

Design:

We implemented a Hadoop style distributed framework called MapleJuice. It has two stages - Maple and Juice, just like MapReduce. We integrated MapleJuice with SDFS from MP3, Gossip Style membership management system from MP2, and Distributed Grep system from MP1.

Architecture:

Executables: We implemented the plugin package from Go which writes the stimulation stage of Maple and Juice into .so shared objects. For the Maple Executable, it basically converts input data into desired key-value pairs. For the Juice Executable, it loads in the output files from the Maple stage and perform operations as needed.

Resource Manager: The leader node from MP3 will play the role of the Resource Manager. It will call the executables and partition the tasks to the worker nodes. It will also manage the task status

Workers: Nodes that are assigned with tasks by the RM.

Workflow:

Maple: User input commands. The resource manager processes the command and call the Maple Executable to generate num_maples input files. The resource manager then partition these input files to all the alive workers using gRPC. After each worker node receives the Maple request, it will first get the input files using SDFS and load these files as key-value pairs. Next, it aggregates the data based on the key, in this case, is the 'objectid' for each row of the dataset. For each key, the worker node outputs a file with user input prefix name and sends back a list of output filenames.

Juice: User input commands. The resource manager processes the command and split the file list from the Maple stage into num_juice groups. Then, they are partitioned and distributed to all the alive workers using gRPC. After each worker node receives the Maple request, it will first get the input files using SDFS and load these files as key-value pairs. Then, it will process the data

based on user input SQL command. Finally, it will output merge the result output and write to destination directory. It finally sends back a list of output file names to RM.