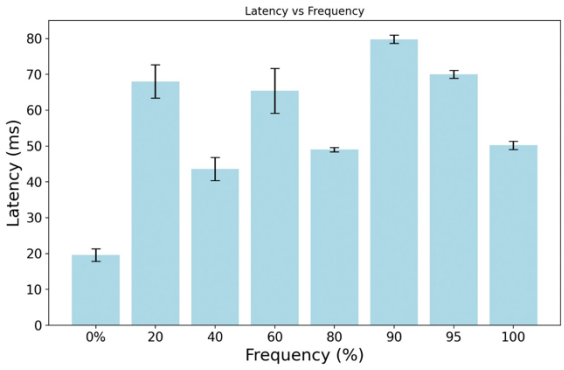
CS425: MP1 Report
By Sagar Abhyankar (sra9) and Aditya Kulkarni (aak14)

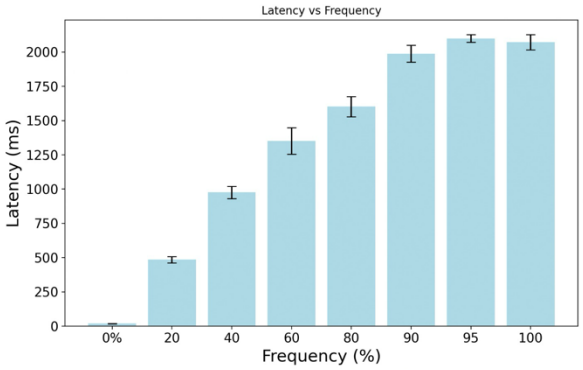
Architecture Design and Implementation:
The distributed log querier is written in go, implemented using socket programming and message passing. The topology used is full mesh. Each client program functions both as a client and a server, eliminating the need for a centralized leader. Each message type uses a message header such as "CONN INIT" or "PEXCG" for easier message structuring and identification. For fault tolerance, we have implemented an alive acknowledge system which runs right before every grep multicast. Machines that don't respond within a static 1-second delay are excluded for that particular grep round but may rejoin in future rounds. Once a query is executed on any of the client programs, the program (querier) runs the multicast and local grep command parallelly using go routines, then the querier waits for all the alive peers to respond, it then prints the latency, matching line counts for each machine, and the cumulative total matches to the terminal. All the matching lines of the latest run are also stored in an output txt file.

Unit Tests:
The unit tests written cover full mesh connection setup, execution of different types of grep patterns, frequent and infrequent, with and without regex. generation of log files on distributed VMs and subsequent verification of matching line counts. This also includes coverage of fault tolerance behavior, i.e. whenever a VM does not respond or has crashed. Tests were written using the "testing" package from go.

Latency Discussion:



(1)



(2)

The plots show the average latency (in milliseconds, 5 trials per data point) versus the percentage of matching lines across all log files (provided, 60MB each) in a setup with 4 VMs and 1 querier. The left plot depicts query latency using the -c (count) flag in the grep command, while the right plot shows latencies for the same patterns run without -c flag. We infer from the upward trend in plot 2 that the increase in latency is directly proportional to the amount of data transmitted and processed over the network, largely independent of the percentage of matching lines, except for minor fluctuations (~10-20ms) in plot 1 that maybe due to network congestion/delays. For reference, a 0% match indicates pattern has no matching lines, while 80% corresponds to around 870,000 matching lines (out of 1,091,212 total lines across 4 VMs). We conclude that performance of a distributed log querier system largely depends on the log file sizes and the network speed.