OpenSAF and VMware from the Perspective of High Availability

Ali Nikzad, **Ferhat Khendek**   Maria Toeroe

Concordia University   Ericsson

SVM'2013 – Zurich – October 2013

I

Outline

▸ **Introduction & Objectives**

▸ **Some Background**

▸ **Testbed, Failures and Metrics**

▸ **The baseline architectures**

  ▸ Measurements

  ▸ Analysis

▸ **Architectures combining OpenSAF and virtualization**

▸ **Conclusion**

Friday, October 18, 13

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**

Friday, October 18, 13

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**
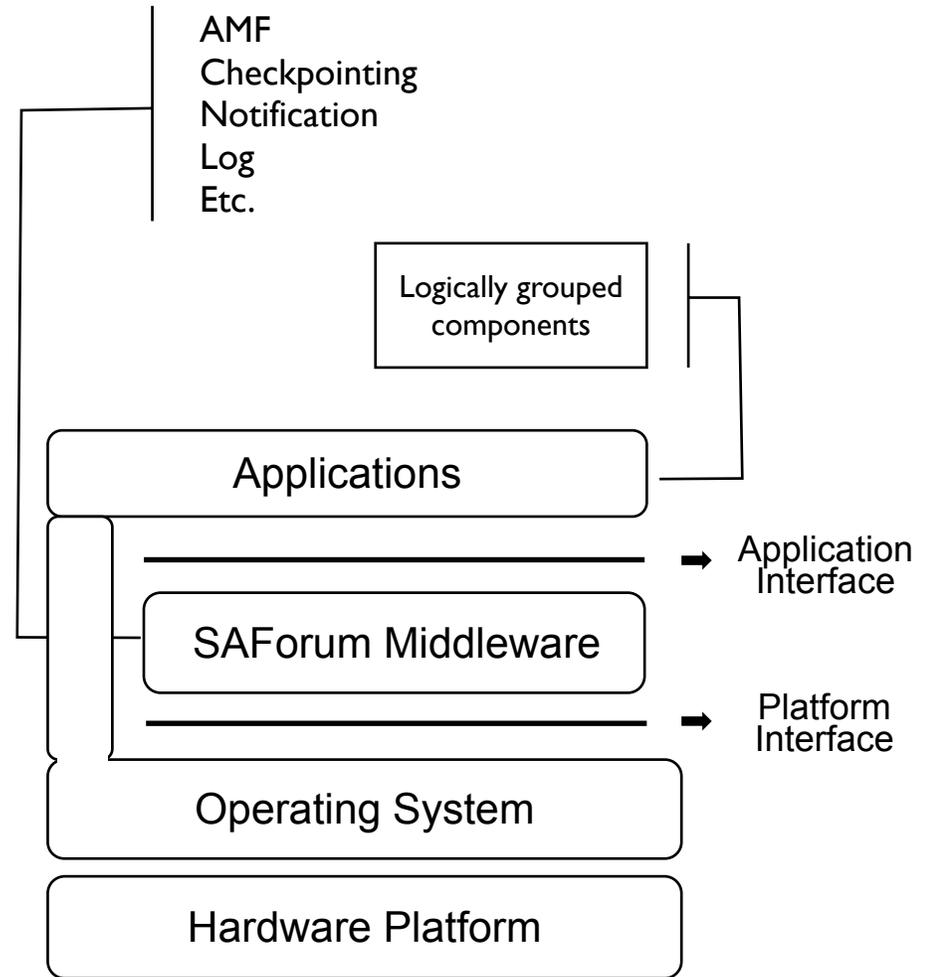
Friday, October 18, 13

Background – SAForum and OpenSAF

SAForum is a consortium of telecommunication and computing companies developing standards for enabling highly available services and applications
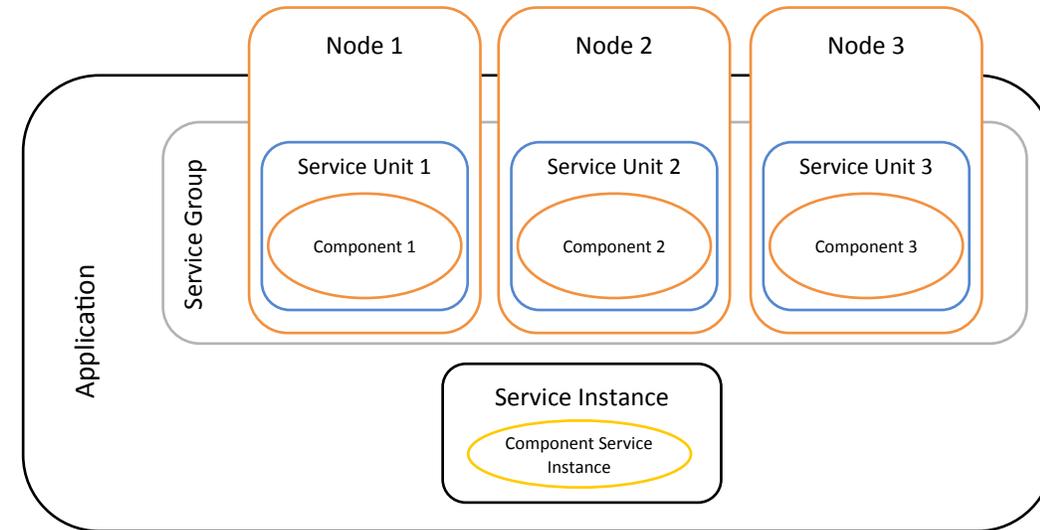
OpenSAF is an open source project focused on Service Availability implementing the SAForum service specifications, among others AMF, SMF and IMM.

AMF
Checkpointing
Notification
Log
Etc.

Logically grouped components

Applications

→ Application Interface

SAForum Middleware

→ Platform Interface

Operating System

Hardware Platform

Friday, October 18, 13

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 — | 2N | N+M | N-Way | N-Way-Active | No-Redundancy |
  - Application

	Node 1	Node 2	Node 3
	Service Unit 1	Service Unit 2	Service Unit 3
	Component 1	Component 2	Component 3

Application / Service Group

Service Instance — Component Service Instance

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**

Friday, October 18, 13

Background - VMware Availability solutions

▸ **VMware HA**
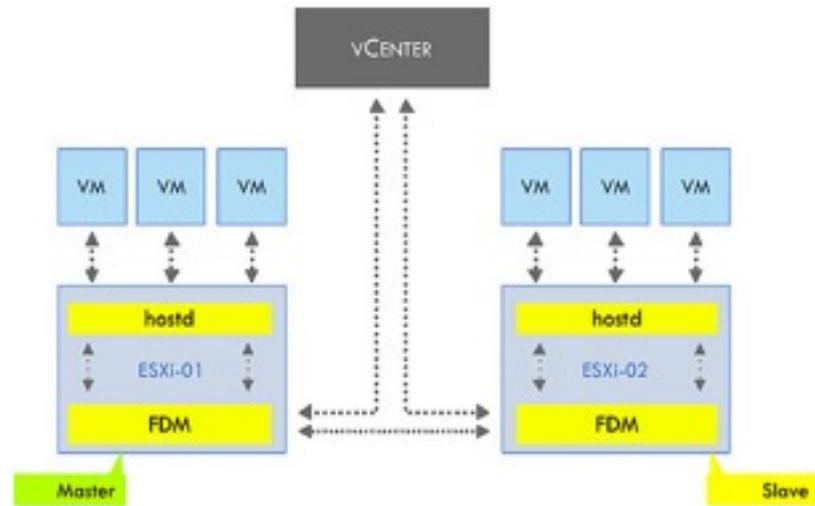
▸ **VMware FT**
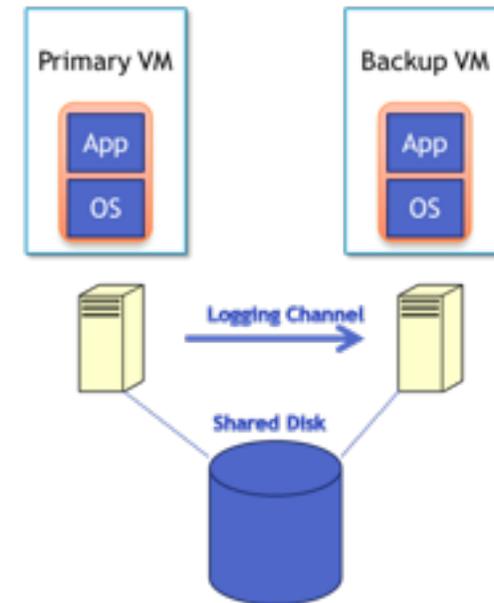
Testbed, Failures and Metrics

- **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

Friday, October 18, 13

Testbed – Metrics and Failures

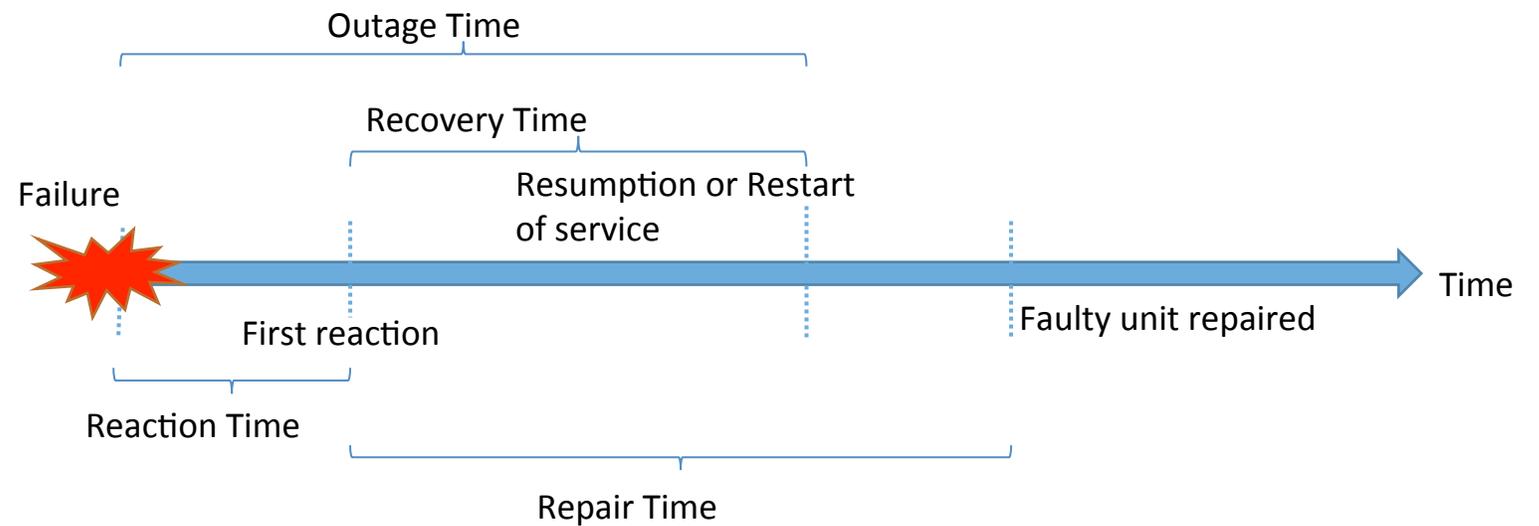▸ **Metrics**

   ▸ 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

Outage Time

Recovery Time

Resumption or Restart of service

Failure

First reaction

Reaction Time

Time

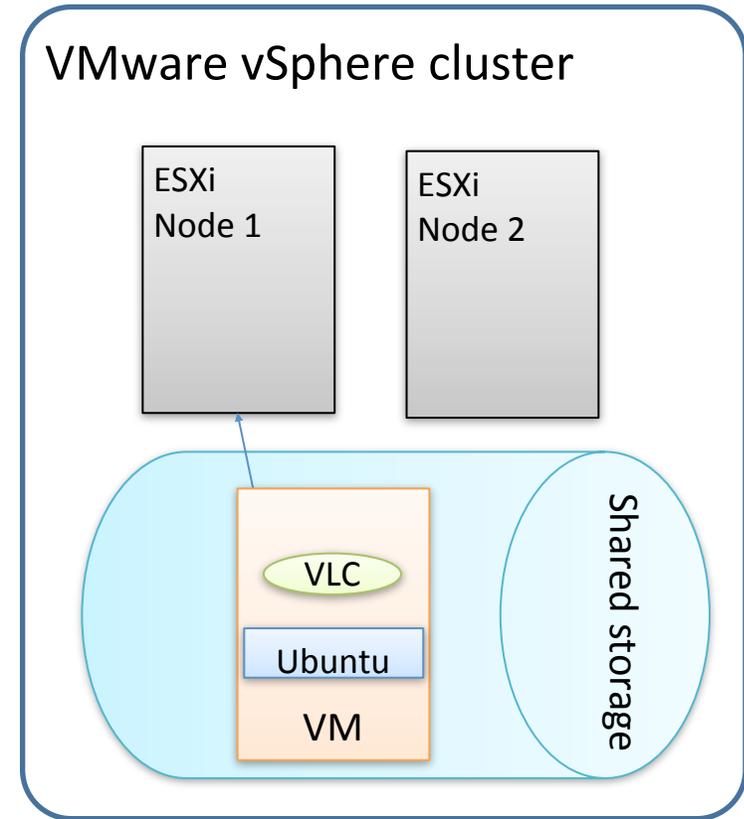Faulty unit repaired

Repair Time

Baseline architectures

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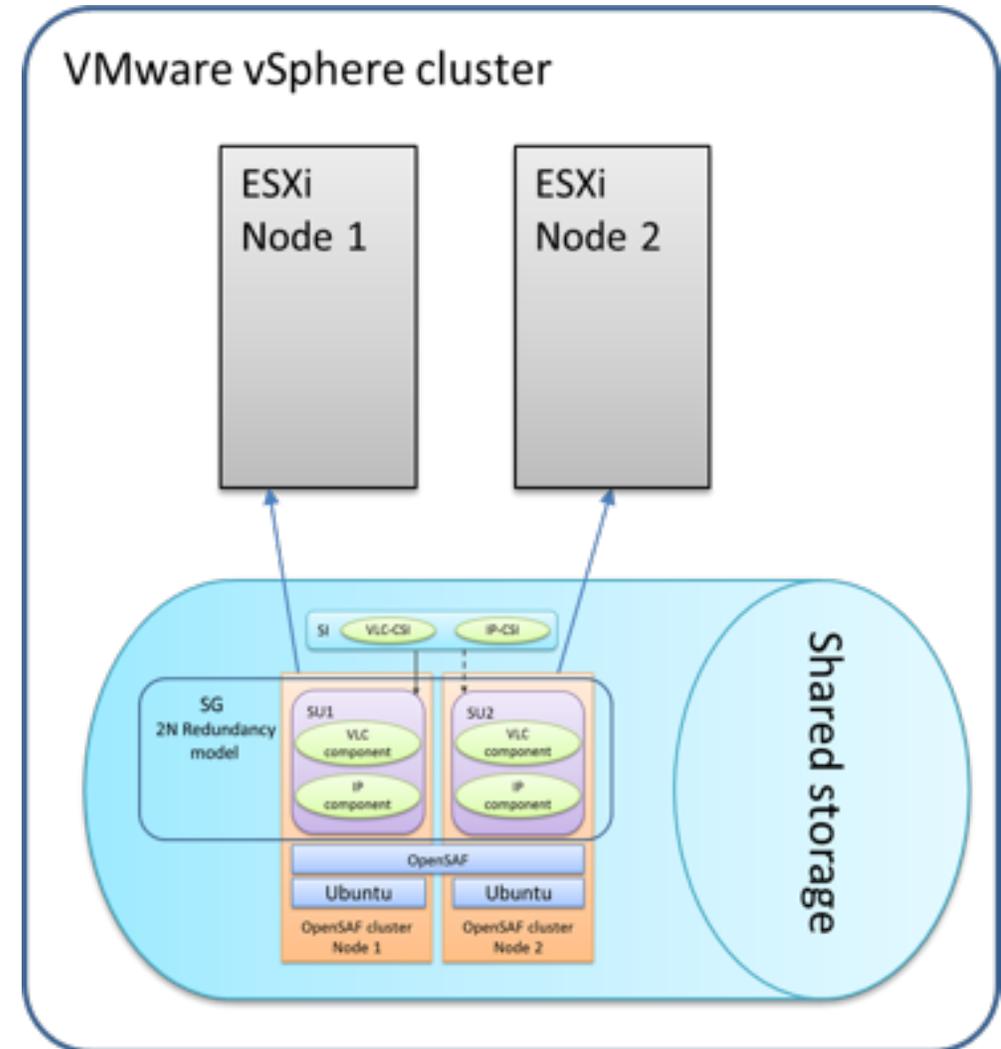VM availability with VMware HA
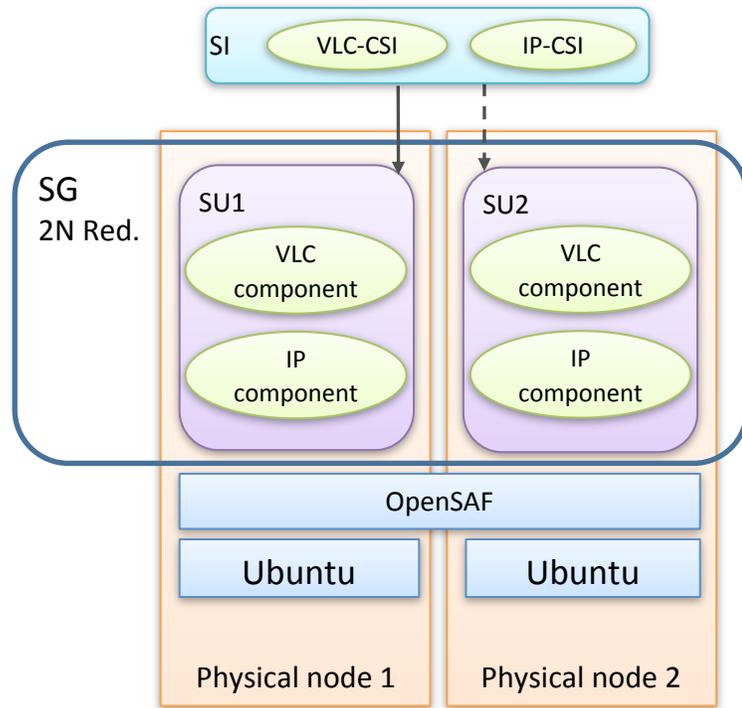
SAF based architecture

VMware HA

Baseline architectures

- OpenSAF on VMware HA

- VMs' lifecycle depends on the hypervisor's availability mechanisms

- VM migration is not considered

Baseline architectures

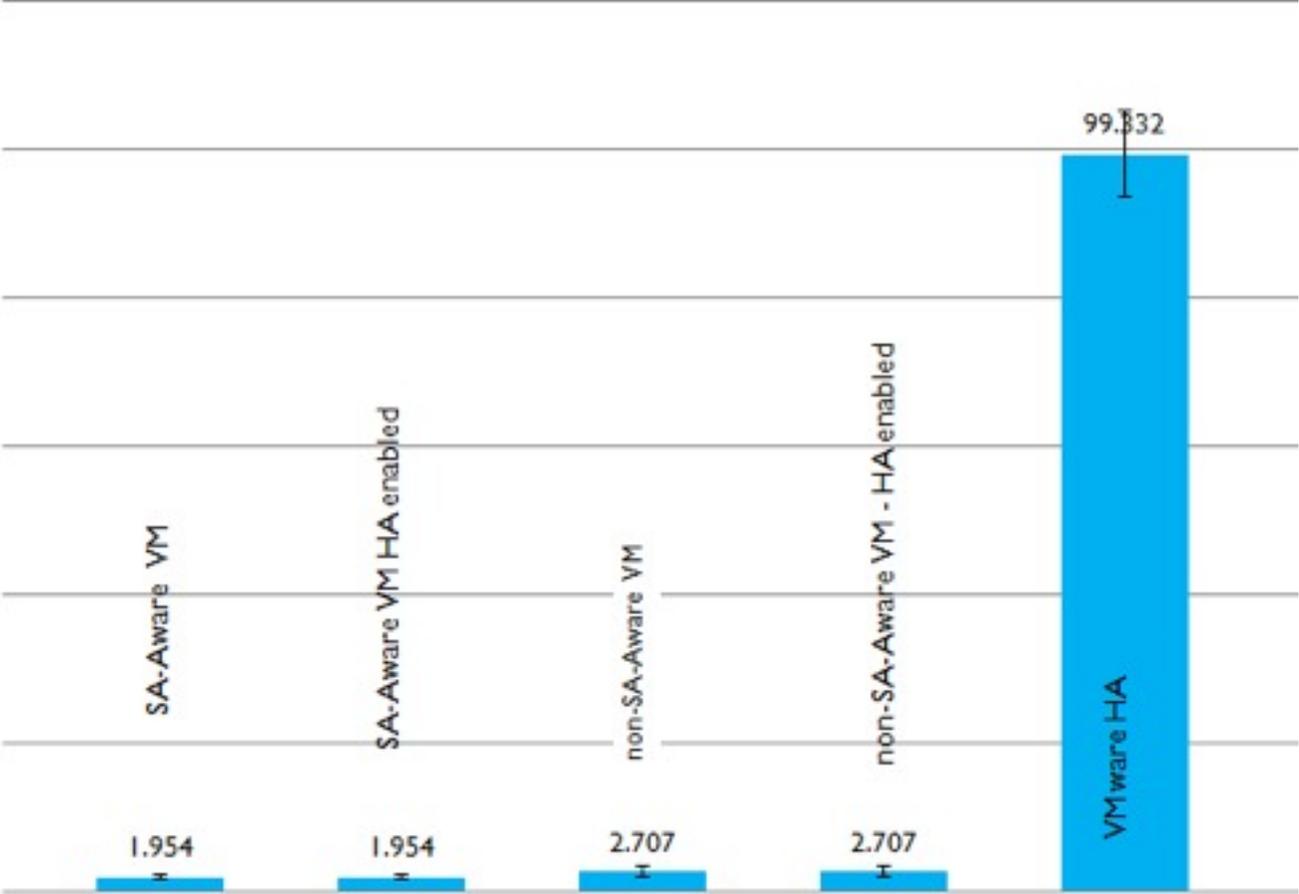| Architectures	VLC component failure	VM failure	Node Failure
OpenSAF on physical nodes with SA-Aware VLC component	√	Not applicable	√
OpenSAF on physical nodes with Non-SA-Aware VLC component	√	Not applicable	√
OpenSAF on virtual nodes with SA-Aware VLC component (with/without VMware HA enabled)	√	√	√
OpenSAF on virtual nodes with Non-SA-Aware VLC component (with/without VMware HA enabled)	√	√	√
VMware HA	Not detectable	√	√

Baseline architectures: Measurements

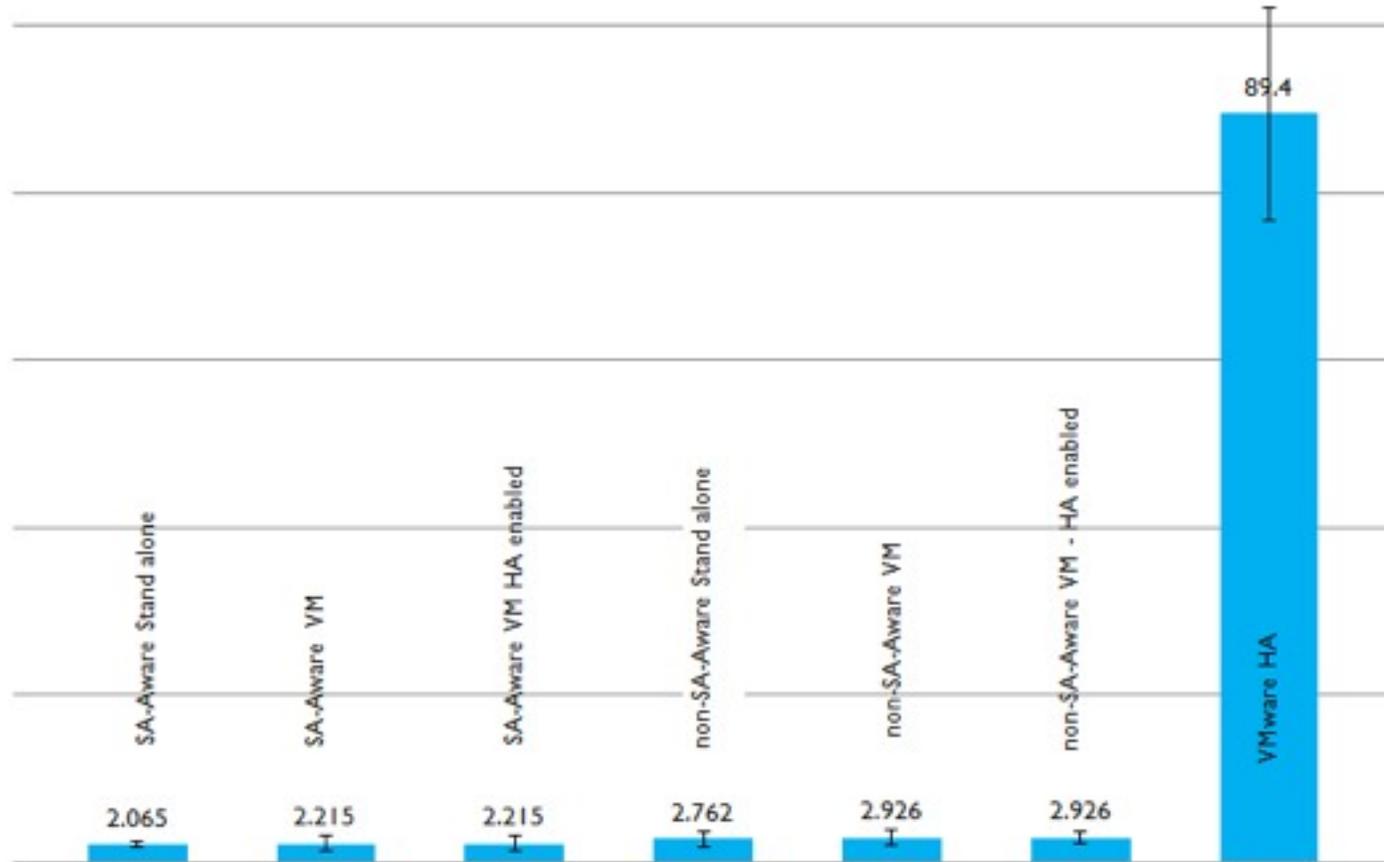## Outage due to VLC component failure (application failure)
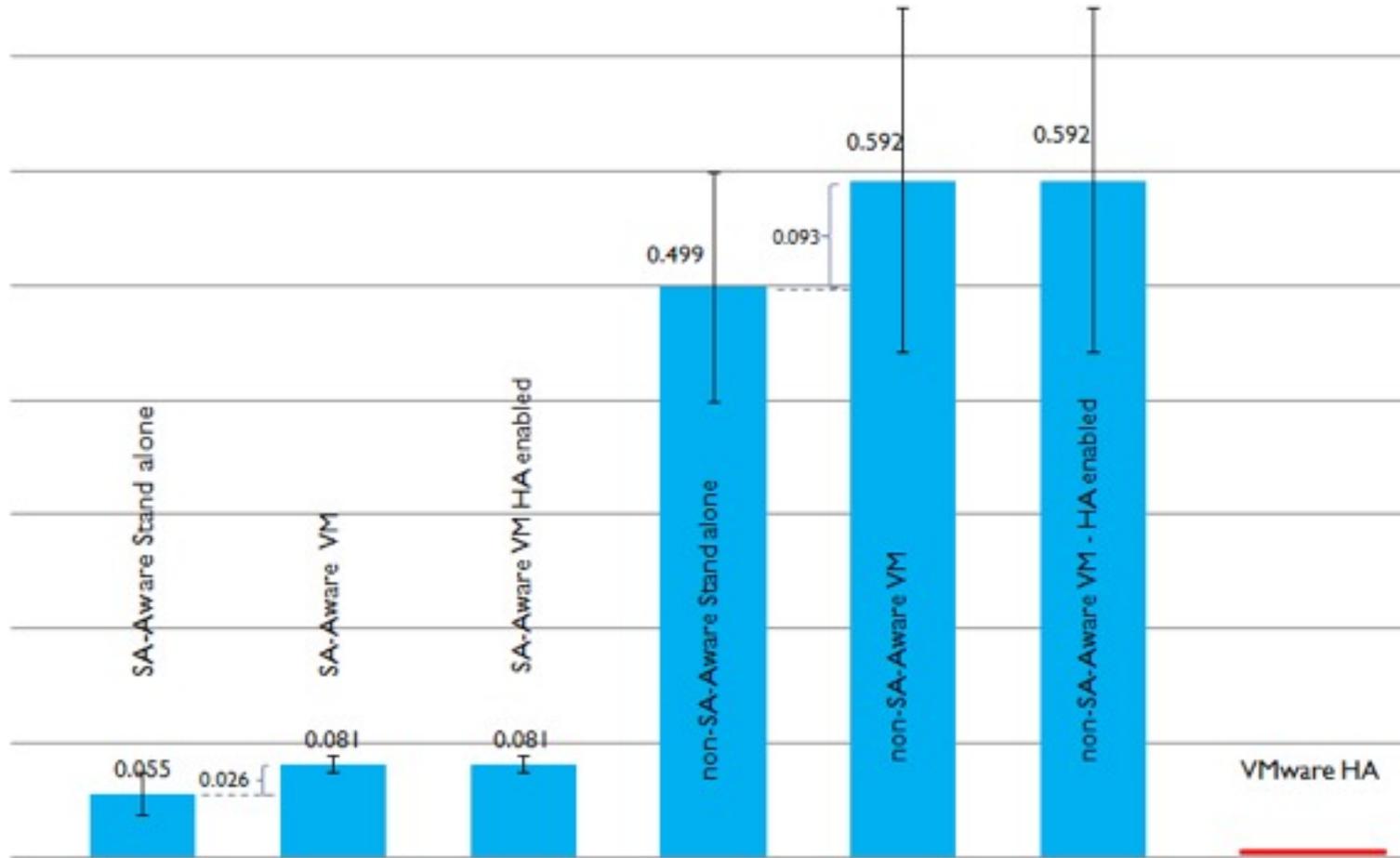
Baseline architectures: Measurements

Outage due to Virtual Machine failure

Friday, October 18, 13

Baseline architectures: Measurements

Outage due to Node Failure

Friday, October 18, 13

Baseline architectures: Measurements

Repair of the faulty unit

	Repair of		
	Failed component	**Failed VM**	**Failed Node**
OpenSAF on Standalone machine	Yes	-	No
OpenSAF in VM	Yes	No	No
VMware HA	No	Yes (restarting the VM on another host)	Yes(restarting the VM on another host)
OpenSAF in VM + HA	**Yes**	**Yes**	**Yes**(restarting the VM on another host)

Friday, October 18, 13

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
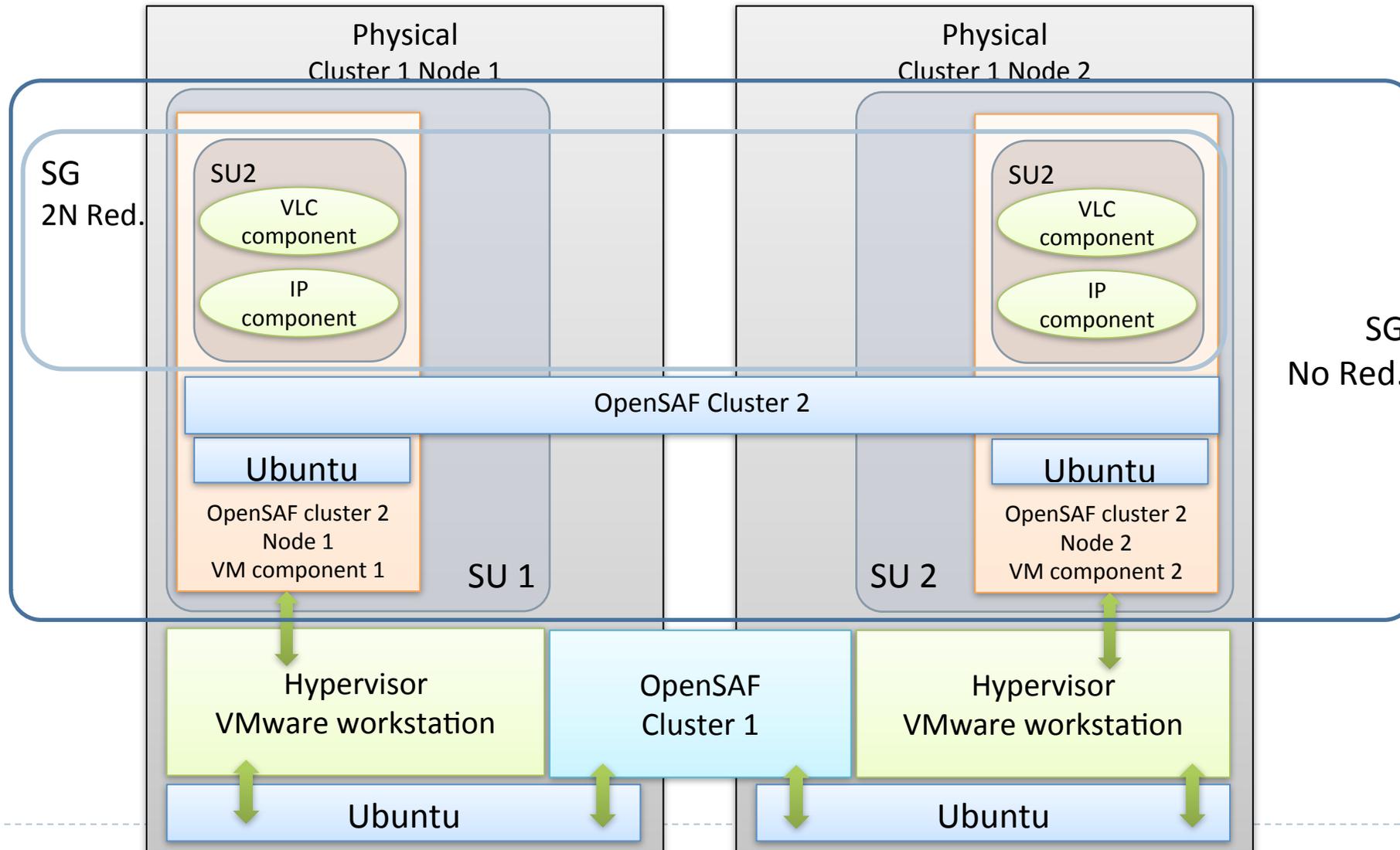 - ✗ but, Very long repair time (~100s) due to no redundancy of the VM

Friday, October 18, 13

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

	Reaction	Repair	Recovery	Outage
VMware HA	72.166	27.166		**99.332**
OpenSAF with ESXi (VMware HA manages the VMs)	1.905	**107.90**	0.047	**1.953**
The new availability management in non-bare-metal hypervisor	3.449	**3.73**	0.056	**3.505**

*Times are in seconds

Friday, October 18, 13

Architectures combining OpenSAF and virtualization: Availability in non-bare-metal hypervisor

Failure of the SA-Aware VLC component in different architectures

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

The non-bare-metal hypervisor imposes delays on some of the measured times like repair and recovery

bare-metal hypervisor can potentially resolve these delays

Architectures combining OpenSAF and virtualization: Availability in bare-metal hypervisor
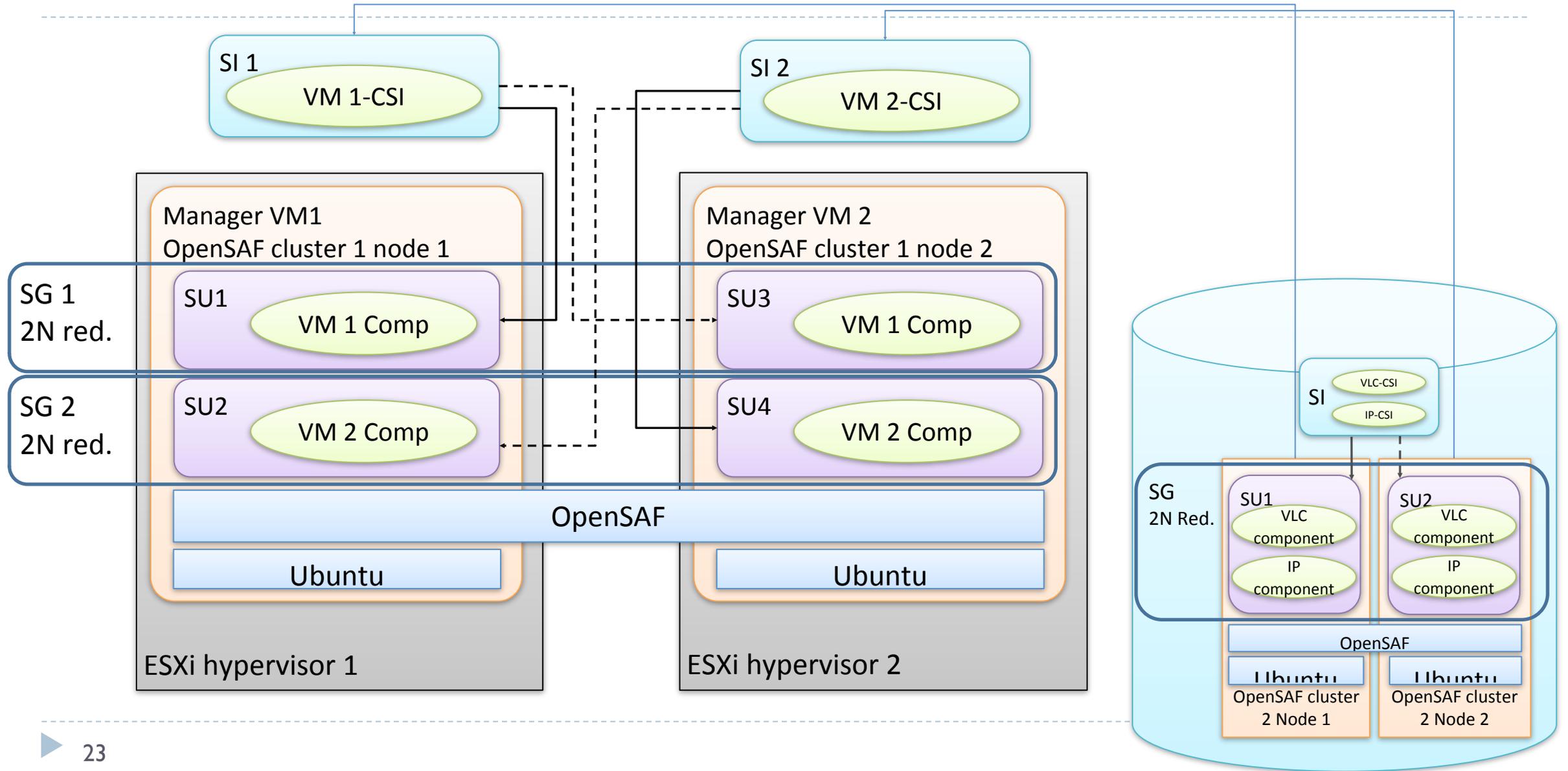
Architectures combining OpenSAF and virtualization: Availability in bare-metal hypervisor... VM Failure
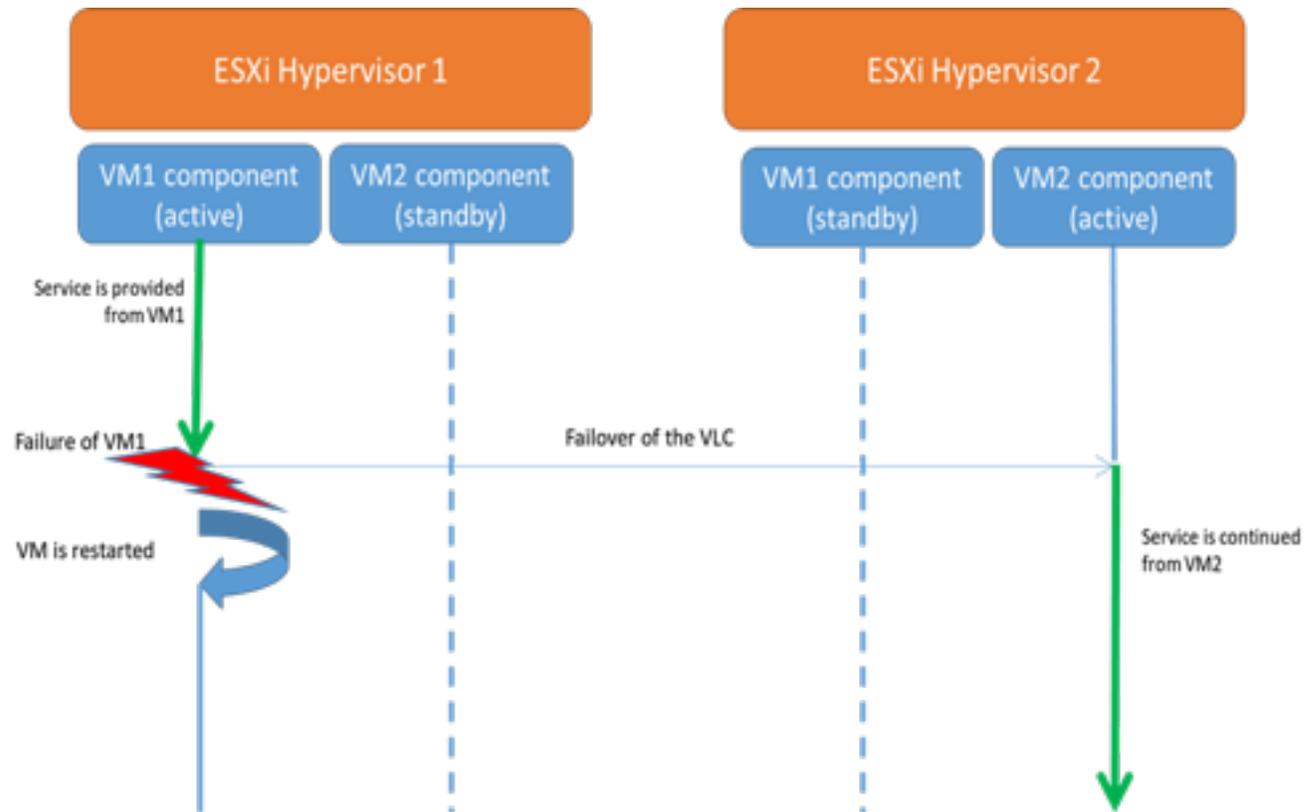
Friday, October 18, 13

Architectures combining OpenSAF and virtualization : Availability in bare-metal hypervisor ... Physical Node Failure

Friday, October 18, 13

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

  ‣

Friday, October 18, 13

Thank you for your attention!