Towards an Architecture for Monitoring Private Cloud

Shirlei Aparecida de Chaves, Rafael Brundo Uriarte, Carlos Becker Westphall

Federal University of Santa Catarina
Networks and Management Laboratory

Outline

1. ABSTRACT
2. INDROTUCTION
3. BACKGROUND

3.1. Cloud Computing Service Models

3.2. Cloud Computing Deployment Models

3.3. Cloud Computing Standards

Outline

4. MONITORING ARCHITECTURE AND PCMONS

4.1. Architecture

4.2. Implemantation

5. CASE STUDY

6. RELATED WORK

6.1. Grid Monitoring

6.2. Cloud Monitoring

Outline

7. KEY LESSONS LEARNED

7.1. Related to Test-Bed Preparation

7.2. Design and Implementation

7.3. Standardization and Available Implementations

8. CONCLUSIONS AND FUTURE WORKS

9. REFERENCES

1. ABSTRACT

This presentation describes:

- our experience with a private cloud;
- the design and implementation of a Private Cloud MONitoring System (PCMONS); and
- its application via a case study for the proposed architecture, using open source solutions and integrating with traditional tools like Nagios.

2. INTRODUCTION

- Cloud computing provides several technical benefits including flexible hardware and software allocation, elasticity, and performance isolation.
- Cloud management may be viewed as a specialization of distributed computing management, inheriting techniques from traditional computer network management.

2. INTRODUCTION

The intent of this presentation is to:

- Provide insight into how traditional tools and methods for managing network and distributed systems can be reused in cloud computing management.
- Introduce a Private Cloud MONitoring System (PCMONS) we developed to validate this architecture, which we intend to open source.

2. INTRODUCTION

- Help future adopters of could computing make good decisions on building their monitoring system in the cloud.
- We chose to address private clouds because they enable enterprises to reap cloud benefits while keeping their mission-critical data and software under their control and under the governance of their security policies.

3. BACKGROUND

3.1. Cloud Computing Service Models

- Software-as-a-Service (SaaS): The consumer uses the provider's applications, which are hosted in the cloud.
- Platform-as-a-Service (PaaS): Consumers deploy their own applications into the cloud infrastructure. Programming languages and applications development tools used must be supported by the provider.

3. BACKGROUND

3.1. Cloud Computing Service Models

- Infrastructure-as-a-Service (IaaS): Consumers are able to provision storage, network, processing, and other resources, and deploy and operate arbitrary software, ranging from applications to operating systems.
- This presetation focuses on IaaS model.

3. BACKGROUND

3.2. Cloud Computing Deployment Models

- Public: Resources are available to the general public over the Internet. In this case, “public” characterizes the scope of interface accessibility.
- Private: Resources are accessible within a private organization. This environment emphasizes the benefits of hardware investments.

3. BACKGROUND

3.2. Cloud Computing Deployment Models

- Community: Resources on this model are shared by several organizations with a common mission.
- Hybrid: This model mixes the techniques from public and orivate clouds. A private cloud can have its local infrastructure supplemented by computer capacity from public cloud.

3. BACKGROUND

3.3. Cloud Computing Standards

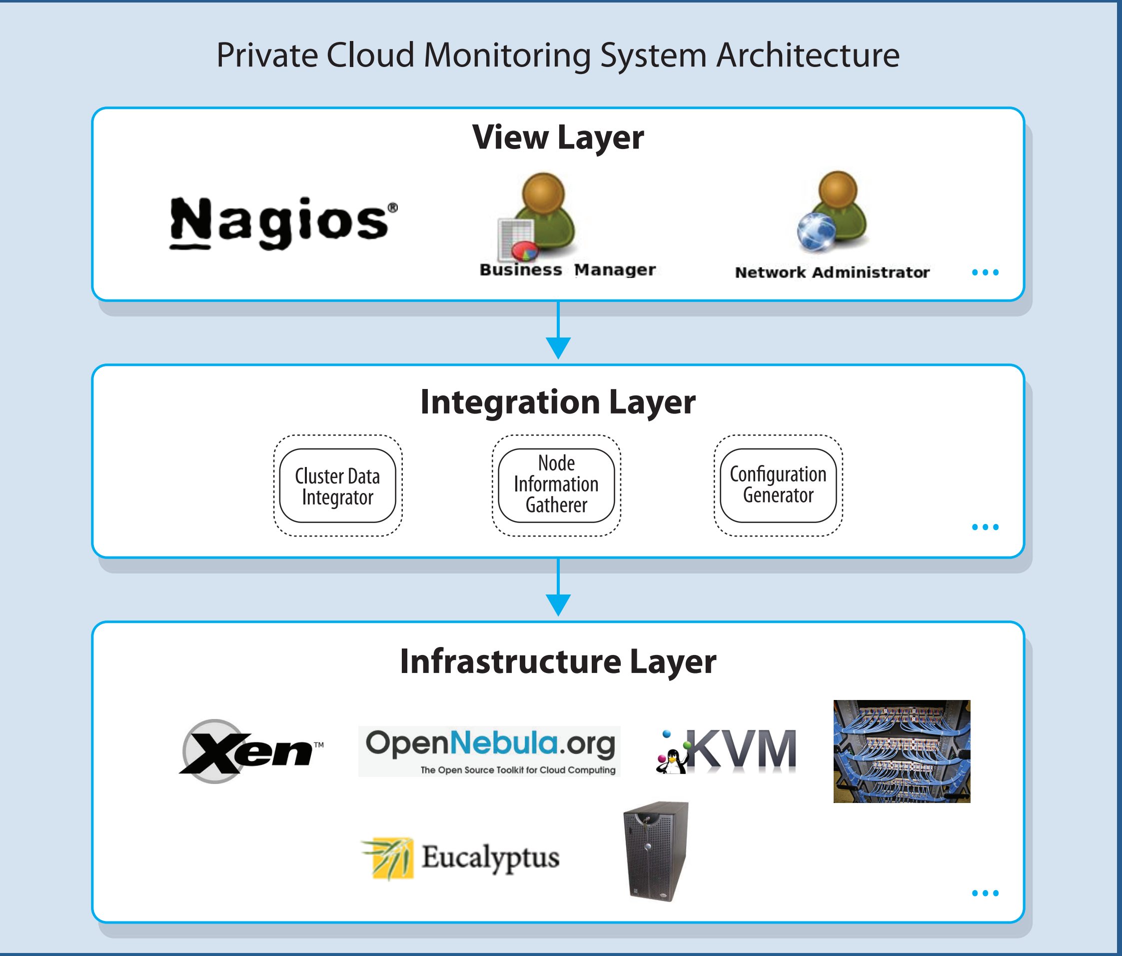
- Open Cloud Computing Interface: This Open Grid Forum group has a focus on specifications for interfacing “*aaS” cloud computing facilities.
- OCCI in Eucalyptus, OCCI in OpenStack, OCCI in OpenNebula...

3. BACKGROUND

3.3. Cloud Computing Standards

- Open Cloud Standards Incubator: This initiative, from Distributed Management Task Force (DMTF), focuses on interactions between cloud environments, their consumers, and developers.
- Example of document: “Use cases and Interactions for Managing Clouds”.

4. MONITORING ARCHITECTURE AND PCMONS



4. MONITORING ARCHITECTURE AND PCMONS

4.1. Architecture

- Three layers address the monitoring needs of a private cloud.
- Infrastructure layer:
- Basic facilities, services, and installations, such as hardware and networks;
- Available software: operating system, applications, licenses, hypervisors, and so on...

4. MONITORING ARCHITECTURE AND PCMONS

4.1. Architecture

- Integration layer:
- The monitoring actions to be performed in the infrastructure layer must be systematized before passed to the appropriate service running in the integration layer.
- The integration layer is responsible for abstracting any infrastructure details.

4. MONITORING ARCHITECTURE AND PCMONS

4.1. Architecture

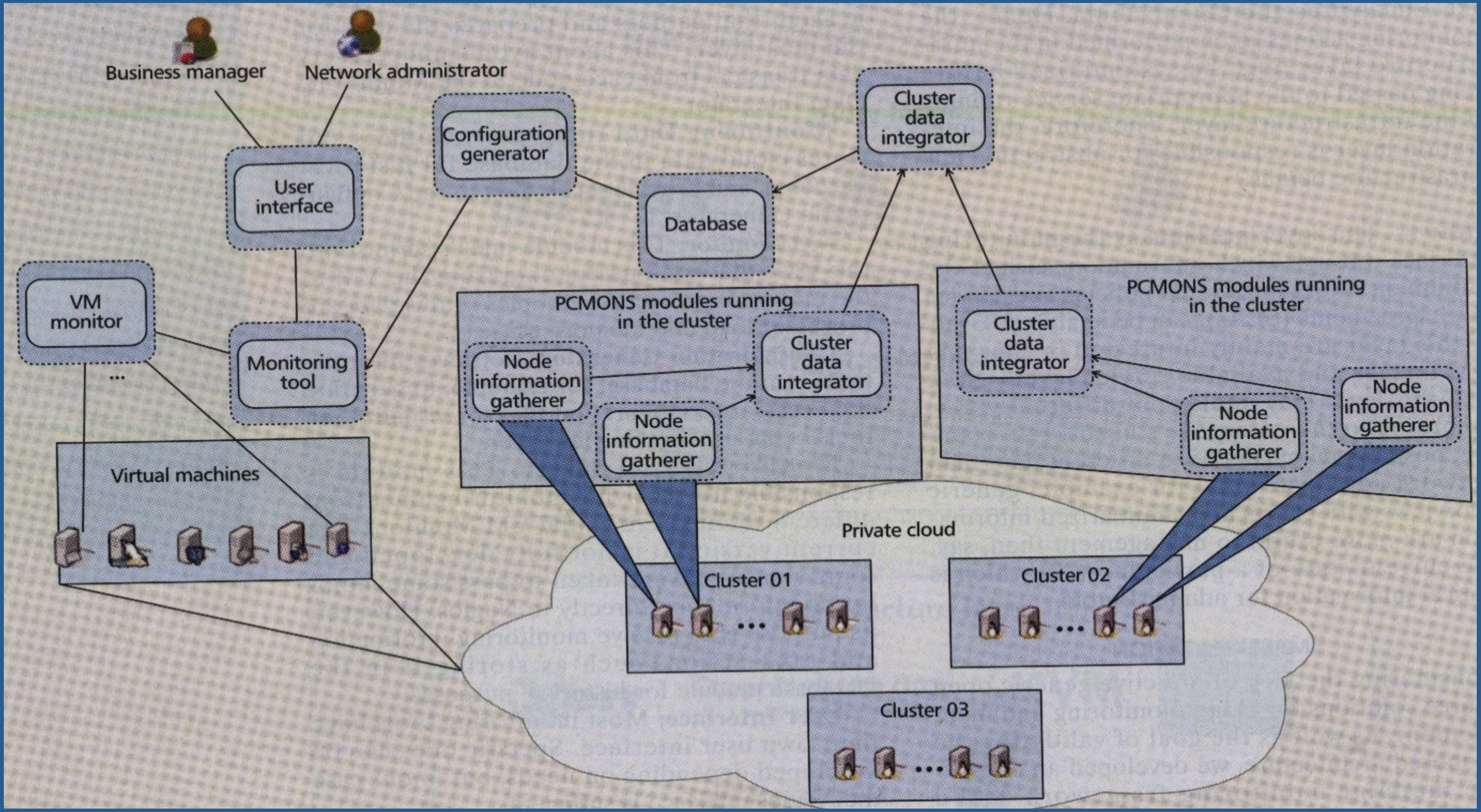
- View layer:
- This layer presents as the monitoring interface through which information, such as the fulfillment of organizational policies and service level agreements, can be analyzed.
- Users of this layer are mainly interested in checking VM images and available service levels.

4. MONITORING ARCHITECTURE AND PCMONS

4.2. Implementation

- The current PCMONS version acts principaly on the integration layer, by retrieving, gathering, and preparing relevant information for the visualization layer.
- The system is divided into the modules presented in the next figure and described below.

A typical deployment scenario for PCMONS



4. MONITORING ARCHITECTURE AND PCMONS

4.2 Implementation

- Node Information Gatherer: This module is responsible for gathering local information on a cloud node. It gathers information about local VMs and sends it to the Cluster Data Integrator.
- Cluster Data Integrator: It is a specific agent that gethers and prepares the data for the next level.

4. MONITORING ARCHITECTURE AND PCMONS

4.2 Implementation

- Monitoring Data Integrator: Gathers and stores cloud data in the database for historical purposes, and provides such data to the Configuration Generator.
- VM Monitor: This module injects scripts into the VMs that send useful data from the VM to the monitoring system.

4. MONITORING ARCHITECTURE AND PCMONS

4.2 Implementation

- Configuration Generator: Retrieves information from the database to generate configuration files for visualization tools.
- Monitoring Tool Server: Its purpose is to receive monitoring information and take actions such as storing it in the database module for histirical purposes.

4. MONITORING ARCHITECTURE AND PCMONS

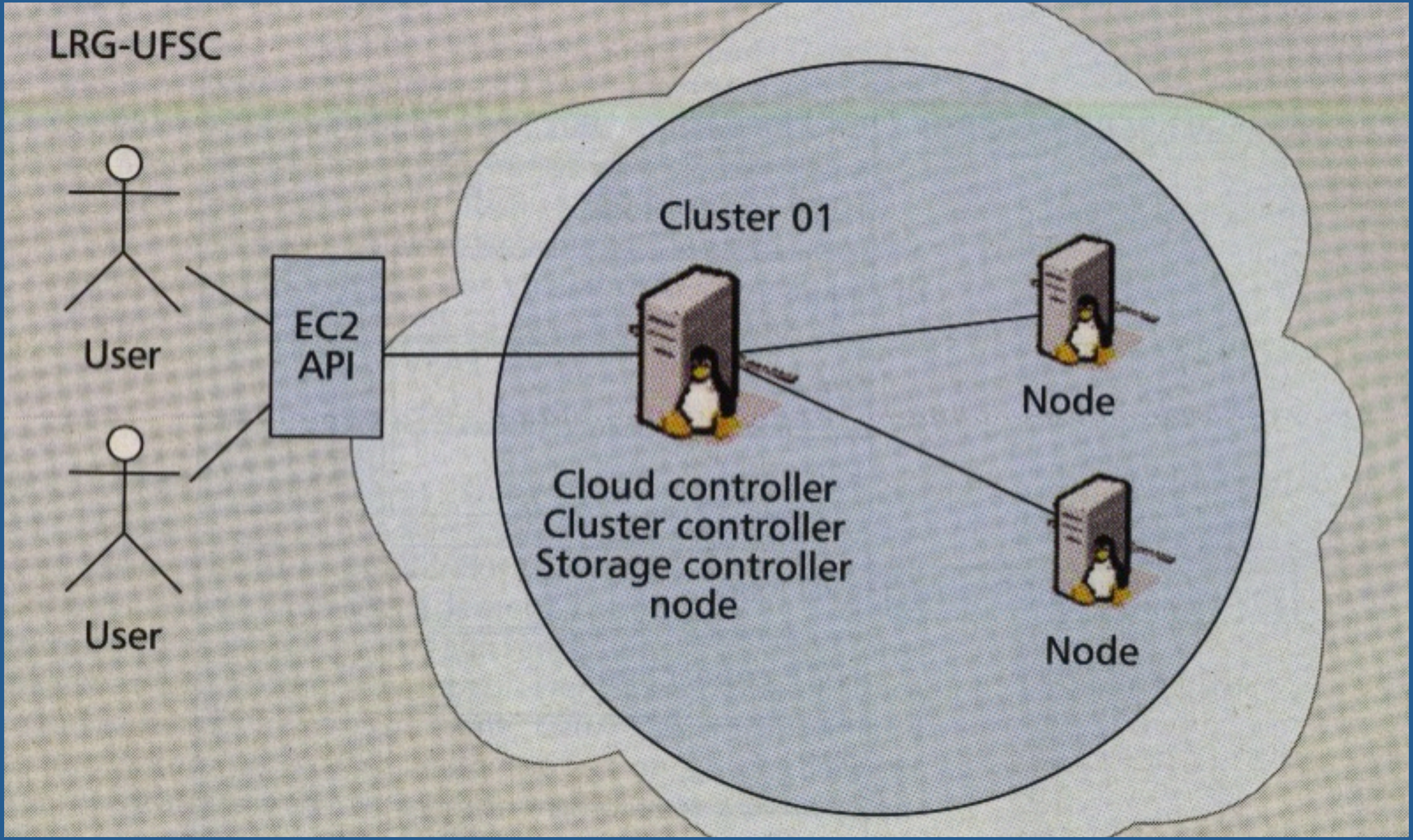
4.2 Implementation

- User Interface: Most monitoring tools have their own user interface. Specific ones can be developed depending on needs, but in our case the Nagios interface is sufficient.
- Database: Stores data needed by Configuration Generator and the Monitoring Data Integrator.

5. CASE STUDY

- We built an environment where VM images are available for users that instantiate a web server, thus simulating web hosting service provision.
- Instantiated VMs are Linux servers providing a basic set of tools, acting as web hosting servers.
- Apache Web Server, PHP language, SQLite.

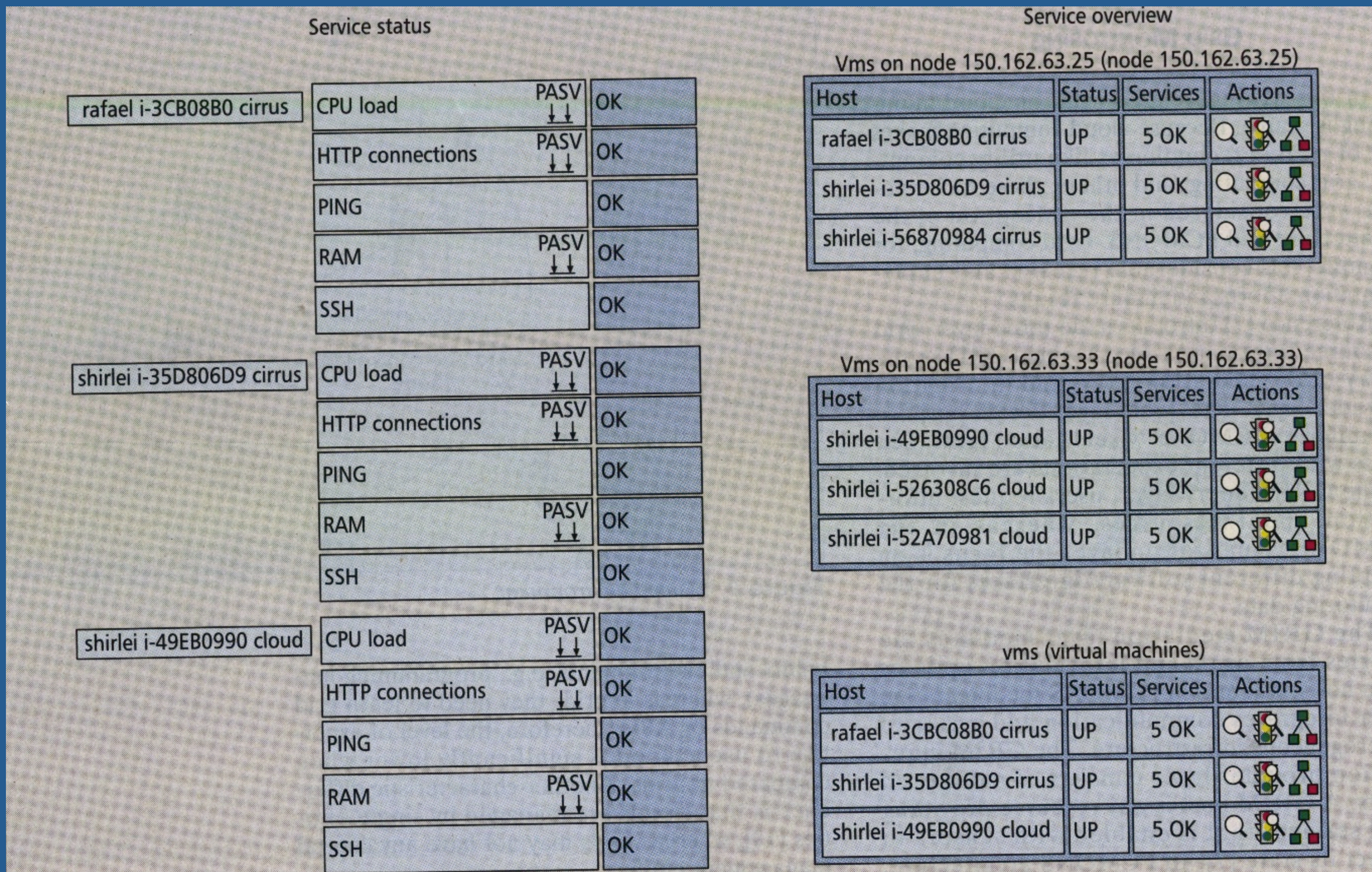
Testbed environment



5. CASE STUDY

- Open SUSE was chosen as the operating system of the physical machines (Xen and YaST).
- Eucalyptus (interface compatible with Amazon's EC2). VM images were downloaded from the Eucalyptus website.
- VM Monitor module is injectec into the VM during boot, allowing data monitoring.

Representative Nagios interface of the monitored cloud services



5. CASE STUDY

- First column shows object names (VM, PM, ROUTERS...). VM names are an aggregation of user name, VM ID, and name of PM where the VM is running.
- The other two columns show service names and their status (OK, Warning, Critical).
- It shows host group created by PCMONS and VM/VP mapping.

6. RELATED WORK

6.1. Grid Monitoring

- Reference [7] introduces the three-layer Grid Resource Information Monitoring (GRIM).
- Several design issues that ashould be considered when constructing a Grid Moitoring System (GMS) are preented in [8]. We have selected some and correlated then with PCMOMS.

6. RELATED WORK

6.1. Grid Monitoring

- Reference [9] identifies some differences between cloud monitoring and grid monitoring, especially in termes of interfaces and service provisioning.
- Another diference is that clouds are managed by single entities [10], whereas grids may not have any central management entity.

6. RELATED WORK

6.2. Cloud Monitoring

- Reference [11] defines general requirements for cloud monitoring and proposes a cloud monitoring framework.
- PCMONS supports two approches, agents and central monitoring, and is highly adaptable, making the migration to a privite cloud straighforward.

7. KEY LESSONS LEARNED

7.1. Related to Test-Bed Preparation

- Software platforms for cloud computing, such as Eucalyptus and OpenNebula, support a number of different hypervisors, each with its own characteristics.
- An example is the KVM hypervisor: it has great performance but requires hardware virtualization that not all processors provide.

7. KEY LESSONS LEARNED

7.2. Design and Implementation

- We opted for solutions well established in the market to facilatate the use of PCMONS in the running structures with little effort and prioritized an adaptable and extensible solution.
- We planned to define some basic common metrics for private clouds, but later found that metrics are often specific to each case.

7. KEY LESSONS LEARNED

7.3. Standardization and Available Implementations

- Before choosing a specific tool for private clouds, it is important to verify to what extent cloud standards are implemented by the tool.
- Some tools, such as OpenNebula, have begun implementing standardization efforts, including the OCCI API.

8. CONCLUSION AND FUTURE WORK

- This presentation summarizes some cloud computing concepts and our personal experience with this new paradigm.
- The current portfolio of open tools lacks open source, interoperable management and monitoring tools. To address this critical gap, we designed a monitoring architecture, and validade the architecture by developing PCMONS.

8. CONCLUSION AND FUTURE WORK

- To monitor specific metrics, especially in an interface-independent manner, a set of preconfigured monitoring plug-ins must be developed.
- For future work, we intend to improve PCMONS to monitor other metrics and suport other open source tools like OpenNebula, OpenStack...

9. REFERENCES

References indicated in this presentation:

- [7] W. Chung and R. Chang, “A New Mechanism for Resource Monitoring in Grid Computing,” Future Gen. Comp. Sys. Jan. 2009.
- [8] M. Yiduo et al., “Rapid and Automated Deployment of Monitoring Services in Grid Environments,” APSCC, 2007.
- [9] L. Wang et al., “Scientific Cloud Computing: Early Definition and Experience,” IEEE Int’l. Conf. High Perf. Computing and Commun., 2008.

9. REFERENCES

References indicated in this presentation:

- [10] M. Brock and A. Goscinski, “Grids vs. Clouds,” IEEE 2010 5th Int’l. Conf. Future Info. Tech., 2010.
- [11] P. Hasselmeyer and N. d’Heureuse, “Towards Holistic Multi-Tenant Monitoring for Virtual Data Centers,” IEEE/IFIP NOMS Wksps., 2010.