

CS-425 MP4 Report Group No: 44

Lin Lyu(linlyu2), Ching-Hua Yu(cyu17)

I. Design

Architecture:

File Server: This part is based on MP3.

Maple JUICE: Maple juice has exclusive ports to complete tasks and receive related messages. For maple, a server will first set up a maple task and ask the master to coordinate. The master get the information and assign the maple task to n servers(n is entered by users). Each one of the n servers will get part of the input file from SDFS and call the application maple task. When they finished, they will upload those intermediate results to SDFS in case the server fails in the future. When master get the information of 'FINISH' from all n machines, it will indicate that it's ready to juice. For juice, we use the same machine as those in the maple task. The master will tell those machines to start to juice and those machines will get the intermediate files with their index from SDFS and aggregate the result in the way that suggested in the application.

Master: Master is in charge of coordinate file transmit tasks as MP3 did. Besides, the master is responsible for assigns maple and juice work to servers. Take the juice results of each server with juice tasks and aggregate them into one single file.

Applications: Applications are named after its functionality such as wordcount.h. They work by implementing API listed in the programming framework below in maple_juice.h.

Programming Framework:

MapReduce APIs:

- maple(): Take the contents and perform Map specific to the application. Files whose name contains key are generated and their names will be returned.
- juice(): Take the files and aggregate the result.

II. How MP1 used in M

III. P3