



# 채팅 STOMP 부하테스트

🕒 생성일	@2024년 2월 19일 오후 9:38
🏷 태그	

개요

[Test Plan](#)

[Results Tree](#)

[Results](#)

[thread 갯수\(유저\): 1](#)

[thread 갯수\(유저\): 100](#)

[thread 갯수\(유저\): 1000](#)

[Connection 유지하는 시나리오](#)

[thread 갯수\(유저\): 1](#)

[thread 갯수\(유저\): 100](#)

[thread 갯수\(유저\): 500](#)

[thread 갯수\(유저\): 1000](#)

[thread 갯수\(유저\): 3000](#)

[결론](#)

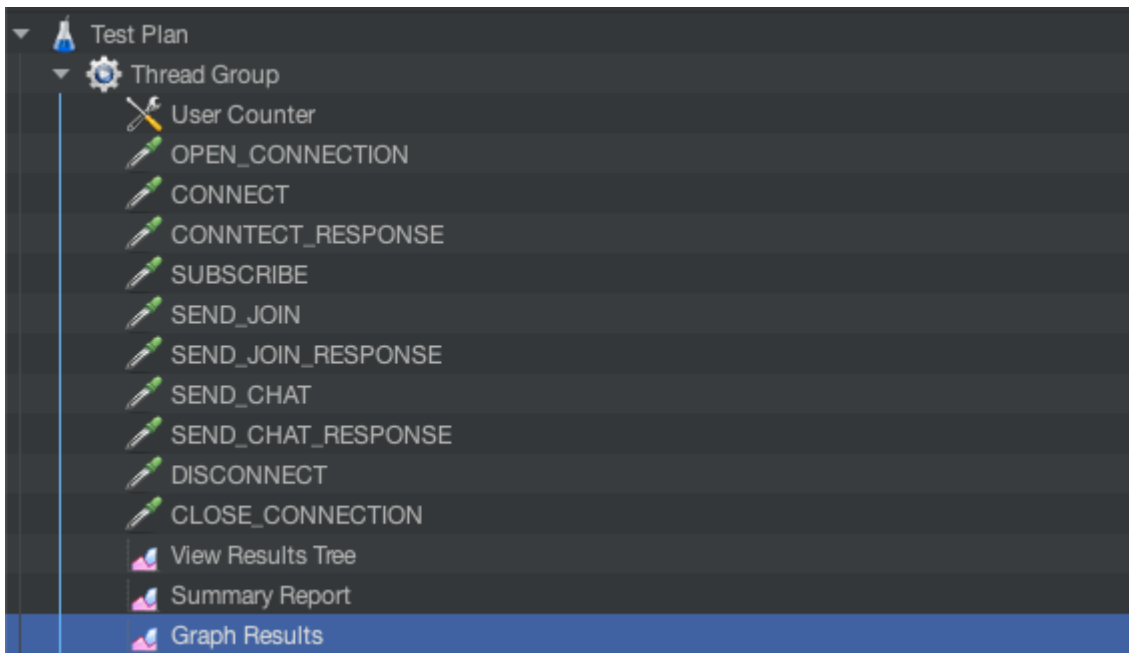
## 개요

JMeter + STOMP Plugin 사용한 채팅 서버 성능테스트입니다.

~~컴퓨터 성능의 문제인지 1000명대 테스트부터는 테스트 중 인터넷 접속이 끊겨서 매번 도커 up, down, volume 날리기, 재시동 등 작업을 반복해 매우 귀찮았습니다..~~ → 쓰레드 수 총합만 맞추면 되는데 제가 오버딩킹해서 1000\*1000 의 부하를 주고 있었네요

케이스 당 5번씩 진행하였습니다.

## Test Plan



**Counter**

Name:

Comments:

Starting value

Increment

Maximum value

Number format

Exported Variable Name

☐ Track counter independently for each user

☐ Reset counter on each Thread Group Iteration

- Summary Report를 통해 전체 쓰레드(유저)가 수행한 결과의 평균치를 report 받습니다.
- Counter manager를 통해 `${변수명}` 식으로 변수 사용 가능합니다. 제 경우에는 쓰레드당 접속하는 유저 데이터로써 사용했습니다.

## Results Tree

```
totalPlaylistTime: playlist=Playlist(id=qwer1234, name=플레이리스트1, playlistOwnerId=1, playlistOwnerNickName=유지닉네임1, playlistOwnerProfileImage=https://currentPlaylistTimeInSeconds=176
music.getId()=1
ChatMessageDto(id=65d358f4f7d84d462e6cbcf, messageType=ENTRANCE, roomId=65d34cfef7d84d462e6cbc22, senderId=9, nickName=유지닉네임9, senderProfileImage=url
채팅방 id: 65d34cfef7d84d462e6cbc22, 발신자 id: 9, 메시지: 유지닉네임9님이 입장하셨습니다.
메시지 저장 성공 message=ChatMessage(id=65d358f4f7d84d462e6cbcf0, messageType=MSG6, roomId=65d34cfef7d84d462e6cbc22, senderId=9, nickName=유지닉네임9, senderPro
메시지 전송 완료 - message=ChatMessageDto(id=null, messageType=MSG6, roomId=65d34cfef7d84d462e6cbc22, senderId=9, nickName=유지닉네임9, senderProfileImage=url9.
채팅방 id: 65d34cfef7d84d462e6cbc22, 발신자 id: 9, 메시지: hi9i
메시지 저장 성공 message=ChatMessage(id=65d358f4f7d84d462e6cbcf2, messageType=MSG6, roomId=65d34cfef7d84d462e6cbc22, senderId=10, nickName=유지닉네임10, senderF
메시지 전송 완료 - message=ChatMessageDto(id=null, messageType=MSG6, roomId=65d34cfef7d84d462e6cbc22, senderId=10, nickName=유지닉네임10, senderProfileImage=url
채팅방 id: 65d34cfef7d84d462e6cbc22, 발신자 id: 10, 메시지: hi10i
room=Room(id=65d34cfef7d84d462e6cbc22, roomName=플레이리스트1, playlist=Playlist(id=qwer1234, name=플레이리스트1, playlistOwnerId=1, playlistOwnerNickName=유지닉
elapsedTime=PT16M53.908299S
totalPPlaylistTime: playlist=Playlist(id=qwer1234, name=플레이리스트1, playlistOwnerId=1, playlistOwnerNickName=유지닉네임1, playlistOwnerProfileImage=https://
currentPlaylistTimeInSeconds=176
music.getId()=1
ChatMessageDto(id=65d358f4f7d84d462e6cbcf1, messageType=ENTRANCE, roomId=65d34cfef7d84d462e6cbc22, senderId=10, nickName=유지닉네임10, senderProfileImage=u
채팅방 id: 65d34cfef7d84d462e6cbc22, 발신자 id: 10, 메시지: 유지닉네임10님이 입장하셨습니다.
메시지 저장 성공 message=ChatMessage(id=65d358f4f7d84d462e6cbcf3, messageType=MSG6, roomId=65d34cfef7d84d462e6cbc22, senderId=11, nickName=유지닉네임11, senderF
메시지 전송 완료 - message=ChatMessageDto(id=null, messageType=MSG6, roomId=65d34cfef7d84d462e6cbc22, senderId=11, nickName=유지닉네임11, senderProfileImage=url
채팅방 id: 65d34cfef7d84d462e6cbc22, 발신자 id: 11, 메시지: hi11i
room=Room(id=65d34cfef7d84d462e6cbc22, roomName=플레이리스트1, playlist=Playlist(id=qwer1234, name=플레이리스트1, playlistOwnerId=1, playlistOwnerNickName=유지닉
elapsedTime=PT16M53.908637S
totalPlaylistTime: playlist=Playlist(id=qwer1234, name=플레이리스트1, playlistOwnerId=1, playlistOwnerNickName=유지닉네임1, playlistOwnerProfileImage=https://
currentPlaylistTimeInSeconds=176
music.getId()=1
ChatMessageDto(id=65d358f4f7d84d462e6cbcf4, messageType=ENTRANCE, roomId=65d34cfef7d84d462e6cbc22, senderId=11, nickName=유지닉네임11, senderProfileImage=u
채팅방 id: 65d34cfef7d84d462e6cbc22, 발신자 id: 11, 메시지: 유지닉네임11님이 입장하셨습니다.
room=Room(id=65d34cfef7d84d462e6cbc22, roomName=플레이리스트1, playlist=Playlist(id=qwer1234, name=플레이리스트1, playlistOwnerId=1, playlistOwnerNickName=유지닉
elapsedTime=PT16M53.917639S
totalPlaylistTime: playlist=Playlist(id=qwer1234, name=플레이리스트1, playlistOwnerId=1, playlistOwnerNickName=유지닉네임1, playlistOwnerProfileImage=https://
currentPlaylistTimeInSeconds=176
music.getId()=1
ChatMessageDto(id=65d358f4f7d84d462e6cbcf5, messageType=ENTRANCE, roomId=65d34cfef7d84d462e6cbc22, senderId=12, nickName=유지닉네임12, senderProfileImage=u
채팅방 id: 65d34cfef7d84d462e6cbc22, 발신자 id: 12, 메시지: 유지닉네임12님이 입장하셨습니다.
메시지 저장 성공 message=ChatMessage(id=65d358f4f7d84d462e6cbcf6, messageType=MSG6, roomId=65d34cfef7d84d462e6cbc22, senderId=12, nickName=유지닉네임12, senderF
메시지 전송 완료 - message=ChatMessageDto(id=null, messageType=MSG6, roomId=65d34cfef7d84d462e6cbc22, senderId=12, nickName=유지닉네임12, senderProfileImage=url
채팅방 id: 65d34cfef7d84d462e6cbc22, 발신자 id: 12, 메시지: hi12i
```

인텔리제이 로그를 통해 STOMP 통한 입장, 메시지 전송이 아주 잘 되고 있음이 보입니다.

```
Request URL
http://localhost:18000/v1/api/rooms/65d34cfef7d84d462e6cbc22

Server response
Code      Details
200
Response body
{
  "uid": 13,
  "nickName": "유저닉네임13",
  "profileImage": "url13.url"
},
{
  "uid": 7,
  "nickName": "유저닉네임7",
  "profileImage": "url7.url"
},
{
  "uid": 8,
  "nickName": "유저닉네임8",
  "profileImage": "url8.url"
},
{
  "uid": 16,
  "nickName": "유저닉네임16",
  "profileImage": "url16.url"
},
{
  "uid": 17,
  "nickName": "유저닉네임17",
  "profileImage": "url17.url"
}
```

실제로 방 조회 API 통해서 확인해보면 비 동기적으로 유저들이 접속하고 있습니다.

```
Request URL
http://localhost:18000/v1/api/chat/history/65d34cfef7d84d462e6cbc...

Server response

Code    Details

200

Response body

{
  {
    "id": "65d350f4f7d84d462e6cbcf3",
    "messageType": "MSG",
    "roomId": "65d34cfef7d84d462e6cbc22",
    "senderId": 11,
    "nickName": "유저닉네임11",
    "senderProfileImage": "url11.url",
    "content": "hi!!i",
    "createdAt": "2024-02-19T22:00:36.626"
  },
  {
    "id": "65d350f4f7d84d462e6cbcf4",
    "messageType": "ENTRANCE",
    "roomId": "65d34cfef7d84d462e6cbc22",
    "senderId": 11,
    "nickName": "유저닉네임11",
    "senderProfileImage": "url11.url",
    "content": "유저닉네임11님이 입장하셨습니다.",
    "createdAt": "2024-02-19T22:00:36.626"
  },
  {
    "id": "65d350f4f7d84d462e6cbcf1",
    "messageType": "ENTRANCE",
    "roomId": "65d34cfef7d84d462e6cbc22"
```

메시지도 잘 쌓이고 저장되고 있습니다.

View Results Tree를 통해 테스트의 각 단계가 어떻게 수행되었는지 하나씩 살펴 볼 수 있습니다.

The screenshot shows the 'View Results Tree' window with the 'Text' tab selected. The list of steps includes: OPEN\_CONNECTION (selected), CONNECT, CONNTECT\_RESPONSE, SUBSCRIBE, SEND\_JOIN, SEND\_JOIN\_RESPONSE, SEND\_CHAT, SEND\_CHAT\_RESPONSE, DISCONNECT, and CLOSE\_CONNECTION. The right pane shows the 'Request' tab with the 'Request Body' sub-tab. The content displays the 'Connect URL:' as 'ws://localhost:18000/ws-chat'.

The screenshot shows the 'View Results Tree' window with the 'Text' tab selected. The list of steps includes: OPEN\_CONNECTION, CONNECT (selected), CONNTECT\_RESPONSE, SUBSCRIBE, SEND\_JOIN, SEND\_JOIN\_RESPONSE, SEND\_CHAT, SEND\_CHAT\_RESPONSE, DISCONNECT, and CLOSE\_CONNECTION. The right pane shows the 'Request' tab with the 'Request Body' sub-tab. The content displays the 'Connect URL:' as 'ws://localhost:18000/ws-chat (using existing connection)' and the 'Request data:' as 'CONNECT', 'accept-version:1.1,1.0', and 'heart-beat:10000,10000'.

Text

- OPEN\_CONNECTION
- CONNECT
- CONNTECT\_RESPONSE
- SUBSCRIBE
- SEND\_JOIN
- SEND\_JOIN\_RESPONSE
- SEND\_CHAT
- SEND\_CHAT\_RESPONSE
- DISCONNECT
- CLOSE\_CONNECTION

Sampler result Request Response data

Response Body Response headers

CONNECTED  
version:1.1  
heart-beat:0,0

Text

- OPEN\_CONNECTION
- CONNECT
- CONNTECT\_RESPONSE
- SUBSCRIBE
- SEND\_JOIN
- SEND\_JOIN\_RESPONSE
- SEND\_CHAT
- SEND\_CHAT\_RESPONSE
- DISCONNECT
- CLOSE\_CONNECTION

Sampler result Request Response data

Request Body Request Headers

1 Connect URL:  
2 ws://localhost:18000/ws-chat  
3 (using existing connection)  
4  
5 Request data:  
6 SUBSCRIBE  
7 id:sub-0  
8 destination:/chat/topic/room/65d34cfef7d84d462e6cbc22  
9  
10  
11  
12

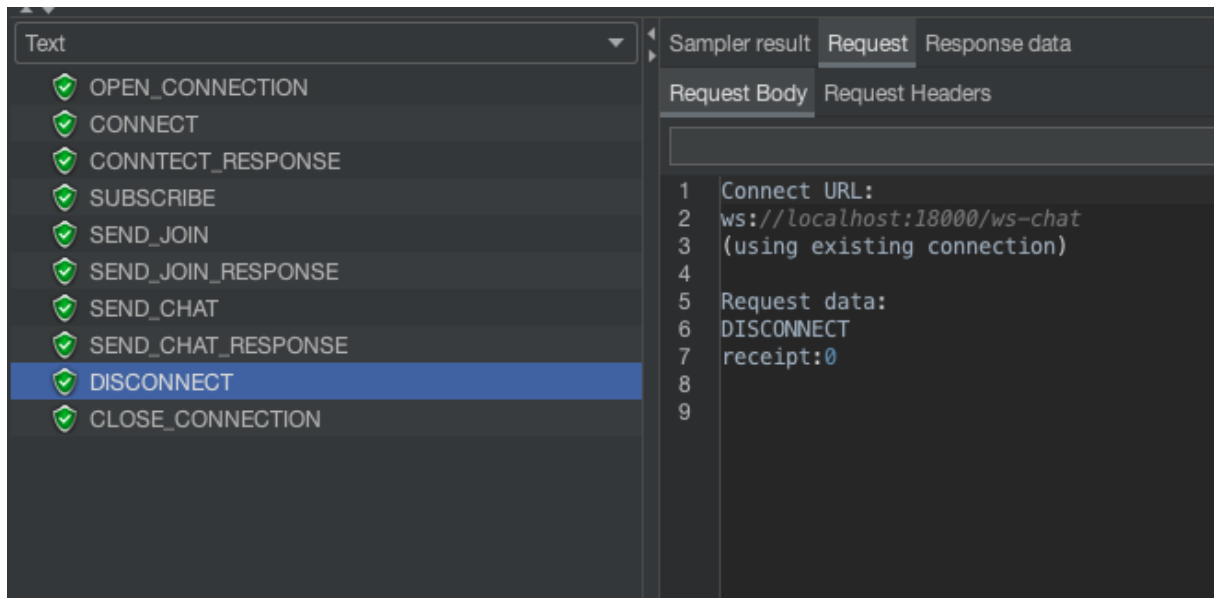
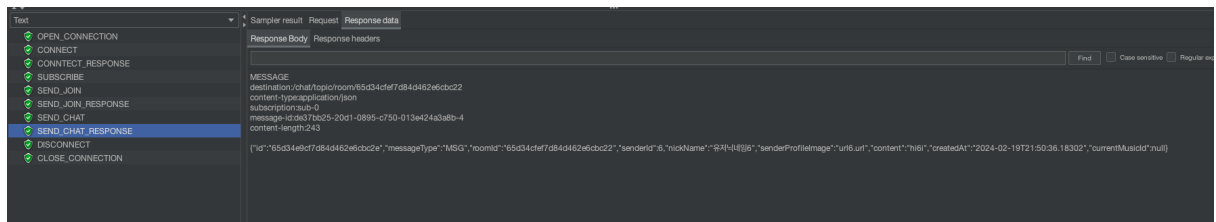
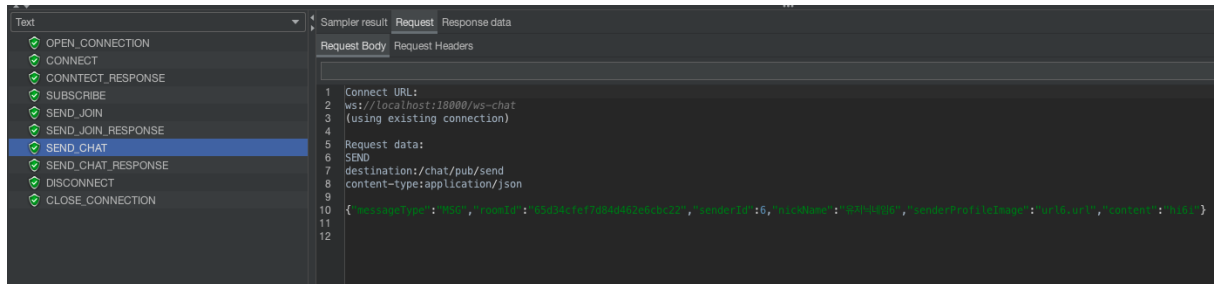
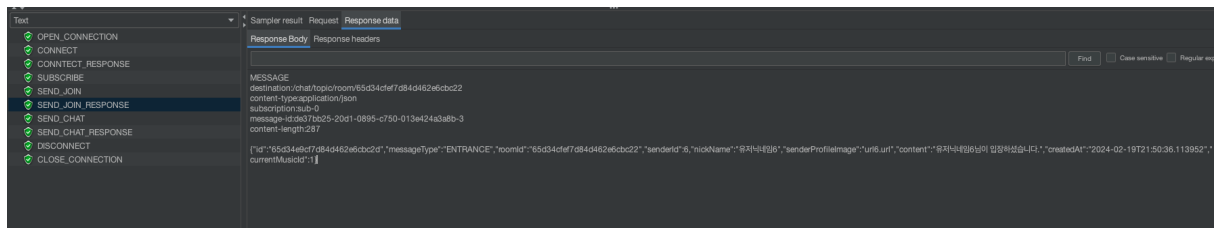
Text

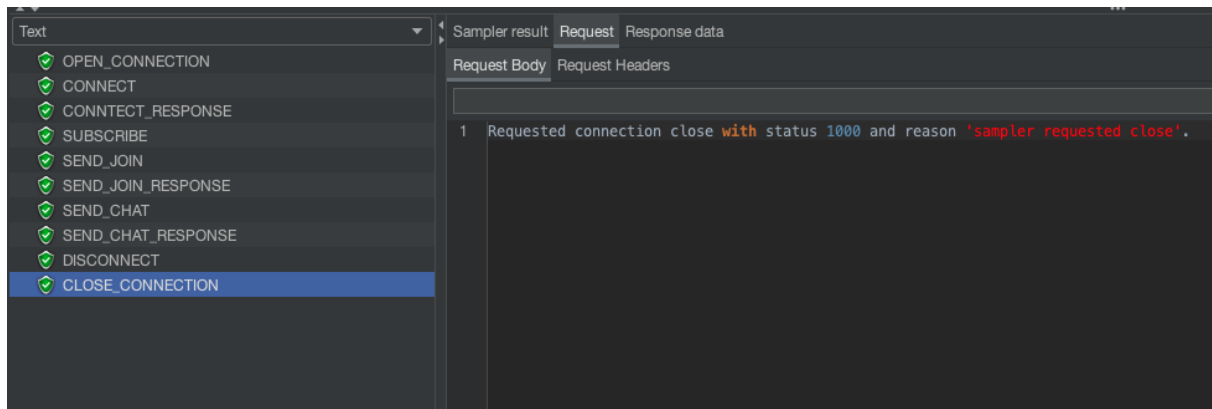
- OPEN\_CONNECTION
- CONNECT
- CONNTECT\_RESPONSE
- SUBSCRIBE
- SEND\_JOIN
- SEND\_JOIN\_RESPONSE
- SEND\_CHAT
- SEND\_CHAT\_RESPONSE
- DISCONNECT
- CLOSE\_CONNECTION

Sampler result Request Response data

Request Body Request Headers

1 Connect URL:  
2 ws://localhost:18000/ws-chat  
3 (using existing connection)  
4  
5 Request data:  
6 SEND  
7 destination:/chat/pub/join  
8 content-type:application/json  
9  
10 {"roomId":"65d34cfef7d84d462e6cbc22","senderId":6,"nickName":"부자H996","senderProfileImage":"url6.url"}  
11  
12





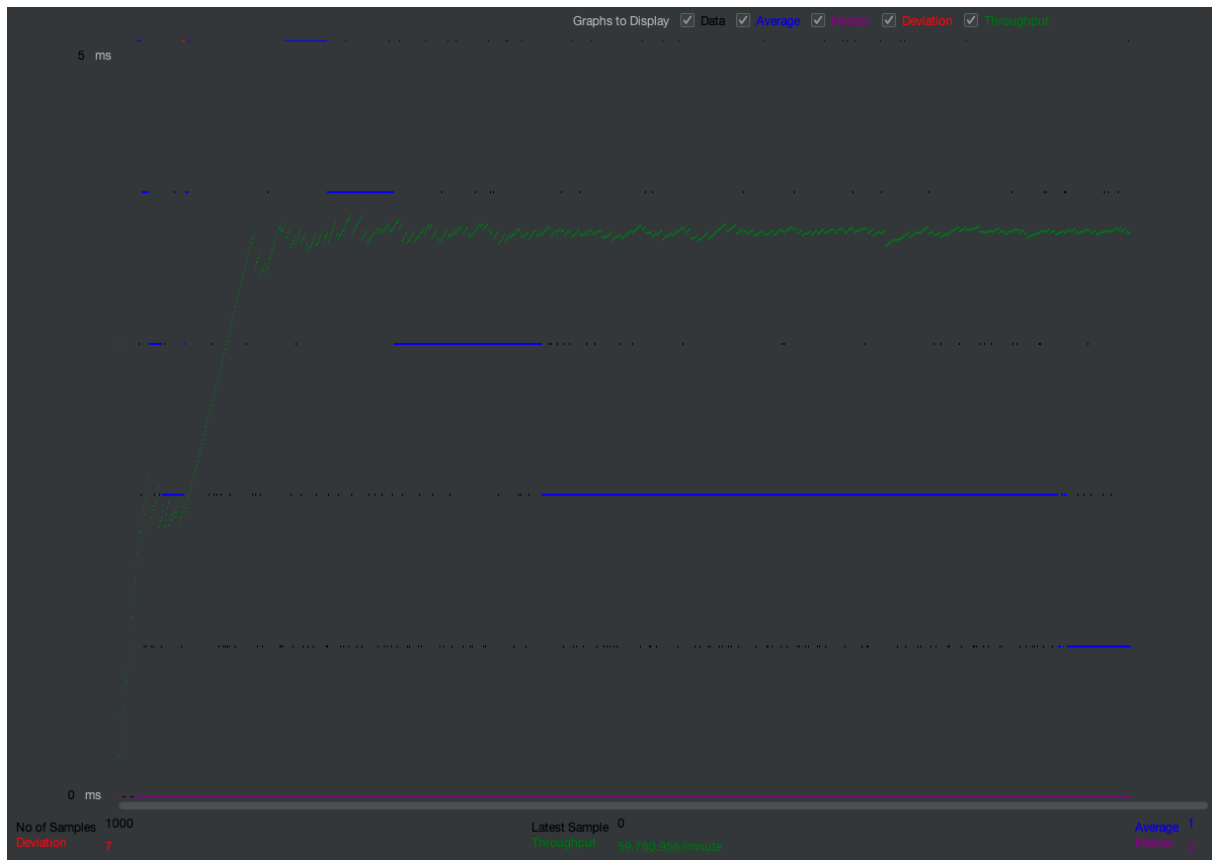
## Results

thread 갯수(유저): 1

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
OPEN_CONNECTION	1	15	15	15	0.00	0.00%	66.7/sec	9.57	10.48	147.0
CONNECT	1	0	0	0	0.00	0.00%	0/hour	0.00	0.00	.0
CONNECT_RESPONSE	1	4	4	4	0.00	0.00%	250.0/sec	10.01	0.00	41.0
SUBSCRIBE	1	0	0	0	0.00	0.00%	0/hour	0.00	0.00	.0
SEND_JOIN	1	0	0	0	0.00	0.00%	0/hour	0.00	0.00	.0
SEND_JOIN_RESPONSE	1	78	78	78	0.00	0.00%	12.8/sec	5.92	0.00	473.0
SEND_CHAT	1	0	0	0	0.00	0.00%	0/hour	0.00	0.00	.0
SEND_CHAT_RESPONSE	1	20	20	20	0.00	0.00%	50.0/sec	20.95	0.00	429.0
DISCONNECT	1	0	0	0	0.00	0.00%	0/hour	0.00	0.00	.0
CLOSE_CONNECTION	1	1	1	1	0.00	0.00%	1000.0/sec	26.37	30.27	27.0
TOTAL	10	11	0	78	23.09	0.00%	84.0/sec	9.17	6.34	111.7

thread 갯수(유저): 100

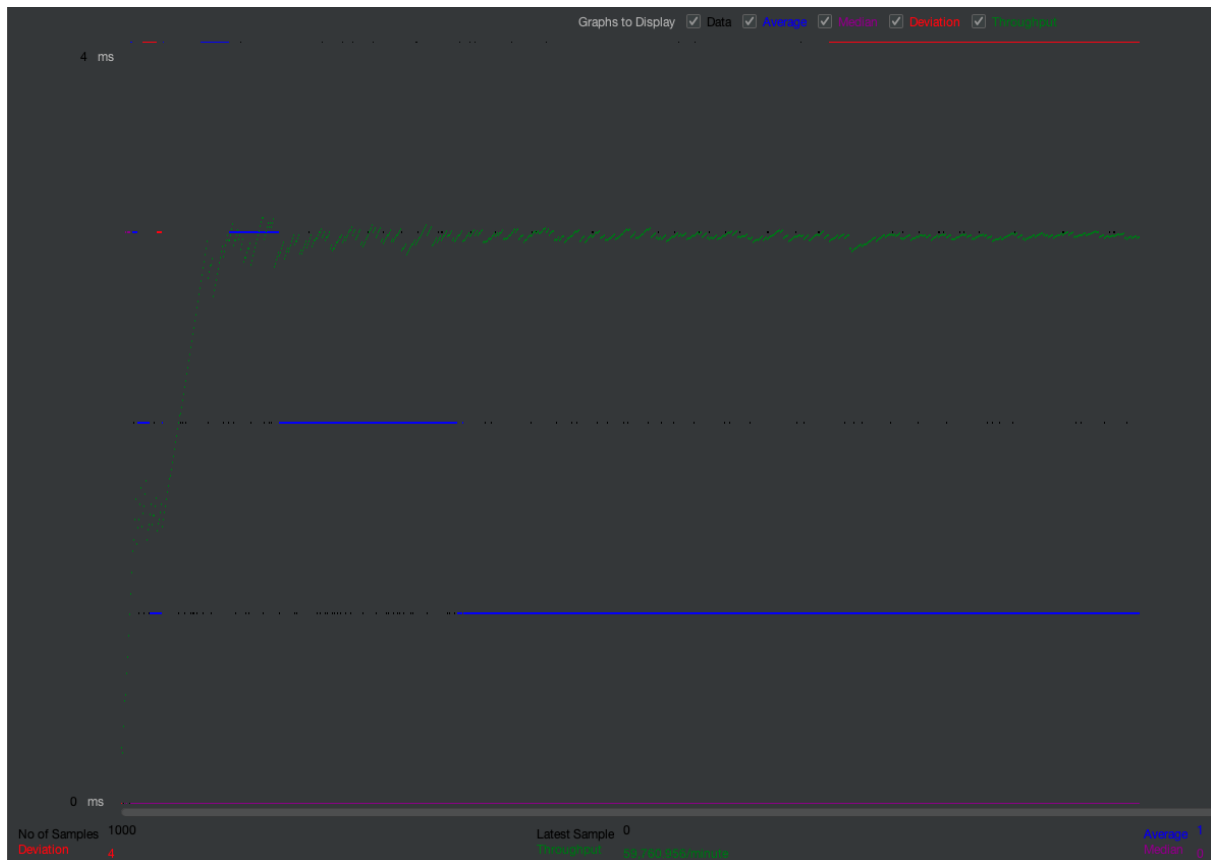
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
OPEN_CONNECTION	100	3	1	37	4.79	0.00%	100.8/sec	14.47	15.85	147.0
CONNECT	100	0	0	1	0.10	0.00%	104.7/sec	0.00	6.44	.0
CONNECT_RESPONSE	100	0	0	7	1.40	0.00%	104.7/sec	4.19	0.00	41.0
SUBSCRIBE	100	0	0	1	0.10	0.00%	105.5/sec	0.00	8.45	.0
SEND_JOIN	100	0	0	1	0.26	0.00%	105.5/sec	0.00	19.35	.0
SEND_JOIN_RESPONSE	100	11	0	88	21.46	0.00%	104.7/sec	47.17	0.00	461.2
SEND_CHAT	100	0	0	1	0.17	0.00%	115.2/sec	0.00	25.40	.0
SEND_CHAT_RESPONSE	100	4	0	27	3.43	0.00%	114.5/sec	50.90	0.00	455.0
DISCONNECT	100	0	0	1	0.10	0.00%	114.8/sec	0.00	3.14	.0
CLOSE_CONNECTION	100	0	0	1	0.32	25.00%	114.9/sec	2.27	3.48	20.2
TOTAL	1000	1	0	88	7.82	2.50%	996.0/sec	109.38	75.74	112.4



thread 갯수(유저): 1000

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
OPEN_CONNECTION	100	1	0	22	2.47	0.00%	100.5/sec	14.43	15.80	147.0
CONNECT	100	0	0	1	0.17	0.00%	102.8/sec	0.00	6.32	.0
CONNECT_RESPON...	100	0	0	5	0.83	0.00%	102.8/sec	4.12	0.00	41.0
SUBSCRIBE	100	0	0	1	0.14	0.00%	103.3/sec	0.00	8.27	.0
SEND_JOIN	100	0	0	1	0.20	0.00%	103.3/sec	0.00	19.27	.0
SEND_JOIN_RESPON...	100	7	0	57	11.15	0.00%	102.9/sec	47.22	0.00	470.0
SEND_CHAT	100	0	0	0	0.00	0.00%	109.3/sec	0.00	24.55	.0
SEND_CHAT_RESPO...	100	3	0	7	1.35	0.00%	108.7/sec	47.48	0.00	447.3
DISCONNECT	100	0	0	1	0.10	0.00%	108.8/sec	0.00	2.98	.0
CLOSE_CONNECTION	100	0	0	1	0.27	26.00%	108.8/sec	2.12	3.29	20.0
TOTAL	1000	1	0	57	4.33	2.60%	996.0/sec	109.45	76.45	112.5



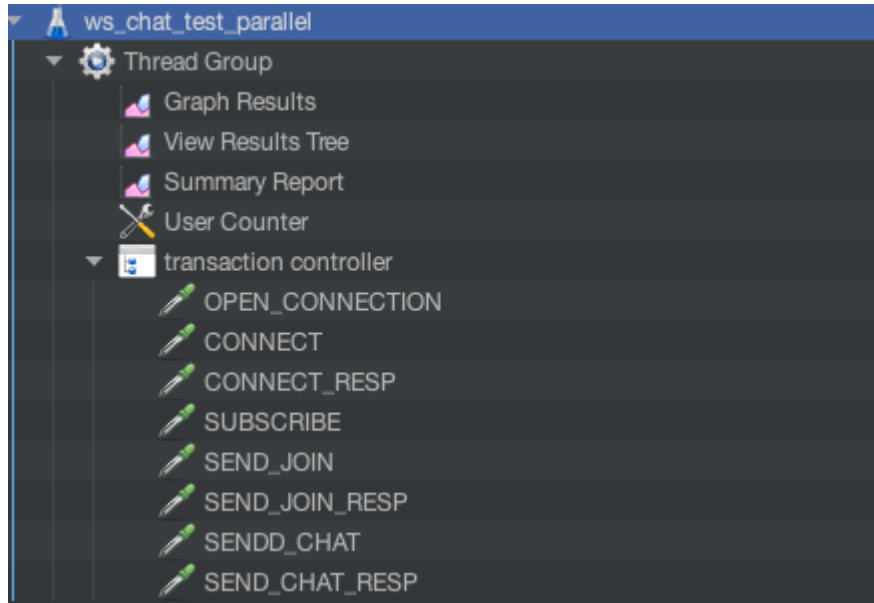


## Connection 유지하는 시나리오

생각해보니 여러명이 동시에 response 받도록 접속, 연결, 구독, 전송 이후에도 Connection 유지하도록 수정해서 테스트하는 시나리오가 대규모 단체 채팅의 부하 테스트에는 더 적합한 게 아닌가 라는 생각이 들어 disconnect, close connection을 disable하고 테스트 해 보았습니다.

일반적으로 JMeter의 Transaction Controller는 해당 트랜잭션에 속하는 모든 요청을 순차적으로 실행합니다. 그러나 이는 그룹화된 요청들이 각각 병렬로 실행되지 않음을 의미하지는 않습니다. 모든 요청이 동시에 실행되도록 보장하려면 여러 개의 Transaction Controller를 사용하거나 병렬 처리를 지원하는 다른 컨트롤러를 사용해야 합니다.

병렬 처리를 원하는 경우, Parallel Controller를 사용할 수 있습니다. 하지만 Jmeter 버전과 해당 플러그인 호환성 이슈로 현재 사용하지 못하여 Transaction Controller를 사용했습니다.



## thread 갯수(유저): 1

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
OPEN_CONNECTION	100	3	1	40	5.45	0.00%	101.0/sec	14.50	15.88	147.0
CONNECT	100	0	0	1	0.30	0.00%	105.6/sec	0.00	6.50	.0
CONNECT_RESP	100	2	0	23	4.67	0.00%	105.6/sec	4.23	0.00	41.0
SUBSCRIBE	100	0	0	1	0.22	0.00%	107.2/sec	0.00	8.58	.0
SEND_JOIN	100	0	0	1	0.24	0.00%	107.2/sec	0.00	19.71	.0
SEND_JOIN_RESP	100	35	1	200	60.85	0.00%	106.8/sec	45.37	0.00	434.9
SEND_CHAT	100	0	0	1	0.32	0.00%	135.0/sec	0.00	30.23	.0
SEND_CHAT_RESP	100	2	0	12	2.47	0.00%	134.6/sec	57.29	0.00	435.9
transaction controller	100	43	4	254	64.14	0.00%	100.1/sec	103.50	70.74	1058.8
TOTAL	900	9	0	254	33.67	0.00%	900.9/sec	207.00	141.49	235.3

## thread 갯수(유저): 100

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
OPEN_CONNECTION	100	4	1	29	5.09	0.00%	100.7/sec	14.46	15.83	147.0
CONNECT	100	0	0	2	0.36	0.00%	101.6/sec	0.00	6.25	.0
CONNECT_RESP	100	11	0	105	21.25	0.00%	101.6/sec	4.07	0.00	41.0
SUBSCRIBE	100	0	0	2	0.34	0.00%	101.8/sec	0.00	8.15	.0
SEND_JOIN	100	0	0	1	0.34	0.00%	101.8/sec	0.00	18.99	.0
SEND_JOIN_RESP	100	46	0	244	63.22	0.00%	100.6/sec	43.00	0.00	437.7
SEND_CHAT	100	0	0	2	0.39	0.00%	132.3/sec	0.00	30.10	.0
SEND_CHAT_RESP	100	3	0	21	4.54	0.00%	130.5/sec	55.89	0.00	438.4
transaction controller	100	67	6	257	62.76	0.00%	98.2/sec	102.08	70.03	1064.2
TOTAL	900	15	0	257	38.49	0.00%	884.1/sec	204.17	140.06	236.5

- avg: 3-15

## thread 갯수(유저): 500

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
OPEN_CONNECTION	500	6	0	54	7.97	0.00%	500.5/sec	71.85	78.69	147.0
CONNECT	500	0	0	1	0.19	0.00%	515.5/sec	0.00	31.71	.0
CONNECT_RESP	500	76	0	230	61.92	0.00%	485.4/sec	19.44	0.00	41.0
SUBSCRIBE	500	0	0	9	0.58	0.00%	495.0/sec	0.00	39.64	.0
SEND_JOIN	500	0	0	5	0.48	0.00%	495.0/sec	0.00	92.34	.0
SEND_JOIN_RESP	500	66	5	259	56.99	0.00%	478.5/sec	205.43	0.00	439.6
SEND_CHAT	500	0	0	2	0.17	0.00%	578.7/sec	0.00	131.68	.0
SEND_CHAT_RESP	500	6	0	48	10.58	0.00%	575.4/sec	246.40	0.00	438.5
transaction controller	500	156	13	410	71.06	0.00%	454.1/sec	472.83	323.75	1066.2
TOTAL	4500	34	0	410	63.58	0.00%	4087.2/sec	945.66	647.49	236.9

- avg: 30-35

## thread 갯수(유저): 1000

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
OPEN_CONNE...	1000	8	0	61	11.23	0.00%	1002.0/sec	143.84	157.54	147.0
CONNECT	1000	0	0	13	0.84	0.00%	1050.4/sec	0.00	64.63	.0
CONNECT_RE...	1000	168	0	523	178.55	0.00%	903.3/sec	36.17	0.00	41.0
SUBSCRIBE	1000	0	0	22	0.99	0.00%	918.3/sec	0.00	73.53	.0
SEND_JOIN	1000	0	0	22	1.01	0.00%	918.3/sec	0.00	173.97	.0
SEND_JOIN_R...	1000	230	13	628	124.81	0.00%	760.5/sec	325.86	0.00	438.8
SENDD_CHAT	1000	0	0	12	0.58	0.00%	1230.0/sec	0.00	284.68	.0
SEND_CHAT_...	1000	38	0	106	24.67	0.00%	1230.0/sec	527.30	0.00	439.0
transaction con...	1000	446	28	813	194.27	0.00%	724.1/sec	753.66	521.16	1065.8
TOTAL	9000	99	0	813	176.34	0.00%	6517.0/sec	1507.31	1042.33	236.8

- avg: 92~99

## thread 갯수(유저): 3000

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
OPEN_CONNE...	3000	47	0	214	48.52	0.00%	1509.1/sec	216.63	237.26	147.0
CONNECT	3000	0	0	15	0.44	0.00%	1601.7/sec	0.00	98.54	.0
CONNECT_RE...	3000	546	1	1194	285.95	0.00%	1404.5/sec	56.23	0.00	41.0
SUBSCRIBE	3000	0	0	19	0.78	0.00%	1489.6/sec	0.00	119.28	.0
SEND_JOIN	3000	0	0	23	1.06	0.00%	1490.3/sec	0.00	282.34	.0
SEND_JOIN_R...	3000	2876	192	6026	2710.97	42.80%	374.3/sec	91.86	0.00	251.3
SENDD_CHAT	3000	0	0	29	2.13	0.00%	413.2/sec	0.00	95.64	.0
SEND_CHAT_...	3000	2573	0	6006	2965.28	42.80%	226.2/sec	55.53	0.00	251.4
transaction con...	3000	6044	772	12904	5644.03	42.80%	210.4/sec	141.92	151.43	690.7
TOTAL	27000	1343	0	12904	3047.23	14.27%	1893.5/sec	283.84	302.85	153.5

- 에러가 발생하기 시작했습니다. 절반에 가까운 (1500) 유저가 서비스 응답(RESP)을 제대로 못받고 있습니다.

## 결론

1000명까진 괜찮은 것 같다...

?