Hadoop (pseudo-distributed) installation and configuration

1. Operating systems.
Linux-based systems are preferred, e.g., Ubuntu or Mac OS X.

2. Install Java.
For Linux, you should download JDK 8 under the section of Java SE Development Kit 8u11 from the website: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html. If your machine is 64-bit, download: jdk-8u11-linux-x64.tar.gz. If your machine is 32-bit, download: jdk-8u11-linux-i586.tar.gz.

For the Mac OS X, you can download: jdk-8u11-macosx-x64.dmg.

3. Set up JAVA_HOME.
Once the JDK is installed, you can define JAVA_HOME system variable by adding the following line into the file .bash_profile (for Mac) or.bashrc (for Ubuntu) under your home directory. If the file does not exist, create a new one.
Below is for Mac environment:

```
$vi .bash_profile
  export JAVA_HOME=$(/usr/libexec/java_home)
```

Then save and exit the file.

```
$source .bash_profile
$echo $JAVA_HOME
/Library/Java/JavaVirtualMachines/1.8.0_11.jdk/Contents/Home
```

Below is for Ubuntu environment:

```
$vi .bashrc
```

At the end of the file, add the follow lines:

```
JAVA_HOME=/usr/lib/jvm/java-8-sun
export JAVA_HOME
PATH=$PATH:$JAVA_HOME
export PATH
```

Then save and exit the file.

```
$echo $JAVA_HOME
/usr/lib/jvm/java-8-sun
```

4. SSH: set up Remote Desktop and Enabling Self-Login.
For Mac, go to System Preferences→Sharing, check Remote Login. Then go to your home directory under the terminal, do the following steps:

```
$ssh-keygen -t rsa -P ""
$cat .ssh/id_rsa.pub >> .ssh/authorized_keys
```

Now try:

$ssh localhost

Now you should be able to log in without any password. Don't forget to exit the localhost environment by typing exit.

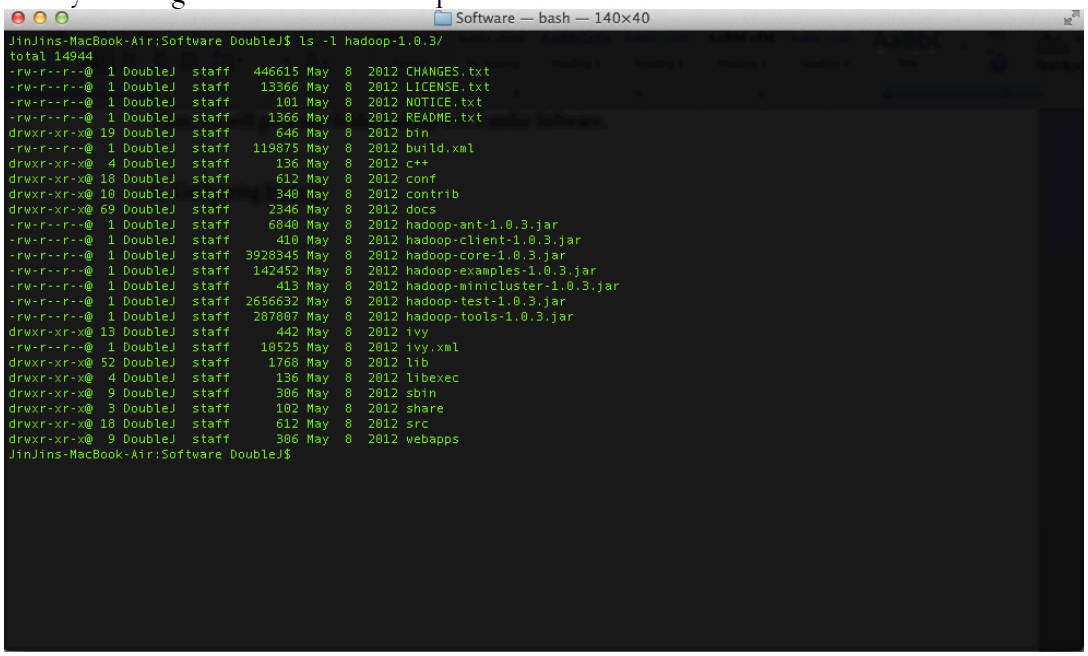
5. Downloading and Unpacking Hadoop.

For learning purpose in this course, we use the stable version of Hadoop. You can try the latest version, which has different Hadoop framework (including resource management, YARN), if you are interested. Download Hadoop 1.0.3 file: hadoop-1.0.3.tar.gz from the website: http://archive.apache.org/dist/hadoop/core/hadoop-1.0.3/.

Unpack the hadoop-1.0.3.tar.gz in the directory of your choice. I place mine in the directory of ~/Software/. Go to ~/Software directory, do the following.

$tar –xzvf hadoop-1.0.3.tar.gz

Then you will get a new folder hadoop-1.0.3 under Software.



6. Configuring Hadoop.

There are 4 files that we want to modify when we configure Hadoop: hadoop-env.sh, hdfs-site.xml, core-site.xml, mapred-site.xml.

```
conf — bash — 140×40
-rw-r--r--@  1 DoubleJ  staff   119875 May  8  2012 build.xml
drwxr-xr-x@  4 DoubleJ  staff      136 May  8  2012 c++
drwxr-xr-x@ 18 DoubleJ  staff      612 May  8  2012 conf
drwxr-xr-x@ 10 DoubleJ  staff      340 May  8  2012 contrib
drwxr-xr-x@ 69 DoubleJ  staff     2346 May  8  2012 docs
-rw-r--r--@  1 DoubleJ  staff     6840 May  8  2012 hadoop-ant-1.0.3.jar
-rw-r--r--@  1 DoubleJ  staff      410 May  8  2012 hadoop-client-1.0.3.jar
-rw-r--r--@  1 DoubleJ  staff  3928345 May  8  2012 hadoop-core-1.0.3.jar
-rw-r--r--@  1 DoubleJ  staff   142452 May  8  2012 hadoop-examples-1.0.3.jar
-rw-r--r--@  1 DoubleJ  staff      413 May  8  2012 hadoop-minicluster-1.0.3.jar
-rw-r--r--@  1 DoubleJ  staff  2656632 May  8  2012 hadoop-test-1.0.3.jar
-rw-r--r--@  1 DoubleJ  staff   287807 May  8  2012 hadoop-tools-1.0.3.jar
drwxr-xr-x@ 13 DoubleJ  staff      442 May  8  2012 ivy
-rw-r--r--@  1 DoubleJ  staff    10525 May  8  2012 ivy.xml
drwxr-xr-x@ 52 DoubleJ  staff     1768 May  8  2012 lib
drwxr-xr-x@  4 DoubleJ  staff      136 May  8  2012 libexec
drwxr-xr-x@  9 DoubleJ  staff      306 May  8  2012 sbin
drwxr-xr-x@  3 DoubleJ  staff      102 May  8  2012 share
drwxr-xr-x@ 18 DoubleJ  staff      612 May  8  2012 src
drwxr-xr-x@  9 DoubleJ  staff      306 May  8  2012 webapps
JinJins-MacBook-Air:Software DoubleJ$ cd hadoop-1.0.3/conf/
JinJins-MacBook-Air:conf DoubleJ$ ls -l
total 152
-rw-r--r--@ 1 DoubleJ  staff  7457 May  8  2012 capacity-scheduler.xml
-rw-r--r--@ 1 DoubleJ  staff   535 May  8  2012 configuration.xsl
-rw-r--r--@ 1 DoubleJ  staff   178 May  8  2012 core-site.xml
-rw-r--r--@ 1 DoubleJ  staff   327 May  8  2012 fair-scheduler.xml
-rw-r--r--@ 1 DoubleJ  staff  2237 May  8  2012 hadoop-env.sh
-rw-r--r--@ 1 DoubleJ  staff  1488 May  8  2012 hadoop-metrics2.properties
-rw-r--r--@ 1 DoubleJ  staff  4644 May  8  2012 hadoop-policy.xml
-rw-r--r--@ 1 DoubleJ  staff   178 May  8  2012 hdfs-site.xml
-rw-r--r--@ 1 DoubleJ  staff  4441 May  8  2012 log4j.properties
-rw-r--r--@ 1 DoubleJ  staff  2033 May  8  2012 mapred-queue-acls.xml
-rw-r--r--@ 1 DoubleJ  staff   178 May  8  2012 mapred-site.xml
-rw-r--r--@ 1 DoubleJ  staff    10 May  8  2012 masters
-rw-r--r--@ 1 DoubleJ  staff    10 May  8  2012 slaves
-rw-r--r--@ 1 DoubleJ  staff  1243 May  8  2012 ssl-client.xml.example
-rw-r--r--@ 1 DoubleJ  staff  1195 May  8  2012 ssl-server.xml.example
-rw-r--r--@ 1 DoubleJ  staff   382 May  8  2012 taskcontroller.cfg
JinJins-MacBook-Air:conf DoubleJ$
```

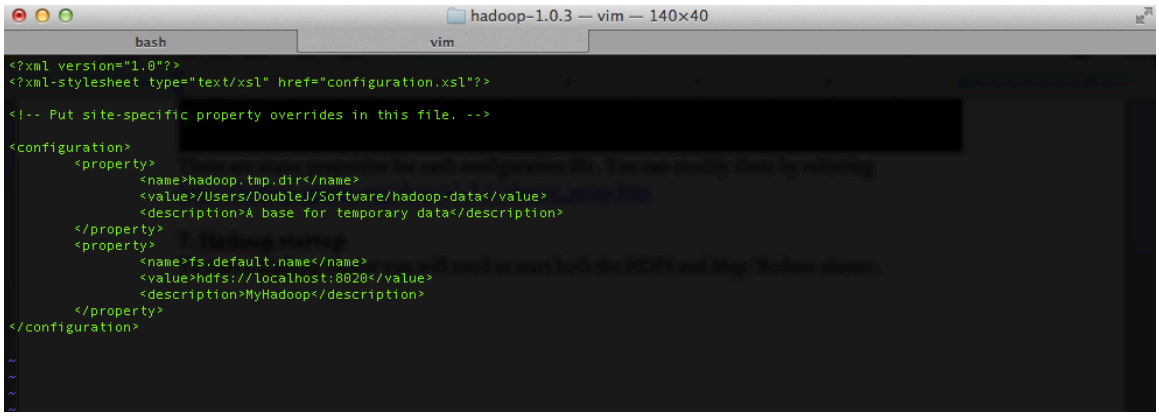
Edit the hadoop-env.sh file

$vi hadoop-env.sh
Uncomment export JAVA_HOME and change to your JAVA_HOME.
Uncomment export HADOOP_HEAPSIZE and change the size of heap depending on your choice.

```
conf — vim — 140×40
# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME.  All others are
# optional.  When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.  Required.
export JAVA_HOME=$(/usr/libexec/java_home)

# Extra Java CLASSPATH elements.  Optional.
# export HADOOP_CLASSPATH=

# The maximum amount of heap to use, in MB. Default is 1000.
export HADOOP_HEAPSIZE=5000

# Extra Java runtime options.  Empty by default.
# export HADOOP_OPTS=-server

# Command specific options appended to HADOOP_OPTS when specified
export HADOOP_NAMENODE_OPTS="-Dcom.sun.management.jmxremote $HADOOP_NAMENODE_OPTS"
export HADOOP_SECONDARYNAMENODE_OPTS="-Dcom.sun.management.jmxremote $HADOOP_SECONDARYNAMENODE_OPTS"
export HADOOP_DATANODE_OPTS="-Dcom.sun.management.jmxremote $HADOOP_DATANODE_OPTS"
export HADOOP_BALANCER_OPTS="-Dcom.sun.management.jmxremote $HADOOP_BALANCER_OPTS"
export HADOOP_JOBTRACKER_OPTS="-Dcom.sun.management.jmxremote $HADOOP_JOBTRACKER_OPTS"
# export HADOOP_TASKTRACKER_OPTS=
# The following applies to multiple commands (fs, dfs, fsck, distcp etc)
# export HADOOP_CLIENT_OPTS

# Extra ssh options.  Empty by default.
# export HADOOP_SSH_OPTS="-o ConnectTimeout=1 -o SendEnv=HADOOP_CONF_DIR"

# Where log files are stored.  $HADOOP_HOME/logs by default.
# export HADOOP_LOG_DIR=${HADOOP_HOME}/logs

# File naming remote slave hosts.  $HADOOP_HOME/conf/slaves by default.
# export HADOOP_SLAVES=${HADOOP_HOME}/conf/slaves

# host:path where hadoop code should be rsync'd from.  Unset by default.
-- INSERT --
```

Edit the core-site.xml file and modify the following properties

```
<property>
        <name>hadoop.tmp.dir </name>
        <value>/Users/DoubleJ/Software/hadoop-data</value>
        <description>A base for temporary data</description>
</property>
<property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:8020</value>
        <description>MyHadoop</description>
</property>
```



Edit the hdfs-site.xml file and modify the following properties

dfs.name.dir

- Path on the local file system where the NameNode stores the namespace and transactions log persistently.
- If this is a comma-delimited list of directories then the name table is replicated in all of the directories, for redundancy.

dfs.data.dir

- Comma separated list of paths on the local filesystem of a DataNode where it should store its blocks.
- If this is a comma-delimited list of directories, then data will be stored in all named directories, typically on different devices.

dfs.replication

- Each data block is replicated for redundancy.

```
conf — vim — 140×40
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>dfs.name.dir</name>
                <value>/Users/DoubleJ/Software/hadoop-data/name</value>
                <description>directory to store namespace for namenode</description>
        </property>
        <property>
                <name>dfs.data.dir</name>
                <value>/Users/DoubleJ/Software/hadoop-data/data</value>
                <description>directory to store block data for datanode</description>
        </property>
        <property>
                <name>dfs.replication</name>
                <value>3</value>
                <description>the number of duplicates for each block</description>
        </property>
</configuration>
~
~
```

Edit mapred-site.xml and modify the following properties

```
<property>
        <name>mapred.job.tracker</name>
        <value>localhost:50300</value>
        <description>MyJobTracker</description>
</property>
```

```
conf — vim — 140×40
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>mapred.job.tracker</name>
                <value>localhost:50300</value>
                <description>MyJobTracker</description>
        </property>
</configuration>
~
~
```

There are many properties for each configuration file. You can modify them by referring http://hadoop.apache.org/docs/r1.2.1/cluster_setup.htm

7. Hadoop startup.

To start a Hadoop cluster you will need to start both the HDFS and Map/Reduce cluster.
Format a new distributed filesystem:

$ bin/hadoop namenode -format

```
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop namenode -format
14/07/28 18:41:39 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = JinJins-MacBook-Air.local/192.168.0.6
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 1.0.3
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.0 -r 1335192; compiled by 'hortonfo' on Tue May  8 2
0:31:25 UTC 2012
************************************************************/
14/07/28 18:41:39 INFO util.GSet: VM type       = 64-bit
14/07/28 18:41:39 INFO util.GSet: 2% max memory = 88.89 MB
14/07/28 18:41:39 INFO util.GSet: capacity      = 2^23 = 8388608 entries
14/07/28 18:41:39 INFO util.GSet: recommended=8388608, actual=8388608
14/07/28 18:41:40 INFO namenode.FSNamesystem: fsOwner=DoubleJ
14/07/28 18:41:40 INFO namenode.FSNamesystem: supergroup=supergroup
14/07/28 18:41:40 INFO namenode.FSNamesystem: isPermissionEnabled=true
14/07/28 18:41:40 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
14/07/28 18:41:40 INFO namenode.FSNamesystem: isAccessTokenEnabled=false accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
14/07/28 18:41:40 INFO namenode.NameNode: Caching file names occuring more than 10 times
14/07/28 18:41:40 INFO common.Storage: Image file of size 113 saved in 0 seconds.
14/07/28 18:41:40 INFO common.Storage: Storage directory /Users/DoubleJ/Software/hadoop-data/name has been successfully formatted.
14/07/28 18:41:40 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at JinJins-MacBook-Air.local/192.168.0.6
************************************************************/
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$
```

In addition, the Hadoop system automatically creates some directories as you specified during the configuration period.

```
JinJins-MacBook-Air:Software DoubleJ$ ls -l
total 1898608
-rw-r--r--   1 DoubleJ  staff  972087296 Jul 24 13:16 SW_DVD5_Office_Mac_Standard_2011_English_MLF_X16-99088.ISO
drwxr-xr-x@ 28 DoubleJ  staff        952 Jul 28 18:42 hadoop-1.0.3
drwxr-xr-x   6 DoubleJ  staff        204 Jul 28 18:42 hadoop-data
JinJins-MacBook-Air:Software DoubleJ$ ls -l hadoop-data/
total 0
drwxr-xr-x  8 DoubleJ  staff  272 Jul 28 18:45 data
drwxr-xr-x  3 DoubleJ  staff  102 Jul 28 18:42 dfs
drwxr-xr-x  3 DoubleJ  staff  102 Jul 28 18:42 mapred
drwxr-xr-x  6 DoubleJ  staff  204 Jul 28 18:45 name
JinJins-MacBook-Air:Software DoubleJ$
```

To start Hadoop, do the following step:

$./bin/start-all.sh

```
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop namenode -format
14/07/28 18:41:39 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = JinJins-MacBook-Air.local/192.168.0.6
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 1.0.3
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.0 -r 1335192; compiled by 'hortonfo' on Tue May  8 2
0:31:25 UTC 2012
************************************************************/
14/07/28 18:41:39 INFO util.GSet: VM type       = 64-bit
14/07/28 18:41:39 INFO util.GSet: 2% max memory = 88.89 MB
14/07/28 18:41:39 INFO util.GSet: capacity      = 2^23 = 8388608 entries
14/07/28 18:41:39 INFO util.GSet: recommended=8388608, actual=8388608
14/07/28 18:41:40 INFO namenode.FSNamesystem: fsOwner=DoubleJ
14/07/28 18:41:40 INFO namenode.FSNamesystem: supergroup=supergroup
14/07/28 18:41:40 INFO namenode.FSNamesystem: isPermissionEnabled=true
14/07/28 18:41:40 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
14/07/28 18:41:40 INFO namenode.FSNamesystem: isAccessTokenEnabled=false accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
14/07/28 18:41:40 INFO namenode.NameNode: Caching file names occuring more than 10 times
14/07/28 18:41:40 INFO common.Storage: Image file of size 113 saved in 0 seconds.
14/07/28 18:41:40 INFO common.Storage: Storage directory /Users/DoubleJ/Software/hadoop-data/name has been successfully formatted.
14/07/28 18:41:40 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at JinJins-MacBook-Air.local/192.168.0.6
************************************************************/
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/start-all.sh
starting namenode, logging to /Users/DoubleJ/Software/hadoop-1.0.3/libexec/../logs/hadoop-DoubleJ-namenode-JinJins-MacBook-Air.local.out
localhost: starting datanode, logging to /Users/DoubleJ/Software/hadoop-1.0.3/libexec/../logs/hadoop-DoubleJ-datanode-JinJins-MacBook-Air.lo
cal.out
localhost: starting secondarynamenode, logging to /Users/DoubleJ/Software/hadoop-1.0.3/libexec/../logs/hadoop-DoubleJ-secondarynamenode-JinJ
ins-MacBook-Air.local.out
starting jobtracker, logging to /Users/DoubleJ/Software/hadoop-1.0.3/libexec/../logs/hadoop-DoubleJ-jobtracker-JinJins-MacBook-Air.local.out
localhost: starting tasktracker, logging to /Users/DoubleJ/Software/hadoop-1.0.3/libexec/../logs/hadoop-DoubleJ-tasktracker-JinJins-MacBook-
Air.local.out
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$
```

To stop Hadoop, do the following step:
$./bin/stop-all.sh

```
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/stop-all.sh
stopping jobtracker
localhost: stopping tasktracker
stopping namenode
localhost: stopping datanode
localhost: stopping secondarynamenode
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$
```

If the Hadoop system is running well, it should be like the following when you type jps in the command terminal.

```
Desktop — bash — 140×40
bash    bash
JinJins-MacBook-Air:Desktop DoubleJ$ jps
624 SecondaryNameNode
1201 Jps
696 JobTracker
792 TaskTracker
428 NameNode
526 DataNode
```

Browse the web interface for the NameNode and the JobTracker; by default they are available at:
NameNode - http://localhost:50070/
JobTracker - http://localhost:50030/

Note: Don't forget to stop Hadoop when you shut down your computer. Every time you have problems with Hadoop, I suggest you delete your temporary data folder: ~/Software/hadoop-data and redo everything from the scratch: reformat NameNode and restart Hadoop. You do not need to reconfigure configuration files.

Note: If you want to leave safemode, do the following:
$./bin/hadoop dfsadmin -safemode leave

Hadoop running example – word count

1. create a folder under hadoop user home directory

For my hadoop configuration, my hadoop home directory is: /user/DoubleJ/
$./bib/hadoop fs –mkdir input
$./bin/hadoop fs –ls

```
hadoop-1.0.3 — bash — 140×40
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -mkdir input
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -ls
Found 1 items
drwxr-xr-x   - DoubleJ supergroup          0 2014-07-28 20:19 /user/DoubleJ/input
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$
```

2. copy local files to remote HDFS

In our pseudo-distributed Hadoop system, both local and remote machines are your laptop. Suppose you have two text files on your local desktop: file1.txt and file2.txt.
file1.txt

Hello World, Bye World!
file2.txt
Hello Hadoop, Goodbye to hadoop.
$./bin/hadoop fs -put ~/Desktop/file1.txt input
$./bin/hadoop fs -put ~/Desktop/file2.txt input
$./bin/hadoop fs -ls input

```
hadoop-1.0.3 — bash — 140×40
bash    bash
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -mkdir input
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -ls
Found 1 items
drwxr-xr-x   - DoubleJ supergroup          0 2014-07-28 20:19 /user/DoubleJ/input
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -put ~/Desktop/file1.txt input
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -put ~/Desktop/file2.txt input
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -ls input
Found 2 items
-rw-r--r--   3 DoubleJ supergroup         24 2014-07-28 20:26 /user/DoubleJ/input/file1.txt
-rw-r--r--   3 DoubleJ supergroup         33 2014-07-28 20:26 /user/DoubleJ/input/file2.txt
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$
```

3. run hadoop job

download the wordcount.jar and put it on your local desktop, then run the following command. Output is the directory which will be automatically generated under your remote HDFS system.

$./bin/hadoop jar ~/Desktop/wordcount.jar WordCount input output
$./bin/hadoop fs -ls output

```
hadoop-1.0.3 — bash — 140×52
bash    bash
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop jar ~/Desktop/AMCIS2014/wordcount.jar WordCount input output
14/07/28 20:45:37 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where app
licable
14/07/28 20:45:37 WARN snappy.LoadSnappy: Snappy native library not loaded
14/07/28 20:45:37 INFO mapred.FileInputFormat: Total input paths to process : 2
14/07/28 20:45:37 INFO mapred.JobClient: Running job: job_201407282013_0003
14/07/28 20:45:38 INFO mapred.JobClient:  map 0% reduce 0%
14/07/28 20:45:52 INFO mapred.JobClient:  map 66% reduce 0%
14/07/28 20:45:58 INFO mapred.JobClient:  map 100% reduce 0%
14/07/28 20:46:01 INFO mapred.JobClient:  map 100% reduce 22%
14/07/28 20:46:10 INFO mapred.JobClient:  map 100% reduce 100%
14/07/28 20:46:15 INFO mapred.JobClient: Job complete: job_201407282013_0003
14/07/28 20:46:15 INFO mapred.JobClient: Counters: 27
14/07/28 20:46:15 INFO mapred.JobClient:   Map-Reduce Framework
14/07/28 20:46:15 INFO mapred.JobClient:     Spilled Records=18
14/07/28 20:46:15 INFO mapred.JobClient:     Map output materialized bytes=129
14/07/28 20:46:15 INFO mapred.JobClient:     Reduce input records=9
14/07/28 20:46:15 INFO mapred.JobClient:     Map input records=2
14/07/28 20:46:15 INFO mapred.JobClient:     SPLIT_RAW_BYTES=309
14/07/28 20:46:15 INFO mapred.JobClient:     Map output bytes=93
14/07/28 20:46:15 INFO mapred.JobClient:     Reduce shuffle bytes=123
14/07/28 20:46:15 INFO mapred.JobClient:     Map input bytes=57
14/07/28 20:46:15 INFO mapred.JobClient:     Reduce input groups=8
14/07/28 20:46:15 INFO mapred.JobClient:     Combine output records=9
14/07/28 20:46:15 INFO mapred.JobClient:     Reduce output records=8
14/07/28 20:46:15 INFO mapred.JobClient:     Map output records=9
14/07/28 20:46:15 INFO mapred.JobClient:     Combine input records=9
14/07/28 20:46:15 INFO mapred.JobClient:     Total committed heap usage (bytes)=565706752
14/07/28 20:46:15 INFO mapred.JobClient:   File Input Format Counters
14/07/28 20:46:15 INFO mapred.JobClient:     Bytes Read=63
14/07/28 20:46:15 INFO mapred.JobClient:   FileSystemCounters
14/07/28 20:46:15 INFO mapred.JobClient:     HDFS_BYTES_READ=372
14/07/28 20:46:15 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=86375
14/07/28 20:46:15 INFO mapred.JobClient:     FILE_BYTES_READ=117
14/07/28 20:46:15 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=67
14/07/28 20:46:15 INFO mapred.JobClient:   File Output Format Counters
14/07/28 20:46:15 INFO mapred.JobClient:     Bytes Written=67
14/07/28 20:46:15 INFO mapred.JobClient:   Job Counters
14/07/28 20:46:15 INFO mapred.JobClient:     Launched map tasks=3
14/07/28 20:46:15 INFO mapred.JobClient:     Launched reduce tasks=1
14/07/28 20:46:15 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=16890
14/07/28 20:46:15 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
14/07/28 20:46:15 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=23294
14/07/28 20:46:15 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
14/07/28 20:46:15 INFO mapred.JobClient:     Data-local map tasks=3
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -ls output
Found 3 items
-rw-r--r--   3 DoubleJ supergroup          0 2014-07-28 20:46 /user/DoubleJ/output/_SUCCESS
drwxr-xr-x   - DoubleJ supergroup          0 2014-07-28 20:45 /user/DoubleJ/output/_logs
-rw-r--r--   3 DoubleJ supergroup         67 2014-07-28 20:46 /user/DoubleJ/output/part-00000
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$
```

To view the output file, you can either view it under HDFS or copy the output file from the HDFS to local system and then view it locally.

a. View it remotely:

$./bin/hadoop fs -cat output/part-00000

```
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -cat output/part-00000
Bye     1
Goodbye 1
Hadoop, 1
Hello   2
World!  1
World,  1
hadoop. 1
to      1
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$
```

b. Copy it from HDFS to local Desktop and then view it locally:

$./bin/hadoop fs -get output ~/Desktop/

$cat ~/Desktop/output/part-00000

```
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ ./bin/hadoop fs -get output ~/Desktop
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$ cat ~/Desktop/output/part-00000
Bye     1
Goodbye 1
Hadoop, 1
Hello   2
World!  1
World,  1
hadoop. 1
to      1
JinJins-MacBook-Air:hadoop-1.0.3 DoubleJ$
```

Note: Don't forget to check http://localhost:50030 and http://localhost:50070 when you run your Hadoop application. If you want to run it again, you have to delete the directory of output from the HDFS first using the command: $./bin/hadoop fs -rmr output