2nd Workshop on Software Services: Cloud Computing and Applications based on Software Services

Timisoara, June 6-9, 2011

Auto-administration of gLite-based grid sites

Alexandru Stanciu, Bogdan Enciu, Gabriel Neagu

National Institute fot R&D in Informatics Bucharest

Outline

- Motivation
- Overview of system management issues
- Specific case of gLite grid based infrastructure
- IGOS project
  - Infrastructure operation support system
  - Use a configuration management system (Puppet)
- Conclusions and future work

Motivation

- A general problem of large distributed ICT infrastructures is their extremely high complexity which makes their management especially difficult and error prone.
  - Automatically detect and fix errors occurring in operation of gLite-based Grid infrastructure.
- There is a growing need to replace manual operations in the field of systems administration and the ICT infrastructure operation.
  - Manual control is time-consuming, expensive, and error-prone.
  - It is not scalable

General system management requirements

- Need a standard way for deploying, configuration and management of systems
  - Either on physical machines or virtual machines in a cloud
  - Control the infrastructure in a consistent way
- Need a flexible and scalable way to manage the infrastructure
  - Now the cloud has increased the systems density dramatically
  - Virtualized environments double at least the number of machines that need to be managed
  - Add to the mix the integration of public and/or private clouds

General system management requirements (cont.)

- Automate admin task like OS and applications installation, configuration and patching
  - Keep the systems running up-to-date
  - Make sure there is no deviation from the initial system state
- Need fully integration of operational tools
  - Deployment, configuration management, monitoring, trouble ticketing, audit, etc.
  - Try to get as simple as possible

General system management requirements (cont.)

- Portable configuration
  - Generic description of the system configuration
  - Use resource abstractions
  - Manage heterogeneous systems: Windows, Linux, OSX, Solaris
- Use industry standards for security

Wrap-up and one step further: Self-Management

- Self-managing distributed computing resources transparently to operators and users.
- We need computer systems to manage their own operation without human intervention.
- We need a system to make decisions on its own, using high-level policies
- Human operator should not control the system directly. Instead, she must define general policies and rules that serve as an input for the self-management processes.

More context: gLite-based grid infrastructure

- RoGrid-NGI National Grid Infrastructure
- Includes national research Institutes and Universities with consolidated experience in operating grid clusters working in production regime
  - Incubated in the European Grid Infrastructures EGEE and SEE-GRID
  - Operational autonomy in the framework of the European Grid Infrastructure (EGI)
  - National multidisciplinary VO Gridmosi

RoGrid-NGI National Grid Infrastructure

- Currently there are registered 18 sites in EGI-inSPIRE infrastructure
- Certified for production: 11 sites, 3000 CPUs, 1.6 PB storage
- Main user comunity is HEP
- Other VOs/communities: Environmental, Earth Science, Biomed

RO-01-ICI grid site

- Working as production site in the EGEE and SEE-GRID infrastructures since 2004
- Significant up-grade of the RO-01-ICI site in 2011, based on the financial support provided by the Sectorial Operational Program “Increase of Economic Competitiveness”, Operation 2.2.3: “Development of networks of R&D centres, coordinated at the national level and connected to European and international networks (GRID, GEANT)”

RO-01-ICI grid site (cont.)

- Computing platform
  - Computing capacity: 264 cores (22 servers IBM BladeCenter HS22, 2 x Intel Xeon X5680 CPUs)
  - Interconnect: QDR 40Gb/s InfiniBand
  - Storage capacity: 36 TB (IBM System Storage DS3512 Express)
  - Cluster management (4 IBM System x3650 M3 servers)
- Software components
  - Software platform for modelling-simulation-optimization based on MATLAB and Simulink product families
  - IGOS project - Prototype for monitoring and support of grid resource operation

IGOS in a nutshell

- Prototype for an intelligent Grid-based infrastructure operation support system
- Automatic control of a gLite based Grid infrastructure
- Automatically detect and fix errors occurring in operation the Grid infrastructure
- More details about the project (Romanian language): http://ro01ici.grid.ici.ro/

YAIM

- YAIM is the default tool to configure the gLite middleware
- A collection of scripts which configure all the components of the gLite services
- PROs
  - Very easy to setup and use
- CONs
  - Need to find a way to automate its usage as it is not scalable
  - Has a its own specific configuration format and it is not easy to extend it to configure other applications
  - Re-configuration of a service is not always a solution

Solution: Configuration management system

- Configure the systems by defining their desired state
- Performed automatically and transparently, no manual interventions
- Try to minimize the need of custom in-house scripts as much as possible
  - Over time they tend to become unmaintainable
  - No need to replicate work

Configuration management system - Puppet

- Using a declarative language to describe the systems state
  - Define **the infrastructure as code** - create a model of the infrastructure
  - Describe de systems configurations, relationships between them
  - Then quickly and easily make changes to the infrastructure by changing the code in one single place, one can change the systems configurations across datacenters

Configuration management system – Puppet (cont.)

- Compare current system state with the desired system state
- Iterative change management process
- Convergent
- Policy-enforced consistency
- Auditing

Why Puppet?

- There are other similar applications like Cfengine, Chef, Quattor which can provide the same functionality as Puppet
- But Puppet is more easily to integrate with other operational tools

Employing Puppet for IGOS

- We want to specify the "policy" that must be maintained by each system which is part of a Grid infrastructure.
- The "policy" means the system state, namely how it is configured
- When an error occurs, it can be detected and resolved automatically by the configuration management system because the system state becomes inconsistent with the "policy" defined for it, and so must be corrected.

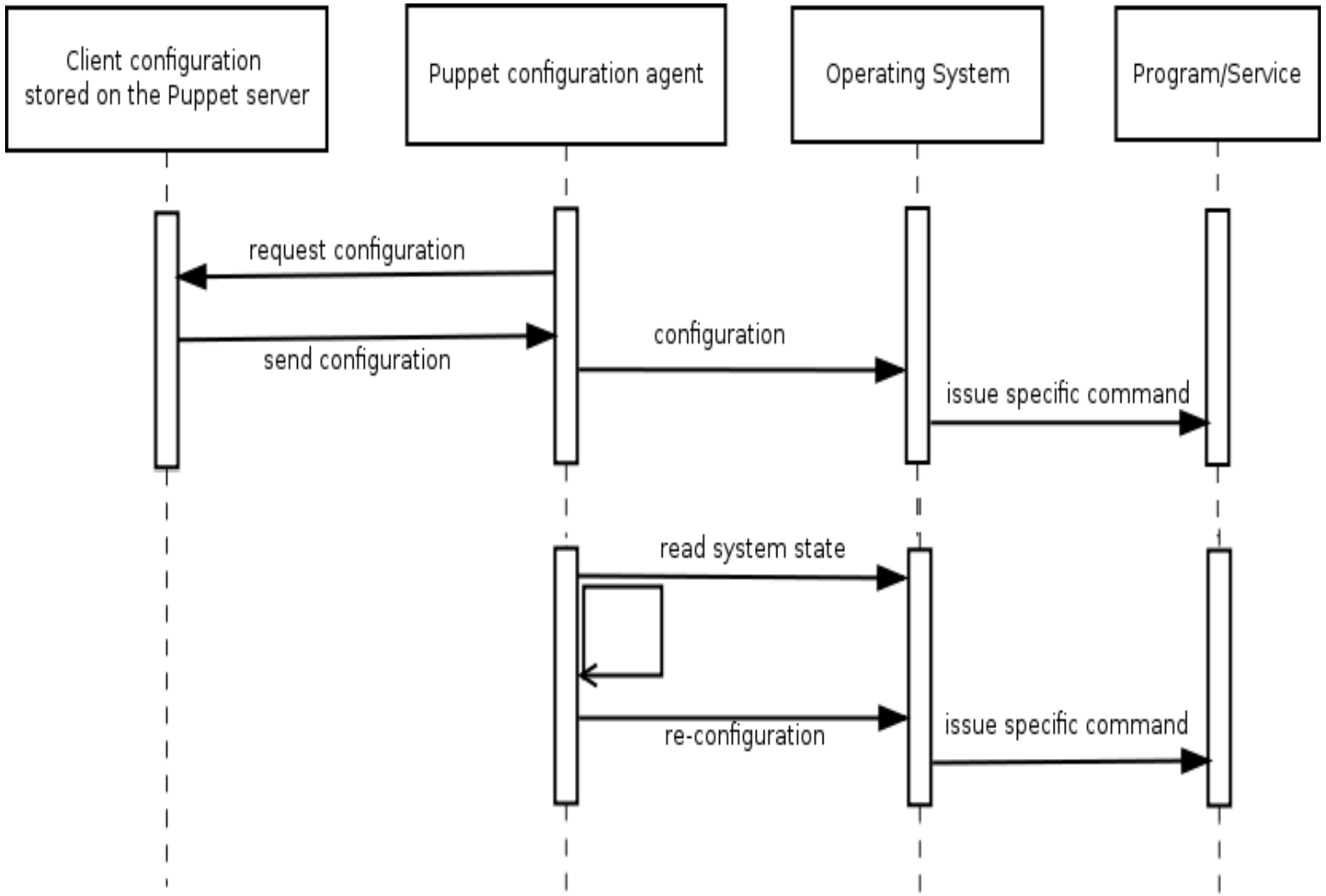
Employing Puppet for IGOS (2)

- Thus the anomaly detected is reverted to a correct state, implicitly solving the error which occurred in the system operation.
- Still, one aspect to be considered very carefully is that the configuration policy is enforced but is not guaranteed because in practice one can never guarantee that a system is exactly in an ideal state or a desired state, but rather that it is converging to the desired state

Employing Puppet for IGOS (3)

- When Puppet notices that the current state is different than the specified state, it will automatically re-configure the system to the desired state, which means that the system should get back to normal operation.
- Puppet thus must act to restore the system to the properly configured state, and must meet the "policy" that has been defined for the node. This action implicitly corrected the error detected.

Employing Puppet for IGOS (4)



Employing Puppet for IGOS (5)

- We need to create the Puppet's manifest file containing the definition of the desired configuration of the system.
- We started by using Puppet to deploy YAIM and other applications' configuration files from a central repository
  - Define a class for yaim conf.
  - A class for the distributed file system - glusterfs conf.
  - A class for LRMS – torque,
  - Etc...
  - Define the nodes which use these classes
- Then we can use YAIM tool to configure the gLite Grid middleware

Using troubleshooting guides for defining the configuration

- Created and maintained by experts
- Collection of errors, diagnosis and solutions for issues encountered with gLite middleware
- A very common problem is when the Workload Management Service (WMS) fails to receive the final status of the job execution
  - Usually a problem with Local Resource Management System
  - A possible case an improper configuration for the "/home" directory which we may solve easily with a definition in Puppet's manifest file

Conclusions and Further work

- Using a configuration management system such as Puppet we can create an autonomous system which can **auto-administrate** itself and perform self-repair actions:
  - if there are defined **exhaustive rules** to achieve a desired system state which can be defined as a “**configuration policy**"
  - this policy always has to be enforced whenever there is a deviation from it.
- As further work we plan to
  - completely replace the YAIM tool which was used for configuration of the gLite middleware, and define the “policy” for gLite Grid services in the Puppet's manifest especially taking into account the specific information in gLite troubleshooting guides.
  - develop a Puppet module to configure gLite services

Thank you for your attention!

Q/A