GettingStartedWithHadoop

Note: for the 1.0.x series of Hadoop the following articles will probably be easiest to follow:

- Hadoop Single-Node Setup
- Hadoop Cluster Setup

The below instructions are primarily for the 0.2x series of Hadoop.

Downloading and installing Hadoop

Hadoop can be downloaded from one of the Apache download mirrors. You may also download a nightly build or check out the code from subversion and build it with Ant. Select a directory to install Hadoop under (let's say `/foo/bar/hadoop-install`) and untar the tarball in that directory. A directory corresponding to the version of Hadoop downloaded will be created under the `/foo/bar/hadoop-install` directory. For instance, if version 0.21.0 of Hadoop was downloaded untarring as described above will create the directory `/foo/bar/hadoop-install/hadoop-0.21.0`. The examples in this document assume the existence of an environment variable `$HADOOP_INSTALL` that represents the path to all versions of Hadoop installed. In the above instance `HADOOP_INSTALL=/foo/bar/hadoop-install`. They further assume the existence of a symlink named `hadoop` in `$HADOOP_INSTALL` that points to the version of Hadoop being used. For instance, if version 0.21.0 is being used then `$HADOOP_INSTALL/hadoop -> hadoop-0.21.0`. All tools used to run Hadoop will be present in the directory `$HADOOP_INSTALL/hadoop/bin`. All configuration files for Hadoop will be present in the directory `$HADOOP_INSTALL/hadoop/conf`.

Startup scripts

The `$HADOOP_INSTALL/hadoop/bin` directory contains some scripts used to launch Hadoop DFS and Hadoop Map/Reduce daemons. These are:

- `start-dfs.sh` - Starts the Hadoop DFS daemons, the namenode and datanodes. Use this before `start-mapred.sh`
- `stop-dfs.sh` - Stops the Hadoop DFS daemons.
- `start-mapred.sh` - Starts the Hadoop Map/Reduce daemons, the jobtracker and tasktrackers.
- `stop-mapred.sh` - Stops the Hadoop Map/Reduce daemons.
- `start-all.sh` - Starts all Hadoop daemons, the namenode, datanodes, the jobtracker and tasktrackers. Deprecated; use `start-dfs.sh` then `start-mapred.sh`
- `stop-all.sh` - Stops all Hadoop daemons. Deprecated; use `stop-mapred.sh` then `stop-dfs.sh`

It is also possible to run the Hadoop daemons as Windows Services using the Java Service Wrapper (download this separately). This still requires Cygwin to be installed as Hadoop requires its df command. See the following JIRA issues for details:

- https://issues.apache.org/jira/browse/HADOOP-1525
- https://issues.apache.org/jira/browse/HADOOP-1526

Configuration files

Hadoop Cluster Setup/Configuration contains a description of Hadoop configuration for 0.21.0. The information on this wiki page is not current. See also QuickStart which is current for 0.21.0.

The `$HADOOP_INSTALL/hadoop/conf` directory contains some configuration files for Hadoop. These are:

- `hadoop-env.sh` - This file contains some environment variable settings used by Hadoop. You can use these to affect some aspects of Hadoop daemon behavior, such as where log files are stored, the maximum amount of heap used etc. The only variable you should need to change in this file is `JAVA_HOME`, which specifies the path to the Java 1.5.x installation used by Hadoop.
- `slaves` - This file lists the hosts, one per line, where the Hadoop slave daemons (datanodes and tasktrackers) will run. By default this contains the single entry `localhost`
- `hadoop-default.xml` - This file contains generic default settings for Hadoop daemons and Map/Reduce jobs. **Do not modify this file.**
- `mapred-default.xml` - This file contains site specific settings for the Hadoop Map/Reduce daemons and jobs. The file is empty by default. Putting configuration properties in this file will override Map/Reduce settings in the `hadoop-default.xml` file. Use this file to tailor the behavior of Map/Reduce on your site.
- `hadoop-site.xml` - This file contains site specific settings for all Hadoop daemons and Map/Reduce jobs. This file is empty by default. Settings in this file override those in `hadoop-default.xml` and `mapred-default.xml`. This file should contain settings that must be respected by all servers and clients in a Hadoop installation, for instance, the location of the namenode and the jobtracker.

More details on configuration can be found on the HowToConfigure page.

Setting up Hadoop on a single node

This section describes how to get started by setting up a Hadoop cluster on a single node. The setup described here is an HDFS instance with a namenode and a single datanode and a Map/Reduce cluster with a jobtracker and a single tasktracker. The configuration procedures described in Basic Configuration are just as applicable for larger clusters.

Basic Configuration

Take a pass at putting together basic configuration settings for your cluster. Some of the settings that follow are required, others are recommended for more straightforward and predictable operation.

- **Hadoop Environment Settings** - Ensure that `JAVA_HOME` is set in `hadoop-env.sh` and points to the Java installation you intend to use. You can set other environment variables in `hadoop-env.sh` to suit your requirments. Some of the default settings refer to the variable `HADOOP_HOME`. The value of `HADOOP_HOME` is automatically inferred from the location of the startup scripts. `HADOOP_HOME` is the parent directory of the `bin` directory that holds the Hadoop scripts. In this instance it is `$HADOOP_INSTALL/hadoop`.
- **Jobtracker and Namenode settings** - Figure out where to run your namenode and jobtracker. Set the variable `fs.default.name` to the Namenode's intended host:port. Set the variable `mapred.job.tracker` to the jobtrackers intended host:port. These settings should be in `hadoop-site.xml`. You may also want to set one or more of the following ports (also in `hadoop-site.xml`):
  - `dfs.datanode.port`
  - `dfs.info.port`
  - `mapred.job.tracker.info.port`
  - `mapred.task.tracker.output.port`
  - `mapred.task.tracker.report.port`
- **Data Path Settings** - Figure out where your data goes. This includes settings for where the namenode stores the namespace checkpoint and the edits log, where the datanodes store filesystem blocks, storage locations for Map/Reduce intermediate output and temporary storage for the HDFS client. The default values for these paths point to various locations in `/tmp`. While this might be ok for a single node installation, for larger clusters storing data in `/tmp` is not an option. These settings must also be in `hadoop-site.xml`. It is important for these settings to be present in `hadoop-site.xml` because they can otherwise be overridden by client configuration settings in Map/Reduce jobs. Set the following variables to appropriate values:
  - `dfs.name.dir`
  - `dfs.data.dir`
  - `dfs.client.buffer.dir`
  - `mapred.local.dir`

An example of a `hadoop-site.xml` file:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/tmp/hadoop-${user.name}</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
</property>
<property>
  <name>mapred.job.tracker</name>
  <value>hdfs://localhost:54311</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>8</value>
</property>
<property>
  <name>mapred.child.java.opts</name>
  <value>-Xmx512m</value>
</property>
</configuration>
```

Formatting the Namenode

The first step to starting up your Hadoop installation is formatting the Hadoop filesystem, which is implemented on top of the local filesystems of your cluster. You need to do this the first time you set up a Hadoop installation. **Do not** format a running Hadoop filesystem, this will cause all your data to be erased. Before formatting, ensure that the `dfs.name.dir` directory exists. If you just used the default, then `mkdir -p /tmp/hadoop-username/dfs/name` will create the directory. To format the filesystem (which simply initializes the directory specified by the `dfs.name.dir` variable), run the command:

```
% $HADOOP_INSTALL/hadoop/bin/hadoop namenode -format
```

If asked to [re]format, you must reply Y (not just y) if you want to reformat, else Hadoop will abort the format.

Starting a Single node cluster

Run the command:

```
% $HADOOP_INSTALL/hadoop/bin/start-all.sh
```

This will startup a Namenode, Datanode, Jobtracker and a Tasktracker on your machine.

Stopping a Single node cluster

Run the command

```
% $HADOOP_INSTALL/hadoop/bin/stop-all.sh
```

to stop all the daemons running on your machine.

Separating Configuration from Installation

In the example described above, the configuration files used by the Hadoop cluster all lie in the Hadoop installation. This can become cumbersome when upgrading to a new release since all custom config has to be re-created in the new installation. It is possible to separate the config from the install. To do so, select a directory to house Hadoop configuration (let's say `/foo/bar/hadoop-config`. Copy all conf files to this directory. You can either set the `HADOOP_CONF_DIR` environment variable to refer to this directory or pass it directly to the Hadoop scripts with the `--config` option. In this case, the cluster start and stop commands specified in the above two sub-sections become
`% $HADOOP_INSTALL/hadoop/bin/start-all.sh --config /foo/bar/hadoop-config` and
`% $HADOOP_INSTALL/hadoop/bin/stop-all.sh --config /foo/bar/hadoop-config`.
Only the absolute path to the config directory should be passed to the scripts.

Starting up a larger cluster

- Ensure that the Hadoop package is accessible from the same path on all nodes that are to be included in the cluster. If you have separated configuration from the install then ensure that the config directory is also accessible the same way.
- Populate the `slaves` file with the nodes to be included in the cluster. One node per line.
- Follow the steps in the *Basic Configuration* section above.
- Format the Namenode
- Run the command `% $HADOOP_INSTALL/hadoop/bin/start-dfs.sh` on the node you want the Namenode to run on. This will bring up HDFS with the Namenode running on the machine you ran the command on and Datanodes on the machines listed in the slaves file mentioned above.
- Run the command `% $HADOOP_INSTALL/hadoop/bin/start-mapred.sh` on the machine you plan to run the Jobtracker on. This will bring up the Map/Reduce cluster with Jobtracker running on the machine you ran the command on and Tasktrackers running on machines listed in the slaves file.
- The above two commands can also be executed with a `--config` option.

Stopping the cluster

- The cluster can be stopped by running `% $HADOOP_INSTALL/hadoop/bin/stop-mapred.sh` and then `% $HADOOP_INSTALL/hadoop/bin/stop-dfs.sh` on your Jobtracker and Namenode respectively. These commands also accept the `--config` option.