CS 425 MP-1 REPORT

Simrita Jayasimha & Arjun Rajesh Kenath Panikkath

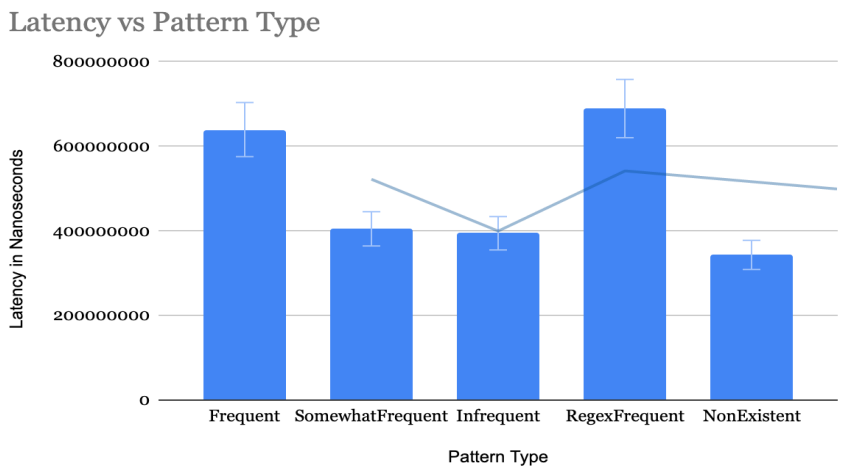
DESIGN & IMPLEMENTATION

1. The architecture consists of 3 entities: Client, Coordinator, Worker. Communication between these is done using RPCs. Implementation done using Go.
2. The client executes a grep command by contacting the coordinator (the first available non-faulty one).
3. The coordinator takes this input command and makes a call to each of the workers in the VMs (concurrently using goroutines).
4. The workers execute the grep command and return the output to the coordinator.
5. The coordinator collects these outputs, calculates the total number of successful matches (from non-faulty nodes) and returns the final result for the client to display.

TESTING

The test client assumes VM2 as the coordinator and establishes the RPC connection. The tests are then run across all VMs against the given log files and are marked as successful as long as the outputs received from non-faulty nodes are as expected. Test cases covered are for all of the pattern types.

OUTPUT ANALYSIS



1. Latency is directly proportional to the number of matches in the log files. This is as expected since it takes slightly longer to sum up the match counts when there are more matching words.
2. Regex matching takes longer than exact matching. This is expected since matching a word as a substring for eg, is more computationally intensive than looking for exact matches of a word.
3. The deviation across each of the patterns increases as the frequency of the patterns increase. This is expected since matching of each word has variability in the computation time, hence the deviation scales with the number of matches.