Testing Active MQ broker - report

Wojciech Czech

July 30, 2007

Introduction

This document presents results of experiments on Active MQ massage broker. The aim of our tests was to evaluate capabilities of broker in different communication scenarios. The main subject of our interests is number of messages broker can forward per second. We also try to investigate how this measure changes with increasing number of parallel producers and consumers or message topics.

Experiments

The tested communication is asynchronous, without subscriber acknowledgments, receipts and persistent messages.

Software

The testing framework consists of several elements:

- AMQ message broker 4.1.1 (stable) running inside Tomcat 5.5.
- Simple STOMP publisher, written as shell script using `netcat` program to write STOMP messages directly to open socket of AMQ STOMP connector.
- Simple STOMP subscriber written with the use of `netcat` program, getting messages directly from STOMP connector.
- STOMP publisher written in Python which parses gFTP logs and sends messages created on this basis to specified topic.
- STOMP subscriber written in Python which get get messages from any topic.
- Message summarizer (Java, JMS API) which consumes messages from given topic and counts how many messages it received in given period (e.g., per second), then it publishes this information on given topic. This program allows us to measure performance of broker in different configurations. We can also use the variant of summarizer which performs a little bit more processing - it consumes gFTP logs and counts number of bytes forwarded by given gFTP server within specified period.

Hardware

- `lxb6117`: two CPUs Intel Xeon 2.80GHz, 2GB RAM
- `lxb6118`: two CPUs Intel Xeon 2.80GHz, 2GB RAM
- *nemrod*: Intel Core Duo 2 1.83GHz, 2GB RAM
- `lxplus096`: two CPUs Intel Xeon 2.80GHz, 2GB RAM
- `lxplus097`: two CPUs Intel Xeon 2.80GHz, 2GB RAM
- `lxb6117` connected to `lxb6118` via Gigabit Ethernet

Results

Test 1a

Configuration

- `lxb6117` - broker
- `lxb6118` - netcat producer, prepared set of 1048578 39B messages, sending time measured with the use of `time` program
- `nemrod` - netcat consumer
- message size: 39B

Results

- average CPU load (broker process): 101%
- average MEM usage (broker process): 19.6%
- messages per second: 18489.16 (producer)

Test 1b

Configuration

- `lxb6117` - broker
- `lxb6118` - netcat producer, prepared set of 3524578 1KB messages, sending time measured with the use of `time` program
- `nemrod` - netcat consumer
- message size: 1KB

Results

- average CPU load (broker process): 101%
- average MEM usage (broker process): 20.7%
- messages per second: 18108.19 (producer)

Test 1c

Configuration

- `lxb6117` - broker
- `lxb6118` - netcat producer, prepared set of 40357 10KB messages, sending time measured with the use of `time` program, netcat consumer
- message size: 10KB

Results

- average CPU load (broker process): 128%
- average MEM usage (broker process): 20.6%
- messages per second: 1841.10 (producer)

Test 2

Configuration

- `lxb6117` - broker
- `lxb6118` - Java summarizer (counts number of messages caught per second), netcat producer, python consumer (gets messages published by summarizer)
- message size: 39B

Results

- average CPU load (broker process): 117%
- average MEM usage (broker process): 19.6%
- average CPU load (summarizer process): 63%
- average MEM usage (summarizer process): 1.4%
- summarizer gets approximately 9500 messages per second

Test 3

Configuration

- `lxb6117` - broker
- `lxb6118` - Java summarizer, 10 netcat producers, python consumer (gets messages published by summarizer)
- message size: 39B

Results

- average CPU load (broker process): 165%
- average MEM usage (broker process): 19.6%
- average CPU load (summarizer process): 70%
- average MEM usage (summarizer process): 1.4%
- summarizer gets approximately 12000 messages per second

Test 4

Configuration

- `lxb6117` - broker
- `lxb6118` - Java summarizer, 100 netcat producers, python consumer (gets messages published by summarizer)
- message size: 39B

Results

- average CPU load (broker process): 169%
- average MEM usage (broker process): 20.7%
- average CPU load (summarizer process): 75%
- average MEM usage (summarizer process): 2.6%
- summarizer gets approximately 12000 messages per second

Test 5

- `lxb6117` - broker
- `lxb6118` - Java summarizer, python producer, python consumer (gets messages published by summarizer)
- `nemrod` - python producer
- message size: 39B

Results

- average CPU load (broker process): 191%
- average MEM usage (broker process): 20.7%
- average CPU load (summarizer process): 67%
- average MEM usage (summarizer process): 1.4%
- summarizer gets approximately 12500 messages per second

Test 6

- `lxb6117` - broker
- `lxb6118` - Java summarizer, 20 python producers, python consumer (gets messages published by summarizer)
- `nemrod` - 2 python producers
- message size: 39B

Results

- average CPU load (broker process): 141%
- average MEM usage (broker process): 20.7%
- average CPU load (summarizer process): 47%
- average MEM usage (summarizer process): 1.6%
- summarizer gets approximately 9000 - 11000 messages per second with short, periodic decreases to 3000 msg/s

Test 7

- `lxb6117` - broker
- `lxb6118` - Java summarizer, 20 python producers, python consumer (gets messages published by summarizer)
- `nemrod` - 20 python producers
- message size: 39B

Results

- average CPU load (broker process): 115%
- average MEM usage (broker process): 20.7%
- average CPU load (summarizer process): 45%
- average MEM usage (summarizer process): 4%
- summarizer gets approximately 5000 - 11800 messages per second, strong fluctuations

Test 8

- `lxb6117` - broker
- `lxb6118` - netcat producer, 2 netcat consumers
- `nemrod` - 2 netcat consumers
- message size: 39B

Results

- average CPU load (broker process): 101%
- average MEM usage (broker process): 20.7%
- producer sends 18483.62 msgs/s (1048576 messages in 56.733s)

Test 9

- `lxb6117` - broker
- `lxb6118` - Java summarizer, python consumer (gets messages published by summarizer)
- `lxplus96` - python consumer
- `lxplus97` - python consumer
- message size: 39B

Results

- average CPU load (broker process): 30%
- average MEM usage (broker process): 20.6%
- summarizer gets at the beginning approximately 6000 messages per second, after a while the efficiency decreases to approximately 400 msgs/s

Test 9

- `lxb6117` - broker
- `lxb6118` - Java summarizer, python consumer (gets messages published by summarizer)
- `lxplus96` - python consumer
- `lxplus97` - python consumer
- message size: 39B

Results

- average CPU load (broker process): 30%
- average MEM usage (broker process): 20.6%
- summarizer gets at the beginning approximately 6000 messages per second, after a while the efficiency decreases to approximately 400 msgs/s

Test 10

- `lxb6117` - broker
- `lxb6118` - Java summarizer, python consumer (gets messages published by summarizer)
- 4 different topics
- 4 pairs producer/consumer: (`lxb6118` - Java summarizer, `nemrod` - netcat), (`lxplus97` python producer, `lxplus96` python consumer), (`lxplus96` python producer, `lxplus97` python consumer), (`nemrod` python producer, `lxb6118` python consumer)

- message size: 39B

Results

- average CPU load (broker process): 60%
- average MEM usage (broker process): 20.7%
- summarizer gets at the beginning approximately 6000 messages per second, after a while the efficiency decreases to approximately 400 msgs/s

Summary

- AMQ broker provides fast message forwarding with approximate rate 10000 msg/s
- In case of multiple subscriptions or multiple topics the efficiency is worse (in proportion to number of subscribers). Differences in speed of subscribers may result in asymmetric transfer rates.