

Assignment 1

Big Data Processing

Wenyao JIN

Hadoop Setup

1. Version

```
[cloudera@quickstart wordcount]$ hadoop version
Hadoop 2.6.0-cdh5.5.0
Subversion http://github.com/cloudera/hadoop -r fd21232cef7b8c1f536965897ce20f50b83ee7b2
Compiled by jenkins on 2015-11-09T20:37Z
Compiled with protoc 2.5.0
From source with checksum 98e07176d1787150a6a9c087627562c
This command was run using /usr/jars/hadoop-common-2.6.0-cdh5.5.0.jar
```

2. Checknative

```
[cloudera@quickstart wordcount]$ hadoop checknative
17/02/13 05:27:44 INFO bzip2.Bzip2Factory: Successfully loaded & initialized native-bzip2 library system-native
17/02/13 05:27:44 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
Native library checking:
hadoop: true /usr/lib/hadoop/lib/native/libhadoop.so.1.0.0
zlib: true /lib64/libz.so.1
snappy: true /usr/lib/hadoop/lib/native/libsnappy.so.1
lz4: true revision:99
bzip2: true /lib64/libbz2.so.1
openssl: false Cannot find AES-CTR support, is your version of Openssl new enough?
-
```

3. Dfsadmin

```
[cloudera@quickstart wordcount]$ hadoop dfsadmin -report
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
```

```
Configured Capacity: 58665738240 (54.64 GB)
Present Capacity: 47308513280 (44.06 GB)
DFS Remaining: 46522322944 (43.33 GB)
DFS Used: 786190336 (749.77 MB)
DFS Used%: 1.66%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0
```

```
-----
Live datanodes (1):
```

```
Name: 127.0.0.1:50010 (quickstart.cloudera)
Hostname: quickstart.cloudera
Decommission Status : Normal
Configured Capacity: 58665738240 (54.64 GB)
DFS Used: 786190336 (749.77 MB)
Non DFS Used: 11357224960 (10.58 GB)
DFS Remaining: 46522322944 (43.33 GB)
DFS Used%: 1.34%
DFS Remaining%: 79.30%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 2
Last contact: Mon Feb 13 05:31:42 PST 2017
```

4. Virtual Machine Specs

Number of Cores: 4

Number of Ram: 4 GB

HDD: 64 GB

5. The default settings are not changed. The number of Ram is tested to be just enough for the virtual machine to run fluently.

Implementation

1. Word Count

The word count job is generally the same as the example given on assignment 1. The mapper keeps 1 as values, and the reducer sum all the values together. By adding a filter on the write phase of the reducer, we output only results with frequency higher than 4000.

2. Invert Table

The mapper's output of this job takes Text(Word) as key and Text(Filename) as value. The catch is that we need to construct a filter list of stop words. So I overridden the setup method of the super class to load the output of Word Counts when the program constructs the mapper.

The reducer's output of this job takes Text(Word) as keys and an Array of Text as value. The difficulty will be to implement an output of array (List of files). In concern of modularization, I chose an implementation of subclass TextArray (Inheriting ArrayWritable) as the output of Word, then override the toString method to conform to the desired output format.

An enum is created for a user defined Counter to print out the length of only words for each document.

3. Invert Table Extension

For this extension job, the mapper remains unchanged.

For the reducer, a frequency needs to be counted. I used a java List to store the mappers' output (filename as values), then a HashSet to retrieve unique filenames.

Collections.frequency is used to calculate the occurrence of each filename. Then combined with frequency, an output String is constructed as the output of reducer.

Assumption of data processing

1. After an analyse of the data set, I decided to use space and '--' as separators to split data. A regex expression is used to mark separators when one or more spaces are encountered or more than one '-' are encountered.
2. Capital is not used to distinguish to words. So for convenience, every word is transformed to lower case. As for the punctuations in the data, they were also wiped out of the words.

Text scenario and Result

1. Word Count with 10 reducer

User:	cloudera
Name:	WordCount
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Sun Feb 12 09:48:55 -0800 2017
Elapsed:	2mins, 20sec
Tracking URL:	History
Diagnostics:	

Job Name:	WordCount
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Feb 12 09:48:55 PST 2017
Started:	Sun Feb 12 09:49:04 PST 2017
Finished:	Sun Feb 12 09:51:15 PST 2017
Elapsed:	2mins, 10sec
Diagnostics:	
Average Map Time	34sec
Average Shuffle Time	33sec
Average Merge Time	4sec
Average Reduce Time	6sec

Execution time : 130 s

2. Word Count with 10 reducer and combiner

User:	cloudera
Name:	WordCount
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Sun Feb 12 09:56:28 -0800 2017
Elapsed:	1mins, 40sec
Tracking URL:	History
Diagnostics:	

Job Name:	WordCount
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Feb 12 09:56:28 PST 2017
Started:	Sun Feb 12 09:56:35 PST 2017
Finished:	Sun Feb 12 09:58:09 PST 2017
Elapsed:	1mins, 33sec
Diagnostics:	
Average Map Time	29sec
Average Shuffle Time	25sec
Average Merge Time	0sec
Average Reduce Time	2sec

The execution time is reduced to 93 s, with merge and reduce time significantly reduced. It proved that with the combiner grouping the mapped values before passing to reducers, data size to pass can be largely reduced thus reduce time in each phrase.

3. Word Count with 10 reducers, combiner, and default codec compressor

User:	cloudera
Name:	WordCount
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Sun Feb 12 10:02:37 -0800 2017
Elapsed:	1mins, 37sec
Tracking URL:	History
Diagnostics:	

Job Name:	WordCount
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Feb 12 10:02:37 PST 2017
Started:	Sun Feb 12 10:02:44 PST 2017
Finished:	Sun Feb 12 10:04:14 PST 2017
Elapsed:	1mins, 30sec
Diagnostics:	
Average Map Time	23sec
Average Shuffle Time	25sec
Average Merge Time	0sec
Average Reduce Time	2sec

The execution time is reduced to 90 s, with map time reduced. It proved that by compressing the output of mapper, we can reduce the transfer time in the map phrase.

4. Word Count with 50 reducers

User:	cloudera
Name:	WordCount
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Sun Feb 12 10:09:52 -0800 2017
Elapsed:	5mins, 36sec
Tracking URL:	History
Diagnostics:	

Job Name:	WordCount
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Feb 12 10:09:52 PST 2017
Started:	Sun Feb 12 10:09:59 PST 2017
Finished:	Sun Feb 12 10:15:28 PST 2017
Elapsed:	5mins, 28sec
Diagnostics:	
Average Map Time	26sec
Average Shuffle Time	29sec
Average Merge Time	0sec
Average Reduce Time	2sec

The explosion of the elapsed time is probably due to the framework overhead with two many reducers to manage. The reduce time is unchanged, because not all reducers are running on parallel. In this case, the number of reducers is set too high!

5. Invert Table

User:	cloudera
Name:	InvertTable
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Sun Feb 12 15:37:40 -0800 2017
Elapsed:	1mins, 44sec
Tracking URL:	History
Diagnostics:	

Job Name:	InvertTable
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Feb 12 15:37:40 PST 2017
Started:	Sun Feb 12 15:37:45 PST 2017
Finished:	Sun Feb 12 15:39:24 PST 2017
Elapsed:	1mins, 38sec
Diagnostics:	
Average Map Time	23sec
Average Shuffle Time	24sec
Average Merge Time	2sec
Average Reduce Time	7sec

A piece of the result file:


```

younker pg100.txt,pg31100.txt
youth   pg31100.txt,pg100.txt,pg3200.txt
zartlichsten   pg3200.txt
zealous pg3200.txt,pg31100.txt,pg100.txt
zeilerus      pg3200.txt
ziani   pg3200.txt
zip      pg3200.txt,pg100.txt
zoes     pg3200.txt
zones    pg3200.txt
zounds   pg100.txt
zu        pg3200.txt
zulus    pg3200.txt
zwaggerd      pg100.txt_

```

6. Counter

In my case, 70102 unique words are found in the document corpus. The built-in counter: Map-reduce Framework. Reduce output record, revealed this information.

The user-defined counter yields values: PG100=12113, PG31100=2324, PG3200=35183

7. Invert Table Extension

User:	cloudera
Name:	InvertTableExtension
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Mon Feb 13 08:43:33 -0800 2017
Elapsed:	2mins, 47sec
Tracking URL:	History
Diagnostics:	

Job Name:	InvertTableExtension
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Mon Feb 13 08:43:33 PST 2017
Started:	Mon Feb 13 08:43:42 PST 2017
Finished:	Mon Feb 13 08:46:20 PST 2017
Elapsed:	2mins, 38sec
Diagnostics:	
Average Map Time	51sec
Average Shuffle Time	42sec
Average Merge Time	3sec
Average Reduce Time	10sec

A piece of the output:

```

yokd      pg100.txt#2
yongrey   pg100.txt#1
youd      pg3200.txt#156,pg100.txt#17
youfl     pg100.txt#1
youngly   pg100.txt#2
yunker    pg31100.txt#1,pg100.txt#3
youth     pg31100.txt#65,pg100.txt#277,pg3200.txt#239
zartlichsten  pg3200.txt#2
zealous   pg100.txt#6,pg3200.txt#10,pg31100.txt#7
zeilerus  pg3200.txt#1
ziani     pg3200.txt#1
zip       pg3200.txt#1,pg100.txt#1
zoes      pg3200.txt#1
zones     pg3200.txt#5
zounds    pg100.txt#24
zu        pg3200.txt#21
zulus     pg3200.txt#8
zwaggerd  pg100.txt#1

```

Conclusion

All the source codes are provided at the git lab repository. Due to the size of the output files, result of the output files is presented with screen capture of fragments.