

# Yearly Team Meeting

Mateusz Guzek

Supervisors: Pascal Bouvry, Johnatan Pecero, Grégoire Danoy

November 23, 2012

- 1 Intro
- 2 Holistic Model
- 3 Multi-Agent Systems

1 Intro

2 Holistic Model

3 Multi-Agent Systems

# Background

- Master

- ① Wrocław Uni. of Tech. 2006-2008
- ② Uni. of Southern Denmark 2008-2009
- ③ Uni.lu 2009-2011
- ④ Thesis title: Performance and Energy optimization by a Multi-Objective Evolutionary Algorithm in Large-Scale Distributed Systems (→ DAG scheduling with DVFS)  
Advisors: Pascal Bouvry, Johnatan E. Pecero, Bernabe Dorronsoro

- PhD

- ① Uni.lu 2011-2014
- ② Thesis title: Holistic autonomic Energy and thermal aware Resource Allocation in cloud computing.

- 1 Intro
- 2 Holistic Model**
- 3 Multi-Agent Systems

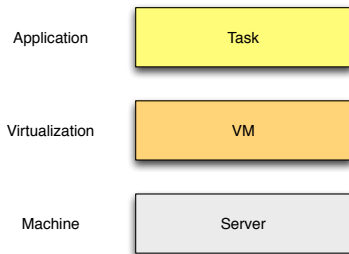
# Holistic model overview

## Model dimensions

- 1 Software: Task, Virtual Machine, Physical Machine
- 2 Hardware: Servers, Switches, Routers
- 3 Power Consumption and Thermal Phenomena
- 4 Storage: Data location
- 5 Networking: Data transfer

# Software

Extending scheduling (queueing) models by virtualization layer.



Motivation:

- Configuration overhead inclusion (time and resources for deployment and running)
- VM Migrations

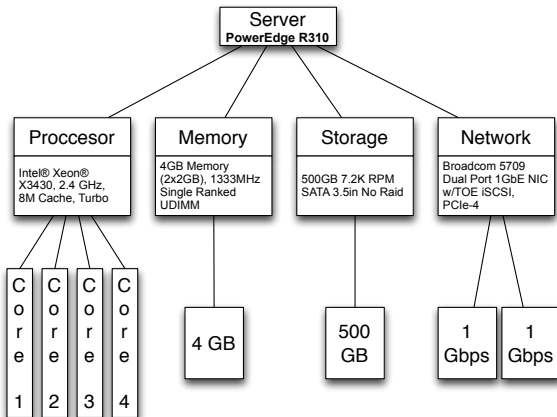
# Services

- Each task belongs to a set of tasks called service and have common Service Level Agreement function.
- Task can be known a priori (batch) or they can come from a queues with arbitrary (possibly known) probability.



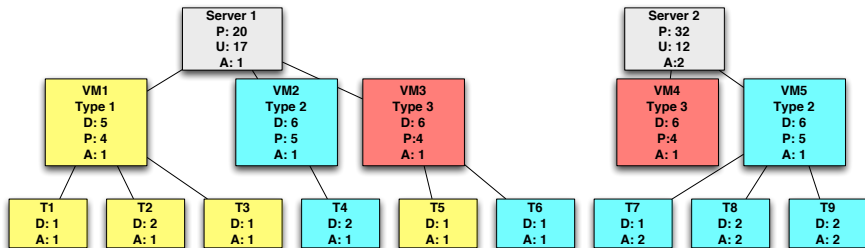
# Hardware

Resource vector  $res$  express resources provision and demand.



# Virtualization

VM demand resources from servers and provide them to tasks.  
Each task has defined a set of VM which can process it.



# Network - intrasite

- Assumption of non-blocking core. Data transfer time is estimated using the sum of latencies between source and destination and the available Tx/Rx at source/destination.

# Network - intrasite

- Assumption of non-blocking core. Data transfer time is estimated using the sum of latencies between source and destination and the available Tx/Rx at source/destination.



$$T_{i,j} = \sum_{x_k \in path_{i,j}} lat_k + \frac{data}{\min(cap_{net,l}^i, cap_{net,m}^j)}, \quad (1)$$

where  $path_{i,j}$  is a communication path between  $i$  and  $j$ ,  $cap_{net,l}^i$  and  $cap_{net,m}^j$  are capacities of network interfaces  $l$  and  $m$  of resources  $i$  and  $j$  respectively.

# Storage

Composed of two entities:

- ① Data object – files, data in memory
- ② Data storage – file systems, memory

A task has data component: receive input, process it, send output.

# Objectives

- 1 Energy consumption of:
  - Servers
  - Network
  - Cooling
- 2 Quality of Service, based on SLA functions.

# The CC test bed

## Grid5000:

- Monitoring
  - Energy: Wattmeters in Lyon, Grenoble, Toulouse; IMPI for Grenoble
  - Thermal: IPMI
  - Resources utilization: VM Hypervisors
  - QoS: System probing and logs
- Control
  - Possibility of deploying OS image = full control
  - Reservation of a set of nodes and deploying one of CC infrastructures (e.g. OpenNebula, OpenStack)
- Size – depends on the grid utilization, policies etc.

- 1 Intro
- 2 Holistic Model
- 3 Multi-Agent Systems



# MAS and organizations

- 1 The aim of an organization is to facilitate reaching the desired states by the system.
- 2 Exist in any MAS (e.g. pattern of communications, authority, responsibilities ...).
- 3 We adopted organizational model ( $\mathcal{M}oise^+$ ), that defines organization in 3 dimensions:
  - Structural: roles and groups
  - Functional: schemas, goals, missions
  - Deontic: obligations, permissions

# SotA

Existing reorganization approaches:

- Limited capabilities or
- Sequential and centralized process that requires halting entire MAS.

But if the system is dynamic, organization shall be changed.

# Proposed solution

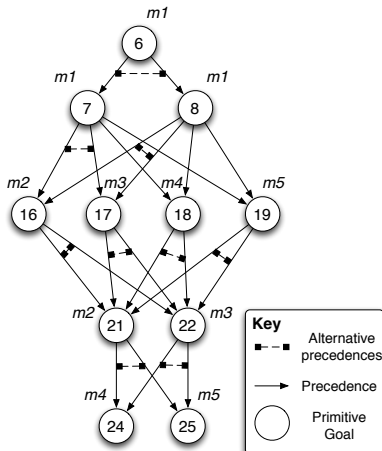
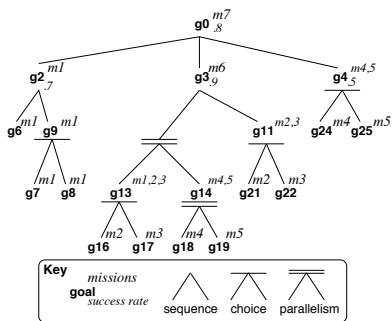
New organizational model ParaMoise:

- Allows to change each organization element
- Reorganization is parallel, online process

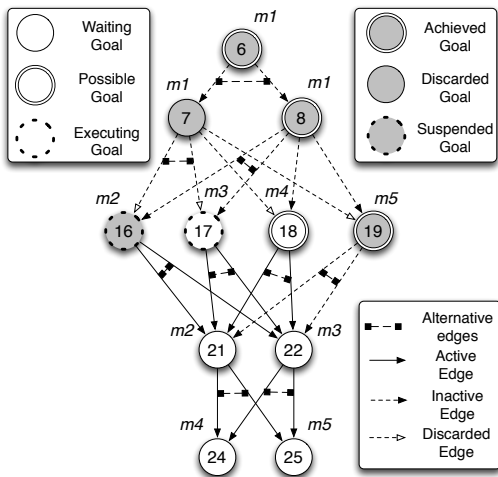
It is based on:

- Workflow specification
- Transactions mechanism
- Locks

# Workflow specification



# Workflow execution



# Locks & Transactions

Locks define parts that must remain unchanged:

- 1 Read
- 2 Write

Transactions defines what to do in case of goal failure.

**Thank you for your attention!**

**Any questions?**