

Vector Clock Report

1. Implementation

1) Entity.

- a) Data. Key-value store
- b) Server. IP:port as id, clock and update Data
- c) Vector_Clock. Servers with its clock

2) Service.

- a) read (Data) returns (Data).
Send key return value
- b) add_update (Data) returns (stream Server).
Send Data and increase its own clock, then propagate to all replicas
- c) Propagate (Vector_clock) returns (stream Server)
Send clock to other servers return Server which has conflict clock

3) Test on Docker. Details in docker.log.txt

```
docker build -t vector-clock .  
docker run -d -p 7000:3000 --name vector-clock-server vector-clock
```

```
< vector-clock-server vector-clock  
EXITED (0) LOGS INSPECT STATS  
Starting server on port 7000  
Starting server on port 7002  
Starting server on port 7004  
Starting server on port 7001  
Server started!  
Starting server on port 7003  
Server started!  
Server started! Server and its replicas start on port  
7000, 7001, 7002, 7003, 7004  
Server started!  
Server started!  
---update---  
key: "testData"  
val: "10" add new {testData, 10} to port 7000  
  
localhost:7000 propagate... port 7000 starts to propagate its clock  
localhost:7001 current clock{localhost:7000=1, localhost:7001=1, localhost:7002=0, localhost:7003=0, localhost:7004=0}  
propagate to localhost:7001 0.821s time to receive response  
localhost:7002 current clock{localhost:7000=1, localhost:7001=0, localhost:7002=1, localhost:7003=0, localhost:7004=0}  
propagate to localhost:7002 0.544s  
localhost:7003 current clock{localhost:7000=1, localhost:7001=0, localhost:7002=0, localhost:7003=1, localhost:7004=0}  
propagate to localhost:7003 0.614s  
localhost:7004 current clock{localhost:7000=1, localhost:7001=0, localhost:7002=0, localhost:7003=0, localhost:7004=1}  
propagate to localhost:7004 0.503s  
localhost:7000 current clock{localhost:7000=1, localhost:7001=0, localhost:7002=0, localhost:7003=0, localhost:7004=0}  
Update successfully, No Conflict No conflict  
---read---  
key: "testData" read value from replicas on port 7004  
10
```

Update simultaneously on replicas 7002 and 7003

```
java -jar Client-1.0-SNAPSHOT-jar-with-dependencies.jar -server_ip localhost -port 7002 -update testData,5
&
java -jar Client-1.0-SNAPSHOT-jar-with-dependencies.jar -server_ip localhost -port 7003 -update testData,4
```

```
---update---
key: "testData"
val: "4"

localhost:7003 propagate...
```

on 7003, update testData to 4

```
---update---
key: "testData"
val: "5"

localhost:7002 propagate...
```

on 7002 update testData to 5

```
localhost:7001 current clock{localhost:7000=1, localhost:7001=1, localhost:7002=0, localhost:7003=0, localhost:7004=0}
propagate to localhost:7001 0.953s
localhost:7002 current clock{localhost:7000=1, localhost:7001=0, localhost:7002=2, localhost:7003=0, localhost:7004=0}
propagate to localhost:7002 0.114s
localhost:7000 current clock{localhost:7000=2, localhost:7001=0, localhost:7002=0, localhost:7003=2, localhost:7004=0}
propagate to localhost:7000 0.083s
---conflicts---
localhost:7004 current clock{localhost:7000=1, localhost:7001=0, localhost:7002=0, localhost:7003=0, localhost:7004=1}
propagate to localhost:7004 0.13s
localhost:7003 current clock{localhost:7000=1, localhost:7001=0, localhost:7002=0, localhost:7003=2, localhost:7004=0}
id: "localhost:7001"
data {
  key: "testData"
  val: "10"
}

---conflicts---
id: "localhost:7002"
data {
  key: "testData"
```

```

id: "localhost:7001"
data {
  key: "testData"
  val: "10"
}

---conflicts---
id: "localhost:7000"
data {
  key: "testData"
  val: "4"
}

---conflicts---
id: "localhost:7003"
data {
  key: "testData"
  val: "4"
}

---conflicts---
id: "localhost:7004"
data {
  key: "testData"
  val: "10"
}

---read---
key: "testData"
10

```

4 conflicts

because of conflict, the value does not change

4) Test on Google Cloud.

Five instances in us-central, Hong Kong, London, Sydney, Singapore

Filter Enter property name or value									
<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect	
<input type="checkbox"/>	✓	instance-1	us-central1-a Iowa			10.128.0.2 (nic0)	34.123.165.94	SSH ▾	⋮
<input type="checkbox"/>	✓	instance-2	asia-east2-a Hong Kong			10.170.0.3 (nic0)	34.150.58.108	SSH ▾	⋮
<input type="checkbox"/>	✓	instance-3	europa-west2-c London			10.154.0.3 (nic0)	35.189.121.64	SSH ▾	⋮
<input type="checkbox"/>	✓	instance-4	australia-southeast1-b Sydney			10.152.0.3 (nic0)	34.151.69.229	SSH ▾	⋮
<input type="checkbox"/>	✓	instance-5	asia-southeast1-b Singapore			10.148.0.2 (nic0)	34.87.0.118	SSH ▾	⋮

Start the main server instance-1, and all replicas except instance-5

Add {testData, 10} to the main server — No conflict

```
aarontao16@instance-1: ~ - Google Chrome
ssh.cloud.google.com/projects/fit-asset-343701/zones/us-central1-a/instances/instance-1?authuser=0&hl=en_US&projectNumber=82574...

aarontao16@instance-1:~$ java -jar Server-1.0-SNAPSHOT-jar-with-dependencies.jar -host 10.128.0.2 -port 7000 -N 5 -ho
sts 10.170.0.3,10.154.0.3,10.152.0.3,10.148.0.2 -ports 7000,7000,7000,7000
Starting server on port 7000
Server started!
---update---
key: "testData"
val: "10"

10.128.0.2:7000 propagate...
propagate to 10.170.0.3:7000 0.835s
propagate to 10.154.0.3:7000 0.521s
propagate to 10.152.0.3:7000 0.748s
Server Error: 10.148.0.2:7000 UNAVAILABLE: io exception
propagate to 10.148.0.2:7000 0.198s
10.128.0.2:7000 current clock{10.154.0.3:7000=0, 10.148.0.2:7000=0, 10.128.0.2:7000=1, 10.152.0.3:7000=0, 10.170.0.3:
7000=0}

aarontao16@instance-1: ~ - Google Chrome
ssh.cloud.google.com/projects/fit-asset-343701/zones/us-central1-a/instances/instance-1?authuser=0&hl=en_US&projectNumber=825748472519...

aarontao16@instance-1:~$ java -jar Client-1.0-SNAPSHOT-jar-with-dependencies.jar -server_ip 10.128.0.2 -port 7000 -update te
stData,10
Update successfully, No Conflict
aarontao16@instance-1:~$
```

Read testData from instance-2 — result: 10

```
aarontao16@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/fit-asset-343701/zones/asia-east2-a/instances/instance-2?authuser=0&hl=en_US&projectNumber=825748472519&use...

aarontao16@instance-2:~$ java -jar Server-1.0-SNAPSHOT-jar-with-dependencies.jar -host 10.170.0.3 -port 7000 -N 5 -hosts 10.128
.0.2,10.154.0.3,10.152.0.3,10.148.0.2 -ports 7000,7000,7000,7000
Starting server on port 7000
Server started!
10.170.0.3:7000 current clock{10.154.0.3:7000=0, 10.148.0.2:7000=0, 10.128.0.2:7000=1, 10.152.0.3:7000=0, 10.170.0.3:7000=1}
---read---
key: "testData"

aarontao16@instance-1: ~ - Google Chrome
ssh.cloud.google.com/projects/fit-asset-343701/zones/us-central1-a/instances/instance-1?authuser=0&hl=en_US&projectNumber=825748472519...

aarontao16@instance-1:~$ java -jar Client-1.0-SNAPSHOT-jar-with-dependencies.jar -server_ip 10.128.0.2 -port 7000 -update te
stData,10
Update successfully, No Conflict
aarontao16@instance-1:~$ java -jar Client-1.0-SNAPSHOT-jar-with-dependencies.jar -server_ip 10.170.0.3 -port 7000 -read test
Data
10
```

Instance-5 is back, the send update {testData, 5} to it —conflict!

```
aarontao16@instance-5: ~ - Google Chrome
ssh.cloud.google.com/projects/fit-asset-343701/zones/asia-southeast1-b/instances/instance-5?authuser=0&hl=en_US&projectNumber=825748472...

aarontao16@instance-5:~$ java -jar Server-1.0-SNAPSHOT-jar-with-dependencies.jar -host 10.148.0.2 -port 7000 -N 5 -ho
sts 10.170.0.3,10.154.0.3,10.152.0.3,10.128.0.2 -ports 7000,7000,7000,7000
Starting server on port 7000
Server started!
---update---
key: "testData"
val: "4"

10.148.0.2:7000 propagate...
propagate to 10.170.0.3:7000 0.264s
propagate to 10.154.0.3:7000 0.866s
propagate to 10.152.0.3:7000 0.315s
propagate to 10.128.0.2:7000 0.566s
10.148.0.2:7000 current clock{10.148.0.2:7000=1, 10.154.0.3:7000=0, 10.128.0.2:7000=0, 10.152.0.3:7000=0, 10.170.0.3:
7000=0}

aarontao16@instance-1: ~ - Google Chrome
ssh.cloud.google.com/projects/fit-asset-343701/zones/us-central1-a/instances/instance-1?authuser=0&hl=en_US&projectNumber=825748472519...

aarontao16@instance-1:~$ java -jar Client-1.0-SNAPSHOT-jar-with-dependencies.jar -server_ip 10.148.0.2 -port 7000 -update_te
stData,4
---conflicts---
id: "10.170.0.3:7000"
data {
  key: "testData"
  val: "10"
}

---conflicts---
id: "10.154.0.3:7000"
data {
  key: "testData"
  val: "10"
}

---conflicts---
id: "10.152.0.3:7000"
data {
  key: "testData"
  val: "10"
}

---conflicts---
id: "10.128.0.2:7000"
data {
  key: "testData"
  val: "10"
}
```

Latency (server in US to other replicas)

US	Hong Kong	London	Sydney	Singapore
Propagate (round)	0.53s	0.30s	0.53s	0.63s
distance	12,719 km	6,848 km	12013.75 km	15289.17 km