**Implementation of Two-Phase Commit using thrift RPC**

Authors:

Gangadhara Maddur Srikantaiah ([gmaddur1@binghamton.edu](mailto:gmaddur1@binghamton.edu))

Sachin Mahadevaprasad ([smahade2@binghamton.edu](mailto:smahade2@binghamton.edu))

**Code Structure:**

I would live to partition code part into three groups namely,

* Client end
* Coordinator end
* Participant end

**Client end:**

Below mentioned java classes are used in client end.

FileHandlingClient.java, FileStore.java, Participant.java, RFile.java, Status.java, StatusReport.java, SystemException.java, Transaction.java

FileHandlingClient.java spawns threads and issues random read and write on random files. This files does this by RPC to coordinator. Remaining classes are helper classes.

**Coordinator end:**

Below mentioned java classes are used in Controller end.

Coordinator.java, CoordinatorHandler.java, FileStore.java, Participant.java, RFile.java, Status.java, StatusReport.java, SystemException.java, Transaction.java

Coordinator.java servers as server and listens for requests from client. CoordinatorHandler.java contains all the methods to implement 2PC algorithm.

**Participant end:**

Below mentioned java classes are used in participant end.

FileHandlingClient.java, FileStore.java, Participant.java, ParticipantHandler.java, ParticipantServer.java, RFile.java, Status.java, StatusReport.java, SystemException.java, Transaction.java

ParticipantServer.java acts as server and listens for all the requests from coordinator. ParticipantHandler.java contains methods that implements write and read operations on file.

**RPC interface participants expose to the coordinator**

Following are the interfaces exposed by participants for coordinator

StatusReport writeFile(1: RFile rFile throws (1: SystemException systemException)

RFile readFile(1: string filename, 2: UserID owner) throws (1: SystemException systemException)

StatusReport canCommit(1: Transaction transaction) throws (1: SystemException systemException)

StatusReport doCommit(1: Transaction transaction) throws (1: SystemException systemException)

void doAbort(1: Transaction transaction) throws (1: SystemException systemException)

StatusReport isAborted(1:Transaction transaction) throws (1: SystemException systemException)

**Failure Detection**

**During read:**

If any of the participant is down, then read operation is achieved by doing read RPC to any of the remaining participant and result will be sent back to the client.

If all the participants are down, then Coordinator will try to contact all the participants then it will send “Participants are down” exception message to the client.

**During write:**

If any of the participant is down, then write operation will be aborted by the controller.

**Generic failure detection:**

Failure is detected by using the “catch block” with the exception “TTransportException”

After catching the exception, coordinator tries to communicate again after sometime.

If coordinator itself fails, exception is cought at the client end client tries to communicate to coordinator after some time.

**Test cases explored**

**1. Durability:**

Run participants, coordinator and client. After some time kill client and participants using the shell scripts provided. Then script will automatically bring back the participants first then client. Mean time in the listOfOperations.txt file remove write operation, so after crash recovery first operation issued will be read.

**Output:**

remote01:~/TestFolder/gmaddur1-project3-copy1> bash participantServer.sh participant1 9096

Participant started in port 9096

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

remote02:~/TestFolder/gmaddur1-project3-copy1> bash participantServer.sh participant2 9097

Participant started in port 9097

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

remote03:~/TestFolder/gmaddur1-project3-copy1> bash coordinatorServer.sh branches.txt 9098

Cooordinator started in port 9098

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

transaction 1 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 1

participant2 voted VOTE\_COMMIT for transaction 1

Final decision is made as COMMIT for transaction 1

transaction 2 arrived.

voting started

Final decision is made as ABORT for transaction 2

participant1 voted VOTE\_ABORT for transaction 2

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

remote04:~/TestFolder/gmaddur1-project3-copy1> bash client.sh 128.226.180.165098 listOfFiles.txt listOfOperations.txt 1

Client called threadId 1

operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}^Cremote04:~/TestFolder/gmaddur1-project3-copy1>

**2. Concurrency Control:**

Run participants, coordinator and clients (project for this test case is built without using sleep)for sometime. You can notice few failure cases to write in the client console.

**Logs at Participant 2:**

remote03:~/Distributed/gmaddur1-sachin-cocurrency test> ./participantServer.sh participant2 1234

Participant started in port 1234

Voted VOTE\_COMMIT for transaction 1

Voted VOTE\_COMMIT for transaction 3

Voted VOTE\_COMMIT for transaction 4

Voted VOTE\_COMMIT for transaction 2

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

doCommit called in participant for transaction 3

doCommit called in participant for transaction 2

Voted VOTE\_COMMIT for transaction 7

doCommit called in participant for transaction 4

doCommit called in participant for transaction 1

Voted VOTE\_COMMIT for transaction 9

Voted VOTE\_COMMIT for transaction 10

Voted VOTE\_COMMIT for transaction 8

doCommit called in participant for transaction 7

doCommit called in participant for transaction 9

doAbort called in participant

doAbort called in participant

doCommit called in participant for transaction 8

doCommit called in participant for transaction 10

Voted VOTE\_COMMIT for transaction 15

Voted VOTE\_COMMIT for transaction 14

Voted VOTE\_COMMIT for transaction 18

Voted VOTE\_COMMIT for transaction 17

Voted VOTE\_COMMIT for transaction 16

doCommit called in participant for transaction 15

doCommit called in participant for transaction 14

doCommit called in participant for transaction 18

doCommit called in participant for transaction 16

**Logs at participant 1:**

remote02:~/Distributed/gmaddur1-sachin-cocurrency test> ./participantServer.sh participant1 1235

Participant started in port 1235

Voted VOTE\_COMMIT for transaction 1

Voted VOTE\_COMMIT for transaction 1

Voted VOTE\_COMMIT for transaction 4

Voted VOTE\_COMMIT for transaction 3

Voted VOTE\_COMMIT for transaction 2

doCommit called in participant for transaction 1

doCommit called in participant for transaction 3

doCommit called in participant for transaction 4

doCommit called in participant for transaction 1

doCommit called in participant for transaction 2

Voted VOTE\_COMMIT for transaction 7

Voted VOTE\_COMMIT for transaction 8

doCommit called in participant for transaction 7

Voted VOTE\_COMMIT for transaction 9

doCommit called in participant for transaction 9

Voted VOTE\_COMMIT for transaction 10

Voted VOTE\_COMMIT for transaction 8

Voted VOTE\_COMMIT for transaction 9

doAbort called in participant

doAbort called in participant

doAbort called in participant

doCommit called in participant for transaction 10

doCommit called in participant for transaction 8

Voted VOTE\_COMMIT for transaction 15

Voted VOTE\_COMMIT for transaction 14

Voted VOTE\_COMMIT for transaction 17

**Logs at Coordinator:**

remote01:~/Distributed/gmaddur1-sachin-cocurrency test> ./coordinatorServer.sh branches.txt 1234

Cooordinator started in port 1234

transaction 1 arrived.

transaction 4 arrived.

transaction 3 arrived.

transaction 2 arrived.

transaction 1 arrived.

voting started

voting started

voting started

voting started

voting started

participant1 voted VOTE\_COMMIT for transaction 1

participant1 voted VOTE\_COMMIT for transaction 2

participant1 voted VOTE\_COMMIT for transaction 1

participant1 voted VOTE\_COMMIT for transaction 4

participant1 voted VOTE\_COMMIT for transaction 3

participant2 voted VOTE\_COMMIT for transaction 1

participant2 voted VOTE\_COMMIT for transaction 3

participant2 voted VOTE\_COMMIT for transaction 4

participant2 voted VOTE\_COMMIT for transaction 2

participant2 voted VOTE\_COMMIT for transaction 1

Final decision is made as COMMIT for transaction 1

Final decision is made as COMMIT for transaction 4

Final decision is made as COMMIT for transaction 3

Final decision is made as COMMIT for transaction 1

Final decision is made as COMMIT for transaction 2

transaction 5 arrived.

voting started

Final decision is made as ABORT for transaction 5

participant1 voted VOTE\_ABORT for transaction 5

transaction 6 arrived.

voting started

Final decision is made as ABORT for transaction 6

participant1 voted VOTE\_ABORT for transaction 6

transaction 7 arrived.

voting started

transaction 8 arrived.

participant1 voted VOTE\_COMMIT for transaction 7

voting started

participant2 voted VOTE\_COMMIT for transaction 7

participant1 voted VOTE\_COMMIT for transaction 8

transaction 9 arrived.

Final decision is made as COMMIT for transaction 7

voting started

transaction 8 arrived.

voting started

Final decision is made as ABORT for transaction 8

participant1 voted VOTE\_COMMIT for transaction 9

participant2 voted VOTE\_ABORT for transaction 8

transaction 9 arrived.

voting started

participant2 voted VOTE\_COMMIT for transaction 9

transaction 10 arrived.

voting started

Final decision is made as COMMIT for transaction 9

participant1 voted VOTE\_COMMIT for transaction 10

**Logs at Client:**

remote00:~/Distributed/gmaddur1-sachin-cocurrency test> ./client.sh 128.226.180.163 1234 listOfFiles.txt listOfOperations.txt 5

Client called threadId 1

Client called threadId 5

Client called threadId 4

Client called threadId 3

Client called threadId 2

operation is write threadId 2

operation is write threadId 1

operation is write threadId 5

operation is write threadId 3

operation is write threadId 4

fileName is test.txt threadId is 5

fileName is test.txt threadId is 3

fileName is test.txt threadId is 4

fileName is test.txt threadId is 1

fileName is test.txt threadId is 2

StatusReport(status:SUCCESSFUL)

operation is write threadId 4

fileName is test.txt threadId is 4

StatusReport(status:FAILED)

operation is write threadId 4

fileName is test.txt threadId is 4

StatusReport(status:FAILED)

operation is write threadId 4

fileName is test.txt threadId is 4

StatusReport(status:SUCCESSFUL)

operation is write threadId 3

fileName is test.txt threadId is 3

StatusReport(status:SUCCESSFUL)

operation is write threadId 2

fileName is test.txt threadId is 2

StatusReport(status:SUCCESSFUL)

operation is write threadId 1

fileName is test.txt threadId is 1

StatusReport(status:FAILED)

operation is write threadId 3

fileName is test.txt threadId is 3

StatusReport(status:SUCCESSFUL)

operation is write threadId 5

fileName is test.txt threadId is 5

StatusReport(status:SUCCESSFUL)

operation is write threadId 4

fileName is test.txt threadId is 4

StatusReport(status:FAILED)

operation is write threadId 3

fileName is test.txt threadId is 3

StatusReport(status:FAILED)

operation is write threadId 4

fileName is test.txt threadId is 4

StatusReport(status:SUCCESSFUL)

operation is write threadId 2

fileName is test.txt threadId is 2

StatusReport(status:FAILED)

operation is write threadId 3

fileName is test.txt threadId is 3

StatusReport(status:FAILED)

**3. Coordinator Failure before voting started:**

Kill coordinator using the script provided as soon as it receives the transaction. Script automatically brings back the coordinator after some time, then notice how coordinator behaves.

remote01:~/TestFolder/gmaddur1-project3-copy1> bash testCaseP1.sh participant1 9096

Participant started in port 9096

returned status as ABORT for transaction 1

Voted VOTE\_COMMIT for transaction 1

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

returned status as ABORT for transaction 2

Voted VOTE\_COMMIT for transaction 3

remote02:~/TestFolder/gmaddur1-project3-copy1> bash testCaseP1.sh participant2 9097

Participant started in port 9097

returned status as ABORT for transaction 1

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

returned status as ABORT for transaction 2

remote03:~/TestFolder/gmaddur1-project3-copy1> bash coordinatorServer.sh branes.txt 9098

After timeout, operation is aborted for transaction 2

Cooordinator started in port 9098

readFIle for file name TestFile.txt

transaction 3 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 3

remote04:~/TestFolder/gmaddur1-project3-copy1> bash client.sh 128.226.180.165 9098 listOfFiles.txt listOfOperations.txt 1

Client called threadId 1

operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is write threadId 1

fileName is TestFile.txt threadId is 1

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

operation is read threadId 1

operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is write threadId 1

fileName is TestFile.txt threadId is 1

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is write threadId 1

fileName is TestFile.txt threadId is 1

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

**4. Coordinator Failure after voting started**

**a. before final decision is made**

Kill the coordinator before final decision is made (each stage status will be printed in the coordinator console). Script will bring back the coordinator after some time, then notice how coordinator behaves.

remote01:~/TestFolder/gmaddur1-project3-copy1> ./participantServer.sh participant1 9096

Participant started in port 9096

Voted VOTE\_COMMIT for transaction 1

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

Voted VOTE\_COMMIT for transaction 2

remote02:~/TestFolder/gmaddur1-project3-copy1> ./participantServer.sh participa

nt2 9097

Participant started in port 9097

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

remote03:~/TestFolder/gmaddur1-project3-copy1> bash coordinatorServer.sh branches.txt 9098

Cooordinator started in port 9098

transaction 1 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 1

remote03:~/TestFolder/gmaddur1-project3> bash testCase34.sh branches.txt 9098

conduct voting again

transaction 1 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 1

participant2 voted VOTE\_COMMIT for transaction 1

Final decision is made as COMMIT for transaction 1

Cooordinator started in port 9098

transaction 2 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 2

remote04:~/TestFolder/gmaddur1-project3-copy1> bash client.sh 128.226.180.165 9098 listOfFiles.txt listOfOperations.txt 1

Client called threadId 1

operation is write threadId 1

fileName is TestFile.txt threadId is 1

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

operation is write threadId 1

fileName is TestFile.txt threadId is 1

Coordinator went down, going to wait state. Will start communicating once coordinator is up

**b. after final decision is made**

Kill the coordinator after final decision is made using the script provided. Script will bring back the coordinator after some time, then notice how coordinator behaves.

remote01:~/TestFolder/gmaddur1-project3-copy1> ./participantServer.sh participant1 9096

Participant started in port 9096

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

Voted VOTE\_COMMIT for transaction 2

doCommit called in participant for transaction 2

Voted VOTE\_COMMIT for transaction 3

^Cremote02:~/TestFolder/gmaddur1-project3-copy1./participantServer.sh participant2 9097

Participant started in port 9097

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

Voted VOTE\_COMMIT for transaction 2

doCommit called in participant for transaction 2

remote03:~/TestFolder/gmaddur1-project3-copy1> bash coordinatorServer.sh branches.txt 9098

Cooordinator started in port 9098

transaction 1 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 1

participant2 voted VOTE\_COMMIT for transaction 1

Final decision is made as COMMIT for transaction 1

transaction 2 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 2

participant2 voted VOTE\_COMMIT for transaction 2

Final decision is made as COMMIT for transaction 2

^Cremote03:~/TestFolder/gmaddur1-project3> bash testCase34.sh branches.txt 9098

Ask participants to commit for transaction 2

Cooordinator started in port 9098

transaction 3 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 3

remote04:~/TestFolder/gmaddur1-project3-copy1> bash client.sh 128.226.180.165 9098 listOfFiles.txt listOfOperations.txt 1

Client called threadId 1

operation is write threadId 1

fileName is TestFile.txt threadId is 1

StatusReport(status:SUCCESSFUL)

operation is write threadId 1

fileName is TestFile.txt threadId is 1

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

Coordinator went down, going to wait state. Will start communicating once coordinator is up

operation is write threadId 1

fileName is TestFile.txt threadId is 1

Coordinator went down, going to wait state. Will start communicating once coordinator is up

**5. Participant Failure:**

Kill one of the participant after it voted “COMMIT” to the coordinator using the script provided. Script will bring back the coordinator after some time, then notice how coordinator behaves.

remote01:~/TestFolder/gmaddur1-project3-copy1> ./participantServer.sh participant1 9096

Participant started in port 9096

Voted VOTE\_COMMIT for transaction 1

remote01:~/TestFolder/gmaddur1-project3-copy1> bash testCase5.sh participant1 9096

doCommit called in participant for transaction 1

Participant started in port 9096

Voted VOTE\_COMMIT for transaction 3

remote02:~/TestFolder/gmaddur1-project3-copy1> ./participantServer.sh participant2 9097

Participant started in port 9097

Voted VOTE\_COMMIT for transaction 1

doCommit called in participant for transaction 1

remote03:~/TestFolder/gmaddur1-project3-copy1> bash coordinatorServer.sh branches.txt 9098

Cooordinator started in port 9098

readFIle for file name TestFile.txt

transaction 1 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 1

participant2 voted VOTE\_COMMIT for transaction 1

Final decision is made as COMMIT for transaction 1

transaction 2 arrived.

voting started

Final decision is made as ABORT for transaction 2

participant1 voted VOTE\_ABORT for transaction 2

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

readFIle for file name TestFile.txt

transaction 3 arrived.

voting started

participant1 voted VOTE\_COMMIT for transaction 3

remote04:~/TestFolder/gmaddur1-project3-copy1> bash client.sh 128.226.180.165 9098 listOfFiles.txt listOfOperations.txt 1

Client called threadId 1

operation is read threadId 1

SystemException(message:File doesn't exist)

operation is write threadId 1

fileName is TestFile.txt threadId is 1

StatusReport(status:SUCCESSFUL)

operation is write threadId 1

fileName is TestFile.txt threadId is 1

StatusReport(status:FAILED)

operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is read threadId 1

{"2":{"str":"asdkajslfakaksjfalsfka\n"},"4":{"i32":0}}operation is write threadId 1

fileName is TestFile.txt threadId is 1