Aayush Joshi, Ben Hinchliff, Gabriel Rodriguez, Tymon Vu
CSC 569
Dr. Pantoja

Lab 4 - MapReduce

MapReduce was a paradigm introduced back in the early days of Google's inception to deal with the parallel processing of its different products. It was greatly utilized for PageRank, Google Maps, and clustering articles. It does as its namesake with working in 2 stages, the mapping stage and the reduce stage. The mapping stage sees the task at hand being split into multiple parts for easier processing across nodes. Where each node then maps the data into key-value pairs. Once the data is mapped, the reducing stage tasks the different key-value files and "reduces" them into a singular instance of all the keys and the number of times the key appears. Couple that with the other methods we've implemented throughout this quarter and we have a rather robust system of nodes that is able to parse and map a ton of files quickly.

Our implementation of MapReduce follows this system by basing it on our previous labs. Gossip and Raft have enabled us to keep track of our nodes, their liveliness, and passing information between them. Our previous lab had us implement leaders through elections but didn't give us much to do when our leaders were actually elected but now with MapReduce, we were able to give our leaders the task of assigning tasks to other nodes and being the source of truth for tasks. With our implementation of MapReduce, we keep track of the state of the different tasks with task_log.jsonl(Pantoja approved this). While also writing down the results of the key-value mapping to files mark mr-tasknum-seqnum to be later used to reduce down further. If any workers die while on the job, we can sweep them under the rug and have another take their place and finish the job. In the case of a leader dying, we take an honorable break to elect another and proceed to finish the tasks, if any are left. Due to the quick nature of the jobs and the small amount of data initially given, we decided to add a few more free books from the Project Gutenberg library to the ./data folder to give us a bit more insight on how the process was running in case we saw any weird behaviors occur.

Note: There was a weird behavior in one of the nodes in the video demo, but the output cleared the sequential diff.