**CS425, Distributed Systems: Fall 2015**

**Machine Programming 1 – Distributed Log Querier**

The Distributed Log Querier consists for four main components:-

|  |  |
| --- | --- |
| **Class** | **Description** |
| Catalog.java | Stores information about the hosts, including host name, IP address and port number. |
| RemoteGrepClient.java | Client program that sends `grep' command to servers, and output matched lines received from servers to standard output. |
| Server.java | Server program that takes the socket, binds and listens on it. It also calls execute function of Grep class to run grep functionality. |
| Grep.java | Main class to implement grep functionality along with all its options(including regexps,etc.) |

The RemoteGrepClient and Server are both deployed in all the machines of the distributed system. The Client program can run the grep command from any of the machines and query the logs present on all the other machines and display the results on its terminal.

Client and server connection is established using socket programming.

Apache Commons CLI library is used for parsing command line options passed to programs. JSON is also used to send string array using json format in case the string has quote or space.

The MP is implemented in Java.

**Catalog**

Catalog stores information about the whole system. It consists of class Host that stores information about all the machines present in the distributed system including the host name, ip address and port number.

The details of the machine is stored in “conf/host\_list” file.

**RemoteGrepClient**

RemoteGrepClient is a client program which sends `grep' command to servers, and output matched lines received from servers to standard output. The usage of RemoteGrepClient is exactly the same as Grep, except that at least one file should be given (i.e. cannot read from standard input, which does not make sense in distributed settings).

**Server**

This is the main class for Server. Every machine in system runs an instance of Server class. Whenever a request is received by the server, the Server accepts the connection and spawns a thread (an instance of ServerThread) to handle the request. This worker thread is responsible for handling the request from the client. It receives the arguments (grep options) as a args object from the client and performs the grep operation on the file “machine.i” where i is the host name of the server by calling the execute function of Grep class. The result of the grep operation is sent back to the client over the client socket. The data is streamed and not kept in the memory. Hence the memory usage of the server and the worker threads is minimal.

It is assumed that the message the servers receives from the client is valid and that any mistakes should be detected at the client side itself.

**Grep**

Grep is a utility which is like UNIX command: egrep. This is the main class which implements the grep functionality.

The command line syntax for performing grep is: **java Grep [-options] [pattern] [file ...]**

To find which options are available, printHelp() can be called.

**Unit Test Case**

GrepTest.java contains the test case to verify that the distributed grep functionality produces the desired outputs for appropriate inputs.

**Performance**

**100 MB log file**

**Frequent-**

**Avg. Query Latency**

**Somewhat Frequent-**

**Avg. Query Latency**

**Rare-**

**Avg. Query Latency**

**Very Rare-**

**Avg. Query Latency**