6.824 2021 Midterm Exam

This is a 90-minute exam. Please write down any assumptions you make. You are allowed to consult 6.824 papers, notes, and lab code. If you have questions, please ask them on Piazza in a private post to the staff, or ask in the Zoom chat if you are taking the exam during lecture time. Please don't collaborate or discuss the exam with anyone.

In particular, please do not discuss the contents of the exam with anyone at all besides the course staff until at least 3:00 PM EDT on Friday, April 2nd, to allow students in other timezones to take the test.

MapReduce

In lab 1, a worker in the MapReduce library doesn't hand out any reduce tasks until all map tasks have completed and have produced all their intermediate files. The implementation described in the "MapReduce" paper by Dean and Ghemawat allows reduce workers to start reading intermediate files *before* all map tasks have completed.

A. Briefly describe the advantage of allowing reduce workers to start reading intermediate files early.

B. Describe how you would modify your lab implementation to allow reading of intermediate files early, as Google's MapReduce library does, while maintaining correctness. (You don't have to write code, but sketch out a design in words. Keep your answer brief.)

MapReduce (Lab)

Zara Zoom has implemented MapReduce for 6.824 Lab 1. Their worker code loops, and asks the coordinator for tasks using a `GetTask` RPC call. Their coordinator code to handle `GetTask` RPCs and assign tasks to workers is as follows:

```
func (c *Coordinator) HandleGetTask(args *GetTaskArgs, reply *GetTaskReply) error {
    c.mu.Lock()
    defer c.mu.Unlock()

    for {
        // iterate through tasks that aren't done
        for i := c.numCompletedMapTasks; i < c.numTotalMapTasks; i++ {
            if c.mapStarted[i] == false || time.Since(c.mapWhen[i]).Seconds() > 5) {
                c.mapStarted[i] = true
                c.mapWhen[i] = time.Now()

                reply.TaskType = MapTask
                reply.TaskNum = i
                // fill in rest of GetTaskReply fields

                // remember that we assigned a task
                return nil
            }
        }

        // if we've completed all map tasks, break out of the loop that
        // assigns map tasks
        if c.numCompletedMapTasks >= c.numTotalMapTasks {
            break
        }

        // if we got here, all map tasks have been started,
        // but some have not finished.
        // wait for either a map task to complete,
        // or for some timeout to occur to wake us up.
        c.cond.Wait()
    }

    // all maps have completed. move on to reduce phase.
    ...
    // when all reduces have completed, tell worker to exit with a
    // DoneTask.
    reply.TaskType = DoneTask
    return nil
}
```

Zara also has their workers send `FinishedTask` RPCs to tell the coordinator when a particular map task has finished.