

Tapir experiments

In this experiment, I want to get the throughput of TAPIR, varying the number of CPU cores with 3 replicas and benchClient (*simple transactional benchmark*), exactly as Meerkat (a paper in Eurosys 2020) did. We also set the number of shard for each replica as 1 to simplify the experiment.

Now, I have 4 machines running: three of them are replicas and the other one serves as the client issuing transactions. All machines are running on Ubuntu-18.04 LTS. We named three replicas and one client as breakout, pitfall, qbert and spyhunter. We have to start a timeserver + server for every replica and a benchClient for every client. The details are as follows

Bash

```
1 # on breakout
2 /home/weihsen/Desktop/tapir/timeserver/timeserver -c /home/weihsen/Desktop/tap
ir/store/tools/shard.tss.config -i 0
3 # start shard0
4 /home/weihsen/Desktop/tapir/store/tapirstore/server -m txn-l -f /home/weihsen/
Desktop/tapir/store/tools/keys -k 100000 -e 0 -s 0 -c /home/weihsen/Desktop/tap
ir/store/tools/shard0.config -i 0
5
6 # on pitfall
7 /home/weihsen/Desktop/tapir/timeserver/timeserver -c /home/weihsen/Desktop/tap
ir/store/tools/shard.tss.config -i 1
8 # start shard0
9 /home/weihsen/Desktop/tapir/store/tapirstore/server -m txn-l -f /home/weihsen/
Desktop/tapir/store/tools/keys -k 100000 -e 0 -s 0 -c /home/weihsen/Desktop/tap
ir/store/tools/shard0.config -i 1
10
11 # on qbert
12 /home/weihsen/Desktop/tapir/timeserver/timeserver -c /home/weihsen/Desktop/tap
ir/store/tools/shard.tss.config -i 2
13 # start shard0
14 /home/weihsen/Desktop/tapir/store/tapirstore/server -m txn-l -f /home/weihsen/
Desktop/tapir/store/tools/keys -k 100000 -e 0 -s 0 -c /home/weihsen/Desktop/tap
ir/store/tools/shard0.config -i 2
15
16 # on spyhuynter
17 /home/weihsen/Desktop/tapir/store/benchmark/benchClient -c /home/weihsen/Deskt
op/tapir/store/tools/shard -N 1 -f /home/weihsen/Desktop/tapir/store/tools/keys
-d 10 -l 2 -w 0 -k 100000 -m txn-l -e 0 -s 0 -z -1
```



The Meerkat also implements or tests on the TAPIR, there is no TimeServer and ViewChange inside it, thus I also comment on this piece of code in the TAPIR.

Also, the starting script `run_test.sh` is outdated and has many errors, thus we have to start `benchClient` and `server` manually.

I have succeeded in setting up the environment and compiling the project as follows.

1. Install dependencies on four machines

Bash

```
1  sudo apt install openssh-server
2  sudo apt install gcc g++
3  sudo apt install make
4
5  sudo apt install libprotobuf-dev
6  apt-get install libevent-dev
7  sudo apt-get install libssl-dev
8  sudo apt-get install -y libevent-pthreads-2.1-6
9  sudo apt-get install libssl-dev
10 sudo apt-get install protobuf-compiler
11 sudo apt install maven
12 sudo apt-get install -y pkg-config
13 sudo apt install python3.8
14 sudo apt install python
15 sudo apt install git
16
17 # install JDK
18 sudo apt install openjdk-11-jdk
19 # export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/
20 # export JAVA_INCLUDE_DIR=/usr/lib/jvm/java-11-openjdk-amd64/include
```

2. Create VM machines

- Keep all machines with the exact username for convenience.
- Make sure 4 machines can connect to each other via ssh
- Add host alias to /etc/hosts so that we can connect to other VMs with alias, i.e., ssh breakout.

Bash

```
1  # replica-0
2  192.168.139.157 breakout
3  # replica-1
4  192.168.139.158 pitfall
5  # replica-2
6  192.168.139.159 qbert
7  # the client
8  192.168.139.160 spyhunter
```

3. Compile repo on the four machines

Apache

```
1 cd ~/Desktop/
2 mkdir -p ~/logs/tapir
3 git clone https://github.com/shenweihai1/tapir.git
4
5 # MODIFY: ./store/tools/run_test.sh accordingly
6 # srcdir="/home/weihshen/Desktop/tapir"
7 # logdir="/home/weihshen/logs/tapir"
8
9 # MODIFY: comment #DNASSERT for
10
11 cd ./tapir/
12 # generate dataset
13 python key_generator.py 100000 > keys
14 make -j8
```

4. Run the experiment:

Bash

```
1 # replica0
2 ./r0.sh
3
4 # replica1
5 ./r1.sh
6
7 # replica2
8 ./r2.sh
9
10 # client: in this client-server, I will start 10 processes of Clients
11 ./c10.sh
```