

Design:

Our MapleJuice implementation is made up of two types of nodes, a master node and several worker nodes. The master is the node that receives the work requests, and distributes the work to the various workers. Once the master receives a work order, it creates a number of jobs as prescribed by the input parameters (num_maples or num_juices), and then runs the assigned jobs. If the master node is given a second task before the first set of tasks are complete, it queues the assigned tasks and runs them as tasks complete and machines become available. The master node does not overlap assigned jobs, rather, it runs a second job only after all tasks for the first job have completed. Failure detection is done by any worker, however, once failure is communicated to the master, besides updating local membership lists, the worker is done with that aspect of failure detection. If the master becomes aware of a failure, it will reassign/reschedule any jobs currently assigned to the failed node, and run that task on another node. If the master becomes aware of an addition to the cluster, it will reshuffle its waiting tasks and allow for use of the new addition in processing the queued jobs. We assume that the master will not fail for this purpose, though in order to implement this we might have created a task queue on the worker side that would hold the list of tasks to complete, which the master would send, rather than holding the queue of jobs internally. We could then elect a master on another machine in a similar way to our implementation for MP3.

Our submission for sdfs on MP3 was flawed, and as such attempted to use the recommended MP3 given on the course webpage. We however struggled with installation of Google's protobuf library, and eventually abandoned that course of action. Because of this concern, we created a pseudo-sdfs where all submitted files are backed up on one specified VM, and distributed from there to the various maple and juice jobs. However, our original plan was to implement our MP3 as the backbone connection for this assignment, utilizing the failure detection, leader election, and sdfs capabilities in that assignment to streamline the development of our MapleJuice implementation.

When running Maple, our master node partitions the input and creates nearly-equivalently sized maple files, which are stored on the sdfs while running. Each maple worker is assigned one of those input files, and creates several output files as specified. When Maple completes and Juice begins, those intermediate files are loaded with filename prefixes created from Maple execution, and processed according to the specified executable. Both numbers of Maple and Juice jobs are specified. Juice tasks receive several keys with their requisite prefixed filenames as determined by the master node using range-based or hash-based partitioning, complete processing those files, and create the sdfs output file. We have an option to delete or to not delete the previously mentioned intermediate files.

We wrote only the Wordcount application for this task, and did not have time to test this to obtain the requested plots.

Hadoop: We did not manage to get Hadoop running on our cluster, and as such have not tested it.

We have no plots, and therefore this discussion area is almost blank.

As an aside, this task was markedly more difficult than we expected, given our deficit in our MP3 application and the difficulty in using the provided MP3 application and Hadoop installation as well. While we admittedly could have used our time more optimally, the aforementioned difficulties combined with the larger scope of this project and tighter work-schedules as the end of the year neared handcuffed our efforts substantially. While we understand that the requirements were clear and specific, we can only ask for leniency and understanding.

Despite our struggles with the 4th-credit projects, I very much enjoyed this class, it was fast-paced, interesting, and extremely informative. That said, the difference in time requirements between the 3-credits and 4-credits versions of this class was similar to the difference between taking another class and not taking that class, which I was not prepared for.

Happy holidays, and farewell.

Joe and Siddarth