

배경

1:1 채팅 기능이 구현되어있고 상대방을 어떤 이유로 신고하는 기능이 있다고 하자. 그리고 신고할 수 있는 횟수가 제한이 없다고하면 상대방을 계속 신고할 수 있다.

이때, 여러명이 상대방을 쉽없이 신고할 경우를 가정하여 테스트를 진행해보았다.

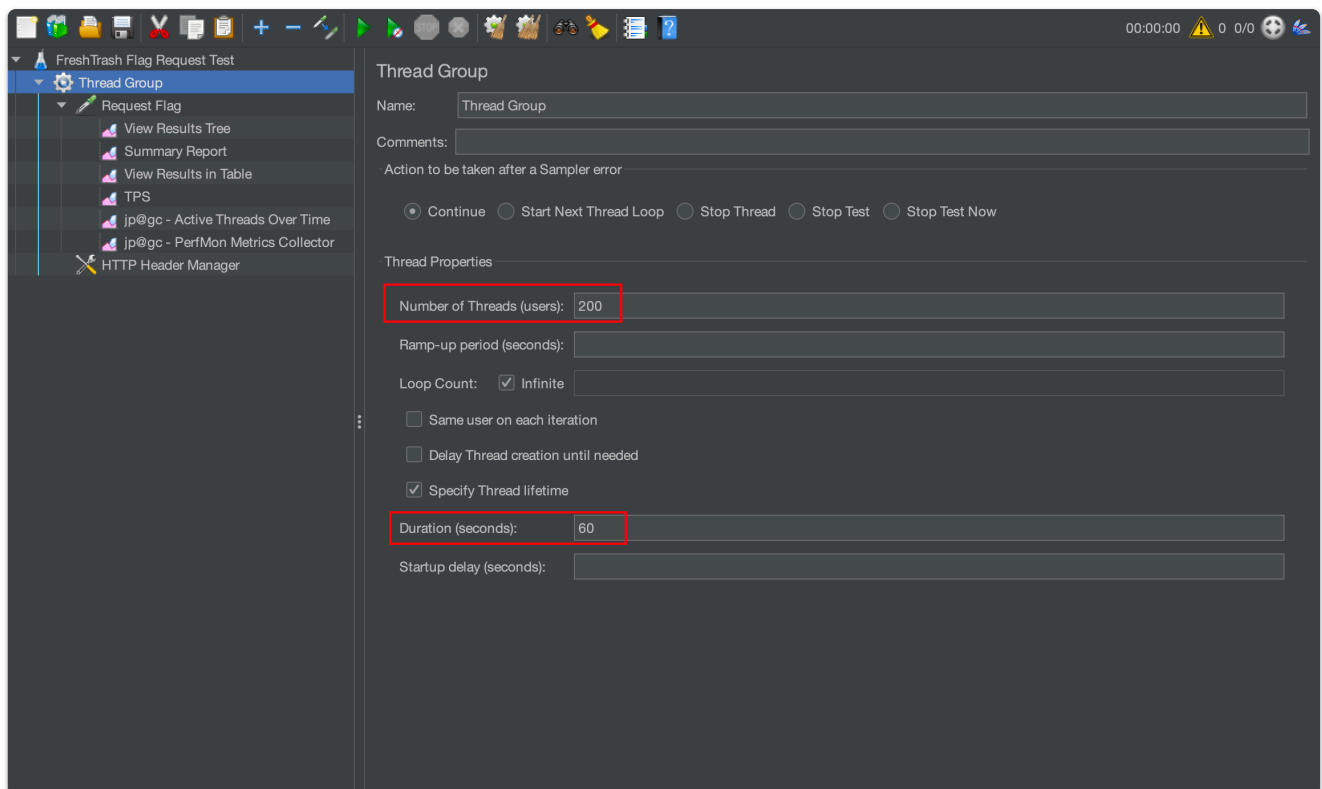
부하 테스트

Spring Boot로 구현한 API는 Docker 환경에 올려서 실행했으며, [CPU 2, Memory 4G]로 Resource 를 설정했다.

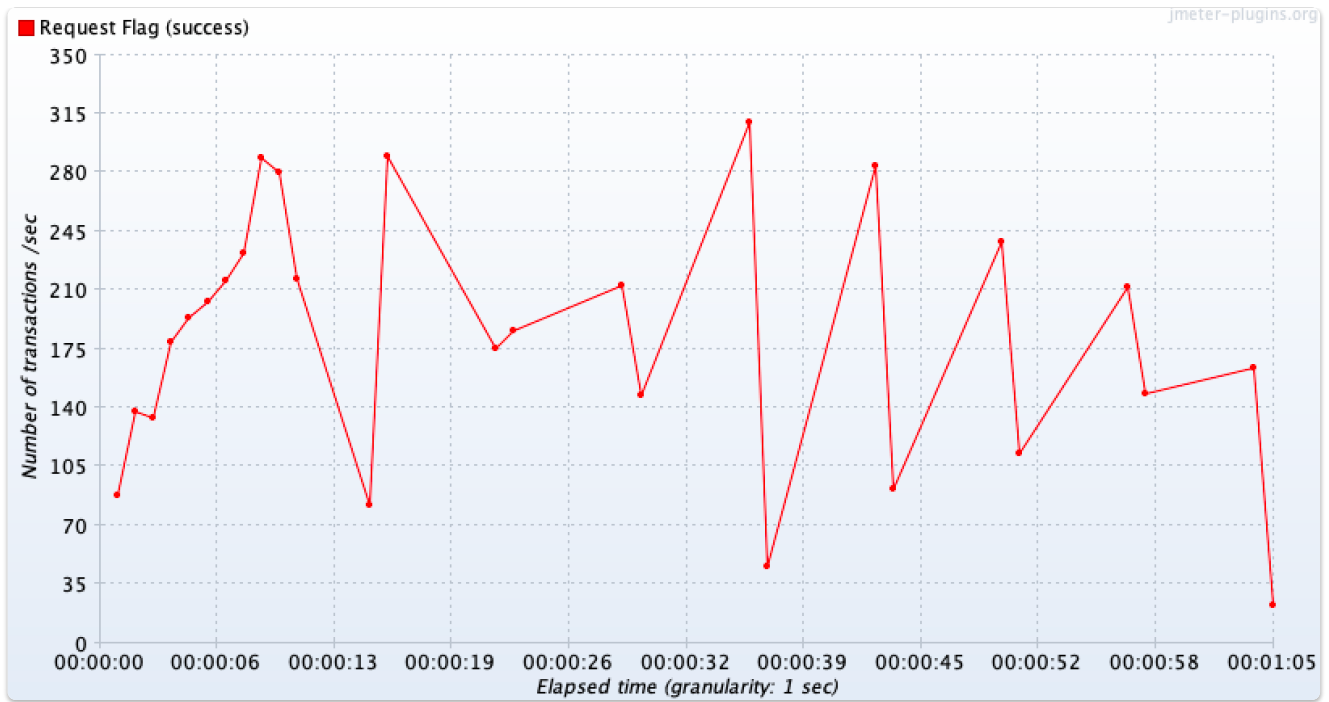
Jmeter 테스트

부하 테스트는 Jmeter를 사용하여 수행했다.

아래 설정은 **200명의 사용자가 60초 동안** 쉽없이 상대방을 신고하는 경우를 테스트하기 위한 설정이다.



실행 후 TPS 그래프를 확인해보면 7 ~ 8 초 이후 갑자기 TPS가 떨어지고 중간중간에 멈추는 구간이 발생하는 것을 볼 수 있었다.

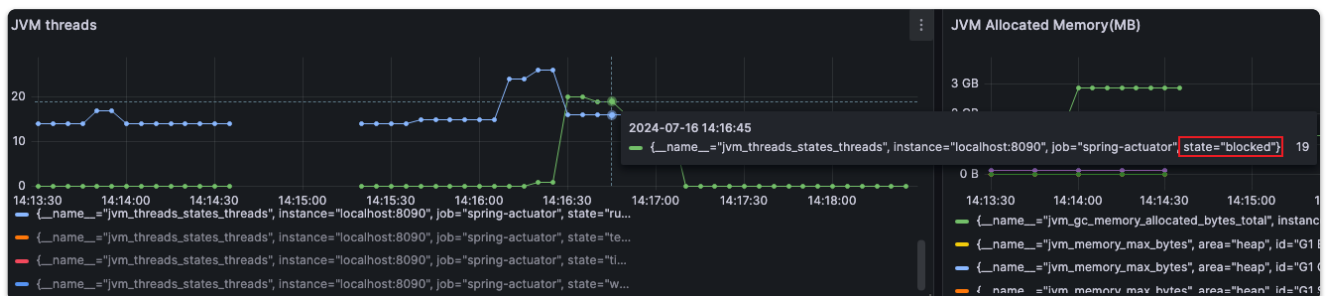


왜 요청을 잘 처리하다가 멈추게 되는 걸까?

모니터링

그 이유를 찾기위해 프로메테우스와 그라파나를 통해 모니터링을 해보았다.

갑자기 요청을 제대로 처리하지 못하는 구간에 어떤 부분이 문제인지 확인해보자.



JVM 스레드가 갑자기 block되는 것 같다.

스레드가 block되는 원인으로 뭐가 있을까?

1. 동기화 블록을 사용할 때, 하나의 스레드가 락을 획득하면 다른 스레드들은 해당 락이 해제될 때까지 블록된다.
2. I/O 작업을 수행하는 동안 해당 작업이 완료될 때까지 블록될 수 있다.
3. 데드락이 발생할 경우 블록된다.
4. 메모리 또는 스레드 고갈 등의 자원 고갈로 인해 블록될 수 있다.

현재 동기화 블록은 사용하지 않고 데드락이 걸릴 수 있는 부분은 코드에서 확인할 수 없었다.

정확하게 어떤 부분에서 스레드가 블록되었는지 확인할 수 있는 방법은 없을까?

스레드 덤프(Thread Dump)

스레드 덤프(Thread Dump)는 JVM 내의 모든 스레드의 상태와 스택 트레이스를 보여준다.

스레드 덤프를 생성하는 방법은 다음과 같이 다양하다.

1. `jstack [PID]`
2. `kill -3 [PID]`
3. `Thread.getAllStackTraces()`
4. VisualVM

여기서 간단하게 `jstack`으로 스레드 덤프를 확인해보겠다.

블록된 스레드 중 하나를 확인해보면 다음과 같다.

```
"http-nio-8080-exec-161" #248 daemon prio=5 os_prio=0 cpu=223.83ms
elapsed=3993.40s tid=0x0000ffff400c9f80 nid=0x104 waiting for monitor
entry [0x0000fffed2a4000]
    java.lang.Thread.State: BLOCKED (on object monitor)
        at
        java.io.PrintStream.writeln(java.base@17.0.11/PrintStream.java:718)
            - waiting to lock <0x00000000c2a1db18> (a java.io.PrintStream)
        at
        java.io.PrintStream.println(java.base@17.0.11/PrintStream.java:1028)
            at
            org.hibernate.engine.jdbc.spi.SqlStatementLogger.logStatement(SqlStatement
            Logger.java:147)
                at
                org.hibernate.engine.jdbc.spi.SqlStatementLogger.logStatement(SqlStatement
                Logger.java:125)
                    at
                    org.hibernate.engine.jdbc.internal.StatementPreparerImpl$StatementPreparat
                    ionTemplate.prepareStatement(StatementPreparerImpl.java:170)
                    ...
```

```
- waiting to lock <0x00000000c2a1db18> (a java.io.PrintStream)
```

`PrintStream`으로 출력하는 작업을 수행하는 과정에서 문제가 생겼다는 사실을 확인할 수 있다. 즉, I/O 작업을 수행하는 동안 스레드가 블록된 경우라는 것을 확인할 수 있었다.

그리고 `SqlStatementLogger.logStatement`를 통해 SQL 쿼리문을 출력에 대한 작업이라는 것을 알 수 있다.

결론

위와 같은 사실들을 바탕으로 수많은 쿼리문을 출력하는 과정에서 다른 스레드들은 출력이 완료될때까지 블록이 된다는 것을 알 수 있었다.

따라서 운영(배포) 환경에서는 Hibernate의 쿼리문 출력을 하지 않도록 설정해야한다는 결론을 낼 수 있었다.

```
spring.jpa.show-sql: false
```