# CS 425 MP1 Report
## GA10 - Sunveg Nalwar (snalwar2) and Sudarshan Shinde (sshinde5)

**Design:**

Coordinator: reads *cluster.properties*, fans out concurrent RPCs (one goroutine per worker), server-streams responses, aggregates:

- lines mode: print lines as they arrive.
- count mode: sum per-worker counts into TOTAL_COUNT.

Worker: shells out to grep over matched files (-glob within -logdir), streams lines or returns summed count.

Fault tolerance: failures logged; coordinator continues with remaining workers (partial results allowed).

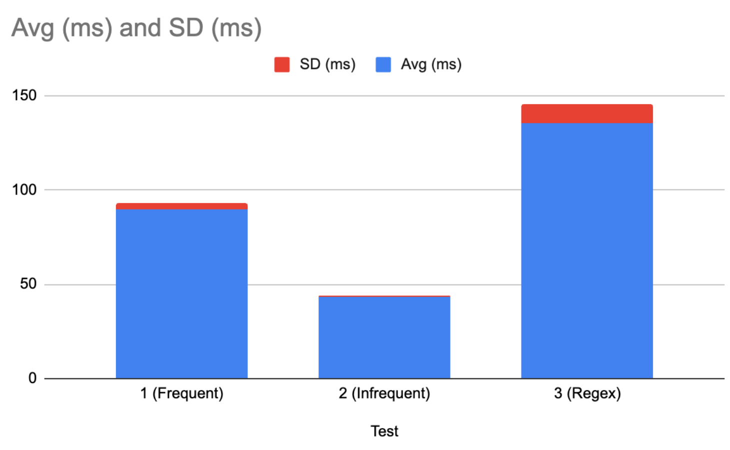
Concurrency: fan-out/fan-in with WaitGroup; per-worker stream is blocking within its goroutine; across workers it's parallel.

**Unit tests:**

Correctness: local grep vs distributed results (frequent, infrequent, regex; lines and counts).

Edge cases: empty matches; single matched file; multiple files; invalid glob.

Fault case: kill one worker; verify remaining counts and TOTAL_COUNT match sum of healthy workers.



| | Frequent | Infrequent | Regex |
|---|---|---|---|
| **Test 1 (ms)** | 88 | 44 | 131 |
| **Test 2 (ms)** | 92 | 44 | 138 |
| **Test 3 (ms)** | 94 | 42 | 154 |
| **Test 4 (ms)** | 85 | 42 | 128 |
| **Test 5 (ms)** | 91 | 43 | 125 |
| **Avg (ms)** | 90 | 43 | 135.2 |
| **SD (ms)** | 3.16 | 0.89 | 10.34 |

These tests were averaged out on 10 machines and below are out observations:

Regex queries are slowest (avg 135 ms, SD 10.3) due to more expensive matching, with noticeably higher variability across runs.

Frequent substring queries are slower than infrequent probably because more matches increase grep work and streaming/aggregation overhead.

Variance is small for infrequent/frequent patterns, indicating stable performance; most jitter appeared only in regex workloads.

All workers have around the same time_elapsed. This denotes all workers run grep in parallel