Energy-aware server provisioning

Daniel Balouek-Thomert [1,2]
Under the supervision of Eddy Caron, Gilles Cieza and Laurent Lefèvre

[1] Avalon Team
LIP, ENS Lyon

[2] NewGeneration SR

GreenDays
Toulouse, France
16-17 Mars 2015

2nd year PhD Student

newgenerationSR
Créateur de valeur pour les entreprises et les territoires

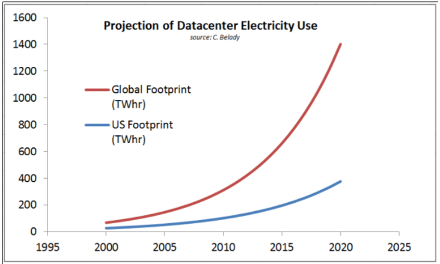
- French start-up revolving around technology and environmental concerns.
- Investigating software oriented energy aware techniques in large scale and distributed environments
- http://www.newgeneration-sr.com

ENS DE LYON

- AVALON research team in LIP laboratory
- Design models, systems, and algorithms to execute applications on resources
- http://avalon.ens-lyon.fr

Context

- Electric consumption of servers accross the world doubled between 2005 and 2010
- ICT = 2 % of C02 emissions

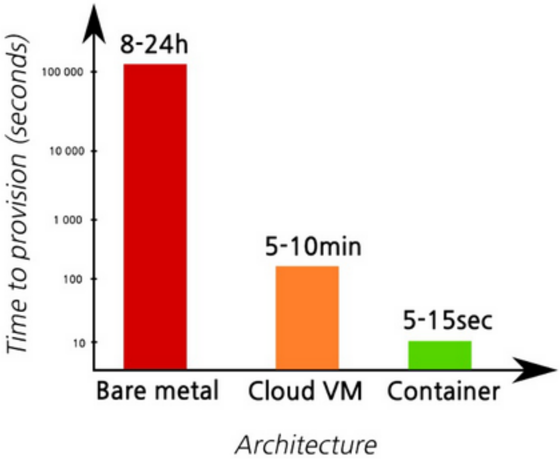


– PROJECTING ANNUAL NEW DATACENTER CONSTRUCTION MARKET SIZE – March 2011

Explosion of services: The Apple Example

- 300,000 Apps for iPad/800,000 for pour iPhone
- 45 000 square meters datacenter dedicated to the selling of Apps and the operating of Itunes software

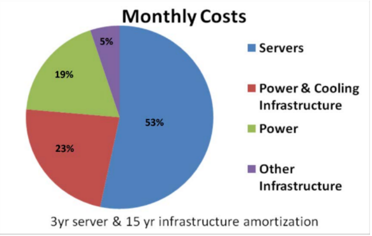
Time to launch a new instance



Motivation

Electric consumption reprensents more than 42 % of a datacenter total budget

- Supply of electricity
- Cooling of components



Source: Hamilton 2009

Aim

- Consuming less energy
- Generating less heat
- Minimizing performance degradation
- Keeping a scalable infrastructure

Our Approach

- Profiling: Know your hardware before you get to know your jobs
- Placement: Where should I put this task?
- Event management: What is happening on my platform?

Profiling of servers

Goal : Favorize the output of servers

Static Profile

- Initial calibration of the hardware
- Observation of disparities (up to 20 %) between similar nodes (Diouri et al., 2013)

Do not trust the hardware

Dynamic Profile

- Systematic collection of usage metrics
- Maximization of the server's output
- Dynamic adaptation of the workload

Provider and User preferences

Aim: Taking into account the willingness to be energy-efficient

User preference

- Indicate a trade-off between performance and energy savings

$Preference_{user} \in [-1, 1]$.

$$Preference_{user} \begin{cases} -1 & \Leftrightarrow & \text{maximize performance} \\ 0 & \Leftrightarrow & \text{no preference} \\ 1 & \Leftrightarrow & \text{maximize energy efficiency} \end{cases}$$

Provider preference

- Determine the number of resources available for computation

Be c the electricity cost and u the resource usage

$$Preference_{provider}(u, c) \rightarrow \frac{(1-c)+2u}{3}$$

Event management

Goal : Reactive dimensioning of the resources

Energy cost

- Favor the use of resources in off-peak periods
- Taking advantage of the negotiations cost

Conditions of temperature

- Avoiding excessive wear of components
- Prevent exploitation incidents

The DIET Middleware

Distributed Interactive Engineering Toolbox

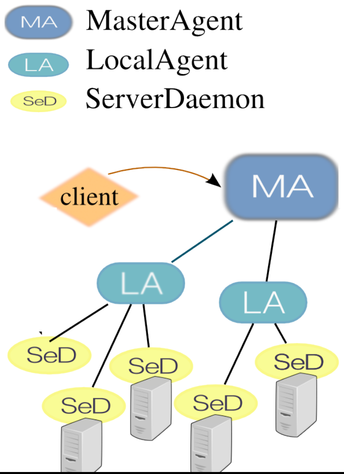
Middleware for high-performance computing in heterogeneous and distributed environments

- Grid-RPC Paradigm
- Hierarchical structure : Scalability and Performance
- Open-Source, based on standards protocols
- Workstations, clusters, grids, clouds

Use in various scientific fields

Simulation, BioInformatics, Cosmology, Meteorology, ...

Scheduling process



Master Agent Propagates the requests and applies a scheduling criteria

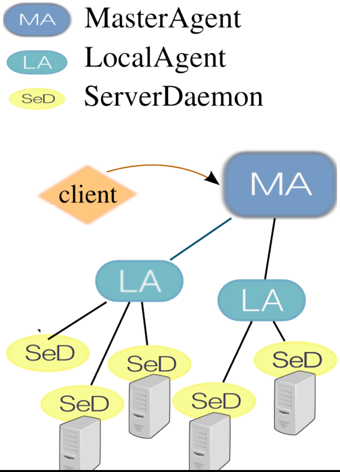
Local Agent Performs (if needed) a sort of the servers

Server Daemon Collects and sends the performance estimation each computational resource

Plug-in schedulers

- Allow the developer to address specific needs over the scheduling subsystem
- Collection of performance estimation values

Scheduling process



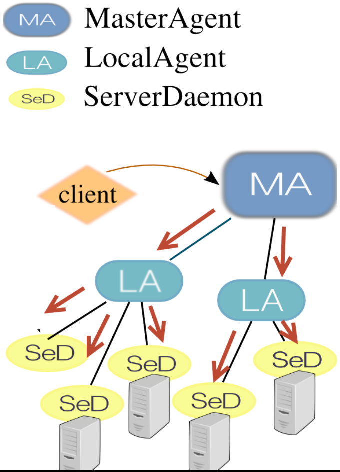
Master Agent Propagates the requests and applies a scheduling criteria

Local Agent Performs (if needed) a sort of the servers

Server Daemon Collects and sends the performance estimation each computational resource

A client submits a request

Scheduling process



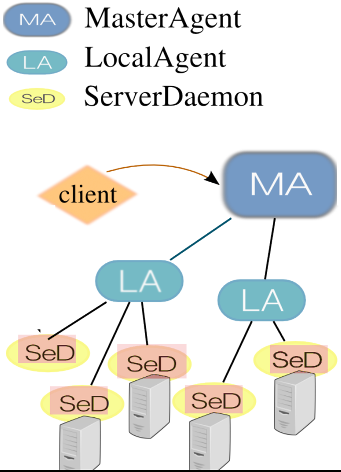
Master Agent Propagates the requests and applies a scheduling criteria

Local Agent Performs (if needed) a sort of the servers

Server Daemon Collects and sends the performance estimation each computational resource

The Master Agent contacts the available SeDs

Scheduling process



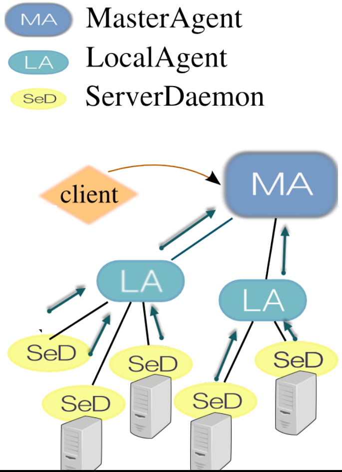
Master Agent: Propagates the requests and applies a scheduling criteria

Local Agent: Performs (if needed) a sort of the servers

Server Daemon: Collects and sends the performance estimation each computational resource

Each SeD retrieves its performance metrics

Scheduling process



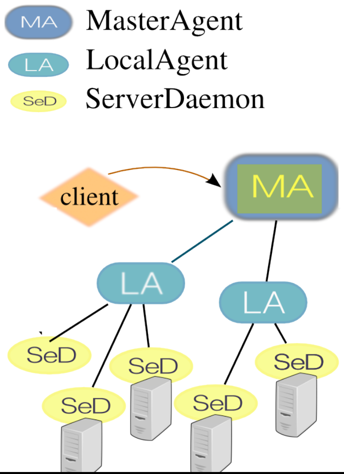
Master Agent Propagates the requests and applies a scheduling criteria

Local Agent Performs (if needed) a sort of the servers

Server Daemon Collects and sends the performance estimation each computational resource

Performance metrics are forwarded to the Master Agent

Scheduling process



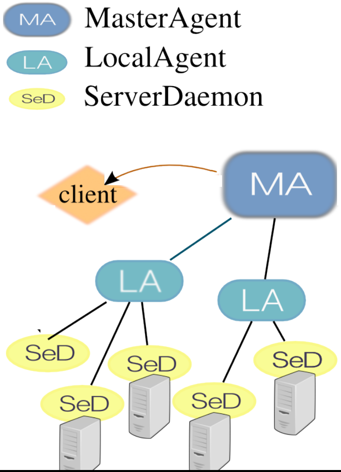
Master Agent Propagates the requests and applies a scheduling criteria

Local Agent Performs (if needed) a sort of the servers

Server Daemon Collects and sends the performance estimation each computational resource

SeDs are sorted based on the scheduling criteria

Scheduling process



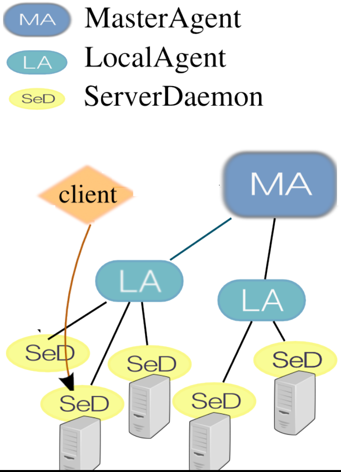
Master Agent Propagates the requests and applies a scheduling criteria

Local Agent Performs (if needed) a sort of the servers

Server Daemon Collects and sends the performance estimation each computational resource

The name of the first server is returned to the client

Scheduling process



Master Agent Propagates the requests and applies a scheduling criteria

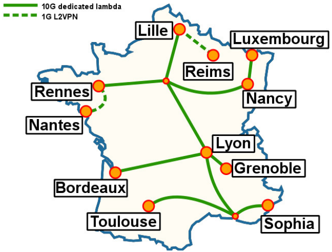
Local Agent Performs (if needed) a sort of the servers

Server Daemon Collects and sends the performance estimation each computational resource

The client adresses his request at the selected server

The Grid'5000 testbed

- Use of 3 clusters from Lyon site (Taurus, Orion, Sagittaire)
- Power consumption measures obtained using the Grid'5000 API

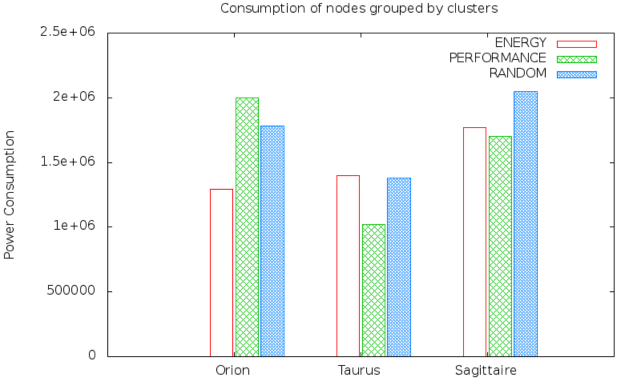


Comparison of 3 scheduling criteria

- ENERGY : Number of requests/power consumption
- PERFORMANCE : Number of floating operations per seconds (flops)
- RANDOM : Random distribution

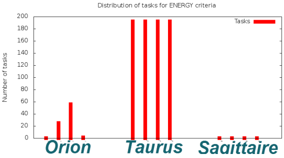
Results (1/2) : Comparison of the scheduling policies

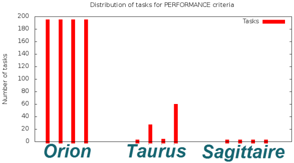
- Execution of 1000+ tasks (CPU intensive)

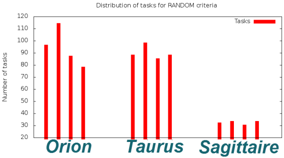


- Significant Energy Savings (up to 25%)
- Minor performance degradations (6

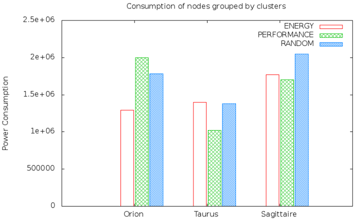
Distribution of tasks (4 nodes per cluster)







- ENERGY Criterion
- PERFORMANCE Criterion
- Random distribution



Context-aware provisionning : Events and Metrics

Events considered

- Scheduled events : Energy provider planning, normal conditions of temperature
- Unexpected events : Heat alerts, Hardware incidents

Considered metrics

- Energy cost
- Temperature

Taking into account the context of execution : Agenda et administrator rules

At each time interval, the scheduler checks the values of the metrics...

Comparaison with thresholds

```
<timestamp value="1385896446">
  <temperature>23.5</temperature>
  <candidates>8</candidates>
  <energy_cost>0.6</energy_cost>
</timestamp>
```

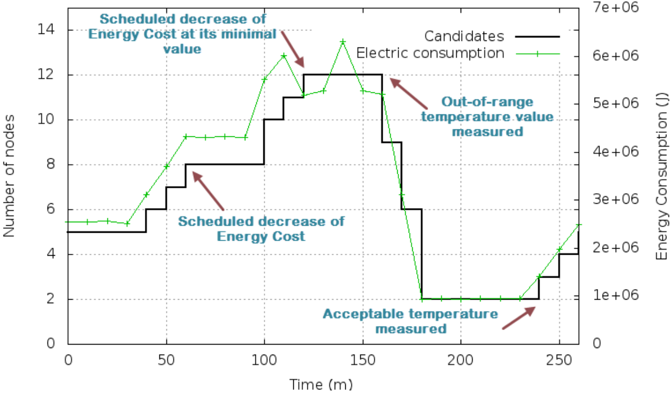
Figure: Sample of the agenda file

... and apply predefined rules.

Example : If the electricity price is below 0.5, the number of available servers (i.e. candidate servers) is set to its maximum value

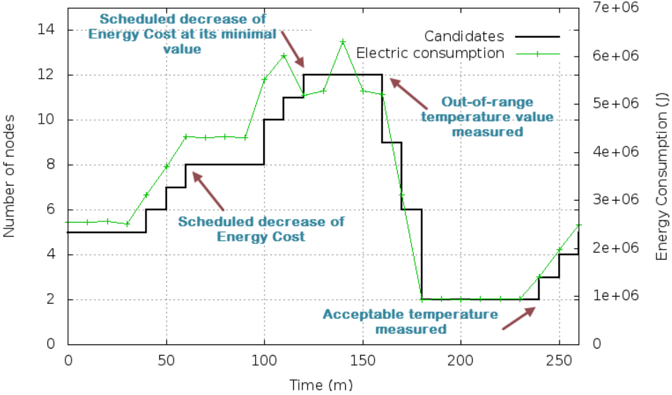
Results(2/2) : Automation of event management

Simulating the fluctuations of energy price and temperature



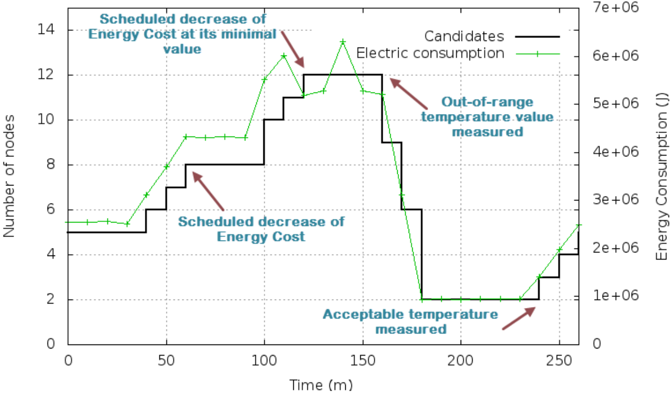
Startup : 5 computational servers available

Results(2/2) : Automation of event management



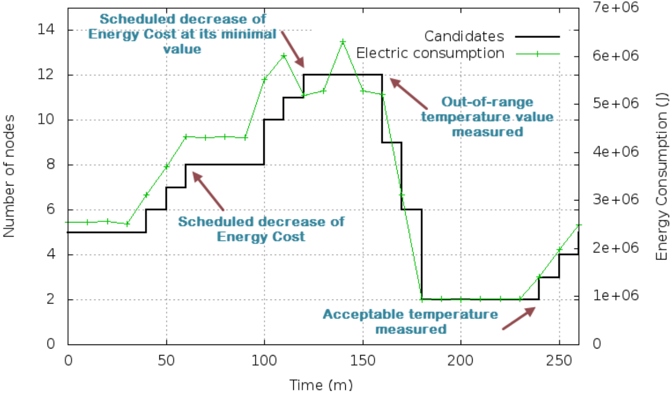
Event 1 : First decrease of energy cost. Progressive incrementation by a subset of nodes

Results(2/2) : Automation of event management



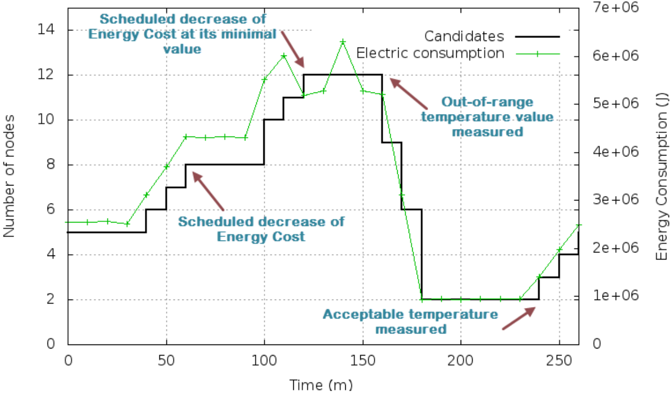
Event 2 : Second decrease of energy cost. The whole set of computationnal servers is in use

Results(2/2) : Automation of event management



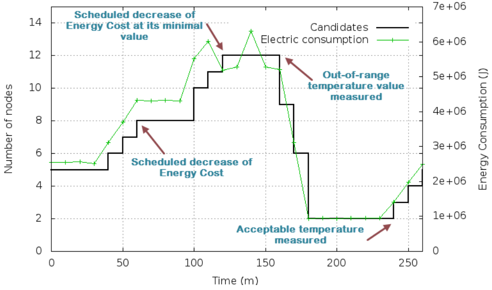
Event 3 : Instant raise of temperature. Minimal use of the infrastructure (2 servers)

Results(2/2) : Automation of event management



Event 4 : Acceptable temperature is measured. The number of nodes is incremented.

Results(2/2) : Automation of event management



Simulation of fluctuations of energy price and temperature

- The energy consumption of the infrastructure is automatically adapted according to the happening of events
- Third party monitoring/predicting tools can be integrated

Ongoing work

Towards a dynamic Service Level Agreement

- Taking into account the side effects of consolidation
- Maintaining an efficient use of resources

Integrating the cost of operations in the model

- Migrations, Reconfigurations, Standby, Shutdown... When is it relevant to execute them?

Summary

Dynamic characterization of servers profile

- Computational behavior on-the-fly
- Distribution of tasks according specific criterias
- Significant energy saving with minimal impact on the performance

Managing energy-related events

- Use of pre-defined thresholds
- Simple integration of third-party tools

Perspectives

- Taking into account spatial information
- Budget constrained scheduling

More information

To appear

Daniel Balouek-Thomert, Eddy Caron, Laurent Lefevre
Energy-Aware Server Provisioning by Introducing Middleware-Level Dynamic Green Scheduling
In HPPAC 2015: Workshop on High-Performance, Power-Aware Computing

Tutorial at the Grid'5000 School

- Using real-time data in your experiments

https://www.grid5000.fr/mediawiki/index.php/Kwapi_2014_School_tutorial

Thanks

Eddy Caron, Laurent Lefevre, Gilles Cieza, Barbara Walter, Joran Bigalet, Arrate Magdaleno, Laurent Pouilloux, François Rossigneux

Thanks for your attention!

Any questions?

daniel.balouek@ens-lyon.fr
http://graal.ens-lyon.fr/~dbalouek