

# Dynamic throttling of an ActiveMQ message queue with Camel

Asked 15 years, 11 months ago   Modified 12 years, 8 months ago

Viewed 4k times



2

I am an ActiveMQ / Camel noob with a specific scenario in mind, I wonder firstly if it is possible and secondly whether someone might provide a little direction.



Basically I need to perform dynamic throttling off the queue. I.E the ability to set **at runtime** the rate a particular group of messages will be consumed from the queue.



So I might, for example, add a group of messages that are to be consumed at 10 per second, another group which should be consumed at 1 per second and so forth.

I know the basics of setting up routes in camel and message grouping onto the queue etc, but just can't figure this out from the docs.

java

jms

activemq-classic

apache-camel

Share

Improve this question

Follow

edited Apr 2, 2012 at 21:14



davemyron

2,542 ● 3 ● 24 ● 33

asked Jan 17, 2009 at 21:24



mysomic

1,556 ● 4 ● 21 ● 34

4 Answers

Sorted by:

Highest score (default)



4

You could just use Camel's existing [throttler](#) then using a different queue for each type of messages where you need to configure a different throttle rate?



e.g.



```
from("activemq:Queue1.Input").
    throttle(20).
    to("activemq:Queue1.Output");
from("activemq:Queue2.Input").
    throttle(5).
    to("activemq:Queue2.Output");
```

Share Improve this answer

answered Jan 19, 2009 at 12:38

Follow



James Strachan

9,188 ● 36 ● 31



3

Why don't you add a RFE in the Apache Camel JIRA?



What are your logic for determining the rate for a given group of messages?



If different group of messages passes throughout the same throttler it can get complex. It kinda need discriminator to determine any Message which group it



belongs to and thus which rate it should pass the throttler with.

If you take some time to fill in your use case and log the RFE then I am sure the Camel community, developers can help.

You can try to implement it yourself. Basically anything is a Processor, so you can do a  
`from("activemq:queue:foo").process(myOwnThrottler).to("bean:handleMessage");`

You can extend some of the classes in Camel: -  
DelegateProcessor - DelayProcessorSupport - Throttler

---

Claus Ibsen Apache Camel Committer

Open Source Integration: <http://fusesource.com> Blog:  
<http://davsclaus.blogspot.com/>

Share Improve this answer

answered Jan 18, 2009 at 9:08

Follow



Claus Ibsen

57.1k ● 7 ● 52 ● 66

- 
- 1 Just bought a EAP copy of your book "Camel in Action" Claus. It is excellent and I can definitely recommend it for anyone else getting to grips with, or wanting to learn more about Camel. Very in-depth stuff. – [mysomic](#) Sep 5, 2010 at 12:21
-



Yeah looks like you are looking for broker side throttling to avoid consumers to block.

1



Have you raised your request at the ActiveMQ user/dev forum?



Share Improve this answer

answered Feb 17, 2009 at 5:25



Follow



[Claus Ibsen](#)

57.1k ● 7 ● 52 ● 66



OK, I'll lay out the scenario in a bit more detail and highlight the main blocker as far as I can tell.

0



I have 2 groups of messages (in reality the scale will be much larger), each with a different throttling requirement - Let's say, I specify this in the message header as flowRate and flowTime.



- Group 1: FlowRate = 1; flowTime = 60 (1 per minute)
- Group 2: FlowRate = 1; flowTime = 1 (1 per second)

I implement a Processor as per [Claus](#) which examines the header fields and uses them as delayer input.

I add 20000 messages from group 1 and 20000 from group 2

Because the throttler is consumer side, the delayer activated by group 1 will cause it to become slow by

quickly filling its input buffer and group 2 messages will then become stuck...even if I use multiple queues as per [James](#).

I realise that I can group the message using the JMXGroupID header and implement multiple consumers, but don't think that would scale to the requirement for accommodating  $n$  groups.

What I was wondering, basically, is if there is any way to do broker, rather than consumer side throttling, or some other solution, whereby the consumer can throttle without ultimately blocking.

Hope I've explained myself clearly and thanks for the suggestions so far.

Share Improve this answer

Follow

edited May 23, 2017 at 12:11



Community Bot

1 • 1

answered Jan 20, 2009 at 21:24



mysomic

1,556 • 4 • 21 • 34