Practical Byzantine Fault Tolerance Consensus and A Simple Distributed Ledger Application

Hao Xu Muyun Chen Xin Li

Abstract

Along with cryptocurrencies become a great success known to the world, how to deploy a large scale, robust Byzantine Fault Tolerant system turns into an interesting challenge in the technical community. We as a group of distributed system are practitioners in implementing the core consensus used in the distributed ledger - Practical Byzantine Fault Tolerance (known as PBFT, in Liskov), and designing а simple distributed ledger application of simulating the peer-to-peer transactions, in order to have a principle understanding the PBFT protocol, and its powerful strength to survive various software errors and malicious attacks.

1. Introduction

The objective of Byzantine fault tolerance is to be able to defend against Byzantine failures, in which components of a system fail with symptoms that prevent some components of the system from reaching agreement among themselves, where such agreement is needed for the correct operation of the system. Correctly functioning components of a Byzantine fault tolerant system will be able to provide the system's service, assuming there are not too many faulty components.

The application basically simulates the account transactions (deposit, withdraw, move, etc) of the bank system, which is distributed with data replicated. During the process of the simulation, there might encounter PBF causing some nodes problem, but with the distributed ledger technology, the non-fault nodes can reach consensus to make the transaction succeed and correct.

One example of BFT in use is bitcoin, a peer-to-peer digital currency system. The bitcoin network works in parallel to generate a chain of Hashcash style proof-of-work. The proof-of-work chain is the key to overcome Byzantine failures and to reach a coherent global view of the system state.

In our system, the application basically simulates the account transactions (deposit, withdraw, move, etc) of the bank system, which is distributed with data replicated. To simulate the Byzantine fault during the process, some fundamental standups below are giving us a

- a. Any node can crash and recover at any time.
- b. Use UDP to communicate between replicas, so the messages sent to each node might be lost, duplicated or disordered;
- c. Client send request to replica to deposit and retrieve money, and double check the consistency between replicas.
- d. Anytime with the 3f + 1 nodes, the system is able to survive f fault nodes.

2. Background

Figure 1 shows a normal case operation described in the paper. With four replicas (one primary) the system is able to tolerate one faulty node at a time.

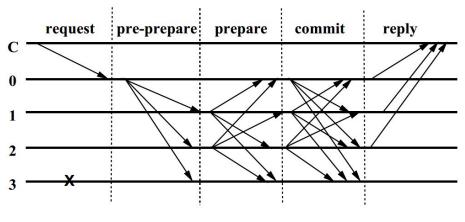


Figure 1.

Another important protocol is checkpoint. we will describe more detail of checkpoint endpoints in Section 3, here we only pinpoint that the checkpoint and digest are important to maintain data consistency in database.

3. Distributed Ledger

Basic: Build account database (including account information, balance, etc.) and replicated it into multiple nodes. Launch replica process in each replica node and each Process or some Processes share one database; Distributed: distributed ledger is а а peer-to-peer system, each node can do transaction (communicate with other node) asynchronously;

- 1.1) support command line input to start the transaction in the client, eg: get current balance for A deposit \$100 to A move \$100 from B to A withdraw \$200 from B
- 1.2) Information updated automatically among different nodes. The consensus among the working nodes is supported by PBFT, and we also rely on database's log system to commit, redo and undo. After the consensus and commit, the information

should be updated into each replica;

1.3) Simulate PBFT to cause some nodes failed during the transaction. Since the messages are transferred with UDP, which is not reliable, the system can detect and process UDP related issues to keep accordance. We also manually shut down replica to simulate fault process.

4. Implementation

4.1 **PBFT Service**

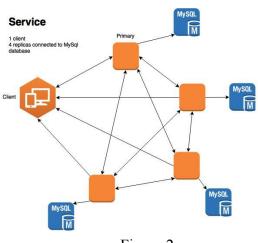
We reuse MIT BFT open source library [2], and sfslite, a cryptography software tool [3] to help design our PBFT service.

The *libbyz* library implement the PBFT algorithm described in the paper. It provides one client interface *invoke*, which sends the request operation to replicas, and one main server side interface *execute*, to receive and execute the requested operations.

The *mysql* library provides the endpoints connected to Mysql database, it has two interface for both single-thread and multi-thread execution.

The *PBFTservice* library provide the endpoint to connect between PBFT distributed system and databases. And it simulated some simple transaction samples.

The service system is shown in Figure 2.





For each replica:

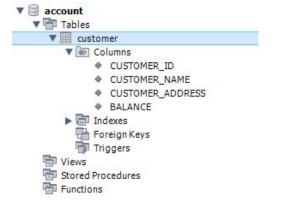
- wait infinitely for new transaction log.
- For the primary node, receive the input transaction from client and multicast to other replicas.
- Each replica will communicate with DB to execute and commit with log read and write one they receive the request.
- Communicate with other nodes with log read and write.

4.2) AWS environment setup

We initiated five AWS instances for simulating the distributed system. We will not give too much detail about the environment. But we are using network packet IO data to evaluate the throughput of the system. This can be optimized to use other AWS services to have more accurate evaluation.

4.3) Database

Install MySql Workbench to create bank account database, the table structure is like:



We implement connection pool in the replica to communicate between each replica and database, so the transaction is committed within the connection pool managed by each replica. Also MySql provide interfaces to maintain and manage the log and data and called by the replicas and clients.

5. Simulation

The main procedure involves three steps:

- 1) The client main procedure start transaction;
- 2) The replica main procedures start the pbft phases;
- 3) The replica executes the request and persists result into MySQL db in the commit phase (after committed-local is true).

We will present more specific details in the Appendix Cases. You can have a better understanding of how we simulate the protocols.

6. Discussion

correctness

- 1. The system can run successfully with at most 1 fault node, the remaining replicas keep consistent.
- 2. The combination of *view_change* and *check_point* enable the replicas to have consistency data after a node recover from the network partition (but it cannot recover from crash, because we don't persist logs into disk, while we do persist current state into db). Basically the returned replica can obtain missing messages from other replicas.

performance

1. We run simulation in both read-only mode and read-write mode. In the read-write mode, there are 50% read operations and 50% write executed in random sequence. The read-only mode is about 1.7 times faster than the read-write mode, which is as expected because of the read-only optimization

in PBFT algorithms.

Table 1 Simulation result. No.iterations: the number of iteration, read: the number of read operation, write: the number of write operation per simulation. time: the avg response time of a request. view_change: how many view_change happened in the simulation (totally). network partition times: manually turn down the aws network and then turn it on.

No. iterations	read	write	time(ms)/request	view_change	network partition times		
1000	1000	0	1.34	0	0		
1000	500	500	2.31	3	0		
5000	2500	2500	2.56	5	1		
10000	5000	5000	2.35	4	2		

Table 1.

2. While restart the fault server the operation slow down dramatically. We assume it is because after the replica comes back to network, there are more communications between replicas and clients to resume lost messages, which will consume more times.

7. Improvement

An interesting implementation of PBFT is the peer-to-peer transaction system. For instance, some of the blockchain techniques are using the *Proof-of-work* based on the PBFT algorithm, which would be an future development of our system.

8. Conclusion and Acknowledgement

The main goal of the project is to get all of us more familiar with the PBFT protocols. Even though the algorithm is published about 20 years ago, we can still find its significant influence over the technology world. We started from implementing the algorithm, integrating open source tools, and then proceeded to simulating the transaction and distributing it to databases. Finally now we have better understanding of the PBFT not only its protocol, but also its implement. Even the simulation is simple and may contain flaws while handling complicated cases, but it is enough to maintain as a study case, which can be scale up to larger systems, and can be further designed to peer-to-peer transaction system.

9. Reference

- [1] Castro, Miguel, and Barbara Liskov. "Practical Byzantine fault tolerance." OSDI. Vol. 99. 1999.
- [2] Programming Methodology Group, MIT, http://www.pmg.csail.mit.edu/bft/
- [3] Sfslite, https://github.com/OkCupid/sfslite/wiki
- [4] Source code: https://github.com/cmuhao/CS244b_final_project

Appendix

- 1) Experiment 1: write one and read, which means replicas show the specified one customer's balance related information after it wirted.
 - Request from client: **deposit A 100, get A** Result from replicas:

Command Pron	1pt - ssh -i *,pem ec2-us	501@151571521240		Command Prompt - ssh -i *.pem ec2-user@54.193.27.25				
eplica 1:	·	±		Replica 2:				
CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	BALANCE	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	BALANCE	
	Michael Chen	NULL	200	3	Michael Chen	NULL	200	
Command Pron	npt - ssh -i *.pem ec2-us	ser@52.53.168.124	++	Command Pron	npt - ssh -i *.pem ec2-u	ser@54.183.251.79	+	
_	npt - ssh -i *.pem ec2-us	ser@52.53.168.124	++		npt - ssh -i *.pem ec2-u	ser@54.183.251.79	+	
_	npt - ssh -i *.pem ec2-us	ser@52.53.168.124	++	Command Pron Replica 4:	npt - ssh -i *.pem ec2-u	ser@54.183.251.79	+	
Command Prom plica 3: CUSTOMER_ID	npt - ssh -i*.pem ec2-us CUSTOMER_NAME	+	BALANCE	Replica 4: +	+	ser@54.183.251.79	BALANCE	

2) Experiment 2: 1000, 5000, and 10000 iterations, random deposit and withdrawal. Result from replicas:

X root@ip-172-31-8-111/home/ec2-user/CS244b_final_project/rsc/bft/bft-simple (ssh)	X motilie.172.21	1-1-74/home/ec2-user/C5244b final or	niarthra-Drhhhhaimeis (mh)
x rootgrp-1/2-31-8-111/home/ecz-user/Cs2446_tna_project/rsc/ott/ott-simple (son)		count set balance-balance+10 W	
ν Γrootθip-172-31-8-111 bft-simple]∉ ,/simple-client	sqL: UPLATE ACC execute success		acke name= A
WANING: disabled multicost	incoming reques		
		ance FROM account INHERE name="	
noot#ip=172-31-8-111 bft-simple]∉ ./simple-client	execute success		
ARNING: disabled multicast	5070		
lapsed time 2.470501 for 1000 iterations of operation 0	incoming reques	t: DEPOSIT 10 TO A	
root#ip-172-31-8-111 bft-simple]# ./simple-client	sql: UPDATE acc	count set balance-balance+10 W	
ARNING: disabled multicast	execute success		
lapsed time 124.810318 for 1000 iterations of operation 0	incoming reques		
root8ip-172-31-8-111 bft-simple]# ./simple-client			
ARNING: disabled multicast	execute success		
lapsed time 2.314138 for 1000 iterations of operation 0	5880		
root#ip=172-31-8-111 bft-simple]# ./simple-client -t 5000			
ARNING: disabled multicast		count set balance-balance+10 W	(ERE name='A'
nter command:command:	execute success		
eply: Elopsed time 2.142641 for 1000 iterations of operation 0	incoming reques		
root8ip-172-31-8-111 bft-simple]# ./simple-client -i 5000			
WRMING: disabled multicast Elapsed time 18.341013 for 5000 iterations of operation 0	execute success 5898		
rootBip-172-31-8-111 bft-simple]# ./simple-client -i 10000		IL: DEPOSIT 10 TO A	
/Soterp-1/2-31-8-111 Dft-Simplej# ./Simple-client -1 10000 WANING: disabled multicost		cc: DEPOSIT 10 TO A	100 111
lapsed time 31.824320 for 10000 iterations of operation 0	execute success		IENE RUICH A
root#ip-172-31-8-111 bft-simple]# ^C	incoming reques		
[root@ip-172-31-8-111 bft-simple]# ./simple-client 10000		ance FROM account NHERE name="	
WRMING: disabled multicost	execute success		
clapsed time 2.315865 for 1000 iterations of operation 0	5100		
root8ip-172-31-8-111 bft-simple]#	1		
	5		
	× root@ip-172-31-5-155:/home/ec2-user/CS244b_final_project/rsc/bft/bft-sim	nple (ssh) 📖	X root@ip-172-31-15-153;/home/ec2-user/CS244b_final_project/rsc/bft/bft-simple (ssh)
	sql: UPDATE account set balance=balance=10 WHERE name='A'		
xecute success	execute success		execute success
xxecute success ncoming request: GET A	execute success incoming request: GET A		execute success incoming request: GET A
xecute success ncoming request: GET A ql: SELET bulance FRDM account WHERE name='A'	execute success incoming request: GET A sql: SELECT balance FROM account WHERE name='A'		execute success incoming request: GET A sql: SELECT balance FROM account WHERE name="A"
xecute success ncoming request: GET A ql: SELEC bolance FROM account WHERE nome="A" Neorite success	execute success incoming request: GET A sql: SELECT balance FROM account WHERE name='A' execute success		execute success incoming request: GET A sql: SELECT balance FROM account IMERE name='A' execute success
viscute success ncoming request: GET A di SELECT balance FROM account WHERE nome="A" neorie success Arg	execute success incoming request: GET A sql: SeLLEC balance FROM account MMERE name="A" execute success 2650		execute success inconing request: GET A sql: SELEC balance FROM account MHERE name-'A' execute success serve
viente success nording request: GT A ql: SELECT balance FRDM account WHEEE nume-'A' works success APD nording request: DEPOSIT 10 TO A	execute success incosting request: GET A sql: SELECT balance FRDM account MHERE name='A' execute success 2620 incosting request: DEPOSIT 10 TO A		exercite success inconting request: GET A sql: SELECT boliones FROM account WHERE name-'A' exercite success S070 inconting request: DEPOSITI 10 TO A
vincete success oncette request: GET A 4): SULCE Tolance HRM account MERE name-'A' societe success concenting request: DEPOSIT 38 TO A (): UPGNT account est balance-balances-38 MERE name-'A'	execute success inconing request: GET A sql: SELEC balance FROM account MHERE name-'A' execute success 2020 inconing request: DEPOSIT 10 TO A sql: UFUNE counts set balance-balance-10 MHERE name-'A'		ennots success Incoming request: GFT A sql: SUIC' balance HFDM account WREE nome-'A' encoder success encoder success enc
vinote success sonote property: UPT A state of the sonote state of the sonote state state of the sonote state of the sonote state sonote sonote sonote state state sonote soccess society access	exacute success incoming request: GET A sql: SILCT bollows FROM account WHERE name='A' execute success 2020 incoming request: DEPOSIT 10 TO A sql: UPATE account set bollonce+b0ince+10 WHERE name='A'		neucots accoss [coording request: UFT A eq: 24.027 billions FROM account WHEEE nume-14* 58/09 tocoming properties 100/005/138 TO A injuncties success
whorts microses sounds provide (FTA) works microses m	execute success incoming regards: GFT A sql: SILUT bolance HRM ecount MERE name-A' secute success 2020 INFORM programs: GRIONT: 10 TO A INFORM PROVINCE STATUS INFORMATION AND AND AND AND AND AND INFORMATION AND AND AND AND AND AND AND INFORMATION AND AND AND AND AND AND AND INFORMATION AND AND AND AND AND AND AND AND INFORMATION AND AND AND AND AND AND AND AND AND INFORMATION AND AND AND AND AND AND AND AND AND AN		menter sectors incoming request GET A sql: SULCT balance RRM eccurit MEEE name-'A' execute success SMP descriptions: concentration of the A sector success incoming requests: GET A SMP Sectors SMP Sectors
winche aucossis andre maante (EFA A config maante (EFA A De ancessis) config maante (EFA A (EFA A) (EFA A	encode success (conting request: GFT A sql: SULCT building: FMB count MERE name-A ¹ option sql: SULCT boint and the sql (conting request: SUPOSIT 30 TA sql: SULCT boint set building-salarses) MERE name-'A ¹ second access sql: SULCT boint set Red account MERE name-'A ¹		menche success Loording magnets: GPT A hei: SILLT balance RMM account MREE name-'A' hei: SILLT balance RMM account MREE name-'A' menche Rangests: GPT A spil: SILLT balance RM account RMEE name-'A' menche Rangests spil: SILLT balance RM account RMEE name-'A'
whords excessions on only request ICS and the excession there are a second to be excession of the excession of the excession of the excession of the control of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the excession of the	execute success Locating request: GT A 441 Idle To before FRDE council WHEE nome-A' 2020 Locating program (LIC) (LIC) (LIC) (LIC) Locating program (LIC) (LIC) (LIC) Locating program (LIC) (LIC) (LIC) Locating program (LIC) (LIC) Locating program (LIC) (LIC) Locating program (LIC) (LIC) LIC) (LIC) (LIC) (LIC) (LIC) LIC) (LIC) (LIC) (LIC) (LIC) LIC) (LIC) (LIC) (LIC) (LIC) (LIC) LIC) (LIC) (LIC) (LIC) (LIC) (LIC) LIC) (LIC) (LIC		encode secons Looning request: GFT A 401: SLEVET hallows RMA excount WEEE nome-'A' MANN PARAMENT AND A SCHLEME AND A SCHLEME Howard Parament And I balance-balance-ball WEEE nome-'A' Becode secons Looning request: GFT A encode secons Looning request: GFT A
whords access is and access is a constant sets name-'A' whore access is a constant access is a constant sets name-'A' and names names (1: UPDT bolino m RM access) BHEE name-'A' and a names (1: SULT) bolino m RM access) BHEE name-'A' and a names	encode success (conting request: GFT A sql: SULCT building: FMB count MERE name-A ¹ option sql: SULCT boint and the sql (conting request: SUPOSIT 30 TA sql: SULCT boint set building-salarses) MERE name-'A ¹ second access sql: SULCT boint set Red account MERE name-'A ¹		menche success locardy anguest. GPT A set: SELTC balance 700% accurst WEEE nome-'A' segne success inconfig request: E07051T IP TO A sectors success sectors success sectors success set: SELCT balance Placecourts WEEE nome-'A' set: SELCT balance Placecourts WEEE nome-'A'
winche aucossis 41: SILLE follower HIPE Reme-'A' 41: SILLE follower HIPE Reme-'A' Provide State State State State State State State State State State State State State State (1): SILLE State State Follower'A' State S	encute success (control progents GET A spic SELET balance RRM scores WEEE name-'A' market success (control progents DROSIT 38 TO A spic URVET scores to balance-balance-LB WEEE name-'A' encodes success (control progents GET A spic SELECT balance RRM sccoret WEEE name-'A' spic SELECT balance RRM sccoret WEEE name-'A' SEN		encode access transfer quages (OF 1. transfer quages (OF 1. transfer quages (OF 1. test) (OF 1.
whords accesses whords accesses accesses and accesses and accesses	encode accoss locaring request: GPT A sci: Statt: Tollano: RXM account WEEE name-'A' Sci Sci Locaring request: GPT A locaring request: GPT A locaring request: GPT A sci : SL(TC) Biology Content: B WEEE name-'A' content request: GPT A sci : SL(TC) Biology Content: B WEEE name-'A' sci sci : SL(TC) Biology Content: B WEEE name-'A' sci sci sci sci sci sci sci sci		encode success forming request: GFT A Not: SELECT balance NPU account WHEE nome-A' SHO incoming request: EPOSIT Ja TO A Incoming request: GFT A set Jalance Selection and Selection and forming request: GFT A secode success provide second request: EPOSIT Ja TO A
whords access is and a second for measure of the second second second second second measure access is access as a second second second second second second second (1) (POTI Second second second second second second second second second second second second second second second (1) (Second second (1) (Second second second second second second second second second second second second second second second second second second s	encies success locating regards: UT A success program : UT A success program : UT A success program : UT A Component : UT A (UT A) (UT		encode success Locating request: GFLA depicts for the second WEEE nome-'A' depicts for the second WEEE nome-'A' How Provide success Network Provide How Provide success How Provide success Ho
whoth success whoth success 41. SUEC To Marco ANN REFE NUMe-'A' BALLED STATE AND ANN ANN ANN ANN ANN ANN ANN ANN ANN	encode successi transfer appendix II and Records HEEE non-'A' encode successi and appendix II and II and II and II and II and II and II and II and II and II and II		encode access transfer quagest CBT A transfer quagest CBT A tencing to constrain the second the second the second tencing the second tencing the second tencing tencing the second tencing the second tencing
whords access whords access whord a protect IC and whord a protect IC and whord a protect IC and whord a protect IC and IC and IC and and IC and IC and IC and IC and IC and access access whords access whords in protect IC and IC and IC and IC and whord IC and IC and IC and IC and IC and IC and whords access whords access whords access whords access whords access access access (I) IF ANT access and I ablance-bit access if BEEL name-'A' access access (I) I ANT access access field access if BEEL name-'A' access access (I) I ANT access access field access if BEEL name-'A' access access (I) I ANT access access field access if BEEL name-'A' (I) I ANT access field access field access field access if BEEL name-'A'	mencies success incontary anguest: GPT A leg. SINCL' formane FROM exclosel MEEE none-A' leg. Since formation and the second memory and formation grangest: GPT AIT DA leg. SINCL' biology and Records MEEE none-A' media since formation and and and and and and and since formation and and and and and and and and since formation and and and and and and and and and since formation and and and and and and and and and since formation and and and and and and and and and since formation and and and and and and and and and an		renords access foroning request: GPT A Hot: Starts Nations RAD account WEER nome-'A' Starts Inconting request: GPTOST 18 TO A Inconting request: GPTOST 18 TO A Inconting request: GPT A Hot: Starts National Access Inconting request: GPT A Hot: Starts National Access Inconting request: GPT A Hot: Starts National Access Inconting request: GPT A Hot: Starts RAD Access Inconting request: GPT A Hot: Starts RAD Access Inconting request: GPT A
whorks success whorks a most of properties (FM account: WEEE name-'A' success burgers and the success and the success and the success and the	mencies success locating request: 2015 A 402, 1942 CH Innien FRM KCourt HEEE new-A' 402, 1942 CH Innien FRM KCourt HEEE new-A' 2008 Inconting request: 2015 A 104, 1942 Ch Innie FRM KCourt HEEE new-A' 402, 1942 Ch Innie FRM KCourt HEEE new-A' 402, 1942 Ch Innie FRM KCourt HEEE new-A' 403, 1942 Ch Innie FRM KCourt HEEE new-A' 404, 1947 Charles FRM KCourt HEEE new-A' 404, 1947 Charles FRM KCourt HEEE new-A' 404, 1947 Charles FRM KCourt HEEE new-A' 405, 1947 Ch Innien FRM KCourt HEEE new-A' 405, 1947 Ch Innien FRM KCourt HEEE new-A'		encode success (conting request: GPT A Ref: BLECT balance RDM account WEEE nome-'A' Ref: BLECT balance RDM account WEEE nome-'A' RDM account property in the balance-balance-100 WEEE nome-'A' RDM account property in the balance-balance-100 WEEE nome-'A' RDM account property in the balance-balance-100 WEEE nome-'A' Ref: BLECT balance RDM account WEEE nome-'A' Ref: RDM account set balance-balance-100 WEEE nome-'A'
whords access whords access access of a project (2011) access of the project (2011) access o	encies success (contrar superior UT A contrar superior UT A success success and success the BER sum-A* encies success 2009 opportunity (FORT 14 Th A (contrar success success) (contrar success success and and and and and and and and and (contrar success) (contrar succes		encode success Locardig request: GFLA locardig request: GFLA locardig request: GFLA locardig request: GFLA second success second success locardig request: GFLA locardig request: GFLA locard
whords access whords access access whord a model of the second states name 'A' whord a model of the second states name 'A' and a model of the second states name 'A' and a model name access access access access access access access access access access access access access access access ac	Periods success (control property IT) and the second second REE (new 'A' encode success (control property IT) (control program (control program (control program (control program (control program (control program) (control program (control program) (control program)		encode access transfer quegate: CET 1 transfer quegate: CET 1 transfer quegate: CET 1 transfer quegate: CET 1 transfer quegate: CET 10 To A test: EMPTI Eccount set bilance-bilance: 30 #EEE name-'A' test: EMPTI Eccount set bilance-bilance: 30 #EEE name-'A' test: Set CET 10 To A test: Set CET 10 Toronting request: CET 10 To A test: Set CET 10 Toronting request: CET 10 Toront
Nuncti success Nuncti success	mencies accessi accession appropriate all constraints and accession accession accession accession accession accession accession accession accession accession accession accession accession accession acc		encode access teaching regards (EFT / Benoting regards (EFT / Benoti
whoth second whoth second III whoth second IIII whoth second IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	encode successi (contrary magent): GPTA successions: GPTA Contrart MEER mane-VA encode successi SUG of preparent contrarts and successions (contrary magent): GPTA 10 A (contrary magent): GPTA 10 A		encode access Localing inquest: Gran Access and access of RDA account BEEE name-'A' encode access of RDA account BEEE name-'A' BYO Property Construction and Indonesian access of RDA account seconds access
whords access 51 whords access 53 whords access 54 whore a more second meter name 'A' more than access 54 whore access 5	Periods accessing control in property and a motor to access and a second accessing accessing accessing there non-'A' accessing accessing accessing there non-'A' accessing accessing accessing there non-'A' accessing accessing accessing accessing accessing there non-'A' accessing accessing accessing accessing accessing there non-'A' accessing accessing accessing accessing accessing there non-'A' accessing accessing accessing therein accessing accessing accessing accessing therein accessing therein accessing therein accessing accessing accessing accessing therein accessing therein accessing therein accessing therein accessing therein accessing therein accessing the accessing therein accessing therein accessing therein accessing therein accessing the accessing therein accessing therein accessing therein accessing therein accessing therein accessing the accesi		encode access transfer quegate: CET 1 transfer quegate: CET 1 transf
(a) (UPDT account set balance-balance-JB BEEE name-'A' hards account set balance-balance-JB BEEE name-'A' (a): SLLT balance FRM account BEEE name-'A' (b): SLLT balance FRM account BEEE name-'A' (c): SLLT balance FRM account BEEE name-'A'	mencies excessi control my request at an excession tests numeric's excess excess pointing request at an excession tests numeric's excess excess pointing request: ETA to A excession tests and the excession tests numeric's excession tests and the excession tests and the excession tests numeric's excession tests and the excession tests numeric's excession tests and the excession tests numeric's excession tests and tests numeric's excession tests and tes		encode access terrority reparts: CET 1 terrority reparts: CET 2 terrority reparts: CET 2 ter
Nexts Accounts Nexts Accounts FIGH Eccounts REEE nome-'A' Nexts Accounts FIGH Eccounts REEE nome-'A' Nexts Accounts Accounts Accounts REEE nome-'A' Accounts Accounts Accounts Accounts REEE nome-'A' Accounts Accounts Accounts Accounts REEE nome-'A' Accounts Accounts Accounts REEE nome-'A'	Periods accessing control in property and a motor to access and a second accessing accessing accessing there non-'A' accessing accessing accessing there non-'A' accessing accessing accessing there non-'A' accessing accessing accessing accessing accessing there non-'A' accessing accessing accessing accessing accessing there non-'A' accessing accessing accessing accessing accessing there non-'A' accessing accessing accessing therein accessing accessing accessing accessing therein accessing therein accessing therein accessing accessing accessing accessing therein accessing therein accessing therein accessing therein accessing therein accessing therein accessing the accessing therein accessing therein accessing therein accessing therein accessing the accessing therein accessing therein accessing therein accessing therein accessing therein accessing the accesi		encode access transfer quegate: CET 1 transfer quegate: CET 1 transf
Nuncti success Nuncti success	mencies excessi control my request at an excession tests numeric's excess excess pointing request at an excession tests numeric's excess excess pointing request: ETA to A excession tests and the excession tests numeric's excession tests and the excession tests and the excession tests numeric's excession tests and the excession tests numeric's excession tests and the excession tests numeric's excession tests and tests numeric's excession tests and tes		encode access terrority reparts: CET 1 terrority reparts: CET 2 terrority reparts: CET 2 ter

(1000 iterations)

× root@ip-172-31-8-111./home/ec2-user/CS244b_final_project/rso/bf/bft-simple	E × rost@ip-172-31-1-74./home/ec2-user/CS244b	nal_project/soc/bftlb/t-simple (ssh)
[root@ip-172-31-8-111 bft-simple]# ./simple-client	sql: UPDATE account set balance-balance	
WARNING: disabled multicast	execute success	
Elapsed time 124.810318 for 1000 iterations of operation 0	incoming request: GET A	
[root@ip-172-31-8-111 bft-simple]# ./simple-client	sql: SELECT balance FROM account WHERE	
WARNING: disabled multicast	execute success	
Elapsed time 2.314138 for 1000 iterations of operation 0	50878	
<pre>[root@ip-172-31-8-111 bft-simple]# ./simple-client -t 5000</pre>	incoming request: DEPOSIT 10 TO A	
WARNING: disabled multicast	sql: UPDATE account set balance=balance	-10 WHERE name='A'
Enter command:	execute success	
reply: Elapsed time 2.142641 for 1000 iterations of operation 0	incoming request: GET A sol: SELECT balance FROM account WHERE	
<pre>[root@ip-172-31-8-111 bft-simple]# ./simple-client -i 5000 NARNING: disabled multicast</pre>	sql: SELECT balance FNDM account WHERE execute success	same='A'
Elapsed time 18.341013 for 5000 iterations of operation 0	EXECUTE SUCCESS 50080	
[root@ip-172-31-8-111 bft-simple]∉ ./simple-client -i 10000	incoming request: DEPOSIT 10 TO A	
WARDING: disabled multicast	sql: UPDATE account set balance-balance	10 WHERE roman's'
Elapsed time 31.824320 for 10000 iterations of operation 0	execute success	
FrontBig-172-31-8-111 bft-simple1# ^C	incoming request: GET A	
[root#ip-172-31-8-111 bft-simple]# ./simple-client 10000	sql: SELECT balance FROM account WHERE	same="A"
WARNING: disabled multicast	execute success	
Elapsed time 2.315865 for 1000 iterations of operation 0	50090	
[root#ip-172-31-8-111 bft-simple]# ^C	incoming request: DEPOSIT 10 TO A	
<pre>[rootfip-172-31-8-111 bft-simple]# ./simple-client -i 10000</pre>	sql: UPDATE account set balance=balance	10 WHERE name='A'
WARNING: disabled multicast	execute success	
Elapsed time 22.484482 for 10000 iterations of operation 0	incoming request: GET A	
<pre>[root@ip-172-31-8-111 bft-simple]# ./simple-client -i 10000</pre>	sql: SELECT balance FROM account WHERE	name="A"
NARNING: disabled multicast Elassed time 22.545240 for 10000 iterations of operation 0	execute success 59109	
[rootBip-172-31-8-111 bft-simple]# S	50100	
[room/p-1/2-31-8-111 pre-subje]+ s	Ц	
	X root@ip-172-31-5-155/home/ec2-user/CS244b_final_project/rsc/bft/bft-simple (ssh)	× root@ip-172-31-15-153.(home/ec2-user/CS244b_final_project/rsq/bft/bft-simple (ssh)
sql: UPDATE account set balance+balance+10 WHERE name='A'	sql: UPDATE account set balance-balance+10 WHERE name='A'	sql: UPDATE account set balance=balance+10 WHERE name="A"
sql: UPDATE account set balance-balance+10 WHERE name='A' execute success	sql: UPGATE account set balance=balance=10 WHERE name='A' execute success	sql: UPDATE account set balance=balance+10 NHERE name="A" execute success
sql: UPDATE account set balance-balance+10 WHERE name='A' axacute success incoming request: GET A	sql: UPDATE account set balance-balance+10 WHERE name-'A' execute success incoming request: GET A	sql: UPDATE account set bolance-bolance-10 MHERE name='A' execute success incoming request: GET A
sql: UPDATE account set balance-balance-10 WHERE name-'A' execute success incoming request: GET A sql: SELECT balance FRDM account WHERE name-'A'	sql: UPDATE account set balance-balance-l0 WHERE name-'A' execute success incoming nequest: GET A sql: SELECT balance FROM account WHERE name-'A'	sql: UPDATE account set balance-balance-b0 MHERE name-'A' execute success incoming request: GET A sql:SELECT balance RPDM account WHERE name-'A'
sql: UPDATE account set balance-balances-10 WERE name-'A' execute success Laconfug request: GET A sql: SELECT balance FROM account WHERE name-'A' execute success	sql: UPDATE account set balance-balance-10 WERE name-"A" execute success Incoming request GET A sql: SELECT balance FROM account WERE name-"A" execute success	sql: UPGNTE account set balance-balance-s0 MHERE name-'A' execute success incoming measer: GET A sql: SELET balance MRM account WHERE name-'A' execute success
ngi: URMIT account set balance-balance-bala monate success transfir graphest: GET A success success success success 90070	<pre>sql: UPDATE account set bolance-bolance-i0 WEEE nome-'A' wexcure success incounty negate: GET A sql: SELET bolance-BAM account WEEE nome-'A' wexcute success Seara</pre>	eqt: UPDATE account set balance-ba alance-balance-balance-balance-balance-balance-balance-balance-balance-balance-balance-balance-balance-balance-ba
ng): UPATT SOCUME Set bulance-bulance-10 MERE nome-'A' socurate success socurate success socurate success socurate socurate SMA socurate MERE nome-'A' socurate socurate SMA socurate SMA socurate (social programment: SMA SOCURE 10 TO A	(a): UMMTE account set billionce-bill WERE nome-'A' execute success incentry request: GT A appl: SILIT billione FRM account WERE nome-'A' deexit success incentry request: MPCNT 18 TO A	(a): UPATT account set balance-balance-18 WEEE nome-'A' manatis success toostig request: GFT A agri: SiLLY balance HOM account WEEE nome-'A' agrids success incompley necessis: DPOST 19 TO A
sal: INFOIT account set billow-balance-18 WEEE nome-'A' secrets access tooming reparts UT A secrets access means access a	(a) URATI account set bioinen-bidinon-18 MEEE nume-14' securits success incoming request; or FRA exact set for the securit success securits success securits success (a) URATI account Set bioinen-bidinential MEEE nume-14' (a) URATI account set bioinen-bidinential MEEE nume-14'	st: UPDTI count is to bioinventillences.sB MERE nome-A' execute socies ROP accurit MERE nome-A'
ng): UPATT SOCUME Set bulance-bulance-10 MERE nome-'A' socurate success socurate success socurate success socurate socurate SMA socurate MERE nome-'A' socurate socurate SMA socurate SMA socurate (social programment: SMA SOCURE 10 TO A	(a): UMMTE account set billionce-bill WERE nome-'A' execute success incentry request: GT A appl: SILIT billione FRM account WERE nome-'A' deexit success incentry request: MPCNT 18 TO A	(a): UPATT account set balance-balance-18 WEEE nome-'A' manatis success toostig request: GFT A agri: SiLLY balance HOM account WEEE nome-'A' agrids success incompley necessis: DPOST 19 TO A
al : IFATE account set bulance-bala	(c): IVPAIT SCOUNT Set bilance-bilance-18 WEEE nume-'A' seconds success (conting requests: GT A sci: SLICT bilance Folk scount HEEE name-'A' except seconds source (scound) set bilance-bilance-18 WEEE name-'A' except seconds	(a): 1997E account set bulance-bulance-18 WEEE name-'A' manatus success (nooting request): GT A sql: SL(T) bulance FRM account: WEEE name-'A' execute success 2000 repart: RFDIT is not success and information of the success and WEEE name-'A' execute success and information of the success and WEEE name-'A' execute success
will in DMT scoots the biologeneous/active BHEE nome-'A' scoots scoots scoots scoots scoots scoots scoots are scoot scoots scoots are scoot scoots scoots are scoots scoots are scoots scoots are scoots scoots are scoots scoots are scoots scoots are specific are scoots scoots are scoots scoots are scoots scoots are scoots scoots are scoots scoots are scoots scoots are scoots scoot	(c): UPTL SCORE at billowed-science 39 BEEL name-'A' scored access access access access access control access at a control negative to the science access access access access control negative to billowed-access 18 BEEL name-'A' access access access access access access access control negative to bill non-access to BEEL name-'A' access access access access access access access control negative to A access acce	(a): 1997b account set bulance-bulance-18 WHEE nome-'A' execute success (nonling request: GT A sql: SQLT) Bulance FMM account WHEE nome-'A' examine success (nonling request: SPRDIT 18 TO A sql: UPDAT account set bulance-bulance-18 WHEE nome-'A' execute success (nonling request: SPRDIT 18 TO A sql: UPDAT account set bulance-bulance-18 (sql: UPDAT account set bulance-bulance-18 (sql: UPDAT account set bulance-bulance-18) (sql: UPDAT account set bulance-18 (sql: UPDAT account set bulance-18) (sql: UPDAT account set bulance-18) (
ILL INFORT EXCOUNT BE INDERCOMMENDATIONS IN BREEF NAME-14* INCOMENT PROJECT CAT INCOMENT PROJECT CAT INCOMENT PROJECT CAT INFORMATION CONTRACT INFORMATION INFORMATION INFORMATION CONTRACT INFORMATION INFORMATION INFORMATION CONTRACT INFORMATION INFORMATION INFORMATION CONTRACT INFORMATION INFORMATION INFORMATION CONTRACTORY INFORMATION INFORMATION INFORMATION INFORMAT	Indi: UPUT SCOUNT SEt billow-billow-18 WEE name-'A' scontor success isometing request: CFA Scontor progent: CFA Scontor progent: CFA Scontor progent: DEPORT 18 TO A scontor progent: CFA scontor progent: CFA	a): UMDTL scours tes tholoso-balance-bill WEEE name-'A' execute account set to balance-bill WEEE name-'A' execute account weeks name-'A' execute account are balance-bill weeks name-'A' execute account are balance-bill WEEE name-'A' actionation measures: CET A actionation measures CET Account WEEE name-'A' actionation account account weeks name-'A'
set: EVENT EXCOUNT SET billionshall BREEF nome-V* seconts Seconds Seconds to purphy reparts or FIRM excounts WREEF nome-V* methods Seconds Seconds Seconds 11: EVENT Excounts the billionshall Seconds BREEF nome-V* seconds Seconds 11: EVENT Excounts the billionshall Seconds BREEF nome-V* seconds Seconds 11: EVENT Excounts MREEF nome-V* seconds Seconds 11: EVENT Excounts MREEF nome-V* seconds Seconds 11: EVENT Excounts INFERE nome-V*	(a): UPUT SCOUNT SET billow-billow-b3 WEE nom-'A' social access social access social access social access social socia	<pre>st: UPDTE count is to bioins-billines-sil# MHTE nome-A' execute scores to count is not an an</pre>
ni Li 1971 Scoott se biolosobilance 38 MEE nome-4' social socia	Weill store and the information of the informati	<pre>sql: URDTL account are bilance-bila WEEE nome-'4' watching account are bilance Fill WEEE nome-'4' sql: SIGIT bilance FMM count WEEE nome-'4' sql: SIGIT bilance FMM count will BEEE nome-'4' sql: SIGIT bilance FMM count will bilance FMM count will be the sql bilance FMM count will bilance FM</pre>
ILE LEVEL EXCLUSE LAS LABORADORASIA BREER NAME-A' HEARTER SECONS LEVELSE AND	(a): UPAIL SCOUNT SEt billow-billow-bill WEE nome-'A' execute aurosa scouter property: UPA scouter property: UPA scouter aurosa WOR execute aurosa scouter aurosa scoute	set: UPDTI account set biotoco-bill wette nome-'A' execute social for requests: GT A county requests: GT A description for account WEEE nome-'A' description for account WEEE nome-'A' description for account wetter nome-'A' execute social for account WEEE nome-'A' execute social county of request: GT A execute social county of request: GT A execute social county of request: GT A execute social county of request: SPORT 30 TO A vector social county of request: SPORT 30 TO A
III I I I I I I I I I I I I I I I I I	[41] UPUT SCOLIN SEE Ibilized-billion-18 WEE nome-'A' societa access and access access and access	<pre>st: UPDTE count is to bioins-bill wells num-'4' execute scores execute scores first and the score state of the score state count is the score state count is to bioins-bill wells count is to bioins-bill wells count is to bioins-bill wells count is to bioins-bill cou</pre>
ni L'EDRIT SCOOMT SE billoreadinanci & HEEE nome-'A' matori mouto moutor mouto moutor m	The second access second access se	<pre>sql: UMTAT account as bilance-bila WHE nome-'A' executed account of the bilance-bila WHE nome-'A' executed account of the bilance influence influence influence standard account as bilance-bilance bilance influence (counting requests: EMPCIT 18 TO A executed account as bilance-bilance bilance account (counting requests: EMPCIT 18 TO A executed account as bilance-bilance bilance account (counting requests: EMPCIT 18 TO A executed account account BMEE nome-'A' accounting requests: EMPCIT 18 TO A sector account account bilance bilance bilance account (counting requests: EMPCIT 18 TO A executed account account bilance bilance bilance bilance bilance account account account bilance bilance bilance bilance bilance execute account account bilance bilance bilance bilance bilance bilance account account bilance field account BMEE nome-'A' execute account account bilance bilance</pre>
ILL INFORT EXCOUNT SET INTO AND	<pre>sql: UPDIX SCOUNT SEt billnow-billnow-bill WEEE nome-'A' execute success scound sprogram: LET A scound program: LET A scound program: LET A scound program: LET A scound success sound scound program: LET A scound pro</pre>	act: UPDTI account se bioinno-ballmon-ball MEER nom-'A' execute social for projects: CFT County Projects: CFT Account Projects: CFT
IL I UPUT EXCOUNT SEE INDERCOMMENDANCES BREEF nome-'A' secont success success success I I I I I I I I I I I I I I I I I I I	(4): UPUT SCOUT SET billow-billow-bill WEE nom-'A' sociat access and access access and access acc	<pre>st: UPDTE count is to bioince-bill were nom-'4' execute scores execute scores count is an observed the count HEEE nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute count is to bioince-bill were nom-'4' execute count is to bioince-bill were nom-'4' execute and the count is to bioince-bill were nom-'4' execute and the count is to bioince-bill were nom-'4' execute and the count is to bioince-bill were and the count is to bioince and the coun</pre>
nd : URDY Scott, the biologicalization of BEEE nome-'A' secont scot	(c): UPUT scout at bidrowkalanck3 BREE name-'A' scouts access access access access access access access and access acc	<pre>init UMTR count as bilance-bilawels memory." init UMTR count as bilance-bilawels memory." init UMTR count as bilance-bilawels memory." init UMTR count memory." init U</pre>
IL I UPUT EXCOUNT SEE INDERCOMMENDANCES BREEF nome-'A' secont success success for the success I I I I I I I I I I I I I I I I I I I	(4): UPUT SCOUT SET billow-billow-bill WEE nom-'A' sociat access and access access and access acc	<pre>st: UPDTE count is to bioince-bill were nom-'4' execute scores execute scores count is an observed the count HEEE nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute scores and the count is to bioince-bill were nom-'4' execute count is to bioince-bill were nom-'4' execute count is to bioince-bill were nom-'4' execute and the count is to bioince-bill were nom-'4' execute and the count is to bioince-bill were nom-'4' execute and the count is to bioince-bill were and the count is to bioince and the coun</pre>
ni: 1971 Excount set buildnesshall BREE nome-14' sectors sectors teaching request. 1974 Annual Sectors 1974 Excount Sectors 1974 Excount Sectors 1976 Excount Sectors 1976 Excount Sectors 1976 Excount Sectors 1977 Excount Sectors 1978 Excord 1978 Excount Sectors 1978 Excount Sectors 1978 Excount	(a): UPTA SCOUNT SET ID/SCOUND-SH #SEE nome-'A' seconds accoss scounds accoss scounds accoss scounds reparation EFT / scounds reparation EFT / scounds accoss scounds accoss scounds accoss scounds accoss scounds reparation (SCOUT SH TA) seconds accoss scounds reparation (SCOUT SH TA) seconds accoss scounds accoss scounds reparation (SCOUT SH TA) seconds accoss scounds accoss s	st: UMDTE scouts se bioteneoidinee-si# MERE nome-'4' execute scouts se bioteneoidinee-si# MERE nome-'4' execute scouts 2005 2005 2005 2005 2005 2005 2005 200
Init I SPATE SECOND. In the biologoachicanship BREE nome-'A' second socies second socies I second socies I SPATE Socies Heblic Readingtones (B REE nome-'A' social Socies I second property (I SPATE I) TO A social Socies I second property (I SPATE Socies) Heblic Readingtones (I SPATE Socies)	(4): UPUT SCOLT SET Discontinues/3 WEE nome-'A' sectors access	<pre>stillUPTI count is this black-bill WEER nome-'A' execute scores weeks the scores weeks the scores count is this score the score the score the score the score count is scores count is this score that the score th</pre>
Init I UPDE accords the biologowall-discretely BREE nome-'A' seconds Society Devices for TRAN accords REEE nome-'A' seconds Society Societ	(4): UPUT SCOLT at DOLARD SEE DATA SEE Nome-'A' seconds access and seconds access	<pre>stillWDTE scouts iss bioinco-bill WEEE nome-'A' execute scouts iss bioinco-bill weEE nome-'A' execute scouts et bioinco-bill weEE nome-'A' execute scouts et bioinco-billworts WEEE nome-'A' execute scouts scouts MEEE nome-'A' execute scouts scouts MEEE nome-'A' execute scout</pre>
set : UPUT eccords tes biologoadalaces/3 BREE nome-14' seconds access seconds access seconds access seconds access	(c): UPTL SCOLT SET (ACCOUNTS EVER Insue-A' social accost scored accost scored accost scored accost scored accost scored accost set (ACCOST ACCOST EVER Insue-A' social scored accost set (ACCOST ACCOST EVER Insue-A' social scored accost scored accost s	<pre>set: UPDTE account as the bioteneoblaces-18 MERE nome-'A' execute social information information information control information information information control information information control cont</pre>
Init I UPDE accords the biologowall-discretely BREE nome-'A' seconds Society Devices for TRAN accords REEE nome-'A' seconds Society Societ	(4): UPUT SCOLT at DOLARD SEE DATA SEE Nome-'A' seconds access and seconds access	<pre>stillWDTE scouts iss bioinco-bill WEEE nome-'A' execute scouts iss bioinco-bill weEE nome-'A' execute scouts et bioinco-bill weEE nome-'A' execute scouts et bioinco-billworts WEEE nome-'A' execute scouts scouts MEEE nome-'A' execute scouts scouts MEEE nome-'A' execute scout</pre>

(10000 iterations)

3) Experiment 3: write some and read, which means replicas show the specified customers' balance related information after they writed.

Request from client: deposit A 100, get A, deposit B 200, get B, withdraw C 100, get C Result from replicas:

Command Prom	npt - ssh -i *.pem ec2-user@	⊉13.57.32.246		Command Prompt - ssh -i *.pem ec2-user@54.193.27.25			
Replica 1:				Replica 2:			
CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	BALANCE	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	BALANCE
3 4 5	Michael Chen Wei Qin Saniav Ghemawat	NULL NULL NULL	300 1000 300	4 3 4 5	Michael Chen Wei Qin Saniav Ghemawat	NULL NULL NULL	300 1000 300
+				1			
Command Prom	npt - ssh -i *.pem ec2-user@	₽52.53.168.124		Command Pro	mpt - ssh - <mark>i</mark> *.pem ec2-use	·@54.183.251.79	
Command Prom Replica 3:	npt - ssh -i *.pem ec2-user(@52.53.168.124		Replica 4:	mpt - ssh -i *,pem ec2-user	r@54.183.251.79	
	npt - ssh -i *.pem ec2-user@ + CUSTOMER_NAME	©52.53.168.124	BALANCE		+	©54.183.251.79 + CUSTOMER_ADDRESS	BALANCE