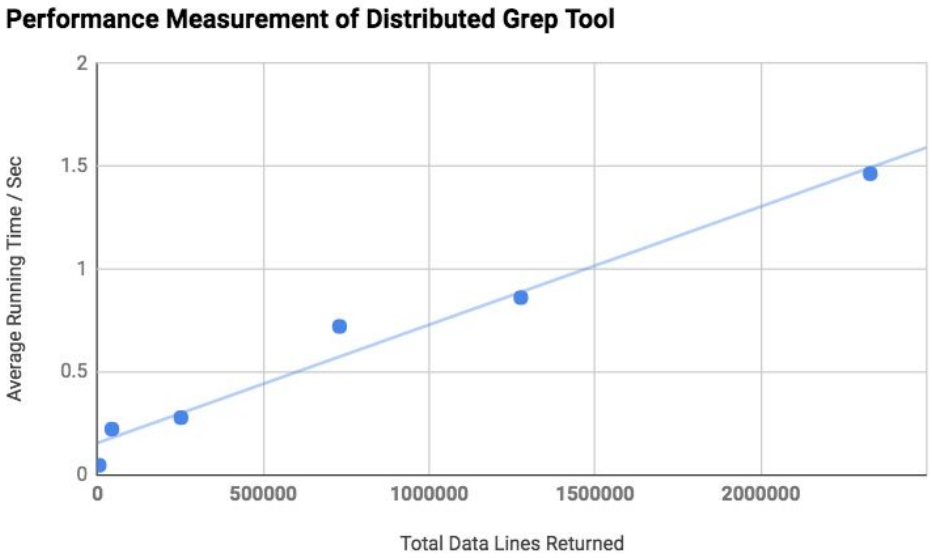
CS-425 MP1 Report

Group No: 59.
Member: Jialin Liu(jialin2), Zijun Cai(zijunc2)

Measurements:

We perform several queries with different workload, each with 5 times, and get the average result. The plot is shown in below



Design:

Since the overall requirement of this project is to grep contents from multiple nodes, the planning can be split into two separate parts. One is to grep content on single node, the other is to build reliable communication mechanism from one master to N slaves (N<=10 in this project).

There are two ways to achieve the functionality of grep on single machine. The first is implement the file pattern match on your own, the second is to call the shell command to get the result. In practice, we choose to use the second one, since it saves us from massive coding workload and have a better performance guarantee.

In the master-slave communication part, we choose to use the classic client-server design. That is, on each virtual machine there is a socket server daemon running, listening for a specific port. The distributed-grep is implemented as an client, to issue grep request to these servers in parallel. Once a request is caught, the server forks a new process to handle this event using popen function.

Unit Tests:

The test program firstly generates random log to each vm and secondly compares the result of grep and distributed-grep with words in different frequency, like "HTTP"(frequent), ".com"(sometimes), "uffba.msfyqmhq.xzvqj.elc.net"(rare) and etc.