Hybrid cloud monitoring

Jernej Vičič, Andrej Brodnik
UP IAM
CLASS/Monitor

• Provides monitoring of IaaS by service provider and client
  – (internal and external monitoring)
• Provides tracking of adherence to SLA agreement
  – In real-time
  – notification by E-mail and SMS
• Supports hybrid cloud
• The observed cloud platforms:
  – VMWare
  – OpenStack
  – HyperV
CLASS/Monitor

• We implemented two monitoring systems:
  – BBmon
  – Cacti based solution (proof of concept)
• BBmon tested on three different IaaS – VMWare, OpenStack, and HyperV;
• Cacti based solution supports VMWare only
CLASS/Monitor

• Encountered problems:
  – different degree of maturity of IaaS,
  – differences in supported functionality,
  – APIs are based on different concepts
  – OpenStack lacks monitoring of the host what we solved with Ganglia

• We plan to augment the system with failover servers for higher availability
Monitoring properties comparison

• Reference feature set:
  • observed needed features
  • literature,
  • independent research.

• Observed feature sets (all platforms),
• intersection is almost full (except OpenStack).
Monitoring properties comparison

• Openstack feature set
  • acceptable in all feature sets except:
    • low level (mostly hardware) monitoring.

• Proposed solution:
  • Ganglia monitoring system,
  • opensource,
  • simple interface,
  • vast palette of installations.
Consideration of possible architectures: using a gateway

Properties:
• The control system communicates via the same protocol with different virtual platforms.
• No need to install special software on each virtual platform.
• A Gateway (translation interface) has to be implemented for each virtual platform.
Consideration of possible architectures: using a gateway

Pros:
• all controlled systems are presented in the same way
  • easier to scale,
  • similar approach is known in network management (SNMP).

Cons:
• Complex implementation.
Consideration of possible architectures: using a direct interface

Properties:
• The control system communicates directly through the interface.
• The access to these interfaces is possible via additional software.
• Modules can also be implemented as plug-ins.
Consideration of possible architectures: using a direct interface

**Pros:**
- easier debugging,
- better error control,
- simpler implementation.

**Cons:**
- difficult maintenance,
- poor scalability.
The implemented architecture of CLASS/Monitor

KC Class

SLA Monitoring/Management

Cacti

BBmon

vSphere API

Powershell

vSphere API

vCenter

API

ESX

API

Hyper-V

API

OpenStack

API

KVM

Ganglia

API

SCVMM

WMI

vSphere API
The testing environment

Monitoring System
- Cacti on VM
- Bbmon on VM

VMware cloud
- Machine nr1
- Machine nr2

Hyper-v cloud
- Machine nr3
- Machine nr4

OpenStack cloud
- Machine nr5
- Machine nr6
KC Class SLA Monitoring
BBmon Monitoring and BBmonSLA
BBmon Monitoring

• Each or group of metrics represent BBmon test (different states)
Virtual Infrastructure Monitoring

- VMware: Perl SDK API
- OpenStack: XML-RPC with Ganglia
- HyperV: WMI and client with PowerShell
Monitoring

• State
• Metrics with trends
• Thresholds
BBmonSLA

- SLA reports
- SLO objects
- Time periods exclusion
- Scheduling
BBmonSLA – SLA Report Configuration
BBmonSLA – SLO Objects

• Duration of events:

\[
SLA_{\text{level}} = \frac{T_{1\text{\_OK}} + T_{2\text{\_OK}} + T_{3\text{\_OK}} - T_{2\text{\_alarm}}}{\text{Total}_{\text{SLA}}_{\text{Time}} - T_{2\text{\_alarm}}}
\]

T2_alarm - excluded

• Duration of single event:

\[
T_{2\text{\_alarm}} \text{ is the longest}
\]

• Number of events:

3 alarms
# BBmonSLA – SLA Report

## SLA Report
Report time: 23.10.2012, 08:32

### SLA OPENSTACK - FROM: 01.08.2012, 01:00:00 TO: 17.08.2012, 18:00:00

<table>
<thead>
<tr>
<th>SLO</th>
<th>Object</th>
<th>Interval</th>
<th>Duration</th>
<th>Measured Value</th>
<th>Threshold</th>
<th>Status</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability - conn</td>
<td>KC_openstack - conn</td>
<td>382200 s</td>
<td>117 s</td>
<td>99.97%</td>
<td>99.5%</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Availability - conn</td>
<td>OpenStack Host 10.40.10.0 - conn</td>
<td>382200 s</td>
<td>118 s</td>
<td>99.97%</td>
<td>99.5%</td>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

### SLA OPENSTACK - FROM: 01.08.2012, 01:00:00 TO: 17.08.2012, 18:00:00

<table>
<thead>
<tr>
<th>SLO</th>
<th>Object</th>
<th>Interval</th>
<th>Duration</th>
<th>Measured Value</th>
<th>Threshold</th>
<th>Status</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability - http</td>
<td>KC_openstack - conn</td>
<td>382200 s</td>
<td>117 s</td>
<td>99.97%</td>
<td>97%</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Availability - http</td>
<td>OpenStack Host 10.40.10.0 - conn</td>
<td>382200 s</td>
<td>118 s</td>
<td>99.97%</td>
<td>97%</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Availability - http</td>
<td>OpenStack Host 10.40.10.0 - http</td>
<td>382200 s</td>
<td>34375 s</td>
<td>99.97%</td>
<td>97%</td>
<td>ALERT</td>
<td></td>
</tr>
</tbody>
</table>

### SLA OPENSTACK - FROM: 01.08.2012, 01:00:00 TO: 17.08.2012, 18:00:00

<table>
<thead>
<tr>
<th>SLO</th>
<th>Object</th>
<th>Interval</th>
<th>Duration</th>
<th>Measured Value</th>
<th>Threshold</th>
<th>Status</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability - ocpu</td>
<td>KC_openstack - conn</td>
<td>382200 s</td>
<td>117 s</td>
<td>99.97%</td>
<td>95%</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Availability - ocpu</td>
<td>OpenStack Host 10.40.10.0 - conn</td>
<td>382200 s</td>
<td>118 s</td>
<td>99.97%</td>
<td>95%</td>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>