Distributed Monitoring of Web Services using OpenNMS

Dr Craig Gallen EngD C.Eng MBA

entimOSS Limited
6 Burnett Close
Bitterne Park
Southampton
Hampshire
England
SO18 1JD

Email: craig.gallen@entimoss.com
cgallen@opennms.org

Mobile: +44 (0) 7789 938012

The OpenNMS Group, Inc.
220 Chatham Business Drive
Pittsboro
NC 27312
United States

e-mail sales@opennms.com
www.opennms.com

Phone: +1 919-533-0160
Fax: +1 503-961-7746
End user Services (Apps) are a mash-up of web services accessed through standard and proprietary protocols; HTTP, REST, SOAP, RSS etc.

Services hosted in ‘Cloud’ designed to scale through addition of VM resources.

Underlying physical infrastructure:
- Commodity hardware
- Geographical Diversity
- Rapid Churn
- Network Connectivity

Gartner: Market for public cloud services to grow from $1.7 billion in 2009 to $12.4 billion in 2014.
Management as another web service

Management integration enabled through standard interfaces
OpenNMS Strategic Overview

- **OpenNMS Core Overview**
  - Introducing the OpenNMS Project
  - Network and service problem management
  - SLA reporting

- **OpenNMS Remote Poller Project**
  - Distributed End User Experience Monitoring of web services and mash ups.

- **Cloud Services Broker Project**
  - Proof of concept for commercial cloud service monitoring

- **TM Forum Interfaces Project**
  - Model driven interface design
  - Semantic Interfaces

- **OpenNMS Infrastructure Projects**
  - IPv6
  - Cassandra
  - OSGi
The OpenNMS Project

- **OpenNMS**
  - Open Network Management System
  - OpenNMS is the world's first Enterprise and Carrier grade network management platform developed under the open source model.

- **Technology**
  - Written in Java
  - Packaged for Windows, Linux and most Unix distributions
  - Proven scalability
    - 300,000 data points every 5 minutes
    - automatically discover core nodes with 5000+ interfaces

- **Websites**
  - [www.opennms.org](http://www.opennms.org)
  - [http://sourceforge.net/projects/opennms/](http://sourceforge.net/projects/opennms/)
Wide community of commercial users

- Papa Johns Pizza [http://www.papajohns.com/]
- Minnesota Children's Hospital [http://www.childrensmn.org/]
- Oregon State University [http://www.oregonstate.edu]
- Permanente Medical Group [www.permanente.net]
- Myspace [www.myspace.com]
- Ocado [www.ocado.com]
- FreshDirect [http://www.freshdirect.com]
- Fox TV (Australia) [http://www.foxtel.com.au]
- BBC Monitoring [www.monitor.bbc.co.uk]
- FastSearch [http://www.fastsearch.com/]
- New Edge Networks [http://www.newedgenetworks.com/]
- Rackspace [http://www.rackspace.com]
- Swisscom Eurospot [http://www.swisscom-eurospot.com]
- Wind Telecomunicazioni SpA (Italy) [http://www.wind.it]

- And many more - 4000 downloads per month
OpenNMS history

- We've been around since 1999 and were registered on Sourceforge in March of 2000 (by comparison NetSaint, the first name of Nagios, was registered just two months earlier).

Mar /April 2000
OpenNMS Announced & Code released
10 OcuLAN employees
50 contributors

Sept 2004
OpenNMS group lunched
3 staff
16 contributors

Jan 1999
Steve Gilles & Brian Weaver
Prototype Bluebird

May 2002
OcuLAN forks OpenNMS
Tarus Balog becomes Maintainer (Sortova Consulting)

2009
over 100 OpenNMS group customers
4000 complete downloads / month
6 staff
35 core developers (OGP)

Open Source Product Maturity
Community and Governance

- **User community**
  - There are around 1000 people subscribed to the discuss list, but when I (Tarus Balog) teach classes I find that less than 10% of the people in the class actually use the discuss list, so my guess is that the active user community is probably closer to 10,000 people.

- **Developer Community**
  - We have 35 developers with commit access to the repository.

- **Assets**
  - Licence GPL
  - The IPR is owned by The OpenNMS Group, Inc.
  - OpenNMS Trademark owned by The OpenNMS Group

- **Governance**
  - The community is managed by The Order of the Green Polo.
  - All active OGP members have a vote on the direction of the project, but there is no charter and no one restricts what can and can't go into OpenNMS, as long as it is good.
  - For example, the OTRS integration that Jonathin Sartin (Ocado/Truephone) did was pretty much on his own.

- **Foundation**
  - We would like to create a foundation separate to the OpenNMS Group when sponsorship is available to do so.

DEV-JAM Atlanta July 2008
OpenNMS Infrastructure Management

In development
• IPv6 – to be released corresponding with IPv6 day 6 Jun 2006
• OSGi infrastructure

Service Discovery and data Collection
- Windows WMI
- Windows Services (SNMP-based)
- Citrix
- DHCP
- DNS
- Domino IIOP
- FTP
- General Purpose (script based)
- HTTP
- HTTPS
- ICMP
- IMAP
- JBOSS (JMX)
- JDBC
- JSR160 (JMX)
- K5
- LDAP
- Microsoft Exchange
- MX4J
- Notes HTTP
- NSClient (Nagios Agent)
- NRPE (Nagios Plugin Executor)
- NTP
- POP3
- Radius
- SMB
- SMTP
- SNMP
- SSH
- TCP
OpenNMS Core Functionality

Problem Management Work Flow

- **Event Collection**
  - OpenNMS can record all event occurrences

- **Alarm Correlation**
  - OpenNMS uses an Alarm Mechanism to convert configurable ‘alarm raising traps’ or ‘alarm clearing traps’ into a manageable alarm cycle.
  - User configured ‘automations’ can process the alarm list for more sophisticated analysis.
  - Jboss Rules correlation engine for more sophisticated down stream alarm suppressing.

- **User Notifications and scheduled escalation**
  - OpenNMS supports multiple users and an Notification escalation mechanism between users. If a severe event is detected (such as a major alarm), this generates a Notification which is escalated over time through a list of users if it is not acknowledged. The system can also generate external paging, emails or instant messaging messages to attract attention to a notification.

- **Trouble ticket integration**
  - If the basic escalation mechanism is not enough, OpenNMS also has a Trouble ticket interface for integrating with a number of trouble ticket systems including open source trouble ticket implementations, RT and OTRS.
Performance and SLA Management

- **Performance Data Collection and Management**
  - Like other network management tools such as Nagios or Cricket, OpenNMS stores performance data in RRD files. It can use RRDTOOL to do the collection, but the preferred library is JRobin which is a Java implementation of RRD.
  - OpenNMS has MIBs already installed for most large vendors equipment but users can add their own configurations. The user community often share this work and experience of new equipment.
  - However unlike these tools, all of the scheduling of data collection is controlled by a Java process entirely within OpenNMS which makes the solution very scalable.
  - Data can be collected from a variety of sources; SNMP polling and trap management, Ascii Syslog messages, TL1, JMX. there is also an integration with Nagios to allow the use of Nagios plugins. OpenNMS has also been integrated with Snort.

- **Data visualisation**
  - OpenNMS presents performance data as graphs. These graphs can also be exported in the form of performance reports.
  - Threshold events. OpenNMS can generate Threshold crossing alarms based on changes in the data. OpenNMS also performs synthetic transactions to test the availability of services on nodes. This can be done centrally or through a distributed collection of remote rollers as described above.

- **Service Quality Management**
  - SLA Alarms can be escalated based upon threshold crossing events. Every performance data collection point can be assigned a lo/high threshold with hysteresis to avoid ‘bouncing’ alarms
Original Use Case

Remote needs to ensure connectivity to central hosts

1. Remote client downloads OpenNMS Remote Poller from OpenNMS using java web start
2. OpenNMS Remote Poller polls central services using synthetic transactions
3. Remote Poller sends results to OpenNMS
Remote Poller Installation

- Web Start a Remote Poller
Remote Poller Management

Definitions

• **Application**
  — collection of services to monitor
  — Services on URI's

• **Remote Poller Location**
  — Remote locations from which to monitor
    one or more applications

Response times from remote locations

![Graph showing response times](image)

<table>
<thead>
<tr>
<th>Area</th>
<th>Definition Name</th>
<th>ID</th>
<th>Hostname</th>
<th>IP Address</th>
<th>Status</th>
<th>Last Check-in Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>NJ</td>
<td>55</td>
<td>p-brane</td>
<td>192.168.211.132</td>
<td>Started</td>
<td>9/2/10 1:07:17 AM</td>
</tr>
<tr>
<td>USA</td>
<td>NY</td>
<td>60</td>
<td>p-brane.local</td>
<td>10.71.8.25</td>
<td>Started</td>
<td>9/2/10 1:07:14 AM</td>
</tr>
</tbody>
</table>
Visualisation: Poller Location

- Remote Poller locations – for each Application
Visualisation: Application Location

- Application locations
Managing Within The Cloud

Cloud as a Service

Provisioning of service

Provisioning and reporting of SLA

Cloud Services Broker Proof of Concept

- [http://www.tmforum.org/CloudServiceBroker/8437/home.html](http://www.tmforum.org/CloudServiceBroker/8437/home.html)
  - Sponsor BT
  - Participants
    - OpenNMS, Infonova, Comptel, square Hoop
  - BT Paper IM2011 Dublin 2011
Flexibility through Standard Interfaces

- **Service definition can be arbitrarily complex but provisioning of Instance of service is easy**
  - Advanced service correlation uses complex OpenNMS features. Needs expert definition.
    - Alarm reduction
    - Correlation
    - Event Translation based on external data lookup
  - Service Instance provisioned through UI or through XML API from another system

- **Open Source and virtualisation**
  - Allows deployment of multiple OpenNMS instances with radically different configurations
  - Interfaces allow information to be shared through a hierarchy of systems

Flexible Deployment Strategies
Same framework for all interfaces
• Can be extended to new managed technologies (Modelling: 3G, Ethernet etc)
• Can be extended to different management technologies (WSDL, JAVA, etc)
• Can be offered to other standards organisations (ITU-T, 3GPP etc)
• Can be used internally by SP’s, SI’s, Equipment vendors
Open Source RI Design Goals

- **Major Sponsors**
  - DT, Telcordia, HP, Nortel, Ciena
  - Leveraging open source projects
    - Tigerstripe
    - OpenNMS
  - Delivering libraries for Interface Specification Development and Implementation

---

- **Eclipse Tigerstripe Workspace**
  - Maven 2
  - TIP Model Project
  - Java Implementation Plugin
  - Transitory Jar in local repository

- **Java Implementation Project**
  - Maven 2
  - CTK, RI, Libraries
  - Interface library can be used by any OSS Component

- **OpenNMS**
  - Reference Implementation in OpenNMS
Summary

- An overview of OpenNMS
- Application management
  — Remote Poller Project
- Virtual infrastructure management
  — Cloud Services Broker
- Sharing Management information
  — TM Forum Interface Program
- The OpenNMS project is at the forefront of research and standardisation of large scale web infrastructure management
Thank You For Listening

Any Questions?