CNRS - INP - UT3 - UT1 - UT2J

Institut de Recherche en Informatique de Toulouse

Comment évaluer / réduire / optimiser la consommation d'énergie dans les Data Center / Cloud instances et l'impact sur l'environnement

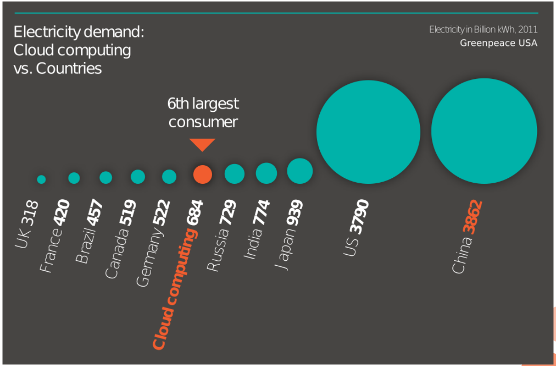
Georges Da Costa

dacosta@irit.fr

ENERGUMEN data ZERO neOCampus

IT impact on electricity

- Recent datacenters: 40000 servers, 500000 services (virtual machines). Google, Facebook > 1 million servers
- One major power consumer
  - 2000 : 70 TWh
  - 2007 : 330 TWh, 2% of CO_2 world production
  - 2011 : 6^{th} electricity consumer in the world
  - 2020 : 1000 TWh
- Rising
  - 2014 to 2016: 90% of datacenters were expected to need hardware upgrades

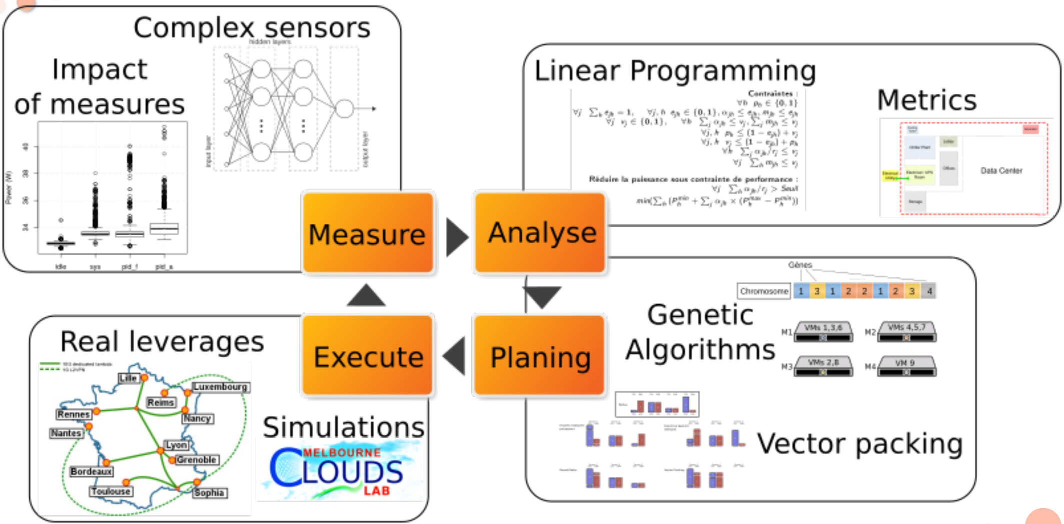


Sustainable datacenters

- Action can be done at several different levels
  - Hardware level: changing servers or cooling system
    - If entropy is constant, theoretical energy consumtion is 0 !
  - Application level: rewrite applications while changing paradigm*or library
  - Middleware level: manages servers and services/applications
- Middleware: minimal cost, maximal impact
  - Clouds: OpenStack: 30% of market share in 2014
    - OpenSource solutions: 43% in 2014 (+72% in 2 years)
  - HPC: Slurm, LSF, plugin based

* Georges Da Costa et al. *Exascale machines require new programming paradigms and runtimes*, SFI journal, 2015

Complete loop



Plan

Model a system

To manage a system, we need to:

- Know all possible actions
- Know which is(are) the best one(s)

It can be translated into:

- Modeling impact and means (time, energy,...) of these actions
- Being able to compare two scenarios

Impact of leverages, an example with DVFS

Dynamic electric power consumed by a CMOS component:

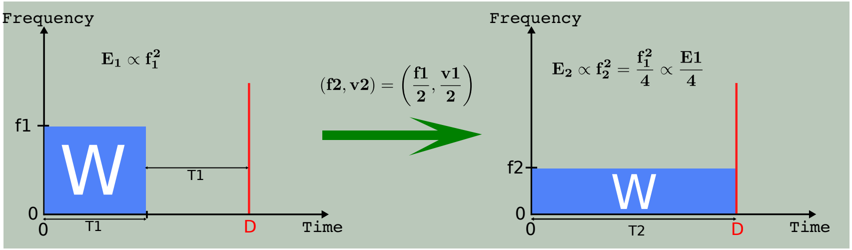
$P_{cmos} = C_{eff} \times V^2 \times f$

with, C_{eff} the effective capacitance *, V the voltage and f the frequency

* physical quantity: capacity of a component to resist to the change of voltage between its pins

Energy consumed for each tasks:

$$E = P * T \propto T * V^3 \text{ , avec } V \propto f \text{ et } T \propto 1/F \text{, alors } E \propto f^2$$



Even models are complex

Electrical power models for a single server:

- Classical : linear (error E∼10-15%)

$$Power = P_{min} + Load \times (P_{max} - P_{min})$$

Even models are complex

Electrical power models for a single server:

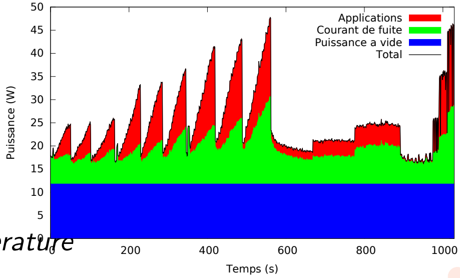
- Classical : linear (error E∼10-15%)
- Finer : Processor voltage/frequency (E∼5-9%)

$$Power = P_{min} + Load \times \alpha Voltage^2 Frequency$$

Even models are complex

Electrical power models for a single server:

- Classical : linear (error E∼10-15%)
- Finer : Processor voltage/frequency (E∼5-9%)
- Even finer: Processor temperature (E∼4-7%)



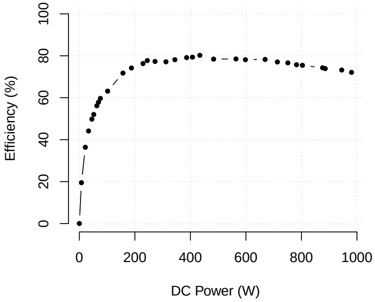
$$Power = P_{min} + Load \times \alpha Voltage^2 Frequency + \lambda Temperature$$

Even models are complex

Electrical power models for a single server:

- Classical : linear (error E∼10-15%)
- Finer : Processor voltage/frequency (E∼5-9%)
- Even finer: Processor temperature (E∼4-7%)
- Do not forget about bias: **power supply unit E∼2-3%**, cooling, ...

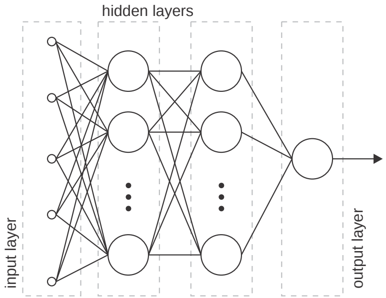
$$Power_{DC} = \omega_0 + \omega_1 Power_{AC} + \omega_2 Power_{AC}^3$$



Even models are complex

Electrical power models for a single server:

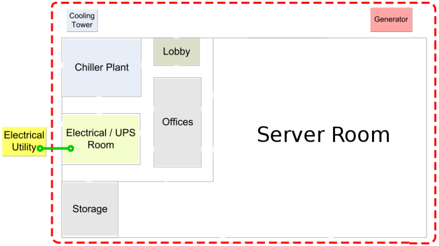
- Classical : linear (error E∼10-15%)
- Finer : Processor voltage/frequency (E∼5-9%)
- Even finer: Processor temperature (E∼4-7%)
- Do not forget about bias: **power supply unit E∼2-3%**, cooling, ...
- Learning methods (neural networks, E∼2%) *



* Da Costa et al., *Effectiveness of neural networks for power modeling for Cloud and HPC: It's worth it!*, Transactions on Modeling and Performance Evaluation of Computing Systems journal, 2020

PUE : Power Usage Effectiveness

- Ratio Total electricity/IT electricity
- Mean value: 1.7 in 2014
- Standard initiated by GreenGrid
- Where does the IT part stops?
  - Power Supply Unit? Fans on the motherboard? Processor?
- Useful only in a very specific case

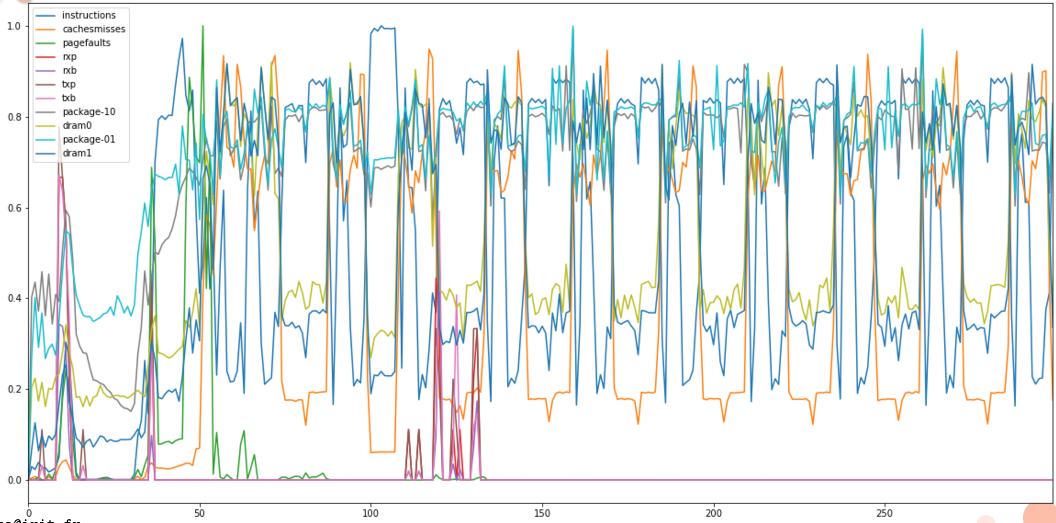


PUE : Power Usage Effectiveness

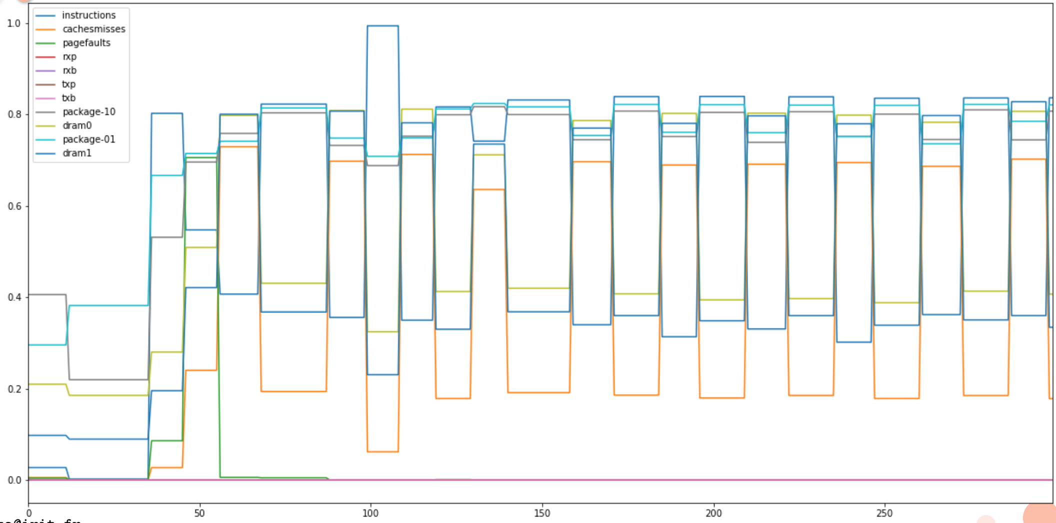
- Ratio Total electricity/IT electricity
- Mean value: 1.7 in 2014
- Standard initiated by GreenGrid
- Where does the IT part stops?
  - Power Supply Unit? Fans on the motherboard? Processor?
- Useful only in a very specific case

- Constant overhead (100), IT part 100 to 200 depending of the load
- For the same service provided by two softwares
  1. Mean load 75%
     PUE = 275/175 = 1.57
  2. Mean load 100%
     PUE = 300/200 = 1.5

LU Diagonalization, raw values



LU Diagonalization, after phase detection



External application identification

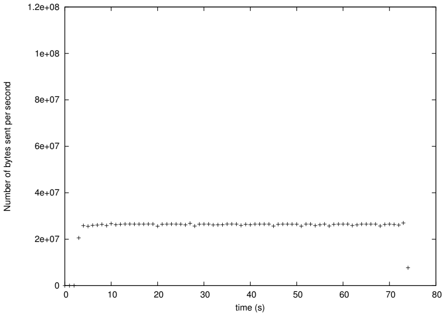
- Monitoring system values is intrusive
- Reduce the number of values monitored
- Using external values has lower impact (power, network)
- Authorize statistic tools
- Study the behavior during time

Georges et al., *Characterizing applications from power consumption : A case study for HPC benchmarks*, ICT-GLOW Symposium, 2011

External application identification

- Monitoring system values is intrusive
- Reduce the number of values monitored
- Using external values has lower impact (power, network)
- Authorize statistic tools
- Study the behavior during time

Georges et al., *Characterizing applications from power consumption : A case study for HPC benchmarks*, ICT-GLOW Symposium, 2011

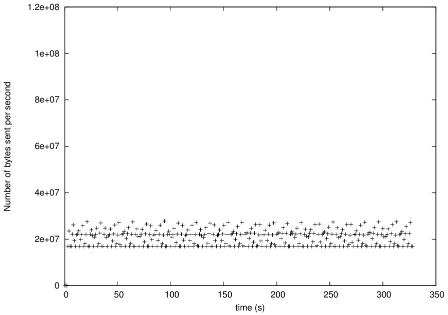


benchmark CG (NPB)

External application identification

- Monitoring system values is intrusive
- Reduce the number of values monitored
- Using external values has lower impact (power, network)
- Authorize statistic tools
- Study the behavior during time

Georges et al., *Characterizing applications from power consumption : A case study for HPC benchmarks*, ICT-GLOW Symposium, 2011

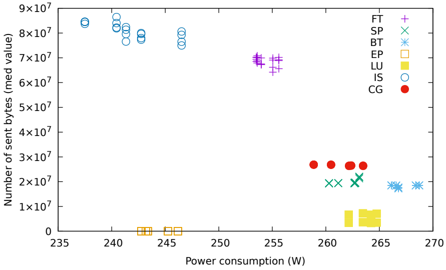


benchmark SP (NPB)

External application identification

- Monitoring system values is intrusive
- Reduce the number of values monitored
- Using external values has lower impact (power, network)
- Authorize statistic tools
- Study the behavior during time

Georges et al., *Characterizing applications from power consumption : A case study for HPC benchmarks*, ICT-GLOW Symposium, 2011



Evaluation tools: Experimentation, Simulation

- To improve, comparison is necessary
- Three main methods
  - Mathematical models
  - Simulation
  - Experiments

Linear programming

- Describe all constraints with linear equations

Example : A task is on a unique server

- Let e_{jh} the fact that task j runs on server h
- $e_{jh} = 1$ iif task j is on server h
- $\forall j, h \;\; e_{jh} \in \{0, 1\}$,
  $\forall j \;\; \sum_h e_{jh} = 1$

Linear programming

- Describe all constraints with linear equations
- Describe the objective as a function to minimize

Example : Minimize the total power consumed

- P_h^{stat} et P_h^{dyn} : static and dynamic power of server h (linear model)
- Let α_{jh} the processor fraction of task j on server h
- $min \sum_h (P_h^{stat} + \sum_j \alpha_{jh} P_h^{dyn})$

Linear programming

- Describe all constraints with linear equations
- Describe the objective as a function to minimize
- Formalize leverages and their impact
- Approximation of real world (quadratic phenomena)
- Exact resolution for small cases

Constraints :

$$\forall h \quad p_h \in \{0,1\}$$
$$\forall j \quad \sum_h e_{jh} = 1, \quad \forall j,h \quad e_{jh} \in \{0,1\}, \alpha_{jh} \leq e_{jh}, m_{jh} \leq e_{jh}$$
$$\forall j \quad v_j \in \{0,1\}, \quad \forall h \quad \sum_j \alpha_{jh} \leq v_j, \sum_j m_{jh} \leq v_j$$
$$\forall j,h \quad p_h \leq (1 - e_{jh}) + v_j$$
$$\forall j,h \quad v_j \leq (1 - e_{jh}) + p_h$$
$$\forall h \quad \sum_j \alpha_{jh}/r_j \leq v_j$$
$$\forall j \quad \sum_h m_{jh} \leq v_j$$

Minimize power under performance constraints:

$$\forall j \quad \sum_h \alpha_{jh}/r_j > Threshold$$
$$min(\sum_h (P_h^{min} + \sum_j \alpha_{jh} \times (P_h^{max} - P_h^{min}))$$

Damien et al., *Energy-Aware Service Allocation*, FGCS journal, 2011

Simulation

- Large number of simulators: SimGrid, DCWorms, CloudSim, ...
- Particular needs for our research
  - Cloud models (migration, Over-allocation of resources, federation†)
  - DVFS
  - Electrical power
  - Temperature
- Situation is steadily improving
  - DVFS and fine-grained management of clouds in CloudSim
  - Thermal simulation in DCWorms*
  - DVFS and energy in SimGrid
- Now mainly BatSim

* Wojtek et al., *Energy and thermal models for simulation of workload and resource management in computing systems*, SMPT journal, 2015. †Thiam et al., *Cooperative Scheduling Anti-load balancing Algorithm for Cloud*, CCTS workshop, 2013

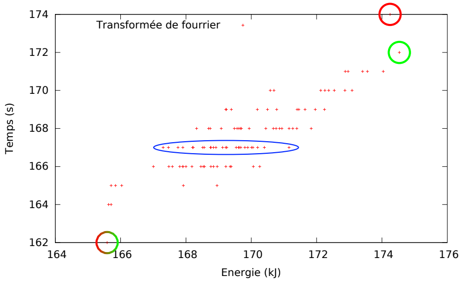
Experimentation

- A model is always an approximation
- Final validation by experiment
- Complex because of the need to have electrical measures
  - At ENS-Lyon, they where one of the first to experiment with watt-meters at large scale (GreenNet)*
- Problem of distributed measures, electrical conversions, impact of measures (performance counters)
- Reproducibility problem

* Georges et al., *The green-net framework: Energy efficiency in large scale distributed systems*, IPDPS, 2009

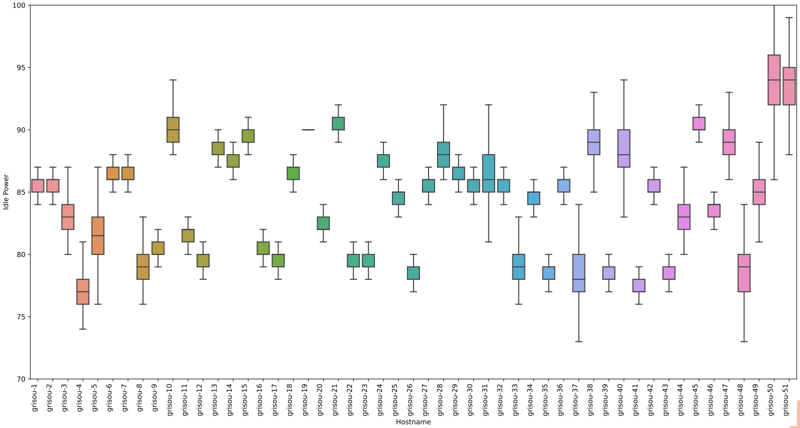
No stability of experiments

- *Simple* experiment of Fast Fourier Transform (NPB)
- 100 experiments on exactly the same hardware (Grid'5000)
- Large variations
  - Time: 12s, 7% (Std. Dev. 3.2s)
  - Energy: 9.3kJ, 5.5% (3kJ)
- For the same time, 167s
  - Difference of 4kJ
- Time $\neq$ Energy



Heterogeneity between servers

- *Idle* servers
- Same hardware (Grid'5000)
- Same O/S: Debian 10.3, Linux kernel 4.19.0-8-amd64 Duration 500s
- Large variations
  - Min: 77.4W
  - Max: 93.8W



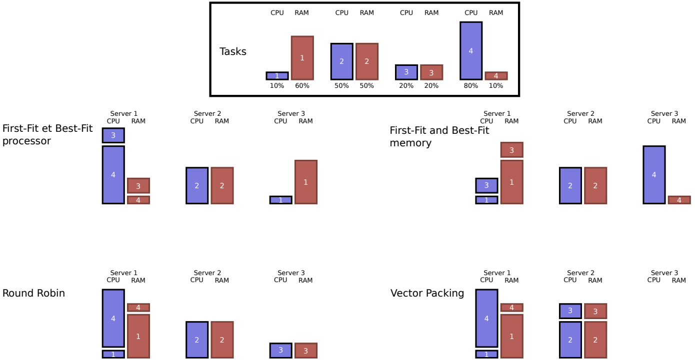
Plan

Exact Approaches and heuristics

- Two problems
  - Placement
  - Temporality
- Classical heuristics for placement
  - Greedy: Best Fit, First Fit
  - Vector Packing (*Gourmet* Greedy)
  - Genetic algorithms

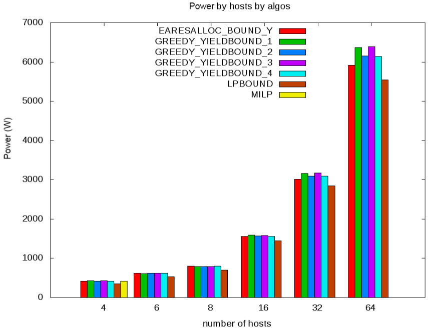
Classical greedy algorithms

- Characteristics
  - Memory
  - Processor
- Sort services
- Sort servers
- No coming back on previous decisions



Gourmet Vector packing

- 4 objectives in the sort function
  - Server is attractive from an energy point of view
  - Add the task do not overload the server
  - Server already switched on
  - The tasks brings back the balances of resources
- Time "only" in $\mathcal{O}(J \times H \ln(H))$
- But the solution of the *Gourmet* is difficult to qualify

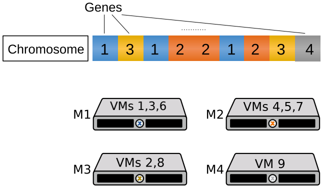


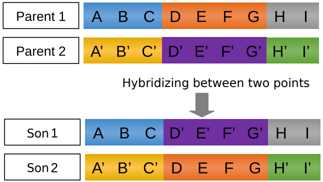
Damien et al., *Energy-Aware Service Allocation*, FGCS journal, 2012.

Genetic Algorithms

- Chromosome = Allocation
- Initial random generation
- At each generation:
  - Hybridizing and mutation
  - Sort on the objective metric
  - Keep only the best

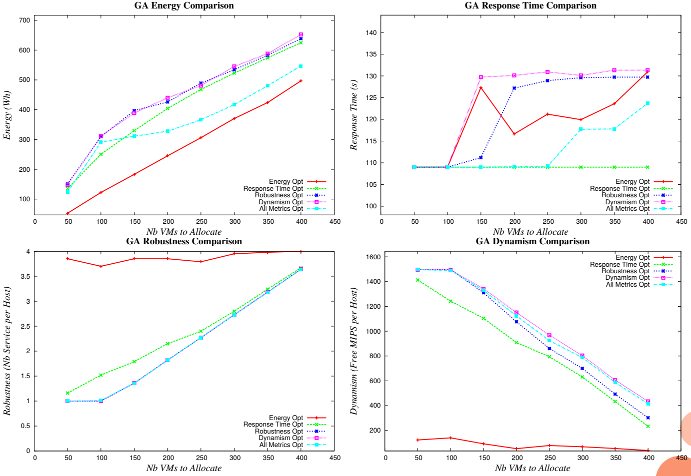
Tom et al., *Quality of Service Modeling for Green Scheduling in Clouds*, SUSCOM journal, 2014





Results of the genetic algorithm

- Each algorithm is the best in its own domain (Energy)
- GA_All Very good everywhere
- 400 services on 110 servers, approximately 40s
- Taking into account a metric is already very important



Plan

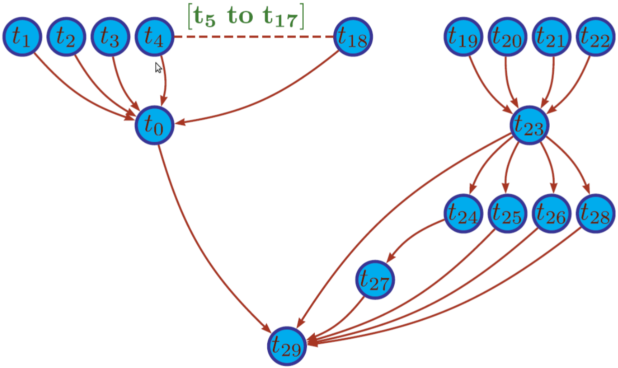
Node optimization

- Three temporalities
  - Large-grained (minute) : Optimal frequency in function of the task graph*
    - 13% of energy savings
  - Medium-grained (second) : Phase detection†
    - 20% of energy savings, 3% of time increase
  - Fined-grained (1/10s) : Frequency policy at the kernel level‡
    - 25% of energy savings, 1% of time decrease
- No coordination between the three temporality, no objectives

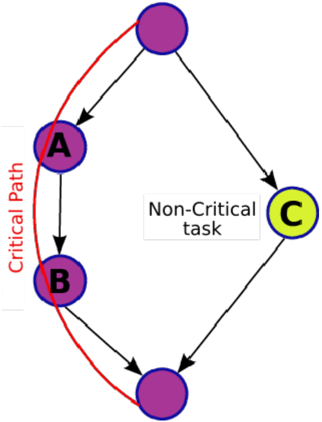
* Tom et al., *Energy-aware simulation with DVFS*, SMPT journal, 2013 †Landry et al., *Exploiting performance counters to predict and improve energy performance of HPC systems*, SUSCOM journal, 2014 ‡Georges et al., *DVFS governor for HPC: Higher, Faster, Greener*, PDP conference, 2015

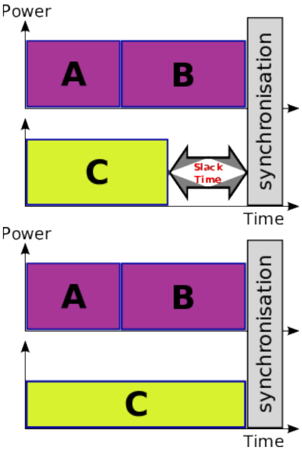
At the scale of a node: Large-grained

- Use of contextual external information
- Example at the scheduler level: Task DAG

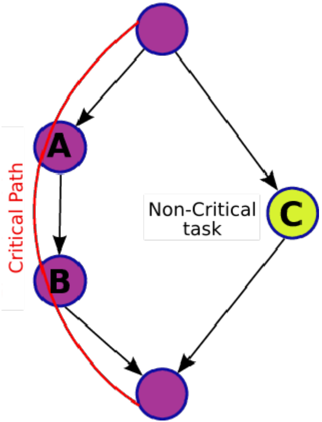


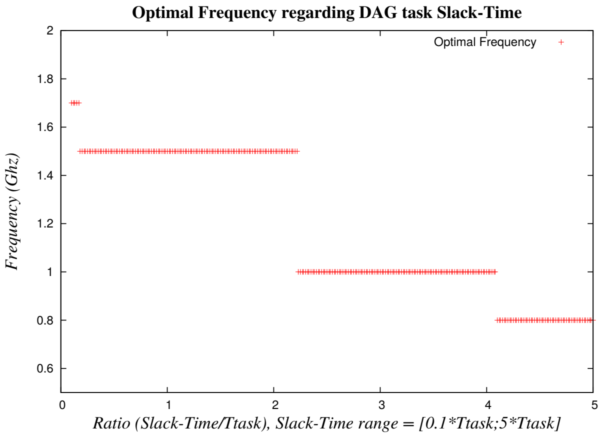
Coordination of node speeds





Coordination of node speeds





Open research questions

- Programming paradigms
  - Ability to describe parallelism intuitively
  - Remove the burden from developer
- Runtimes
  - Capability to adapt to particular profiles and their interactions
    - Monitoring & prediction
  - Ability to change kernels in function of context
- Communication between these two levels
- Improvement of RJMS (Resources and Job Management Systems)
  - Spatial management
  - Temporal management