

Managing Slow Consumers

Overview

Slow consumers are consumers whose dispatch buffer is regularly too full; the broker cannot dispatch messages to them because they have reached the prefetch limit. This can bog down message processing in a number of ways and it can mask problems with a client. One of the major ways it can bog down a broker is by increasing its memory footprint by forcing the broker to hold a large number of messages in memory.

JBoss A-MQ provides two ways of limiting the impact of slow consumers:

- limiting the number of messages retained for a consumer

When using non-durable topics, you can specify the number of messages that a destination will hold for a consumer. Once the limit is reached, older messages are discarded when new messages arrive.

- aborting slow consumers

JBoss A-MQ determines slowness by monitoring how often a consumer's dispatch buffer is full. You can specify that consistently slow consumers be aborted by closing its connection to the broker.

Limiting message retention

Topics typically retain a copy of all unacknowledged messages for each of the consumers subscribed to it. For non-durable topics, the messages are stored in the broker's volatile memory, so if messages begin to pile up the broker's memory footprint begins to balloon.

To address this issue you can set the pending message limit strategy (`pendingMessageLimitStrategy`) on a topic to control the number of messages that are held for slow consumers. When set, the topic will retain the specified number of messages in addition to the consumer's prefetch limit.



Important

The default setting for the strategy is -1, which means that the topic will retain all of the unconsumed messages for a consumer.

There are two ways to configure the pending message limit strategy:

- specifying a constant number of messages over the prefetch limit

The `constantPendingMessageLimitStrategy` implementation allows you to specify constant number of messages to retain as shown in [Example 1 on page 2](#).

Example 1. Constant Pending Message Limiter

```
<broker ... >
  <destinationPolicy>
    <policyMap>
      <policyEntries>
        <policyEntry topic="" >
          <pendingMessageLimitStrategy>
            <constantPendingMessageLimitStrategy limit="50"/>
          </pendingMessageLimitStrategy>
        </policyEntry>
      </policyEntries>
    </policyMap>
  </destinationPolicy>
  ...
</broker>
```

- specifying a multiplier that is applied to the prefetch limit

The `prefetchRatePendingMessageLimitStrategy` implementation allows you to specify a multiplier that is applied to the prefetch limit. [Example 2 on page 2](#) shown configuration that retains twice the prefetch limit. So if the prefetch limit is 3, the destination will retain 6 pending messages for each consumer.

Example 2. Prefetch Limit Based Pending Message Limiter

```
<broker ... >
  <destinationPolicy>
    <policyMap>
      <policyEntries>
        <policyEntry topic="" >
          <pendingMessageLimitStrategy>
            <prefetchRatePendingMessageLimitStrategy multiplier="2"/>
          </pendingMessageLimitStrategy>
        </policyEntry>
      </policyEntries>
    </policyMap>
  </destinationPolicy>
  ...
</broker>
```

Aborting slow consumers

Another strategy for managing slow consumers is to have the broker detect slow consumers and automatically abort consumers that are consistently slow. When a slow consumer is aborted, its connection to the broker is closed.

The broker marks a consumer slow when the broker has messages to dispatch to the consumer, but the consumer's prefetch buffer is full. As the consumer acknowledges consumption of messages from the prefetch buffer and the broker can once again start dispatching messages to the consumer, the broker will stop considering the consumer slow. If the consumer's prefetch buffer fills up again, the broker will again mark the consumer as slow.

The abort slow consumers strategy allows the broker to abort consumers when one of two conditions is met:

- a consumer is considered slow for specified amount of time
- a consumer is considered slow a specified number of times

The abort slow consumer strategy is activated by adding the configuration shown in [Example 3 on page 4](#) to a destination's configuration.

Example 3. Aborting Slow Consumers

```
<broker ... >
...
<destinationPolicy>
  <policyMap>
    <policyEntries>
      <policyEntry topic=">" >
        <slowConsumerStrategy>
          <abortSlowConsumerStrategy />
        </slowConsumerStrategy>
      </policyEntry>
    </policyEntries>
  </policyMap>
</destinationPolicy>
...
</broker>
```

The `abortSlowConsumerStrategy` element activates the abort slow consumer strategy with default settings. Consumers that are considered slow for more than 30 seconds are aborted. You can modify when slow consumers are aborted using the attributes described in [Table 1 on page 4](#).

Table 1. Settings for Abort Slow Consumer Strategy

Attribute	Default	Description
maxSlowCount	-1	Specifies the number of times a consumer can be considered slow before it is aborted. -1 specifies that a consumer can be considered slow an infinite number of times.
maxSlowDuration	30000	Specifies the maximum amount of time, in milliseconds, that a consumer can be continuously slow before it is aborted.
checkPeriod	30000	Specifies, in milliseconds, the time between checks for slow consumers.

For example, [Example 3 on page 4](#) shows configuration for aborting consumers that have been marked as slow 30 times.

Example 4. Aborting Repeatedly Slow Consumers

```
<abortSlowConsumerStrategy maxSlowCount="30" />
```