Mosquitto Cluster Test Report

Menu

Functional Test

Test environment
The cluster contains 3 brokers, which called broker2/ broker3 /broker4.
There's a lot of cluster log which shows the actual message flow inside the cluster.

1. SUBSCRIBE/ UNSUNSCRIBE Messages Test

Non-duplicate SUBSCRIBE, local broker which receive client subscription would send private SUBSCRIBE to other brokers, which contain subscription topic.

2017-12-25 10:30:37:658032 Received SUBSCRIBE from mosqsub|2495-broker2(sock:11)

2017-12-25 10:30:37:658101 /test/sub (QoS 0)

2017-12-25 10:30:37:658125 Client mosqsub|2495-broker2 subscribe for topic: /test/sub. This sub is fresh for local broker, fresh for client.

2017-12-25 10:30:37:658148 [CLUSTER] Sending private subscribe to node:node3 (sub_client_id:mosqsub|2495-broker2 subid:1 topic:/test/sub qos:0 mid:1)

2017-12-25 10:30:37:658179 [CLUSTER] Sending private subscribe to node:node4 (sub_client_id:mosqsub|2495-broker2 subid:1 topic:/test/sub qos:0 mid:1)

2017-12-25 10:30:37:658316 mosqsub|2495-broker2 0 /test/sub

2017-12-25 10:30:37:658360 Sending SUBACK to mosqsub|2495-broker2

Duplicate SUBSCRIBE, local broker would NOT send private SUBSCRIBE to other broker.

2017-12-25 10:30:55:958156 Received SUBSCRIBE from mosqsub|2498-broker2(sock:12)

2017-12-25 10:30:55:958297 /test/sub (QoS 0)

2017-12-25 10:30:55:958327 Client mosqsub|2498-broker2 subscribe for topic: /test/sub. This sub is stale for local broker, fresh for client.

2017-12-25 10:30:55:958364 mosqsub|2498-broker2 0 /test/sub

2017-12-25 10:30:55:958380 Sending SUBACK to mosqsub|2498-broker2

Local SUBSCRIBE clean, if a topic no longer subscribed by any client, the local broker would send a UNSUBSCRIBE message to other brokers.

2017-12-25 10:33:10:151273 [TOPIC_TABLE] Client mosqsub|2495-broker2 disconnecting.. total(1), subscribed topic(0): /test/sub. ref_cnt = 2

2017-12-25 10:33:10:151343 Socket error on client mosqsub|2495-broker2, disconnecting.

…

2017-12-25 10:33:20:789692 [TOPIC_TABLE] Client mosqsub|2498-broker2 disconnecting.. total(1), subscribed topic(0): /test/sub. ref_cnt = 1

2017-12-25 10:33:20:789769 [CLUSTER] sending UNSUBSCRIBE to node: node3 (nrTopics:1,Mid: 2)

2017-12-25 10:33:20:789786 UNSUBSCRIBE topic[0]: /test/sub

2017-12-25 10:33:20:792406 [CLUSTER] sending UNSUBSCRIBE to node: node4 (nrTopics:1,Mid: 2)

2017-12-25 10:33:20:792489 UNSUBSCRIBE topic[0]: /test/sub

2017-12-25 10:33:20:792624 Socket error on client mosqsub|2498-broker2, disconnecting.

Duplicate UNSUBSCRIBE:
To be tested.

Invalid UNSUBSCRIBE(unsubscribe the topic which has not been subscribed by the client):
To be tested.

2. PUBLISH Message Test

mosqsub|2503-broker2 subscribe with the topic "/test/pub" on broker2, client mosqpub|1994-broker3 and mosqpub|2066-broker4 publish a message with topic "/test/pub" on broker3/4, client mosqsub|2503-broker2 receive the PUBLISH message as expected.

2017-12-25 10:39:1:347039 Received SUBSCRIBE from mosqsub|2503-broker2(sock:11)

2017-12-25 10:39:1:347104 /test/pub (QoS 0)

2017-12-25 10:39:1:347125 Client mosqsub|2503-broker2 subscribe for topic: /test/pub. This sub is fresh for local broker, fresh for client.

2017-12-25 10:39:1:347149 [CLUSTER] Sending private subscribe to node:node3 (sub_client_id:mosqsub|2503-broker2 subid:2 topic:/test/pub qos:0 mid:3)

2017-12-25 10:39:1:347177 [CLUSTER] Sending private subscribe to node:node4 (sub_client_id:mosqsub|2503-broker2 subid:2 topic:/test/pub qos:0 mid:3)

2017-12-25 10:39:35:731739 [CLUSTER] Received PUBLISH from node: node3 (d0, q0, r0, m0, '/test/pub', ... (15 bytes))

2017-12-25 10:39:35:731918 Sending PUBLISH to mosqsub|2503-broker2(sock:11) (d0, q0, r0, m0, '/test/pub', ... (15 bytes))

2017-12-25 10:39:46:281111 [CLUSTER] Received PUBLISH from node: node4 (d0, q0, r0, m0, '/test/pub', ... (15 bytes))

2017-12-25 10:39:46:284872 Sending PUBLISH to mosqsub|2503-broker2(sock:11) (d0, q0, r0, m0, '/test/pub', ... (15 bytes))

root@broker2:~# mosquitto_sub -t /test/pub

this is broker3

this is broker4

3. QoS Test

Test PUB/SUB with topic "/test/qos" with a single broker, subscriber choose QoS1, publisher choose QoS2, subscriber receive the PUBLISH message with QoS=1 as expected.

root@broker2:~# mosquitto_sub -t /test/qos -q 1

Pub QoS:2

root@broker2:~# mosquitto_pub -t /test/qos -q 2 -m "Pub QoS:2"

2017-12-25 10:49:4:518885 Received SUBSCRIBE from mosqsub|2506-broker2(sock:11)

2017-12-25 10:49:4:518949 /test/qos (QoS 1)

…

2017-12-25 10:49:14:788929 Received PUBLISH from mosqpub|2507-broker2 (d0, q2, r0, m1, '/test/qos', ... (9 bytes)) at 5954

2017-12-25 10:49:14:789001 Sending PUBREC to mosqpub|2507-broker2 (Mid: 1)

2017-12-25 10:49:14:790264 Received PUBREL from mosqpub|2507-broker2 (Mid: 1)

2017-12-25 10:49:14:790380 Sending PUBCOMP to mosqpub|2507-broker2 (Mid: 1)

2017-12-25 10:49:14:790432 Sending PUBLISH to mosqsub|2506-broker2(sock:11) (d0, q1, r0, m1, '/test/qos', ... (9 bytes))

2017-12-25 10:49:14:791455 Received PUBACK from mosqsub|2506-broker2 (Mid: 1)

Test PUB/SUB with topic "/test/qos_cluster" between brokers, the PUBLISH/SUBSCRIBE message inside cluster choose the original QoS value which has decided by the client. In order to accelerate the messages processing, whatever the QoS is, the broker deal this "internal" PUBLISH/SUBSCRIBE message with QoS=0. The actual QoS decision made on the broker which subscription client connected with.

root@broker3:~# mosquitto_sub -t /test/qos_cluster -q 2

Pub QoS:1

root@broker2:~# mosquitto_pub -t /test/qos_cluster -q 1 -m "Pub QoS:1"

broker2.log:

2017-12-25 10:52:28:544692 [CLUSTER] Received PRIVATE SUBSCRIBE from peer: broker3, topic: /test/qos_cluster, client_id: mosqsub|2003-broker3, sub_id: 1

…

2017-12-25 10:52:39:166417 Received PUBLISH from mosqpub|2510-broker2 (d0, q1, r0, m1, '/test/qos_cluster', ... (9 bytes)) at 6158

2017-12-25 10:52:39:166547 Sending PUBACK to mosqpub|2510-broker2 (Mid: 1)

2017-12-25 10:52:39:166616 Sending PUBLISH to broker3(sock:7) (d0, q1, r0, m1, '/test/qos_cluster', ... (9 bytes))

broker3.log:

2017-12-25 10:52:27:786716 Received SUBSCRIBE from mosqsub|2003-broker3(sock:11)

2017-12-25 10:52:27:786758 /test/qos_cluster (QoS 2)

2017-12-25 10:52:27:786789 [CLUSTER] Sending private subscribe to node:node2 (sub_client_id:mosqsub|2003-broker3 subid:1 topic:/test/qos_cluster qos:2 mid:1)

…

2017-12-25 10:52:38:452424 [CLUSTER] Received PUBLISH from node: node2 (d0, q1, r0, m1, '/test/qos_cluster', ... (9 bytes))

2017-12-25 10:52:38:452692 Sending PUBLISH to mosqsub|2003-broker3(sock:11) (d0, q1, r0, m1, '/test/qos_cluster', ... (9 bytes))

2017-12-25 10:52:38:453673 Received PUBACK from mosqsub|2003-broker3 (Mid: 1)

3. Retain Message test

Publish a retain message at broker3 and broker4 successively, then a client make the subscription at broker2 as a fresh(topic) subscription for broker2. The client received the latest retain message as expected.

root@broker3:~# date && mosquitto_pub -t /test/retain -r -m "retain.broker3.2017/12/25 11:06"

Mon Dec 25 11:07:01 CST 2017

Broker3.log

2017-12-25 11:7:1:124923 Received PUBLISH from mosqpub|2009-broker3 (d0, q0, r1, m0, '/test/retain', ... (31 bytes)) at 7019

root@broker4:~# date && mosquitto_pub -t /test/retain -r -m "retain.broker4.2017/12/25 11:08"

Mon Dec 25 11:08:22 CST 2017

Broker4.log

2017-12-25 11:8:22:603731 Received PUBLISH from mosqpub|2073-broker4 (d0, q0, r1, m0, '/test/retain', ... (31 bytes)) at 7099

root@broker2:~# date && mosquitto_sub -t /test/retain

Mon Dec 25 11:10:28 CST 2017

retain.broker4.2017/12/25 11:08

Broker2.log

2017-12-25 11:10:28:51513 Received SUBSCRIBE from mosqsub|2516-broker2(sock:11)

2017-12-25 11:10:28:51787 /test/retain (QoS 0)

2017-12-25 11:10:28:51855 [CLUSTER] Sending private subscribe to node:node3 (sub_client_id:mosqsub|2516-broker2 subid:4 topic:/test/retain qos:0 mid:7)

2017-12-25 11:10:28:51889 [CLUSTER] Sending private subscribe to node:node4 (sub_client_id:mosqsub|2516-broker2 subid:4 topic:/test/retain qos:0 mid:7)

2017-12-25 11:10:28:99080 [CLUSTER] Receive private retain from node:node4 at:7227 (orig_client_id:mosqsub|2516-broker2 subid:4 topic:/test/retain qos:0 mid:0 orig_rcv_time:7101 payloadlen:31)

2017-12-25 11:10:28:99214 save retain msg for client: mosqsub|2516-broker2, retain topic: /test/retain

2017-12-25 11:10:28:99252 [CLUSTER] Receive private retain from node:node3 at:7227 (orig_client_id:mosqsub|2516-broker2 subid:4 topic:/test/retain qos:0 mid:0 orig_rcv_time:7021 payloadlen:31), but local has a fresh retain(7101)

2017-12-25 11:10:29:100150 Sending PUBLISH to mosqsub|2516-broker2(sock:11) (d0, q0, r1, m0, '/test/retain', ... (31 bytes))

broker3.log

2017-12-25 11:10:27:380256 [CLUSTER] Received PRIVATE SUBSCRIBE from peer: broker2, topic: /test/retain, client_id: mosqsub|2516-broker2, sub_id: 4

2017-12-25 11:10:27:380473 [CLUSTER] Sending private retain to peer:broker2 (remote_client_id:mosqsub|2516-broker2 subid:4 topic:/test/retain qos:0 mid:0 remote_rcv_time:7021 payloadlen:31, local_rcv_time:7019, time_off_set:2)

broker4.log

2017-12-25 11:10:28:221593 [CLUSTER] Received PRIVATE SUBSCRIBE from peer: broker2, topic: /test/retain, client_id: mosqsub|2516-broker2, sub_id: 4

2017-12-25 11:10:28:221755 [CLUSTER] Sending private retain to peer:broker2 (remote_client_id:mosqsub|2516-broker2 subid:4 topic:/test/retain qos:0 mid:0 remote_rcv_time:7101 payloadlen:31, local_rcv_time:7099, time_off_set:2)

Stale local subscription, which means this topic has already been subscribed by some clients on this broker. The latter client receive the retain message immediately since this topic has been subscribed by this broker to other broker.

root@broker2:~# date && mosquitto_sub -t /test/retain

Mon Dec 25 11:15:37 CST 2017

retain.broker4.2017/12/25 11:08

broker2.log

2017-12-25 11:15:37:407604 Received SUBSCRIBE from mosqsub|2519-broker2(sock:12)

2017-12-25 11:15:37:407658 /test/retain (QoS 0)

2017-12-25 11:15:37:407677 Client mosqsub|2519-broker2 subscribe for topic: /test/retain. This sub is stale for local broker, fresh for client.

2017-12-25 11:15:37:407716 mosqsub|2519-broker2 0 /test/retain

2017-12-25 11:15:37:407731 Sending SUBACK to mosqsub|2519-broker2

2017-12-25 11:15:37:408779 Sending PUBLISH to mosqsub|2519-broker2(sock:12) (d0, q0, r1, m0, '/test/retain', ... (31 bytes))

Description:

Broker3 and broker4 save the retain message while no client subscribed this topic.
When broker2 first time receive a client subscription, it would broadcast the PRIVATE SUBSCRIBE, broker3 and broker4 receive this PRIVATE SUBSCRIBE and give a response with PRIVATE RETAIN which contain the saved retain message and the receive timestamp. Broker2 receive these 2 retain messages, after a frozen window(1 second), choose the latest retain message by comparing the timestamp, and send it to the actual subscription client.
The second time broker receive this retain topic subscription, it would NOT broadcast the PRIVATE SUBSCRIBE message since the topic subscription was sent to other brokers so the local has exists the latest retain message.
While all the clients on this broker has unsubscribed this topic, the new “first time” subscription would still cause one second delay as local has clear the inter-cluster subscription.

4. Clean Session Test

Modify the souce code lib/mosquito.c line 1045, sleep(reconnect_delay) to sleep(10000) in order to prevent mosquito_sub process made the auto reconnection.
Client “subscriber_broker2” make a connection with clean_session=true, and subscribe with topic “/test/cleansession1” on broker2, then make the reconnection with clean_session=true and subscribe with topic “/test/cleansession2” also on broker2, then another client publish the messages with “/test/cleansession1” and “/test/cleansession2” on broker3, only message with topic “/test/cleansession2” has been received by “subscriber_broker2” as expected.

```
root@broker2:~# date && mosquitto_sub -t /test/cleansession1 -i subscriber_broker2
Mon Dec 25 11:29:14 CST 2017
root@broker2:~# date && mosquitto_sub -t /test/cleansession2 -i subscriber_broker2
Mon Dec 25 11:30:43 CST 2017
cleansession2.broker3.2017/12/25 11:31
root@broker3:~# date && mosquitto_pub -t /test/cleansession1 -m "cleansession1.broker3.2017/12/25 11:31"
Mon Dec 25 11:31:03 CST 2017
root@broker3:~# date && mosquitto_pub -t /test/cleansession2 -m "cleansession2.broker3.2017/12/25 11:31"
Mon Dec 25 11:31:35 CST 2017
```

While the twice connection and subscriptions set by clean_session=false, the subscriber_broker2 would receive all the message with “/test/cleansession1” and “/test/cleansession2” as expected.

```
root@broker2:~# date && mosquitto_sub -t /test/cleansession1 -c -i subscriber_broker2
Mon Dec 25 11:47:19 CST 2017
root@broker2:~# date && mosquitto_sub -t /test/cleansession2 -c -i subscriber_broker2_cleansession
Mon Dec 25 11:48:14 CST 2017
cleansession1.broker3
cleansession2.broker3
root@broker3:~# date && mosquitto_pub -t /test/cleansession1 -m "cleansession1.broker3"
Mon Dec 25 11:48:31 CST 2017
root@broker3:~# date && mosquitto_pub -t /test/cleansession2 -m "cleansession2.broker3"
Mon Dec 25 11:48:39 CST 2017
```

While the twice connection and subscription was successively made on broker2 and broker3 with clean_session=true, client subscriber_test would only receive the message with the latter subscription “/test/cleansession2” as expected.

root@broker2:~# date && mosquitto_sub -t /test/cleansession1 -i subscriber_test

Mon Dec 25 14:34:25 CST 2017

root@broker3:~# date && mosquitto_sub -t /test/cleansession2 -i subscriber_test

Mon Dec 25 14:34:36 CST 2017

hello, cleansession2

root@broker2:~# date && mosquitto_pub -t /test/cleansession2 -m "hello, cleansession2"

Mon Dec 25 14:36:24 CST 2017

broker2.log

2017-12-25 14:34:25:197292 [CLUSTER] Sending session request to node: node3 (client_id:subscriber_test)

2017-12-25 14:34:25:198120 [CLUSTER] Sending session request to node: node4 (client_id:subscriber_test)

2017-12-25 14:34:25:200892 Received SUBSCRIBE from subscriber_test(sock:11)

2017-12-25 14:34:25:200988 /test/cleansession1 (QoS 0)

2017-12-25 14:34:25:201030 Client subscriber_test subscribe for topic: /test/cleansession1. This sub is fresh for local broker, fresh for client.

2017-12-25 14:34:25:201083 [CLUSTER] Sending private subscribe to node:node3 (sub_client_id:subscriber_test subid:1 topic:/test/cleansession1 qos:0 mid:1)

2017-12-25 14:34:25:201119 [CLUSTER] Sending private subscribe to node:node4 (sub_client_id:subscriber_test subid:1 topic:/test/cleansession1 qos:0 mid:1)

2017-12-25 14:34:36:756268 [CLUSTER] Receive SESSION REQ from peer: broker3, client_id:subscriber_test has found in local db

2017-12-25 14:34:36:756411 [CLUSTER] Client subscriber_test has been connected to remote peer, closing old connection.

2017-12-25 14:34:36:756800 [CLUSTER] sending MULTI UNSUBSCRIBE to node: node3 (nrTopics:1,Mid: 2)

2017-12-25 14:34:36:756923 [CLUSTER] sending MULTI UNSUBSCRIBE to node: node4 (nrTopics:1,Mid: 2)

broker3.log

2017-12-25 14:34:25:194046 [CLUSTER] Receive SESSION REQ from peer: broker2, client_id: not found in local db.

2017-12-25 14:34:36:751671 [CLUSTER] Sending session request to node: node2 (client_id:subscriber_test)

2017-12-25 14:34:36:751861 [CLUSTER] Sending session request to node: node4 (client_id:subscriber_test)

2017-12-25 14:34:36:752760 Received SUBSCRIBE from subscriber_test(sock:11)

2017-12-25 14:34:36:752808 /test/cleansession2 (QoS 0)

2017-12-25 14:34:36:752836 [CLUSTER] Sending private subscribe to node:node2 (sub_client_id:subscriber_test subid:1 topic:/test/cleansession2 qos:0 mid:1)

2017-12-25 14:34:36:752854 [CLUSTER] Sending private subscribe to node:node4 (sub_client_id:subscriber_test subid:1 topic:/test/cleansession2 qos:0 mid:1)

2017-12-25 14:34:36:755907 [CLUSTER] Received UNSUBSCRIBE from peer: broker2

2017-12-25 14:34:36:755995 /test/cleansession1

While the twice connection and subscription was made by clean_session=false on broker2 and broker3 successively, the client would receive all the subscribed topic “/test/cleansession1” and “/test/cleansession2” as expected.

root@broker2:~# date && mosquitto_sub -t /test/cleansession1 -c -i subscriber_test

Mon Dec 25 11:52:42 CST 2017

root@broker3:~# date && mosquitto_sub -t /test/cleansession2 -c -i subscriber_test

Mon Dec 25 11:52:58 CST 2017

cleansession1

cleansession2

root@broker2:~# date && mosquitto_pub -t /test/cleansession1 -m "cleansession1"

Mon Dec 25 11:53:20 CST 2017

root@broker2:~# date && mosquitto_pub -t /test/cleansession2 -m "cleansession2"

Mon Dec 25 11:53:30 CST 2017

broker3.log

2017-12-25 11:52:58:891073 [CLUSTER] Sending session request to node: node2 (client_id:subscriber_test)

2017-12-25 11:52:58:891176 [CLUSTER] Sending session request to node: node4 (client_id:subscriber_test)

2017-12-25 11:52:58:891697 Received SUBSCRIBE from subscriber_test(sock:11)

2017-12-25 11:52:58:891731 /test/cleansession2 (QoS 0)

2017-12-25 11:52:58:891754 [CLUSTER] Sending private subscribe to node:node2 (sub_client_id:subscriber_test subid:2 topic:/test/cleansession2 qos:0 mid:3)

2017-12-25 11:52:58:891774 [CLUSTER] Sending private subscribe to node:node4 (sub_client_id:subscriber_test subid:2 topic:/test/cleansession2 qos:0 mid:3)

2017-12-25 11:52:58:895250 [CLUSTER] Receive SESSION RESP from node: node2 for client: subscriber_test

2017-12-25 11:52:58:895313 SESSION RESP SUBs Total 1, (1): topic:/test/cleansession1 qos:0(in)

2017-12-25 11:52:58:895329 [CLUSTER] Sending private subscribe to node:node2 (sub_client_id:subscriber_test subid:3 topic:/test/cleansession1 qos:0 mid:4)

2017-12-25 11:52:58:895347 [CLUSTER] Sending private subscribe to node:node4 (sub_client_id:subscriber_test subid:3 topic:/test/cleansession1 qos:0 mid:4)

5. Cluster Initialization Test and Node Fault Test

Startup broker2, broker3 and broker4 successively, the cluster setup successful in several seconds as expected.

broker2.log

2017-12-25 14:48:53:96185 mosquitto version 1.4.90 (build date 2017-12-25 10:11:47+0800) starting

2017-12-25 14:48:53:97176 Config loaded from /etc/mosquitto/mosquitto.conf.

2017-12-25 14:48:53:98313 Opening ipv4 listen socket on port 1883.

2017-12-25 14:48:53:99028 Opening ipv6 listen socket on port 1883.

2017-12-25 14:48:53:169403 [CLUSTER] New node: node3 context created.

2017-12-25 14:48:53:170386 [CLUSTER] New node: node4 context created.

2017-12-25 14:48:53:171501 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 14:48:53:172243 [HANDSHAKE] Current cannot handshake with node: node4. reason:Operation now in progress.

2017-12-25 14:48:55:176634 [HANDSHAKE] node: node3 service maybe down, will reconnect later after 2 seconds..

2017-12-25 14:48:55:177501 [HANDSHAKE] node: node4 service maybe down, will reconnect later after 2 seconds..

2017-12-25 14:48:55:530288 New connection from 192.168.52.103 on port 1883, new_context->fd:7,addr:0xa6c330

2017-12-25 14:48:56:533564 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 14:48:56:536123 [HANDSHAKE] Current cannot handshake with node: node4. reason:Operation now in progress.

2017-12-25 14:48:56:900118 New connection from 192.168.52.104 on port 1883, new_context->fd:10,addr:0xa6c780

2017-12-25 14:48:57:906866 New client connected(sockfd=7) from 192.168.52.103 as broker3 (c1, k10).addr:0xa6c330

2017-12-25 14:48:57:907050 Sending CONNACK to broker3 (0, 0)

2017-12-25 14:48:57:909944 Received PINGREQ from broker3

2017-12-25 14:48:57:910638 Sending PINGRESP to broker3

2017-12-25 14:48:58:916047 [HANDSHAKE] Finally handshake with node: node3 success.

2017-12-25 14:48:58:916273 [CLUSTER] Sending CONNECT to node: node3 addr(192.168.52.103:1883)

2017-12-25 14:48:58:916587 [HANDSHAKE] node: node4 service maybe down, will reconnect later after 4 seconds..

2017-12-25 14:48:58:918129 [CLUSTER] Receive CONNECT from peer:(null), node:node4 current disconnected, trigger CONNECT immediately.

2017-12-25 14:48:58:919431 New client connected(sockfd=10) from 192.168.52.104 as broker4 (c1, k10).addr:0xa6c780

2017-12-25 14:48:58:919937 Sending CONNACK to broker4 (0, 0)

2017-12-25 14:48:58:920753 [CLUSTER] Received CONNACK from node: node3.

2017-12-25 14:48:58:921291 Sending PINGREQ to node3

2017-12-25 14:48:58:922383 [HANDSHAKE] Current cannot handshake with node: node4. reason:Operation now in progress.

2017-12-25 14:48:58:923290 Received PINGREQ from broker4

2017-12-25 14:48:58:923354 Sending PINGRESP to broker4

2017-12-25 14:48:58:925338 Received PINGRESP from node3

2017-12-25 14:49:0:927785 [HANDSHAKE] Finally handshake with node: node4 success.

2017-12-25 14:49:0:928009 [CLUSTER] Sending CONNECT to node: node4 addr(192.168.52.104:1883)

2017-12-25 14:49:0:930713 [CLUSTER] Received CONNACK from node: node4.

broker3.log

2017-12-25 14:48:55:477161 mosquitto version 1.4.90 (build date 2017-12-25 10:11:47+0800) starting

2017-12-25 14:48:55:478182 Config loaded from /etc/mosquitto/mosquitto.conf.

2017-12-25 14:48:55:479312 Opening ipv4 listen socket on port 1883.

2017-12-25 14:48:55:480149 Opening ipv6 listen socket on port 1883.

2017-12-25 14:48:55:522757 [CLUSTER] New node: node2 context created.

2017-12-25 14:48:55:523764 [CLUSTER] New node: node4 context created.

2017-12-25 14:48:55:524519 [HANDSHAKE] Current cannot handshake with node: node2. reason:Operation now in progress.

2017-12-25 14:48:55:525562 [HANDSHAKE] Current cannot handshake with node: node4. reason:Operation now in progress.

2017-12-25 14:48:56:530240 New connection from 192.168.52.102 on port 1883, new_context->fd:9,addr:0x88f330

2017-12-25 14:48:56:896570 New connection from 192.168.52.104 on port 1883, new_context->fd:10,addr:0x88f780

2017-12-25 14:48:57:898890 [HANDSHAKE] Finally handshake with node: node2 success.

2017-12-25 14:48:57:899843 [CLUSTER] Sending CONNECT to node: node2 addr(192.168.52.102:1883)

2017-12-25 14:48:57:900737 [HANDSHAKE] node: node4 service maybe down, will reconnect later after 2 seconds..

2017-12-25 14:48:57:902769 [CLUSTER] Received CONNACK from node: node2.

2017-12-25 14:48:57:904107 Sending PINGREQ to node2

2017-12-25 14:48:57:907258 Received PINGRESP from node2

2017-12-25 14:48:58:902419 [CLUSTER] Receive CONNECT from peer:(null), node:node4 current disconnected, trigger CONNECT immediately.

2017-12-25 14:48:58:902616 New client connected(sockfd=10) from 192.168.52.104 as broker4 (c1, k10).addr:0x88f780

2017-12-25 14:48:58:903196 Sending CONNACK to broker4 (0, 0)

2017-12-25 14:48:58:903644 [HANDSHAKE] Current cannot handshake with node: node4. reason:Operation now in progress.

2017-12-25 14:48:58:906206 Received PINGREQ from broker4

2017-12-25 14:48:58:906307 Sending PINGRESP to broker4

2017-12-25 14:48:58:912058 New client connected(sockfd=9) from 192.168.52.102 as broker2 (c1, k10).addr:0x88f330

2017-12-25 14:48:58:912174 Sending CONNACK to broker2 (0, 0)

2017-12-25 14:48:58:920062 Received PINGREQ from broker2

2017-12-25 14:48:58:920167 Sending PINGRESP to broker2

2017-12-25 14:49:0:927748 [HANDSHAKE] Finally handshake with node: node4 success.

2017-12-25 14:49:0:930414 [CLUSTER] Sending CONNECT to node: node4 addr(192.168.52.104:1883)

2017-12-25 14:49:0:937246 [CLUSTER] Received CONNACK from node: node4.

broker4.log

2017-12-25 14:48:56:843987 mosquitto version 1.4.90 (build date 2017-12-25 10:11:47+0800) starting

2017-12-25 14:48:56:845697 Config loaded from /etc/mosquitto/mosquitto.conf.

2017-12-25 14:48:56:875331 Opening ipv4 listen socket on port 1883.

2017-12-25 14:48:56:875907 Opening ipv6 listen socket on port 1883.

2017-12-25 14:48:56:908722 [CLUSTER] New node: node2 context created.

2017-12-25 14:48:56:909450 [CLUSTER] New node: node3 context created.

2017-12-25 14:48:56:909888 [HANDSHAKE] Current cannot handshake with node: node2. reason:Operation now in progress.

2017-12-25 14:48:56:910383 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 14:48:58:914340 [HANDSHAKE] Finally handshake with node: node2 success.

2017-12-25 14:48:58:915296 [CLUSTER] Sending CONNECT to node: node2 addr(192.168.52.102:1883)

2017-12-25 14:48:58:916102 [HANDSHAKE] Finally handshake with node: node3 success.

2017-12-25 14:48:58:916576 [CLUSTER] Sending CONNECT to node: node3 addr(192.168.52.103:1883)

2017-12-25 14:48:58:918991 [CLUSTER] Received CONNACK from node: node3.

2017-12-25 14:48:58:919933 Sending PINGREQ to node3

2017-12-25 14:48:58:920624 New connection from 192.168.52.103 on port 1883, new_context->fd:9,addr:0xdcb070

2017-12-25 14:48:58:921974 Received PINGRESP from node3

2017-12-25 14:48:58:931664 [CLUSTER] Received CONNACK from node: node2.

2017-12-25 14:48:58:932276 Sending PINGREQ to node2

2017-12-25 14:48:58:933459 New connection from 192.168.52.102 on port 1883, new_context->fd:10,addr:0xdcb2b0

2017-12-25 14:48:58:934248 Received PINGRESP from node2

2017-12-25 14:49:0:940136 New client connected(sockfd=10) from 192.168.52.102 as broker2 (c1, k10).addr:0xdcb2b0

2017-12-25 14:49:0:940866 Sending CONNACK to broker2 (0, 0)

2017-12-25 14:49:0:942932 Received PINGREQ from broker2

2017-12-25 14:49:0:943034 Sending PINGRESP to broker2

2017-12-25 14:49:0:950075 New client connected(sockfd=9) from 192.168.52.103 as broker3 (c1, k10).addr:0xdcb070

2017-12-25 14:49:0:951225 Sending CONNACK to broker3 (0, 0)

In order to simulate node fault, kill the mosquitto process on host “broker3”, broker2 and broker4 would try to re-connect with broker3 2 minutes later as expected.

mosquitto@broker3:~$ kill -9 2576

mosquitto@broker3:~$

[1]+ Killed   mosquitto -c /etc/mosquitto/mosquitto.conf

broker2.log

2017-12-25 14:55:20:804242 [CLUSTER] node: node3 down, do_disconnect now.

2017-12-25 14:55:20:804345 Socket error on client broker3, disconnecting.

2017-12-25 14:55:20:804440 context__cleanup,client_id:(null).addr:0xa6c330,do_free:true

…after 120 seconds

2017-12-25 14:57:20:586016 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 14:57:22:592257 [HANDSHAKE] node: node3 service maybe down, will reconnect later after 2 seconds..

2017-12-25 14:57:24:596134 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 14:57:26:600531 [HANDSHAKE] node: node3 service maybe down, will reconnect later after 4 seconds..

2017-12-25 14:57:30:608518 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 14:57:32:616099 [HANDSHAKE] node: node3 service maybe down, will reconnect later after 8 seconds..

2017-12-25 14:57:40:632875 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 14:57:42:638822 [HANDSHAKE] node: node3 service maybe down, will reconnect later after 16 seconds..

2017-12-25 14:57:58:673280 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 14:58:0:677161 [HANDSHAKE] node: node3 service maybe down, will reconnect later after 20 seconds..

2017-12-25 14:58:20:720368 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 14:58:22:726010 [HANDSHAKE] node: node3 service maybe down, will reconnect later after 20 seconds..

broker4.log and broker2.log are the same as above.

Anytime the broker3 startup again, broker2 and broker4 would try to reconnect with broker3 immediately as expected.

2017-12-25 15:1:14:802456 New connection from 192.168.52.103 on port 1883, new_context->fd:8,addr:0xdc0870

2017-12-25 15:1:16:811726 [CLUSTER] Receive CONNECT from peer:(null), node:node3 current disconnected, trigger CONNECT immediately.

2017-12-25 15:1:16:811904 New client connected(sockfd=8) from 192.168.52.103 as broker3 (c1, k10).addr:0xdc0870

2017-12-25 15:1:16:811962 Sending CONNACK to broker3 (0, 0)

2017-12-25 15:1:16:812381 [HANDSHAKE] Current cannot handshake with node: node3. reason:Operation now in progress.

2017-12-25 15:1:18:823555 [HANDSHAKE] Finally handshake with node: node3 success.

2017-12-25 15:1:18:823661 [CLUSTER] Sending CONNECT to node: node3 addr(192.168.52.103:1883)

2017-12-25 15:1:18:829683 [CLUSTER] Received CONNACK from node: node3.

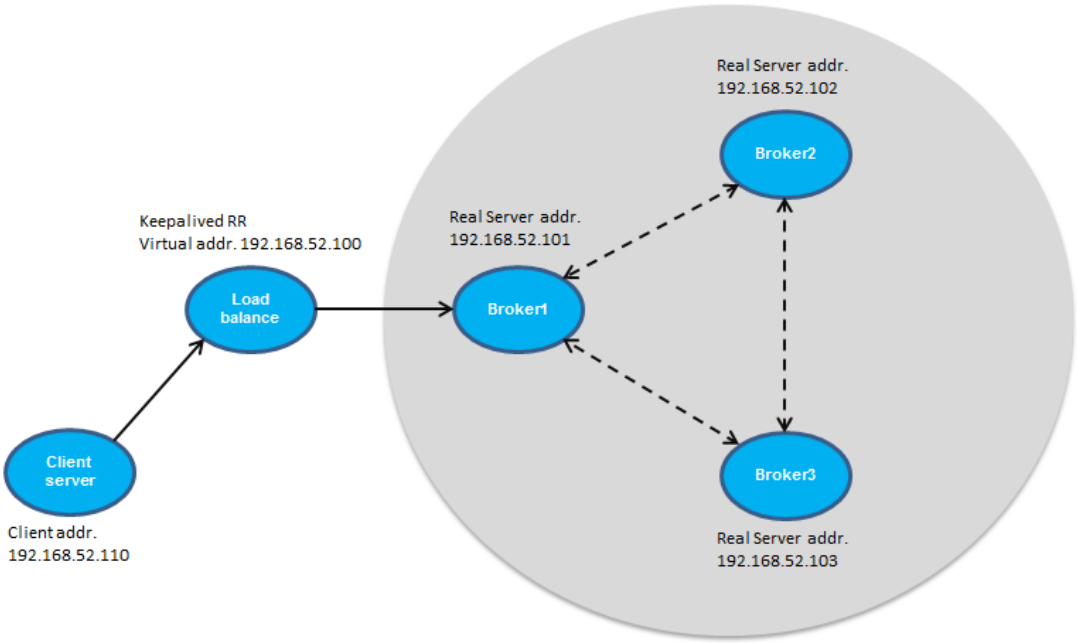
Reboot the OS for broker2:

The cluster also can be recovered as expected.

6. Will Message Test

Performance Test(using Tsung and Keepalived)

Network topology:



Kernel parameters:

root@broker2:~# sysctl -p

net.ipv4.ip_local_port_range = 1024 60000

net.core.rmem_max = 16777216

net.core.wmem_max = 16777216

net.ipv4.tcp_rmem = 4096 87380 16777216

net.ipv4.tcp_wmem = 4096 65536 16777216

net.ipv4.tcp_fin_timeout = 10

net.ipv4.tcp_tw_recycle = 1

net.ipv4.tcp_timestamps = 0

net.ipv4.tcp_window_scaling = 0

net.ipv4.tcp_sack = 0

net.core.netdev_max_backlog = 30000

net.ipv4.tcp_no_metrics_save = 1

net.core.somaxconn = 262144

net.ipv4.tcp_syncookies = 0

net.ipv4.tcp_max_orphans = 262144

net.ipv4.tcp_max_syn_backlog = 262144

net.ipv4.tcp_synack_retries = 2

net.ipv4.tcp_syn_retries = 2

root@broker2:~# ulimit -a

core file size (blocks, -c) unlimited

data seg size (kbytes, -d) unlimited

scheduling priority (-e) 0

file size (blocks, -f) unlimited

pending signals (-i) 3712

```
max locked memory           (kbytes, -l) 64
max memory size             (kbytes, -m) unlimited
open files                          (-n) 80000
pipe size                (512 bytes, -p) 8
POSIX message queues        (bytes, -q) 819200
real-time priority                  (-r) 0
stack size                  (kbytes, -s) 8192
cpu time                    (seconds, -t) unlimited
max user processes                  (-u) 3712
virtual memory              (kbytes, -v) unlimited
file locks                          (-x) unlimited
```

Scenario 1

Simulate the smart home of IOT, 2000 houses which contain 2000 mobile phones(send device control instructions via PUBLISH messages) and 2000 household electrical appliances(receive device control instructions via PUBLISH messages). Each mobile phone produces a PUBLISH message with a 2 minutes cycle.

First arrivalphase, 2000 electrical appliances make the subscription within 20 seconds period, second arrivalphase, 50 mobile phones make a PUBLISH per second.

The Tsung configure as below,

```
    <load>
        <arrivalphase phase="1" duration="20" unit="second">
            <session_setup name="mqtt_subscriber" probability="100"/>
            <users maxnumber="2000" arrivalrate="200" unit="second"/>
        </arrivalphase>
        <arrivalphase phase="2" duration="1800" unit="second">
            <session_setup name="mqtt_publisher" probability="100"/>
            <users maxnumber="36000" arrivalrate="20" unit="second"/>
        </arrivalphase>
    </load>

    <sessions>
        <session name="mqtt_publisher" probability="0" type="ts_mqtt">
            <setdynvars sourcetype="random_number" start="1001" end="3000">
            <var name="pubtopic"/>
            </setdynvars>
            <request>
                <mqtt type="connect" clean_start="true" keepalive="60" will_topic="will_topic" will_qos="0"
will_msg="will_msg" will_retain="false"></mqtt>
            </request>
            <request subst="true">
                <mqtt type="publish" topic="%%_pubtopic%%" qos="1" retained="false">test_message</mqtt>
```

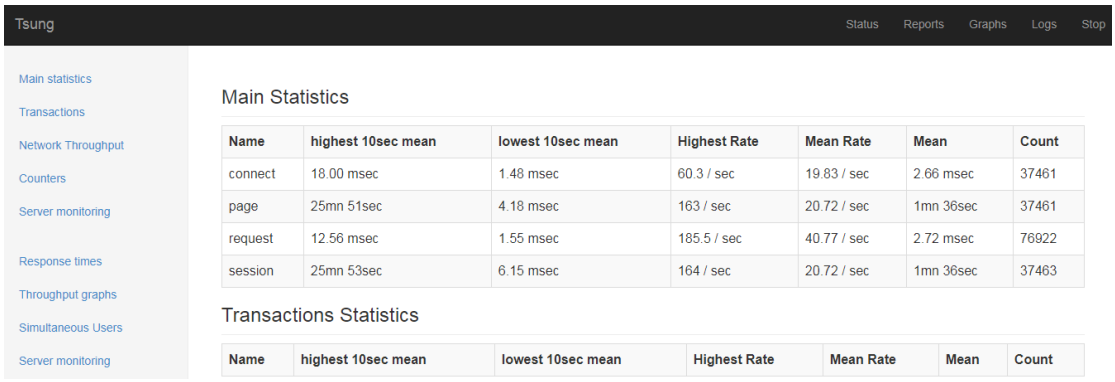
```
        </request>
        <request>
            <mqtt type="disconnect"></mqtt>
        </request>
    </session>

    <session name="mqtt_subscriber" probability="0" type="ts_mqtt">
            <setdynvars sourcetype="random_number" start="1001" end="3000">
            <var name="subtopic"/>
        </setdynvars>
        <request>
            <mqtt type="connect" clean_start="true" keepalive="60"></mqtt>
        </request>
        <request subst="true">
            <mqtt type="subscribe" topic="%%_subtopic%%" qos="1"></mqtt>
        </request>
        <request>
            <!-- wait for 60s -->
            <mqtt type="waitForMessages" timeout="1800"></mqtt>
        </request>
        <request subst="true">
            <mqtt type="unsubscribe" topic="%%_subtopic%%"></mqtt>
        </request>
        <request>
            <mqtt type="disconnect"></mqtt>
        </request>
    </session>
</sessions>
```

loadbalance(with Keepalived)：

```
root@broker1:~# ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.52.100:1883 rr
  -> broker2:1883                 Route   1      667        0
  -> broker3:1883                 Route   1      667        0
  -> broker4:1883                 Route   1      666        0
```

Test report



Tsung

Status Reports Graphs Logs Stop

Main statistics
Transactions
Network Throughput
Counters
Server monitoring

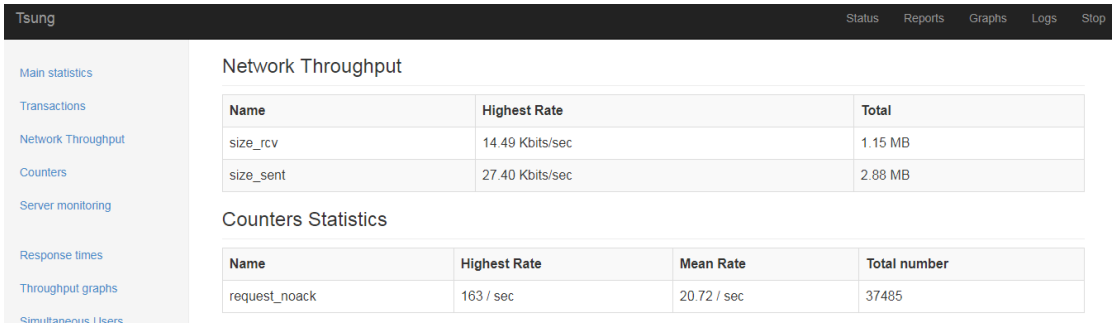
Response times
Throughput graphs
Simultaneous Users
Server monitoring

Main Statistics

Name	highest 10sec mean	lowest 10sec mean	Highest Rate	Mean Rate	Mean	Count
connect	18.00 msec	1.48 msec	60.3 / sec	19.83 / sec	2.66 msec	37461
page	25mn 51sec	4.18 msec	163 / sec	20.72 / sec	1mn 36sec	37461
request	12.56 msec	1.55 msec	185.5 / sec	40.77 / sec	2.72 msec	76922
session	25mn 53sec	6.15 msec	164 / sec	20.72 / sec	1mn 36sec	37463

Transactions Statistics

Name	highest 10sec mean	lowest 10sec mean	Highest Rate	Mean Rate	Mean	Count



Tsung

Status Reports Graphs Logs Stop

Main statistics
Transactions
Network Throughput
Counters
Server monitoring

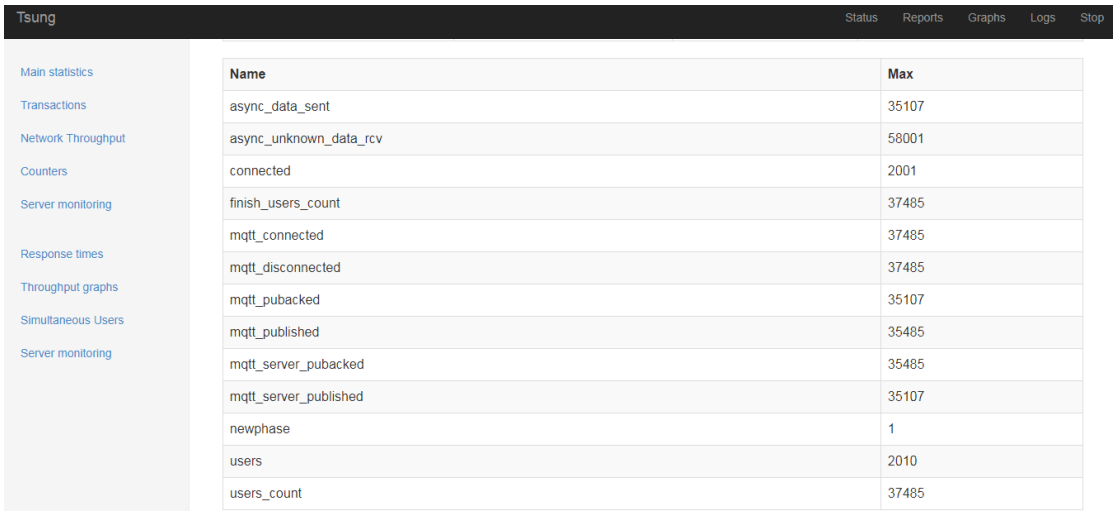
Response times
Throughput graphs
Simultaneous Users

Network Throughput

Name	Highest Rate	Total
size_rcv	14.49 Kbits/sec	1.15 MB
size_sent	27.40 Kbits/sec	2.88 MB

Counters Statistics

Name	Highest Rate	Mean Rate	Total number
request_noack	163 / sec	20.72 / sec	37485



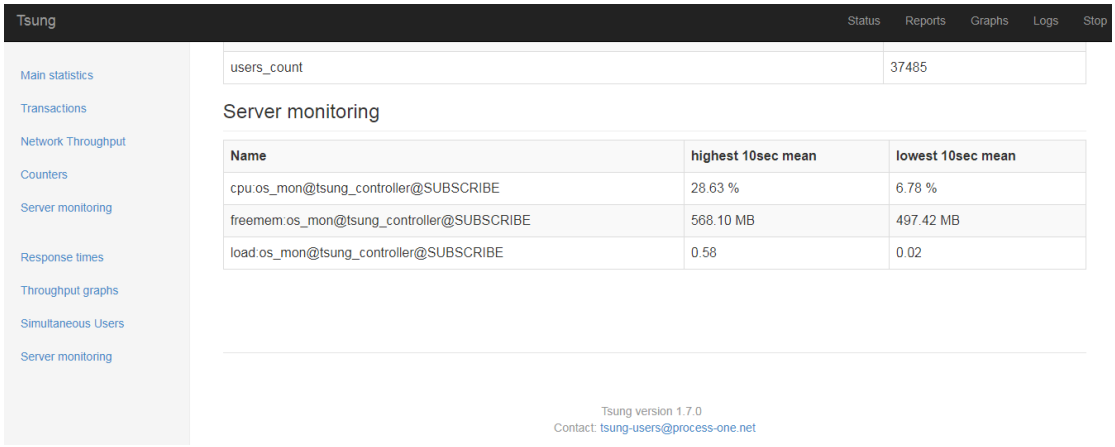
Tsung

Status Reports Graphs Logs Stop

Main statistics
Transactions
Network Throughput
Counters
Server monitoring

Response times
Throughput graphs
Simultaneous Users
Server monitoring

Name	Max
async_data_sent	35107
async_unknown_data_rcv	58001
connected	2001
finish_users_count	37485
mqtt_connected	37485
mqtt_disconnected	37485
mqtt_pubacked	35107
mqtt_published	35485
mqtt_server_pubacked	35485
mqtt_server_published	35107
newphase	1
users	2010
users_count	37485



Tsung

Status Reports Graphs Logs Stop

Main statistics
Transactions
Network Throughput
Counters
Server monitoring

Response times
Throughput graphs
Simultaneous Users
Server monitoring

users_count	37485

Server monitoring

Name	highest 10sec mean	lowest 10sec mean
cpu:os_mon@tsung_controller@SUBSCRIBE	28.63 %	6.78 %
freemem:os_mon@tsung_controller@SUBSCRIBE	568.10 MB	497.42 MB
load:os_mon@tsung_controller@SUBSCRIBE	0.58	0.02

Tsung version 1.7.0
Contact: tsung-users@process-one.net

Scenario 2

To be test.