**Group 9 Members:**

Yiyun Zhang

Michael Karaman

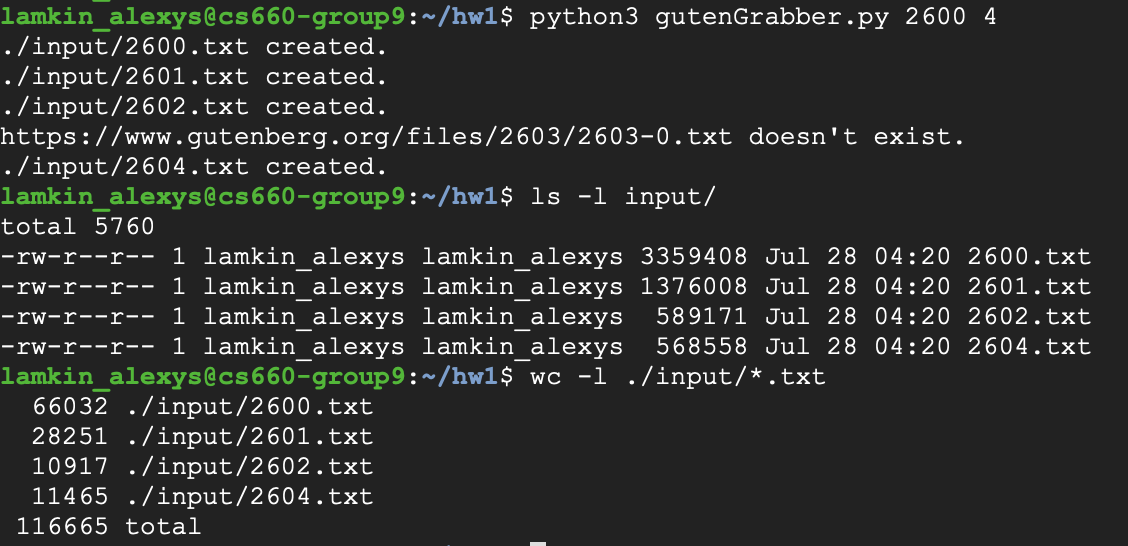
Alexys Lamkin

**Homework 1 Report**

Here is the link to the python implementation of both versions of the inverted index algorithm: <https://colab.research.google.com/drive/1g-qHi0wVuUbJsf-pxCUFO7coeT-gNu1t?usp=sharing>

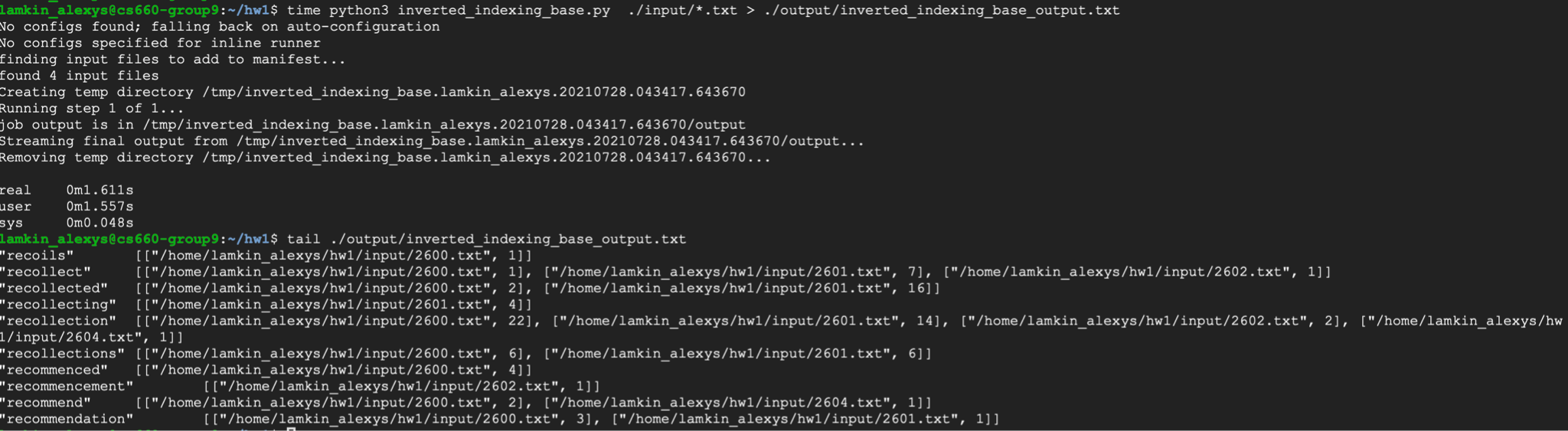
**Generating Input Files**

We first started by generating input files to create our inverted index. We used the 4 books from Gutenberg which totaled 116,6665 lines of text and 1,006,149 words.



**Baseline Inverted Indexing Algorithm**

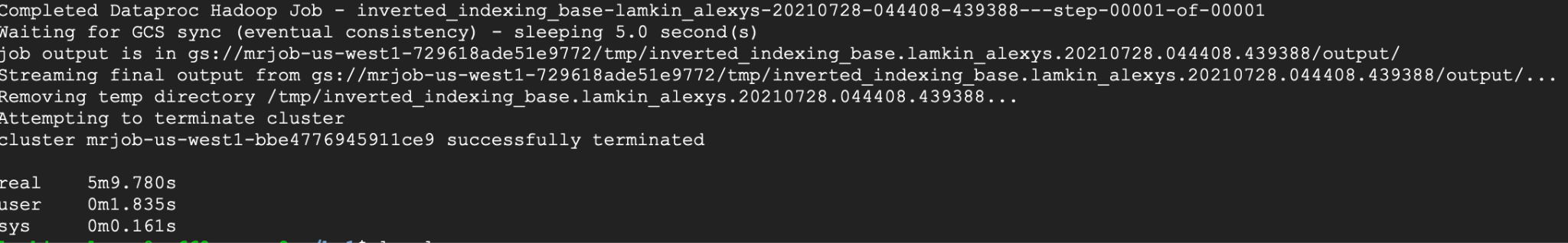
Using the first algorithm from section 4.3 of Data-Intensive Text Processing with MapReduce, without using dataproc, the MapReduce job took under 2 seconds to create the inverted index. Our index consisted of 26,687 keys/unique words, a 97.3% reduction in the overall word count of the input files . Below illustrates five examples of keys and their associated document id + term frequency postings:



We then timed our runs using the non-optimized inverted index algorithm and 2 Hadoop nodes. The run took a little over five minutes:

time python3 inverted\_indexing\_base.py -r dataproc ./input/\*.txt --num-core-instances

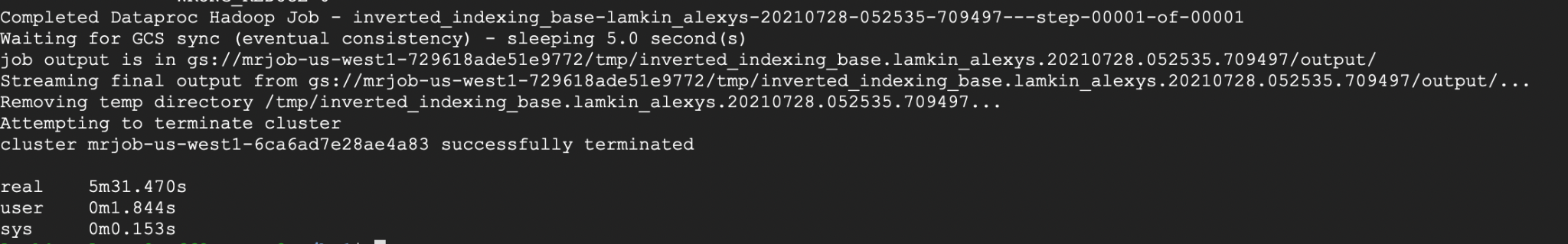
2 > ./output/inverted\_indexing\_base\_output\_2nodes.txt



We compared our first Hadoop run to our second run where we increased the number of nodes to five:

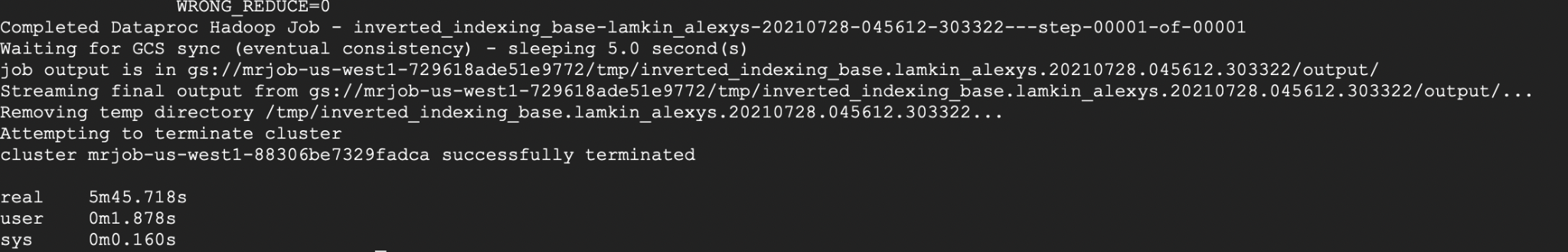
time python3 inverted\_indexing\_base.py -r dataproc ./input/\*.txt --num-core-instances

5 > ./output/inverted\_indexing\_base\_output\_5nodes.txt



Lastly, we compared our first two Hadoop run to our final run where we increased the number of nodes to seven:

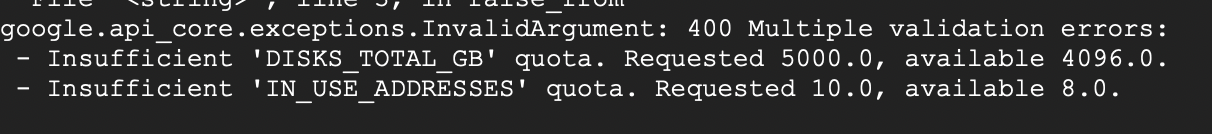
time python3 inverted\_indexing\_base.py -r dataproc ./input/\*.txt --num-core-instances 7 > ./output/inverted\_indexing\_base\_output\_7nodes.txt



Oddly enough, the runtime for all three runs were nearly the same. Though there was a slight increase in runtime when increasing the number of nodes to five and seven, we’d argue that this difference is not significant. In retrospect, a lot of time is spent in organizing the cluster of nodes and starting the jobs within the cluster. Given this, it is hard to measure the true runtime of the MapReduce jobs using hadoop. When using more than seven nodes, we encountered various quota errors:

time python3 inverted\_indexing\_base.py -r dataproc ./input/\*.txt --num-core-instances

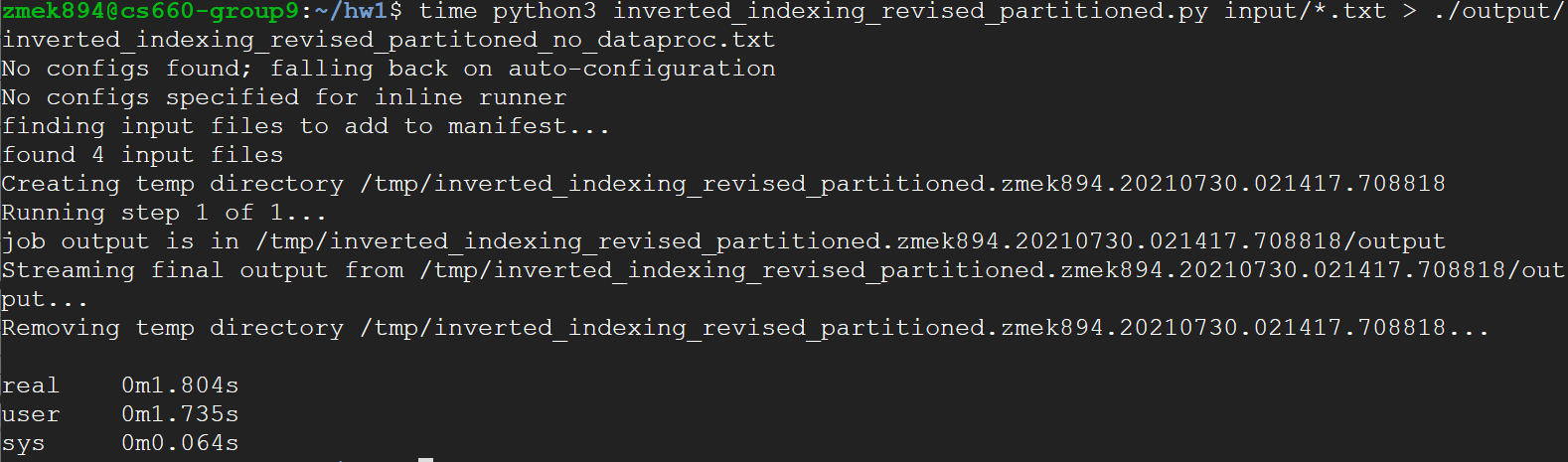
10 > ./output/inverted\_indexing\_base\_output\_10nodes.txt



**Revised Inverted Indexing Algorithm**

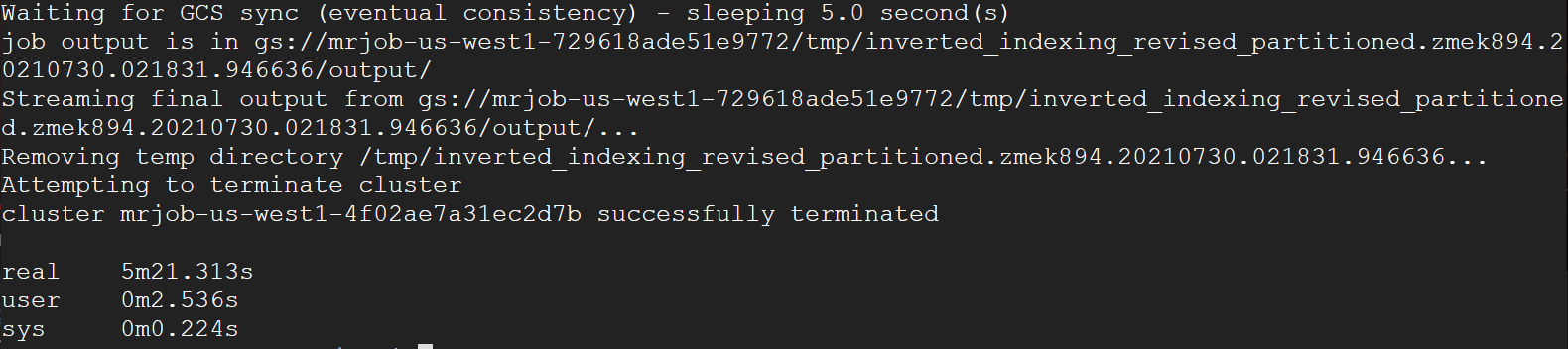
The following tests were done using the algorithm from section 4.4 of Data-Intensive Text Processing with MapReduce. The inverted\_indexing\_revised\_partitioned.py

file from the link at the top of this report was used, and the same input files were used from the previous section. Without dataproc, the job took under 2 seconds.



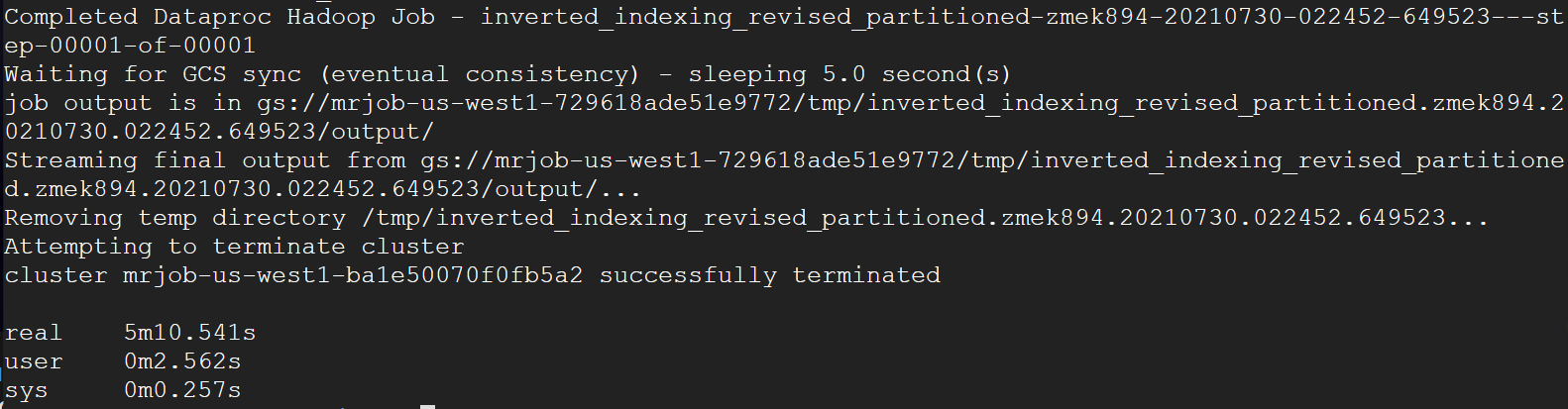
We then timed our runs using 2 Hadoop nodes. The run took a little over five minutes:

time python3 inverted\_indexing\_revised\_partitioned.py -r dataproc ./input/\*.txt --num-core-instances 2 > ./output/inverted\_indexing\_revised\_output\_2\_nodes.txt



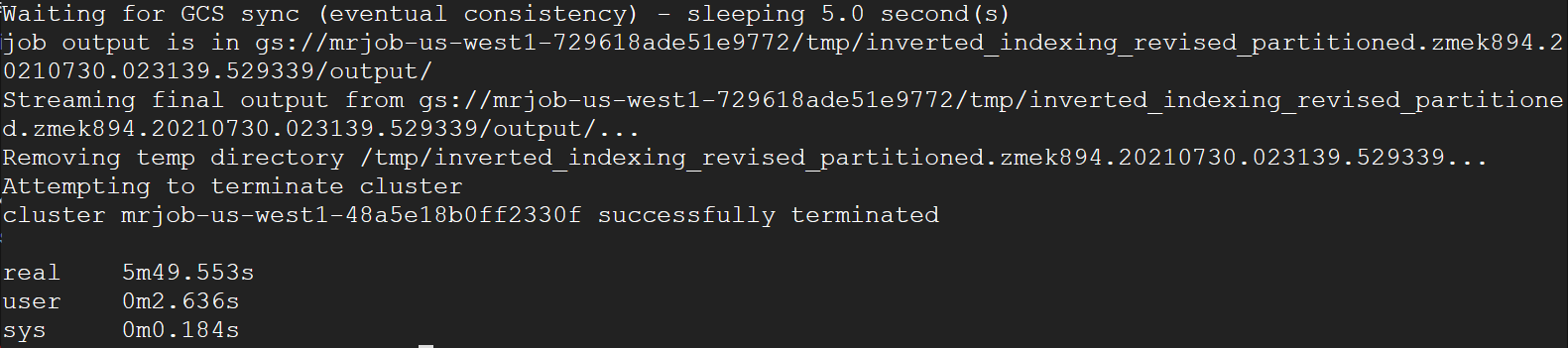
Then on 5 nodes:

time python3 inverted\_indexing\_revised\_partitioned.py -r dataproc ./input/\*.txt --num-core-instances 5 > ./output/inverted\_indexing\_revised\_output\_5\_nodes.txt



Then on 7 nodes:

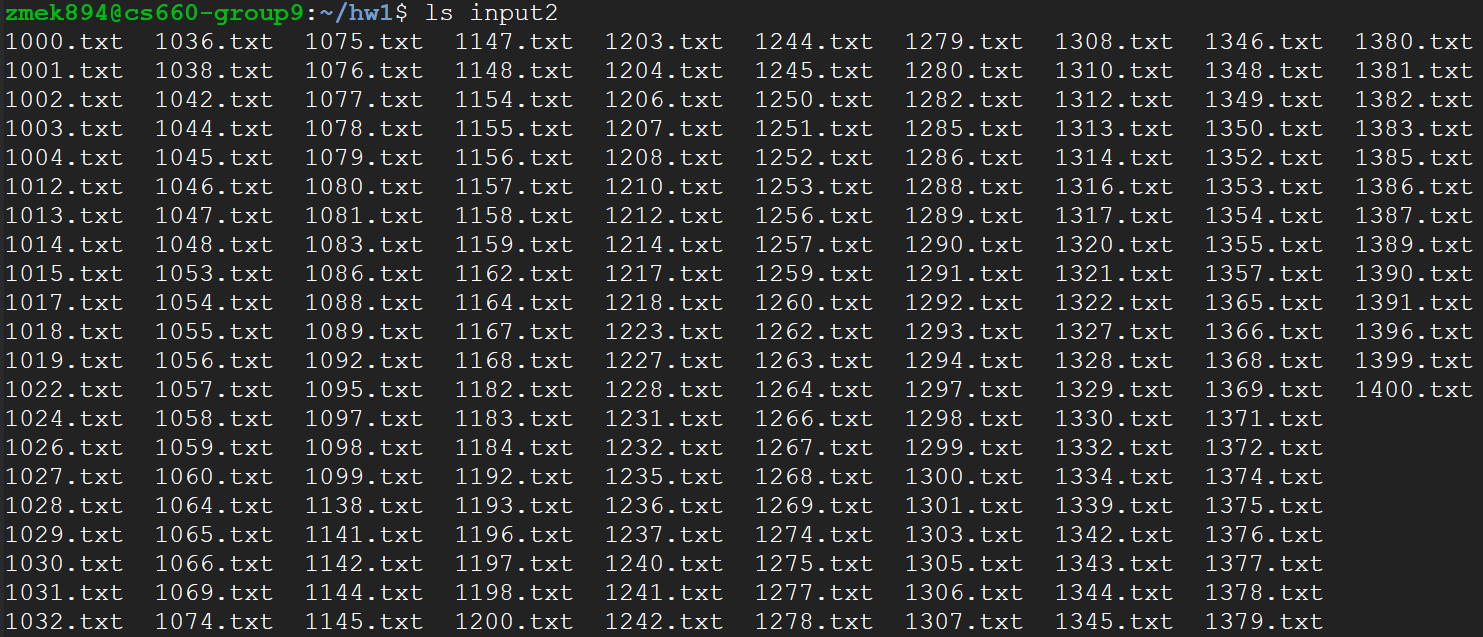
time python3 inverted\_indexing\_revised\_partitioned.py -r dataproc ./input/\*.txt --num-core-instances 7 > ./output/inverted\_indexing\_revised\_output\_7\_nodes.txt



As previously, the job time did not change much with the number of nodes. It seems with the size of the dataset used the overhead cost of setting up the dataproc nodes greatly outweighed the cost of actually completing the job.

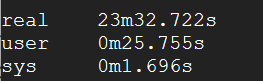
**Larger Jobs**

In order to see the potential benefits a couple of tests were run with 200 books from Gutenberg. The revised algorithm was used with the following results.



On 7 nodes:

time python3 inverted\_indexing\_revised\_partitioned.py -r dataproc ./input2/\*.txt --num-core-instances 7 > ./output/inverted\_indexing\_revised\_output\_7\_nodes.txt



Unfortunately we ran out of time to do more testing on this larger data set, but it seems like at 200 books the time of actual processing outweighs the ~5 minute overhead to setting up the nodes.