Lab Lecture #1

HADOOP Prerequisites

- GNU/Linux computerRequired Software
- Java 1.6 SDK installed
- ssh must be installed and sshd must be running

HADOOP Preparation

- Create the hadoop user account and login as hadoop user
- Download Hadoop-0.20.2.tar.gz in your home dir
- Unpack the downloaded Hadoop distribution in you home dir
- Check that you can ssh to the localhost without a passphrase:
  ```
  hadoop@localhost$ ssh localhost
  ```
  If you cannot ssh to localhost without a passphrase, execute the following commands:
  ```
  hadoop@localhost$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
  hadoop@localhost$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
  ```
- Move to the hadoop distribution dir:
  ```
  hadoop@localhost$ cd $HOME/hadoop-0.20.2
  ```
- Create the HADOOP_HOME environment variable:
  ```
  hadoop@localhost$ export HADOOP_HOME=`pwd`
  ```
- Edit the file conf/hadoop-env.sh to define at least JAVA_HOME to be the root of your Java installation
- Try the following command:
  ```
  hadoop@localhost$ bin/hadoop
  ```
  This will display the usage documentation for the hadoop script. Now you are ready to start your Hadoop cluster in one of the three supported modes: Local (Standalone) Mode, Pseudo-Distributed Mode, Fully-Distributed Mode.

HADOOP Verification

- By default, Hadoop is configured to run in a non-distributed mode (*standalone mode*), as a single Java process. This is useful for debugging.
- The following example copies the unpacked conf directory to use as input and then finds and displays every match of the given regular expression. Output is written to the given output directory.
  ```
  hadoop@localhost$ mkdir input
  hadoop@localhost$ cp conf/*.xml input
  hadoop@localhost$ bin/hadoop jar hadoop-0.20.2-examples.jar grep \
    input output 'dfs[a-z.]+'
  hadoop@localhost$ cat output/*
  ```
- Clean up:
  ```
  hadoop@localhost$ rm –rf input output
  ```
- Hadoop can also be run on a single-node in a *pseudo-distributed mode* where each Hadoop daemon runs in a separate Java process.
- Edit the conf/core-site.xml file:
  ```
  <configuration>
    <property>
      <name>fs.default.name</name>
      <value>hdfs://localhost:9000</value>
    </property>
  </configuration>
  ```
- Edit the conf/hdfs-site.xml file:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

- Edit the `conf/mapred-site.xml` file:

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9001</value>
  </property>
</configuration>
```

- Format a new distributed-filesystem:

```
hadoop@localhost$ bin/hadoop namenode -format
```

- Start the hadoop daemons:

```
hadoop@localhost$ bin/start-all.sh
```

- Browse the web interface for the *NameNode* and the *JobTracker*; by default they are available at:

  NameNode - `http://localhost:50070`
  JobTracker - `http://localhost:50030`

- Copy the input files into the distributed filesystem:

```
hadoop@localhost$ bin/hadoop fs -put conf input
```

- Run some of the examples provided:

```
hadoop@localhost$ bin/hadoop jar hadoop-*-examples.jar grep \
  input output 'dfs[a-z.]+'
```

- Copy the output files from the distributed filesystem to the local filesytem and examine them:

```
hadoop@localhost$ bin/hadoop fs -get output output
hadoop@localhost$ cat output/*
```

- Clean up:

```
hadoop@localhost$ rm –r output
hadoop@localhost$ bin/hadoop fs –rmr input output
```

- When you're done, stop the daemons with:

```
hadoop@localhost$ bin/stop-all.sh
```