EXECUTIVE SUMMARY

Assignment overview:

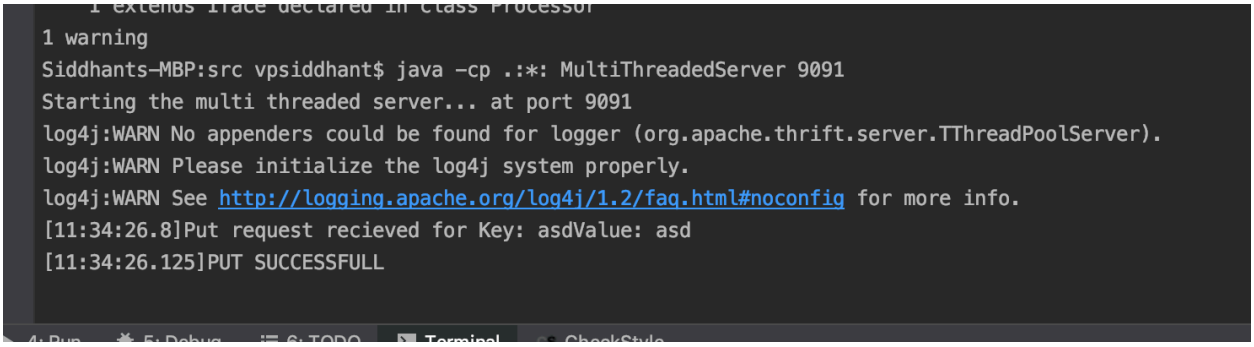
- The main purpose of the assignment was to realize the complexity of data duplication and consensus even on such a small scale.
- The assignment was also to further our understanding of exchange of structured data between multiple PCs using RPCs.
- Designing Interfaces for all the parts of the system was also an important aspect because that helped us allocate different functions to the server, coordinator and the client.
- Understanding synchronization was another important part of the project, because we had to find way to not block a “get” request while a 2 phase commit is happening for a put or delete.
- The assignment also helped us understand the errors that can happen while creating a distributed storage.
- Testing for a single storage is easier, but the testing parameters increase exponentially as multiple servers come into picture and many different failure cases have to be thought out.

Technical impression:

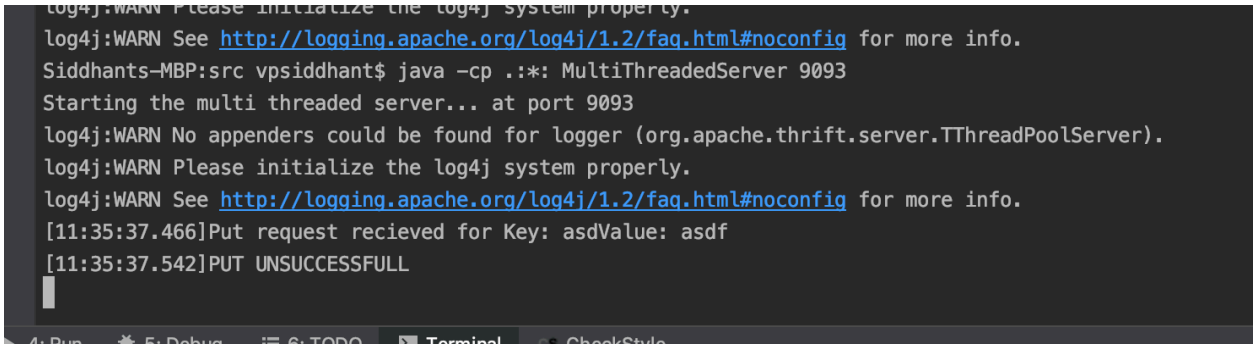
- This project was particularly challenging because many of the requirements seemed ambiguous.
- The first step to approach this problem was to understand the different phases of a two phase commit.
- I read the oracle documentation and the 2-phase commit mentioned in the text book
- But even these resources had little bit of contradiction because the oracle documentation mentioned that there is read lock when a two phase commit starts.
- But one of the objectives given in the project was increasing the bandwidth compared to the last project.
- So I decided to not put a read lock in place during a two phase commit.
- There were multiple ways to do that, but I decided to have a duplicate map just for reading and it get updated only when a two phase commit is successful.
- I decided to use Apache Thrift for this project, because RPCs were an important part of creating this project and Thrift helps make this process smoother.
- Implementing the algorithm, required many design choices.
- Including whether the coordinator should be a separate class or a thread originating from one of the servers ?
- I decided to make coordinator to be a separate class.
- I got stuck multiple times during the project.

- I mainly got stuck when I was thinking of edge cases and implementing them, some part of the system was failing. So I had to correct it along the way.
- Also I made few choices like making coordinator IP static so that while testing it becomes easier.
- I didn't take ip of servers because it was making the number of arguments to the coordinator unnecessarily long without actually increasing the output. I only took unique port numbers.
- Testing:
- I had all 5 servers running and then did a put from one and get from another to the same server.
- I had all 5 servers running and then did a put from one and get from another to the different server.
- I killed one server in between so that all the next put request will be aborted without a live server.
- The server that I killed in last step was revived to show that once the server recovered the put starts working again.
- So I finally decided the system was robust enough.

SUCCESSFUL PUT:



UNSUCCESSFUL PUT:



SUCCESSFUL TWO PHASE COMMIT ON COORDINATOR:

```
[11:34:28.117]Two phase commit completed
doCommit started
log4j:WARN No appenders could be found for logger (org.apache.thrift.server.TThreadPoolServer).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
[11:35:37.523]Two phase commit initiated
can commit for port: 9091
can commit for port: 9095
[11:35:37.526]Two phase commit completed
org.apache.thrift.TApplicationException: Internal error processing doCommit
```

UNSUCCESSFUL TWO PHASE COMMIT WITH ABORT:

```
org.apache.thrift.TApplicationException: Internal error processing doCommit
        at org.apache.thrift.TServiceClient.receiveBase(TServiceClient.java:79)
        at ServerService$Client.recv_doCommit(ServerService.java:175)
        at ServerService$Client.doCommit(ServerService.java:162)
        at CoordinatorHandler.doCommitHelper(CoordinatorHandler.java:83)
        at CoordinatorHandler.startTwoPhase(CoordinatorHandler.java:129)
        at CoordinatorService$Processor$startTwoPhase.getResult(CoordinatorService.java:217)
        at CoordinatorService$Processor$startTwoPhase.getResult(CoordinatorService.java:197)
        at org.apache.thrift.ProcessFunction.process(ProcessFunction.java:38)
        at org.apache.thrift.TBaseProcessor.process(TBaseProcessor.java:39)
        at org.apache.thrift.server.TThreadPoolServer$WorkerProcess.run(TThreadPoolServer.java:310)
        at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1135)
        at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
        at java.base/java.lang.Thread.run(Thread.java:844)
[11:35:37.529]Two phase commit aborted
```