Data Analytics

CloudSuite1.0 Benchmark Suite

The data analytics benchmark relies on using the Hadoop MapReduce framework to perform machine learning analysis on large-scale datasets. Apache provides an machine learning library, Mahout, that is designed to run with Hadoop and perform large-scale data analytics.

Prerequisite Software Packages

1. Hadoop
   We recommended version 20.2, that was used and tested in our environment.
2. Mahout: Scalable machine learning and data mining library
3. Maven: A project management tool (required for installing Mahout)
4. Java JDK

Download the Data Analytics benchmark

Setting up Hadoop

Hadoop contains two main components: the HDFS file system and the MapReduce infrastructure. HDFS is the distributed file system that stores the data on several nodes of the cluster called the data nodes. MapReduce is the component in charge of executing computational tasks on top of the data stored in the distributed file system.

Installing Hadoop on a single node:

1. Create a Hadoop user (Note: This requires root privileges and the commands can be different in various Linux distributions such as useradd vs adduser):
   - sudo groupadd hadoop
   - sudo useradd -g hadoop hadoop
   - sudo passwd hadoop (to setup the password)
2. Preparing SSH
   - su - hadoop
   - ssh-keygen -t rsa -p "" (press enter for any prompts)
   - cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
   - ssh localhost (answer yes to he prompt)
3. Unpack the downloaded Hadoop. Then, chown -R hadoop:hadoop hadoop-0.20.2
4. Update the *.bashrc* file with the following:
   - export HADOOP_HOME=/path/to/hadoop/ (e.g., /home/username/hadoop-0.20.0/).
   - export JAVA_HOME=/pth/to/jdk (e.g., /usr/lib/jvm/java-6-sun).
5. In the *HADOOP_HOME* directory (the main Hadoop folder) make the following modifications to the configuration files:
   - In *conf/core-site.xml* add:
     <property>
     <name> hadoop.tmp.dir</name>
     <value> /path/to/dir</value>
     <description> A base for other temporary directories. Set it to a directory that you have write permissions to.</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>

- In *conf/mapred-site.xml* add
  <property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker runs at. If "local", then jobs are run in-process as a single map and reduce task. </description>
  </property>
  Note that there are other parameters that should be tuned to fit the needs of your environment, such as mapred.tasktracker.map.tasks.maximum, mapred.tasktracker.reduce.tasks.maximum, mapred.map.tasks, mapred.reduce.tasks. For now we will rely on the defaults.
- In *conf/hdfs-site.xml* add
  <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>/path/to/namespace/dir (dir of your choice)</value>
  </property>
  <property>
  <name>dfs.data.dir</name>
  <value>/path/to/file system/dir (dir of your choice)</value>
  </property>

6. Format the HDFS filesystem:
   $HADOOP_HOME/bin/hadoop namenode -format
7. To start Hadoop:
   $HADOOP_HOME/bin/start-all.sh (you can start hdfs then mapreduce by start-dfs.sh and start-mapred.sh respectively)
8. To stop Hadoop:
   $HADOOP_HOME/bin/stop-all.sh (or stop-dfs.sh and stop-mapred.sh)

This link provides a step-by-step guide to install Hadoop on a single node. To install Hadoop on multiple nodes, you need to define the slave nodes in conf/slaves and maintain the node that you already installed Hadoop on as the master. Then, start all the nodes with start-all.sh from the master node. This link provides more details on installing Hadoop on multiple nodes.

Installing Mahout

1. Download Mahout (already included in the analytics benchmark package)
2. For installing Mahout:
   mvn install (from MAHOUT_HOME folder)
   This will build the core package and the Mahout examples. You need to instal Maven to be able to perform this step.
3. For further information regarding Mahout installation please refer to Mahout website.

Running the classification benchmark

This benchmark runs the Wikipedia Bayes example as provided by Mahout package:

1. This benchmark uses a Wikipedia data set of ~30GB size. You can download it at the following address
2. Unzip the bz2 file to get *enwiki-latest-pages-articles.xml*.
3. Create the directory $MAHOUT_HOME/examples/temp and copy *enwiki-latest-pages-articles.xml* into this directory.
4. The next step requires to chunk the data into pieces. For this step run the following:
   $MAHOUT_HOME/bin/mahout wikipediaXMLSplitter -d $MAHOUT_HOME/examples/temp/enwiki-latest-pages-articles.xml -o wikipedia/chunks -c 64
5. After creating the chunks, verify that the chunks were written successfully into HDFS (look for chunk-*.xml files):
   hadoop dfs -ls wikipedia/chunks
6. Before running the benchmark, you need to create the countries based Split of the Wikipedia dataset:

   $MAHOUT_HOME/bin/mahout wikipediaDataSetCreator -i wikipedia/chunks -o wikipediainput -c $MAHOUT_HOME/examples/src/test/resources/country.txt

   Check if the wikipediainput folder was created, you should see *part-r-00000* file inside the *wikipediainput directory*
   hadoop fs -ls wikipediainput

7. Build the classifier model. After completion, the model will be available in HDFS, under the *wikipediamodel* folder:

   $MAHOUT_HOME/bin/mahout trainclassifier -i wikipediainput -o wikipediamodel

8. Test the classifier model:

$MAHOUT_HOME/bin/mahout testclassifier -m wikipediamodel -d wikipediainput

By default the testing phase is running as a standalone application. If you want to use MapReduce for parallelizing the testing phase, use the --method option. To list all the options available use:

$MAHOUT_HOME/bin/mahout testclassifier --help

9. You can measure the time to build the classifier, as well as the time to test it using the *time* system call.