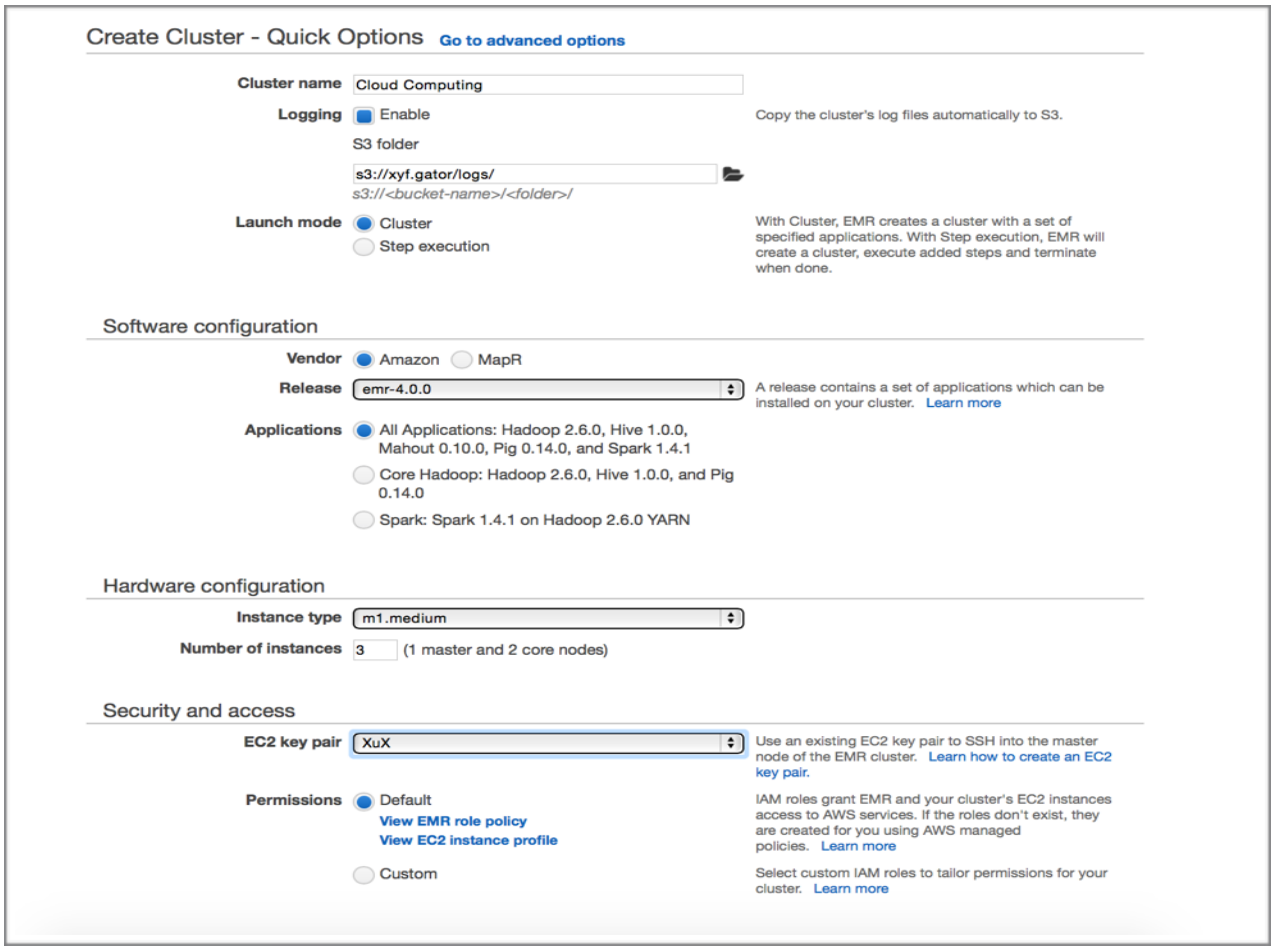
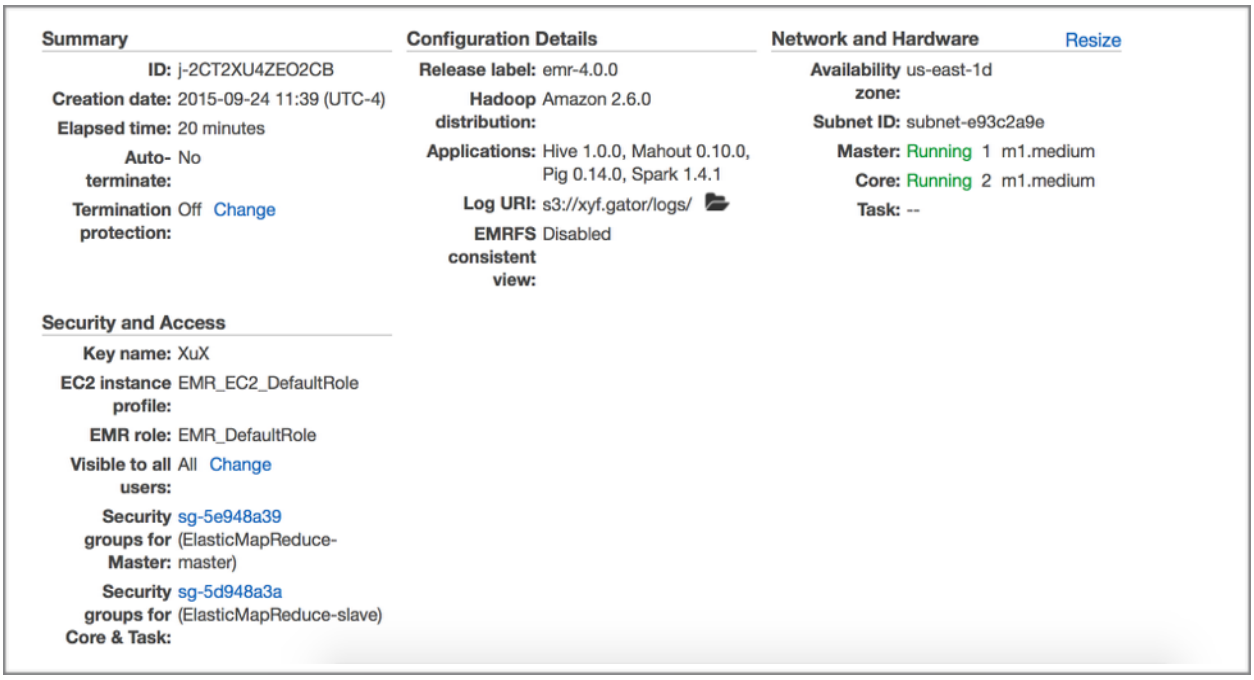
Report of Home Assignment One

Yufan Xu 7909-7439

For this assignment, professor ask use to run three different program in AWS to collect frequency of words in given data source. Firstly, I run Hadoop Pseudo-Distributed model in my localhost to test the correctness and reliability of the code. After this step, I upload my jar files and input source to Amazon Web Service. Since it is very complex to configure Hadoop in EC2 server, so I use Amazon Web Service EMR to conduct MapReduce process. My input data and source file are stored in AWS S3. EMR can easily get data from S3. According to description of assignment, I use three servers, one is master node and two others are slave node. *Picture 1* shows the configuration in AWS EMR. *Picture 2* shows the structure of this cluster.



Picture 1



Picture 2

Amazon EMR provides several ways to get data onto a cluster. The most common way is to upload the data to Amazon S3 and use the built-in features of Amazon EMR to load the data onto your cluster. *Picture 3* shows the data in S3.

Picture 3

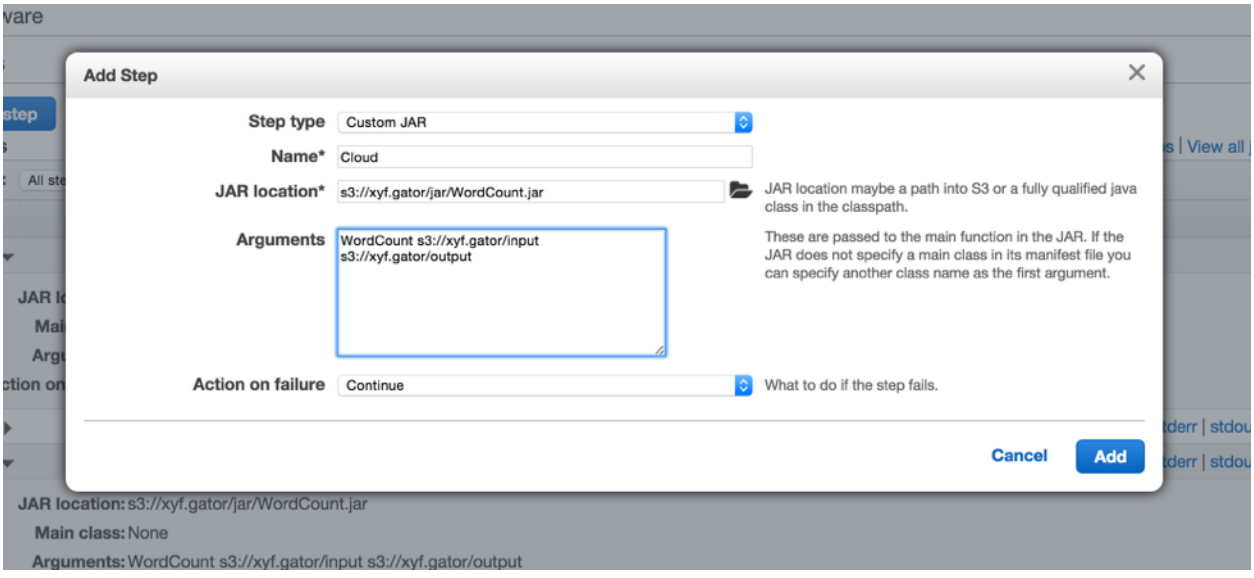
Now, I illustrate the details of result of three program and what I do in this home assignment.

Part I

In this part, we need count one-word frequency. Firstly, we need add MapReduce work in EMR. To submit a custom JAR step using the console, I do following steps.

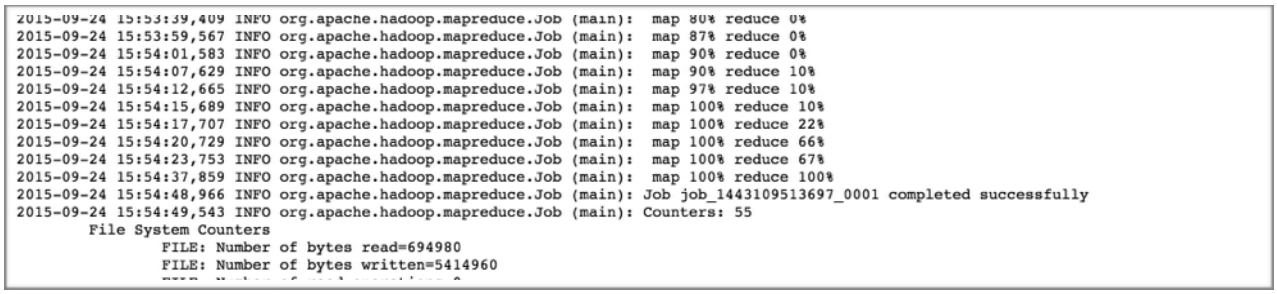
1. Open the Amazon EMR console
2. In the Cluster List, click the name of your cluster.
3. Scroll to the Steps section and expand it, then click Add step.
4. In the Add Step dialog:
    - For Step type, choose Custom JAR.
    - For Name, accept the default name or type a new name.
    - For JAR location, type or browse to the location of your JAR file.
    - For Arguments, type any required arguments as space-separated.
    - For Action on failure, accept the default option.
5. Click Add. The step appears in the console with a status of Pending.

Picture 4 show the console interface in EMR

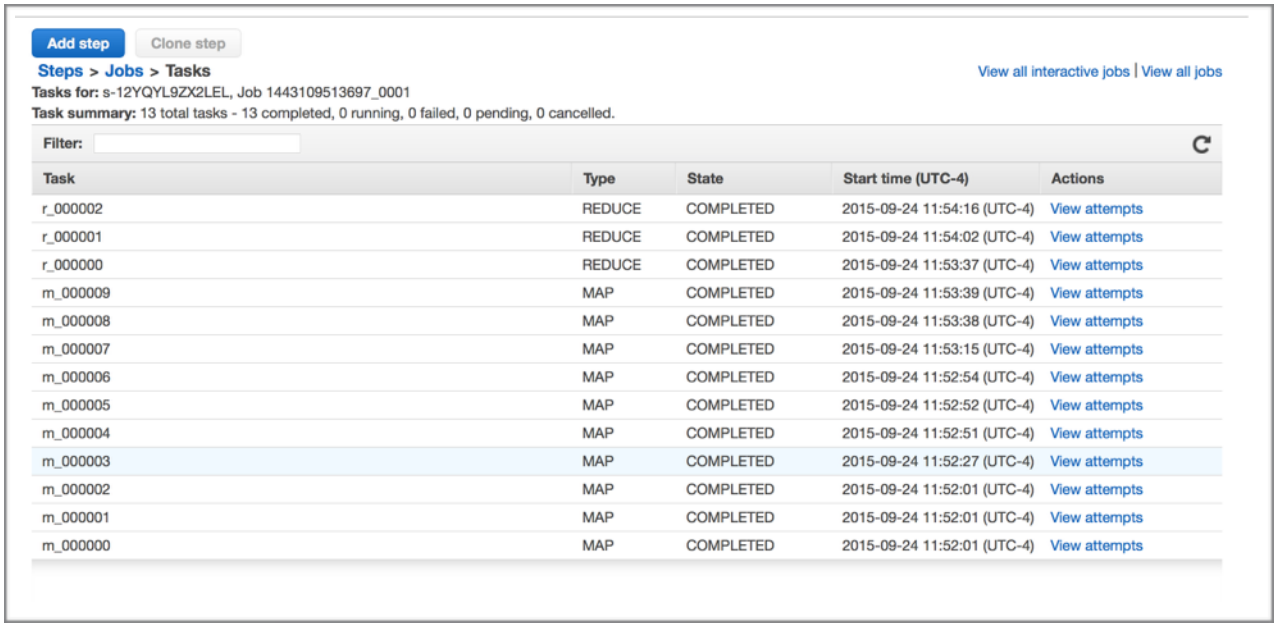


Picture 4

After this, EMR will start MapReduce work and I can get final result in S3 output folder. *Picture 5* and *Picture 6* shows the logs of this work. The output of the job is n number of part files where n is the number of reduce tasks. Each file contains a list of pairs in sorted order.



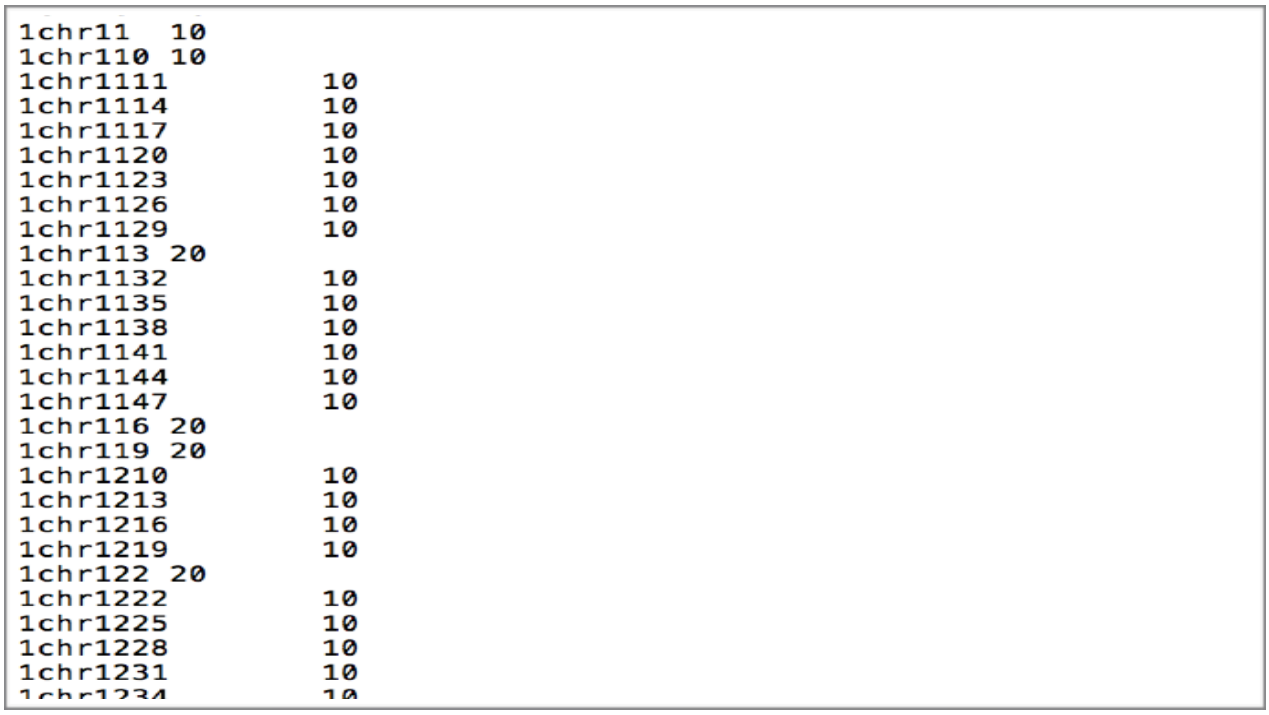
Picture 5



Task	Type	State	Start time (UTC-4)	Actions
r_000002	REDUCE	COMPLETED	2015-09-24 11:54:16 (UTC-4)	View attempts
r_000001	REDUCE	COMPLETED	2015-09-24 11:54:02 (UTC-4)	View attempts
r_000000	REDUCE	COMPLETED	2015-09-24 11:53:37 (UTC-4)	View attempts
m_000009	MAP	COMPLETED	2015-09-24 11:53:39 (UTC-4)	View attempts
m_000008	MAP	COMPLETED	2015-09-24 11:53:38 (UTC-4)	View attempts
m_000007	MAP	COMPLETED	2015-09-24 11:53:15 (UTC-4)	View attempts
m_000006	MAP	COMPLETED	2015-09-24 11:52:54 (UTC-4)	View attempts
m_000005	MAP	COMPLETED	2015-09-24 11:52:52 (UTC-4)	View attempts
m_000004	MAP	COMPLETED	2015-09-24 11:52:51 (UTC-4)	View attempts
m_000003	MAP	COMPLETED	2015-09-24 11:52:27 (UTC-4)	View attempts
m_000002	MAP	COMPLETED	2015-09-24 11:52:01 (UTC-4)	View attempts
m_000001	MAP	COMPLETED	2015-09-24 11:52:01 (UTC-4)	View attempts
m_000000	MAP	COMPLETED	2015-09-24 11:52:01 (UTC-4)	View attempts

Picture 6

Link of first part result(WordCount) is:
https://console.aws.amazon.com/s3/home?region=us-east-1&bucket=xyf.gator&prefix=output/
And following screenshot is part of result (*Picture 7*).

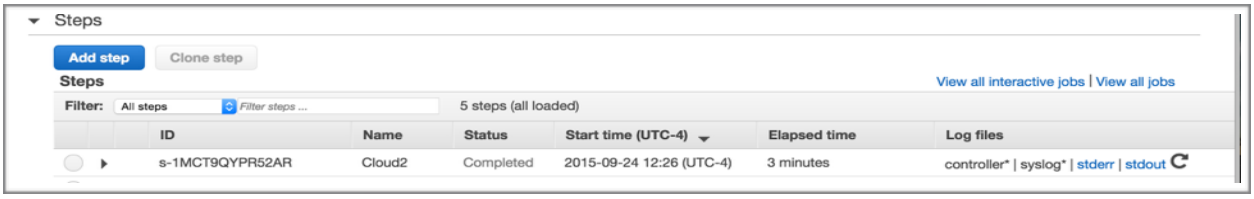


Picture 7

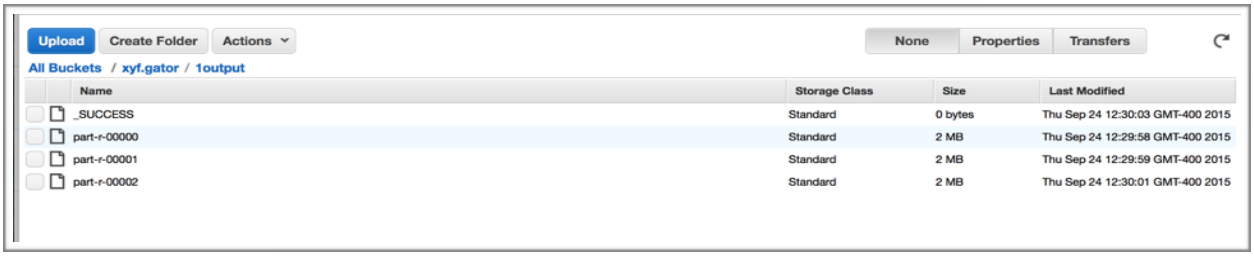
Part II

In this part, I count double-word frequency in the same input file. The precess is same as first part. So, I just show screen shot about part two.
Link of second part result(TwoWordCount) is:
https://console.aws.amazon.com/s3/home?region=us-east-1&bucket=xyf.gator&prefix=1output/

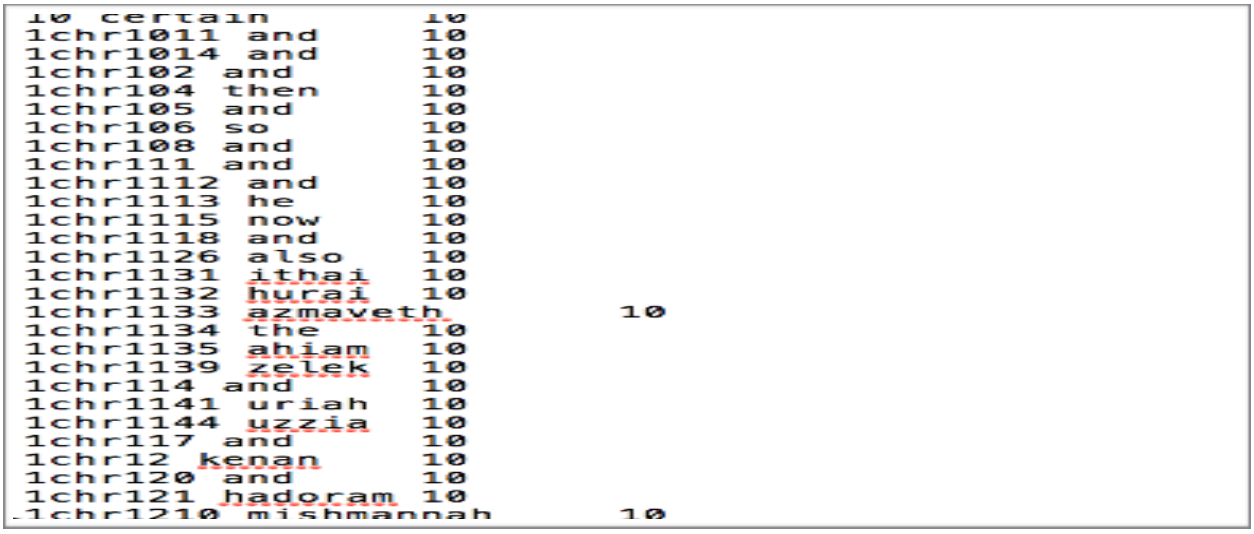
Picture 8, 9 show the result of second program in EMR. The output of the job is n number of part files where n is the number of reduce tasks. Each file contains a list of pairs in sorted order. *Picture 10* shows part result of part two.



Picture 8



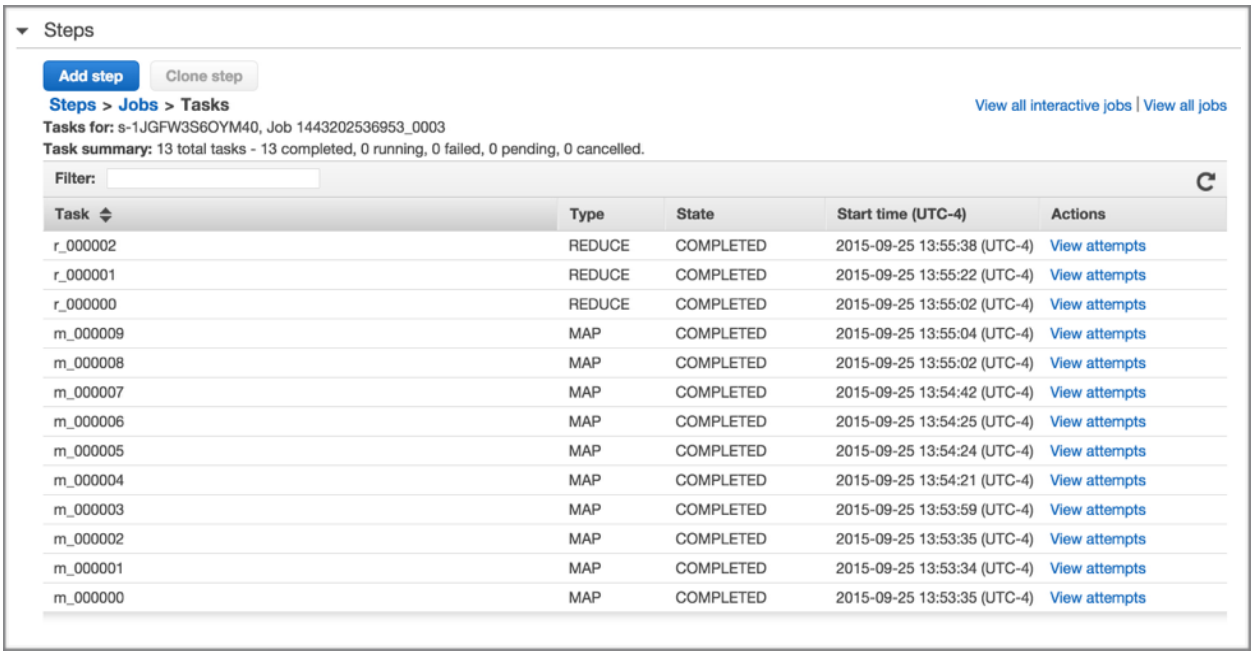
Picture 9



Picture 10

Part III

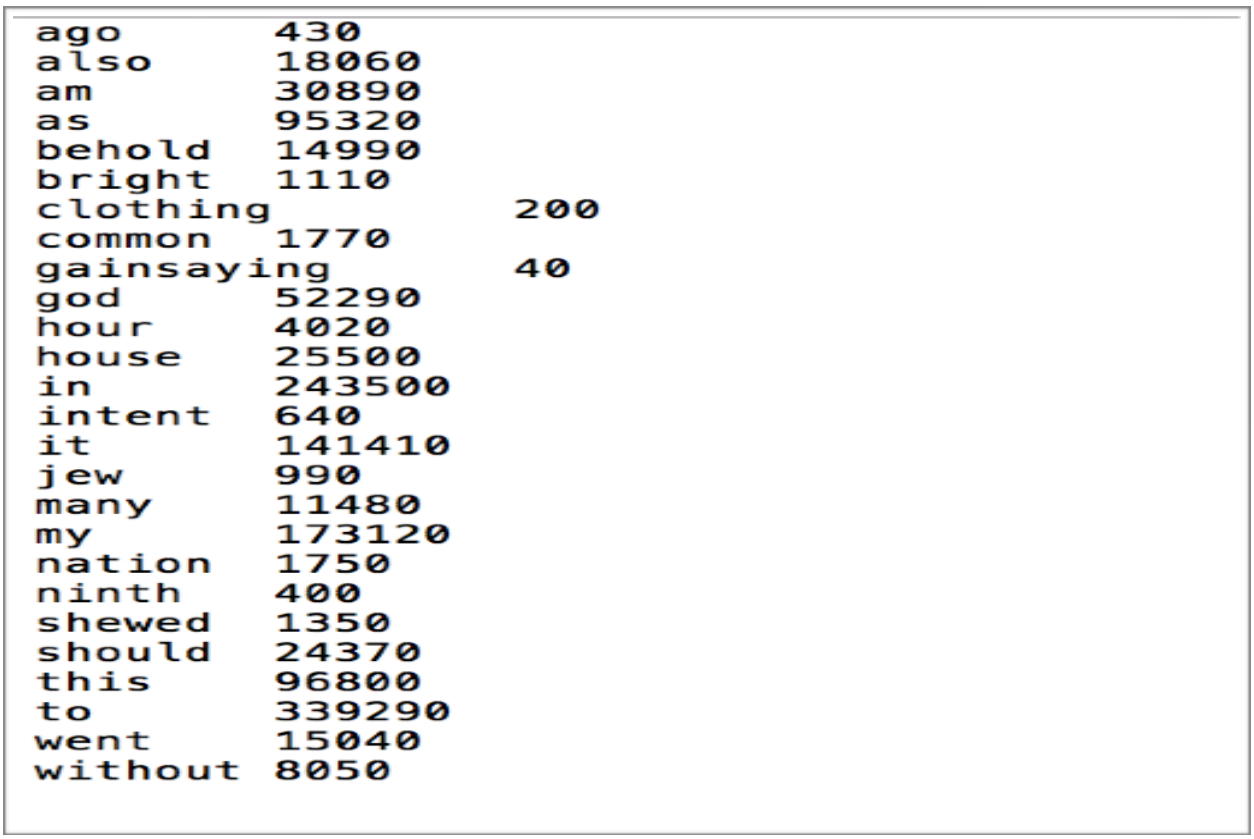
Using DistributedCache. Find frequency of one-words in another given list. It is to use the given list to find one-word frequency in bible.gz file. *Picture 11* shows the schedule process in part three.



Picture 11

The output of the job is n number of part files where n is the number of reduce tasks. Each file contains a list of pairs in sorted order. The output contains only those words which are present in the word-patterns.txt. *Picture 12,13* show the result of part three.

Picture 12



Picture 13

Link of first part result(Distributed) is:
https://console.aws.amazon.com/s3/home?region=us-east-1&bucket=xyf.gator&prefix=2output/

Finally, the command I use in three programs are following.
bin/hadoop jar WordCount.jar WordCount /input /output
bin/hadoop jar TwoWordCount.jar TwoWordCount /input /output
bin/hadoop jar Distributed.jar Distributed /word-patterns.txt /input /output