Q1: Implement the voting phase of 2PC, **using python**.

1. Define vote request and respone:

A computer screen with white text

Description automatically generated

1. Define vote function

A computer screen shot of text

Description automatically generated

1. Build up communication between coordinator and participants

A black background with text

Description automatically generated

1. Send vote request and receive vote commit form participants

A screen shot of a computer code

Description automatically generated

1. Result:

CMD: python -m grpc\_tools.protoc --proto\_path=proto --python\_out=. --grpc\_python\_out=. proto/twopc.proto

docker-compose up --build

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Q2: Implement the decision phase of 2PC, **using node.**

1. The coordinator collects all votes from the participants

A computer screen with text on it

Description automatically generated

1. If Coordinator receive all RPC decide, call Python GlobalDecision

A screen shot of a computer code

Description automatically generated

1. Coordinator receive all RPC decide from participants

A computer code on a black background

Description automatically generated

1. Result:Coordinator request voting, and receive commit from all participants

A screen shot of a computer

Description automatically generated

Coordinator send global decision to all participants, and all participant locally commits the transaction.

A screenshot of a computer program

Description automatically generated

Q3. Implement the leader election of a simplified version of Raft, using python.

1. Define vote and appendEntries message

A screen shot of a computer program

Description automatically generated

1. Define raft node attributes

A screen shot of a computer program

Description automatically generated

1. If a follower fail to receive heartbeat or election ack, it adds term by 1 and become a candidate voted for itself and send **RequestVote to other nodes.**

A computer screen shot of a program

Description automatically generated

1. If candidate receives a majority of votes, it becomes the leader and begins sending periodic **AppendEntries**

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

1. Simulation:

Init 5 nodes as follower

A black background with white text

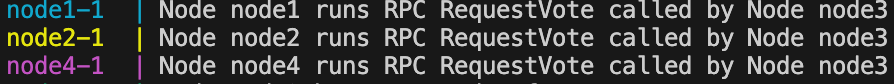
Description automatically generated

If node3 fail to receive heartbeat, add its term by 1 and become candidate. Send vote request to other node.

A black background with white text

Description automatically generated

Other node response request vote call by node3 after compare its term.



After receive majority of ack, node3 becomes leader.( we do not need ack by node5)



Node3 send appendEntries to other nodes, and other nodes receive it.

A screen shot of a computer program

Description automatically generated

Q4. Implement the log replication of a simplified version of Raft, **using node.**

1. If client call a operation, but follower receive it. The operation will be forwarded to leader.

A screen shot of a computer program

Description automatically generated

1. If a leader receive operators, create new entry of appendEntries and multicast appendEntries.

A screen shot of a computer program

Description automatically generated

1. Once leader received majority of acks, response the client operation have been executed.

A screen shot of a computer code

Description automatically generated

1. Client send operation to server.

A screen shot of a computer program

Description automatically generated

1. Simulation:

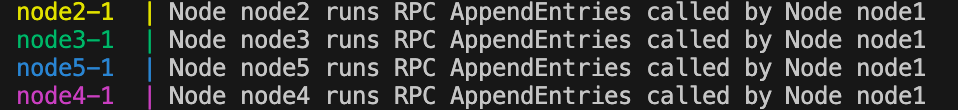
CMD: node client.js 0.0.0.0:5001 "doSomething"

Node 1 is leader and receive operation from client. Multicast to other followers

A black background with white text

Description automatically generated

Followers received request and respone ack to leader



Leader received majority of ack, response to client.



Client got response



CMD: node client.js 0.0.0.0:5002 "doSomething2"

Client send to a follower( node2).

A screen shot of a computer screen

Description automatically generated

