

Discussion 10: Distributed Systems

April 25, 2025

Contents

1	Distributed Data Processing	2
1.1	Concept Check	2
1.2	Leetcode golf	2
2	Distributed Systems	3
2.1	Concept Check	3
2.2	Two Phase Commit	3

1 Distributed Data Processing

1.1 Concept Check

1. True or False: Completed map tasks are re-executed when their worker crashes.

2. True of False: User tasks must be idempotent for MapReduce.

3. How does a reduce worker find and read the relevant KV pairs on workers' disks?

4. What are some performance benefits of lazily evaluating transformations?

1.2 Leetcode golf

Suppose we downloaded a file of all solution statistics from Leetcode. Each file is a comma-separated list of entries formatted as {key: PROBLEM_NAME, value: CODE_LENGTH} (e.g. {"2SUM", 162}). Implement the following MapReduce program to find the code golf champion for each problem!

Note: Code golfing is a competition to solve a given problem with the shortest source code possible.

```
map(key problem_name, value code_length) {  
    Emit(_____, _____);  
}  
  
reduce(key problem_name, List<value> length_lists) {  
    lowest = _____;  
    for (value length : length_lists) {  
        _____;  
    }  
    Emit(_____, _____);  
}
```

2 Distributed Systems

2.1 Concept Check

1. The vanilla implementation of 2PC logs all decisions. How could 2PC be optimized to reduce logging?

2. 2PC exhibits blocking behavior where a worker can be stalled until the coordinator recovers. Why is this undesirable?

2.2 Two Phase Commit

Consider a system with one coordinator (C) and three workers (W_1, W_2, W_3). The following latencies are given for each worker.

Worker	Send/Receive (each direction)	Log
W_1	400 ms	10 ms
W_2	300 ms	20 ms
W_3	200 ms	30 ms

You may assume all other latencies not given are negligible. C has a timeout of 3 s, log latency of 5 ms, and can communicate with all workers in parallel.

1. What is the minimum amount of time needed for 2PC to complete successfully?

2. Consider that all three workers vote to commit during the preparation phase. The coordinator broadcasts a commit decision to all the workers. However, W_2 crashes and does not recover until immediately after the coordinator's timeout phase. Does this transaction commit or abort? What is the latency of this transaction, assuming no further failures?

