom-hadoop-VirtualBox.out
```

- Verify if the processes have started by issuing jps command.

```
hduser@virtualbox:~$ jps
9702 SecondaryNameNode
11295 Jps
11247 GroomServerRunner
9244 NameNode
9471 DataNode
11053 BSPMasterRunner
11018 ZooKeeperRunner
```

- Verify the installation by issuing Hama jobs from the provided examples. We can invoke the BSP PI value estimator example by the following command.

```
hduser@virtualbox:~$ /opt/hama/bin/hama jar hama-examples-0.5.0-incubating.jar pi
12/04/23 23:01:50 INFO bsp.BSPJobClient: Running job: job_201204232255_0001
12/04/23 23:01:53 INFO bsp.BSPJobClient: Current supersteps number: 0
12/04/23 23:02:02 INFO bsp.BSPJobClient: Current supersteps number: 1
12/04/23 23:02:02 INFO bsp.BSPJobClient: The total number of supersteps: 1
12/04/23 23:02:02 INFO bsp.BSPJobClient: Counters: 8
12/04/23 23:02:02 INFO bsp.BSPJobClient:
  org.apache.hama.bsp.JobInProgress$JobCounter
12/04/23 23:02:02 INFO bsp.BSPJobClient:    LAUNCHED_TASKS=3
12/04/23 23:02:02 INFO bsp.BSPJobClient:
  org.apache.hama.bsp.BSPPeerImpl$PeerCounter
12/04/23 23:02:02 INFO bsp.BSPJobClient:    SUPERSTEPS=1
12/04/23 23:02:02 INFO bsp.BSPJobClient:    COMPRESSED_BYTES_SENT=225
12/04/23 23:02:02 INFO bsp.BSPJobClient:    SUPERSTEP_SUM=3
12/04/23 23:02:02 INFO bsp.BSPJobClient:    TIME_IN_SYNC_MS=3457
12/04/23 23:02:02 INFO bsp.BSPJobClient:    COMPRESSED_BYTES_RECEIVED=225
12/04/23 23:02:02 INFO bsp.BSPJobClient:    TOTAL_MESSAGES_SENT=3
12/04/23 23:02:02 INFO bsp.BSPJobClient:    TOTAL_MESSAGES_RECEIVED=3
Estimated value of PI is       3.1516
```

Job Finished in 12.879 seconds

- The logs for the task is written in directory /opt/hama/logs/taskLogs/

HDFS Installation (Multiple node)

In this section, we move forward on our hama cluster deployment. Before we get into distributed multi-node Hama cluster, we need to install HDFS on multiple nodes working with each other. For the sake of simplicity, we would be considering a maximum of 3 nodes as examples for explanation. Also, it is assumed that at this stage you have 3 nodes configured in local mode for HDFS.

Defining cluster

Network Definition

Specify all the hosts in your cluster in /etc/hosts file across all nodes in the cluster. As an example, a 3 node cluster /etc/hosts file would like:

```
192.168.2.6 hadoop-1
192.168.2.7 hadoop-2
192.168.2.8 hadoop-3
```

Use ping command to verify that every host above is reachable from every other two nodes.

Picking a master for Namenode

In this example, we shall pick hadoop-1 as the master. This just means that we have designated the host *hadoop-1* the responsibility to run Namenode process for HDFS.

SSH configuration.

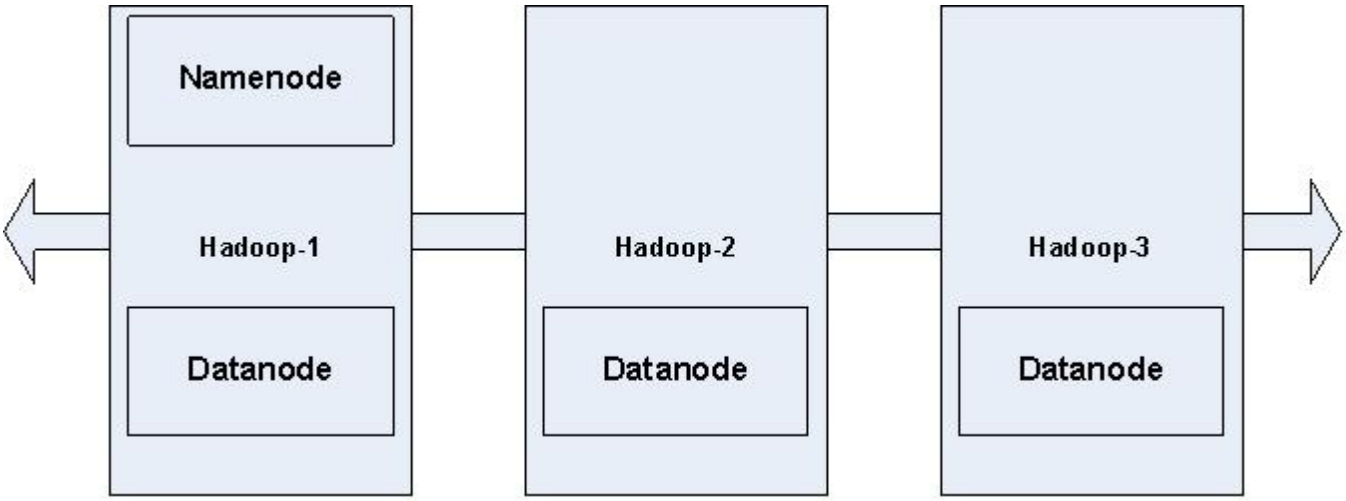
As we did for local installation, we need to get the public ssh-keys exchanged with our trusted nodes in the cluster and make the first connection to each of them so that all of them are added to the known_hosts at master. Run the following commands. Please note that we have to run ssh for hadoop-1 now, because we would now ssh to the machine using hostname ‘hadoop-1’ and not ‘localhost’

```
hduser@hadoop-1:~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@hadoop-2
hduser@hadoop-1:~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@hadoop-3
hduser@hadoop-1:~$ ssh hadoop-1
The authenticity of host 'master (192.168.2.6)' can't be established.
RSA key fingerprint is 3b:21:b3:c0:21:5c:7c:54:2f:1e:2d:96:79:eb:7f:95.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'master' (RSA) to the list of known hosts.
Linux master 2.6.20-16-386 #2 Mon Jan 23 20:16:13 UTC 2007 i686
hduser@hadoop-1:~$ ssh hadoop-2
…....
hduser@hadoop-1:~$ ssh hadoop-3
…....
```

Configuring Namenode and Datanodes.

Node responsibility allocation

In the previous section, we have ironed out the communication protocols between namenode and the datanodes. In this section, we define how we setup our HDFS cluster. The figure below, explains what we are trying to achieve.



Please note that until now we have not considered the BSPMaster and GroomServer allocation. It is advisable that we have separate servers each for BSPMaster and Namenode.The GroomServers and Datanodes should reside on remaining nodes. However, since we are short of machines in this small cluster and to make the example easier to understand, we will be having both BSPMaster and Namenode running on hadoop-1. The GroomServers and Datanodes would be running on all the nodes.

Configuration files

Make hadoop directory as current directory.

```
hduser@hadoop-1:~$cd /opt/hadoop
```

- Update *masters* file with the hostname of namenode. (In this case hadoop-1). This file should have same value for all three nodes.

```
hadoop-1
```

- Update *slaves* file with the hostnames of the slaves on Namenode host. (On hadoop-1 in our case)

```
hadoop-1
hadoop-2
hadoop-3
```

The above file implies that the datanodes would be running on all 3 nodes.
Delete slaves file on all the slave nodes (Nodes with only datanodes configured to run viz. in our case hadoop-2 and hadoop-3)

```
hduser@hadoop-2:~$rm -f conf/slaves
hduser@hadoop-3:~$rm -f conf/slaves
```

- Update conf/hdfs-site.xml

Add/Update the following lines between configuration property in hdfs_site.xml. We specify the data directory and the replication factor in this file. Please paste the text below in between <configuration> …. </configuration> tag. The replication factor could be set depending on the scale of redundancy you need and the size of cluster.

```
<property>
 <name>dfs.replication</name>
 <value>2</value>
 <description>Default block replication.
 The actual number of replications can be specified when the file is created.
 The default is used if replication is not specified in create time.
 </description>
</property>
<property>
 <name>dfs.name.dir</name>
 <value>/opt/hdfs_data/</value>
 <description>Default block replication.
 The actual number of replications can be specified when the file is created.
 The default is used if replication is not specified in create time.
 </description>
</property>
```

- Update conf/core-site.xml

Add/Update the following lines between configuration property in core_site.xml. We specify the temporary directory and the default file system name in the file.Please paste the text below in between <configuration> …. </configuration> tag.

```
<property>
 <name>hadoop.tmp.dir</name>
 <value>/opt/hdfs_tmp/</value>
 <description>A base for other temporary directories.</description>
</property>

<property>
```

```
 <name>fs.default.name</name>
 <value>hdfs://hadoop-1:54310</value>
 <description>The name of the default file system.  A URI whose
 scheme and authority determine the FileSystem implementation.  The
 uri's scheme determines the config property (fs.SCHEME.impl) naming
 the FileSystem implementation class.  The uri's authority is used to
 determine the host, port, etc. for a filesystem.</description>
</property>
```

- If you would like to manage your own Zooker instance, please specify the port as shown below in hama-site.xml

```
<property>
  <name>hama.zookeeper.property.clientPort</name>
  <value>2181</value>
 </property>
```

Deploy HDFS

We now have the configuration required to start HDFS service on multiple nodes. We would be first formatting the HDFS. This is a step required only for the first time when you bring HDFS up. Please note that using the following command would remove data stored in HDFS if any. It is assumed that hadoop command is in the $PATH.

```
user@hadoop-VirtualBox:/opt/hadoop$ sudo su - hduser
hduser@hadoop-VirtualBox:~$ which hadoop
/opt/hadoop/bin/hadoop
hduser@hadoop-VirtualBox:~$ hadoop namenode -format
```

- Now run the following command to start HDFS.

```
hduser@virtualbox:~$/opt/hadoop/bin/start-dfs.sh
starting namenode, logging to /opt/hadoop-1.0.0/libexec/../logs/hadoop-hduser-
namenode-hadoop-VirtualBox.out
localhost: starting datanode, logging to /opt/hadoop-1.0.0/libexec/../logs/hadoop-hduser-
datanode-hadoop-VirtualBox.out
localhost: starting secondarynamenode, logging to /opt/hadoop-1.0.0/libexec/../logs/
hadoop-hduser-secondarynamenode-hadoop-VirtualBox.out
```

- Issue ‘jps’ command and verify if you have the following commands running.

```
8260 NameNode
```

```
8486 DataNode
8733 SecondaryNameNode
8777 Jps
```

- To stop HDFS run the following command.
As you can see the commands shuts down Namenode on hadoop-1, Datanodes on all three nodes and Secondary Namenode on hadoop-1

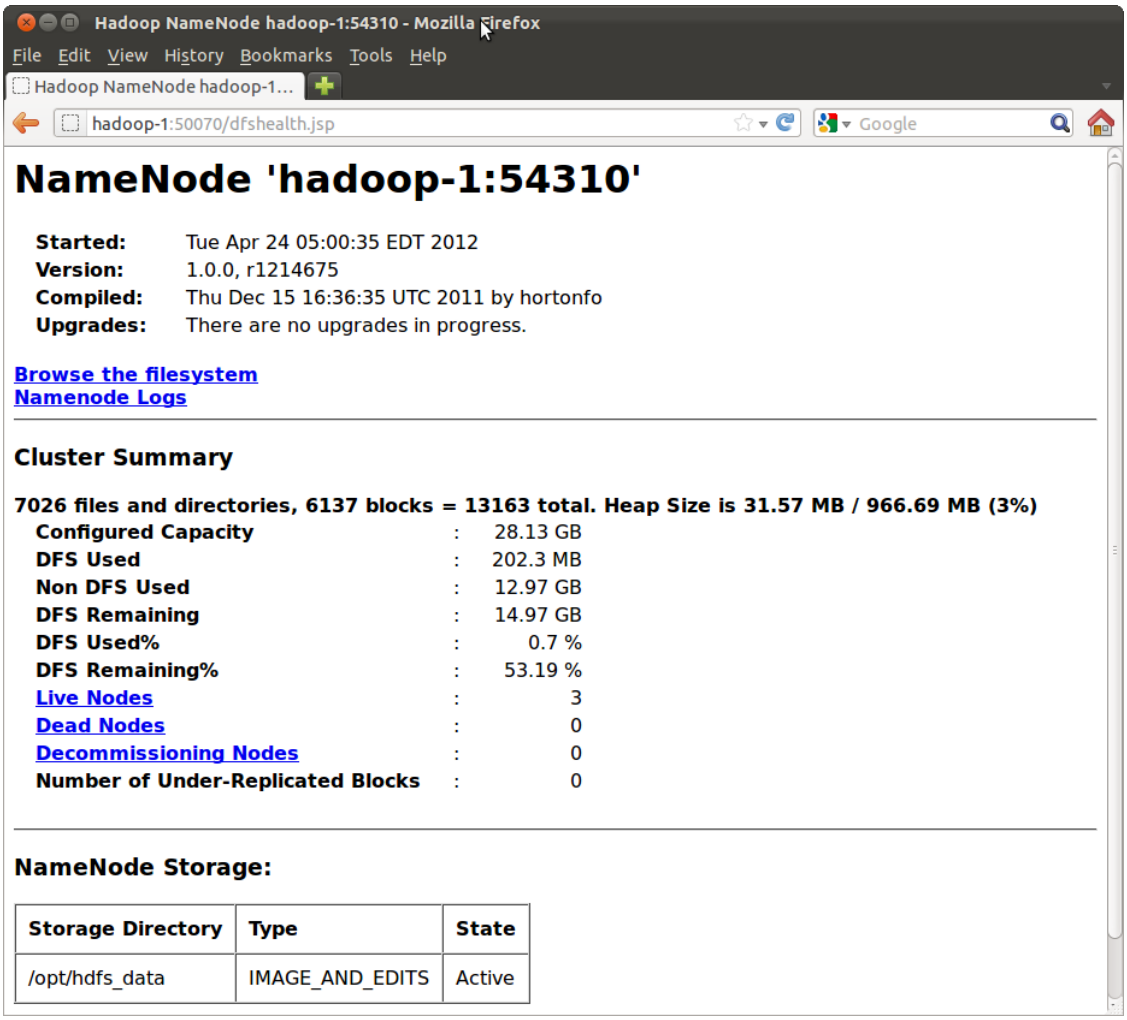
```
hduser@hadoop-1:~$/opt/hadoop/bin/stop-dfs.sh
stopping namenode
hadoop-1: stopping datanode
hadoop-2: stopping datanode
hadoop-3: stopping datanode
hadoop-1: stopping secondarynamenode
```

Restart the HDFS service.
To check the status of HDFS, issue the following command. You can see that it shows the HDFS status for all the three nodes.

```
hduser@hadoop-1:~$  hadoop dfsadmin -report


Upgrade for version -32 has been completed.
Upgrade is not finalized.
Configured Capacity: 30207897600 (28.13 GB)
Present Capacity: 16279965696 (15.16 GB)
DFS Remaining: 16067842048 (14.96 GB)
DFS Used: 212123648 (202.3 MB)
DFS Used%: 1.3%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0

-------------------------------------------------
Datanodes available: 3 (3 total, 0 dead)

Name: 192.168.2.8:50010
Decommission Status : Normal
Configured Capacity: 10069299200 (9.38 GB)
DFS Used: 70606848 (67.34 MB)
Non DFS Used: 4477755392 (4.17 GB)
DFS Remaining: 5520936960(5.14 GB)
DFS Used%: 0.7%
DFS Remaining%: 54.83%
Last contact: Tue Apr 24 04:55:37 EDT 2012


Name: 192.168.2.6:50010
```

```
Decommission Status : Normal
Configured Capacity: 10069299200 (9.38 GB)
DFS Used: 70340608 (67.08 MB)
Non DFS Used: 4974833664 (4.63 GB)
DFS Remaining: 5024124928(4.68 GB)
DFS Used%: 0.7%
DFS Remaining%: 49.9%
Last contact: Tue Apr 24 04:55:37 EDT 2012


Name: 192.168.2.7:50010
Decommission Status : Normal
Configured Capacity: 10069299200 (9.38 GB)
DFS Used: 71176192 (67.88 MB)
Non DFS Used: 4475342848 (4.17 GB)
DFS Remaining: 5522780160(5.14 GB)
DFS Used%: 0.71%
DFS Remaining%: 54.85%
Last contact: Tue Apr 24 04:55:36 EDT 2012
```

You can check the status on Web UI at the following address:**http://hadoop-1:50070/**



NameNode 'hadoop-1:54310'

Started: Tue Apr 24 05:00:35 EDT 2012
Version: 1.0.0, r1214675
Compiled: Thu Dec 15 16:36:35 UTC 2011 by hortonfo
Upgrades: There are no upgrades in progress.

Browse the filesystem
Namenode Logs

Cluster Summary

7026 files and directories, 6137 blocks = 13163 total. Heap Size is 31.57 MB / 966.69 MB (3%)

Configured Capacity	:	28.13 GB
DFS Used	:	202.3 MB
Non DFS Used	:	12.97 GB
DFS Remaining	:	14.97 GB
DFS Used%	:	0.7 %
DFS Remaining%	:	53.19 %
Live Nodes	:	3
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	0

NameNode Storage:

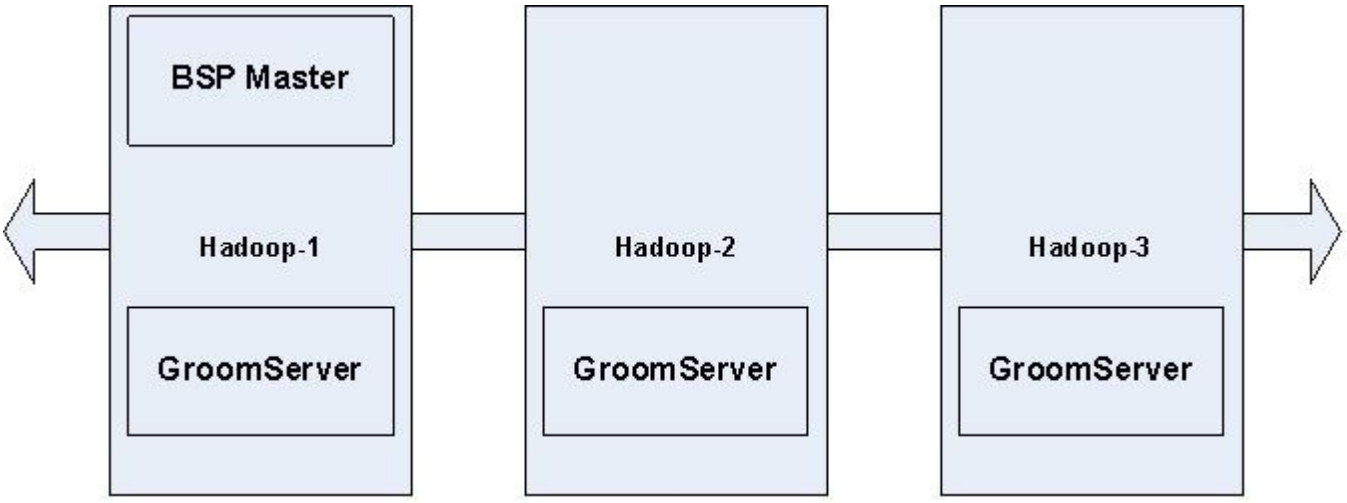
Storage Directory	Type	State
/opt/hdfs_data	IMAGE_AND_EDITS	Active

Hama Installation (Distributed mode)

Until now, we have installed and deployed HDFS on our cluster successfully. To deploy Hama cluster to run jobs over it is a matter of changing some configuration files. Once again, we have assumed that all the nodes are configured in local mode already. If you have navigated straight to this section, it is recommended that you read the local mode instructions first. By this stage, it is assumed that all the nodes have the release downloaded and untarred with the configuration parameters set such that we can run Hama jobs locally.

Define BSPMaster and GroomServers.

As in HDFS, in Hama too we have a master node that co-ordinates tasks with other slave nodes. Let us for this example, select hadoop-1 as the host for BSPMaster process and run GroomServers on all the nodes. This is depicted in the figure below.



Configuration

- Update hama-site.xml on all nodes. In this step we are specifying hadoop-1 as the host for bsp-master and also the directory locations used by Hama for control files and local temporary storage. More importantly, we specify the HDFS filesystem information for Hama to work on. Finally, we define the zookeeper quorum for our cluster. In our example, we have defined hadoop-1 and hadoop-2 for zookeeper quorum.

```
<property>
  <name>bsp.master.address</name>
  <value>hadoop-1</value>
```

```
  <description>The address of the bsp master server. Either the
  literal string "local" or a host[:port] (where host is a name or
  IP address) for distributed mode.
  </description>
</property>
<property>
  <name>bsp.system.dir</name>
  <value>/tmp/hama-hduser/bsp/system</value>
  <description>The shared directory where BSP stores control files.
  </description>
</property>
<property>
  <name>bsp.local.dir</name>
  <value>/tmp/hama-hduser/bsp/local</value>
  <description>local directory for temporal store.</description>
</property>
<property>
  <name>hama.tmp.dir</name>
  <value>/tmp/hama-hduser</value>
  <description>Temporary directory on the local filesystem.</description>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://hadoop-1:54310</value>
  <description>
    The name of the default file system. Either the literal string
    "local" or a host:port for HDFS.
  </description>
</property>
<property>
  <name>hama.zookeeper.quorum</name>
  <value>hadoop-1,hadoop-2</value>
  <description>Comma separated list of servers in the ZooKeeper quorum.
  For example, "host1.mydomain.com,host2.mydomain.com,host3.mydomain.com".
  By default this is set to localhost for local and pseudo-distributed modes
  of operation. For a fully-distributed setup, this should be set to a full
  list of ZooKeeper quorum servers. If HAMA_MANAGES_ZK is set in hama-env.sh
  this is the list of servers which we will start/stop ZooKeeper on.
  </description>
</property>
```

- You can also specify whether Hama can manage its own zookeeper by changing the following line in hama-env.sh

```
# Tell Hama whether it should manage it's own instance of Zookeeper or not.
export HAMA_MANAGES_ZK=true
```

- Now specify all the hosts where you want to run GroomServers in the file conf/groomservers. In our example, we are going to run GroomServers on all the nodes. Hence the following file would be written in the groomservers file.

```
hadoop-1
hadoop-2
hadoop-3
```

- You are free to duplicate hdfs-site.xml and groomservers files over all the nodes in your hama cluster.

Deploy Hama cluster

- Start hama cluster. We can see that it has started GroomServers on all the nodes and BSP Master is started only on hadoop-1. **Note:** It is recommended that you wait until HDFS is out of Safemode. You can check this by issuing the hadoop dfsadmin -report command or on the Namenode Web UI mentioned.

```
hduser@hadoop-1:~$ /opt/hama/bin/start-bspd.sh
hadoop-1: starting zookeeper, logging to /opt/hama/bin/../logs/hama-hduser-zookeeper-hadoop-1.out
hadoop-2: starting zookeeper, logging to /opt/hama/bin/../logs/hama-hduser-zookeeper-hadoop-2.out
starting bspmaster, logging to /opt/hama/bin/../logs/hama-hduser-bspmaster-hadoop-1.out
hadoop-2: starting groom, logging to /opt/hama/bin/../logs/hama-hduser-groom-hadoop-2.out
hadoop-1: starting groom, logging to /opt/hama/bin/../logs/hama-hduser-groom-hadoop-1.out
hadoop-3: starting groom, logging to /opt/hama/bin/../logs/hama-hduser-groom-hadoop-3.out
```

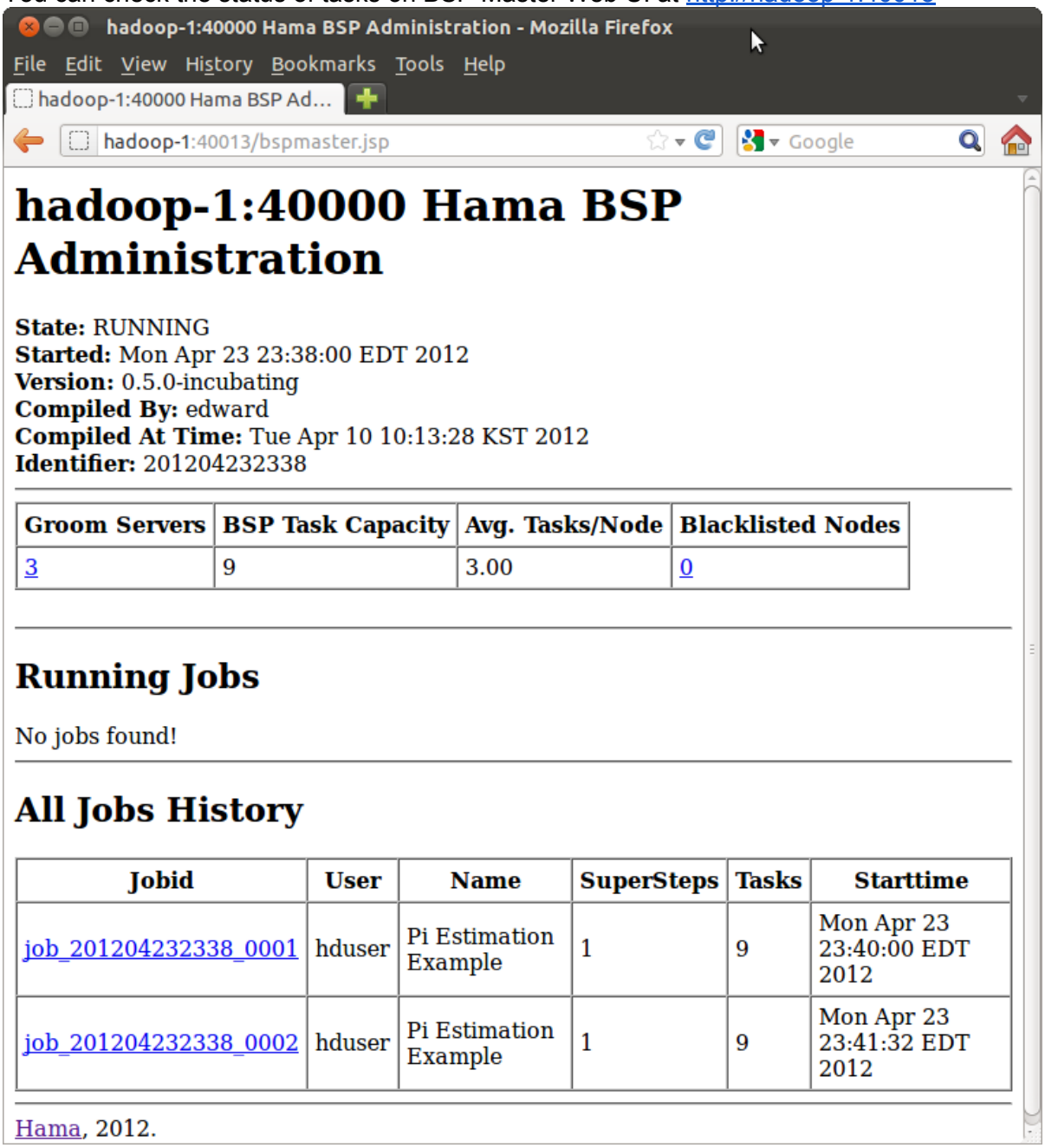
- To stop hama cluster use the following command.

```
hduser@hadoop-1:~$ /opt/hama/bin/stop-bspd.sh
stopping bspmaster
hadoop-3: stopping groom
hadoop-2: stopping groom
hadoop-1: stopping groom
hadoop-1: stopping zookeeper
hadoop-2: stopping zookeeper
```

- Run an example job

```
hduser@hadoop-1:~$ cd /opt/hama/
hduser@hadoop-1:~$ hama jar hama-examples-0.5.0-incubating.jar pi
12/04/23 23:41:32 INFO bsp.BSPJobClient: Running job: job_201204232338_0002
12/04/23 23:41:35 INFO bsp.BSPJobClient: Current supersteps number: 0
12/04/23 23:41:53 INFO bsp.BSPJobClient: Current supersteps number: 1
12/04/23 23:41:53 INFO bsp.BSPJobClient: The total number of supersteps: 1
12/04/23 23:41:53 INFO bsp.BSPJobClient: Counters: 8
12/04/23 23:41:53 INFO bsp.BSPJobClient:
 org.apache.hama.bsp.JobInProgress$JobCounter
12/04/23 23:41:53 INFO bsp.BSPJobClient:    LAUNCHED_TASKS=9
12/04/23 23:41:53 INFO bsp.BSPJobClient:
 org.apache.hama.bsp.BSPPeerImpl$PeerCounter
12/04/23 23:41:53 INFO bsp.BSPJobClient:    SUPERSTEPS=1
12/04/23 23:41:53 INFO bsp.BSPJobClient:    COMPRESSED_BYTES_SENT=675
12/04/23 23:41:53 INFO bsp.BSPJobClient:    SUPERSTEP_SUM=9
12/04/23 23:41:53 INFO bsp.BSPJobClient:    TIME_IN_SYNC_MS=18723
12/04/23 23:41:53 INFO bsp.BSPJobClient:    COMPRESSED_BYTES_RECEIVED=675
12/04/23 23:41:53 INFO bsp.BSPJobClient:    TOTAL_MESSAGES_SENT=9
12/04/23 23:41:53 INFO bsp.BSPJobClient:    TOTAL_MESSAGES_RECEIVED=9
Estimated value of PI is          3.1394222222222226
Job Finished in 21.563 seconds
```

- Web UI
You can check the status of tasks on BSP Master Web UI at http://hadoop-1:40013



Hama on YARN

Hama on Cloud

The following commands install Whirr and start a 5 node Hama cluster on Amazon EC2 in 5 minutes or less.

```
% curl -O http://www.apache.org/dist/whirr/whirr-0.x.0/whirr-0.x.0.tar.gz
% tar zxf whirr-0.x.0.tar.gz; cd whirr-0.x.0
% export AWS_ACCESS_KEY_ID=YOUR_ID
% export AWS_SECRET_ACCESS_KEY=YOUR_SECKEY
% ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa_whirr
% bin/whirr launch-cluster --config recipes/hama-ec2.properties --private -key-file
~/.ssh/id_rsa_whirr
```

Upon success you should see imok echoed to the console, indicating that ZooKeeper is running.
Now you can run an BSP examples as below:

```
edward@domU-12-31-39-0C-7D-41:/usr/local/hama-0.3.0-incubating$ bin/hama jar hama-
examples-0.3.0-incubating.jar
An example program must be given as the first argument.
Valid program names are:
  bench: Random Communication Benchmark
  pagerank: PageRank
  pi: Pi Estimator
  sssp: Single Source Shortest Path
  test: Serialize Printing Test
edward@domU-12-31-39-0C-7D-41:/usr/local/hama-0.3.0-incubating$ bin/hama jar hama-
examples-0.3.0-incubating.jar pi
11/12/25 11:48:11 INFO bsp.BSPJobClient: Running job: job_201112251143_0001
11/12/25 11:48:14 INFO bsp.BSPJobClient: Current supersteps number: 0
11/12/25 11:48:17 INFO bsp.BSPJobClient: Current supersteps number: 1
11/12/25 11:48:20 INFO bsp.BSPJobClient: The total number of supersteps: 1
Estimated value of PI is 3.147866666666667
Job Finished in 9.635 seconds
```

And shut down the cluster with:

```
% bin/whirr destroy-cluster --config recipes/hama-ec2.properties
```

The various options are explained in more detail in the Whirr website

Release Migration

This section covers the steps required to update the Hama version to 0.5 release. Please check the compatibility matrix picked from our Wiki.

Apache Hama Release Version	Apache Hadoop Release Version	Java	Known Compatibility Problems
0.2.0-incubating	0.20.2	1.6	None
0.3.0-incubating	0.20.2	1.6	None
0.4.0-incubating	0.20.2	1.6	None
0.5.0-incubating	1.0.0	1.6	None

Please refer to related sections in the document to get more information on the commands directed below.

Migration to Hama incubating version 0.5

Migrating to 0.5 from older version requires you to first upgrade the HDFS running on your cluster. Detailed instructions to upgrade HDFS on your cluster is explained here - http://wiki.apache.org/hadoop/Hadoop_Upgrade.

- Stop Apache Hama

```
hduser@hadoop-1:~$ /opt/hama/bin/stop-bspd.sh
```

- Then stop HDFS service. It is recommended that you run fsck command and admin -report command to verify if everything is fine and make a copy of the results. Please refer the aforesaid Wiki page for further details.

```
hduser@hadoop-1:~$/opt/hadoop/bin/stop-dfs.sh
```

- Download and rename/move the new hadoop installation as /opt/hadoop on all the nodes. Please refer to single node HDFS installation section for details. Configure the nodes with the same values. In the same way, download and install the Hama release as explained in the local mode Hama installation. Update the hama-site.xml file and groomservers file from the older version files. If you are renaming the releases, it is recomended that you keep the backup of the release until the upgradation is complete and tested.

```
user@virtualbox:~$  sudo cd /opt
user@virtualbox:~$  sudo tar xzvf hadoop-1.0.0.tar.gz

user@virtualbox:~$  sudo ln -s hadoop-1.0.0  hadoop
OR
user@virtualbox:~$  sudo mv hadoop-1.0.0  hadoop

user@virtualbox:~$  sudo chown -R hduser:hadoop hadoop-1.0.0
user@virtualbox:~$  sudo chown -R hduser:hadoop hadoop

user@virtualbox:~$ sudo tar xzvf hama-dist-0.5.0-incubating.tar.gz
user@virtualbox:~$ sudo ln -s hama-0.5.0-incubating hama
OR
user@virtualbox:~$ sudo mv hama-0.5.0-incubating hama
```

- Now start the the HDFS with the upgrade option.

```
hduser@hadoop-1:~$/opt/hadoop/bin/start-dfs.sh -upgrade
```

-Run the report command and wait till the HDFS cluster reports its status as OK.

```
hduser@hadoop-1:~$/opt/hadoop/bin/hadoop dfsadmin -report
```

- Now start Apache Hama processes.

```
hduser@hadoop-1:~$/opt/hama/bin/start-bspd.sh
```

- Verify your installation is working fine and you are able to work on the files on HDFS after upgradation. If everything looks good, you can get rid of the previous version backup by command.

```
hduser@hadoop-1:~$ hadoop dfsadmin -finalizeUpgrade
```