Project 1

jiamin cheng(jic122@pitt.edu)

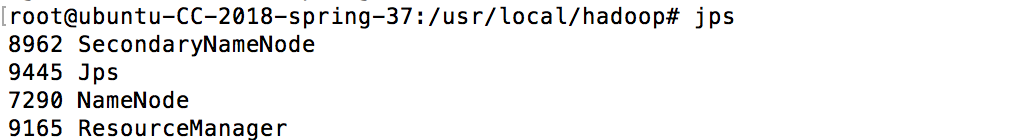
zherui cao (zhc61@pitt.edu)

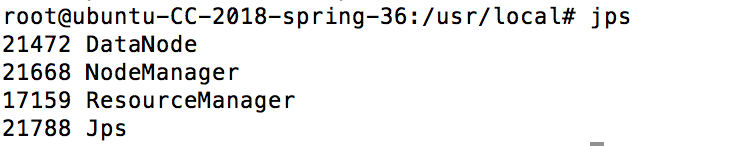
After installing Hadoop on the master machine. We changed the 4 xml files. and copy the whole Hadoop directory to slave machine.

Then we run

|  |
| --- |
| sbin/start-dfs.sh  sbin/start-yarn.sh |

We can see the running processing by enter “jps”





We can see that NameNode is running on master

and DataNode is running on slave

To run the word count

We firstly changed the name of the two machines to master and slave

on slave :

disable the firewall

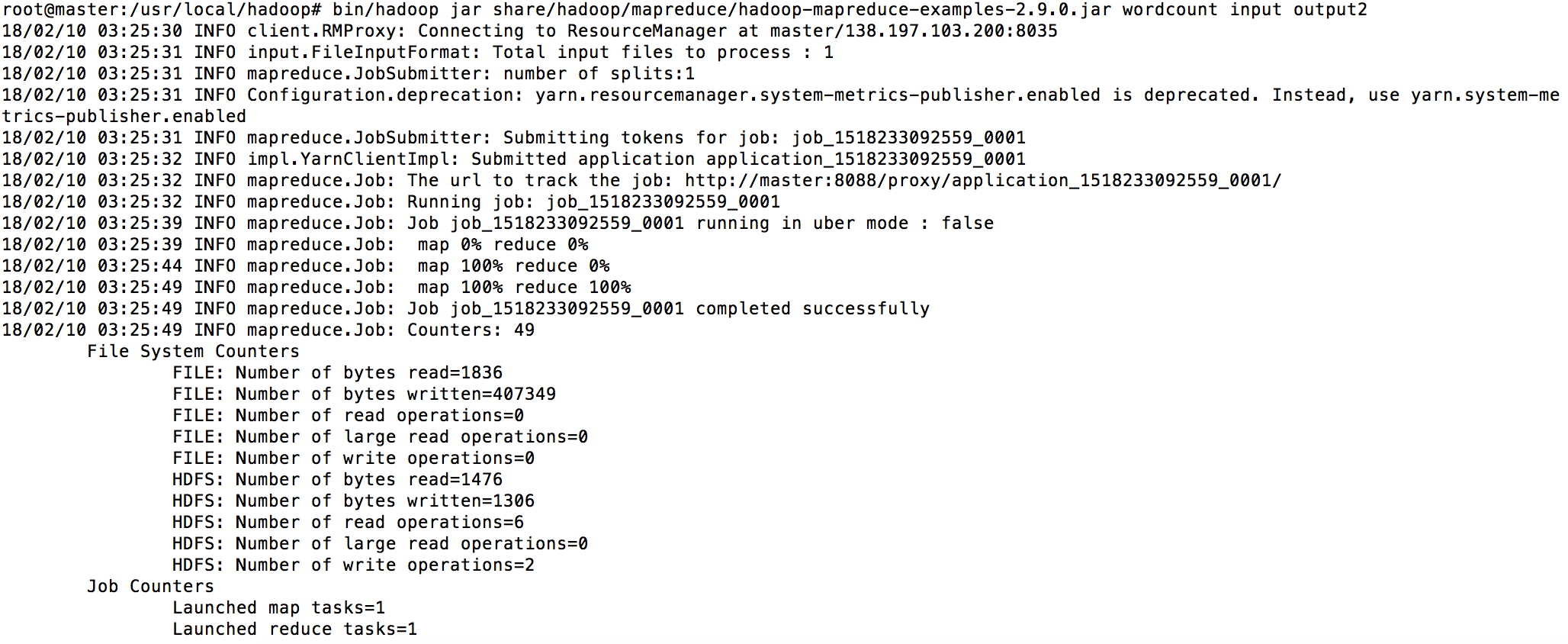
|  |
| --- |
| ufw disable |

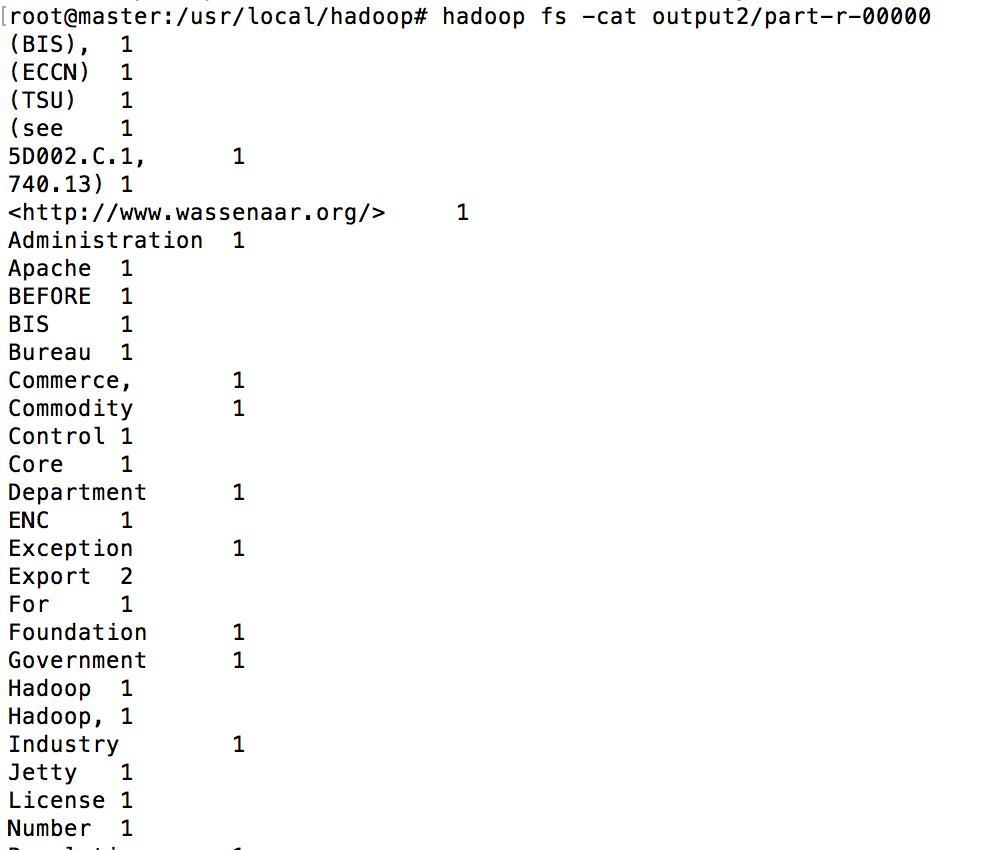
on master:

put file to the hdfs

And the output is

|  |
| --- |
| Hadoop –mkdir input  hadoop fs -put file.txt input  bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.0.jar wordcount input/ output2/ |



we can see the result is 

Part 2

First we use

sudo debootstrap xenial xenial > /dev/null

sudo tar -C xenial -c . | docker import - xenial

docker run xenial cat /etc/lsb-release

to get the ubuntu image of our Ubuntu

Based on this image, we complete our Dockerfile,bootstrap.sh,ssh\_config and other 4 xml files.

Then we

run

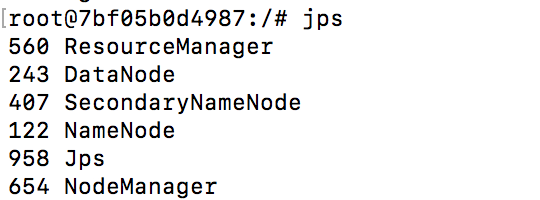
docker build -t mytest .

docker tag mytest cjm8481447/get-start:part3

docker push cjm8481447/get-start:part3

docker run -it cjm8481447/get-start:part3 /etc/bootstrap.sh -bash

we will enter the bash mode

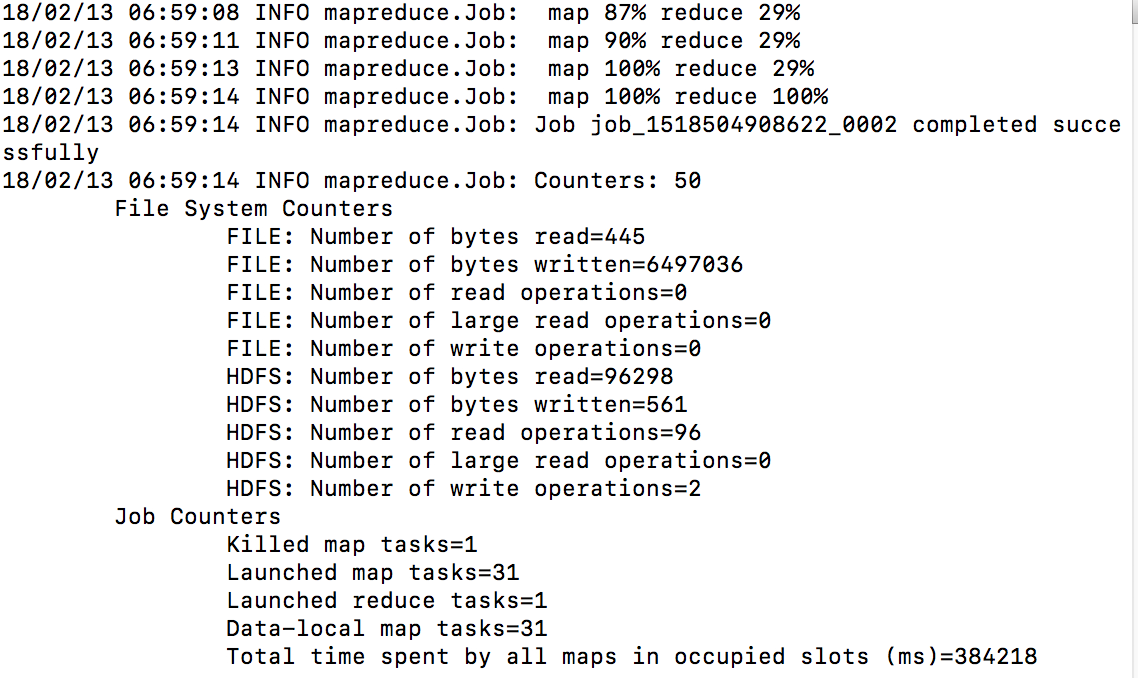


To execute mapreduce, So we run:

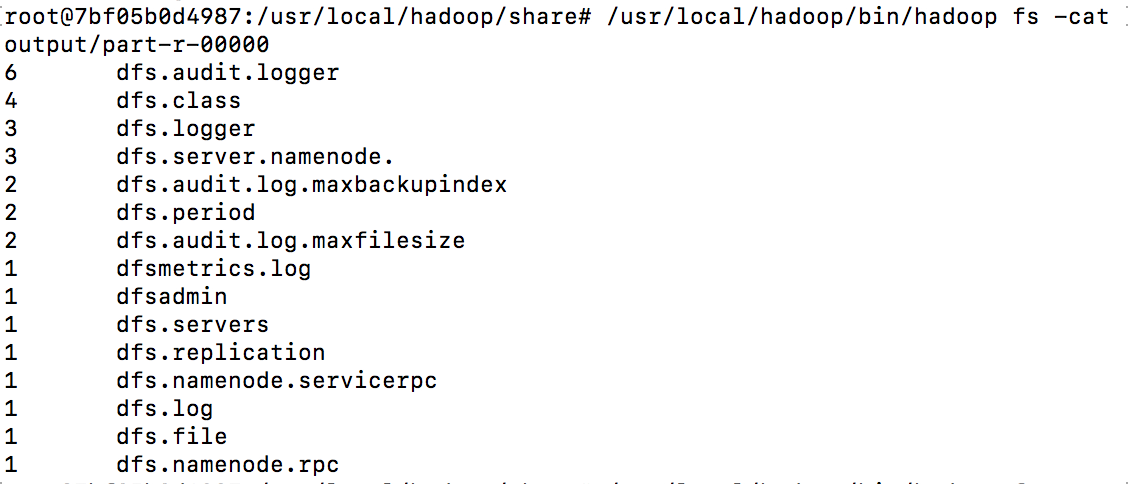
mkdir input

cp etc/hadoop/\*.xml input

bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.0.0.jar grep input output 'dfs[a-z.]+'



Then we run:



Part 3

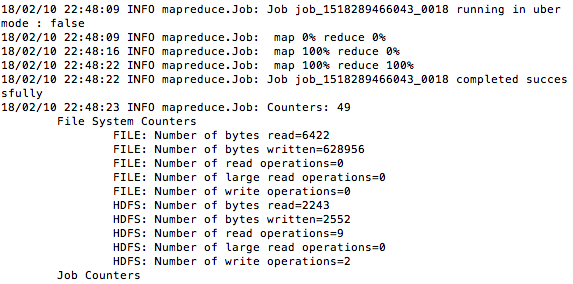
we use Hadoop streaming to run python in Hadoop

we made mapper.py and reducer.py

after finish two files.

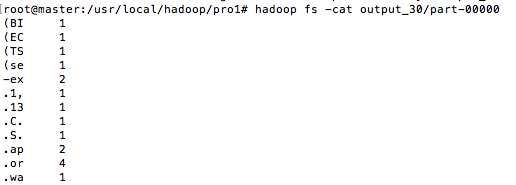
we run

|  |
| --- |
| ../bin/hadoop jar ../hadoop-streaming-2.9.0.jar -files mapper.py,reducer.py -mapper "mapper.py 3" -reducer reducer.py -input input -output output\_30 |



And then

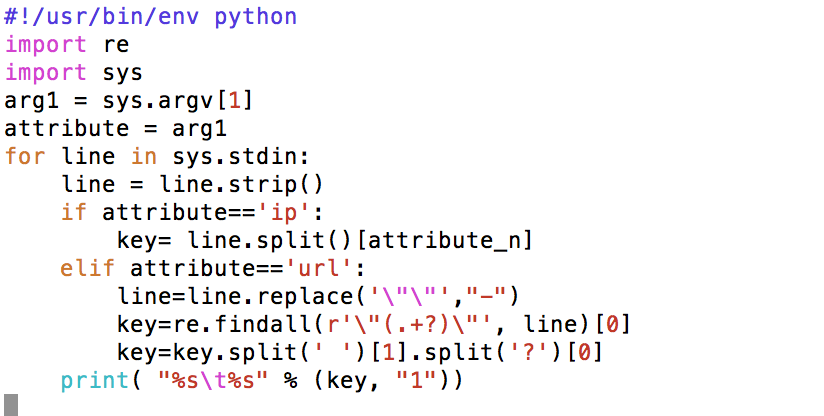
|  |
| --- |
| hadoop fs -cat output\_30/part-00000 |



Part 4

we use the parameter to indicate which column is going to be counted.

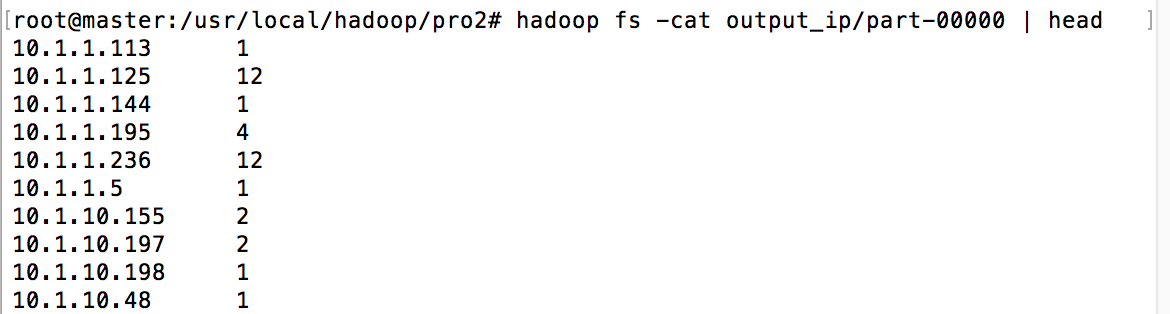
in order to extract information from log file. we use the following code.



**Count IP address:**

|  |
| --- |
| ../bin/hadoop jar ../hadoop-streaming-2.9.0.jar -files mapper.py,reducer.py -mapper "mapper.py ip" -reducer reducer.py -input input\_part4 -output output\_ip |

the head of result



find the answer to question 2

|  |
| --- |
| hadoop fs -cat output\_ip/part-00000 | cat  > ../ipresult.txt  cd ../  grep 10.153.239.5 ipresult.txt |

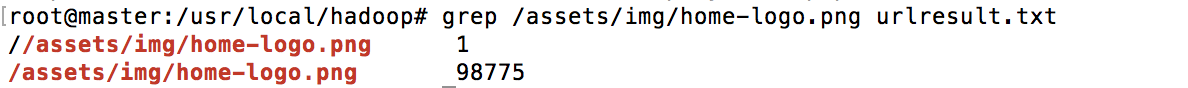


**Count URL:**

|  |
| --- |
| ../bin/hadoop jar ../hadoop-streaming-2.9.0.jar -files mapper.py,reducer.py -mapper "mapper.py url" -reducer reducer.py -input input\_part4 -output output\_url |

find the answer to question 1

|  |
| --- |
| hadoop fs -cat output\_url/part-00000 | cat  > ../urlresult.txt  cd ../  grep /assets/img/home-logo.png urlresult.txt |



**Find max IP**

In this case we will use mapper\_max.py and reducer\_max.py

mapper\_max is going to make every record to a common key, is value is (number\_of\_hit, ip)

reducer\_max is going to find the max value and its corresponding number of hit.

|  |
| --- |
| ../bin/hadoop jar ../hadoop-streaming-2.9.0.jar -files mapper\_max.py,reducer\_max.py -mapper mapper\_max.py -reducer reducer\_max.py -input output\_ip/part-00000 -output output\_ip\_max |

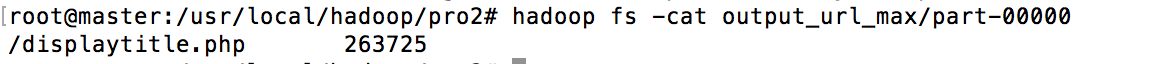
result



**Find max URL**

|  |
| --- |
| ../bin/hadoop jar ../hadoop-streaming-2.9.0.jar -files mapper\_max.py,reducer\_max.py -mapper mapper\_max.py -reducer reducer\_max.py -input output\_url/part-00000 -output output\_url\_max |

result



Reference:

Install Hadoop 2.x on Ubuntu – Hadoop Multi Node Cluster Setup

<https://data-flair.training/blogs/install-hadoop-2-x-on-ubuntu/>

Hadoop Streaming

<https://www.tutorialspoint.com/hadoop/hadoop_streaming.htm>