Cloud Computing Capstone - Data Extraction, Batch Processing with Hadoop

Task 1 Report

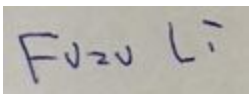
Instructor:

Indranil Gupta, Roy H. Campbell, P. Brighten Godfrey, Ankit Singla, Reza Farivar

ILLINOIS

Submitted as a partial fulfillment of course completion.

Submitted by

Avatar	Name	Signature
	Fuzu Li	

Video Demonstration: https://www.youtube.com/watch?v=j7KBRNtLO4M

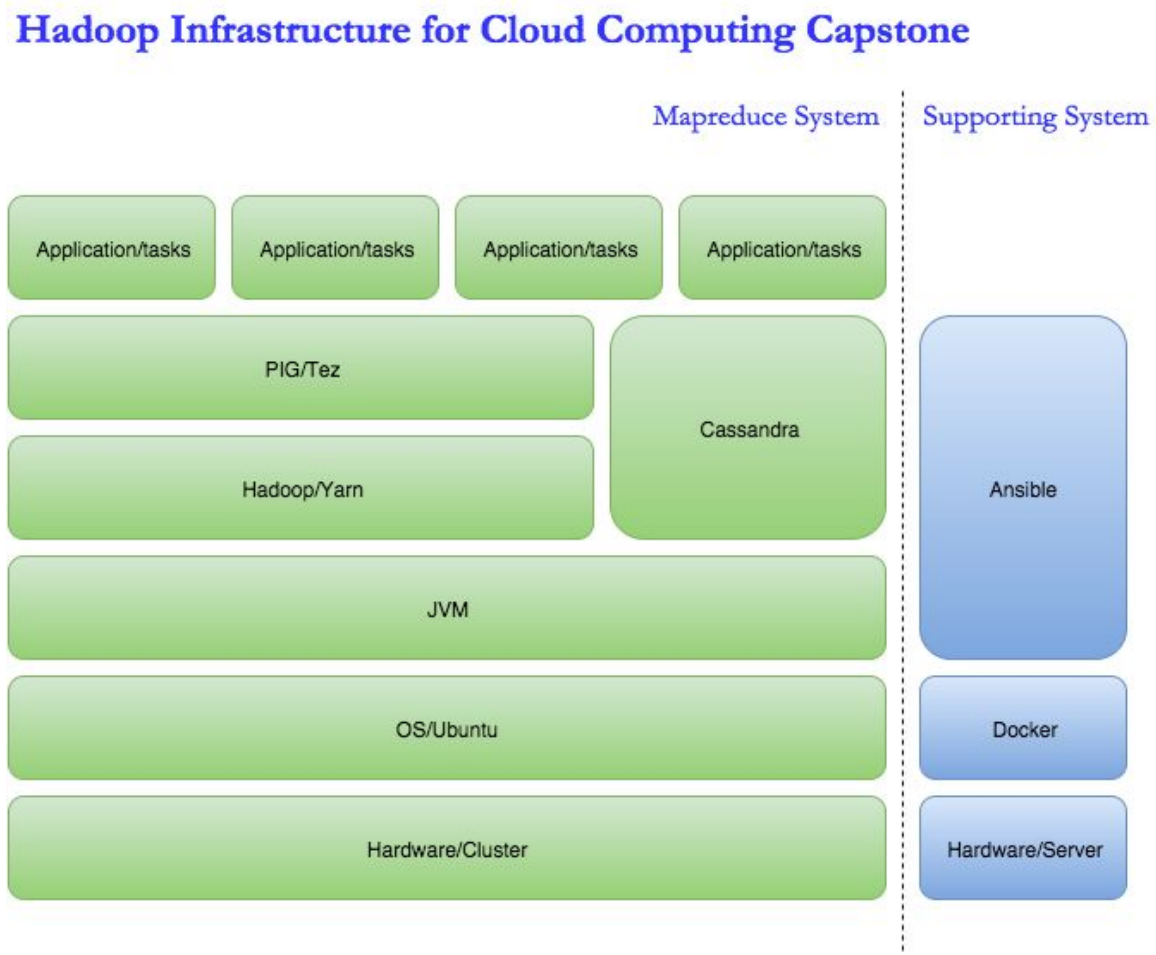
A brief overview of how extracted and cleaned the data.
According to the problem, as analyzed, I found the on-time performance related data is most useful for the problem domain, so I picked up the on-time performance data from the specific folder, then uncompress them to my local folder, then copy them into hadoop file system.

The principle for cleaning the raw data is to prevent errores, reduce duplication, and improve efficiency. For this task, I keep trying best to maintain the original format of the data, remove the unstructured data avoid breaking the data process.

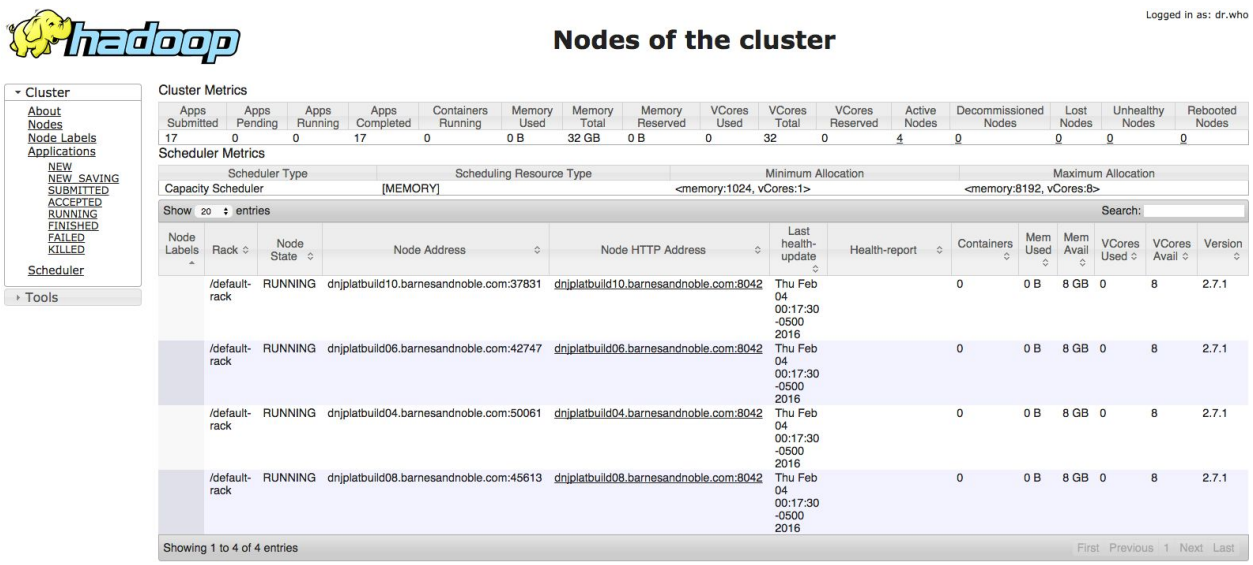
With the principle in mind, I keep the clean structured csv file for On_Time_On_Time_Performance_{1988-2008}_{1-12}.csv, remove the readme.html, in order to be imported by hadoop/pig scripts.

I created an open source repository on github to save some scripts to cleanup the raw data and import to hadoop file system. Please refer to Reference #1 for details.

A brief overview of how integrated each system.
Here is a brief overview for my local hadoop cluster infrastructure:



I have 6 Nodes for Hadoop/PIG, includes 1 for Namenode/ResourceManager, 1 for secondary Namenode, 4 for Datanode/TaskManager.



I have 4 nodes for Cassandra Cluster, some of them share same servers with Hadoop cluster, as following:

```
build@dnjplatbuild02:~$ nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address       Load       Tokens       Owns    Host ID                               Rack
UN  10.5.178.84   367.75 MB  256          ?       dc866979-3427-4bd0-8344-49beea35f5a7  rack1
UN  10.5.178.82   345.01 MB  256          ?       2bfb70fe-99ed-4be8-a5f4-aefe8629caf8  rack1
UN  10.5.178.79   442.47 MB  256          ?       f0abdf90-a967-4798-b2e6-9d89c3f7206b  rack1
UN  10.5.178.91   345.88 MB  256          ?       35b4b2b8-8f54-433d-a1ca-0a8b37075575  rack1
```

For creating and maintaining the cluster, I created another open source repository on github to automatically manage the Hadoop/Cassandra cluster to install Java, Hadoop, PIG, Cassandra, Storm and Spark (for task 2), then config them in a cluster.

Please take a deep look at [Reference #2](#).

The approaches and algorithms to answer each question?

Group 1.1

Group 1.2

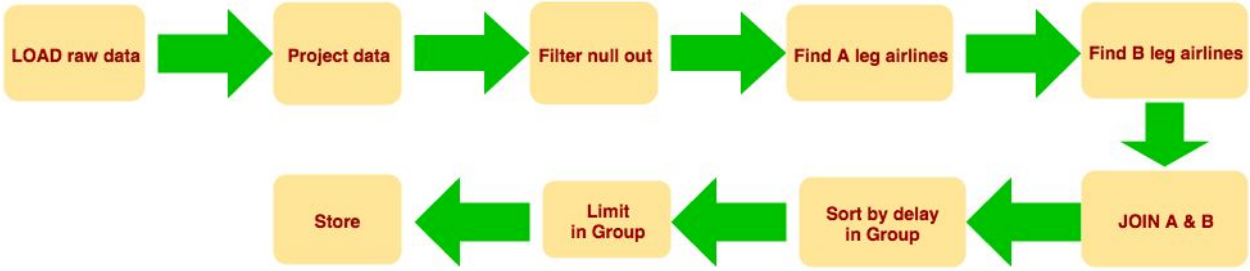
Group 2.1

Group 2.2

Group 2.3

Group 3.1

Group 3.2



The results of each question is as following:

Group 1.1's result:

https://gist.github.com/lifuzu/352989a7a40db1e53b7f

Group 1.2's result:

https://gist.github.com/lifuzu/511b649777b876ba9723

Group 2.1's full result:

https://gist.github.com/lifuzu/c844c01e9824c6d32bc1

Group 2.2's full result:

https://gist.github.com/lifuzu/439933ee2718b92c36c1

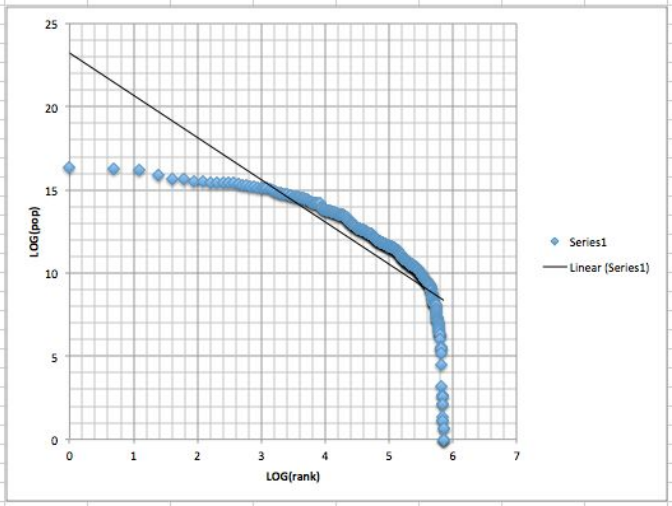
Group 2.3's full result:

https://gist.github.com/lifuzu/629354f6cd916f264997

Group 3.1's result:

RANK	LOG(RANK)	LOG(POP)	Airport
1	0	16.33717929	ORD
2	0.693147181	16.26136639	ATL
3	1.098612289	16.19499215	DFW
4	1.386294361	15.85979062	LAX
5	1.609437912	15.70038598	PHX
6	1.791759469	15.65189072	DEN
7	1.945910149	15.54479551	DTW
8	2.079441542	15.51674959	IAH
9	2.197224577	15.46401783	MSP
10	2.302585093	15.4585811	SFO
11	2.397895273	15.45197416	EWR
12	2.48490665	15.44970664	STL
13	2.564949357	15.41751249	LAS
14	2.63905733	15.38926139	CLT
15	2.708050201	15.28273193	LGA
16	2.772588722	15.27670736	BOS
17	2.833213344	15.221522	PHL
18	2.890371758	15.18573143	PIT
19	2.944438979	15.1544811	SLC
20	2.995732274	15.13372975	SEA
21	3.044522438	15.12973115	MCO
22	3.091042453	15.1196766	CVG
23	3.135494216	15.05793946	DCA
24	3.17805383	14.98822796	BWI
25	3.218875825	14.88315423	SAN
26	3.258096538	14.82129276	MIA
27	3.295836866	14.81137743	CLE
28	3.33220451	14.7522539	IAD
29	3.36729583	14.73681072	JFK
30	3.401197382	14.72553087	TPA
31	3.433987204	14.63862579	HOU
32	3.465735903	14.63604297	MEM
33	3.496507561	14.61423098	BNA
34	3.526360525	14.61303938	MDW

PEARSON Value:	-0.7927981



Pearson Product-Moment Correlation Coefficient: **-0.7927981**

Group 3.2's full result:
https://gist.github.com/lifuzu/7080eadf72d38778f6c7

Some optimizations employed on the task:
I employed the following optimizations on the Hadoop cluster and application level:

1. Project Early and Often
2. Filter Early and Often
3. Drop Nulls Before a Join
4. Use the PARALLEL Clause - SET DEFAULT_PARALLEL 10;

Opinion about whether the results make sense and are useful in any way:
Definitely, the results came from the real data should be very useful to analyze the aviation history and forecast the future to provide convenience to potential passengers.

References:

1. https://github.com/lifuzu/cccapstone-tasks
2. https://github.com/lifuzu/cccapstone-cluster-manager