Operating a Grid Site in the Cloud

C. Loomis (CNRS/LAL)

M.-E. Bégin (SixSq Sàrl), V. Floros (GRNET), I. Llorente (UCM), R. Montero (UCM)

UF4/OGF25 (Catania)

28 May 2009

Contents

- **StratusLab**
  - Motivation
  - Goals
- **Amazon Web Services**
  - Provided Services
  - “Features” of AWS
- **Hurdles**
  - Administrative
  - Middleware
- **Challenges of Dynamic Resources**
- **Conclusions**
- **Future Work**

StratusLab Motivation

- **StratusLab: *Open* collaboration between CNRS/LAL, GRNET, SixSq Sàrl, and UCM.**

- **Determine maturity of cloud resources**
  - Stability
  - Manageability
  - Functionality
- **Promise of Cloud Technologies**
  - Natural convergence between grid and virtualization.
  - Sandboxing
  - Uniformity
  - Dynamic allocation of resources
  - User-level services

Program of Work

- **Run grid site within the cloud**
  - Use standard management tools
  - Gauge maturity/stability of cloud resources
  - Extract detailed usage patterns/costs
- **Investigate cloud use strategies**
  - Bridging of resource centers
  - Resource allocation strategies

- **Use Amazon Web Services:**
  - Mature provider
  - Simple interface
  - Clear cost model
  - Provides HaaS → appropriate for running a grid site

Amazon Web Services

- **Elastic Compute Cloud (EC2)**
  - Provides easy interface for allocating and starting machines.
  - Five different machine configurations available.
  - Supports various *Linux*, OpenSolaris, Windows server.

- **Elastic Block Store (EBS)**
  - Acts as a mountable, persistent disk.
  - Can associate disk with image dynamically at startup.
  - Uses Simple Storage Service (S3) behind the scenes.

- **Elastic IP**
  - Persistent allocation of IPv4 address.
  - Can associate image instance and address dynamically.

"Features" of AWS

- **AWS uses Xen and paravirtualization rather than "hardware" virtualization. This adds some hurdles to adoption.**

- **Xen "features:**
  - Need to have OS be aware of the paravirtualization
- **Linux distributions and versions:**
  - Only endorsed, pre-compiled Amazon kernels
  - Recent OSes generally available, old ones harder to use
  - Making new OS images can be difficult
- **Network configuration challenges**

AWS vs. Virtual Machines

- **AWS uses Xen, but there are important differences between AWS and direct use of Xen.**

- **Network configuration:**
  - All machines have private and public IP addresses. User cannot predict or allocate those addresses before starting the machine.
  - Network interface uses the private address for configuration.
  - DNS contains only public address, not private one.
  - IP address can change when using Elastic IP.

- **Installation:**
  - PXE is not supported for installation of a machine.
  - Must start from existing machine image.

Administrative Hurdles

- **Grid site registration with EGEE:**
  - Procedure for being associated to a ROC worked well.
  - GOC registration afterwards also worked smoothly.

- **Paying for AWS**
  - AWS use is tied to a credit card.
  - Difficult for organization (e.g. CNRS) to pay for those services.
  - StratusLab: SixSq pays for AWS; invoices CNRS.
  - Use may require (institute or Amazon) policy changes

- **Certificates Problematic**
  - Which CA for issuing cloud host certificates?
  - CAs have different policies for obtaining host certificates

Middleware Issues

- **Resource Requirements**
  - Need clear memory/CPU/disk requirements for services
  - Tendency for developers to treat resources as “cheap”
- **Portability**
  - gLite must support recent versions of OSes
  - SL is not really a “standard” platform for many sites/disciplines
  - Treat Xen as separate, supported platform
- **Networking**
  - AWS configuration “strange”: local IP, public IP, elastic IP
  - Hostnames, etc. should be configurable!

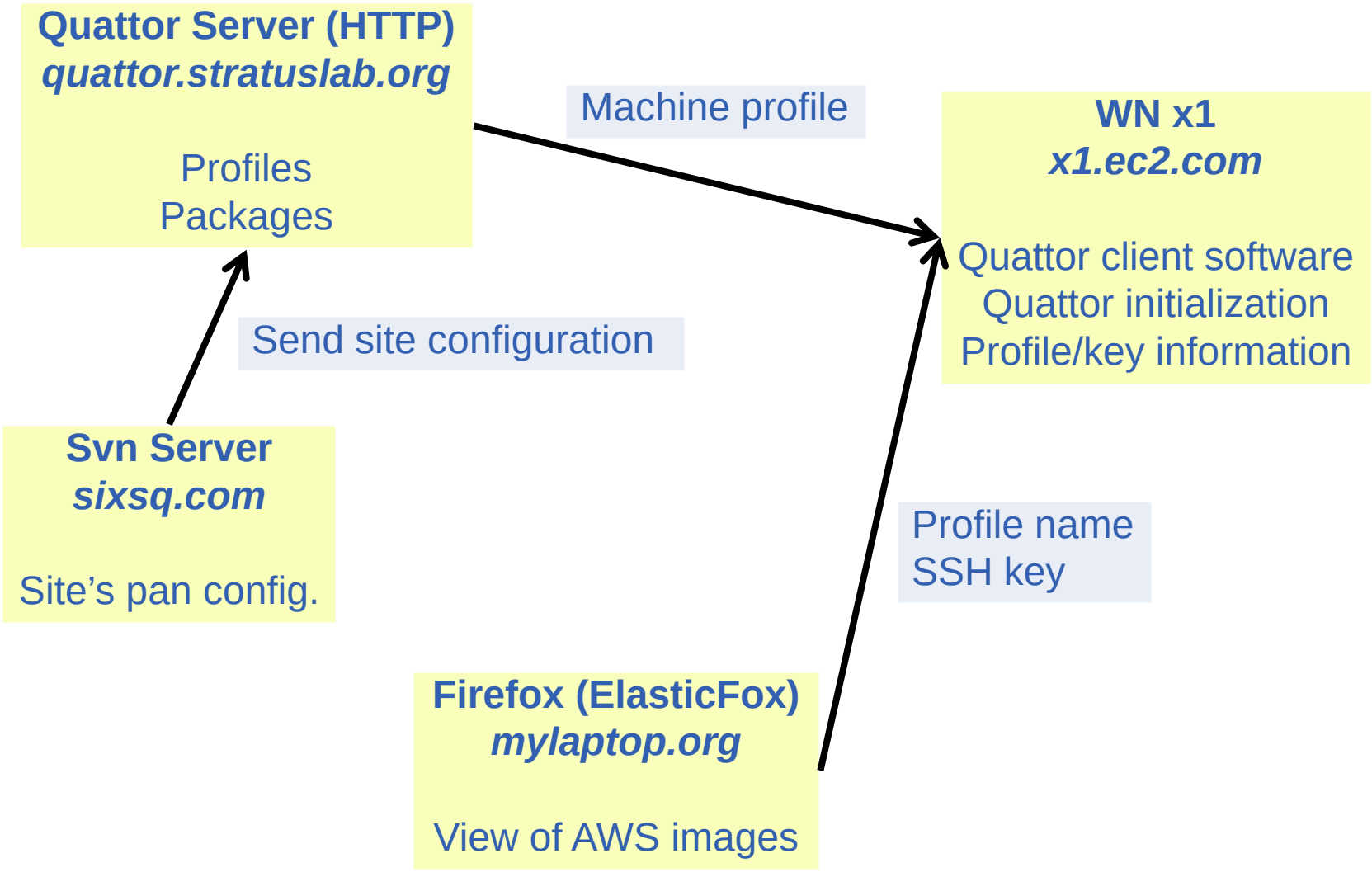
Dynamic Resource Challenges

- **One of the benefits from cloud technologies is the dynamic allocation of resources.**

- **Challenge:**
  - Late knowledge of host names/IP addresses makes configuration challenging.
  - Many server/client systems need to know hostname/IP addresses to work properly or to be properly secured

- **How to ensure that services and operational tools can take advantage of cloud's dynamic nature?**

Quattor in Cloud



Quattor Modifications

- **Standard quattor server deployed and used in AWS.**

- **Minor modifications for machine initialization:**
  - Create AWS image with basic quattor client software.
  - Add init.d script for initial quattor bootstrapping.
  - “User data” at startup selects image profile.
  - Turn off the network configuration
  - May fold changes into standard quattor distribution

Management Issues

- **Multiple machines can use the same “profile”:**
  - Easy and clean way to define only one WN profile for a site.
- **Machine names not known at compilation time:**
  - How to link batch server and clients?
  - How to link NFS servers and clients?
  - Allow late binding for some information?
- **Configuration change notifications fail**
  - No link between profile name and machine name.
  - Allow machines to register for changes?
  - Move to “chat room” messaging for changes?
- **Workflow**
  - How to manage image disks, IP addresses, etc.?
  - How to manage machine lifecycle?

Bridging Cloud & Grid Resources

- **Have already done limited tests of bridging cloud and grid resources.**

- **Setup:**
  - Use torque server from standard grid site.
  - Deploy 5 additional worker nodes in AWS-Europe.
  - Use SL4 image with grid WN software installed.
  - Open ports 22 and 15000-15004.

- **Results:**
  - No real problems!

Summary

- **StratusLab collaboration has started to investigate what is necessary to deploy a full grid site within the cloud.**
  - http://www.stratuslab.org/wiki
- **Widespread use of cloud resources in the grid:**
  - May require policy changes (e.g. policies from CAs)
  - Would be easier with support from the middleware
  - Probably easier with full virtualization (at least from customer's point-of-view)
- **Current status:**
  - Initial deployment of services inside AWS.
  - Limited tests of bridging with using WN's in the cloud.

Future Work

- Get all basic grid services running in the cloud.
- Evaluate various bridging strategies.
- Provide use information and costs to Grid Observatory.

- Work towards open source cloud distribution.