Performance Benchmarking an Enterprise Message Bus

Anurag Sharma
Pramod Sharma
Sumant Vashisth
About the Authors...

Sumant Vashisth is Director of Engineering, Security Management Business Unit at McAfee. Besides, managing and leading teams across various industries during his long career, Sumant is a techie at heart. You can often see him chatting with engineers about the latest in technology and day to day technical challenges.

Pramod Sharma is a Software Architect at McAfee, with more than ten years of industry experience. An acknowledged technologist, Pramod has eight patent pending inventions to his credit. Passionate about software quality and process improvements, Pramod is a vocal champion of adopting changes, and learning from the industry. His interests span security management for embedded and mobile devices, and scalable architectures for distributed systems.

Anurag Sharma is a Principal SDET Engineer at McAfee, with around eight years of industry experience in software testing and quality assurance. He enjoys reading and being up to date on latest technologies.
<table>
<thead>
<tr>
<th>1. Message Bus Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Few Ground Rules</td>
</tr>
<tr>
<td>3. Motivation for this Paper</td>
</tr>
<tr>
<td>4. Key things about measuring message bus performance</td>
</tr>
<tr>
<td>5. Performance categories/factors</td>
</tr>
<tr>
<td>6. Final Notes</td>
</tr>
<tr>
<td>7. McAfee Message Bus Architecture</td>
</tr>
<tr>
<td>8. Q&amp;A</td>
</tr>
</tbody>
</table>
What the hell is this message (service) bus...

- Lightweight and loosely coupled framework.
- Integration across a broad range of enterprise applications across security firewalls, application protocols and languages (messaging backbone).
- Guaranteed delivery of messages among services performing security checks and protocol and data conversions on the fly.
- Fault tolerant and fail-safe
- Ease of development of services
- Ease of deployment
- Features for message manipulation
- Transactional processing

Message Bus Introduction

Cont..

Resolve contention between communicating service components

Control deployment and versioning of services

Cater for commodity services like event handling, data transformation and mapping, message and event queuing and sequencing

Conforms to Advanced Message Queuing Protocol (AMQP) - an open standard application layer protocol for message oriented middleware

What this presentation is not about?

- We are not going to discuss the message bus architecture and design.

- We are not going to talk about numbers (performance data) here.... *sorry to disappoint any mathematicians in the hall.*

- We are not going to prove that McAfee enterprise message bus is superior to all other message buses out there... *although, it is 😊.*
Performance data generated by the different vendors or projects is not consistent – really hard to compare and evaluate.

Incomplete testing is performed just to solve the immediate needs.

Performance tests are too far fetched that they just seem too good to be practical.

Key things to be considered while evaluating performance

• Benchmarks standards and tests should measure the performance of the architecture, as a whole, not the various components in it.

• Benchmark standards should not be implementation specific i.e. based on any particular platform, software, language of implementation, topology etc.

• Benchmark tests and results should not be biased by application domain, any particular feature etc.

• Tests should be reproducible, scalable and exhaustive.
Benchmark Categories – Machine Configuration

- Set of three configurations – one high end (best), one average and one low-end (worst) configuration and execute the benchmark test on all three configurations.

- For out-of-machine (network) communication - network configuration parameters like network topology, the network bandwidth etc. should be identified in advance and documented.

- Platform parameters like – processor (speed and number of cores), operating system, architecture, physical memory etc. should also be documented.

- No other processor or network hogging tasks should be running while the benchmark tests are conducted.
Benchmark Categories - Communication Modes

Client-Server or Request-Response mode - Synchronously or Asynchronously.

Publisher-Subscriber mode – In this mode, the subscribers for some particular “Topic” of interest and then, gets notified when topic of interest is available on the bus.
Message delivery latency, throughput and reliability vary depending on the relative location of client and server.
Any messaging architecture depends on the performance of one or two entities in the architecture for example, the broker or the packet forwarder or the connection manager.

These entities drive the limits of message bus scalability and throughput.

Design test scenarios stressing/loading this component.
In the out of machine tests, the machines or nodes which are distant (IP hops wise, not physically), should be considered for communication.

In-the-process communication configuration is not a good choice for latency tests.

The server or receiving node should perform some minimal processing before sending the response.

Gradually, increase the number of simultaneous connections and message size and record the average time taken for response.
Benchmark Categories – Throughput

Throughput is the measure of the average number of successful messages delivered and responses received over a communication channel.

The throughput is an important factor while designing message bus or choosing any open source message bus for enterprise usage.

Less throughput can be unsuitable for some highly data intensive networks.
The standard machine configuration should be used while designing and executing the throughput tests.

More number of threads spawned at the client and server sides can also increase the throughput, therefore, the throughput values should also correlate with the number of parallel threads of execution.

Some processing at the receiving node should also be taken into account while measuring the throughput.
Benchmark Categories - Thread Pool v/s Thread IO

Thread Pool - Number of threads are created to perform a set of tasks. Typically, more tasks than threads.

Thread IO - A single worker thread performs the tasks sequentially.

Thread pool provides better performance and throughput compared to creating a new thread for each task.

Thread pool, the number of threads in a thread pool is a parameter that can be tuned to provide the best performance.

Benchmark tests should be conducted for both configurations.
Benchmark Categories – Reliability

Reliability in this context is measured in terms of number of packets lost in a given period of time.

Example scenarios -

Push the messages continuously without delay and determine the number of messages dropped (Monitor CPU usage, memory utilization, crashes etc.).

Run the tests for a two to three days continuously and monitor the messages dropped and other vital characteristics.

Varying the network configuration i.e., topology and number of processes/machines communicating, can also generate significant reliability information.
Benchmark Categories – Resource Utilization

- **CPU utilization** – By general principle, more than 50% CPU utilization is not considered good.

- **Memory usage** – Should be kept to minimum.

- **Number of threads** – Always a tradeoff between the throughput and reliability.

- **Disk I/O** – Disk operations are expensive and increase latency. They should be kept to minimum.

- **Number of open sockets/pipes** – Expensive and potential security threats. Try to keep it to minimum. Consider intelligent designs concepts like reusing the sockets/pipes, opening connections on demand etc.

*The ability of software to use an appropriate quantity of resource of certain type.*
• Most message bus architectures deploy some level of security either at message level or at connection level.

• Various security mechanisms like connection based authentication (connection based or message based), encryption, integrity check (hash algorithm), authorization etc., slow down the message bus at varying levels.

• Benchmark tests should be conducted for both secure and non-secure environments.
Throughput, reliability, latency are not independent factors, they should all be considered together while designing the test scenarios.

The performance test scenarios are also end-to-end tests and a great aid for performance fine-tuning.

Performance test scenarios should generally be automated.
McAfee Message Bus Architecture
Few Popular Enterprise Message Buses

- ZeroMQ (Brokerless model)
- ActiveMQ
- RabbitMQ
- Qpid
- HornetMQ
- Apache Qpid
- Apollo
- Avis Event Router (Only PUB-SUB mode)
- D-Bus (Open Source)
- MSMQ
- RavenMQ
- Jboss ESB
References

- Zero MQ performance results at - http://www.zeromq.org/area:results

Image credit – Images are borrowed from Google Image Search
Questions are guaranteed in life; Answers aren’t.
Thanks...