Seminario di inizio terzo anno

Dicembre 7, 2016

Jacopo Pellegrino
- What “cloud” is
- My work on cloud computing

Cloud

- What “ABM & MAS” are
- The idea CIs as agents

ABM & MAS

- Model design
- Tools for implementation:
  - GAMA
  - JADE
  - GitLab
  - PostgreSQL

ABCM

- Current status
- Plans for the last year

Summary
Cloud Computing

- A brief introduction
- Cloud for HEP
- Cloud Toy
- Cloud Bursting
Cloud Computing

The cloud computing a widely adopted paradigm in the context of HEP.

CIs provide access to a shared pool of computational, network and storage resources.

Resources are easily available to users.

Setting up and maintaining a cloud infrastructure may not be trivial by the providers’ point of view.
Cloud Computing

The main idea behind this work is to address two common issues of cloud infrastructures:

- **Usability**: simplifying the setup and installation process.
- **Efficiency**: making the usage of resources more dynamic, flexible and efficient.

The aim is to improve existing cloud infrastructures and give sites with limited manpower/knowledge easier access to cloud technologies.
Cloud Computing

Automatization of Installation process:

- Creation of kickstart file
- Preparation of customized ISO
- Preparation of bootable usb drive
- Installation on server
- Test of the installation
Cloud Computing

● **Creation of kickstart file**: configuration parameters, software packages, disks partitioning, and the like...

● **Preparation of customized ISO**: start from a standard netinstall iso file, make it look for the kickstart at boot time

● **Preparation of bootable usb drive**: burn the iso into a usb drive so that it is possible to boot from it

● **Installation on server**: plug the usb drive and reboot

● **Test of the installation**: the server is ready in about 1 hour
Cloud Computing

Efficiency of the infrastructure:

A CI is used by different stakeholders.

The aim is to dynamically change the quota of resources.

A “private” CI is introduced.

T2 has 100 cores
T4 has 200 cores
Together they have 300

The aim is to manage these 300 cores dynamically for T2 and T4
Cloud Computing

The “public” OpenNebula only knows that T2 and T4 have 300 cores (user-level control).

The “private” OpenNebula can dynamically change quotas between T2 and T4 according to actual needs (full sys-man control).

Variations of quotas can be performed via oneadmin user.
Cloud Computing

The **ONE-cloudbursting-driver** is currently under test.

It enables OpenNebula-based cloud to "burst" Virtual Machines (VM) to external OpenNebula clouds using built-in OpenNebula XML-RPC and OCCI interfaces.

**Aim:** start VMs on Public OpenNebula from the Private OpenNebula.
Cloud Computing

PRIVATE ON

remote VMDIRAC

change quotas

PUBLIC ON

rOCCI server

Instantiate new VM

OCCI client

ONE cloudbursting driver

rOCCI server

Instantiate new VM

rOCCI server

Client

server
Cloud Computing

This work has been presented at:


- **CHEP2016**: A modular (almost) automatic set-up for elastic multi-tenants cloud (micro)infrastructures; F. Astorino, A. Amoroso, S. Bagnasco, N. A. Balashov, F. Bianchi, M. Destefanis, M. Maggiora, J. Pellegrino, L. Yan, T. Yan, X. Zhang, X. Zhao; CHEP2016, 10-14/10/2016; San Francisco, USA
ABM & MAS

- A brief introduction
- An Agent Based Cloud Model
Intelligent Agent: a computer system capable of interaction with the environment and autonomous actions to reach design goals.

Agent-Based Models and Multi-Agent Systems:
- systems made up of agents that interact with each other and the environment they live within to reach personal or global goals
- study the evolution of distributed and dynamic complex systems in many fields
- approach taken from nature (hive, ants...)
ABM & MAS

In *ABM* the focus is on the internal architecture of the agent. In *MAS* the focus is on the interaction between simple agents.

The main idea of these approaches is that agents have simple rules and limited knowledge but a global intelligence **emerges** from the interaction.

*example: segregation model*
ABM & MAS
ABM & MAS
Cloud computing is widely adopted in the context of HEP.

- many sites (globally distributed) have a C.I.
- sites host several experiments
- groups from different sites work on the same experiment
- there may be federations of C.I.

Such a scenario can be viewed as a dynamic and distributed complex system.
ABM & MAS

- Design and implement a **MAS to model a community of cloud infrastructures** devoted to HEP computing.
- Agents are related by means of a **trust matrix** which is updated after every interaction.
- The trust between agents regulates:
  - the propagation of information among the network
  - the possibility to share resources
Agent-Based Cloud Model

- Model Design
- Tools adopted
  - GAMA
  - JADE
  - GitLab
  - PostgreSQL
In the current design of the model agents have “scheduled actions” to perform (sending a job request).

These actions trigger the interaction with other agents (creating a job, sending a cfp, ...).

The outcome of these actions causes variations of the trust matrix.

The interaction triggered by a job request is summarized by the following diagram
ABCM

Create Job Object

Experiment Agent

Done/Failed

Job Request

Propose/Refuse

Accept/Reject

Cloud Agent

CFP

Cloud Agent

Done/Failed

VM Object

Inform CFP

Instantiate VM

Reply CFP

Execute Job

Site Agent

Monitor Agent

Check Template

Group Agent

Experiment Agent
ABCM

**GAMA** 1.6 has been the first modeling tool adopted.

“GAMA is a modeling and simulation development environment for building spatially explicit agent-based simulations.” *

GAML language very close to JAVA with some limits. The platform presented bugs during the simulations. These bugs may have been fixed in version 1.7.

*http://gama-platform.org
JADE 4.4 has been adopted is place of GAMA.

“...is a software Framework fully implemented in the Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications...” *

● PROs:
  ○ Java language, much more freedom
  ○ Industrial Tool
  ○ Agents in different machines

● CONs:
  ○ Lack of GUI

*http://jade.tilab.com
Due to the complexity of the model, a versioning framework was needed to keep track of changes.

For this task GitLab has been adopted.
In order to solve the problem of lacking GUI, the data produced by agents have been collected by a global agent and written into a PostgreSQL database.

Initially an HTML5 GUI has been made which read from the database by means of a jsp page.
The HTML5 GUI required many steps to run and was not easy to set up. GAMA can interact with databases and provided a very easy-to-use GUI.

From these considerations the idea to take advantage of the built-in GUI of GAMA 1.7:
ABCM
Summary

- Current status
- Plans for the last year
- Talks and publications
Current status

- **Cloud**
  - the tool for automatic installation is ready
  - the cloud bursting tool is currently under test

- **ABCM**
  - the model is partially completed
  - documentation is up to date

- **Thesis**
  - first chapters written
Plans for the last year

- **Cloud**
  - set up the cloud bursting tool
  - test the infrastructure

- **ABCM**
  - proceed with the model implementation
  - simulate complex distributed computing real cases (HEP experiments, e.g. BESIII and BELLEII) with nation-wide stakeholders (e.g. INFN)
  - perform analysis of the results

- **Thesis**
  - add and complete the remaining parts
Thank you for the attention
Talks and publications

- **ICME2015**: A Multi Agent Approach for Autonomous Digital Preservation; Pellegrino J., Allasia W., Maggiora M.; ICME2015 conference proceedings, workshop HMMP, 03/07/2015; Torino, Italia
  

Question Time