How to Start Hadoop in Pseudo-distributed Mode

Reference: http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SingleCluster.html

TA Email: caijinjin4@sjtu.edu.cn

---

1. Download Hadoop from mirror site.

   https://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/

   TA used the 2.7.3 version.

2. Install Java environment and ssh.

   Ubuntu:

   `sudo apt-get install openjdk-7-jdk`

   `sudo apt-get install openssh-server`

   CentOS:

   `sudo yum install java-1.7.0-openjdk`

   `sudo yum install openssh-server`

3. Get the path of jvm.

   In ubuntu 14.04, `openjdk` will be installed under the directory: `/usr/lib/jvm/java-7-openjdk-amd64`

   You can also use command `find /usr/ -name jvm -type d` to find the path of jvm in `/usr` directory. Anyway, you need to get the path of jvm at first.

4. Setup passphraseless ssh.

   Now check that you can ssh to the localhost without a passphrase: `ssh localhost` If you cannot ssh to localhost without a passphrase, execute the following commands: `ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa` `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys` `chmod 0600 ~/.ssh/authorized_keys`

5. Setup Hadoop Local (Standalone) Mode.

   - Unpack the downloaded Hadoop distribution.
   - Edit the file `/etc/hadoop/hadoop-env.sh`

     ```
     # set to the root of your Java installation
     export JAVA_HOME=/path/to/jvm_install
     # set to the configuration path of your hadoop installation
     export HADOOP_CONF_DIR=/path/to/hadoop_install/etc/hadoop/
     ```

   - Try the command `./bin/hadoop version` , this will display version information of hadoop.

   - By default, Hadoop is configured to run in a non-distributed mode, as a single Java process. This is useful for debugging. Try the following command to test the wordcount example.

```
#create input folder in the hadoop root directory
mkdir input
#create a file in the input folder, input something in this file, such as "hello world"
#execute wordcount
./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-example-2.7.3.jar wordcount input output
#check the results
cat output/*
```

6. Setup Pseudo-Distributed mode.

    - Edit the file `/etc/hadoop/core-site.xml`

    ```
    <configuration>
        <property>
            <name>fs.defaultFS</name>
            <value>hdfs://localhost:9000</value>
        </property>
    </configuration>
    ```

    - Edit the file `/etc/hadoop/hdfs-site.xml`

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

    - Format the HDFS filesystem

    `./bin/hdfs namenode -format`

    - Start namenode and yarn daemon

    `./sbin/start-all.sh`

    - You can execute command `jps` to check the running java processes, then it will display as below

    ```
    process_id NodeManager
    process_id Jps
    process_id DataNode
    process_id ResourceManager
    process_id SecondaryNamenode
    ```

    - You can also browse the web `http://localhost:50070` to check the status of namenode

    - Make the HDFS directories required to execute MapReduce jobs, and copy the input file used before into the distributed filesystem

    `./bin/hdfs dfs -mkdir /input`

    `./bin/hdfs dfs -put input/file /input`

    - Run wordcount

    `./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-example-2.7.3.jar wordcount /input output`

    - Get the results

```
./bin/hdfs dfs -get output output
```

```
cat output/*
```

or

```
./bin/hdfs dfs -cat output/*
```

- When you’re done, stop the daemons with `./sbin/stop-dfs.sh`