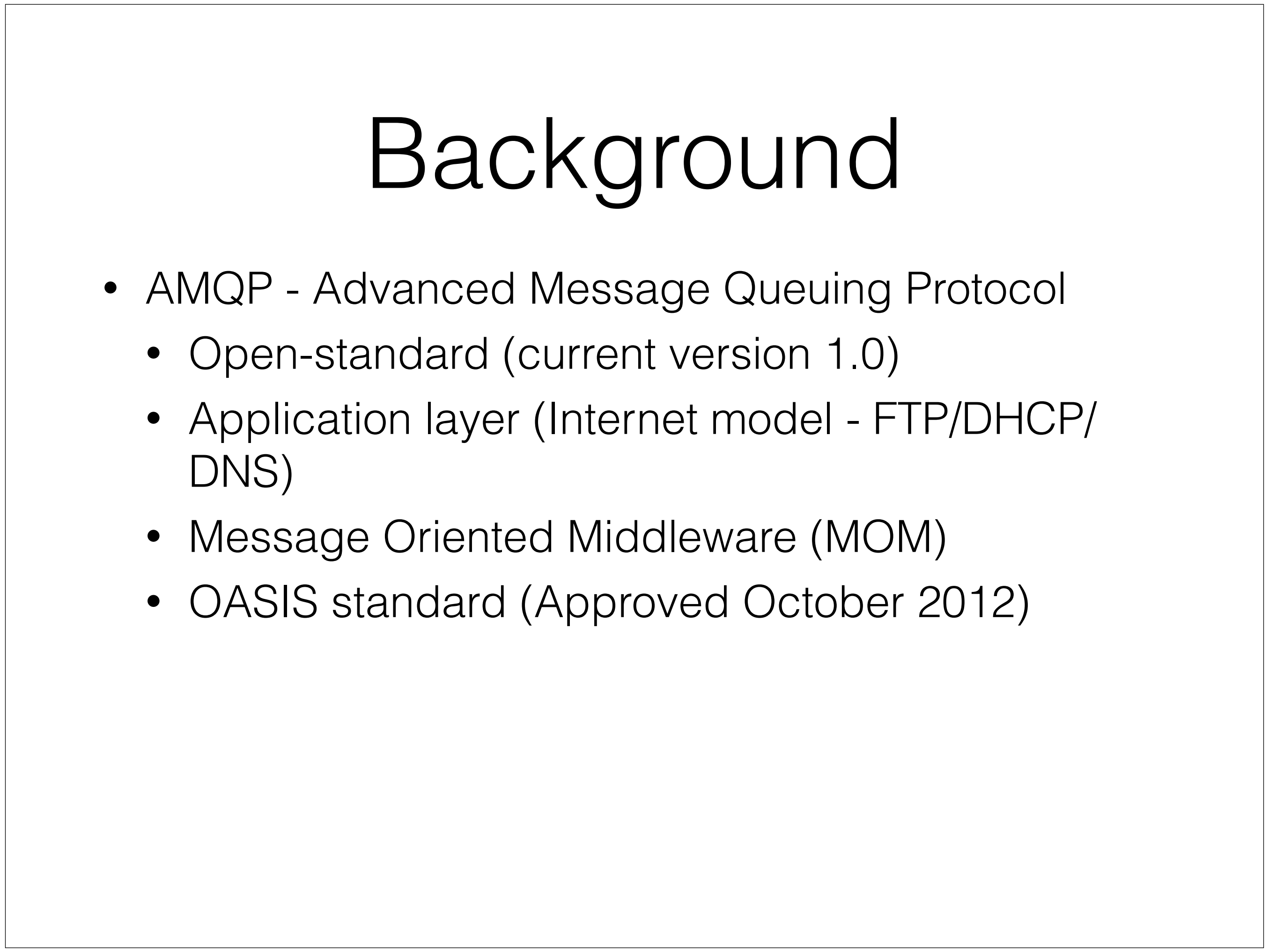
Federating AMQP 1.0 message brokers

David Joe Wade
4/17/2014



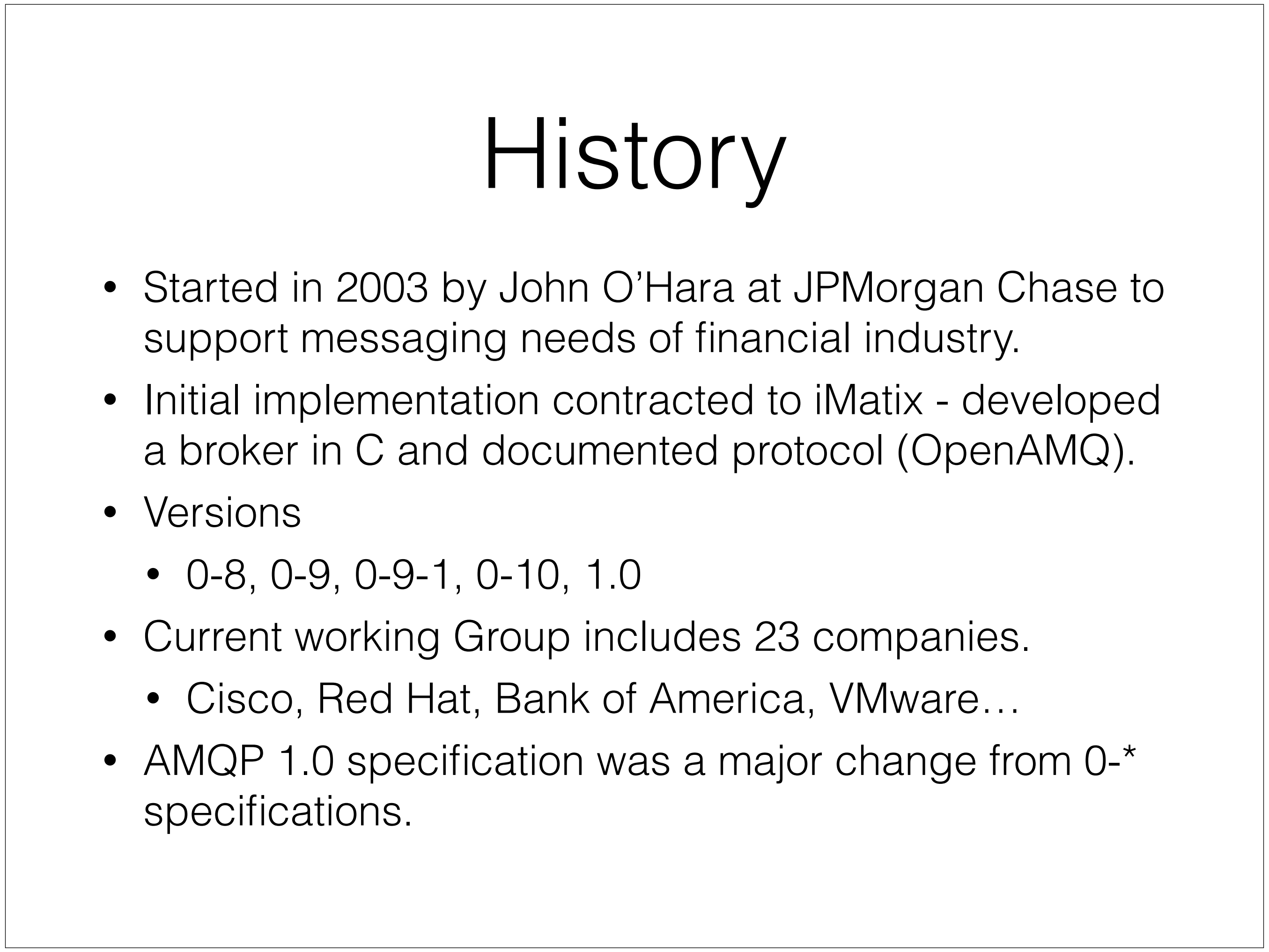
Background

- AMQP - Advanced Message Queuing Protocol
  - Open-standard (current version 1.0)
  - Application layer (Internet model - FTP/DHCP/DNS)
  - Message Oriented Middleware (MOM)
  - OASIS standard (Approved October 2012)

Internet model

Link, Internet, Transport, Application

https://www.oasis-open.org/news/pr/amqp-1-0-approval



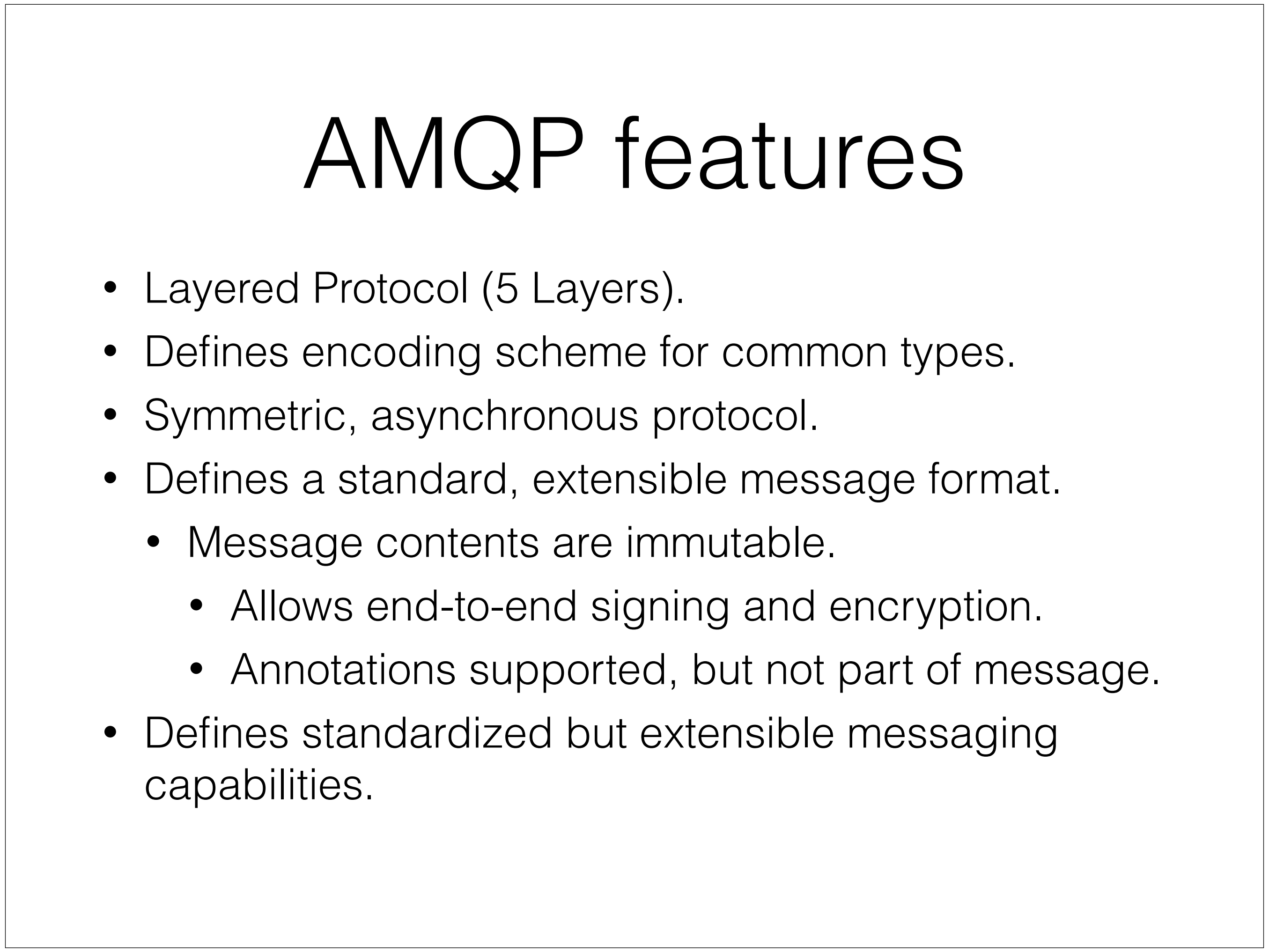
History

- Started in 2003 by John O’Hara at JPMorgan Chase to support messaging needs of financial industry.
- Initial implementation contracted to iMatix - developed a broker in C and documented protocol (OpenAMQ).
- Versions
  - 0-8, 0-9, 0-9-1, 0-10, 1.0
- Current working Group includes 23 companies.
  - Cisco, Red Hat, Bank of America, VMware…
- AMQP 1.0 specification was a major change from 0-* specifications.

0-8 - June 2006
0-9 - December 2006
0-10 February 2008
0-9-1 November 2008

OASIS member - August 2011
1.0 - 30 October 2011
1.0 - First draft - February 2012)
1.0 - Second draft - October 2012

Other working group members
Bank of America
Barclays
Cisco Systems
Credit Suisse
Deutsche Börse Systems
Goldman Sachs
JPMorgan Chase
Microsoft Corporation
Novell
Red Hat
VMware (which acquired Rabbit Technologies)



AMQP features

- Layered Protocol (5 Layers).
- Defines encoding scheme for common types.
- Symmetric, asynchronous protocol.
- Defines a standard, extensible message format.
  - Message contents are immutable.
    - Allows end-to-end signing and encryption.
    - Annotations supported, but not part of message.
- Defines standardized but extensible messaging capabilities.

AMQP1.0 - supports peer-to-peer

Architecture/Terminology

Term	Concept	Example
Exchange	Broker connection point (location)	An airport (Bradley)
Queue/Topic	Message destination	RPI
Binding	Rule that determines path of message from enhance to queue	Flight route to Bradley
Virtual Hosts	Isolated instance of exchanges/queues/bindings	Virtual machine?
Connections	Network connection (TCP session)	POTS Circuit
Channels	Pooled path over connection	Connection pool

AMQP Message brokers

AMQP Brokers

	Version	License	Language	Release date	AMQP version
Apache ActiveMQ	5.9.0	ASL2.0	Java	October 2013	1.0
Apache Apollo	1.7.0	ASL2.0	Java/Scala	February 2014	1.0
Apache Qpid	0.26	ASL2.0	Java	February 2014	1.0
Apache Qpid	0.26	ASL2.0	C++	February 2014	1.0
RabbitMQ	3.2.4	Mozilla 1.1	Erlang	March 2014	**0-9-1**
SwiftMQ	9.5.0	Proprietary	**Unknown**	**Unknown**	1.0
HornetQ	2.4.0	ASL2.0/ LGPL	Java	December 2013	1.0
StormMQ	2010.05. 20	Client - Mozilla 1.1	Java	May, 2010	1.0
Microsoft Service Bus	Unknown	Proprietary	Unknown	Unknown	1.0

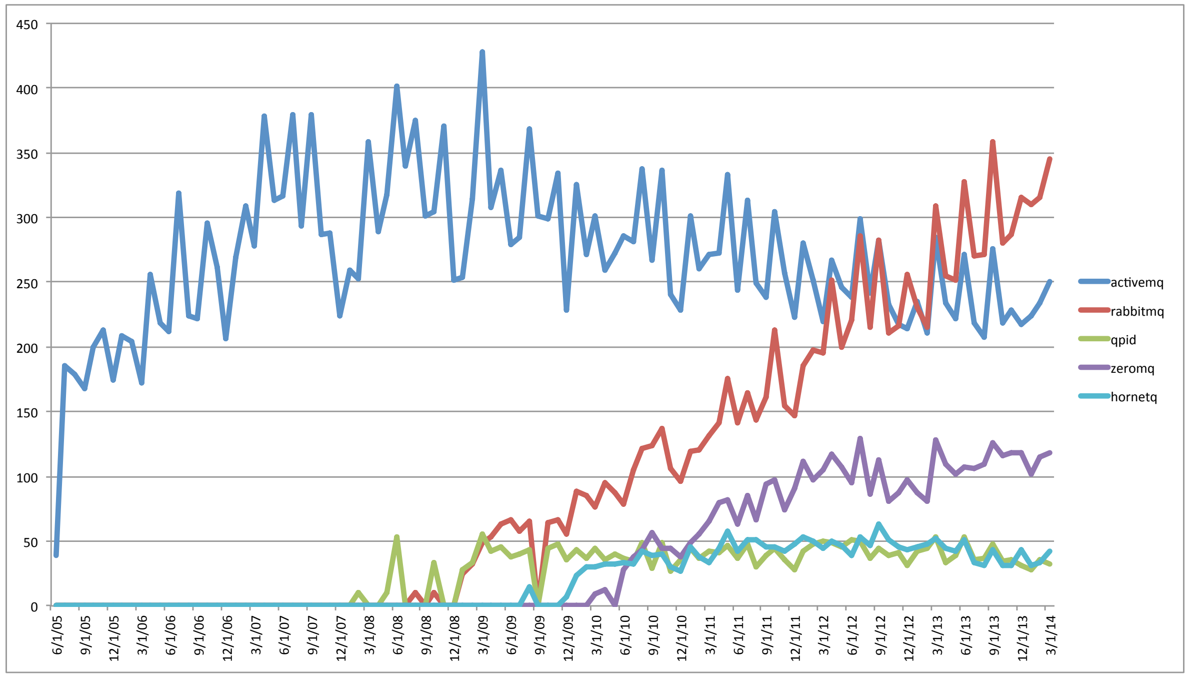
AMQP Clients

Implementation		AMQP Versions				Language bindings			
	0-8	0-9	0-9-1	0-10	1.0	Java	C/C++	.NET	Python
Qpid-JMS	✔	✔	✔	✔	✔	✔			
QPID Proton	✔	✔	✔	✔	✔	✔	✔		
RabbitMQ	✔	✔	✔			✔	✔	✔	✔
.NET Service Bus					✔			✔	
SwiftMQ					✔	✔		✔	

This list is certainly not exhaustive - RabbitMQ lists 175+ RabbitMQ clients in 21 different programming languages.

Most broker implementations that support AMQP 1.0 seem to use the Qpid Proton libraries

Google search popularity

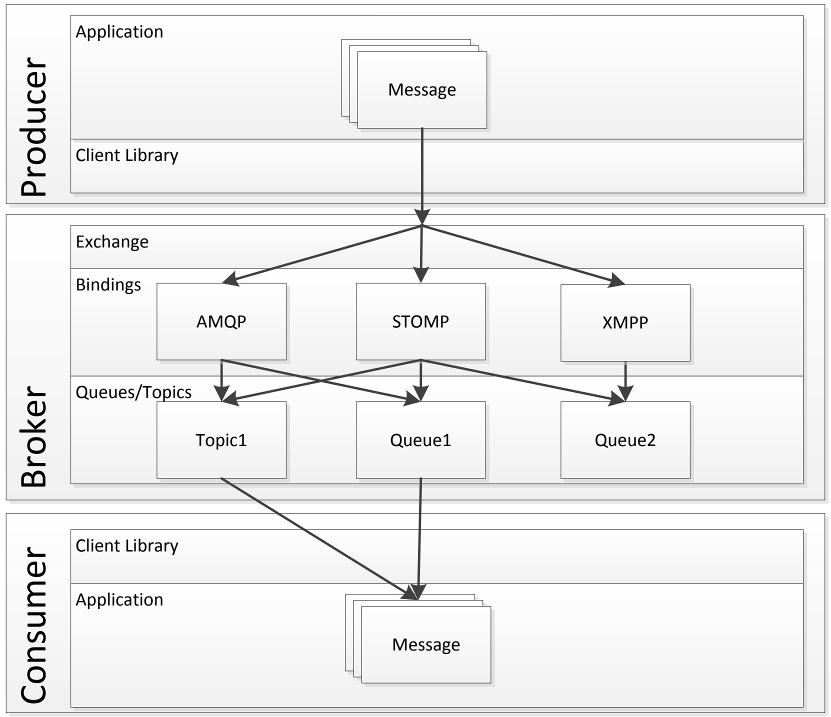


data from: http://www.google.com/trends/explore#q=activemq%2C%20rabbitmq%2C%20qpid%2C%20hornetq%2C%20Apache%20Apollo&cmpt=q

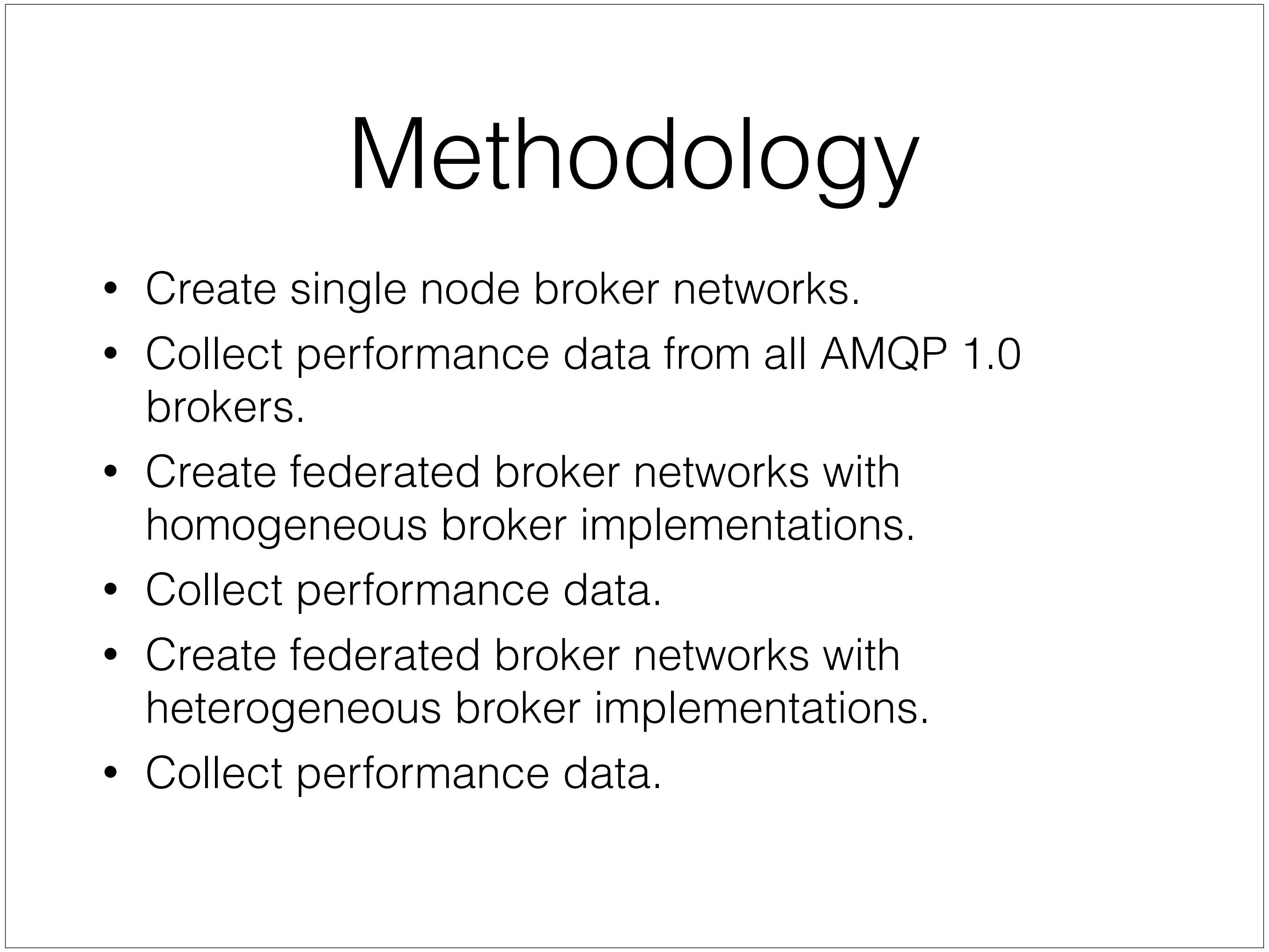
Broker features

- Automatic reconnect.
- Failover/High Availability.
- Persistence.
- Clustering.
- Federation.

Broker Architecture



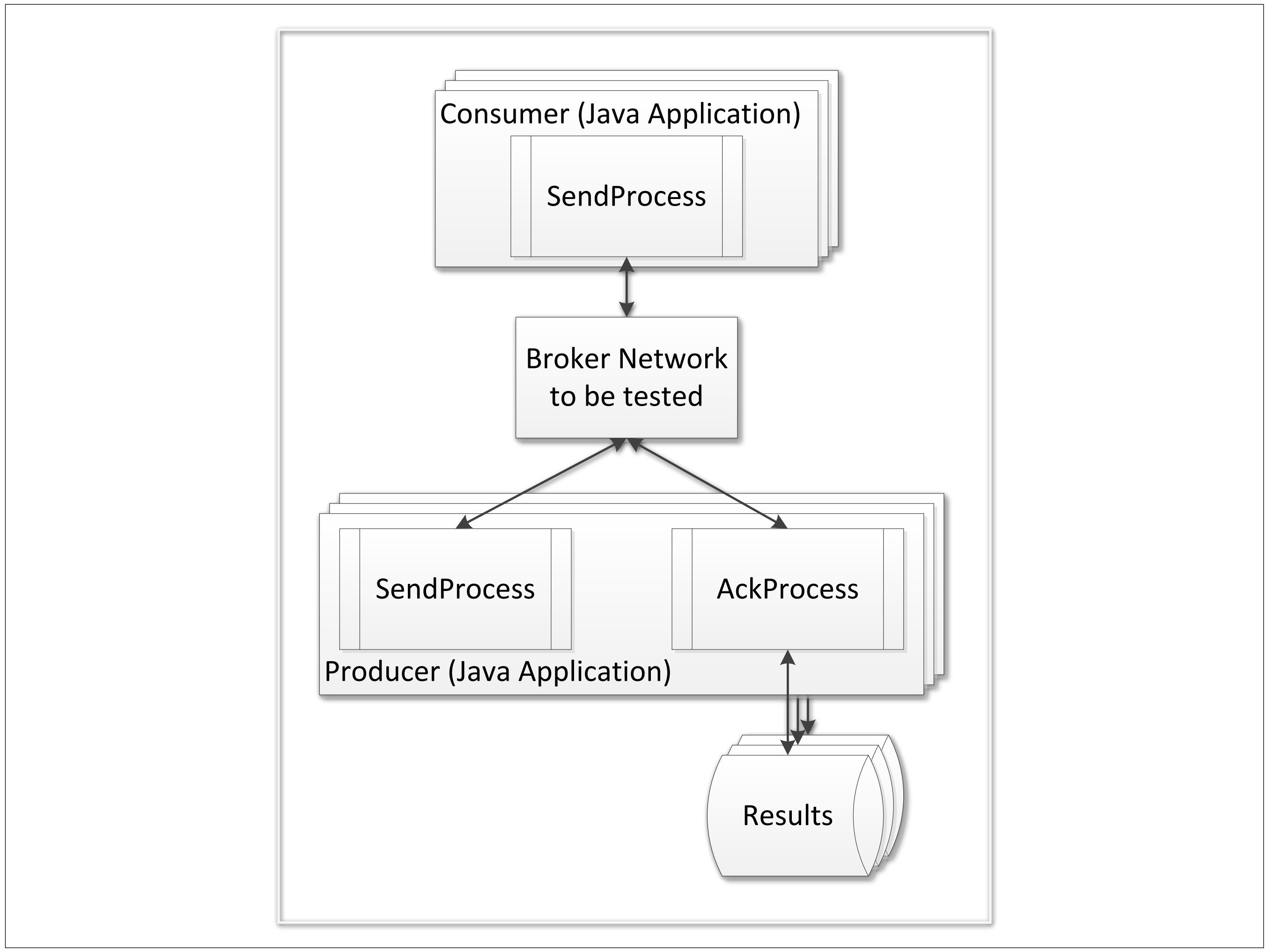
Project design



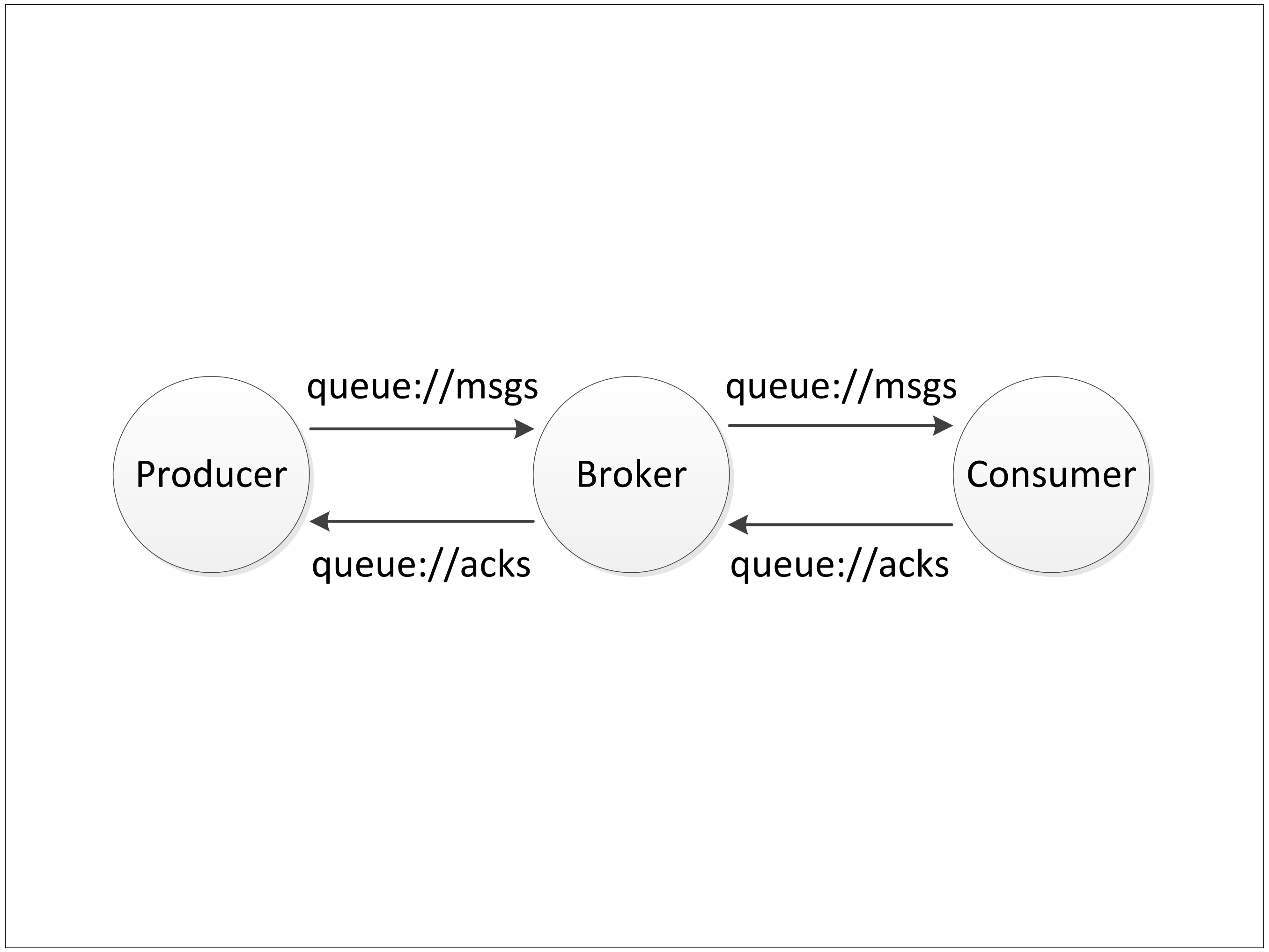
Methodology

- Create single node broker networks.
- Collect performance data from all AMQP 1.0 brokers.
- Create federated broker networks with homogeneous broker implementations.
- Collect performance data.
- Create federated broker networks with heterogeneous broker implementations.
- Collect performance data.

Vary message size (100, 500, 1000, 5000, 10000, 50000, 100000)


Consumer (Java Application)
SendProcess
Broker Network to be tested
SendProcess
AckProcess
Producer (Java Application)
Results

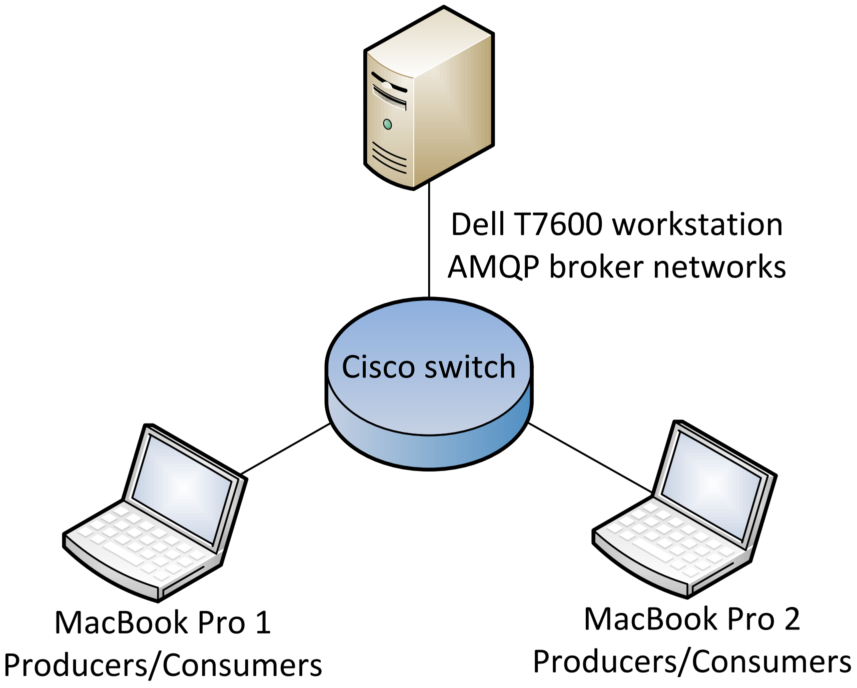
Producer keeping track of results to make supporting multiple clients easier.



AMQP channels are one way - to get keep a single repository for results two separate channels are created.

One Way latency (producer -> consumer)

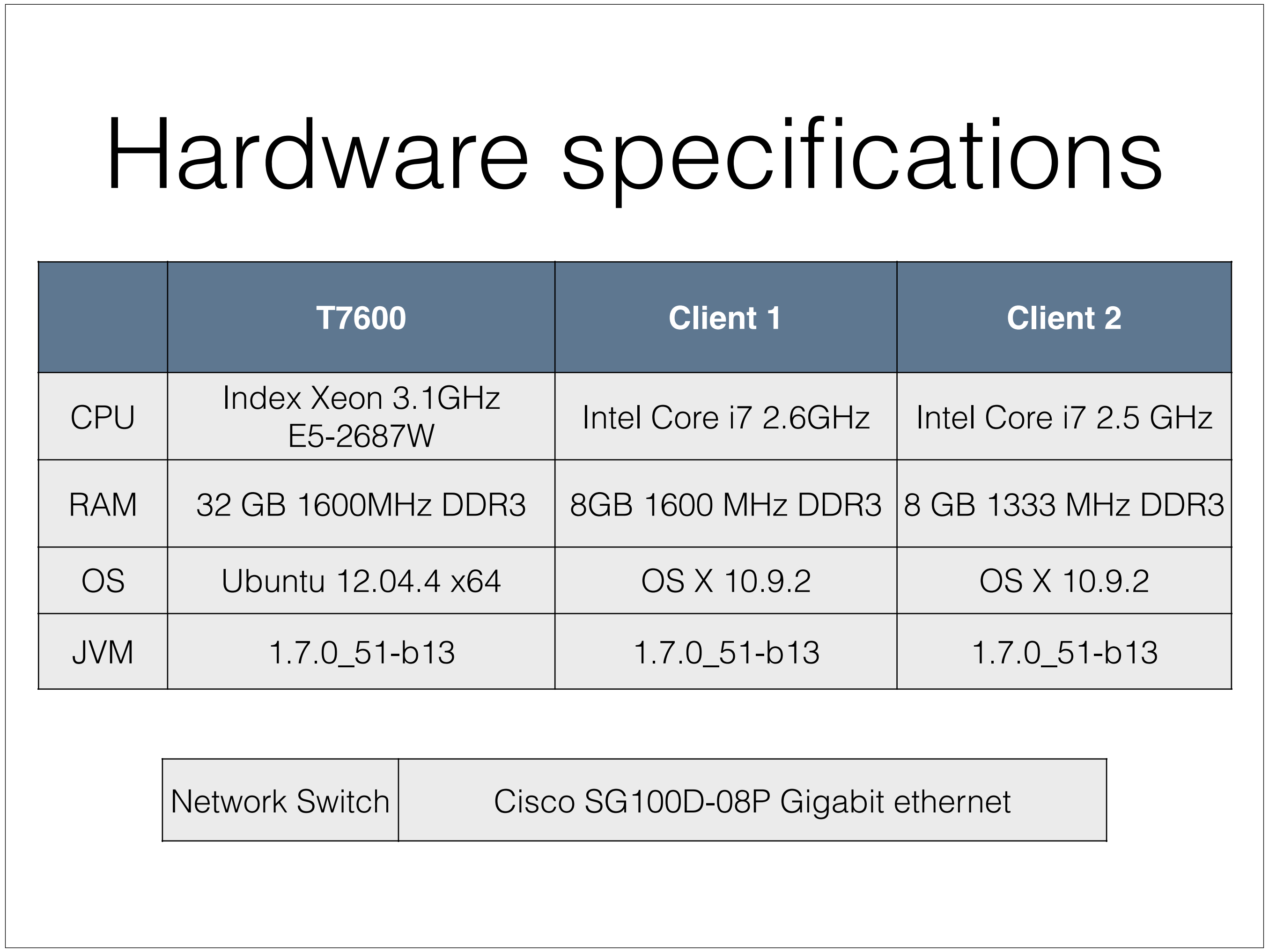
Architecture



Dell T7600 workstation
AMQP broker networks

MacBook Pro 1
Producers/Consumers

MacBook Pro 2
Producers/Consumers



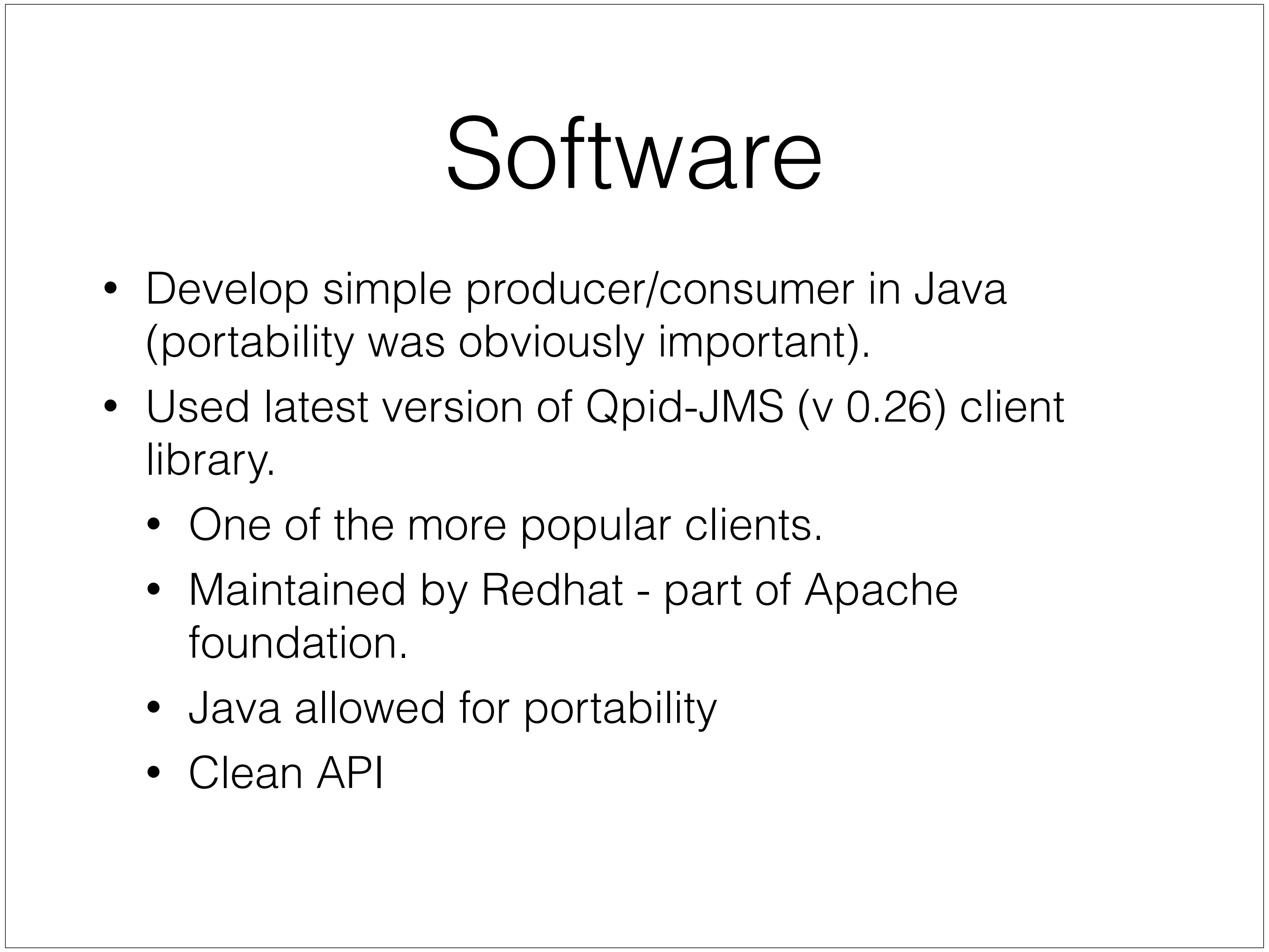
Hardware specifications

	T7600	Client 1	Client 2
CPU	Index Xeon 3.1GHz E5-2687W	Intel Core i7 2.6GHz	Intel Core i7 2.5 GHz
RAM	32 GB 1600MHz DDR3	8GB 1600 MHz DDR3	8 GB 1333 MHz DDR3
OS	Ubuntu 12.04.4 x64	OS X 10.9.2	OS X 10.9.2
JVM	1.7.0_51-b13	1.7.0_51-b13	1.7.0_51-b13

Network Switch	Cisco SG100D-08P Gigabit ethernet

Cisco SG100D-08P switch

Project Software



Software

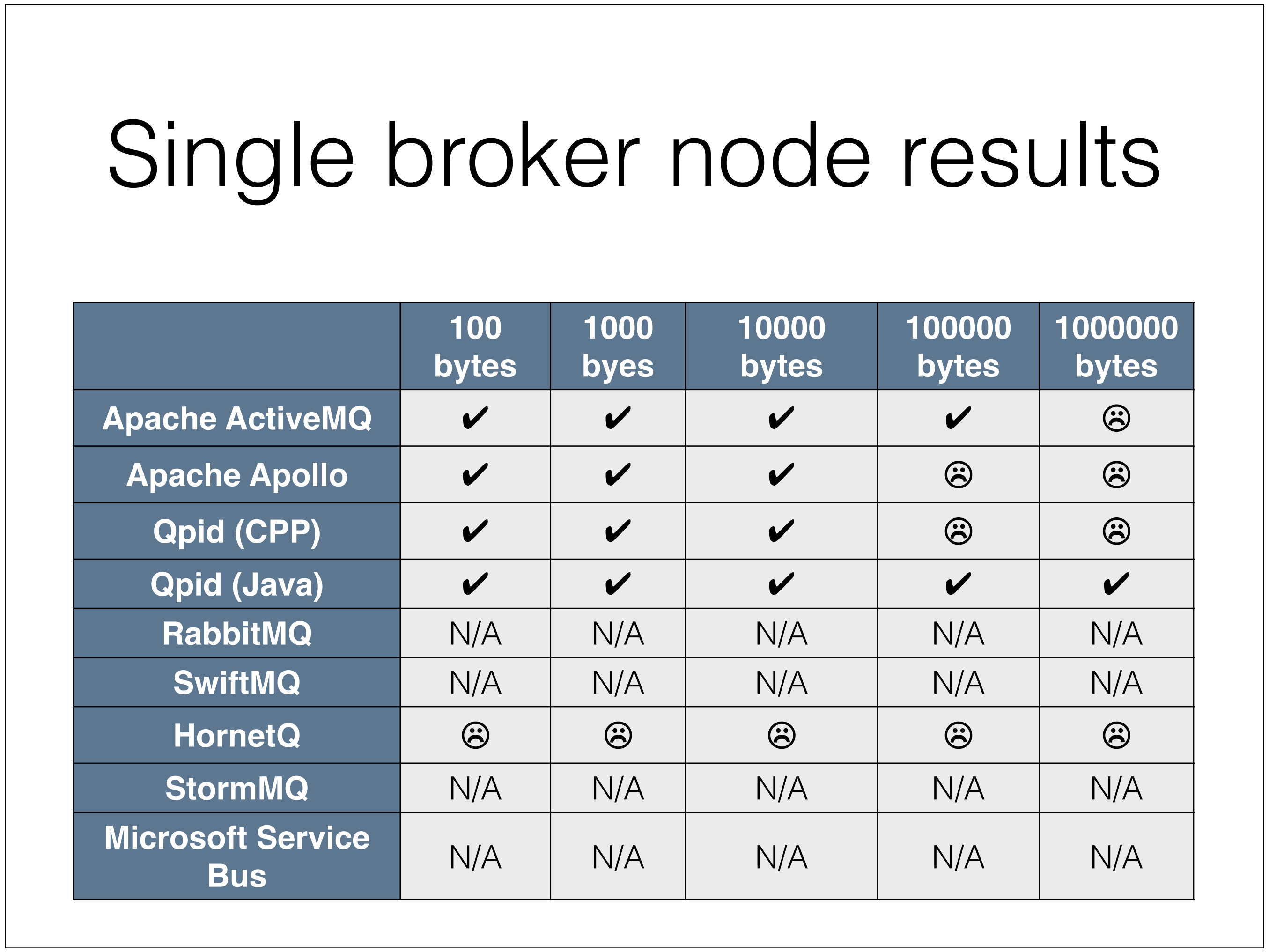
- Develop simple producer/consumer in Java (portability was obviously important).
- Used latest version of Qpid-JMS (v 0.26) client library.
  - One of the more popular clients.
  - Maintained by Redhat - part of Apache foundation.
  - Java allowed for portability
  - Clean API

Also gave me a chance to try maven.

Broker Configuration

- Disable persistence (memory persistence only).
- Minimum authentication.
- Tune memory usage to minimum while ensuring no message loss (in general - use defaults).
- Isolate test network, minimize other processes running on test computers.

Results



Single broker node results

	100 bytes	1000 byes	10000 bytes	100000 bytes	1000000 bytes
Apache ActiveMQ	✔	✔	✔	✔	☹
Apache Apollo	✔	✔	✔	☹	☹
Qpid (CPP)	✔	✔	✔	☹	☹
Qpid (Java)	✔	✔	✔	✔	✔
RabbitMQ	N/A	N/A	N/A	N/A	N/A
SwiftMQ	N/A	N/A	N/A	N/A	N/A
HornetQ	☹	☹	☹	☹	☹
StormMQ	N/A	N/A	N/A	N/A	N/A
Microsoft Service Bus	N/A	N/A	N/A	N/A	N/A

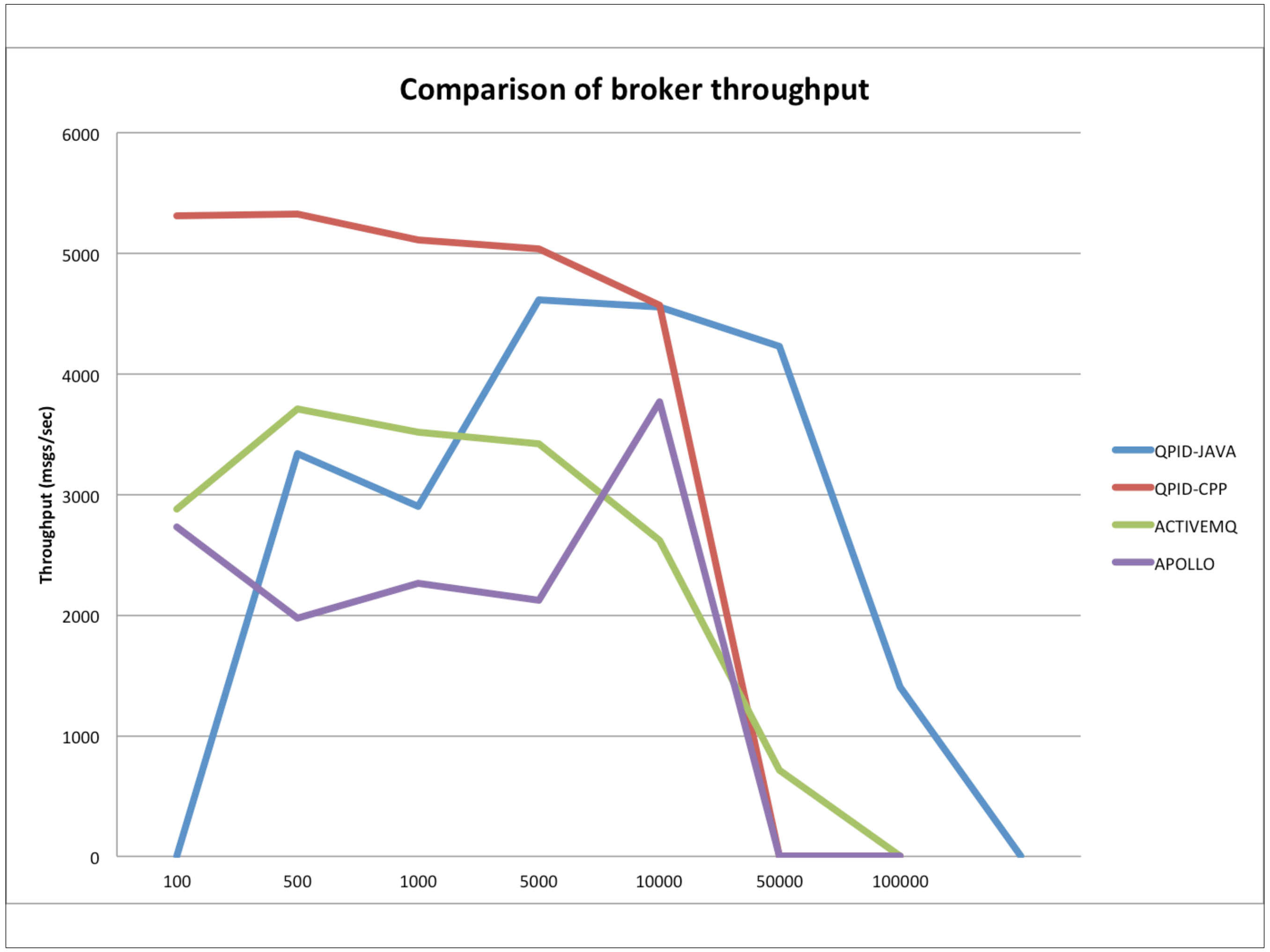
SwiftMQ (actually worked - license prohibits releasing benchmarking information without permission - never got back to me)

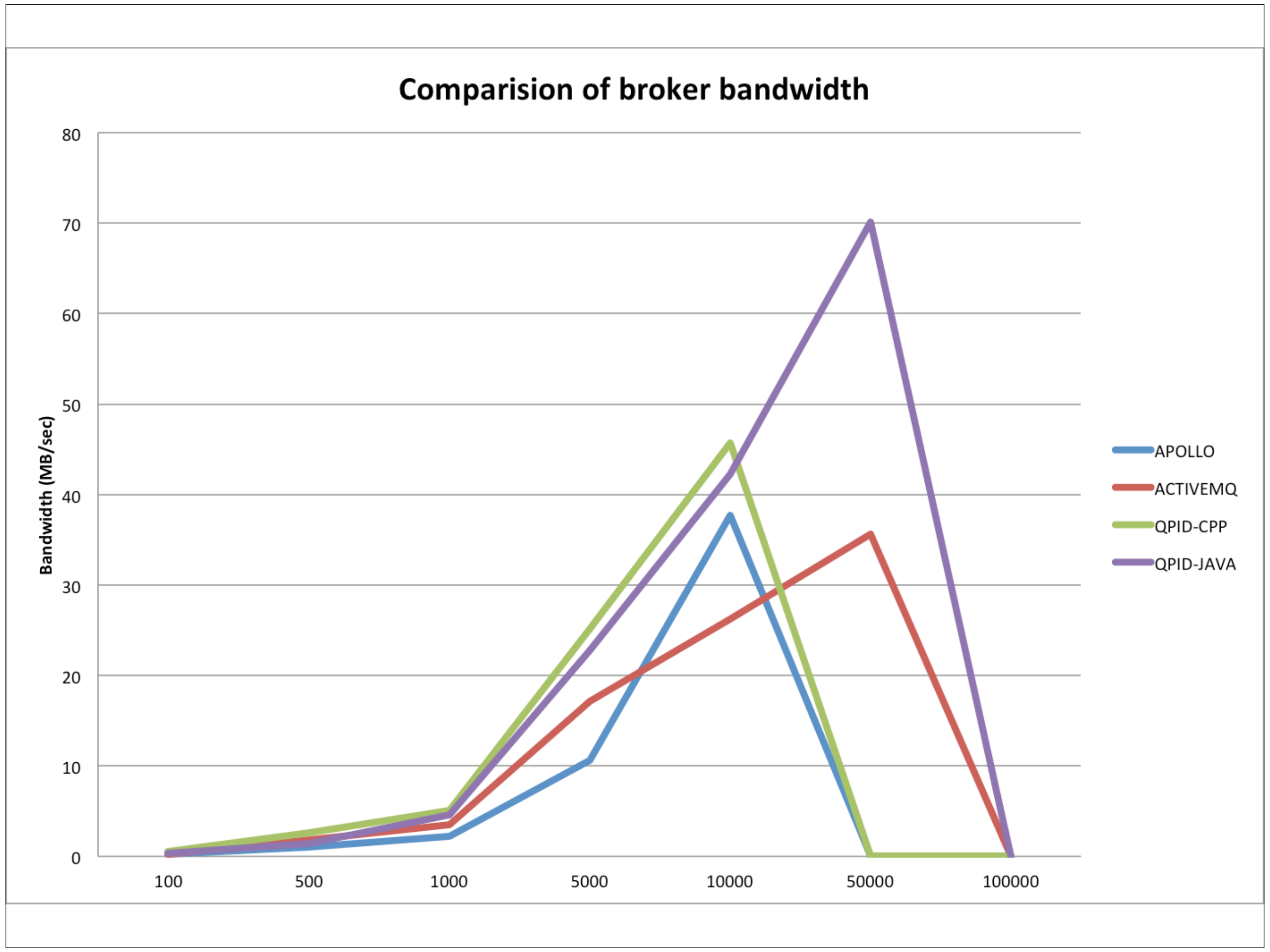
RabbitMQ - does not support AMQP 1.0 (has an experimental plugin which didn't work)

HornetQ - was never able to connect client.

What happened?

- HornetQ never worked. Broker would start, but AMQP clients crashed - suspect is Qpid-JMS client - complaint was about session id in JMS session - not part of AMQP specification.
- RabbitMQ doesn’t support AMQP 1.0 natively. It uses a plugin which didn't work.
- SwiftMQ - doesn’t allow the publication of benchmark data.
- StormMQ - not possible to instantiate a local broker - totally cloud based.
- Microsoft Service Bus - Doesn’t run on Ubuntu.

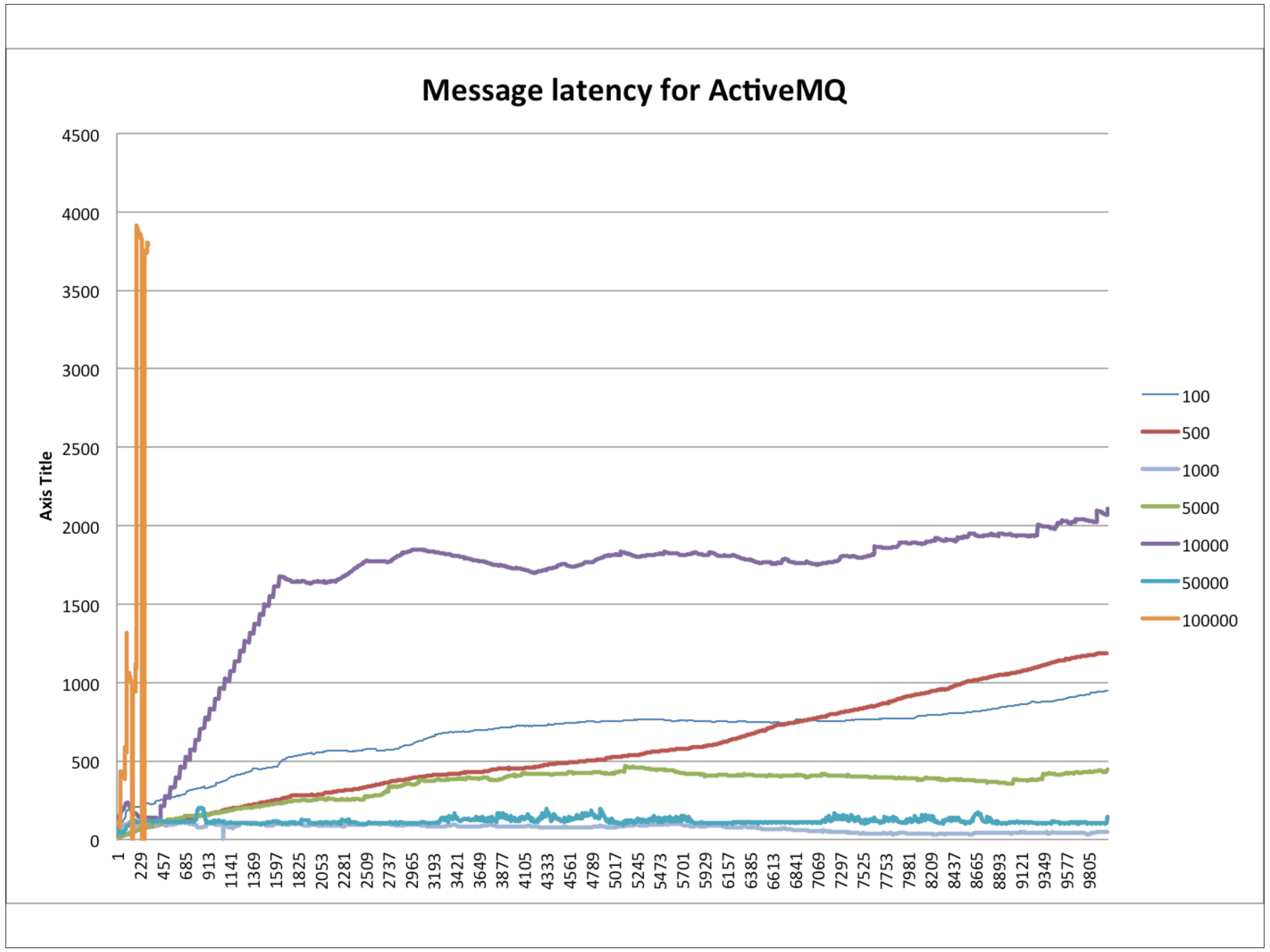

Comparison of broker throughput
Throughput (msgs/sec)
6000
5000
4000
3000
2000
1000
0
100
500
1000
5000
10000
50000
100000
QPID-JAVA
QPID-CPP
ACTIVEMQ
APOLLO



Comparision of broker bandwidth

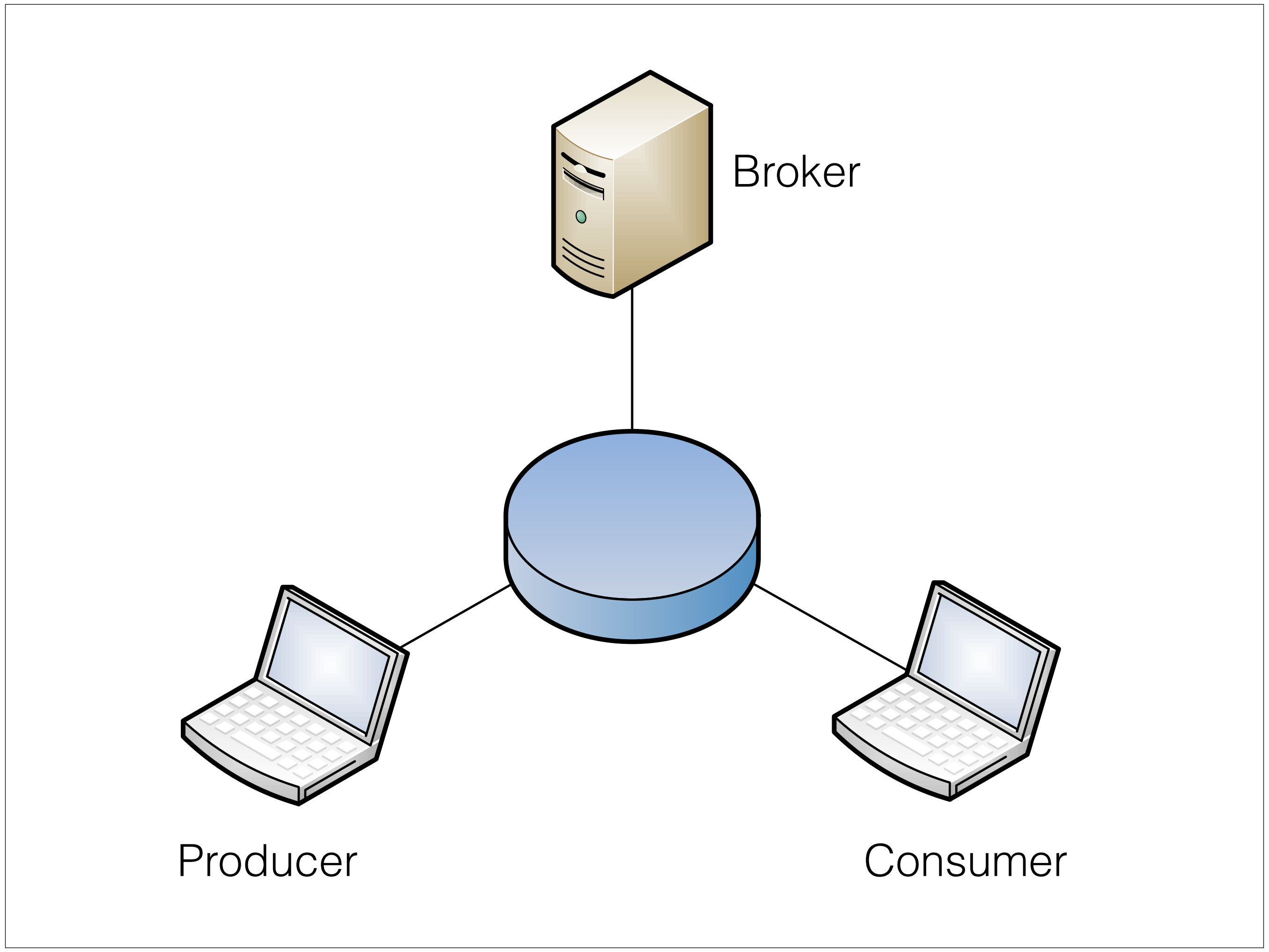
Results

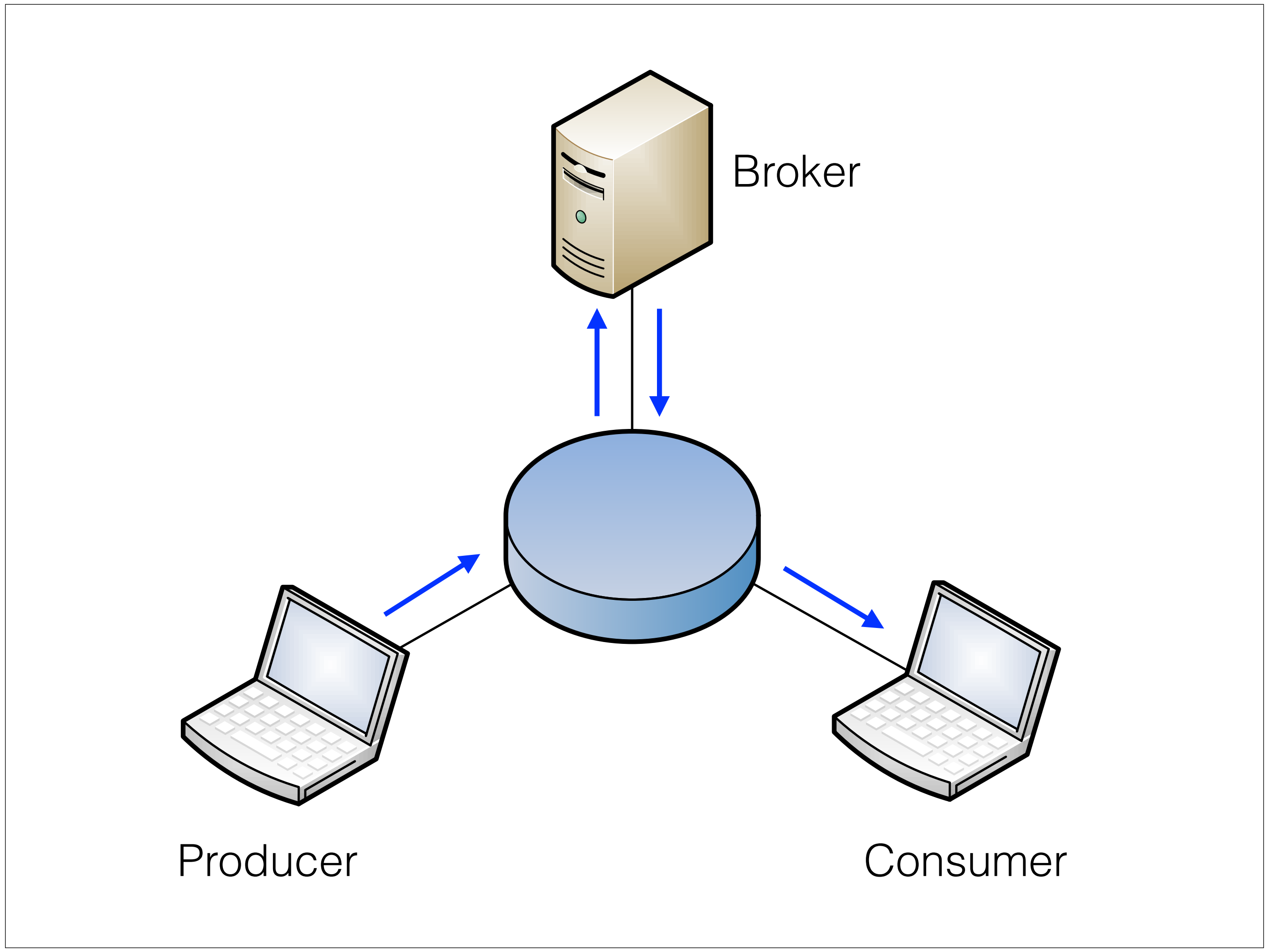
- Bandwidth and Throughput look pretty good and resemble other publicly available results.

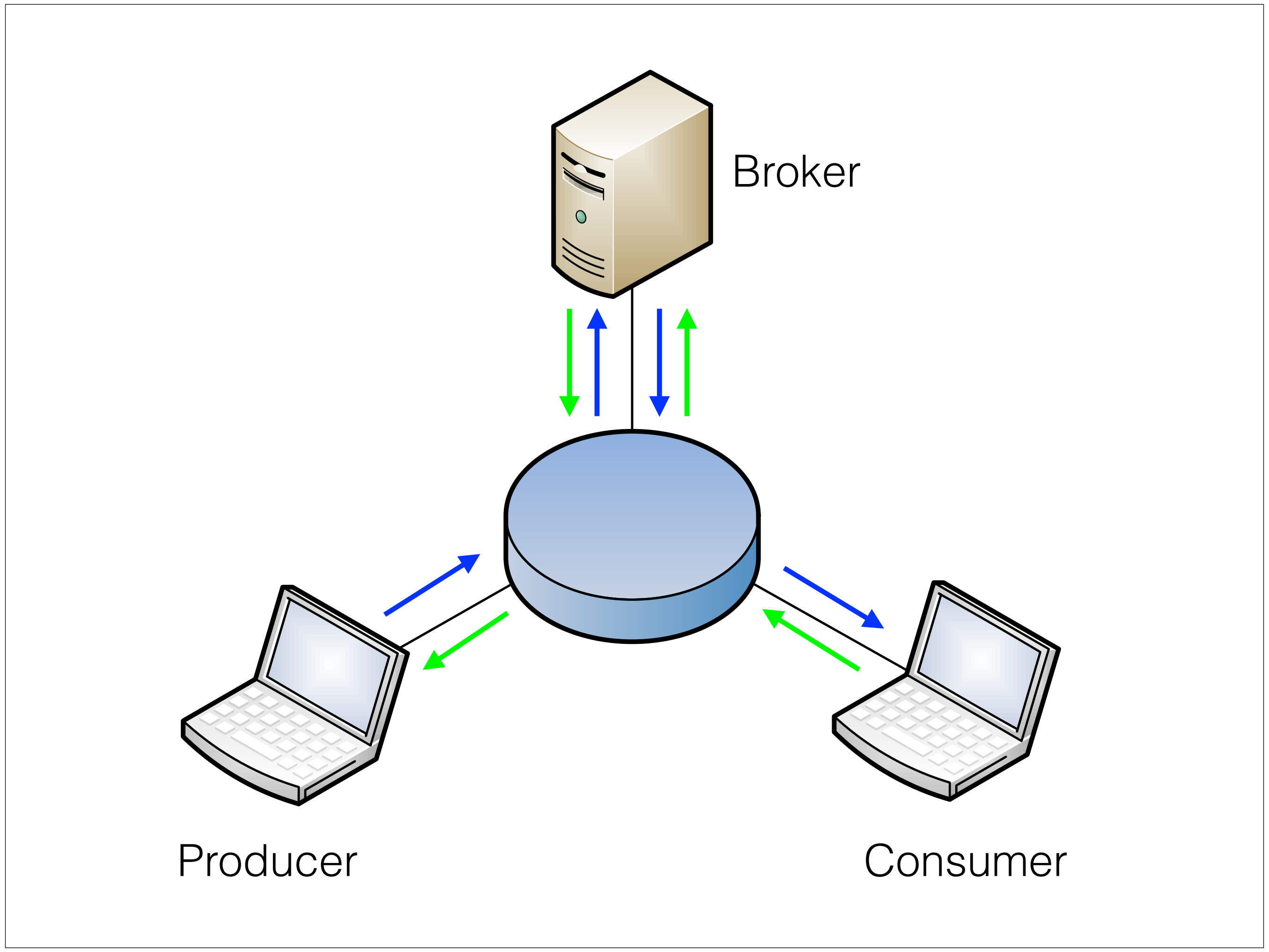

Message latency for ActiveMQ
Axis Title
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1
229
457
685
913
1141
1369
1597
1825
2053
2281
2509
2737
2965
3193
3421
3649
3877
4105
4333
4561
4789
5017
5245
5473
5701
5929
6157
6385
6613
6841
7069
7297
7525
7753
7981
8209
8437
8665
8893
9121
9349
9577
9805
100
500
1000
5000
10000
50000
100000

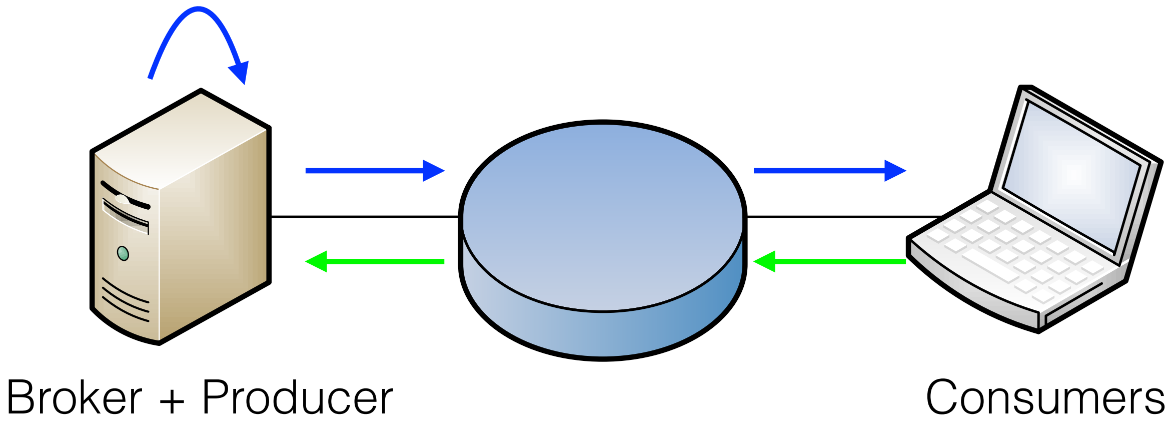
Results - Issues

- A lot going on. JMS client, broker working to maximize performance, not benchmarking accuracy.
- Latency for small messages not accurate.
- Large messages cause the broker/client to fail.
  - Recoverable in some cases, other cases fatal.
- Out of order messages - should not happen with queue exchange.
- Redesign network to simplify measurements and increase accuracy.


Broker
Producer
Consumer


Broker
Producer
Consumer

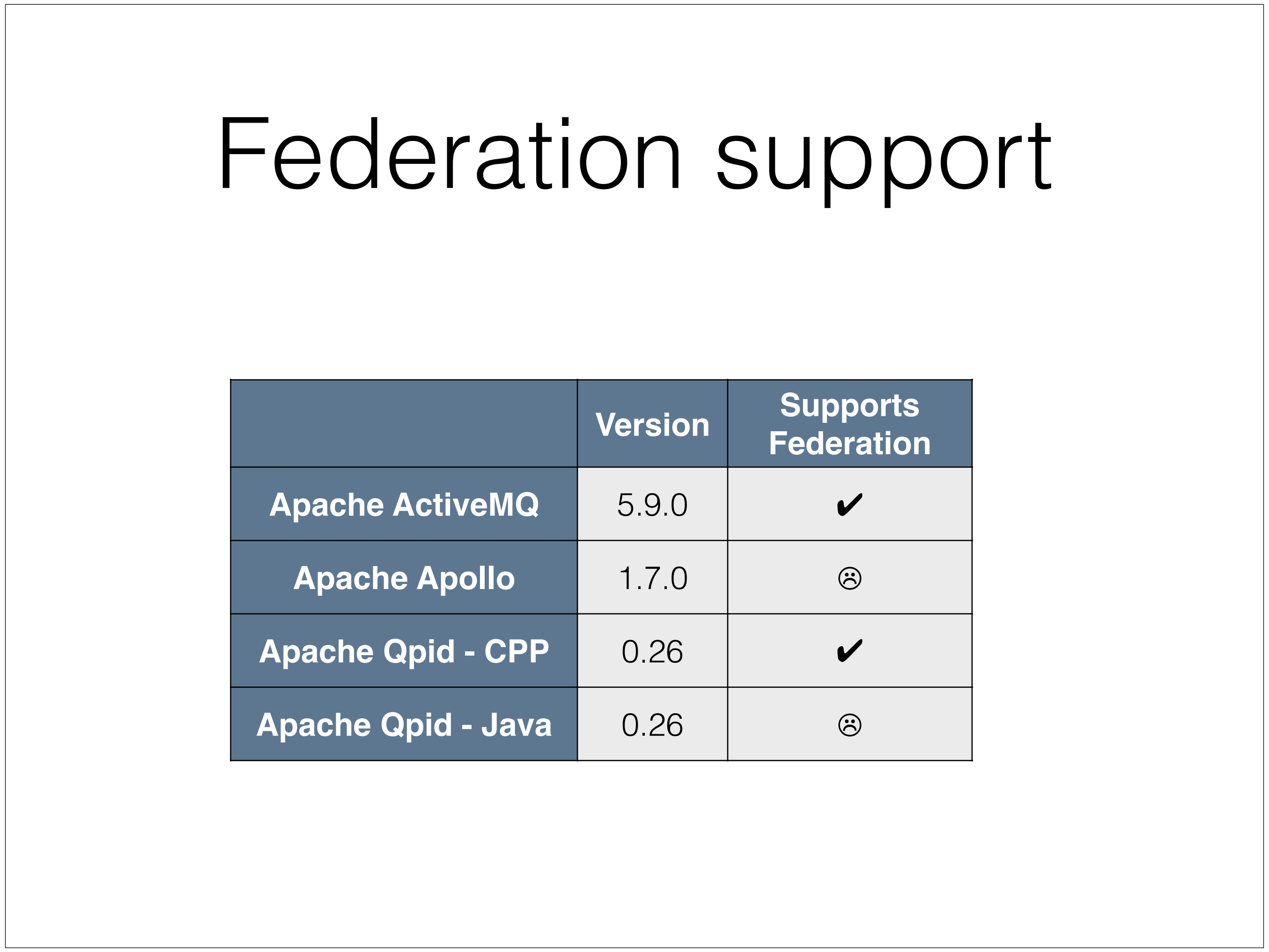

Broker
Producer
Consumer


Broker + Producer
Consumers

Redesign

- Software - ended up requiring two separate benchmarks
  - One for pure bandwidth.
  - One for message latency.
- Needed explicit configuration in clients to support tested brokers.

Federation

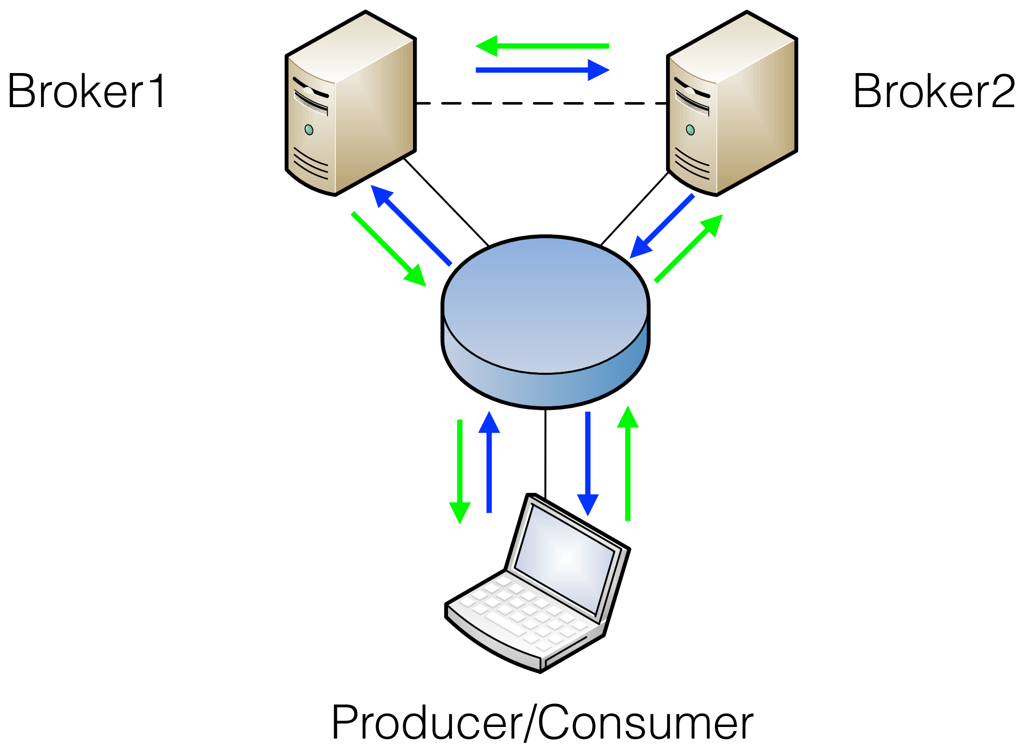


Federation support

	Version	Supports Federation
Apache ActiveMQ	5.9.0	✔
Apache Apollo	1.7.0	☹
Apache Qpid - CPP	0.26	✔
Apache Qpid - Java	0.26	☹

Qpid - Java supports HA (duplication through persistent data store) - but not clustering, was removed in version .18

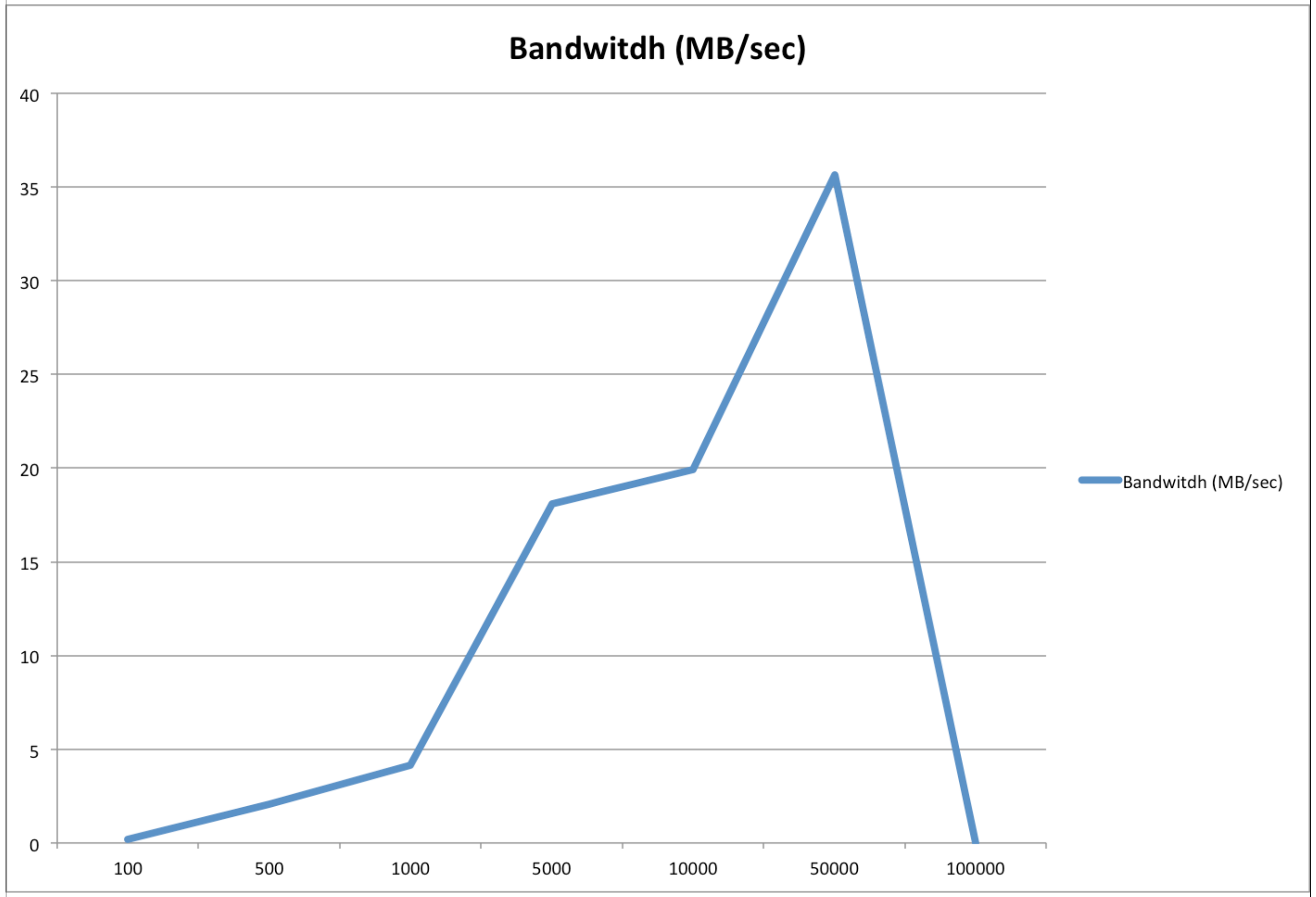
Federated Architecture

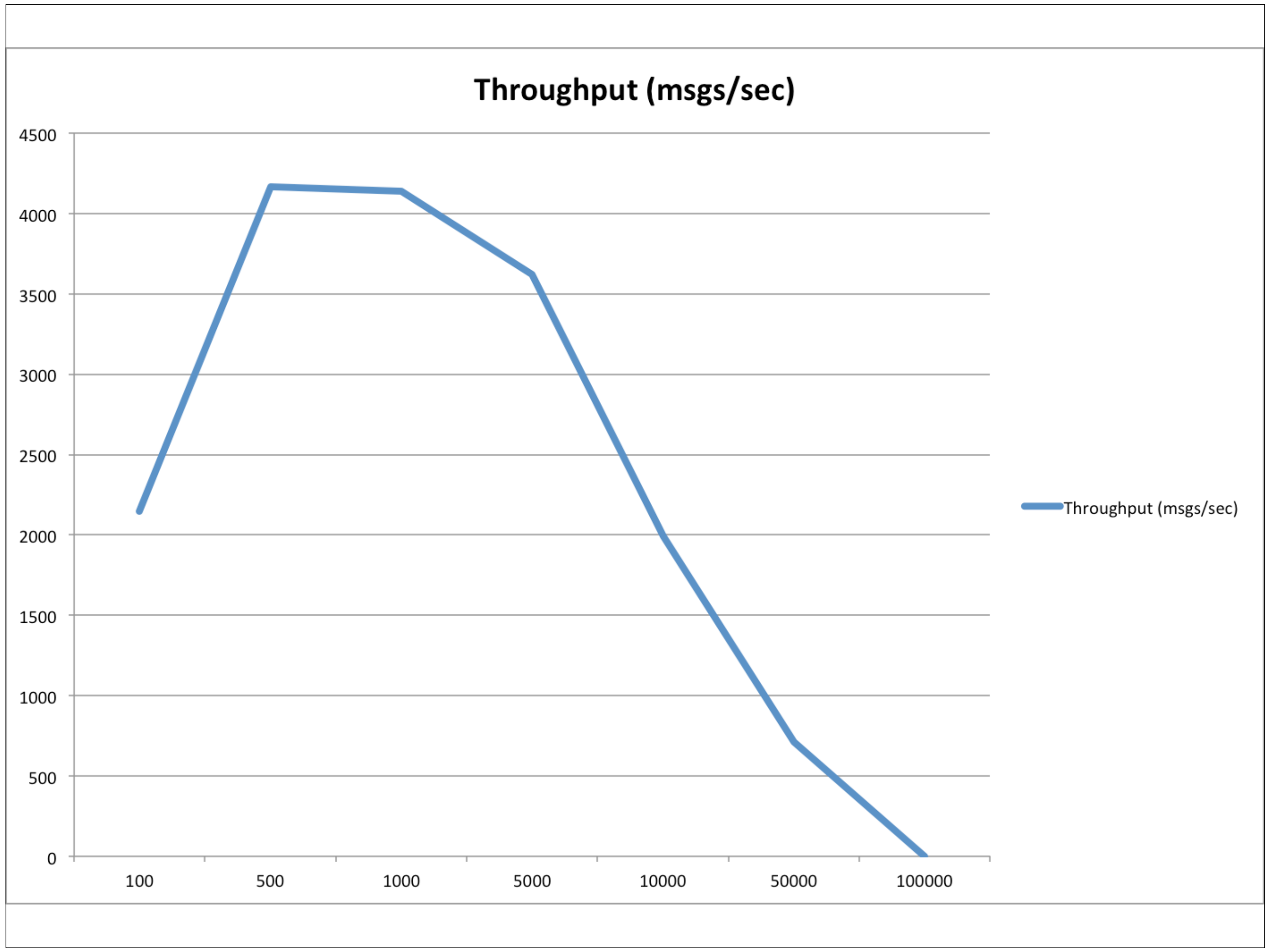


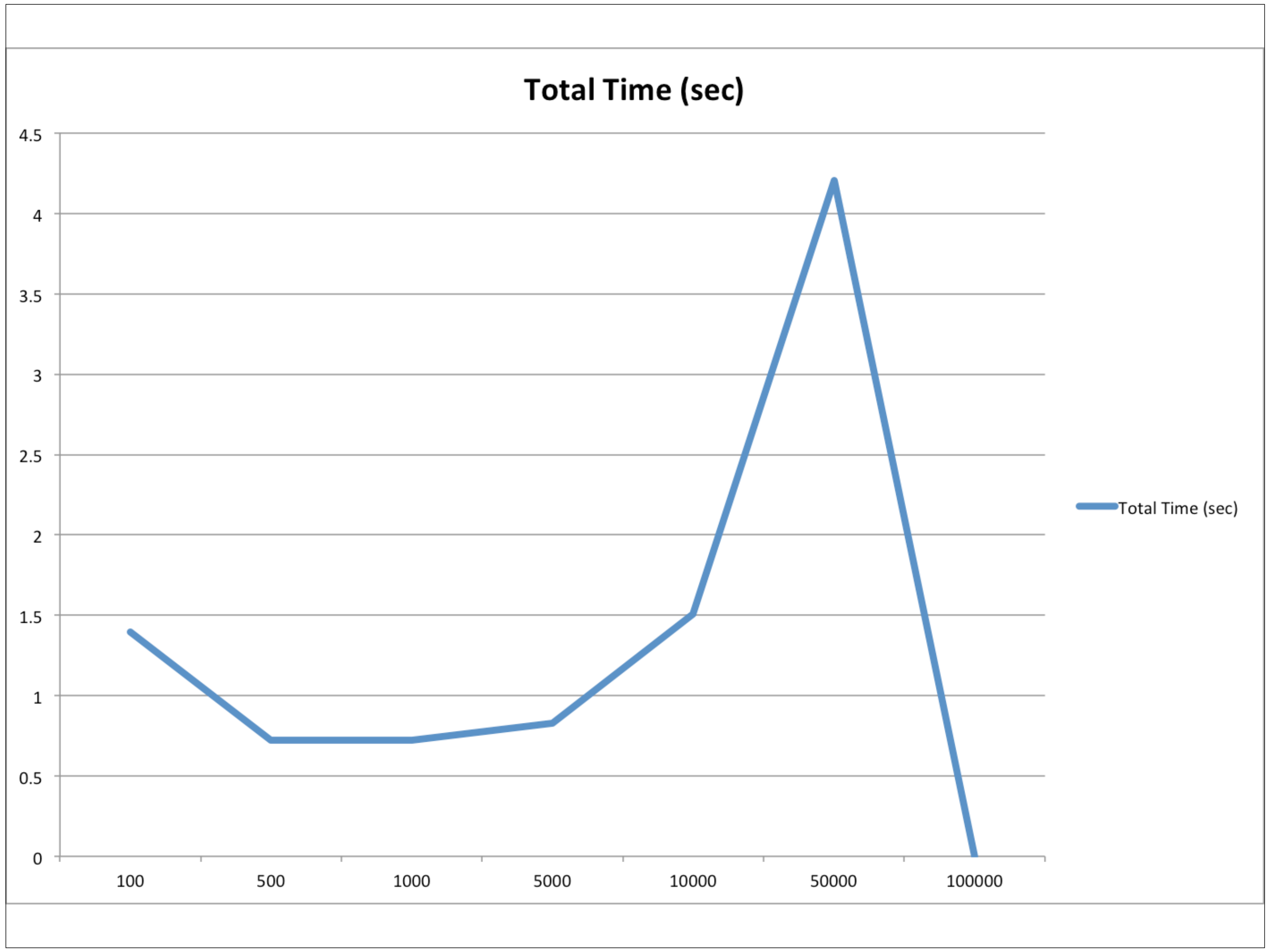
Issues

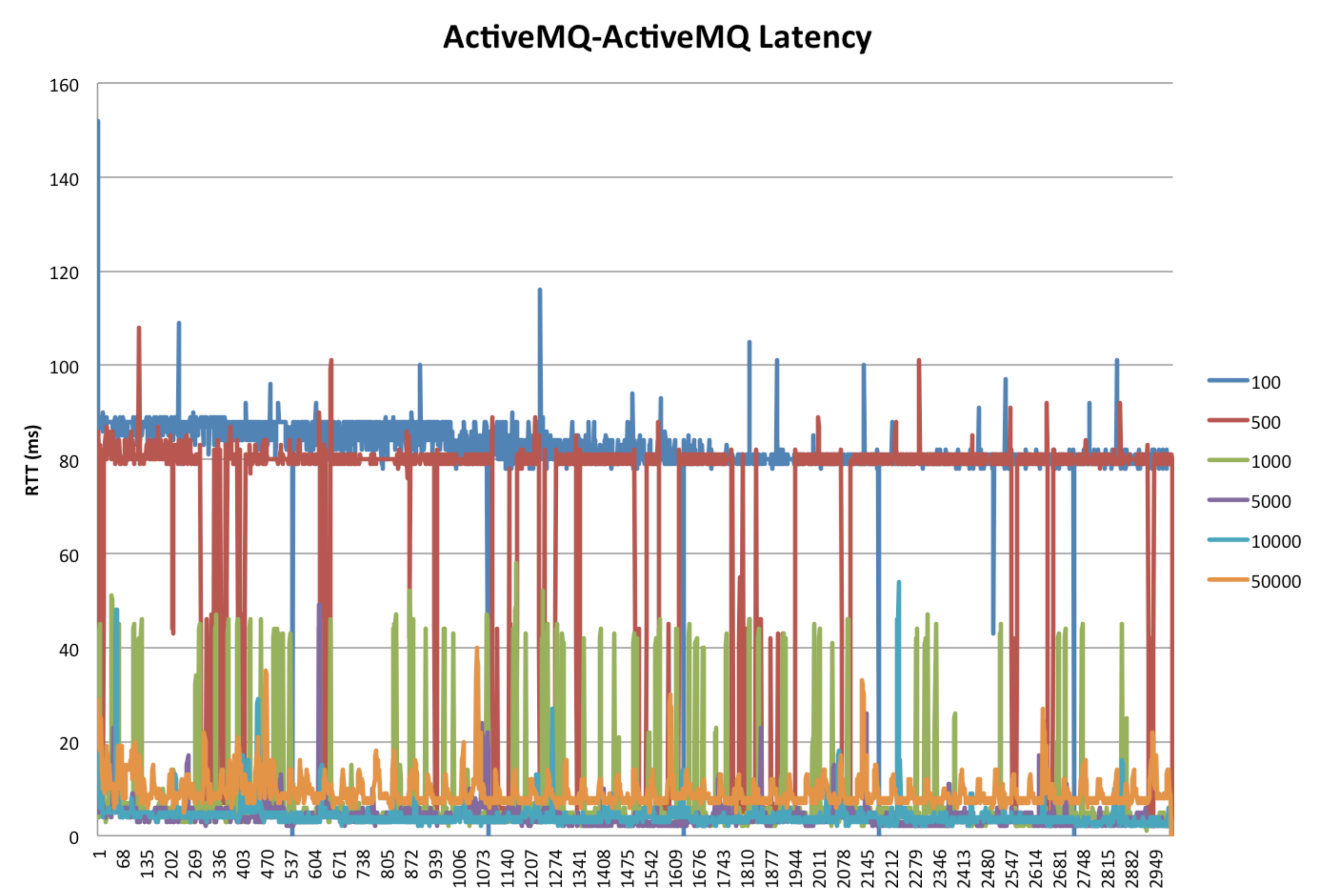
- ActiveMQ worked out of the box
  - **Uses OpenWire not AMQP protocol for federation.**
- Apache Qpid-CPP very difficult to setup - had significant problems with qpid-tools.

ActiveMQ - ActiveMQ

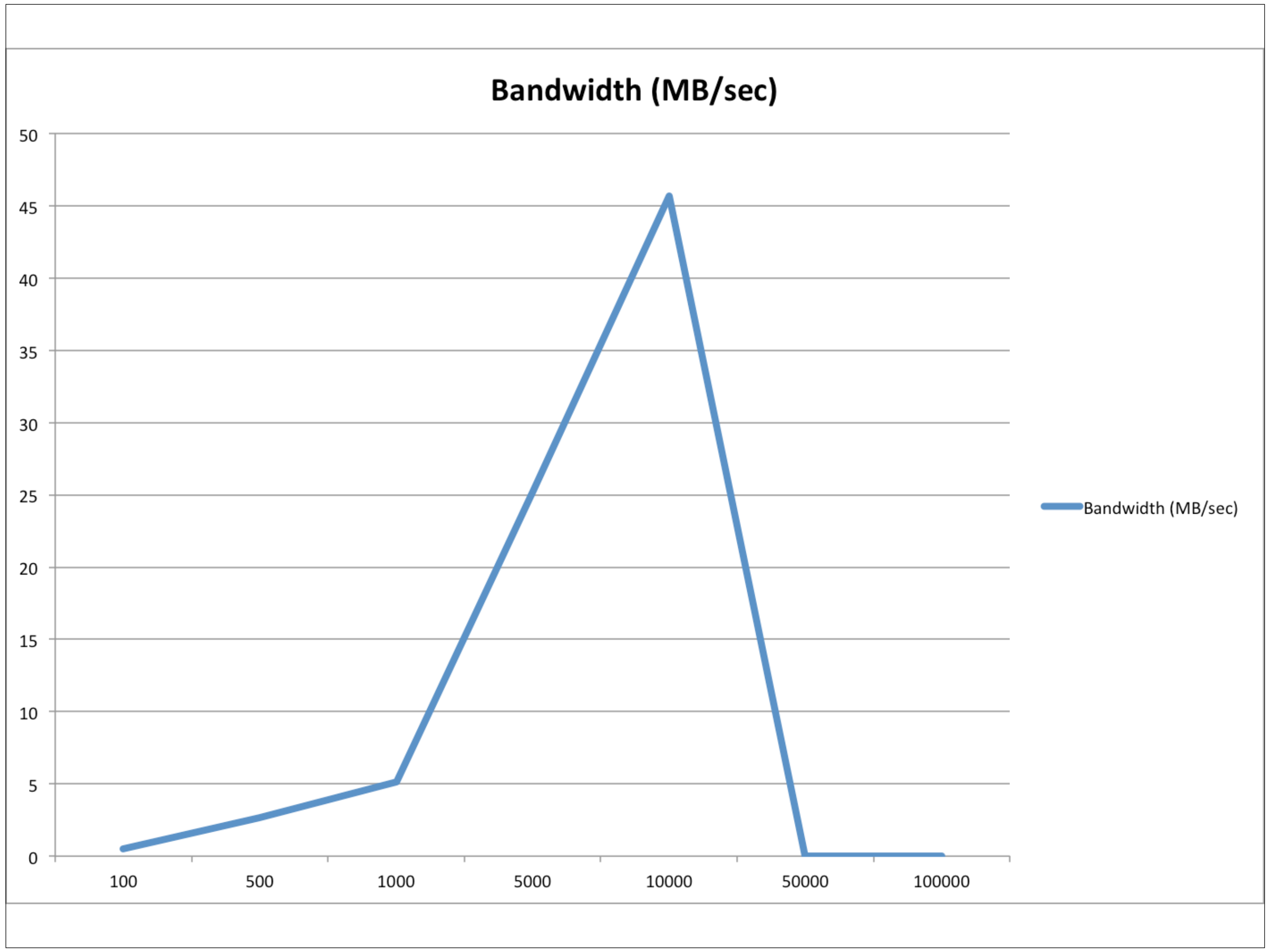

Bandwitdh (MB/sec)
40
35
30
25
20
15
10
5
0
100
500
1000
5000
10000
50000
100000
Bandwitdh (MB/sec)

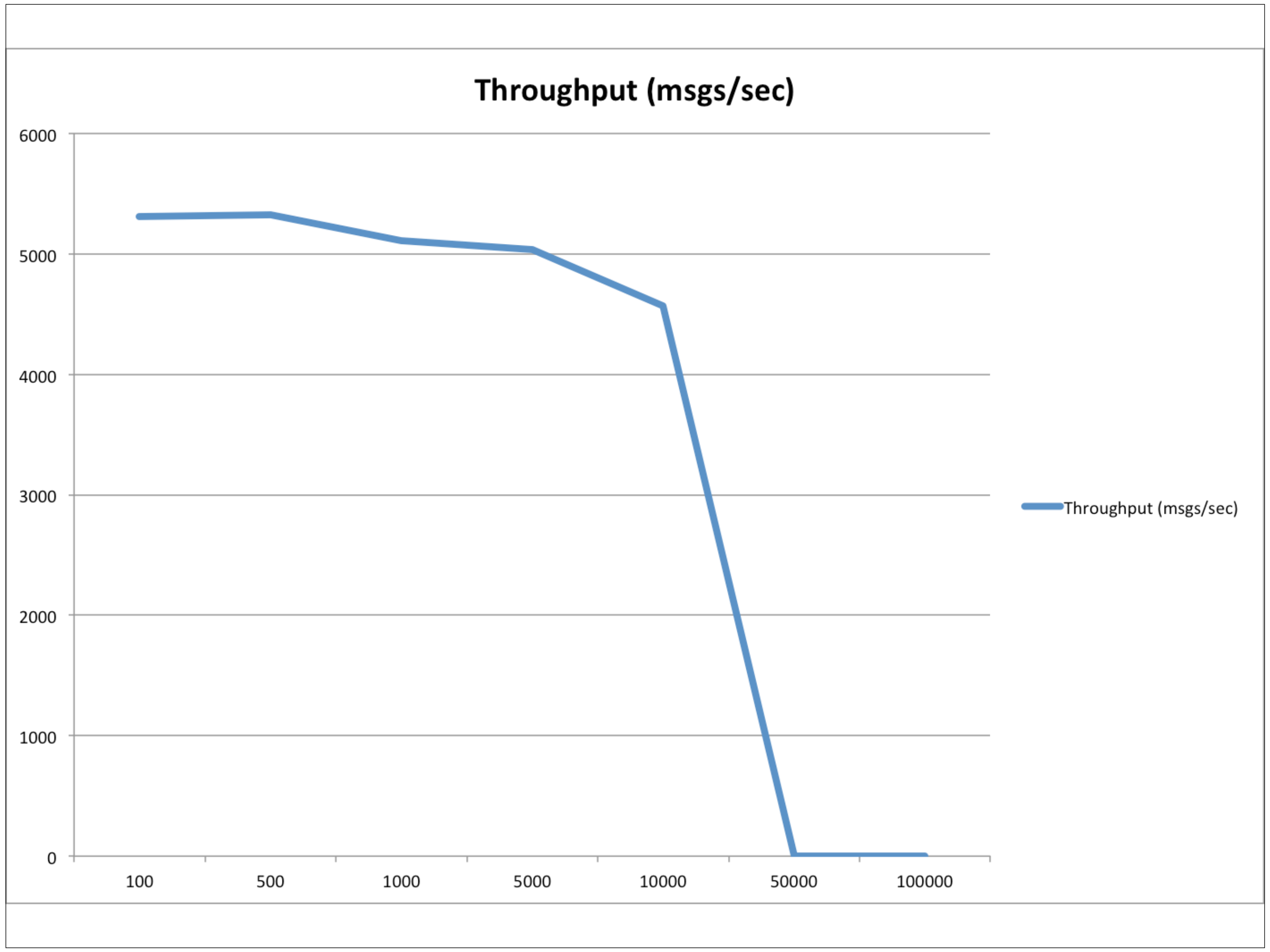

Throughput (msgs/sec)
4500
4000
3500
3000
2500
2000
1500
1000
500
0
100
500
1000
5000
10000
50000
100000
Throughput (msgs/sec)

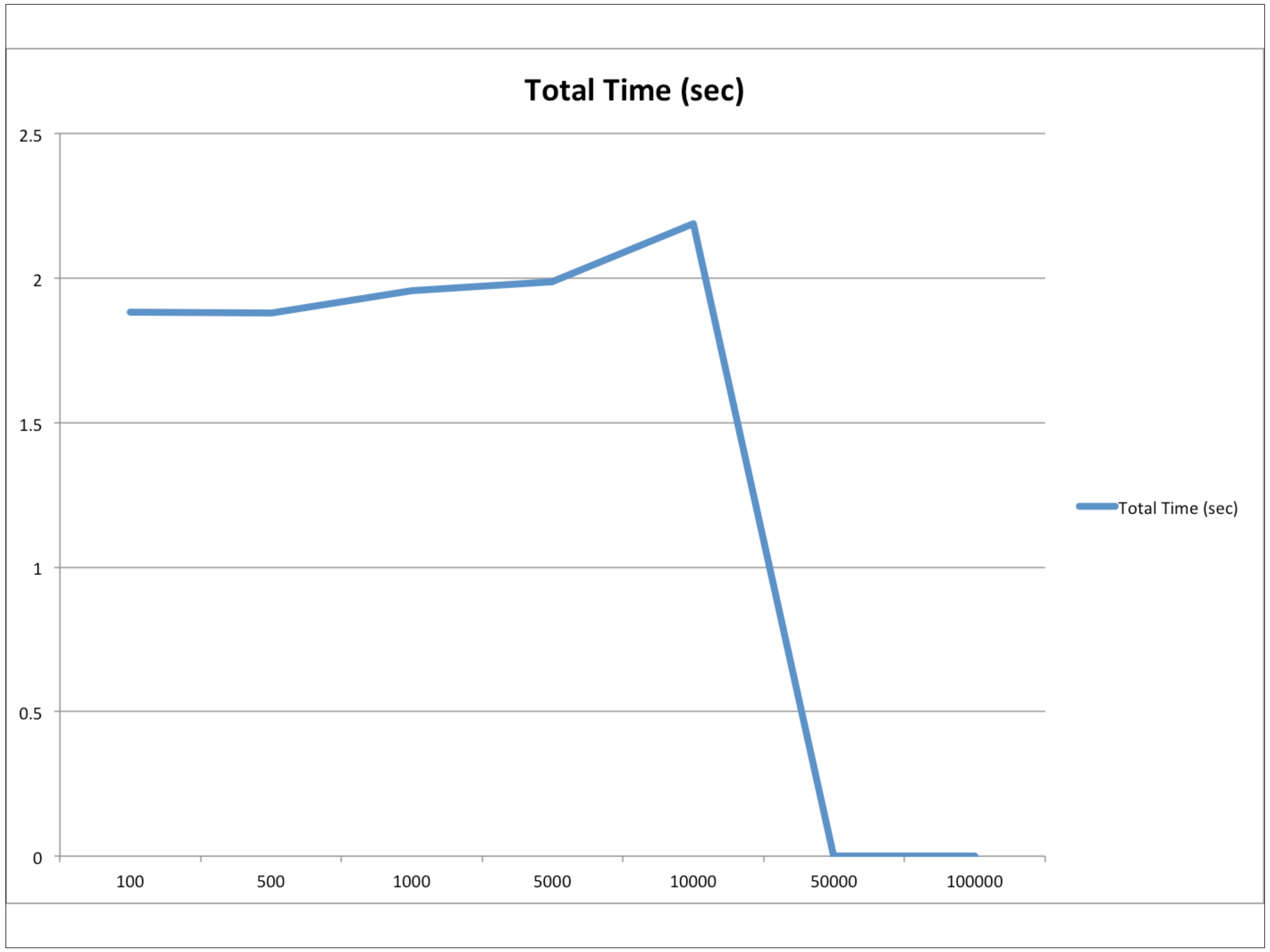

Total Time (sec)
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
100
500
1000
5000
10000
50000
100000
Total Time (sec)

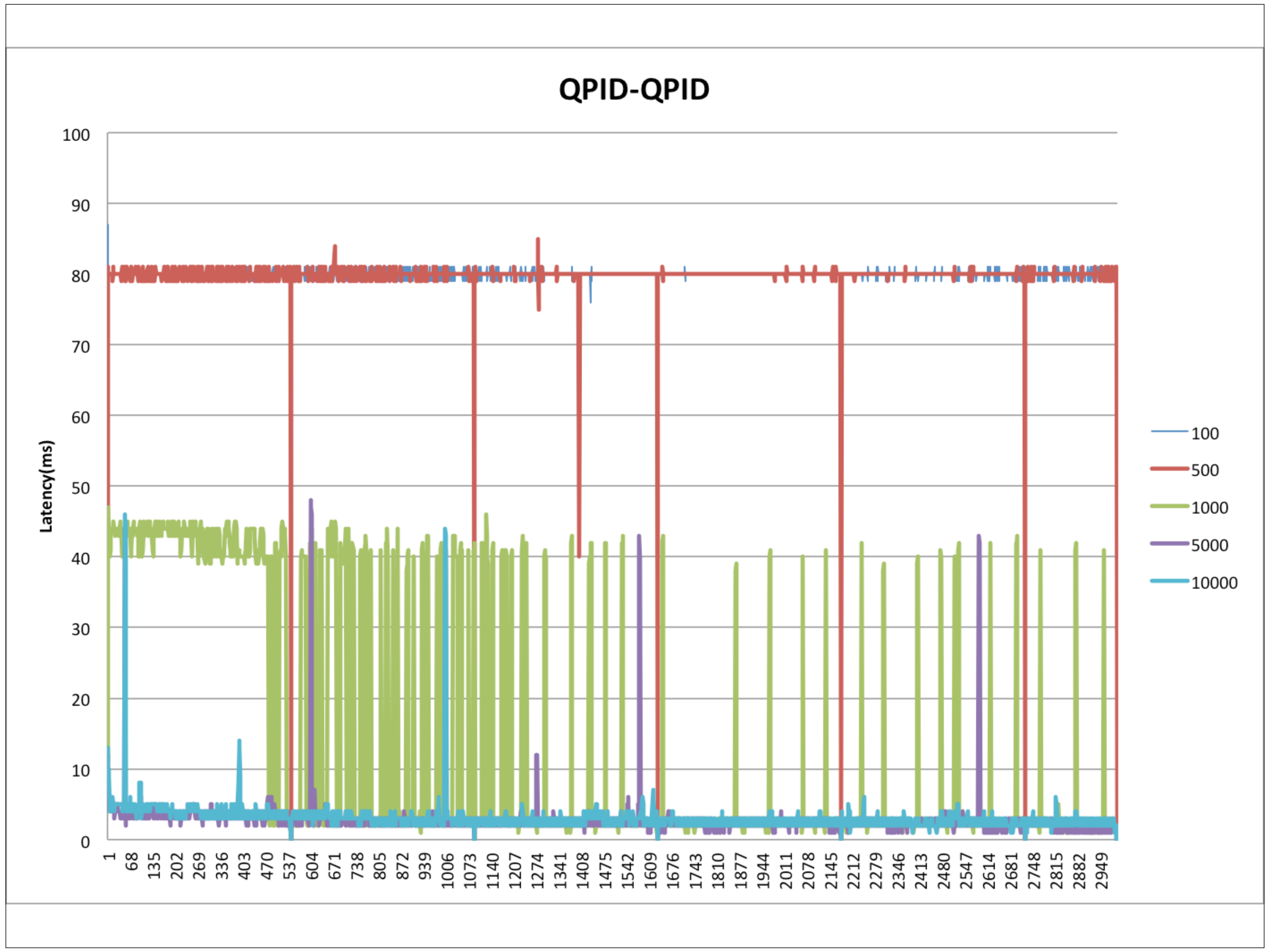

ActiveMQ-ActiveMQ Latency
RTT (ms)
160
140
120
100
80
60
40
20
0
1
68
135
202
269
336
403
470
537
604
671
738
805
872
939
1006
1073
1140
1207
1274
1341
1408
1475
1542
1609
1676
1743
1810
1877
1944
2011
2078
2145
2212
2279
2346
2413
2480
2547
2614
2681
2748
2815
2882
2949
100
500
1000
5000
10000
50000

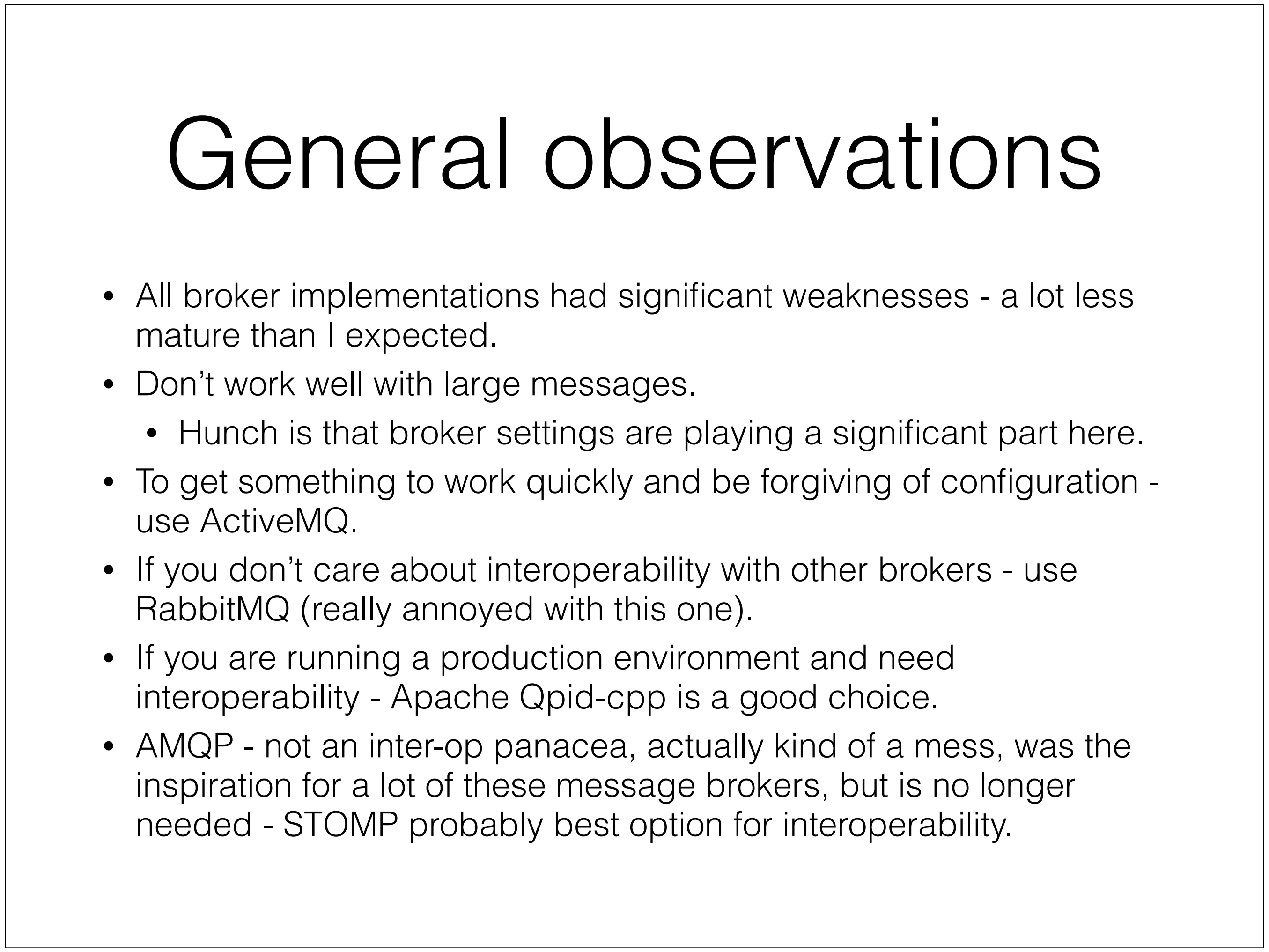
QPID(CPP) - QPID(CPP)


Bandwidth (MB/sec)
50
45
40
35
30
25
20
15
10
5
0
100
500
1000
5000
10000
50000
100000
Bandwidth (MB/sec)


Throughput (msgs/sec)
6000
5000
4000
3000
2000
1000
0
100
500
1000
5000
10000
50000
100000
Throughput (msgs/sec)


Total Time (sec)
2.5
2
1.5
1
0.5
0
100
500
1000
5000
10000
50000
100000
Total Time (sec)


QPID-QPID
Latency(ms)
100
90
80
70
60
50
40
30
20
10
0
1
68
135
202
269
336
403
470
537
604
671
738
805
872
939
1006
1073
1140
1207
1274
1341
1408
1475
1542
1609
1676
1743
1810
1877
1944
2011
2078
2145
2212
2279
2346
2413
2480
2547
2614
2681
2748
2815
2882
2949
100
500
1000
5000
10000



General observations

- All broker implementations had significant weaknesses - a lot less mature than I expected.
- Don’t work well with large messages.
  - Hunch is that broker settings are playing a significant part here.
- To get something to work quickly and be forgiving of configuration - use ActiveMQ.
- If you don’t care about interoperability with other brokers - use RabbitMQ (really annoyed with this one).
- If you are running a production environment and need interoperability - Apache Qpid-cpp is a good choice.
- AMQP - not an inter-op panacea, actually kind of a mess, was the inspiration for a lot of these message brokers, but is no longer needed - STOMP probably best option for interoperability.

Both clients and brokers had issues with large messages. Hunch is that Java Heap configuration is playing a part of this.

ActiveMQ

Better option?

ZeroMQ and Google Protocol Buffers.

+ Forgiving

- large, a bit heavy weight.

- too many configuration options.

RabbitMQ

+ Fast

+ Mature

+ Active development.

Qpid

~ goal to be AMQP 1.0 compliant, yet does not include AMQP 1.0 library in default build.

+ Static config

Questions/Comments/ Death Threats?

sorry it’s so late…