

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ POSTGRES	2
НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ TARANTOOL	4
ВЫВОД	6

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ POSTGRES

В качестве БД – *Postgres 14*

Берем сервис chats, генерируем 10 000 пар друзей, у каждого 20 чатов с 20 сообщениями, итого 4ккк данных в таблицу chats.

Но это долго, поэтому ограничимся 1ккк.

Запрос на чтение:

```
SELECT
  c.id,
  c.sender_id,
  c.receiver_id,
  c.text,
  c.created
FROM chats c
WHERE c.sender_id=$1 AND c.receiver_id=$2
ORDER BY c.created DESC;
```

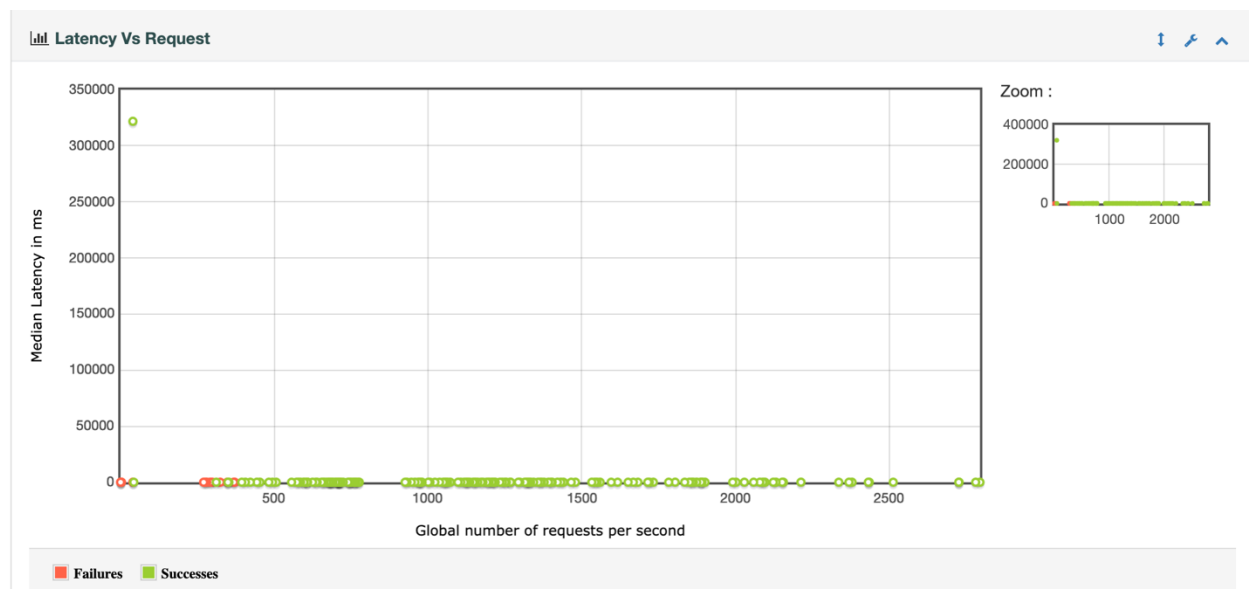
★ Favorites

🏠 Structure

🔍 Pull Requests

Created the tree successfully using ./HTTP_read_chat.jmx
Starting standalone test @ 2023 Mar 6 01:12:56 MSK (1678054376748)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary + 1495 in 00:00:03 = 468.8/s Avg: 1 Min: 1 Max: 60 Err: 0 (0.00%) Active: 1 Started: 1 Finished: 0
summary + 20073 in 00:00:30 = 669.1/s Avg: 1 Min: 0 Max: 84 Err: 0 (0.00%) Active: 1 Started: 1 Finished: 0
summary = 21568 in 00:00:33 = 649.9/s Avg: 1 Min: 0 Max: 84 Err: 0 (0.00%)
summary + 22138 in 00:00:30 = 737.9/s Avg: 2 Min: 1 Max: 156 Err: 0 (0.00%) Active: 10 Started: 11 Finished: 1
summary = 43706 in 00:01:03 = 691.7/s Avg: 1 Min: 0 Max: 156 Err: 0 (0.00%)
summary + 60696 in 00:00:30 = 2023.2/s Avg: 4 Min: 1 Max: 399 Err: 0 (0.00%) Active: 10 Started: 11 Finished: 1
summary = 104402 in 00:01:33 = 1120.3/s Avg: 3 Min: 0 Max: 399 Err: 0 (0.00%)
summary + 41405 in 00:00:30 = 1380.2/s Avg: 12 Min: 1 Max: 2186 Err: 0 (0.00%) Active: 100 Started: 111 Finished: 11
summary = 145807 in 00:02:03 = 1183.6/s Avg: 6 Min: 0 Max: 2186 Err: 0 (0.00%)
summary + 37971 in 00:00:30 = 1265.7/s Avg: 78 Min: 1 Max: 397 Err: 0 (0.00%) Active: 100 Started: 111 Finished: 11
summary = 183778 in 00:02:33 = 1199.7/s Avg: 21 Min: 0 Max: 2186 Err: 0 (0.00%)
summary + 34457 in 00:02:34 = 223.6/s Avg: 1536 Min: 2 Max: 75001 Err: 1890 (5.49%) Active: 1000 Started: 1111 Finished: 111
summary = 218235 in 00:05:07 = 710.1/s Avg: 260 Min: 0 Max: 75001 Err: 1890 (0.87%)
summary + 946 in 00:01:00 = 15.8/s Avg: 75254 Min: 75001 Max: 187082 Err: 946 (100.00%) Active: 54 Started: 1111 Finished: 1057
summary = 219181 in 00:06:07 = 597.0/s Avg: 584 Min: 0 Max: 187082 Err: 2836 (1.29%)
summary + 3 in 00:00:28 = 0.1/s Avg: 212913 Min: 211525 Max: 214748 Err: 3 (100.00%) Active: 51 Started: 1111 Finished: 1060
summary = 219184 in 00:06:35 = 555.2/s Avg: 587 Min: 0 Max: 214748 Err: 2839 (1.30%)
summary + 6 in 00:00:59 = 0.1/s Avg: 244481 Min: 214741 Max: 273700 Err: 6 (100.00%) Active: 45 Started: 1111 Finished: 1066
summary = 219190 in 00:07:34 = 483.0/s Avg: 593 Min: 0 Max: 273700 Err: 2845 (1.30%)
summary + 44 in 00:00:48 = 0.9/s Avg: 318580 Min: 273691 Max: 322022 Err: 40 (90.91%) Active: 0 Started: 1111 Finished: 1111
summary = 219234 in 00:08:22 = 436.6/s Avg: 657 Min: 0 Max: 322022 Err: 2885 (1.32%)
Tidying up ... @ 2023 Mar 6 01:21:18 MSK (1678054878994)

Statistics													
Requests		Executions		Response Times (ms)							Throughput	Network (KB/sec)	
Label ^	#Samples ^	FAIL ^	Error % ^	Average ^	Min ^	Max ^	Median ^	90th pct ^	95th pct ^	99th pct ^	Transactions/s ^	Received ^	Sent ^
Total	219234	2885	1.32%	657.63	0	322022	66.00	25985.00	73970.65	75046.00	436.96	3550.79	180.65
HTTP Request	219234	2885	1.32%	657.63	0	322022	66.00	25985.00	73970.65	75046.00	436.96	3550.79	180.65



docker-compose

```
chat-postgres:
  image: postgres:14
  environment:
    POSTGRES_DB: "default-db"
    POSTGRES_USER: "someuser"
    POSTGRES_PASSWORD: "p@ssw0rd"
  ports:
    - "5432:5432"
```

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ TARANTOOL

В качестве БД – *Tarantool 2.10.5*

Берем сервис chats, генерируем 10 000 пар друзей, у каждого 20 чатов с 20 сообщениями, итого 4ккк данных в таблицу chats.

Но это долго, поэтому ограничимся 1ккк.

Запрос на чтение:

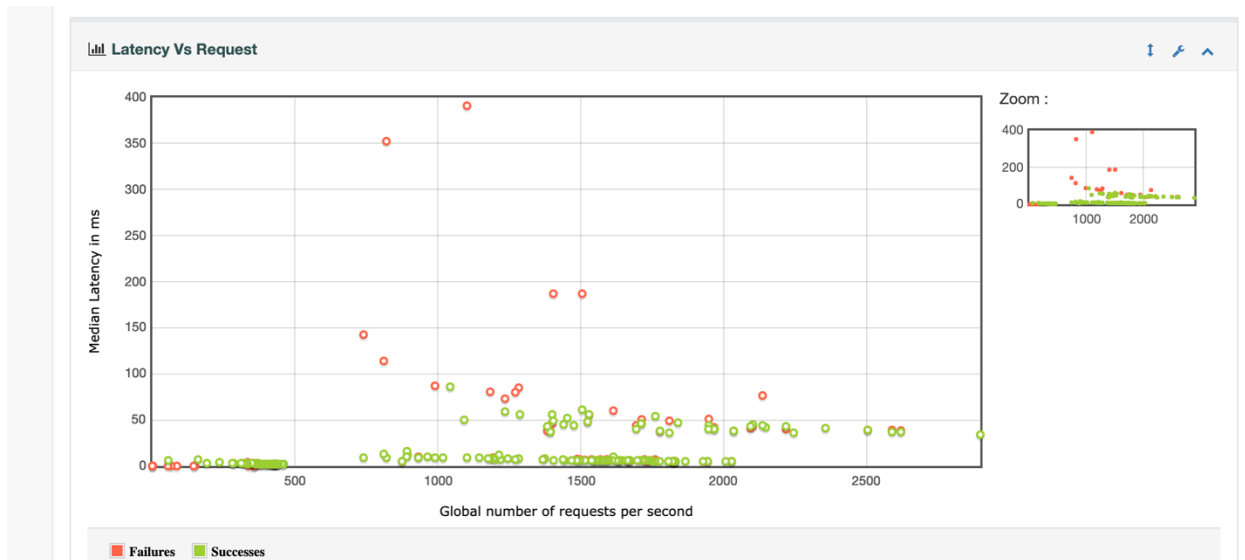
```
function select_chats(sender_id, receiver_id)
  local chats = box.space.chats.index.sender_receiver:select({sender_id,
receiver_id}, {iterator = 'EQ'})
  table.sort(chats, function(a, b) return a[5] > b[5] end)
  return chats
end
```

Push ME

Created the tree successfully using ./HTTP_read_chat.jmx
Starting standalone test @ 2023 Mar 6 02:01:39 MSK (1678057299935)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary + 6604 in 00:00:20 = 330.2/s Avg: 2 Min: 2 Max: 86 Err: 80 (1.21%) Active: 1 Started: 1 Finished: 0
summary + 12235 in 00:00:30 = 407.8/s Avg: 2 Min: 1 Max: 82 Err: 100 (0.82%) Active: 1 Started: 1 Finished: 0
summary = 18839 in 00:00:50 = 376.8/s Avg: 2 Min: 1 Max: 86 Err: 180 (0.96%)
summary + 38404 in 00:00:30 = 1280.1/s Avg: 5 Min: 1 Max: 66 Err: 360 (0.94%) Active: 10 Started: 11 Finished: 1
summary = 57243 in 00:01:20 = 715.5/s Avg: 4 Min: 1 Max: 86 Err: 540 (0.94%)
summary + 48386 in 00:00:30 = 1612.9/s Avg: 6 Min: 3 Max: 85 Err: 520 (1.07%) Active: 10 Started: 11 Finished: 1
summary = 105629 in 00:01:50 = 960.2/s Avg: 5 Min: 1 Max: 86 Err: 1060 (1.00%)
summary + 51852 in 00:00:30 = 1728.4/s Avg: 39 Min: 4 Max: 2186 Err: 500 (0.96%) Active: 100 Started: 111 Finished: 11
summary = 157481 in 00:02:20 = 1124.8/s Avg: 16 Min: 1 Max: 2186 Err: 1560 (0.99%)
summary + 42750 in 00:00:30 = 1425.0/s Avg: 70 Min: 6 Max: 340 Err: 440 (1.03%) Active: 100 Started: 111 Finished: 11
summary = 200231 in 00:02:50 = 1177.8/s Avg: 27 Min: 1 Max: 2186 Err: 2000 (1.00%)
summary + 11667 in 00:01:05 = 180.3/s Avg: 2271 Min: 5 Max: 54640 Err: 1053 (9.03%) Active: 1000 Started: 1111 Finished: 111
summary = 211898 in 00:03:55 = 902.8/s Avg: 151 Min: 1 Max: 54640 Err: 3053 (1.44%)
summary + 937 in 00:01:29 = 10.6/s Avg: 75137 Min: 54640 Max: 143358 Err: 937 (100.00%) Active: 65 Started: 1111 Finished: 1046
summary = 212835 in 00:05:23 = 658.1/s Avg: 481 Min: 1 Max: 143358 Err: 3990 (1.87%)
summary + 64 in 00:01:12 = 0.9/s Avg: 209032 Min: 143365 Max: 215305 Err: 64 (100.00%) Active: 0 Started: 1111 Finished: 1111
summary = 212899 in 00:06:35 = 538.4/s Avg: 544 Min: 1 Max: 215305 Err: 4054 (1.90%)
Tidying up ... @ 2023 Mar 6 02:08:15 MSK (1678057695462)
... end of run
a m tsitu1skiv@macbook-C02F8RA4MD6R highload-architect %

Externally added files can
[View Files](#) [Always Add](#)

Statistics													
Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label ^	#Samples ^	FAIL ^	Error % ^	Average ^	Min ^	Max ^	Median ^	90th pct ^	95th pct ^	99th pct ^	Transactions/s ^	Received ^	Sent ^
Total	212899	4054	1.90%	544.25	1	215305	74.00	361.00	73982.00	75071.98	538.82	4276.28	223.69
HTTP Request	212899	4054	1.90%	544.25	1	215305	74.00	361.00	73982.00	75071.98	538.82	4276.28	223.69



```
chat-tarantool:  
image: tarantool/tarantool  
environment:  
  TARANTOOL_MEMTX_MEMORY: 671088640  
ports:  
  - "3301:3301"
```

ВЫВОД

Обе БД ближе к концу вылетали с «unexpected EOF»

Postgresql с дефолтными настройками лучше держит нагрузку, чем Tarantool

Возможно потому что у Postgres'а был хороший индекс

```
CREATE INDEX IF NOT EXISTS chats_sender_receiver_created_idx ON chats  
(sender_id, receiver_id, created DESC);
```