

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.

Create Cluster - Quick Options [Go to advanced options](#)

Cluster name

Logging ☒ Enable Copy the cluster's log files automatically to S3.

S3 folder s3://<bucket-name>/<folder>/

Launch mode ☒ Cluster With Cluster, EMR creates a cluster with a set of specified applications. With Step execution, EMR will create a cluster, execute added steps and terminate when done.
☐ Step execution

Software configuration

Vendor ☒ Amazon ☐ MapR

Release A release contains a set of applications which can be installed on your cluster. [Learn more](#)

Applications ☒ All Applications: Hadoop 2.6.0, Hive 1.0.0, Mahout 0.10.0, Pig 0.14.0, and Spark 1.4.1
☐ Core Hadoop: Hadoop 2.6.0, Hive 1.0.0, and Pig 0.14.0
☐ Spark: Spark 1.4.1 on Hadoop 2.6.0 YARN

Hardware configuration

Instance type

Number of instances (1 master and 2 core nodes)

Security and access

EC2 key pair Use an existing EC2 key pair to SSH into the master node of the EMR cluster. [Learn how to create an EC2 key pair.](#)

Permissions ☒ Default IAM roles grant EMR and your cluster's EC2 instances access to AWS services. If the roles don't exist, they are created for you using AWS managed policies. [Learn more](#)
[View EMR role policy](#)
[View EC2 instance profile](#)
☐ Custom Select custom IAM roles to tailor permissions for your cluster. [Learn more](#)

Picture 1

Summary	Configuration Details	Network and Hardware	Resize
ID: j-2CT2XU4ZEO2CB Creation date: 2015-09-24 11:39 (UTC-4) Elapsed time: 20 minutes Auto-terminate: No Termination protection: Off Change	Release label: emr-4.0.0 Hadoop distribution: Amazon 2.6.0 Applications: Hive 1.0.0, Mahout 0.10.0, Pig 0.14.0, Spark 1.4.1 Log URI: s3://xyf.gator/logs/ EMRFS consistent view: Disabled	Availability zone: us-east-1d Subnet ID: subnet-e93c2a9e Master: Running 1 m1.medium Core: Running 2 m1.medium Task: --	
Security and Access			
Key name: XuX EC2 instance profile: EMR_EC2_DefaultRole EMR role: EMR_DefaultRole Visible to all users: All Change Security groups for Master: sg-5e948a39 (ElasticMapReduce-Master) Security groups for Core & Task: sg-5d948a3a (ElasticMapReduce-slave)			

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.

Upload

Create Folder

Actions ▾

None

Properties

Transfers

All Buckets / xyf.gator

	Name	Storage Class	Size	Last Modified
<input type="checkbox"/>	 1output	--	--	--
<input type="checkbox"/>	 2output	--	--	--
<input type="checkbox"/>	 input	--	--	--
<input type="checkbox"/>	 jar	--	--	--
<input type="checkbox"/>	 logs	--	--	--
<input type="checkbox"/>	 output	--	--	--
<input type="checkbox"/>	 pattern	--	--	--

Picture 3

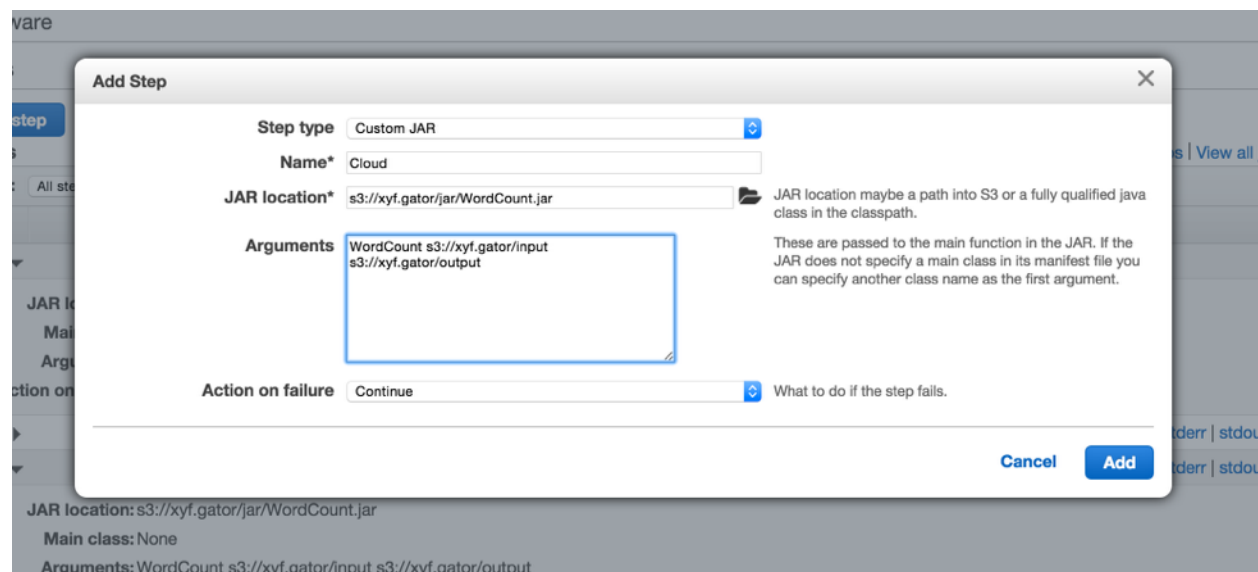
Now, I illustrate the details of result of three program and what I do in this home assignment.

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.

```

2015-09-24 15:53:39,409 INFO org.apache.hadoop.mapreduce.Job (main): map 80% reduce 0%
2015-09-24 15:53:59,567 INFO org.apache.hadoop.mapreduce.Job (main): map 87% reduce 0%
2015-09-24 15:54:01,583 INFO org.apache.hadoop.mapreduce.Job (main): map 90% reduce 0%
2015-09-24 15:54:07,629 INFO org.apache.hadoop.mapreduce.Job (main): map 90% reduce 10%
2015-09-24 15:54:12,665 INFO org.apache.hadoop.mapreduce.Job (main): map 97% reduce 10%
2015-09-24 15:54:15,689 INFO org.apache.hadoop.mapreduce.Job (main): map 100% reduce 10%
2015-09-24 15:54:17,707 INFO org.apache.hadoop.mapreduce.Job (main): map 100% reduce 22%
2015-09-24 15:54:20,729 INFO org.apache.hadoop.mapreduce.Job (main): map 100% reduce 66%
2015-09-24 15:54:23,753 INFO org.apache.hadoop.mapreduce.Job (main): map 100% reduce 67%
2015-09-24 15:54:37,859 INFO org.apache.hadoop.mapreduce.Job (main): map 100% reduce 100%
2015-09-24 15:54:48,966 INFO org.apache.hadoop.mapreduce.Job (main): Job job_1443109513697_0001 completed successfully
2015-09-24 15:54:49,543 INFO org.apache.hadoop.mapreduce.Job (main): Counters: 55
    File System Counters
      FILE: Number of bytes read=694980
      FILE: Number of bytes written=5414960
      ...

```

Picture 5

Add step
Clone step

Steps > Jobs > Tasks
View all interactive jobs | View all jobs

Tasks for: s-12YQYL9ZX2LEL, Job 1443109513697_0001

Task summary: 13 total tasks - 13 completed, 0 running, 0 failed, 0 pending, 0 cancelled.

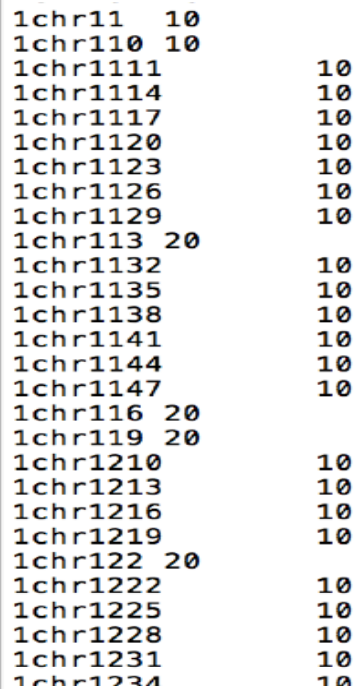
Task	Type	State	Start time (UTC-4)	Actions
r_000002	REDUCE	COMPLETED	2015-09-24 11:54:16 (UTC-4)	View attempts
r_000001	REDUCE	COMPLETED	2015-09-24 11:54:02 (UTC-4)	View attempts
r_000000	REDUCE	COMPLETED	2015-09-24 11:53:37 (UTC-4)	View attempts
m_000009	MAP	COMPLETED	2015-09-24 11:53:39 (UTC-4)	View 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*).



```
1chr11 10
1chr110 10
1chr1111 10
1chr1114 10
1chr1117 10
1chr1120 10
1chr1123 10
1chr1126 10
1chr1129 10
1chr113 20
1chr1132 10
1chr1135 10
1chr1138 10
1chr1141 10
1chr1144 10
1chr1147 10
1chr116 20
1chr119 20
1chr1210 10
1chr1213 10
1chr1216 10
1chr1219 10
1chr122 20
1chr1222 10
1chr1225 10
1chr1228 10
1chr1231 10
1chr1234 10
```

Picture 7

Part II

In this part, I count double-word frequency in the same input file. The process 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.

ID	Name	Status	Start time (UTC-4)	Elapsed time	Log files
s-1MCT9QYPR52AR	Cloud2	Completed	2015-09-24 12:26 (UTC-4)	3 minutes	controller* syslog* stderr stdout

Picture 8

Name	Storage Class	Size	Last Modified
_SUCCESS	Standard	0 bytes	Thu Sep 24 12:30:03 GMT-400 2015
part-r-00000	Standard	2 MB	Thu Sep 24 12:29:58 GMT-400 2015
part-r-00001	Standard	2 MB	Thu Sep 24 12:29:59 GMT-400 2015
part-r-00002	Standard	2 MB	Thu Sep 24 12:30:01 GMT-400 2015

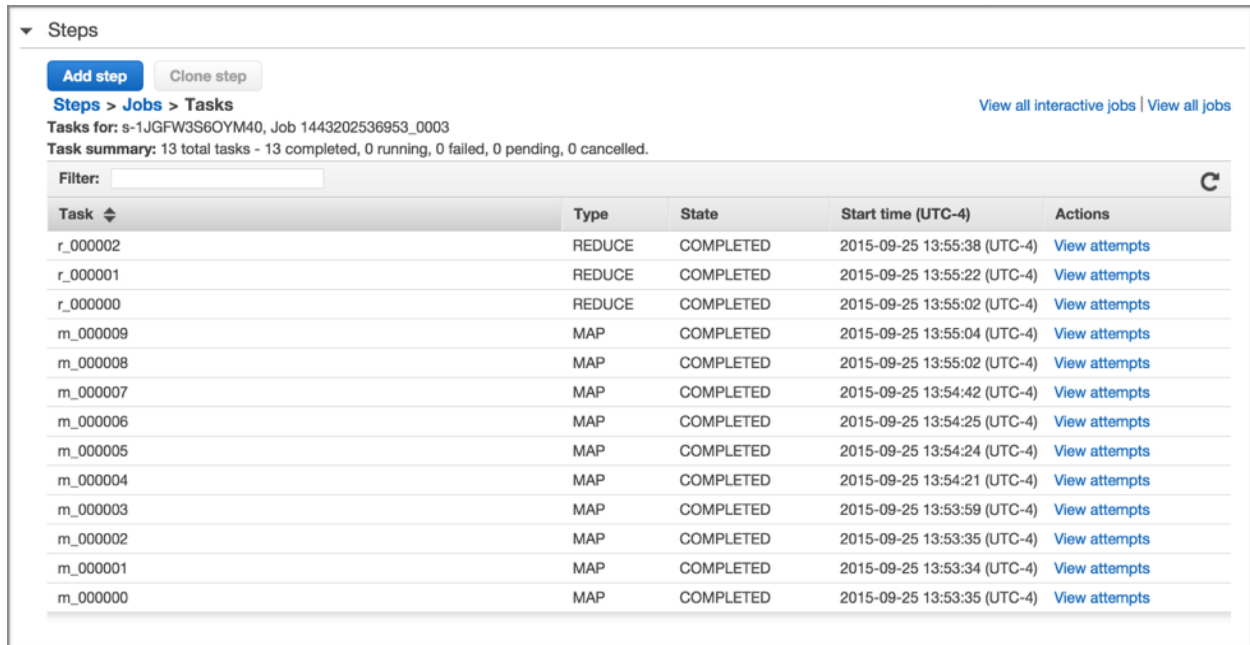
Picture 9

10	certain	10
1chr1011	and	10
1chr1014	and	10
1chr102	and	10
1chr104	then	10
1chr105	and	10
1chr106	so	10
1chr108	and	10
1chr111	and	10
1chr1112	and	10
1chr1113	he	10
1chr1115	now	10
1chr1118	and	10
1chr1126	also	10
1chr1131	ithai	10
1chr1132	hurai	10
1chr1133	azmaveth	10
1chr1134	the	10
1chr1135	ahiam	10
1chr1139	zelek	10
1chr114	and	10
1chr1141	uriah	10
1chr1144	uzzia	10
1chr117	and	10
1chr12	kenan	10
1chr120	and	10
1chr121	hadoram	10
1chr1210	mishmannah	10

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.



Steps

[Add step](#) [Clone step](#)

[Steps > Jobs > Tasks](#) [View all interactive jobs](#) [View all jobs](#)

Tasks for: s-1JGFW3S6OYM40, Job 1443202536953_0003

Task summary: 13 total tasks - 13 completed, 0 running, 0 failed, 0 pending, 0 cancelled.

Filter:

Task	Type	State	Start time (UTC-4)	Actions
r_000002	REDUCE	COMPLETED	2015-09-25 13:55:38 (UTC-4)	View attempts
r_000001	REDUCE	COMPLETED	2015-09-25 13:55:22 (UTC-4)	View attempts
r_000000	REDUCE	COMPLETED	2015-09-25 13:55:02 (UTC-4)	View attempts
m_000009	MAP	COMPLETED	2015-09-25 13:55:04 (UTC-4)	View attempts
m_000008	MAP	COMPLETED	2015-09-25 13:55:02 (UTC-4)	View attempts
m_000007	MAP	COMPLETED	2015-09-25 13:54:42 (UTC-4)	View attempts
m_000006	MAP	COMPLETED	2015-09-25 13:54:25 (UTC-4)	View attempts
m_000005	MAP	COMPLETED	2015-09-25 13:54:24 (UTC-4)	View attempts
m_000004	MAP	COMPLETED	2015-09-25 13:54:21 (UTC-4)	View attempts
m_000003	MAP	COMPLETED	2015-09-25 13:53:59 (UTC-4)	View attempts
m_000002	MAP	COMPLETED	2015-09-25 13:53:35 (UTC-4)	View attempts
m_000001	MAP	COMPLETED	2015-09-25 13:53:34 (UTC-4)	View attempts
m_000000	MAP	COMPLETED	2015-09-25 13:53:35 (UTC-4)	View attempts

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.



All Buckets / xyf.gator / 2output

Name	Storage Class	Size	Last Modified
<input type="checkbox"/> _SUCCESS	Standard	0 bytes	Fri Sep 25 13:56:09 GMT-400 2015
<input type="checkbox"/> part-r-00000	Standard	285 bytes	Fri Sep 25 13:56:04 GMT-400 2015
<input type="checkbox"/> part-r-00001	Standard	274 bytes	Fri Sep 25 13:56:05 GMT-400 2015
<input type="checkbox"/> part-r-00002	Standard	405 bytes	Fri Sep 25 13:56:07 GMT-400 2015

Picture 12

ago	430	
also	18060	
am	30890	
as	95320	
behold	14990	
bright	1110	
clothing		200
common	1770	
gainsaying		40
god	52290	
hour	4020	
house	25500	
in	243500	
intent	640	
it	141410	
jew	990	
many	11480	
my	173120	
nation	1750	
ninth	400	
shewed	1350	
should	24370	
this	96800	
to	339290	
went	15040	
without	8050	

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