

Hi, I'm running the Talend Runtime karaf container from Talend ESB r101800 v.5.3.0 with these features installed and running (see attached).

I'm running one ActiveMQ broker, 4 jobs that produce ActiveMQ messages and 4 that consume them.

No matter how large I make \$JAVA_MAX_MEM, eventually the container occupies all the memory allotted to it, the web consoles become slow and eventually unresponsive. After that, processing of messages stops altogether and I have to force kill the container and restart it.

What I've noticed when checking the ActiveMQ queue stats (activemq:query -QQueue=*) is that some of my queues have a constantly growing ProducerCount number. I don't know if that's related.

Also attached is my ActiveMQ configuration XML.

```
<!--
```

```
Licensed to the Apache Software Foundation (ASF) under one or more  
contributor license agreements. See the NOTICE file distributed with  
this work for additional information regarding copyright ownership.  
The ASF licenses this file to You under the Apache License, Version 2.0  
(the "License"); you may not use this file except in compliance with  
the License. You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.
```

```
-->
```

```
<beans
```

```
xmlns="http://www.springframework.org/schema/beans"
```

```
xmlns:amq="http://activemq.apache.org/schema/core"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xmlns:osgi="http://www.springframework.org/schema/osgi"
```

```
xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
```

```
http://activemq.apache.org/schema/core http://activemq.apache.org/schema/core/activemq-core.xsd
```

```
http://activemq.apache.org/camel/schema/spring http://activemq.apache.org/camel/schema/spring/camel-spring.xsd
```

```
http://www.springframework.org/schema/osgi http://www.springframework.org/schema/osgi/spring-osgi.xsd">
```

```
<!-- Allows us to use system properties as variables in this configuration file -->
```

```
<bean class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"/>
```

```
<broker xmlns="http://activemq.apache.org/schema/core" brokerName="ESB-MSH-01" dataDirectory="${karaf.data}/activemq/ESB-MSH-01" useShutdownHook="false"  
startAsync="true">
```

```
<!--
```

```
For better performances use VM cursor and small memory limit.
```

```
For more information, see:
```

```
http://activemq.apache.org/message-cursors.html
```

```
Also, if your producer is "hanging", it's probably due to producer flow control.
```

```
For more information, see:
```

```
http://activemq.apache.org/producer-flow-control.html
```

```
-->
```

```
<destinationPolicy>
```

```
<policyMap>
```

```
<policyEntries>
```

```
<policyEntry topic=">" producerFlowControl="true" memoryLimit="1mb">
```

```
<pendingSubscriberPolicy>
```

```
<vmCursor />
</pendingSubscriberPolicy>
</policyEntry>
<policyEntry queue="" producerFlowControl="true" memoryLimit="1mb">
<!-- Use VM cursor for better latency
For more information, see:
```

<http://activemq.apache.org/message-cursors.html>

```
<pendingQueuePolicy>
<vmQueueCursor/>
</pendingQueuePolicy>
-->
</policyEntry>
</policyEntries>
</policyMap>
</destinationPolicy>
<!-- Use the following to configure how ActiveMQ is exposed in JMX -->
<managementContext>
<managementContext createConnector="false"/>
</managementContext>
<!--
Configure message persistence for the broker. The default persistence
mechanism is the KahaDB store (identified by the kahaDB tag).
For more information, see:
```

<http://activemq.apache.org/persistence.html>

```
-->
<persistenceAdapter>
<kahaDB directory="${karaf.data}/activemq/ESB-MSH-01/kahadb"/>
</persistenceAdapter>
<!--
The systemUsage controls the maximum amount of space the broker will
use before slowing down producers. For more information, see:
```

<http://activemq.apache.org/producer-flow-control.html>

```
<systemUsage>
<systemUsage>
<memoryUsage>
<memoryUsage limit="20 mb"/>
</memoryUsage>
<storeUsage>
<storeUsage limit="1 gb" name="foo"/>
</storeUsage>
<tempUsage>
<tempUsage limit="100 mb"/>
</tempUsage>
</systemUsage>
</systemUsage>
-->
<shutdownHooks>
<bean xmlns="http://www.springframework.org/schema/beans" id="hook" class="org.apache.activemq.hooks.SpringOsgiContextHook" />
```

```

</shutdownHooks>
<!-- The transport connectors ActiveMQ will listen to -->
<transportConnectors>
<transportConnector name="openwire" uri="tcp://0.0.0.0:61616"/>
<transportConnector name="stomp" uri="stomp://0.0.0.0:61613"/>
</transportConnectors>
</broker>
<bean id="activemqConnectionFactory" class="org.apache.activemq.ActiveMQConnectionFactory">
<property name="brokerURL" value="tcp://0.0.0.0:61616" />
</bean>
<bean id="pooledConnectionFactory" class="org.apache.activemq.pool.PooledConnectionFactory">
<property name="maxConnections" value="8" />
<property name="connectionFactory" ref="activemqConnectionFactory" />
</bean>
<bean id="resourceManager" class="org.apache.activemq.pool.ActiveMQResourceManager" init-method="recoverResource">
<property name="transactionManager" ref="transactionManager" />
<property name="connectionFactory" ref="activemqConnectionFactory" />
<property name="resourceName" value="activemq.ESB-MSH-01" />
</bean>
<osgi:reference id="transactionManager" interface="javax.transaction.TransactionManager" />
<osgi:service ref="pooledConnectionFactory">
<osgi:interfaces>
<value>javax.jms.ConnectionFactory</value>
</osgi:interfaces>
<osgi:service-properties>
<entry key="name" value="ESB-MSH-01"/>
</osgi:service-properties>
</osgi:service>
</beans>

```

Apache ServiceMix :: Bundles :: jdom (1.1.0.4)
Apache ServiceMix :: Bundles :: dom4j (1.6.1.5)
Apache MINA Core (2.0.5)
Apache CXF Compatibility Bundle Jar (2.7.3)
camel-core (2.10.4)
camel-karaf-commands (2.10.5.SNAPSHOT)
camel-spring (2.10.4)
camel-blueprint (2.10.4)
camel-cxf-transport (2.10.4)
camel-cxf (2.10.4)
Commons Pool (1.6.0)
camel-jms (2.10.4)
OPS4J Pax Web - Jsp Support (1.1.12)
Commons XPath (1.3)
Service Activity Monitoring :: Common (5.3.0)
Service Activity Monitoring :: Agent (5.3.0)
Talend :: ESB :: Job :: API (5.3.0)
Talend :: ESB :: LOCATOR :: AUTH (5.3.0)
Hosts: 194
ZooKeeper Bundle (3.3.6)
Fragments: 193

Service Locator Client for CXF (5.3.0)
Apache ServiceMix :: Bundles :: jaxen (1.1.1.2)
Talend :: ESB :: Job :: Controller (5.3.0)
Commons Net (3.1.0)
Apache ServiceMix :: Bundles :: jsch (0.1.48.1)
camel-ftp (2.10.4)
Camel Talend Job Component (5.3.0)
Jolokia Agent (1.0.4)
geronimo-j2ee-management_1.1_spec (1.0.1)
Commons Codec (1.4)
Apache ServiceMix :: Bundles :: velocity (1.7.0.4)
Apache ServiceMix Bundles: oro-2.0.8 (2.0.8.3)
Apache ServiceMix :: Bundles :: jasypt (1.9.0.1)
Apache ServiceMix :: Bundles :: xpp3 (1.1.0.4c_5)
Apache ServiceMix Bundles: dom4j-1.6.1 (1.6.1.2)
Apache ServiceMix Bundles: xstream-1.3 (1.3.0.3)
Apache Aries Transaction Manager (0.3.0)
activemq-core (5.7.0)
kahadb (5.7.0)
activemq-console (5.7.0)
activemq-ra (5.7.0)
activemq-pool (5.7.0)
activemq-karaf (5.7.0)
Apache XBean :: Spring (3.11.1)
activemq-spring (5.7.0)
activemq-web-console (5.7.0)
MyProducer0 (1.9)
MyConsumer1 (0.7)
MyProducer1 (0.3)
MyJob0 (2.2)
ProducerConsumer2 (3.5)
Consumer3 (1.2)
MyJob1 (0.1)
ProducerConsumer4 (2.1)
StoreErrors (0.1)
Apache ServiceMix :: Bundles :: derby (10.8.1.2_1)
Commons DBCP (1.4)
Service Activity Monitoring :: Datasource-derby (5.3.0)
Service Activity Monitoring :: Derby Starter (5.3.0)
Apache ServiceMix :: Bundles :: oscache (2.4.0.5)
Apache ServiceMix :: Bundles :: ibatis-sqlmap (2.3.4.726_4)
Induction_FillBWBContainer__PROD (2.2)
ESB-MSH (01.0.0.broker)