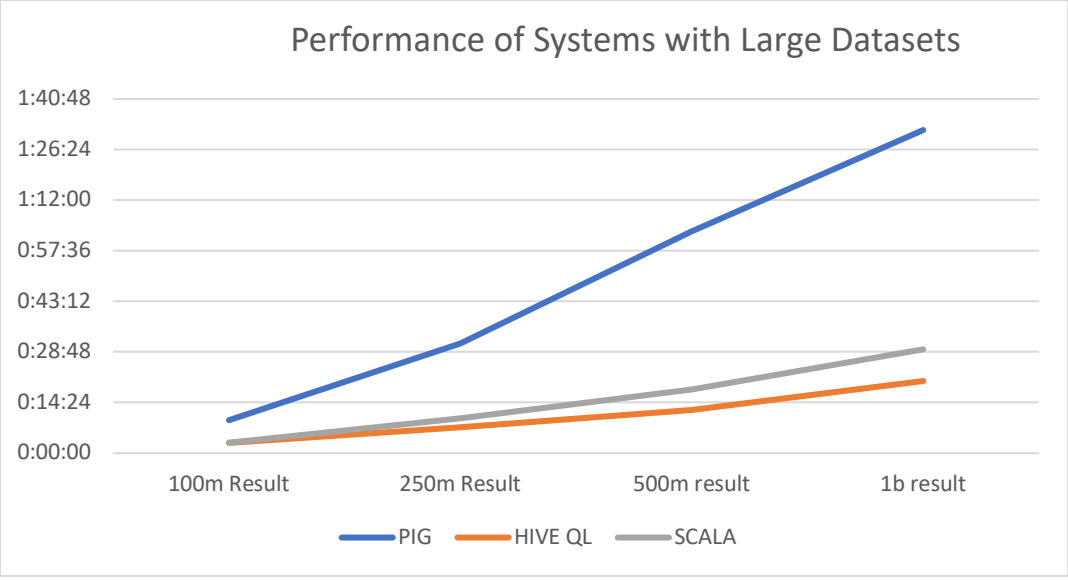
Jeffrey Orcutt
DS 730
Final – Part 3

Description of project: Evaluate the efficiency of three tools in a Hadoop environment in providing basic summary statistics. The tools evaluated were PIG, HiveQL, and Scala and were evaluated on start and stop times with second-level accuracy on subsets of a dataset derived from Gunnar Morling's One Billion Row Challenge[1], which provides a row that contains a city name and a temperature reading. The goal is to read the data, provide the minimum, average, and maximum temperatures, by city, in the quickest method possible.

Testing Environment: After running the code to generate the billion-row dataset, I created 3 smaller testing datasets with 100, 250, and 500 million rows to provide a more robust understanding of the tools' performance. All datasets were tested on a single node Hadoop Cluster inside of a r5a EC2 Ubuntu instance hosted on Amazon Web Services running version 20.04 of Ubuntu. The instance was configured with 100GB of "GP2" SSD storage, 2 x vCPUs and 16GB of RAM. Hive QL and PIG used their native interfaces within Ambari and Scala used Zeppelin Notebook.

Resulting data:

System	100m Result	250m Result	500m result	1b result
PIG	0:09:23	0:31:14	1:03:05	1:31:57
HIVE QL	0:02:56	0:07:18	0:12:16	0:20:32
SCALA	0:02:59	0:09:59	0:18:05	0:29:33



[1] https://github.com/gunnarmorling/1brc

Discussion of Results: HiveQL proved to be the fastest to process the datasets at all tested sizes, followed closely by Scala and standard PIG in a distant third. While monitoring the tests, I used iotop on the instance to monitor the read speed of the various tools. While not accurately measured, I noticed that HiveQL read between 8-10 MB per second, while Scala read closer to 8-9 MB/S, and PIG rarely exceeded 2 MB/S. With single-file datasets ranging between 1.38 GB and 13.8 GB, the individual tools' ability to ingest the data from a CSV file seemed to be the limiting factor rather than the speed of the SSD.

Future ideas for testing: I would have liked to have expanded the size of the cluster to include 2-8 additional nodes to really harness the Hadoop ecosystem. Additionally, I would have liked to have tested the EMR processing power against the dataset. With only minor exposure to the AWS eco-system, I wasn't able to upload the unzipped dataset to my S3 bucket due to the four-hour time limit imposed on us from AWS's Canvas site that would reset the connection. Additionally, I am unsure, but doubtful, that having the data available on s3 would have made any difference in terms of the performance of the tools tested but it is something I would have like to have explored.