Objectives:

* Implement and understand MapReduce jobs on a production level cluster
* Implement and analyze the effects and influences of compression codecs on MapReduce
* Implement and analyze the effects of Intermediate Compression on MapReduce jobs
* Implement and explain the use and effect of Chaining in MapReduce Jobs

Start by creating a folder in your private Github repo under ITMD-521 named week-07. There will be five items. Create a sub-folder for each item, named **item-one, item-two, item-three**, and so forth. Push all of your results there **formatted in a readable** format using Github Markdown (essentially HTML) including all diagrams and charts, in a file named **ReadMe.md.** If a step requires code, include all code needed to successfully reproduce the results.

* <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
* <https://guides.github.com/features/mastering-markdown/>

As a hint – get used to writing and pushing a small amount of your report – do not “write” it in Word and then cut and paste – as the markdown handles the format—you will be wasting time and effort. I should see multiple commits for your deliverables in the history section of Github.

Make use of P. 177 execution times as well.

Deliverable on Blackboard will be your Github URL

**Item-one**

Rerun Week 2 analysis

These full data sets are located in:

* 1990 /user/root/ncdc/1990/
* 1990 & 1992 /user/root/ncdc/1990and1992/
* 90-93 /user/root/ncdc/90-93/

On our cluster - take note of the resources on the cluster (2 processors 16 GB of memory 450 GB of datanode storage x 6 nodes)

Explain why the run times are different - note the amount of resources currently in use (8088) and explain in detail how MR takes advantage of the distributed cluster (Cite diagrams from the various chapters in the book explaining what is going on)

**Item-two:**

Run the above jobs again this time using the MaxTemperature with combiner example code

Repeat your graphing and analysis and explain why and how using Figure 7-1 the execution time changed - note the amount of resources in use at the time of execution (this will vary depending when you are working and results will not be the same for everyone)

**Item-three:**

Based on P 118 of the Text Book (ePub) You are to rerun the above MaxTemperature and MaxTemperature with combiners in matrix of variables

Run the jobs with the input uncompressed, gzip compressed, and bzip2 compressed (over each dataset) Note the execution times, the resources in use, and explain the results for the jobs with a combiner and without a combiner and explain from the book why the results occurred.

**Item-four:**

Repeat the above experiment by enabling intermediate compression (P.118) first with Gzip then with LZO (using the book example New API code)

Explain from the text and Figure 7-1 how intermediate compression increases through put or has a negative effect on throughput.

Cap 5

126 compressing map output 107

23 lecuter

**Item-five:**

Run the jobs again this time using NATIVE HADOOP Compression Codecs. (By default the system will use Java compression codecs). Modify your import statements based on what we learned in chapter 5 of the textbook.

Submit your Mapper, Reducer, and Driver classes and note the changes you made to the import statements to enable NATIVE hadoop comrpession codecs (note by default)

Provide analysis of the percentages increase in performance vs the results that used java compression codecs.

**Item-six:**

Start on P.179 Decomposing a Problem into MapReduce Jobs

Create MapReduce jobs for the two decomposed stages. Use the dataset residing in the Hadoop Directory: **/user/root/ncdc/90-93/** Create the code to run this with two discrete MapReduce jobs (note the execution times and resources used). Then re-write the MapReduce job to use Chaining (on blackboard see listing-5-1 example code) Re-run the analysis and compare execution time and other output data.

Page 183