

Intro & design

We implemented a map-reduce like system (a.k.a MapleJuice) on the basis of MP1 and MP2.

Initiation

Our system is designed so that every node can start a maple or juice process. Once we input the command, the node will signal the master node to initiate INIT_MAPLE or INIT_JUICE.

- INIT_MAPLE
 - The master node distributes all the files under the given input directory to worker nodes based on hash partitioning.
- INIT_JUICE
 - The master node reads the results from the MAPLE phase. It then classifies the output pairs by keys and places them into different files according to the keys.
 - The master send distributes these files to worker nodes

Monitoring

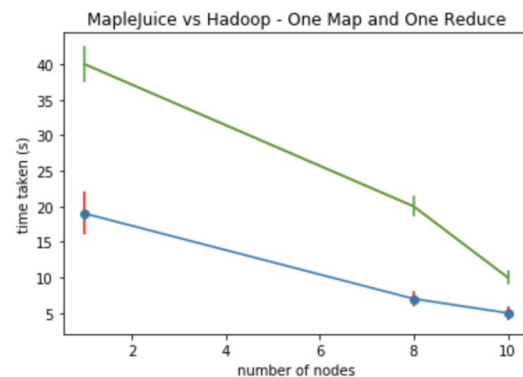
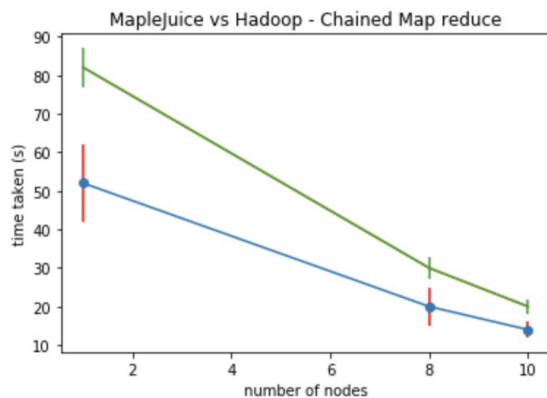
- Once initiated, the master node sets up a channel of ACKs, once it has received the result from a worker node, it will receive an ACK from the channel. It will continue to the next step until all worker nodes that are distributed with a job had returned an ACK
- Lastly, it combines all the files sent from worker nodes to a single file which marks as an end of a phase

Worker Nodes

The worker nodes are assigned with MAPLE or JUICE jobs

- MAPLE
 - The worker will receive a list of input files from the master, it will initiate a GET on every single of them. After that, it will execute the application based on each of the files and combine the results into a single file.
- JUICE
 - The worker does similarly in the JUICE phase as it does in the MAPLE phase

Comparison graphs against hadoop



The input files are around 100 megabytes for each of them. We record the time that Maplejuice takes from receiving INIT_MAPLE/JUICE to the end of the phase.

We did two applications, the first one was the voting set problem from hw1's problem 2. We generated input of 300k pairs of votes, which sum up to be 100 megabytes. We can see that Maplejuice slightly outperformed Hadoop, due to the fact that we have a simpler process, thus involving less overhead. We only record the time which is the summation of times that solely spent for each phase, such that we didn't record the time that is taken as we input commands; This might be part of the reasons as well.

The second one was word frequency function that counts all words appearing in the buildings data plus some extra data that we generated. We could notice that MapleJuice outperforms Hadoop even more because it only has one map and one reduce. Generally, MapleJuice could outperform if the application is simple enough.