Artemis 2.0

Clebert Suconic
RedHat
Artemis 2.0

Artemis 2.1

Clebert Suconic
RedHat

Things are moving fast!!!!!
Agenda

• Origin
• Features
• Development Stream
• Some architecture
• AMQP
• Little demo
ActiveMQ/Artemis

- Message Broker
  - Messages stay in memory (fast delivery)
  - Paging when slow consuming
  - Acked and guaranteed delivery
• HornetQ merged ActiveMQ in Oct 2014
• HornetQ itself had its own history
  • JBoss MQ -> JBoss Messaging -> HornetQ
ActiveMQ Artemis

- Apache Open Source process
- Community oriented
- Lots of improvements on Artemis
- over 3K commits since joined Apache ActiveMQ
- Lots of QE
ActiveMQ Artemis

- OpenWire support
- export data from AMQ5
- OSGI / Karaf
- Security Models
- Improved XA
- NIO Models / Multiple protocols
2.x accomplishments

• Improved AMQP support

• High performance
  • qpid cpp client can generate 70K messages / second on a simple laptop
    • https://github.com/ssorj/quiver

• Stable (a lot of testing done)
Artemis 2.x Clients

AMQP
JMS 2, Python, JavaScript/Node.js, .NET, ...

MQTT

Stomp

OpenWire
JMS, C++, .NET

HornetQ/Artemis JMS

Artemis 2.1
AMQP Clients

- Java JMS 1.1 client (Apache Qpid JMS based on Qpid Proton)
- Reactive C++ client (Apache Qpid Proton)
- Reactive Python client (Apache Qpid Proton)
- Reactive pure JavaScript client w support for Node.js (GitHub Rhea)
- Fully-featured .NET library (GitHub AMQP .NET Lite)
Artemis 2.x

- Messages on their Protocol
- (currently AMQP and Core)

This Means:

Artemis is protocol agnostic now
New Addressing Model

• New address object exposed with two routing type options

  • Anycast
    • Point to point: Messages are distributed amongst many consumers

  • Multicast
    • Publish / Subscribe: Every Queue(subscription) receives a copy of every message

• New Queue fields

  • Allow explicit broker side configuration of queues with different semantics

  • Max-consumers: Restricts sharing

• Purge-on-no-consumers: Behaves like a non-durable subscription
Component approach

Persistence

Broker

Protocol Manager

Network

Clients

Netty
Asynchronous Internal Tasks

Result: Less resources from server
Asynchronous Internal Tasks

Result: Less resources from server
Journal Internal Format

File 1 → File 2 → File 3

Add(1)
Update(1)
Add(2)
Add(2)
Delete(1)
Prepare(2)
Commit(2)
Add(3)
Update(3)
Add(4)

Add(5)
Update(5)
Add(6)
Delete(3)
Delete(4)

Once record 3 and 4 are deleted, page 1 can be reclaimed.
Journal Internal Format

Fast access

Once record 3 and 4 are deleted page 1 can be reclaimed
Journal Internal Format

File 1 → File 2 → File 3

Add(1)
Update(1)
Add(2)
Add(2)
Delete(1)
Prepare(2)
Commit(2)
Add(3)
Update(3)
Add(4)

Add(5)
Update(5)
Add(6)
Delete(3)
Delete(4)

Fast recovery
Fast Writing

Once record 3 and 4 are deleted, page 1 can be reclaimed
Paging internal Format

Address Folder

MSG1
MSG2
MSG3
MSG4
MSG5
MSG6

Fast Write

MSG1
MSG2
MSG3
MSG4
MSG5
MSG6

Reading Cache
Paging internal Format as TX

Address Folder

MSG1   MSG2   MSG3   MSG4   MSG5   MSG6
MSG1   MSG2   MSG3   MSG4   MSG5   MSG6
MSG1   MSG2   MSG3   MSG4   MSG5   MSG6
MSG1   MSG2   MSG3   MSG4   MSG5   MSG6

Fast Write

Reading Cache
TX will require a record on the journal.
Sync of writes

Client

Context

Disk
Sync of writes on replica

- initial catch up
- minimal delay on master
High availability

- Replication
- Split Brain protection
- PINGS
- QUORUM (3+ Pairs (best))
- Shared storage
Clustering

• Artemis Internal Clustering
  • Bridges / Destinations
    • Especial for Topics

• AMQP offers you more opportunities
  • qpid-dispatch-router will offer great possibilities
    • http://qpid.apache.org/components/dispatch-router/
qpid-dispatch Clustering
Pooled Buffers

- Netty Pooled Buffers everywhere on the communication layers
- Improving 2.2.0 now with buffer pooled on bodies
- Low GC pressure