CS425 MP4 Report

Group Members: Sarthak Chakraborty, Raunak Shah

Design

We build on top of our flat distributed filesystem of MP3. When the client receives a message to perform a sql query, it sends a message to the leader, which starts the maple phase. Client also sends the number of maples and juice tasks to execute along with the sql query. The data is partitioned into “num_maples” files and saved as temporary files. A task is created corresponding to each file and enqueued in a queue. Tasks are scheduled at workers randomly. Once a worker completes a maple task, it saves intermediate file output to the SDFS and sends an ack to the leader, which dequeues the task from the queue.

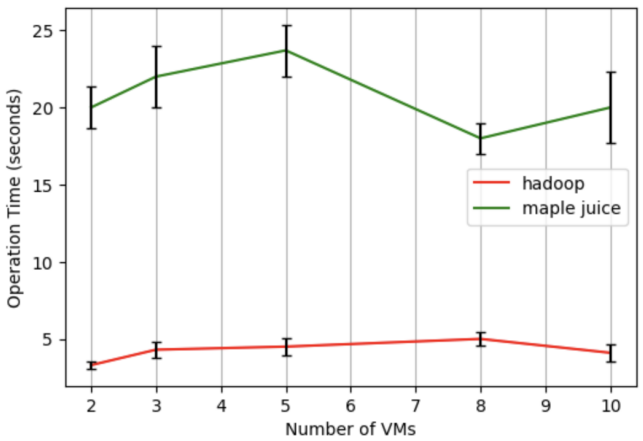
Once all maple tasks are complete, the juice tasks are enqueued into the queue. Each juice task at a worker processes the relevant intermediate files created after the maple step, and the output is appended to a file in the sdfs. Once the leader receives confirmation that all juice tasks are complete, it sends messages to delete all intermediate files that were created in this process. It also sends an ack to the client, which prints a confirmation message.

Failures:

When a failure occurs, any maple or juice tasks that are in progress get aborted (via try-catch for errors). Since the task did not complete successfully, it is added back to the task queue. We make sure that we set the replication factor such that we can support the maximum number of failures that can be expected to occur (in this case, 3, since the demo had 2 failures). Additionally, we make sure that we wait a certain amount of time before scheduling the next task (until re-replication is complete) to avoid potential errors and reducing load on the system. Finally when the maple task is run again, we make sure that it is scheduled on one of the running replicas.

Filter - Simple (VMs vs time taken)

Query - Select * from Table where Interconne = Radio



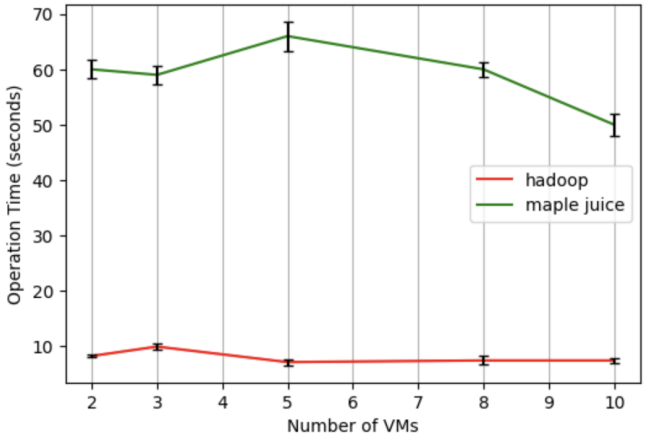
We proportionally increased num_maples and num_juices as we increased the number of VMs. For MapleJuice there is a significant overhead in reading, writing, and deleting intermediate files, which takes up a significantly large amount of time and makes it slower than hadoop (a significantly more optimized system)

Because of this, we chose to use the traffic dataset since it kept the time of our method reasonable and allowed comparison with hadoop as well.

As the number of VMs increases, it is easier to parallelize maplejuice tasks, however there is still significant overhead due to read/write operations. We believe this makes the overall trend less steep.

Filter - Complex Regex (VMs vs time taken)

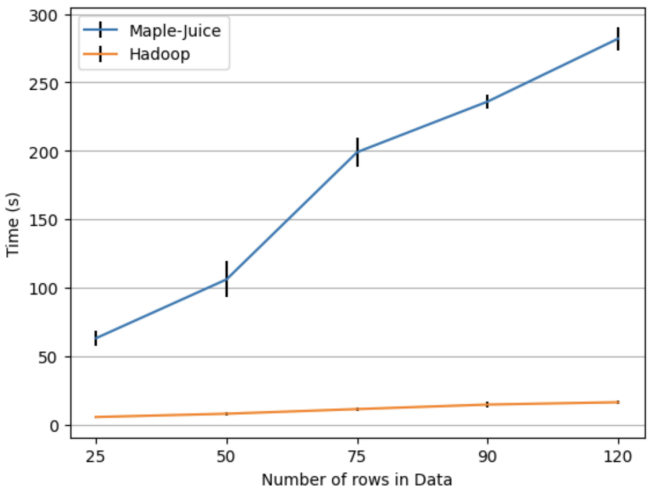
Query - Select * from Table where Detection = ^Video$ and Interconne = ^.*Radio$



The time for the complex regex is greater than the time taken in the simple case for both hadoop and maplejuice, as expected. The overall trend vs number of VMs remains similar as the simple filter case for the same reasons as above.

Join - Simple (Data size vs time taken)

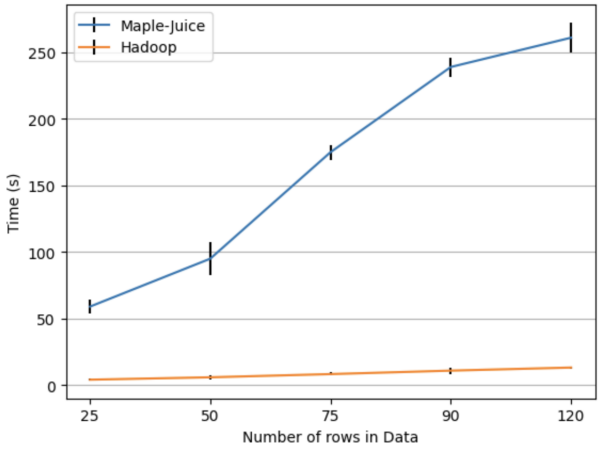
Query - Select * from traffic.csv as D1, traffic.csv as D2 where D1.Interconne = D2.Detection_



We implemented a Reduce-side join for this part. We see from the below plot that the time taken for completing a join query increases with the size of the data.This is because more keys are generated at the Maple phase and hence the Juice phase needs to execute with more number of keys as the data size increases. Even so, the latency of Hadoop increases with data size. However, hadoop processes the data faster than Maple Juice because it is more optimized. Maple Juice takes more time to write and read from the SDFS.

Join - Complex (Data size vs time taken)

Query - Select * from traffic.csv as D1, traffic.csv as D2 where D1.Interconne = D2.Detection_ and D1.Interconne != Fiber



We implemented Reduce-side join for this part as well. Compared to the simple query, complex query includes a new condition which adds a filter to the Interconne column which reduces the number of rows maple produces and hence the number of keys to be processed at the Juice phase.