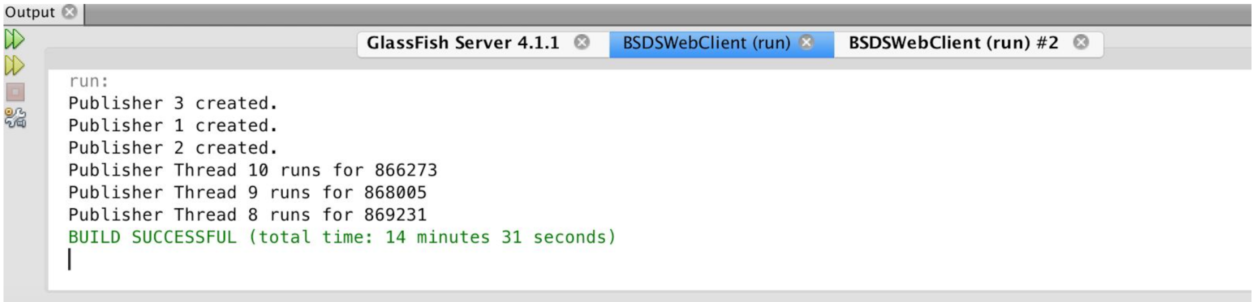
I have refactored my code and created 3 major modules: one EJB module that deals with the major logic for content aggregation; one RESTful web server that accepts incoming PUT and GET HTTP requests and calls the remote methods in the previous EJB module to process these requests; one Message-Driven Bean module along with JDBC that handles read and write to the Postgresql database. In principle these 3 modules should be running on 3 different hosts, but I encountered some problems with calling remote Bean and exchanging messages. I ended up setting up all these 3 modules on a single host, so unsurprisingly my implementation is very inefficient. I'll try to optimize my implementation later but below is what I currently have...

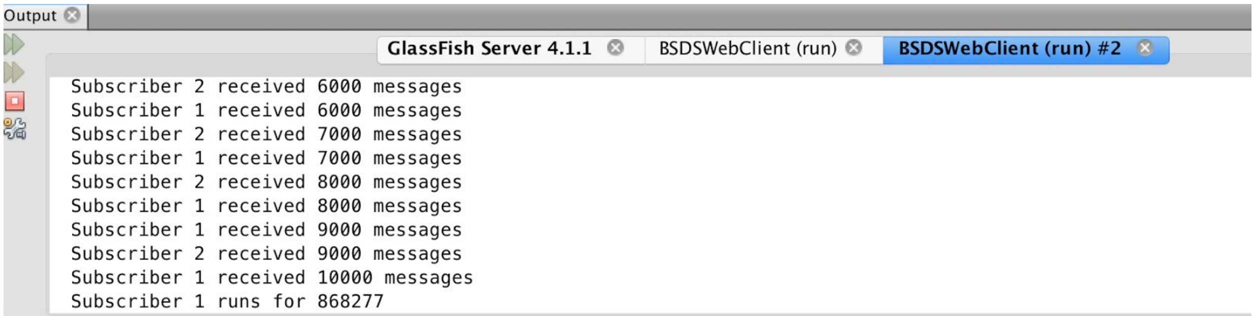
Step 3:

The end point is http://54.89.76.7:8080/BSDSWebApplication/webresources/BSDS/. Running the same publisher and subscriber clients as step 1, which creates 3 publisher threads and 2 subscriber clients. The running result is:



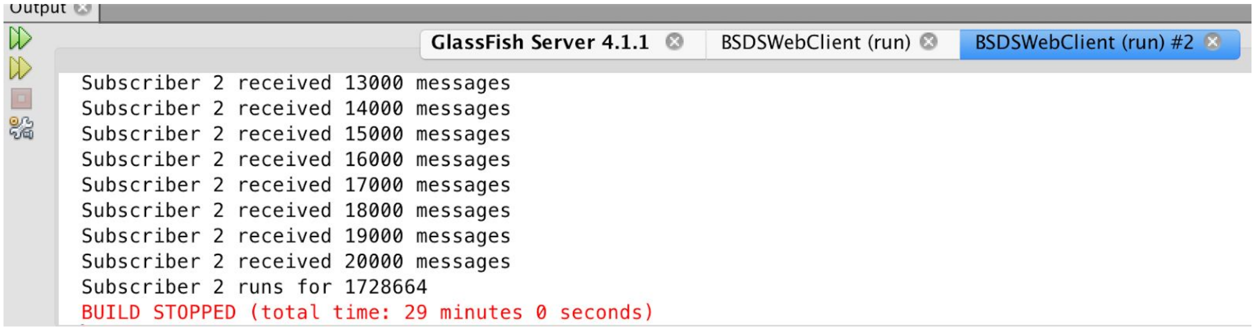
```
run:
Publisher 3 created.
Publisher 1 created.
Publisher 2 created.
Publisher Thread 10 runs for 866273
Publisher Thread 9 runs for 868005
Publisher Thread 8 runs for 869231
BUILD SUCCESSFUL (total time: 14 minutes 31 seconds)
```

All the 3 publishers run for about 14.5 minutes. As a comparison, the publishers in step 1 run for less than 3 minutes.



```
Subscriber 2 received 6000 messages
Subscriber 1 received 6000 messages
Subscriber 2 received 7000 messages
Subscriber 1 received 7000 messages
Subscriber 2 received 8000 messages
Subscriber 1 received 8000 messages
Subscriber 1 received 9000 messages
Subscriber 2 received 9000 messages
Subscriber 1 received 10000 messages
Subscriber 1 runs for 868277
```

Subscriber 1 (which should receive 10000 messages) runs for roughly 14.5 minutes as well, similar to other publishers. In step 1 this subscriber also runs for about 3 minutes.

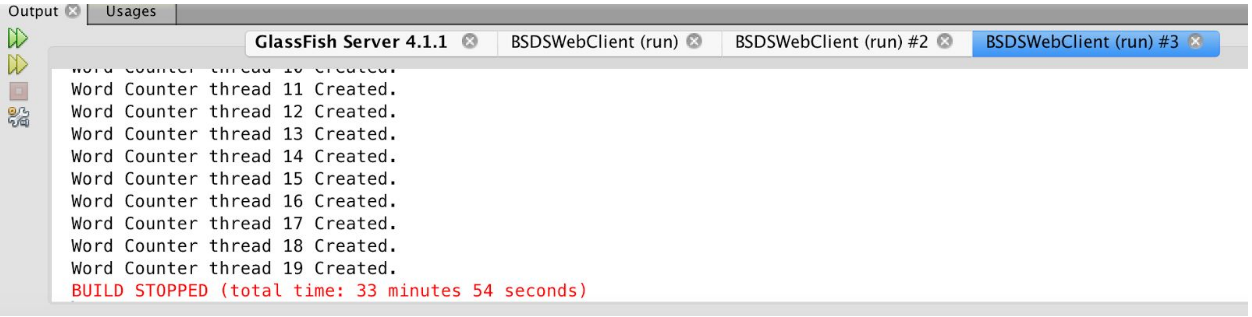


```
Subscriber 2 received 13000 messages
Subscriber 2 received 14000 messages
Subscriber 2 received 15000 messages
Subscriber 2 received 16000 messages
Subscriber 2 received 17000 messages
Subscriber 2 received 18000 messages
Subscriber 2 received 19000 messages
Subscriber 2 received 20000 messages
Subscriber 2 runs for 1728664
BUILD STOPPED (total time: 29 minutes 0 seconds)
```

Subscriber 2 (which should receive 20000 messages) runs for 29 minutes, which is exactly twice as much as that of subscriber 1. This indicates that, in the test the locking mechanism at server side is not a significant factor of efficiency (compared to step 1), instead the most significant factor should be the network transmission.

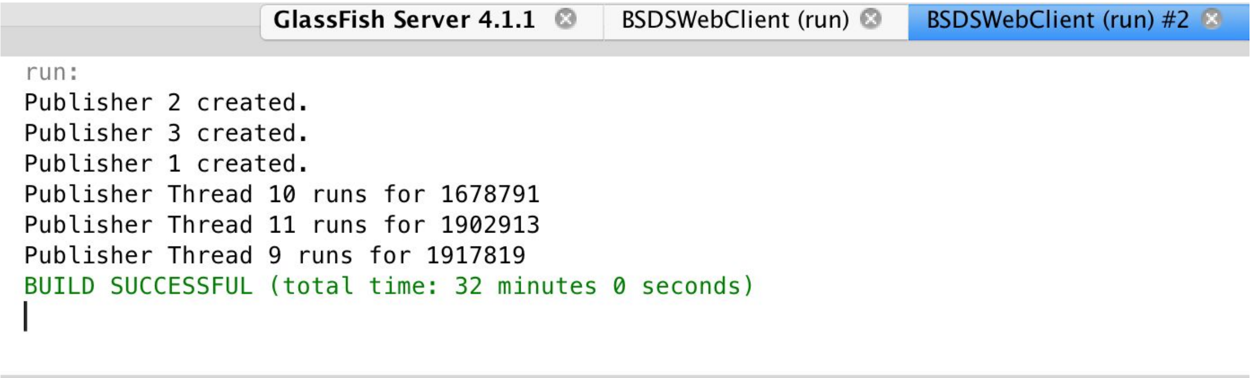
Step 4:

Running the TermCount Client along with the publisher and subscriber clients.



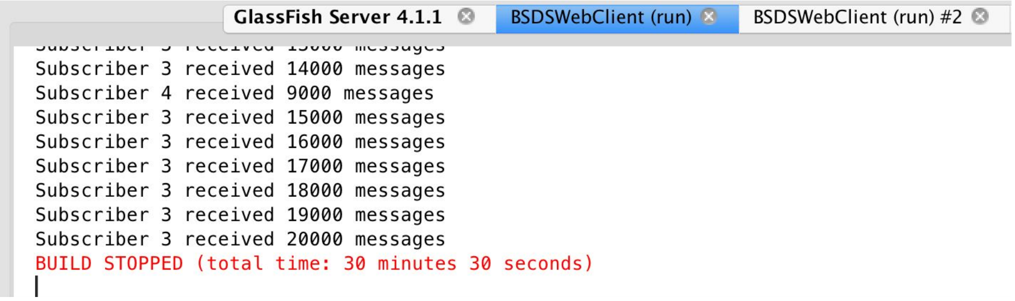
```
Word Counter thread 10 Created.
Word Counter thread 11 Created.
Word Counter thread 12 Created.
Word Counter thread 13 Created.
Word Counter thread 14 Created.
Word Counter thread 15 Created.
Word Counter thread 16 Created.
Word Counter thread 17 Created.
Word Counter thread 18 Created.
Word Counter thread 19 Created.
BUILD STOPPED (total time: 33 minutes 54 seconds)
```

The wall time for subscribers and publishers:



```
run:
Publisher 2 created.
Publisher 3 created.
Publisher 1 created.
Publisher Thread 10 runs for 1678791
Publisher Thread 11 runs for 1902913
Publisher Thread 9 runs for 1917819
BUILD SUCCESSFUL (total time: 32 minutes 0 seconds)
```

One publisher on Topic1 runs for about 27 minutes, and the other two on Topic2 run for about 32 minutes.



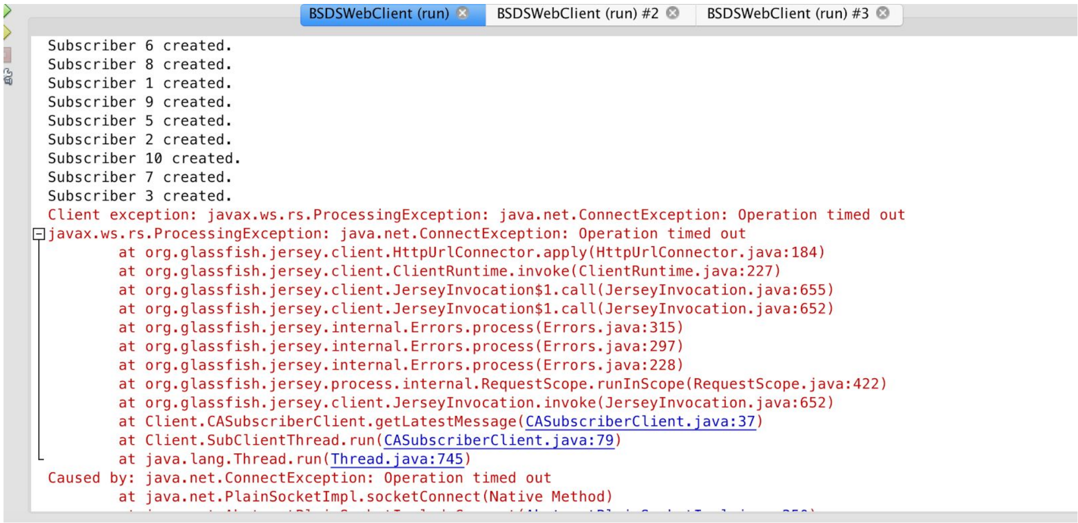
```
Subscriber 3 received 13000 messages
Subscriber 3 received 14000 messages
Subscriber 4 received 9000 messages
Subscriber 3 received 15000 messages
Subscriber 3 received 16000 messages
Subscriber 3 received 17000 messages
Subscriber 3 received 18000 messages
Subscriber 3 received 19000 messages
Subscriber 3 received 20000 messages
BUILD STOPPED (total time: 30 minutes 30 seconds)
```

One subscriber on Topic1 (10000 messages) run for about 27 minutes, and the other (20000 messages) run for about 30.5 minutes.

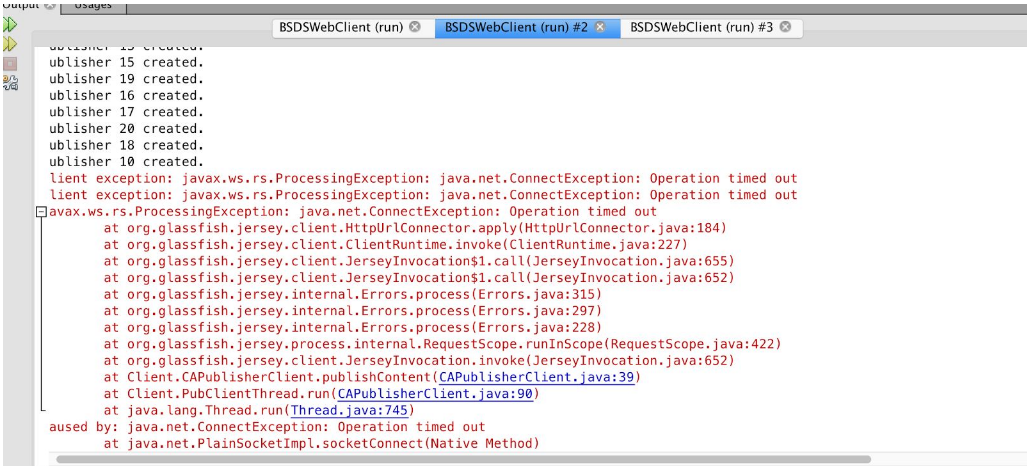
The wall time for both publisher and subscriber has increased compared to Step 3.

Step 5:

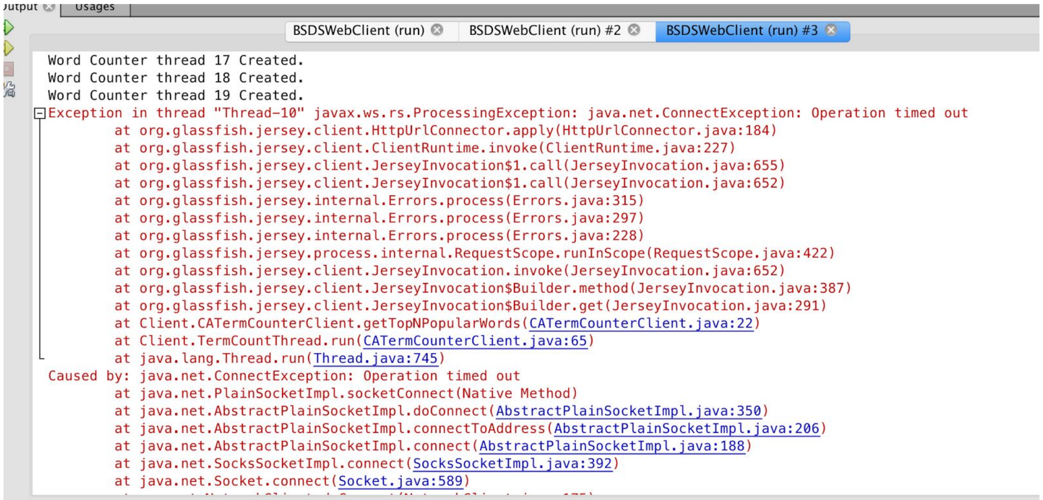
Cannot run this stress test, probably because one single host is accepting too much traffic. After a few minutes, all the API calls failed with a time out:



```
Subscriber 6 created.
Subscriber 8 created.
Subscriber 1 created.
Subscriber 9 created.
Subscriber 5 created.
Subscriber 2 created.
Subscriber 10 created.
Subscriber 7 created.
Subscriber 3 created.
Client exception: javax.ws.rs.ProcessingException: java.net.ConnectException: Operation timed out
javax.ws.rs.ProcessingException: java.net.ConnectException: Operation timed out
        at org.glassfish.jersey.client.HttpUrlConnector.apply(HttpUrlConnector.java:184)
        at org.glassfish.jersey.client.ClientRuntime.invoke(ClientRuntime.java:227)
        at org.glassfish.jersey.client.JerseyInvocation$1.call(JerseyInvocation.java:655)
        at org.glassfish.jersey.client.JerseyInvocation$1.call(JerseyInvocation.java:652)
        at org.glassfish.jersey.internal.Errors.process(Errors.java:315)
        at org.glassfish.jersey.internal.Errors.process(Errors.java:297)
        at org.glassfish.jersey.internal.Errors.process(Errors.java:228)
        at org.glassfish.jersey.process.internal.RequestScope.runInScope(RequestScope.java:422)
        at org.glassfish.jersey.client.JerseyInvocation.invoke(JerseyInvocation.java:652)
        at Client.CASubscriberClient.getLatestMessage(CASubscriberClient.java:37)
        at Client.SubClientThread.run(CASubscriberClient.java:79)
        at java.lang.Thread.run(Thread.java:745)
Caused by: java.net.ConnectException: Operation timed out
        at java.net.PlainSocketImpl.socketConnect(Native Method)
```



```
ublisher 15 created.
ublisher 19 created.
ublisher 16 created.
ublisher 17 created.
ublisher 20 created.
ublisher 18 created.
ublisher 10 created.
lient exception: javax.ws.rs.ProcessingException: java.net.ConnectException: Operation timed out
lient exception: javax.ws.rs.ProcessingException: java.net.ConnectException: Operation timed out
avax.ws.rs.ProcessingException: java.net.ConnectException: Operation timed out
       at org.glassfish.jersey.client.HttpUrlConnector.apply(HttpUrlConnector.java:184)
       at org.glassfish.jersey.client.ClientRuntime.invoke(ClientRuntime.java:227)
       at org.glassfish.jersey.client.JerseyInvocation$1.call(JerseyInvocation.java:655)
       at org.glassfish.jersey.client.JerseyInvocation$1.call(JerseyInvocation.java:652)
       at org.glassfish.jersey.internal.Errors.process(Errors.java:315)
       at org.glassfish.jersey.internal.Errors.process(Errors.java:297)
       at org.glassfish.jersey.internal.Errors.process(Errors.java:228)
       at org.glassfish.jersey.process.internal.RequestScope.runInScope(RequestScope.java:422)
       at org.glassfish.jersey.client.JerseyInvocation.invoke(JerseyInvocation.java:652)
       at Client.CAPublisherClient.publishContent(CAPublisherClient.java:39)
       at Client.PubClientThread.run(CAPublisherClient.java:90)
       at java.lang.Thread.run(Thread.java:745)
aused by: java.net.ConnectException: Operation timed out
       at java.net.PlainSocketImpl.socketConnect(Native Method)
```



```
Word Counter thread 17 Created.
Word Counter thread 18 Created.
Word Counter thread 19 Created.
Exception in thread "Thread-10" javax.ws.rs.ProcessingException: java.net.ConnectException: Operation timed out
        at org.glassfish.jersey.client.HttpUrlConnector.apply(HttpUrlConnector.java:184)
        at org.glassfish.jersey.client.ClientRuntime.invoke(ClientRuntime.java:227)
        at org.glassfish.jersey.client.JerseyInvocation$1.call(JerseyInvocation.java:655)
        at org.glassfish.jersey.client.JerseyInvocation$1.call(JerseyInvocation.java:652)
        at org.glassfish.jersey.internal.Errors.process(Errors.java:315)
        at org.glassfish.jersey.internal.Errors.process(Errors.java:297)
        at org.glassfish.jersey.internal.Errors.process(Errors.java:228)
        at org.glassfish.jersey.process.internal.RequestScope.runInScope(RequestScope.java:422)
        at org.glassfish.jersey.client.JerseyInvocation.invoke(JerseyInvocation.java:652)
        at org.glassfish.jersey.client.JerseyInvocation$Builder.method(JerseyInvocation.java:387)
        at org.glassfish.jersey.client.JerseyInvocation$Builder.get(JerseyInvocation.java:291)
        at Client.CATermCounterClient.getTopNPopularWords(CATermCounterClient.java:22)
        at Client.TermCountThread.run(CATermCounterClient.java:65)
        at java.lang.Thread.run(Thread.java:745)
Caused by: java.net.ConnectException: Operation timed out
        at java.net.PlainSocketImpl.socketConnect(Native Method)
        at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
        at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
        at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
        at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
        at java.net.Socket.connect(Socket.java:589)
```