

## CS425/ECE428 Distributed Systems MP1 report

James Hsu(chsu41), Zitong Chen(zitongc2)

### Design:

We implemented the distributed grep using socket-programming techniques, implemented in Go. It is implemented as a client-server model such that the VM's exchange information by establishing a transport layer connection. If we run "server.go", the VM will be listening on port 3000 for incoming UDP connection. Upon receiving a connection, the server will run grep, parse the information, and send it back to the "controller" that initiated the request. In "controller.go", there is an array of hardcoded ip addresses of all the VM's. The program will simply iterate through all the ip addresses, and establish UDP connection with each one of them to query the logs. Note that we chose UDP over TCP to reduce the transmission overhead.

### Unit tests & Analysis:

For the unit tests, we wrote a script to randomly generate log files around 60MB and placed them into respective VM's. In the logs, there are four hardcoded patterns ("secrethahaha", "onlyfew", "rarestring", "treasure"), with descending frequency of occurrence in the same order. We thought that the average query latency will increase as the pattern occurrence frequency increases, but the data shows it is actually the opposite.

	secrethahaha	onlyfew	rarestring	treasure
run 1	55.862	70.623	62.549	67.292
run 2	50.756	68.865	57.642	63.681
run 3	50.418	65.504	59.17	65.044
run 4	50.422	66.024	58.46	64.936
run 5	57.228	67.854	56.844	66.218
avg	52.9372	67.774	58.933	65.4342
stdev	3.331503144	2.093375	2.201474	1.372947

