A Messaging Infrastructure for WLCG

James Casey, Lionel Cons, Wojciech Lapka, Massimo Paladin, Konstantin Skaburskas

Grid Technology Group
Information Technology Department, CERN

CHEP 2010, Taipei
Message-Oriented Middleware
Message-Oriented Middleware
WLCG infrastructure is complex...
• What is a messaging system?
  – **Standardized, asynchronous** and **scalable** communication between **distributed** entities
  – Broker-mediated communication
  – Messaging is for applications what IM is for people

• Aims of messaging systems
  – Simplify the WLCG infrastructure and make it more effective
  – Allow sharing of the information between distributed components
• Each message received by only one receiver
• Guaranteed message delivery
Publish/Subscribe Messaging

- Each message received by all available subscribers
- Message delivery not guaranteed (except durable subscribers)
Messaging protocols

• Java Message Service (JMS)
  “is a messaging standard that allows application components to create, send, receive, and read messages”

• STOMP (Streaming Text Orientated Messaging Protocol)
  – Simple text based protocol designed for messaging systems
  – Supported by most messaging systems
  – Clients available in all major programming languages

• OpenWire
  – JMS compliant Native Apache ActiveMQ protocol
  – Clients available in C, C++, C#, Java
• **ActiveMQ is a top-level Apache Project**
  – Mature open-source product
  – Written in Java (implements JMS semantics)
  – Excellent cross-language interoperation
  – Enterprise features (*Persistence*, *Failover*, *Clustering*, ...)
  – Many large reference customers – Vodafone, GE, FAA, (CERN)
  – CERN has industry support contract with Progress Software

• **Good performance characteristics**
  – 1K – 20K messages per second depending on features used
Network of Brokers

Diagram showing a network of brokers connected through persistent message stores, with producers and consumers interacting with the brokers.
Broker monitoring

- We use Nagios to monitor
  - Brokers
  - Functionality testing

- Broker monitoring
  - OS: Filesystem full, processes running, socket counts, open file counts
  - ActiveMQ statistics: Store usage, JVM stats, queues with stalled messages

- Functionality testing
  - Nagios sends and receives test messages
<table>
<thead>
<tr>
<th>Host</th>
<th>Service</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Attempt</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>gridmsg101.cern.ch</td>
<td>ActiveMQ.Connections.test</td>
<td>OK</td>
<td>10-15-2010 09:59:15</td>
<td>0d 19h 37m 32s</td>
<td>1/3</td>
<td>OK - 7 plugins checked, 7 ok</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.Consumers</td>
<td>OK</td>
<td>10-15-2010 10:01:37</td>
<td>0d 18h 35m 10s</td>
<td>1/3</td>
<td>check_multi OK - 2 plugins checked, 2 ok</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.inactive_durable_subscribers</td>
<td>OK</td>
<td>10-15-2010 10:00:47</td>
<td>0d 19h 36m 0s</td>
<td>1/3</td>
<td>INACTIVE_DURATION OK - checked 0 durable subscribers, 0 offline</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.Live_threads</td>
<td>WARNING</td>
<td>10-15-2010 10:01:09</td>
<td>0d 0h 11m 38s</td>
<td>3/3</td>
<td>check_multi WARNING - 2 plugins checked, 1 warning (threads_trend), 1 ok</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.Memory</td>
<td>OK</td>
<td>10-15-2010 10:00:31</td>
<td>0d 19h 36m 16s</td>
<td>1/3</td>
<td>check_multi OK - 3 plugins checked, 3 ok</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.Messages_number</td>
<td>OK</td>
<td>10-15-2010 09:57:53</td>
<td>0d 19h 43m 54s</td>
<td>1/3</td>
<td>MESSAGES_NUMBER OK - Rate per second since last check (300 seconds 65)</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.Network.test</td>
<td>OK</td>
<td>10-15-2010 09:59:00</td>
<td>0d 19h 37m 47s</td>
<td>1/3</td>
<td>OK - 3 plugins checked, 3 ok</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.Process</td>
<td>OK</td>
<td>10-15-2010 09:59:37</td>
<td>0d 19h 32m 10s</td>
<td>1/3</td>
<td>PROC OK: 1 process with args '/usr/share/activemq/bin/run.jar'</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.Queues_with_pending_messages</td>
<td>WARNING</td>
<td>10-15-2010 09:56:59</td>
<td>0d 17h 54m 48s</td>
<td>3/3</td>
<td>PENDING_MESSAGES WARNING - 200 queues matching name &quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.TCP.established_connections</td>
<td>OK</td>
<td>10-15-2010 10:01:21</td>
<td>0d 19h 35m 26s</td>
<td>1/3</td>
<td>OK - tcp6162_in is 1, tcp6163_in is 124, tcp6166_in is 2, tcp6167_in is 0</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.TCP.listening_sockets</td>
<td>OK</td>
<td>10-15-2010 09:59:43</td>
<td>0d 19h 32m 4s</td>
<td>1/3</td>
<td>OK - tcp6162_in is 1, tcp6163_in is 1, tcp6166_in is 1, tcp6167_in is 1</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.Wrapper_Process</td>
<td>OK</td>
<td>10-15-2010 09:57:53</td>
<td>0d 19h 33m 54s</td>
<td>1/3</td>
<td>PROC OK: 1 process with args '/usr/share/activemq/bin/linux/wrapper'</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.log_files</td>
<td>WARNING</td>
<td>10-15-2010 09:55:20</td>
<td>0d 18h 6m 27s</td>
<td>1/1</td>
<td>WARNING - 76 warnings</td>
</tr>
<tr>
<td></td>
<td>ActiveMQ.open_files</td>
<td>OK</td>
<td>10-15-2010 10:01:15</td>
<td>0d 19h 35m 32s</td>
<td>1/3</td>
<td>LSOF_NUM_OK - 0.30% file descriptors open (390 / 131072)</td>
</tr>
<tr>
<td></td>
<td>NRPE</td>
<td>OK</td>
<td>10-15-2010 09:55:20</td>
<td>0d 19h 36m 27s</td>
<td>1/4</td>
<td>NRPE v2.12</td>
</tr>
<tr>
<td></td>
<td>NRPE-Push</td>
<td>OK</td>
<td>10-15-2010 09:55:20</td>
<td>220d 15h 0m 47s</td>
<td>1/4</td>
<td>PUSH_NRPE_OK - URL: <a href="https://samnag002.cern.ch/nrpe/gridmsg101.cern.ch">https://samnag002.cern.ch/nrpe/gridmsg101.cern.ch</a> a823a1ae7dd317335adcf1886686a0c2c deployed</td>
</tr>
<tr>
<td></td>
<td>jm4perl</td>
<td>OK</td>
<td>10-15-2010 10:00:37</td>
<td>0d 19h 36m 10s</td>
<td>1/3</td>
<td>HTTP_OK: HTTP/1.1 200 OK - 6923 bytes in 0.023 second response time</td>
</tr>
<tr>
<td></td>
<td>org.nagios.DiskCheck</td>
<td>OK</td>
<td>10-15-2010 09:57:59</td>
<td>0d 19h 33m 48s</td>
<td>1/3</td>
<td>DISK_OK - free space: /var 60474 MB (63% inode=99%)</td>
</tr>
</tbody>
</table>
Live broker monitoring

**Destinations filtering**

- **topic:**
- **queue:**

**Destinations list:**
- all destinations selected

**network**
- vtb-generic-14.cern.ch

Graph showing produced and consumed data over time.
Common messaging use cases

- **Gather**
  - Wide set of nodes send back data which is collected in a DB for further processing

- **Broadcast**
  - Sensors publishing information that can be used by external alarm systems

- **Task Queues**
  - Tasks sent to the queue and executed on distributed worker nodes

- **Remote Procedure Call (RPC)**
  - Distributed query
CERN Control Room

- **LHC monitoring**
  - High availability required (no messages = beam dumped)
    - Achieved by simple configuration
  - 41M messages per day
  - 370 GB incoming data & 2.6 TB outgoing data
  - 7-10k consumers, 7k topics
Service Availability Monitoring (SAM)

- Monitoring of reliability and availability of the distributed computing infrastructure
- 27 national level Nagios servers
  - Will grow to ~40 in next 3 months
- 4 experiment Nagios servers (Alice, Atlas, CMS, LHCb)
- Data received also from OSG
- 700,000 test results/day
- Long-lived and short-lived connections
Service Availability Monitoring (SAM)

- GGUS
- Data Warehouse
- Alarm System

ActiveMQ

Nagios

Site

RSV

Site

Open Science Grid

A Messaging Infrastructure for WLCG - 18
SAM – CE tests

Nagios -> ActiveMQ -> WMS

WMS -> CE

CE -> Worker Nodes
Site Wide Area Testing (SWAT)

- Distributed Tests
- SAM
- ActiveMQ
- Consumer
- Summary Exporter
- Database
- Web frontend

- 2.6M messages per day
- short-lived connections
APEL – replacement of R-GMA transport mechanism

- APEL nodes
- Consumer
- Database
- Nagios
- APEL tests

- 6k messages per day
- Encrypted messages
- x509 authentication and authorisation
„Designing the Next Generation Grid Information Systems” by Laurence Field (Today: 17:30)
• DQ2 Tracing data
  • Data send on every DQ2 command – files transferred, performance data, …
  • 1-1.5M messages/day of ~2-3KB
  • Clients on WN, VOBOX, …
  • Short-lived connections
  • Low reliability requirements – can accept some loss
• LFC -> SE: propagation of changes in the file permissions
• SE -> LFC: propagation of information about lost files
CERN batch service

- Use case: Analyze usage of resources (CPU, IO, Network, ...) by the experiment jobs on 35000 CPU cores
- Solution: Measure live (per minute) per-job usage statistics, and collect them via messaging for subsequent analysis and aggregation.
- Expected rate: ~600 messages per second
• https://tomtools.cern.ch/confluence/display/MIG

• http://activemq.apache.org/

• „Enterprise Integration Patterns”, http://www.eaipatterns.com/

• Contact us: tom-developers@cern.ch
Messaging is a key technology for WLCG