Yang Liu (liu310) and Gohar Irfan Chaudhry (gic2)

CS425 – MP1 Report

(Note: A detailed explanation of each file and the input/output formats is written in the code files and a README.md in the repository.)

We use a modular design in our system to keep all different functionalities separate from one another. All modules communicate by message passing over TCP sockets.

The system has two main modules – one is the **Daemon** and the other is **Grep**. Daemon is the back-end system that runs on each node in the cluster. It has one sub-module (called **Server**) that waits for queries from other nodes. The Daemon also has one module (called **UserHandler**) that listens to requests by the client program (called Grep) and relays the request to all other Daemons in the cluster to get their responses and combines them into one file.

The Grep program is the frontend that a user will execute to search for a pattern in a specified directory. Grep will send this request to the Daemon running locally on that node which will spawn N threads (total number of nodes in the system) to fetch responses and will combine them and alert the Grep program of job completion.

We have written our system in Java and used Maven as the build tool to for compilation and running tests. The tests consist of three categories; basic, complex with no output and complex with long output. All these are part of the build pipeline so whenever any changes are made and compiled in the codebase, all these tests must pass to ensure nothing breaks. Each of these tests spawns a specified number of Daemons on different ports on the same physical node to test that everything in the distributed environment works and the outputs are as expected. The average query latency when 4 machines each store 60 MB log files is approximately 7.5 seconds.

(*Fig 1 shows the topology of the system. Fig 2 shows the architecture of program*)