StratusLab

StratusLab: Cloud-like Resource Delivery for Production Grids

C. Loomis and M. Jouvin (CNRS/LAL)

HEPiX Fall 2010, Cornell

5 November 2010

StratusLab is co-funded by the European Community's Seventh Framework Programme (Capacities) Grant Agreement INFSO-RI-261552

CAPACITIES

Contents

Cloud Technology

StratusLab Project

- Vision and Benefits
- Early Results

Challenges

- Meeting Expectations
- Sharing
- Trust

Conclusions

“The” Cloud

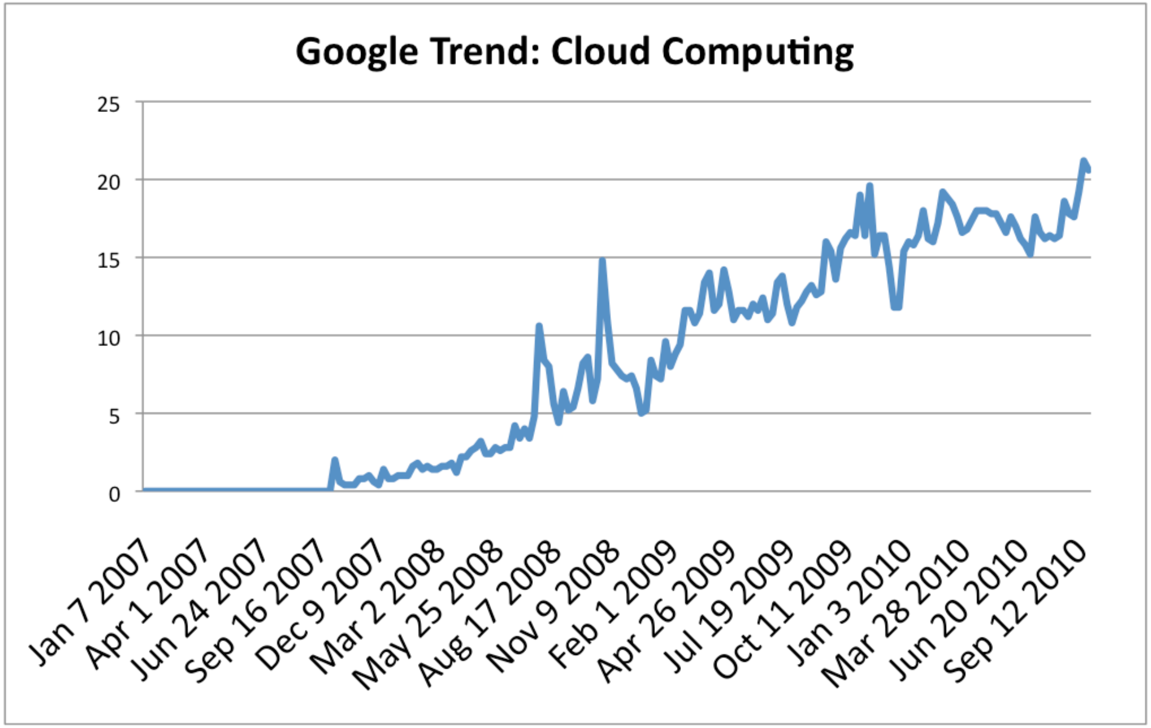
StratusLab

Cloud is the convergence of several ideas:

- Maturity of virtualization technologies
- Appearance of simplified APIs (REST, XMLRPC, …)
- Excess of commercial computing resources

Cloud marketing hype

- Many different, often incompatible definitions
- Used to sell existing software/services
- Even so, interesting, useful ideas at the core



Cloud Taxonomy and Promises

StratusLab

Vendor Lock-In

Flexibility

Software as a Service (SaaS)
Web hosting: easy to use but limited integration and questions on data ownership

Platform as a Service (PaaS)
Framework and infrastructure for web-based apps (web services)

Infrastructure as a Service (IaaS)
Remote access to virtual hardware: customized execution environment but difficulty of creating VM images and lack of standard APIs

Grid Infrastructures

Architecture

- CPU: conceived as a large, distributed batch system
- Data: core services deal with files
- Network: essentially no real management of network resources

Advantages

- Uniform security model
- Sharing of resources, algorithms, and expertise through VOs

Disadvantages

- Tendency toward complexity (APIs, services, etc. often specific to grid)
- All VOs sharing the same execution environment at one site…
- But inhomogeneous environments between sites increases job failure rates
- Porting of applications that are not batch-oriented is difficult
- Difficult to deploy VO and user-level services

Complementary Technologies

Grid: Federate distributed resources through uniform interfaces

- Uniform security model
- Sharing of resources, algorithms, expertise through VOs

Cloud: Instantaneous deployment of customized resources

- Environment is dynamic, elastic, and customized to user/VO needs
- Several levels of abstractions (IaaS, PaaS, et SaaS)
- Based strongly on virtualization technologies

Vital IaaS Features

CPU

- Virtual machines (appliances) created by users
- Repository of virtual machines to share software, expertise, …

Data Management

- The cloud must have mechanisms for efficient data management
- Minimum: file management, disk management

Network

- Dynamic control of inbound and outbound ports/connectivity
- Ability to have a public IP address associated with a machine

Existing virtual machine management tools:

- Nimbus, Eucalyptus, OpenNebula

Benefits

Ease of Deployment

- Prepared virtual machine images for grid services
- Decouple OS on physical systems from grid requirements

Increased Reliability/Robustness

- Migration for load balancing or for avoid hardware failures
- Better isolate user jobs/machines from others

Customized Environments

- Responsibility of application execution environments rests with VOs
- Flexible environment appeals to more diverse scientific community

StratusLab Project

Facts

- 1 June 2010—31 May 2012 (2 yrs.)
- Partners: 6 from 5 countries
  - – Project director: C. Loomis
- Budget: 3.3 M€ (2.3 M€ EC)

Goal

- Provide coherent, open-source private cloud distribution
- Design and implement an open-source API to cloud resources

Contacts

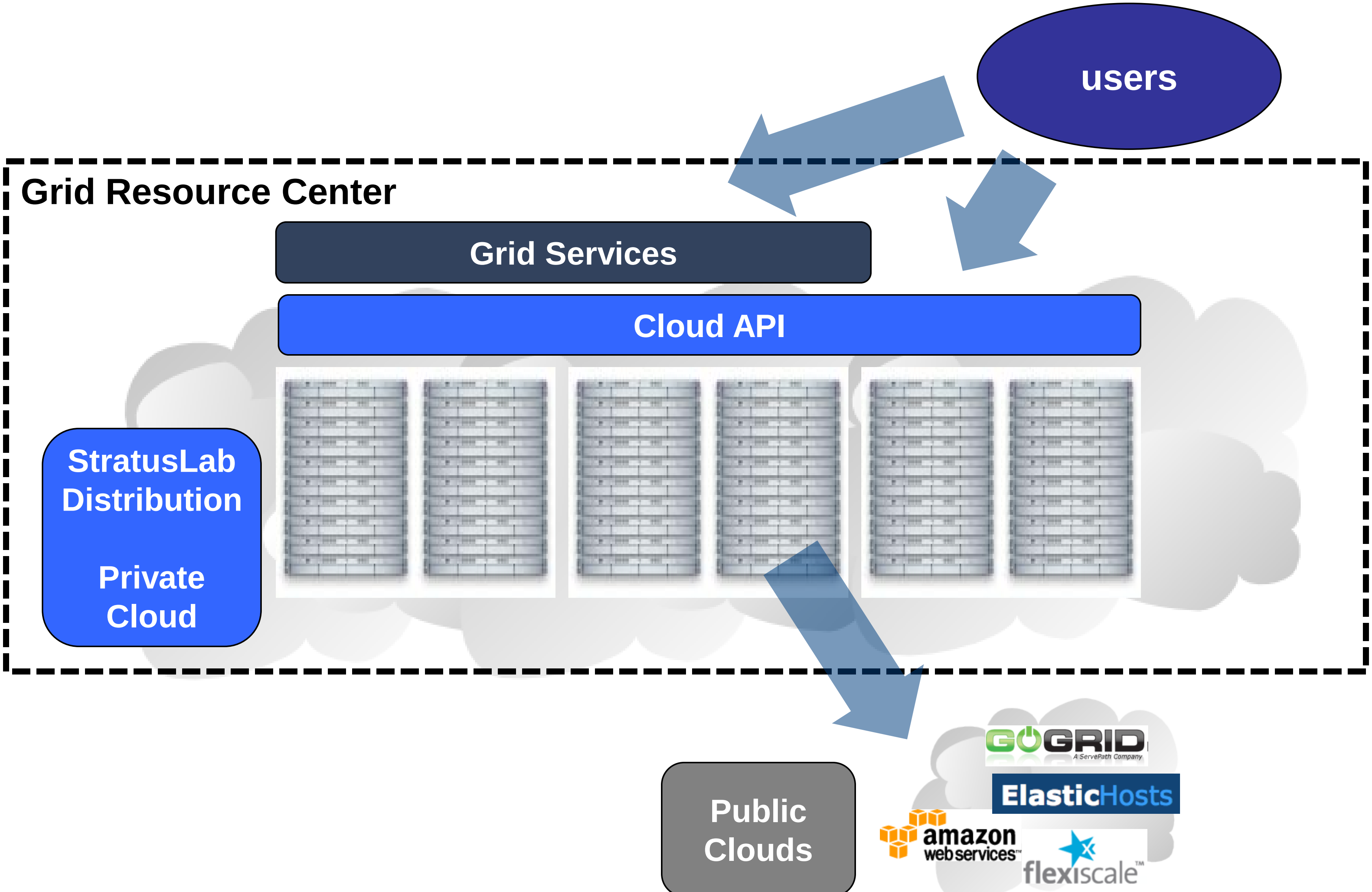
- Website: http://stratuslab.eu/
- Twitter: @StratusLab

CNRS (FR) *UCM (ES)*

GRNET (GR) *SIXSQ (CH)*

TID (ES) *TCD (IE)*

Grid Services Over Cloud Resources



Integration Strategy

Agile Software Processes

- Short project, need to evolve quickly
- Focus on user and administrator requirements
- Planning meeting, 3 week sprint, Demo meeting, Mgt. meeting

Integration Over Development

- Small project with limited effort (15—20 people)
- Many existing services that can be integrated
- Only develop to fill gaps or to expand functionality

Open to Multiple Implementations

- Identify clean interfaces between components
- Standardize those interfaces where possible

User and Administrator Surveys

Surveys for Users and Administrators

- Collect use cases and requirements from target communities
- 248 and 608 unique views, 22 and 56 completed
- Analysis: D2.1 (http://stratuslab.eu/doku.php?id=deliverables)

User Feedback

- 33% of users are regular cloud users
- Most respondents from CS/eng., bioinformatics, and physics
- Most planned for “real” use within 2 years

Administrator Feedback

- 68% of administrators will deploy virtualization/cloud technologies
- Half have already deployed; remainder within 1 year
- Reluctance for administrators to trust user-generated VMs

Early Results

StratusLab

Appliance Repository at TCD

- Based on standard Apache web server
- Contains stock images for supported operating systems
- Will contain grid service images (“appliances”)

Complete Grid Sites at GRNET

- Two StratusLab clouds deployed: Ubuntu 10.04 and CentOS 5.5
- Functional (pre-production) grid site on each cloud

Worker Node Tests at LAL

- Worker nodes deployed via Quattor in StratusLab cloud
- Run in production for since last August without problems

Initial Public Release

Availability

- 9 November 2010

Scope

- Basic cloud distribution
- Generate feedback from users and administrators

Warning

- All APIs, commands, etc. are subject to change!

Documentation

- User tutorial
- Description of reference deployment
- Installation guide

Services

- Appliance repository
- User and administrator support

Software

- Repackaged/patched OpenNebula
- Command line tools (user and administrator)
- Standard images (ttylinux 9.4, Ubuntu 10.04, CentOS 5.5)
- Web monitor
- Manual and Quattor-based installation

Appliance Repository

StratusLab

Appliance

- Virtual machine created to provide a single service or a few tightly coupled services
- Appliances can be used to exchange software and knowledge

Appliance repository

- Repository allows virtual machines to be found and shared
- Current implementation just a simple web server
- Contains base images, will have grid images as well

Future directions

- Worthwhile to use metadata schema like Dublin Core?
- Publish metadata via RDF for others to use?
- Federation of separate appliance repositories?
- Mechanisms/services for distribution of trusted images (HEPiX)

Site Management Tools

Quattor

- Use to demonstrate integration with site management tools
  - Others may come later
- Chosen because of partner experience and use within EGI
- Nearly complete automated installation with Quattor available
  - Some fixing in OpenNebula required to entirely automatic installation
  - Quattor used to provide feedback to ONE developers on configuration process

Virtual machine metadata

- Information about virtual machine contents critical (esp. when shared)
- Tools like Quattor can be used to document those virtual machines
  - Same idea as LBNL *recipes*
- Can be used as mechanism for generating images as well

Meeting Expectations

StratusLab

Bare Metal Performance

- Virtualization tax near zero from technological improvements
- IO still an issue but improvements expected with next CPU generations

Instantaneous (~10s) Availability of Machines

- Fast copies, caching of machine images
- Client site preparation of contextualization image, …
- Optimization of OS boot sequence

Fully Elastic Clouds

- Users expect infinite resources, machine room reality is different!
- How to prevent users from seeing a queue?
- Allow resource searching/matchmaking like the grid?
- Site-level, automated bartering through hybrid clouds?

Sharing Resources

Possible to bring grid's collaborative spirit into the fundamentally "selfish" cloud paradigm?

Authentication/Authorization Mechanisms

- Reuse worldwide trust network and its technologies

Machine Images and Appliances

- Great way to share algorithms, software, packaged data, …
- Issues of transport, caching, access control and trust

Data Management

- Like grid, issues of access, persistency, versioning, …
- Additional issue of service state, logging, accounting, …

Building (More) Trust

Reluctance to Run User-Defined Images

- Realizing full potential of cloud infrastructures will require building more trust between users, VOs, and system administrators
- Trust will vary with actors involved → sliding scale that balances level of trust with allowed capabilities

Requirements

- Define security requirements and best practices for images (HEPiX)
- Trusted images: mechanism to enforce/check/**revoke** endorsement of images (HEPiX)
  - Discussions planned with HEPiX WG about VMIC idea
- Matchmaking: appliance metadata vs. site requirements
- Enhanced monitoring (e.g. ports and connections)
- Additional trust between firewall “controllers” and site administrators

Conclusions

Grid and cloud technologies are complementary

- Easier deployment, maintenance of sites for administrators
- More flexible, capable, and dynamic infrastructure for users
- Cloud makes process more agile, appealing to new scientific communities

StratusLab

- Early results: surveys, grid services over cloud, appliance repository
- Release: expected very soon, accessible cloud infrastructures for tests

Challenges for cloud community

- Meeting expectations of users and administrators
- Sharing easily the diversity of resources in communities
- Building more trust between actors