# RemoraJ stress test report

running Jboss 7.2

Remora 0.1.4-SNAPSHOT built 2020-02-28

With testHarnesses built 2020-02-28:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Threads | Schedulle (every ..) | Target |
| ApacheHttpClientHarness | 50 | 50ms | Same Jboos |
| SQLHarness | 50 | 50ms | SQL server running the same mashine |
| MQReceiveHarness | 50 | 50ms | MQ running on VirtualBox, same queue as send |
| MQSendHarness | 50 | 50ms | MQ running on VirtualBox, same queue as receive |
| WebsocketSendHarness | 50 | 50ms | Same Jboos, testHArnness endpoint |
| KafkaConsumerHarness | 50 | 50ms | Kafka running on the same mashine |
| KafkaProducerHarness | 50 | 50ms | Kafka running on the same mashine |

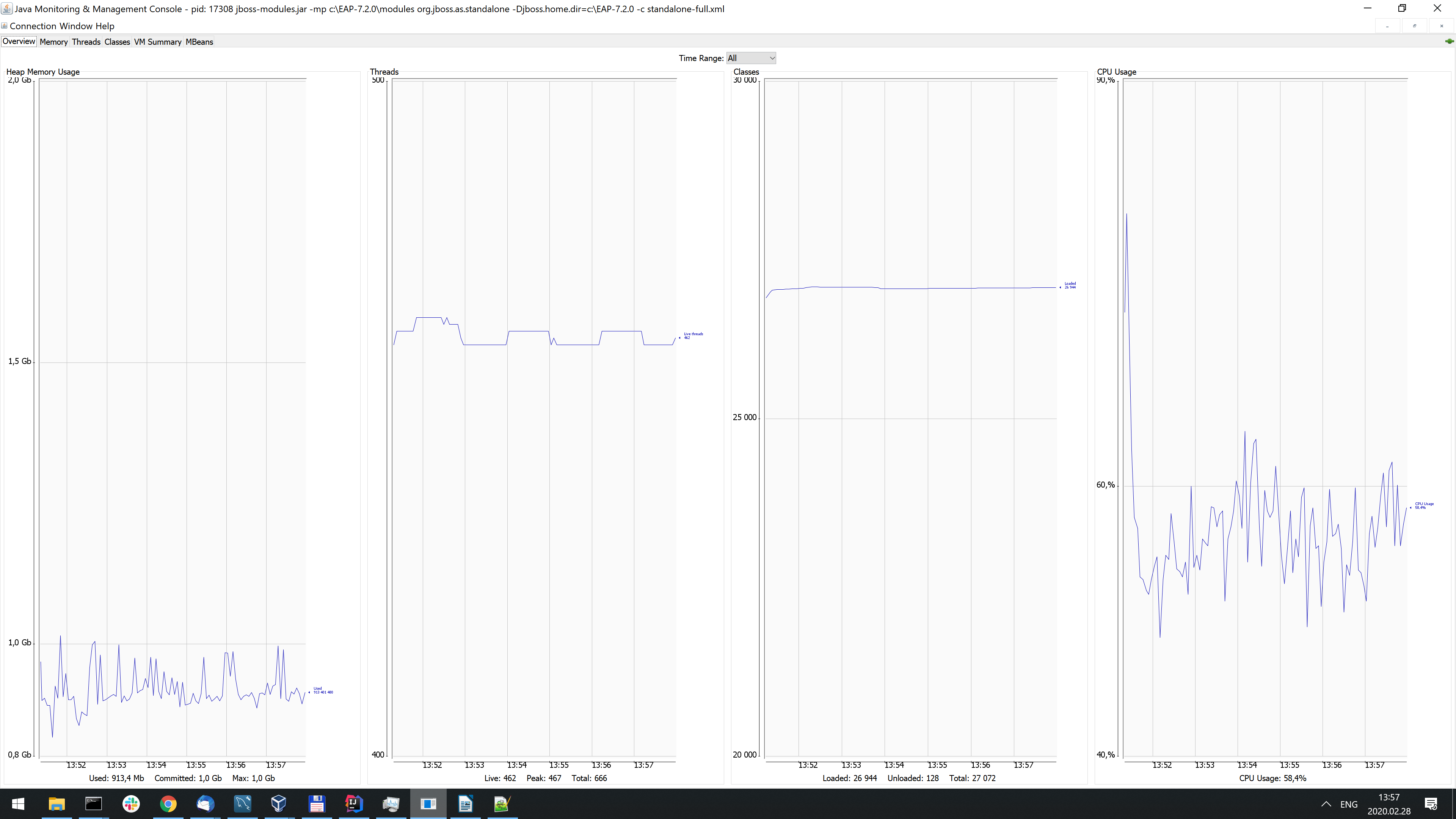
Mesuring with Jconsole.

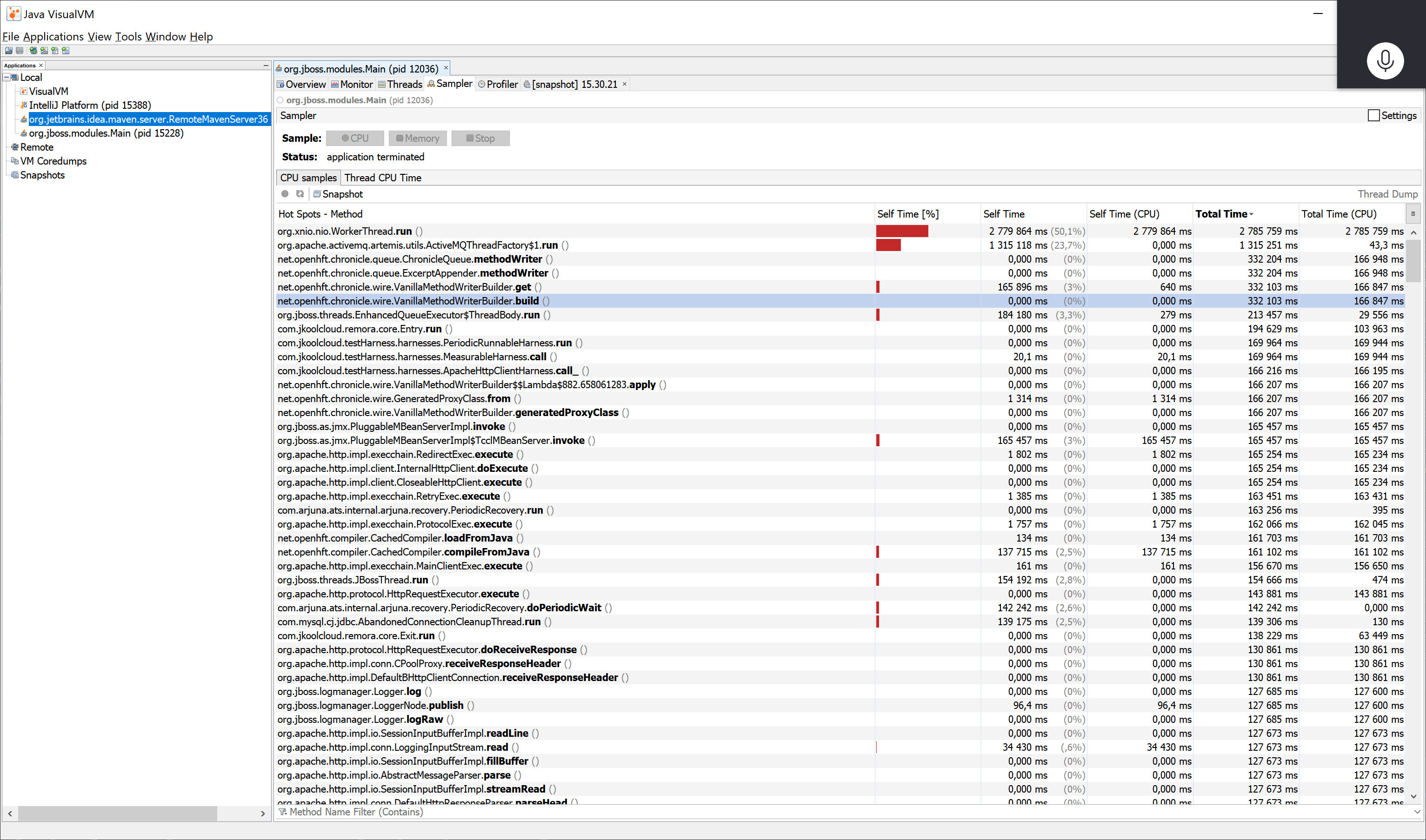
# Results:

### With RemoraJ

Heap Memory Usage: ~900

CPU usage: 50-60



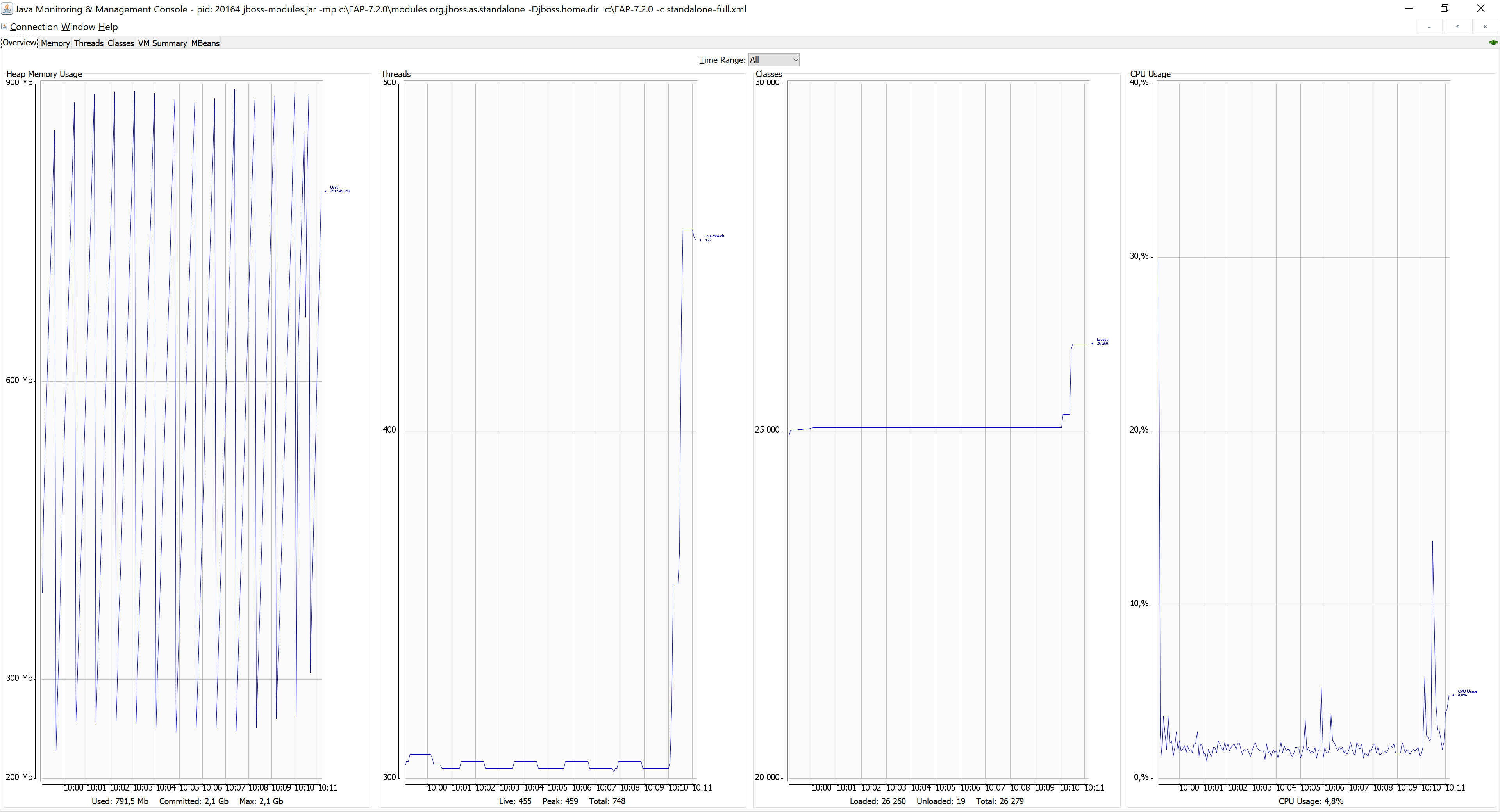


### With RemoraJ without chronicle queue

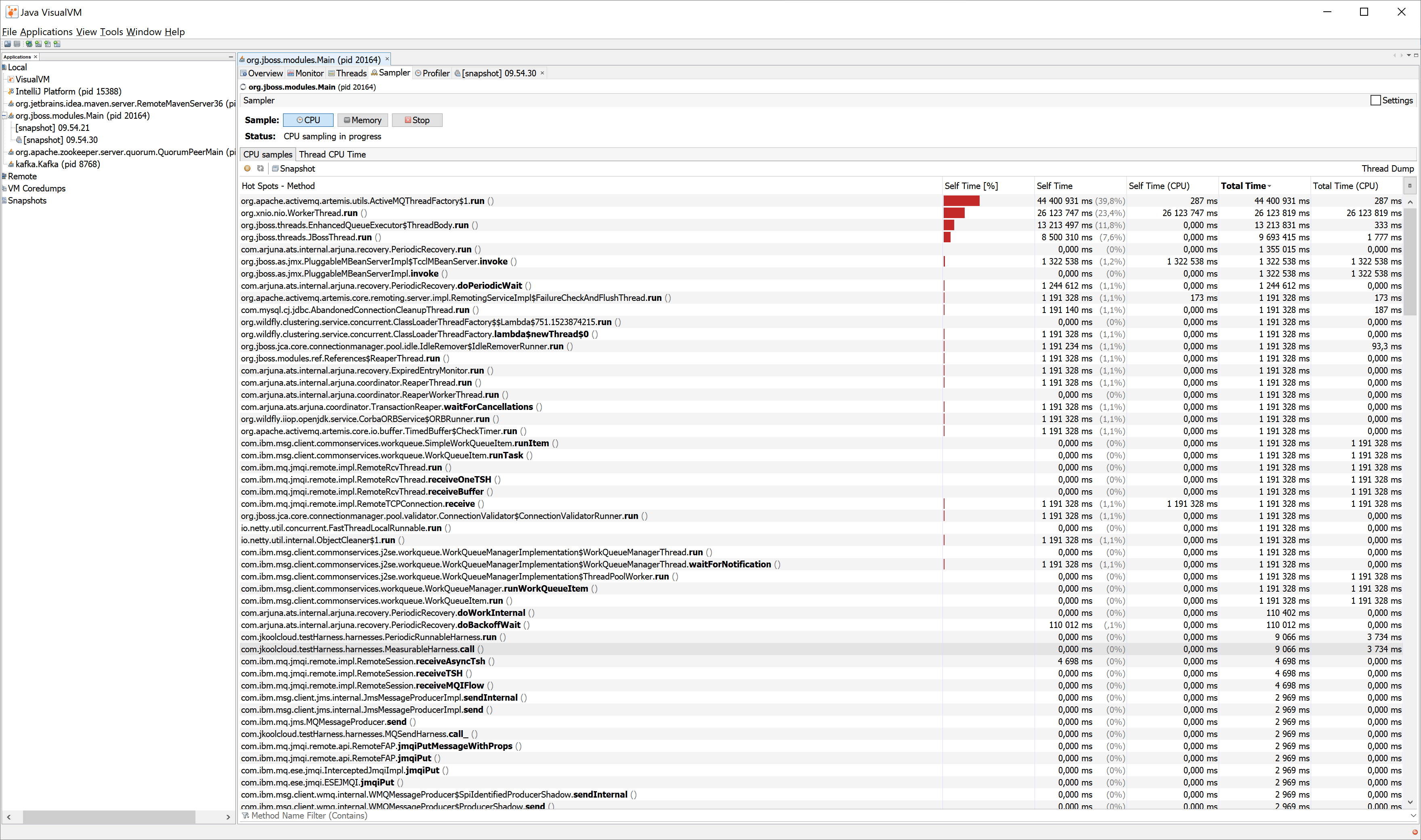
The same test condition except chronicle queue opted out, no files are created or written.

Heap Memory Usage: ~900

CPU usage: 50-55%



.

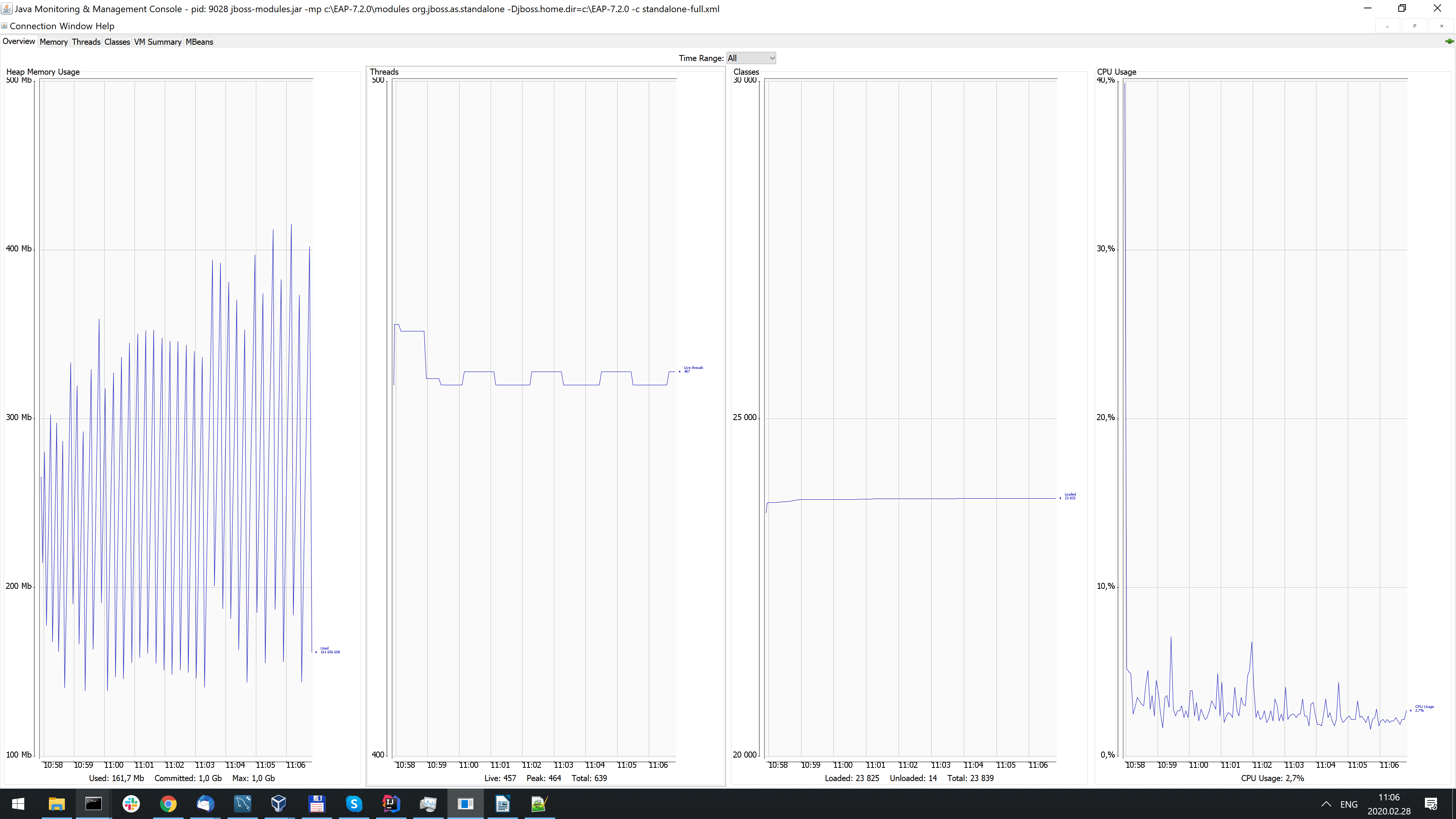


### Without RemoraJ

-javaagent option is commented out;

Heap Memory Usage: ~300

CPU usage: <10%



### Conclusions

For test scenario where ~350 thread are running invoking variuos instrumented remote services (HTTP, JMS send and receive) scenario RemoraJ causes much overhead. This might be caused by GC as Jboss was running with „JAVA\_OPTS=-Xms1G -Xmx1G -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m“. Futher investigation needed.