Yearly Team Meeting

18.10.2013

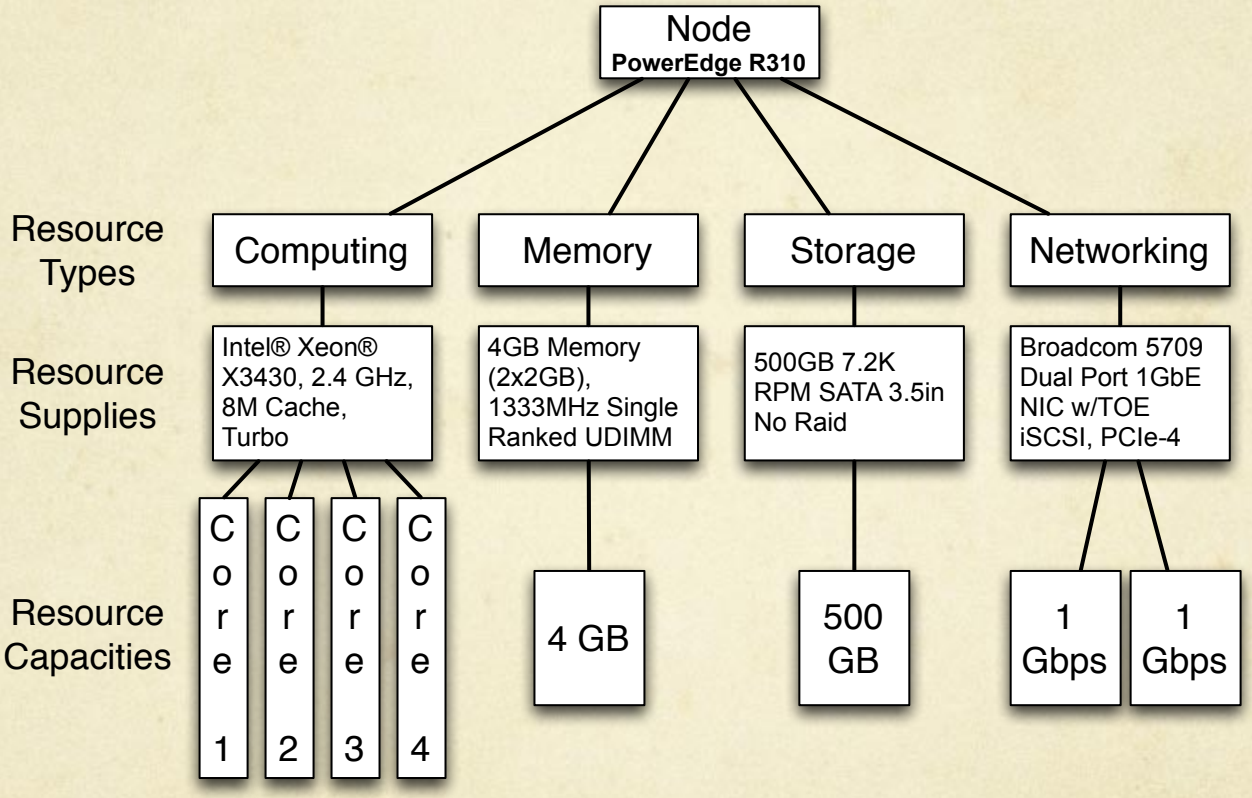
Mateusz Guzek

Presentation Overview

1. Modeling
   - Establishment of power modeling techniques
     and tools for cloud systems
2. Simulation
   - GreenCloud simulator developments
3. Multi-agent systems
   - ParaMoise framework.

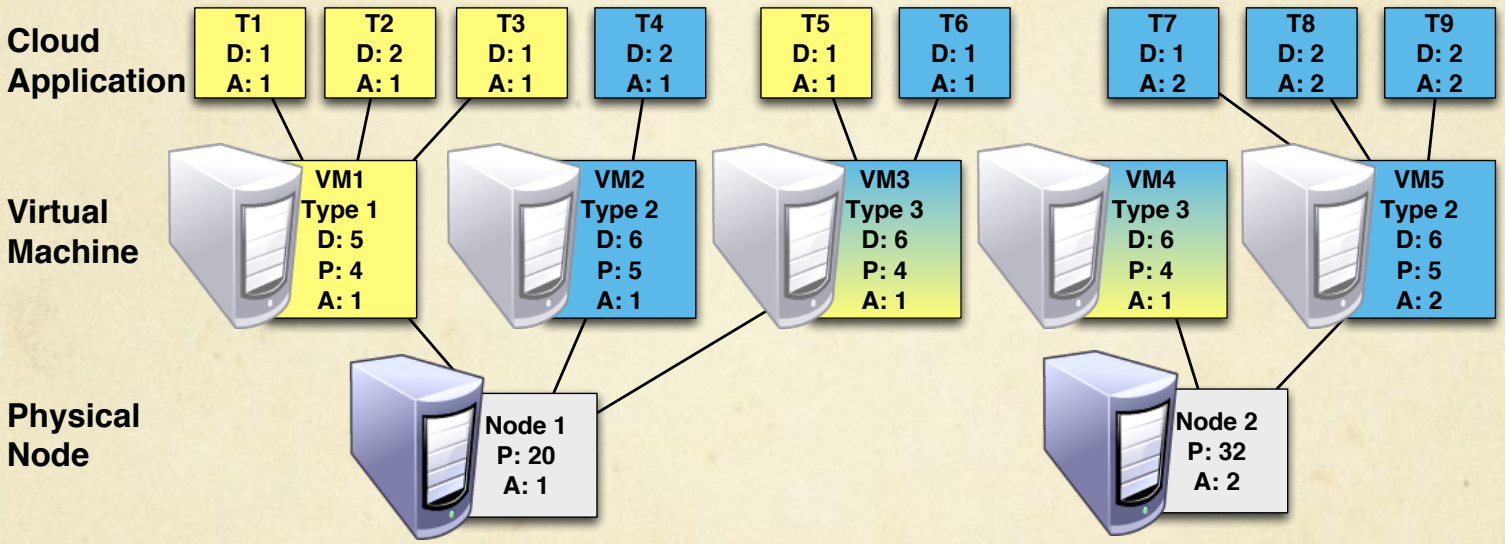
MODELING

Holistic Model overview: resources



Hardware resources are modeled as **Resource Supplies**, grouped by **Resource Type**. Each **Resource Supply** has defined **Capacities** vector and **Architecture**.

Holistic Model overview: allocation



Tasks, Virtual Machines and Nodes are **explicitly** modeled.

Power Modeling – data

- The modeling uses 3 information sources:
  1. Utilization (CPU, RAM, IO, Network)
  2. Thermal onboard sensors
  3. Wattmeters
- The modeling aim is to predict:
  1. Power
  2. Temperature

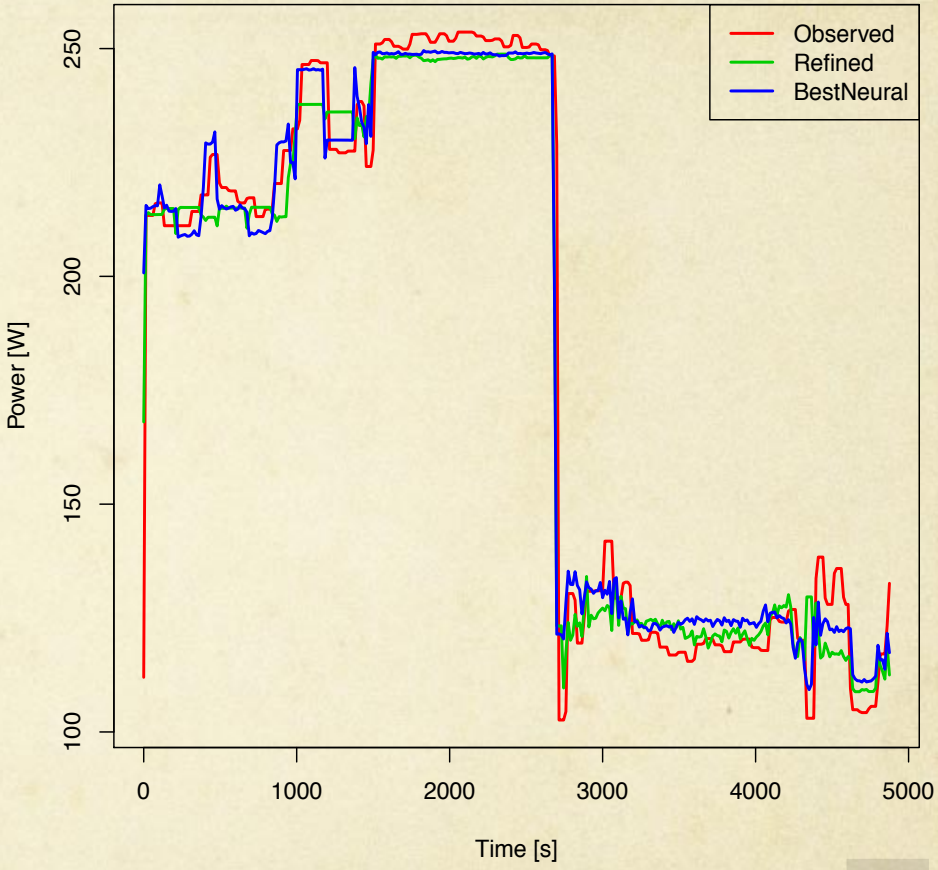
Power Modeling

Achieved:

1. Servers hosting one vm each, running a set of HPC benchmarks on Xen, KVM and ESXi (VMware)
2. Modeling tools: Linear Regression, Neural Networks
3. Deployment and monitoring of mult-tenant virtualization and Iaas Cloud: OpenStack (KVM, Xen)

Next steps:

1. OpenNebula
2. Modeling of multi-tenant cloud running diversified workloads.



[1] **A Holistic Model of the Performance and the Energy-Efficiency of Hypervisors in an HPC Environment** Guzek, Mateusz; Varrette, Sébastien; Plugaru, Valentin; Pecero, Johnatan; Bouvry, Pascal in *Energy Efficiency in Large Scale Distributed Systems (2013), LNCS vol. 8046*

Power Modeling – conclusions

- Linear regressions can be successfully used (0.8-2.6% accumulated error),
- Neural networks training is:
  - longer by orders of magnitude than regressions,
  - fails to correctly converge in ~10% of cases
  - provide worse results (>3% error)
- Using discrete estimators (hypervisor, node id, running application) typically increases accuracy of estimation
- Linear models are applicable for optimization algorithms and simulators

SIMULATION

GreenCloud2

My next goal is to extend GreenCloud simulator by implementing concepts of holistic model:

1. All **provisions** and **demands** of resources are represented as resource vectors
2. Heterogeneous servers are composed of **multiple** hardware components
3. Availability of the resources is used during **resource allocation** process on the **data center** and **server** levels
4. **VMs** are explicitly modeled as an **intermediary** layer between hosts and tasks
5. **Multi-core** processors are supported, including **time-sharing**
6. **Load** of all resources is tracked for **energy prediction** and **result analysis**
7. **Resource specifications** and **power models** are defined in **tcl scripts**

GreenCloud2

Additionally:

1. **Overload** is **detected** on both data center and host levels
2. New **visualizations** are added to graphical output of simulation

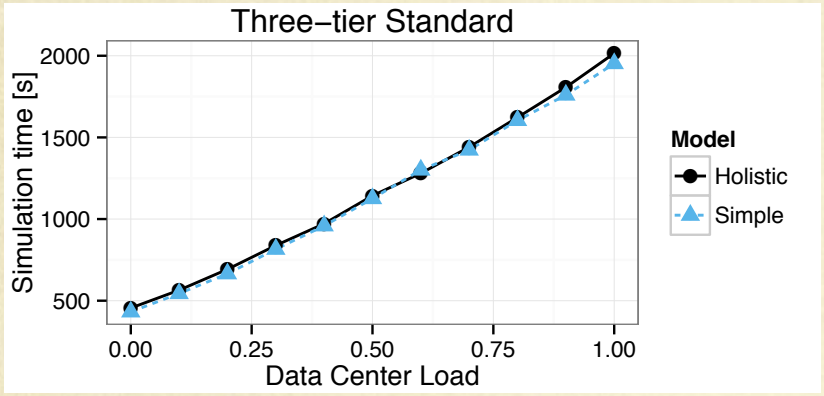
Remaining work packages:

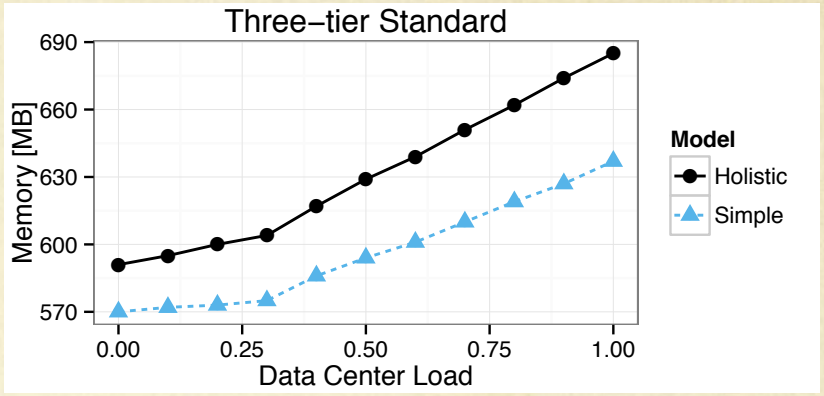
1. Dynamic VM allocation
2. Cloud users utility functions

All these changes greatly change the simulator, so the release of new version is planned (October,2013).

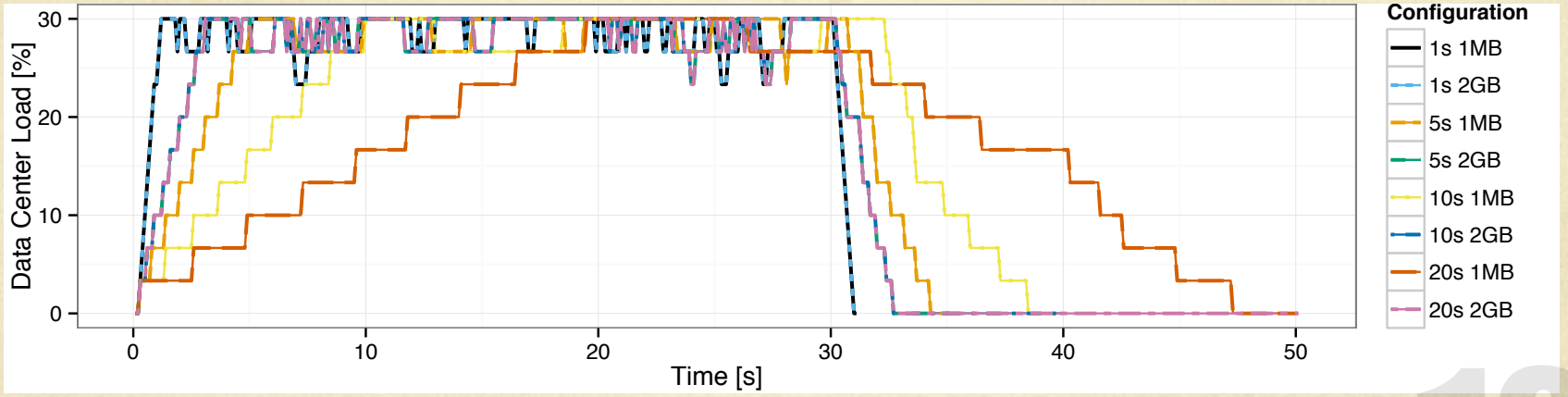
GreenCloud2 – first results

Performance





Sample of traces

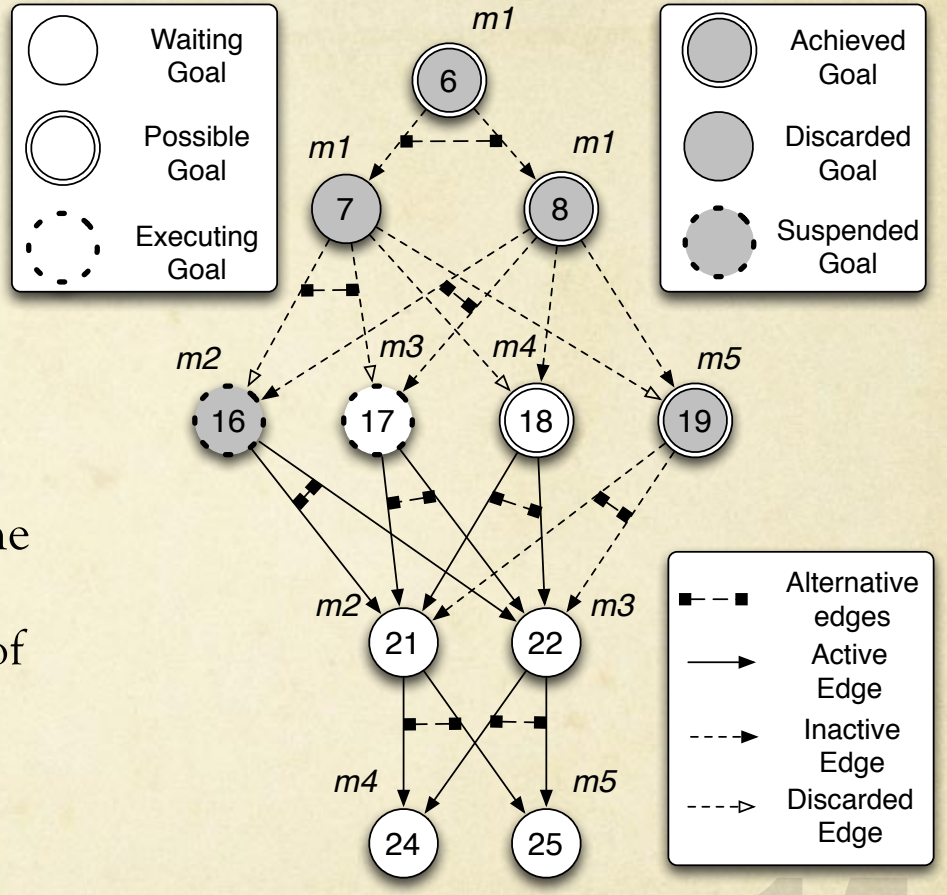


A Holistic Model for Resource Representation in Virtualized Cloud Computing Data Centers Guzek, Mateusz; Kliazovich, Dzmitry; Bouvry, Pascal in *CloudCom2013 (accepted)*

MULTI-AGENT SYSTEMS

Multi-agent Systems – ParaMoise

1. ParaMoise model uses workflows to synchronize cooperation of agents:
   - Enables concurrent, parallel execution and reorganization
2. Efficiency of implementation depends on the system size and the properties of the environment.
   - The reorganization properties of ParaMoise can help
   - Further concepts for MAS organization were presented.



System Design and Implementation Decisions for ParaMoise Organizational Model Guzek, Mateusz; Danoy, Grégoire; Bouvry, Pascal in *AgentDay 2013, FedCSIS2013*

Conclusions – next steps

1. Release of GreenCloud2 – a convergence of main test platform, October 2013
2. Design/implementation of new heuristics for task and vm allocation
3. Identification of the most suitable agent platform (preferably C++)
4. Implementation of multi-agent optimization tool
5. Integration of evolutionary algorithms (ParadisEO) and GreenCloud2
6. Exploitation of the monitoring and modeling framework
7. Implementation of energy-efficient schedulers for OpenStack/ OpenNebula

Thank you for your attention