Why is my ActiveMQ/JMS MDB not scaling as expected?

Why my ActiveMQ/JMS MDB is not scaling as expected?

There are multiple configurations points to ensure you scale as much as you want.

Here some common configuration to validate (note that when possible the sample value is the default):

- The resource adapter thread pool ("worker threads" or `WorkManager`) limits the number of max work threads:

```
<Resource id="my resource adapter" ....>
  # using -1 will make the server using cached threads (unbounded)
  # min recommanded: maxSessions + 1 (for connect())
  threadPoolSize = 30
</Resource>
```

- Then the MDB container itself has a upper bound through `InstanceLimit` which controls the max MDB instance count:

```
<Container id="my mdb container" type="MESSAGE">
    # -1 will disable it
    # min recommanded = maxSessions
    InstanceLimit = 10
</Container>
```

- ActiveMQ `maxSessions` controls how many sessions a MDB can use:

```
@MessageDriven(activationConfig = {
        @javax.ejb.ActivationConfigProperty(propertyName = "maxSessions",
propertyValue = "1"),
        @javax.ejb.ActivationConfigProperty(propertyName = "destination",
propertyValue = "target-queue")
})
public static class MyMdb implements MessageListener {
    @Override
    public void onMessage(final Message message) {
        // ...
    }
}
```

- The ConnectionFactory has also an instance pool through geronimo-connector logic, configuration can make the behavior changing but this is controlled through pool related variables (the pool and resource properties are merged in the definition):

```
<Resource id="my connection factory" type="ConnectionFactory">
    # recommanded to be aligned on maxSessions
    PoolMaxSize = 10
    # for 100% MDB (no client) you can evaluate to turn it off/false, for client it
can still be useful depending what you do
    Pooling = true
</Resource>
```

Slow Message Consumption

If you find you have a slow consumption of messages there are several options to have a look (activemq website explains it very well) but one very impacting option can be the prefetch size.