OpenSAF and VMware from the Perspective of High Availability

Ali Nikzad, Ferhat Khendek, Maria Toeroe
Concordia University, Ericsson

SVM’2013 – Zurich – October 2013
Outline

- Introduction & Objectives
- Some Background
- Testbed, Failures and Metrics
- The baseline architectures
  - Measurements
  - Analysis
- Architectures combining OpenSAF and virtualization
- Conclusion
Introduction & Objectives

- Service availability and continuity have become important requirements in several domains

- High availability (demanded in some domains) is defined as at least 99.999% availability which is maximum of 5.26 minutes of downtime in a year
  - Including scheduled downtime for upgrade for instance

- The computing world is moving toward cloud services and cloud computing
  - Virtualization is an important aspect

- Virtualization has many advantages
Introduction & Objectives

- Evaluate and position virtualization and the SAForum middleware with respect to each other
  - Using OpenSAF and VMware in the test-bed
  - Analyzing the measurements

- Determine pros and cons of using virtualization in/for high availability (HA)

- Propose a solution(s) that combines the benefits from each
Background – SAForum and OpenSAF

**SAForum** is a consortium of telecommunication and computing companies developing standards for enabling highly available services and applications.

**OpenSAF** is an open source project focused on Service Availability implementing the SAForum service specifications, among others AMF, SMF and IMM.
Background-AMF

- AMF is one of the most important services defined by SAForum
- AMF configuration: AMF logical entities ... 
  - AMF Node
  - AMF Cluster
  - Component
    - SA-Aware
    - Non-SA-Aware
    - Pre-instantiable
    - Non-pre-instantiable
  - Component Service Instance (CSI)
  - Service Unit (SU)
  - Service Instance (SI)
  - Service Group (SG)
    - Redundancy Models
  - Application
  - 2N
  - N+M
  - N-Way
  - N-Way-Active
  - No-Redundancy
Background-Virtualization and VMware

- Virtualization is the separation of a resource or request for a service from the underlying physical delivery of that service

- VMs are hosted on software called hypervisor
  - Native (bare metal)
  - Hosted (non-bare metal)

- VMware is one of the leading companies in providing virtualization solutions
Background - VMware Availability solutions

- VMware HA
- VMware FT
Testbed, Failures and Metrics

- **Test-beds:**
  - 5 nodes cluster
  - Case study application:
    - VLC for Video streaming

- **Failures**
  - VLC component failure (application failure)
  - VM failure
  - Physical node failure

  - More than 45 sets of measurements for metrics in different architectures for different failures
Testbed – Metrics and Failures

- **Metrics**
  - **Qualitative (criteria)**
    - Complexity of using the solution
    - Supported Redundancy Models
    - Scope of failure
    - Service continuity
    - Supported platforms
  
  - **Quantitative Metrics**
    - Reaction Time
    - Repair Time
    - Recovery Time
    - Outage Time

- Diagram:
  - Event: Failure
  - Time: Outage Time
  - Event: First reaction
  - Time: Reaction Time
  - Event: Resumption or Restart of service
  - Time: Recovery Time
  - Event: Faulty unit repaired
  - Time: Repair Time
Baseline architectures

SA-Aware and Non-SA-Aware versions of VLC on physical and virtual nodes
VM availability with VMware HA

SAF based architecture

VMware vSphere cluster
Baseline architectures

- OpenSAF on VMware HA
- VMs’ lifecycle depends on the hypervisor’s availability mechanisms
- VM migration is not considered
### Baseline architectures

<table>
<thead>
<tr>
<th>Architectures</th>
<th>VLC component failure</th>
<th>VM failure</th>
<th>Node Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenSAF on physical nodes with SA-Aware VLC component</td>
<td>√</td>
<td>Not applicable</td>
<td>√</td>
</tr>
<tr>
<td>OpenSAF on physical nodes with Non-SA-Aware VLC component</td>
<td>√</td>
<td>Not applicable</td>
<td>√</td>
</tr>
<tr>
<td>OpenSAF on virtual nodes with SA-Aware VLC component (with/without VMware HA enabled)</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>OpenSAF on virtual nodes with Non-SA-Aware VLC component (with/without VMware HA enabled)</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>VMware HA</td>
<td>Not detectable</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>
Baseline architectures: Measurements

Outage due to VLC component failure (application failure)
Baseline architectures: Measurements

Outage due to Virtual Machine failure

![Graph showing outage measurements for different VM configurations]
Baseline architectures: Measurements

Outage due to Node Failure
## Repair of the faulty unit

<table>
<thead>
<tr>
<th></th>
<th>Repair of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Failed component</td>
</tr>
<tr>
<td><strong>OpenSAF on Standalone machine</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>OpenSAF in VM</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>VMware HA</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>OpenSAF in VM + HA</strong></td>
<td>Yes</td>
</tr>
</tbody>
</table>
Baseline architectures: Analysis

- **SAF based architectures**
  - Service continuity with SA-Aware components and better reaction time
  - Support application failure detection and recovery
  - Less overhead
  - No repair for the cluster node

- **VMware HA**
  - Repair of the node is supported
    - but it is very long (~100s)
  - Inevitable 15% to 30% overhead due to the VM compared to the physical host deployment
  - No support for application failure

- **Baseline combined architecture**
  - The advantages of both architectures
    - but, Very long repair time (~100s) due to no redundancy of the VM
Baseline architectures: Analysis

The reaction time to the failure in OpenSAF is much faster

So, why not manage the VMs’ life cycle with OpenSAF
Architectures combining OpenSAF and virtualization: Availability in non-bare-metal hypervisor
Architectures combining OpenSAF and virtualization: Availability in non-bare-metal hypervisor

Comparison of different architectures for VM failure

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Reaction</th>
<th>Repair</th>
<th>Recovery</th>
<th>Outage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware HA</td>
<td>72.166</td>
<td>27.166</td>
<td></td>
<td>99.332</td>
</tr>
<tr>
<td>OpenSAF with ESXi (VMware HA manages the VMs)</td>
<td>1.905</td>
<td>107.90</td>
<td>0.047</td>
<td>1.953</td>
</tr>
<tr>
<td>The new availability management in non-bare-metal hypervisor</td>
<td>3.449</td>
<td>3.73</td>
<td>0.056</td>
<td>3.505</td>
</tr>
</tbody>
</table>

*Times are in seconds*
Architectures combining OpenSAF and virtualization: Availability in non-bare-metal hypervisor

Failure of the SA-Aware VLC component in different architectures

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Reaction</th>
<th>Repair</th>
<th>Recovery</th>
<th>Outage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware HA</td>
<td>Not covered</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenSAF with no virtualization</td>
<td>0.009</td>
<td>0.136</td>
<td>0.046</td>
<td>0.055</td>
</tr>
<tr>
<td>OpenSAF with ESXi (VMware HA manages the VMs)</td>
<td>0.013</td>
<td>0.243</td>
<td>0.068</td>
<td>0.081</td>
</tr>
<tr>
<td>The new availability management in non-bare-metal hypervisor</td>
<td>0.012</td>
<td>0.848</td>
<td>0.580</td>
<td>0.592</td>
</tr>
</tbody>
</table>

The non-bare-metal hypervisor imposes delays on some of the measured times like repair and recovery. The bare-metal hypervisor can potentially resolve these delays.
Architectures combining OpenSAF and virtualization: Availability in bare-metal hypervisor
Architectures combining OpenSAF and virtualization: Availability in bare-metal hypervisor... VM Failure
Architectures combining OpenSAF and virtualization: Availability in bare-metal hypervisor ... Physical Node Failure
Conclusion

- **From baseline architectures to combinations**
  - Use the powerful availability management of OpenSAF and benefiting from virtualization
  - Increased outage time in the new deployment is because of the non-bare-metal hypervisor
    - The bare-metal architecture can possibly fix it…
  - Improve the repair time of a failed VM through management of the VM life cycle
  - Cover the different types of hypervisors (bare-metal and non-bare-metal)

- **The VM’s life cycle management**
  - is not limited to a specific solution
  - can support other hypervisors like XEN and Linux KVM because of using similar and standard interfaces like libvirt
Thank you for your attention!