Energy efficiency and Cloud Computing Works in SEPIA Team

Jean-Marc Pierson

SEPIA Team
Distributed systems, Cloud, HPC
IRIT – Institut de Recherche en Informatique de Toulouse
UPS – UniversitéToulouse 3 Paul Sabatier

CloudDays, Nantes, Septembre 2014

SEPIA

Distributed Operating Systems, from Architecture to Middleware

26 researchers. 11 permanent.
PR: JM. Pierson, D. Hagimont, JP. Bahsoun, A. Mzoughi.
MCF: G. Da Costa, P. Stolf, J. Jorda, A. Tchana, L. Broto.
IE: F. Thiebolt.

- Placement, scheduling, migration of Tasks and Virtual Machines in Datacenters: performance, energy, thermal.
- Autonomic management of Clouds and HPC systems (software and hardware levels): DSL Language and MAPE-K loop: Monitoring, Analysis, Planning, Execution, Knowledge.
- Distributed and secured file systems for HPC and Clouds
- Replication and Maliciousness

Outline

Green IT

- ICT consumes a lot :
  - Estimated to 4-5% of global consumption of electricity (930 TWh per year)
  - Annual growth between 5 to 10%
  - Data Centers themselves account for 2% of the global electrical consumption
  - Reminder: 1 nuclear reactor is producing about 900 MWe. (7 TWh per year)

© ferloo.com

What is it all about?

- A Datacenter : 10000 m2
- Exploitation costs: 20 Meuros per year
- 40000 physical servers
- 500000 VMs
- Electrical Consumption: 10 MW
- PUE : 1.2

Facts, resources:

- France Grilles, 2012. `http://cloudmip.univ-tlse3.fr/`
- Fluids consumption annual cost (est.) : between 4keuros and 10keuros.
- Part of the FG Cloud Federation. May become an EGI node.

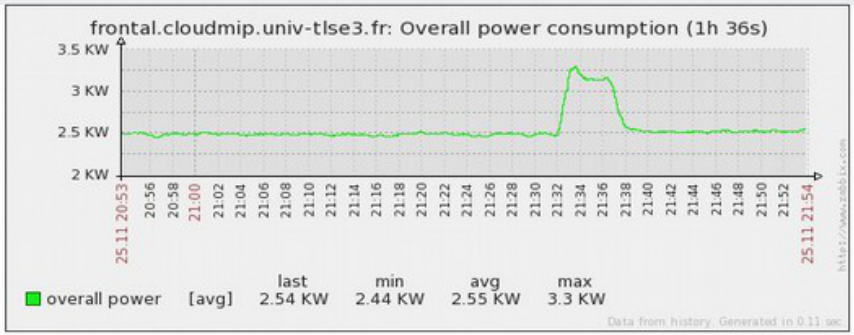
Hardware, OS:

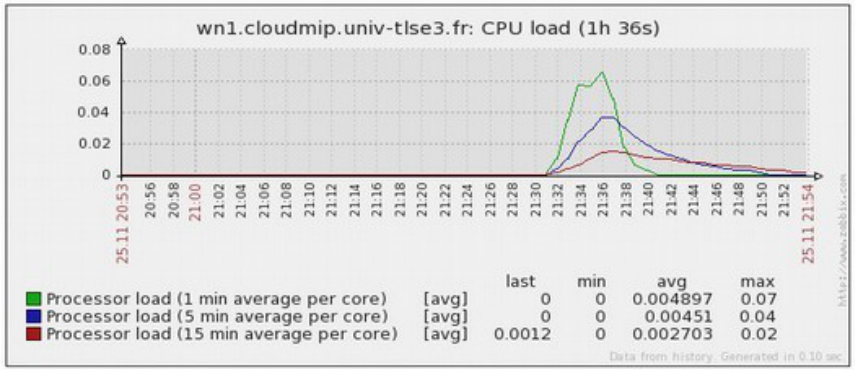
- 32 blades (8 cores @ 2.4Ghz, 32GB ram, 2 x 146GB SAS 15ktpm RAID0), means up to 256 Amazon EC2 M1 instance (1 physical dedicated CPU and 4GB ram)
- System : Scientific Linux 6.5 x86_64,
- OpenNebula 4.2 (KVM) with Qcow2 delta images to speedup deployment (a hundred of VMs in just a few seconds)
- Monitoring 1s : Zabbix —combines Nagios and Ganglia capabilities. `http://cloudmip.univ-tlse3.fr/zabbix`

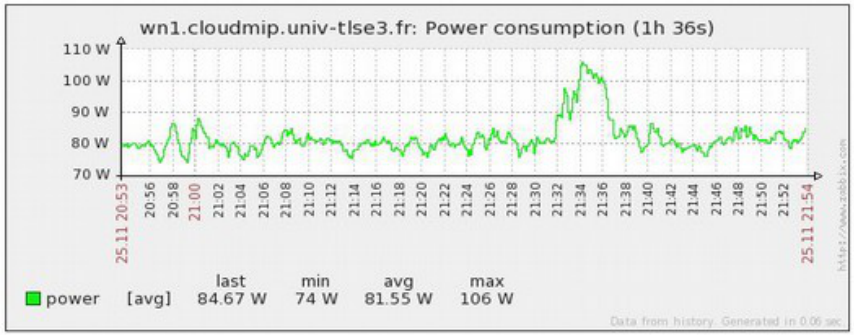
Power consumption on CloudMIP

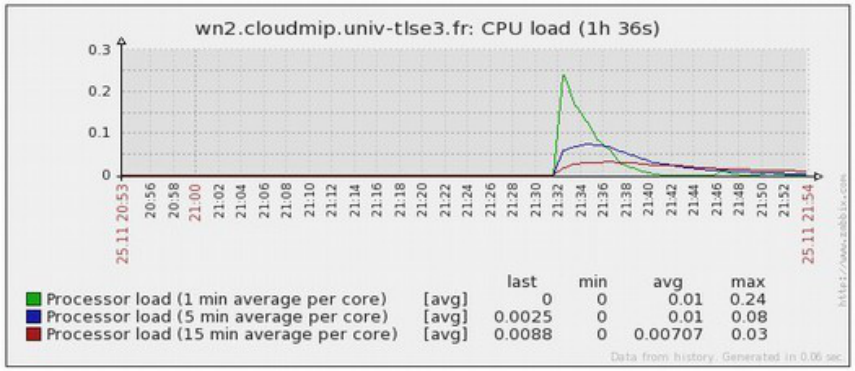
248 VM started: > onetemplate instantiate SL64 -m 248

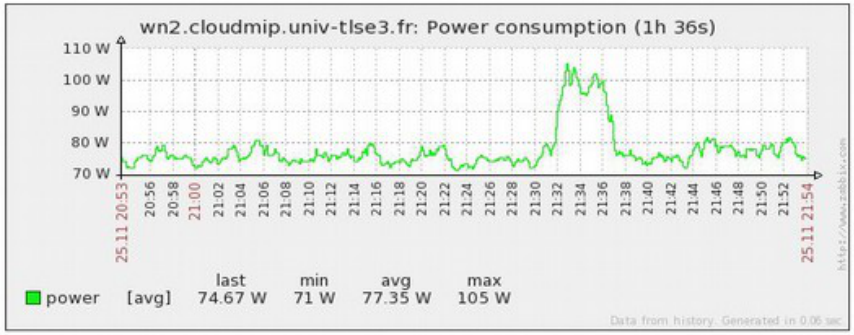
- Leads to 8 VMs on each of the nodes





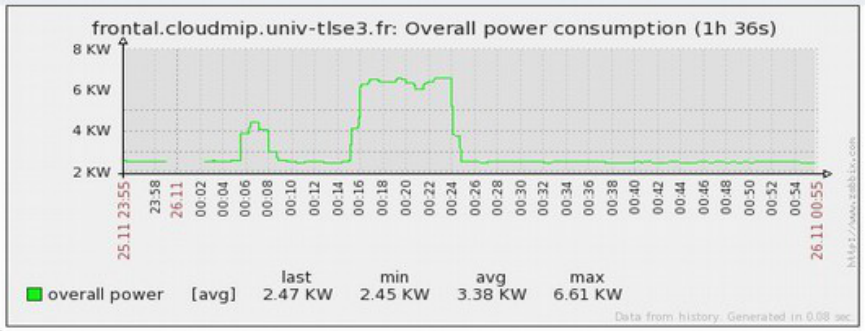


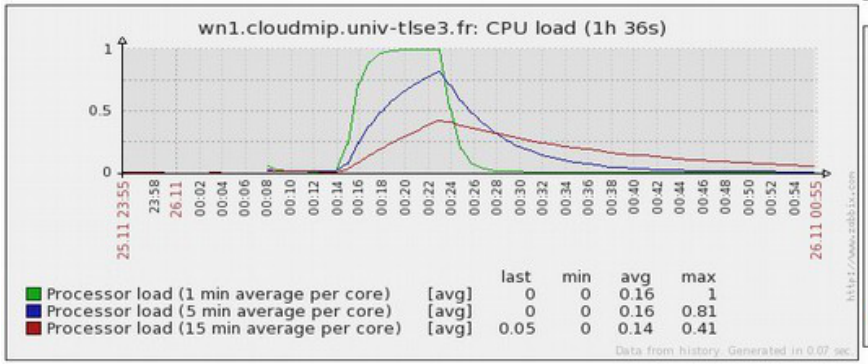


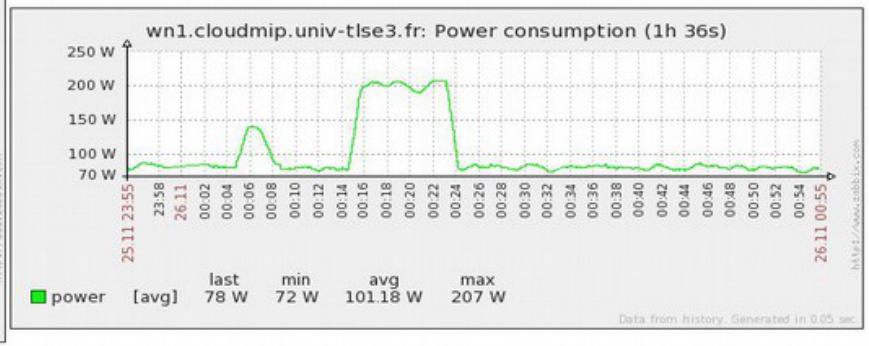


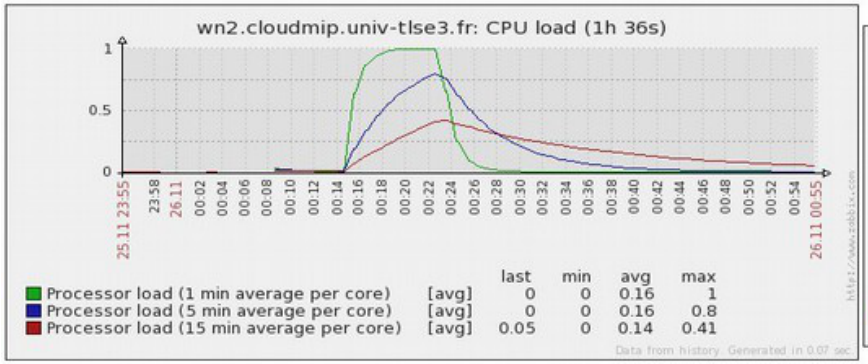
Stress: > pdsh -w wn[1..32] – openssl speed -multi 8

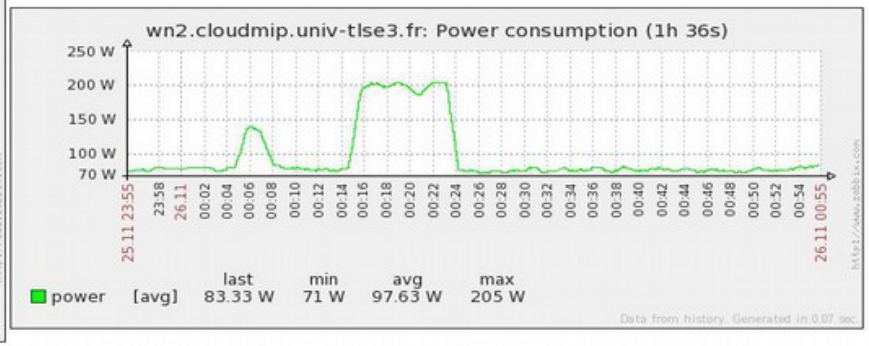
- Power consumption is 6.6kw









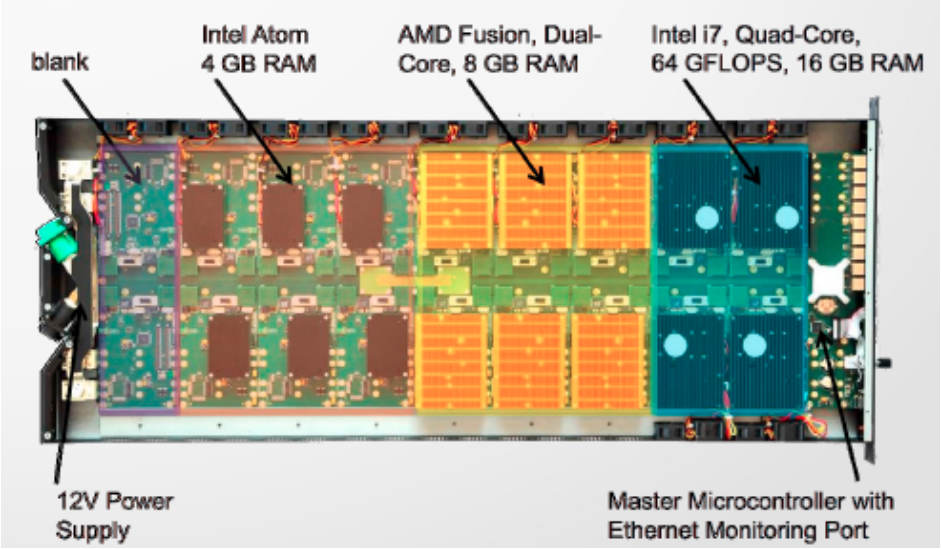


RECS:

- FP7 CoolEmAll, 2012. `http://coolemall.eu/`

Hardware, OS:

- 18 nodes in 1U: 6 i7, 6 Atom, 6 mainboards
- System : Scientific Linux 6.4 x86_64
- Monitoring 1s : Zabbix `http://cloudmip.univ-tlse3.fr/zabbix`



Modeling power consumption

Modeling power consumption

Why measuring, monitoring, estimating power consumed by applications?

1. Make applications' users energy-aware → CO2 labels.
2. Make applications' developers energy aware → applications become energy-friendly
3. Make the operating system energy-friendly:
   - the OS can optimize its behavior according to the profiles of applications
4. Make the middleware energy-aware and energy-friendly:
   - energy-aware: it can redistribute costs to individual users (e.g. in commercial Clouds)
   - energy-friendly: it can optimize dynamically the placement and scheduling of applications on machines

Existing Approaches to Energy Consumption Estimation

Three possibilities for evaluating energy of an application

- Instrument the code of the application, then relate it to actual measures or modeled estimates.
- Monitor the power with (internal or external) power meters, then distribute shares to each process / application based on their resources consumption (CPU, memory, disk, network, ...) ;
- Monitor the usage of resources at the hardware and operating system level (e.g. hardware performance counters, CPU load, ...), then use mathematical models to estimate power consumption share for each process / application ;
  - The Energy Consumption Tools Pack[1]
  - Energy consumption library (libec)
  - Data Acquisition tool (ecdaq). Easy to extend for new power estimators.
  - Data Monitoring tool (ectop and ganglia plugin)
  - Energy profiler (valgreen)

[1] Available under GPL3 licence.

Data Monitoring Tool: ectop

ectop command line tool (a-like 'top', and extensible), based on libec.

```
ectop  - Thu Sep 12 15:31:37 2013
Tasks: 159, CPU (%):0.507353
```

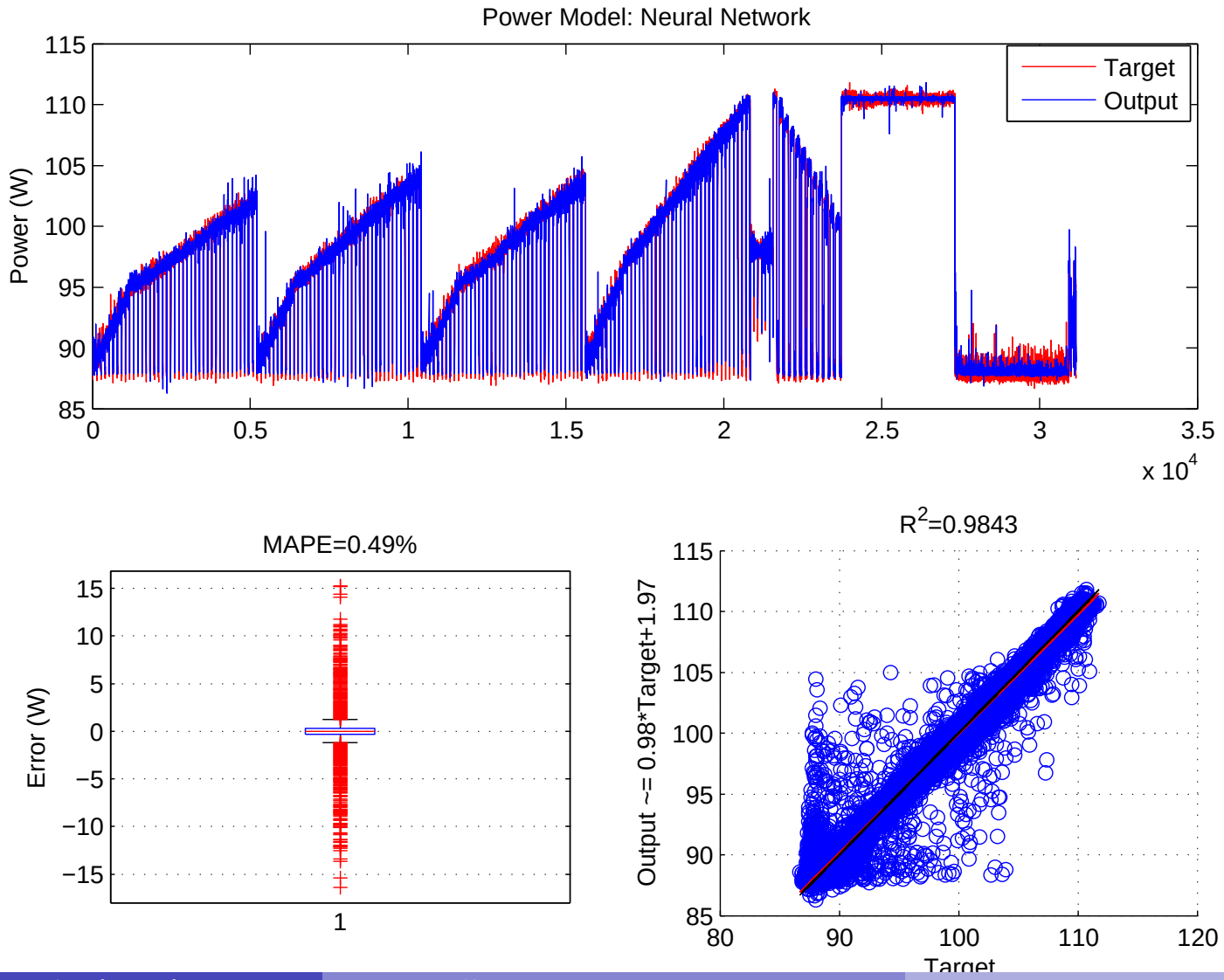
PID	COMMAND	EJIFS	CPUu	MEM_RSS	MEMu	PE_MMC v
11647	qemu-kvm	93	0.227941	194670	0.047595	7.66042
26177	qemu-kvm	52	0.127451	71492	0.0174791	4.34425
28561	lookbusy	17	0.0416667	149	3.64291e-	1.51336
28560	lookbusy	16	0.0392157	149	3.64291e-	1.43248
28562	lookbusy	15	0.0367647	149	3.64291e-	1.3516
28563	lookbusy	12	0.0294118	149	3.64291e-	1.10895
29014	ectop	3	0.0073529	738	0.0001804	0.381012
2223	zabbix_agentd	0	0	455	0.0001112	0.138365
2224	zabbix_agentd	0	0	406	9.92632e-	0.138365
2102	nrpe	0	0	325	7.94594e-	0.138365
24	khelper	0	0	0	0	0.138365
28	sync_supers	0	0	0	0	0.138365
29	bdi-default	0	0	0	0	0.138365
30	kintegrityd/0	0	0	0	0	0.138365
25	netns	0	0	0	0	0.138365
		208	0.509804	287895	0.0703876	38.8235

It exists also in HTML format, with pan view, accumulated values, ...
Lightweight tool: 3 Kb of memory, 0.3% (unsorted columns) to 2% of CPU (when sorted/sum bar/html)

Open source software. Deliverable 5.2 of the CoolEmAll project - October 2012 - Leandro Fontoura-Cupertino
Tested on RECS platform, Grid5000 and CloudMIP

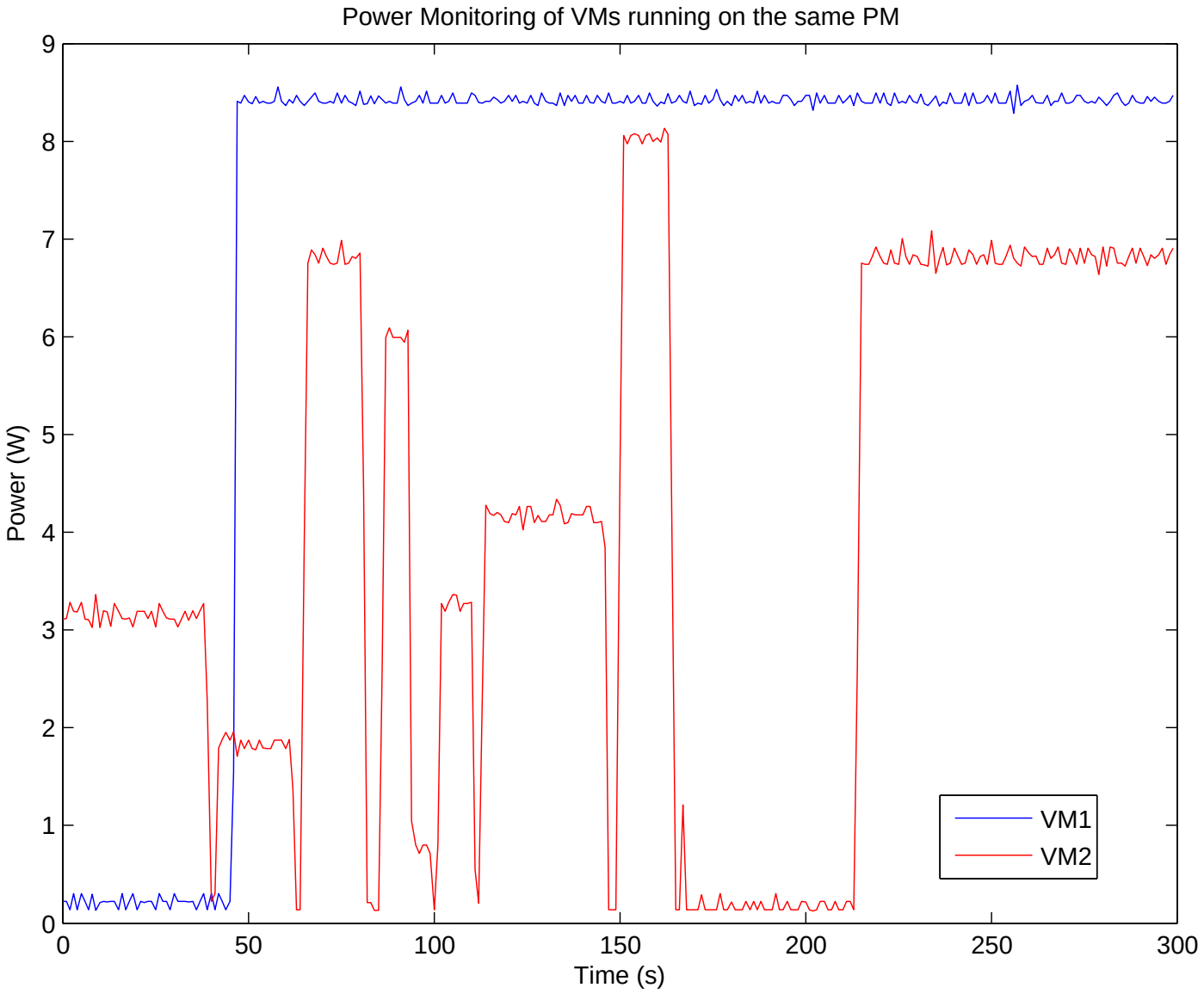
Neural networks models results

High accuracy of the power estimation, compared to actual measurements with precise and reactive power meters.



Following the consumption of VMs

VM are processes, hence we can follow their consumption. Example on CloudMIP



Adapting at system level

Adapting at system level: A three step methodology

Definitions:

- Phase: region of execution of the application/system stable with respect to a given metric
- System: computing or storage node

Phase detection

Discover system's runtime execution patterns

Phase characterization

Associate useful information with known execution patterns

Phase identification and system reconfiguration

Reuse of optimal configuration information for recurring phases

Implemented as MREEF Framework.

MREEF in cloud: experimental protocol and set-up

- 8 VMs deployed on a node having 8 CPU cores
- Each VM executes the following workloads 20 times
  - Cloud applications:
    - ★ Transactional database system (sysbench + MySQL)
    - ★ Web application (siege + Apache HTTP server)
    - ★ IO intensive application (IOzone)
  - HPC application: CG
- Three system configurations
  - on demand, powersave, MREEF
  - Workloads are the same for each system configuration (a random execution order is provided to each VM)

MREEF in cloud: results

- MREEF configuration:
  - Majority-rule-based phase characterization for phase characterization
  - EV classification for workload prediction

MREEF reduces the energy consumption of up to 8% with less than 1% performance degradation
Outperforms powersave

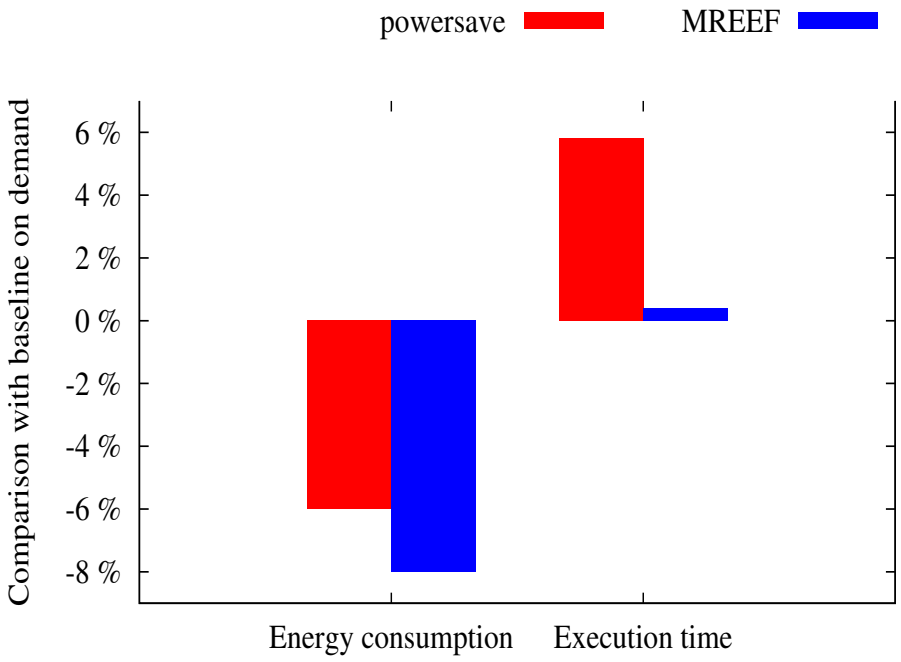
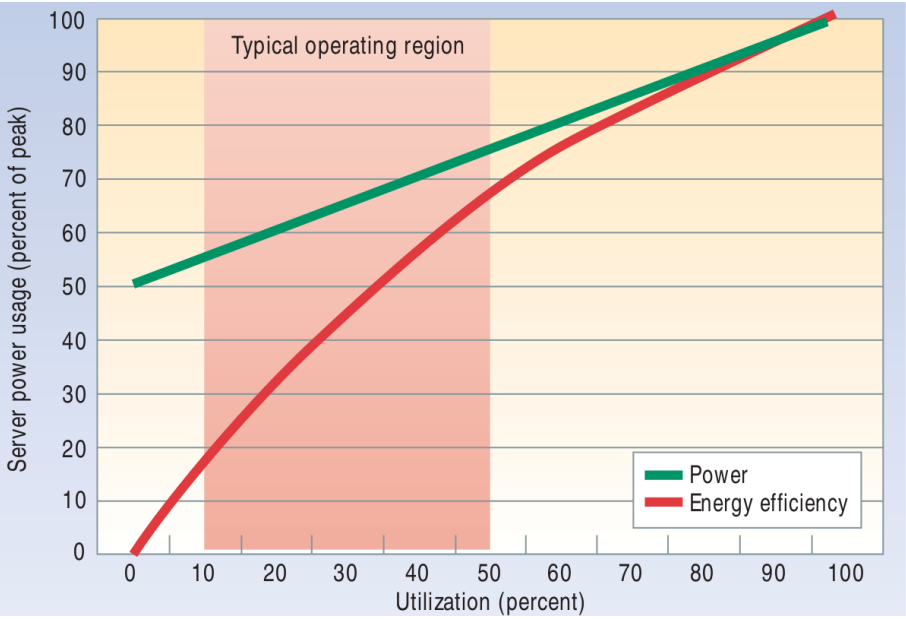


Figure: Baseline on demand versus powersave and MREEF in a cloud environment.

Heterogeneous Computing and Clouds

Needs to achieve Energy Proportionality

Luiz André Barroso and Urs Hölzle, "The case for Energy-Proportional Computing", IEEE Computer, 2007

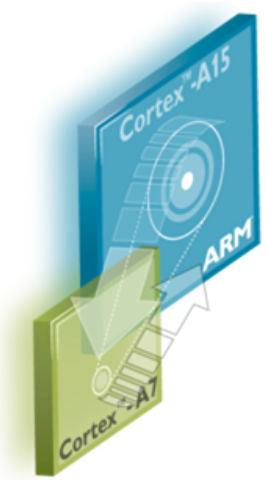


Server power consumption and energy efficiency from 0 to 100% utilization

- **Average load of servers between 10 and 50%**
  Most energy inefficient region
- **High idle consumption**
  Can be up to 50% of the peak power

→ **A perfect proportional curve would bring huge energy savings**
Especially need to reduce energy consumption for low to medium loads

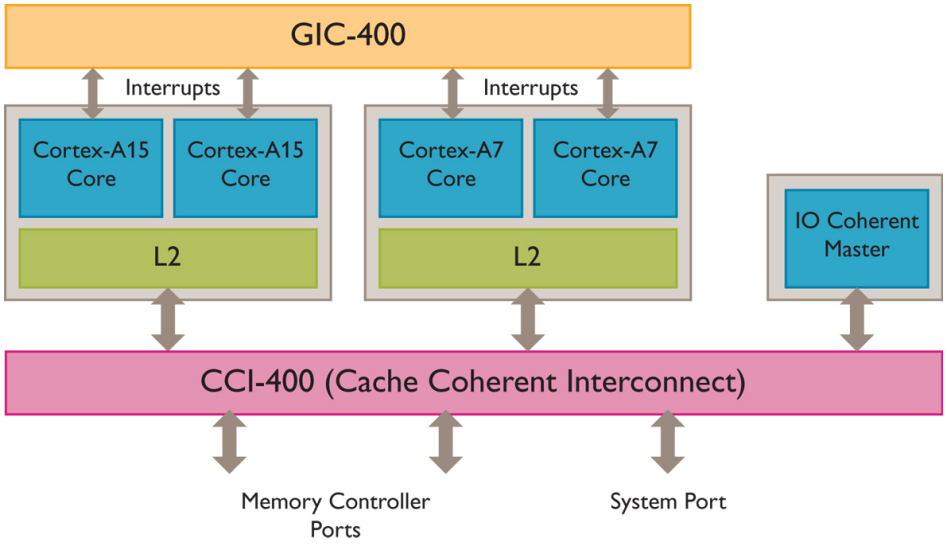
Heterogeneous architectures - Example of ARM big.LITTLE



2 coupled processors :

- big : Cortex-A15
- LITTLE : Cortex-A7

Interconnected by a Cache Coherence system



<u>GOAL</u> : Extend battery life time of mobile devices

⇒ Application of the concept to datacenters :

ARM is not powerful enough for all applications, we need to extend the range of heterogeneous hardware :

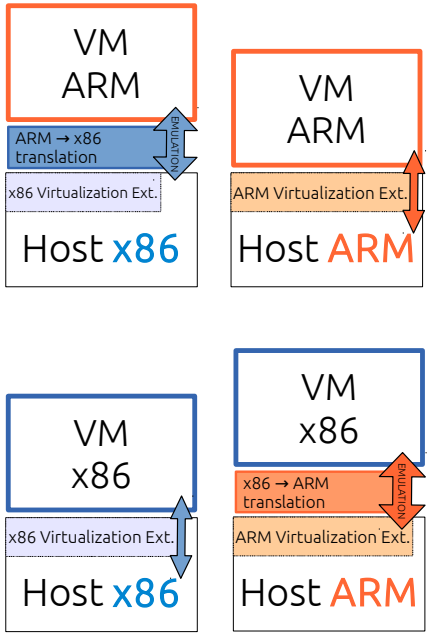
low power processors for low load, regular x86 servers for high performance

First experiments

Summary of selected hardware :

Codename	Chromebook	Taurus	Parapluie
Fullname	Samsung Chromebook	Dell PowerEdge R720	HP Proliant DL165 G7
Architecture	ARMv7 32 bits	x86 64 bits	x86 64 bits
CPU	2 x Cortex-A15	2 x Intel Xeon E5-2630	2 x AMD Opteron 6164
Total cores	2	12	24
Power consumption	5 - 25 W	96 - 227 W	180 - 280 W
Release year	2012	2012	2010

Two alternatives for VM architecture :

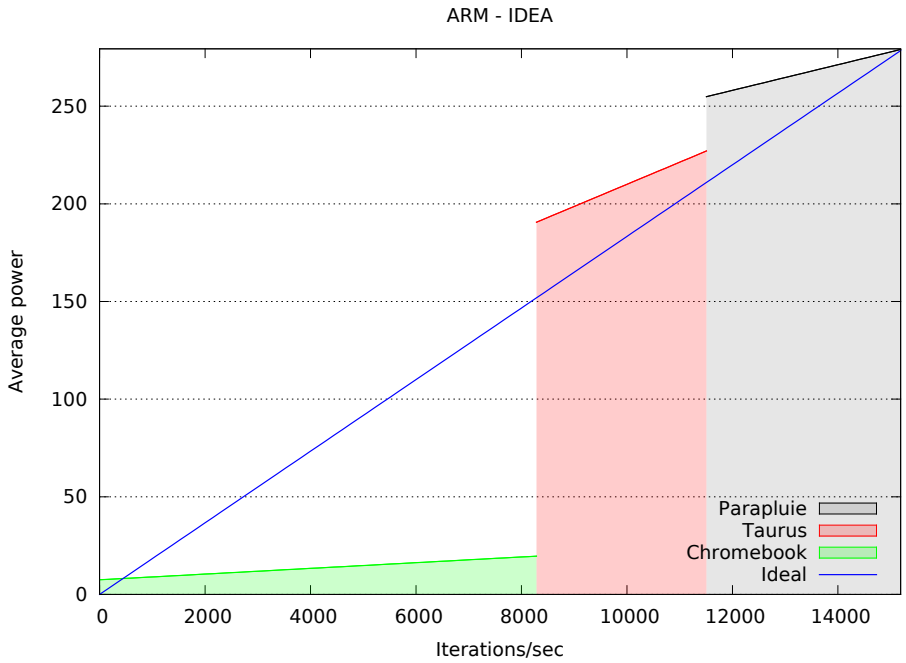


- KVM for virtualization
- QEMU for emulation when VM arch is different from host arch

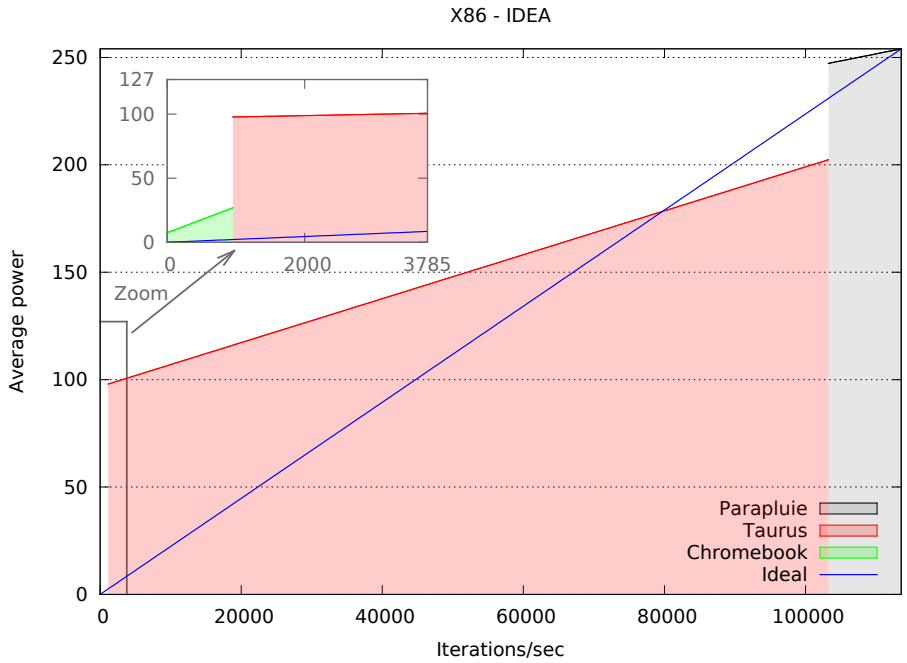
First results and Perspectives

Comparison execution of nbench benchmark for the two VM alternatives :

- ARM VM



- x86 VM



→ Still some work to do to reach perfect energy proportionality

Challenges and Perspectives :

- Effective migrations between heterogeneous architectures
- Other solutions than emulation to benefit from native performances
- Find adaptive solutions to applications types
- Predict applications evolutions to take decisions

Scheduling and Placement

Placement and Scheduling

Placing and Moving applications in the physical infrastructure to optimize energy with losses in QoS.
Questions:

- Where to move / place a VM?
- Which VMs to move?
- When to do it?

Answering the question

- Optimal solutions (using Linear Programming)
- Heuristics

- Centralized algorithm (First-Fit, Vector Packing, Credit-based, Genetic Algorithm)
- Cooperative algorithm

Credit-based Algorithm

Credit-based Algorithm

Algorithm for anti-loadbalancing the load

- The proposed algorithm works by associating a credit value with each node.
- This Credit algorithm is an adaptation of the Comet Algorithm: Each agent is trying to maximize its own credit by moving between nodes

Adaptation to Cloud:

- In our case, each hosts will try to maximize its own credit by offloading VMs to other nodes.

Credit-based Algorithm: Credit calculation

✓ Credit of a node depends on its current load ($c_{i,j}$), its communications behavior and history of task execution.

✓ $\sigma_{i,j} = c_{i,j} - t_{i,j} s_{i,j} + \varepsilon - \gamma$

❑ $s_{i,j} = \frac{c_{i,j}}{r_{i,j}}$, job satisfaction of node j in site i

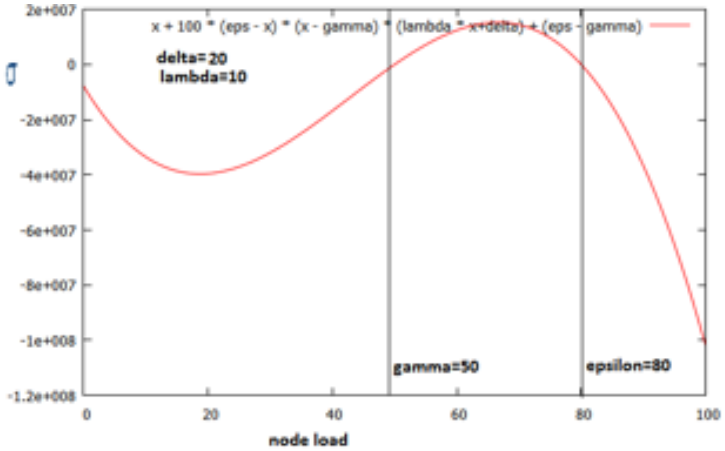
❑ the aggregated demand load of all tasks on node j in site i $r_{i,j} = \sum_{k=1}^{T_i} (l_{i,j,k})$

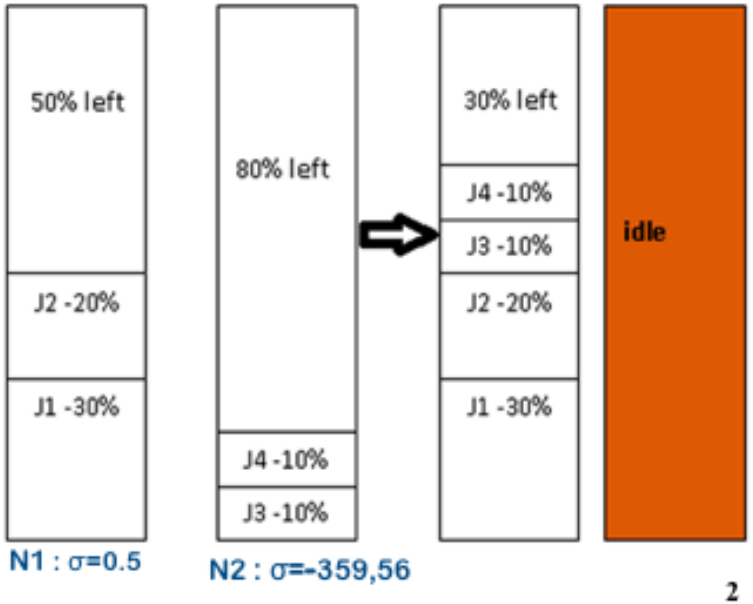
❑ $l_{i,j,k}$ demand load of VM *k in node j in site i*

❑ $t_{i,j} = c_{i,j} + (\varepsilon - c_{i,j})(c_{i,j} - \gamma)(0.1\, c_{i,j}\, 20) / s_{i,j}$

❑ The Credit of a node will be used in the selection policy

$\gamma = 50$ $\varepsilon = 80$

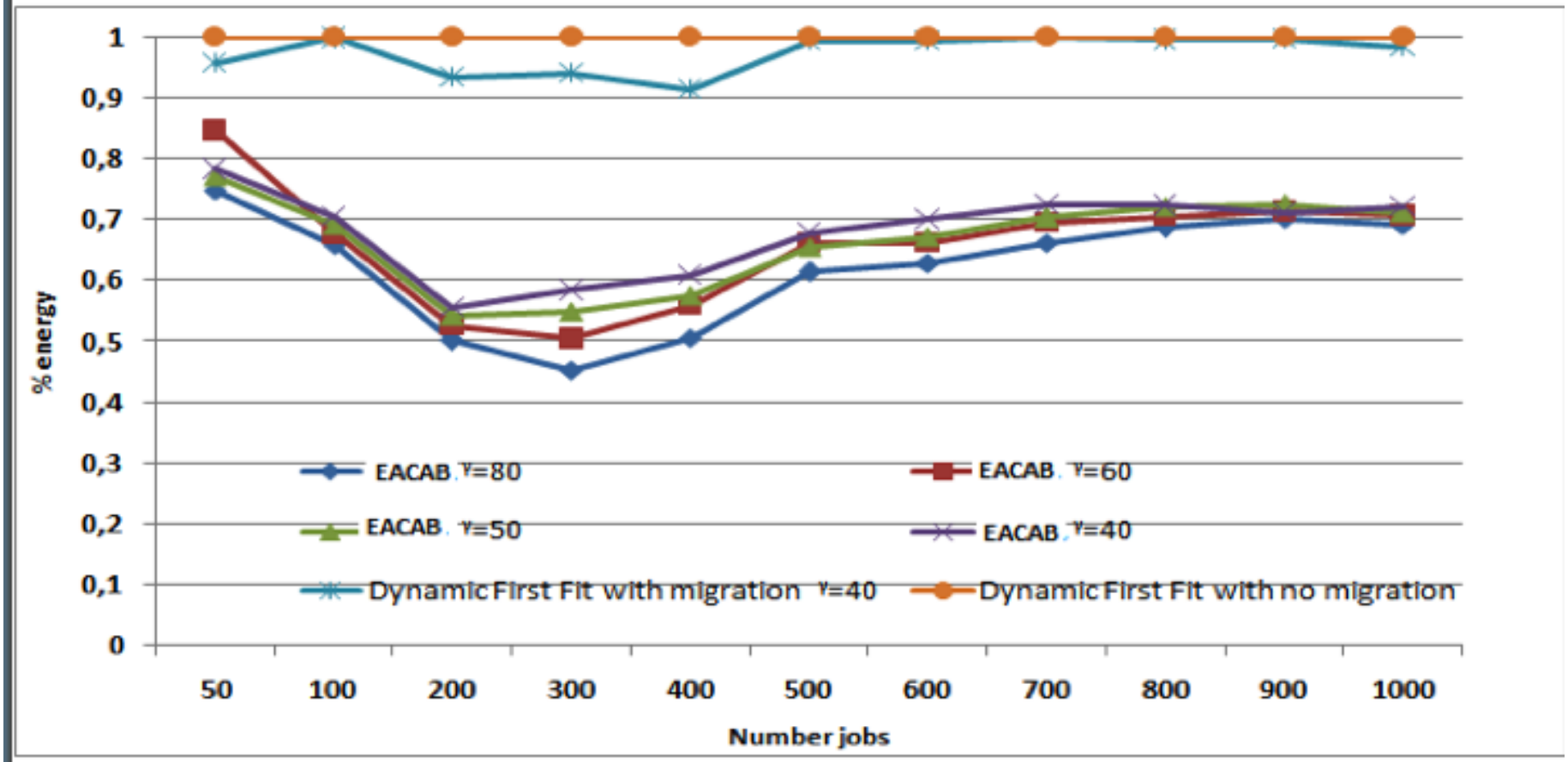




2

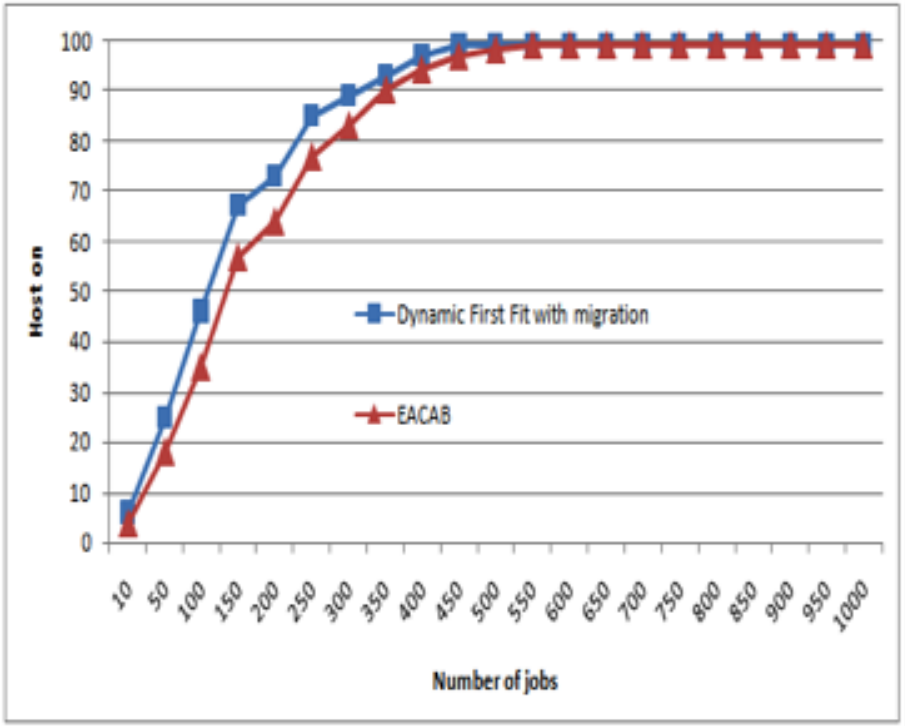
Credit-based Algorithm: Energy Results

Simulation with an extension of CloudSim.

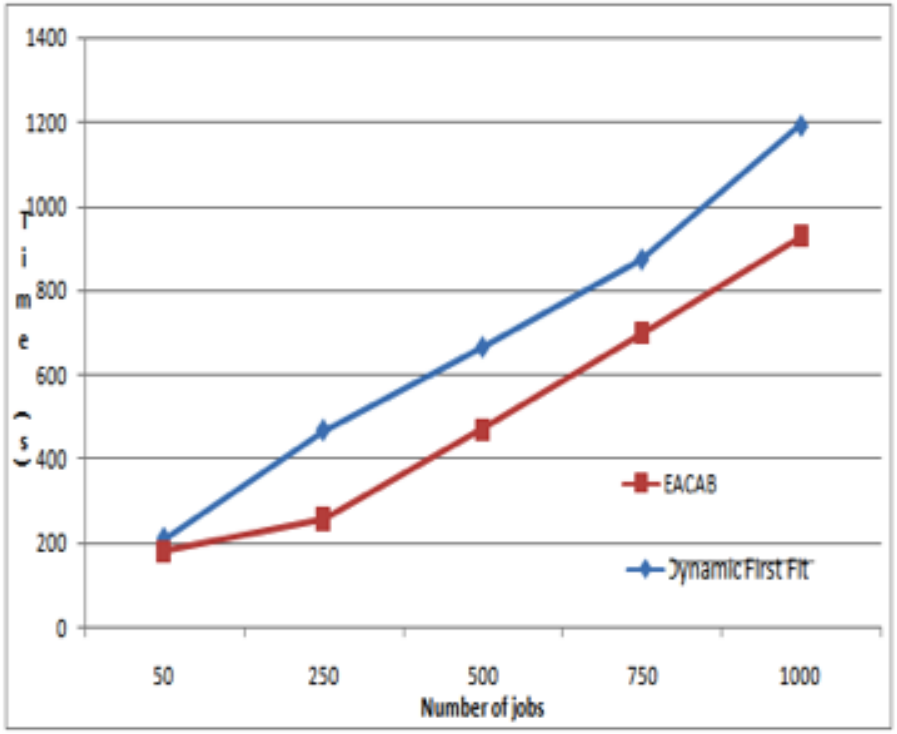


Impact of threshold in energy gain
with the EACAB Lower is better

Credit-based Algorithm: Hosts ON and Makespan



Maximum nodes switched on with EACAB



Makespan

SOP

- SOP: think global Services for persOnal comPuters:
  - aims to make the use of computers as simply as mobile phones
  - builds a hybrid working model
  - takes benefit from a mix between local execution and remote access to software and services located in distant dedicated datacenters or even in other non-used consumers machines
- Partners:
  - Degetel,
  - LAAS,
  - IRIT,
  - Sysfera,
  - QoSDesign
- IRIT is involved:
  - on machine placement: multi-cloud energy-aware scheduling
  - on autonomic manager: graph based approach

SOP VM Scheduling and Hosts Management

- Small experimental cloud with OpenNebula on a RECS platform
- Dynamically manage VMs via consolidation
- Manage hosts to save energy

Host Management Strategies

Powering on and off take time and resource, avoiding unnecessary operations is important.

We used a Pivot based host management strategy:

- Compute expected number H of physical hosts needed
- Add N pivot hosts
- Keep the number of powered on hosts at $H + N$

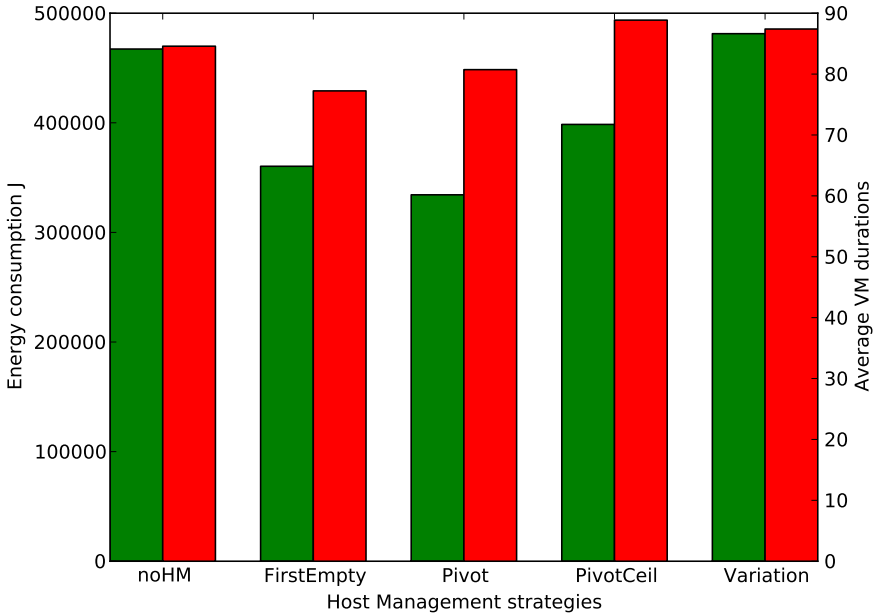


Figure: Green is Energy (J) / Red is effective duration (%)

VM migration overhead

VM Live migration time depends on VM size, profile, and environmental conditions (other VM migrating).

- According to **our infrastructure**, we tried to estimate a number of average VMs it is acceptable to migrate at the same time, on the same link.
- Based on the VM size and host capacity, we can compute x, the number of host the consolidation algorithm will try to offload.

Table: Time spent with concurrent live migrations

Nb. migrations	1	2	3	4	5	6
Avg. Time (s)	5.5	7.44	11.68	12.55	14.05	16.62

SOPVP Approach

- Vector Packing based reallocation algorithm. Aims at reducing resource consumption imbalance on hosts
- Tries to offload x hosts. Enforce migrations only if the host can be emptied
- Comparison between *SOPVP* and OpenNebula's *mm_sched*

Results

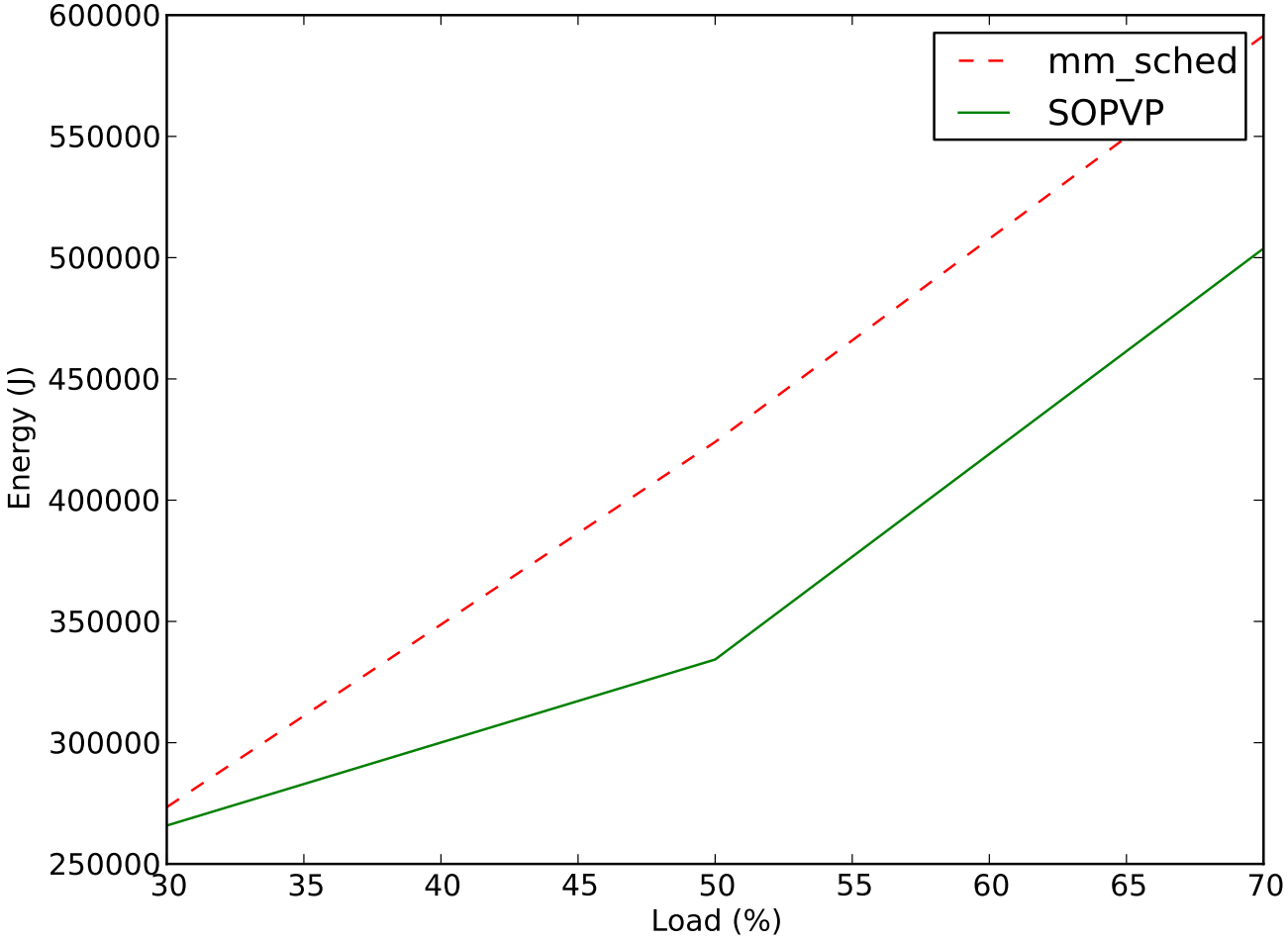


Figure: Energy consumption vs System Load

Results

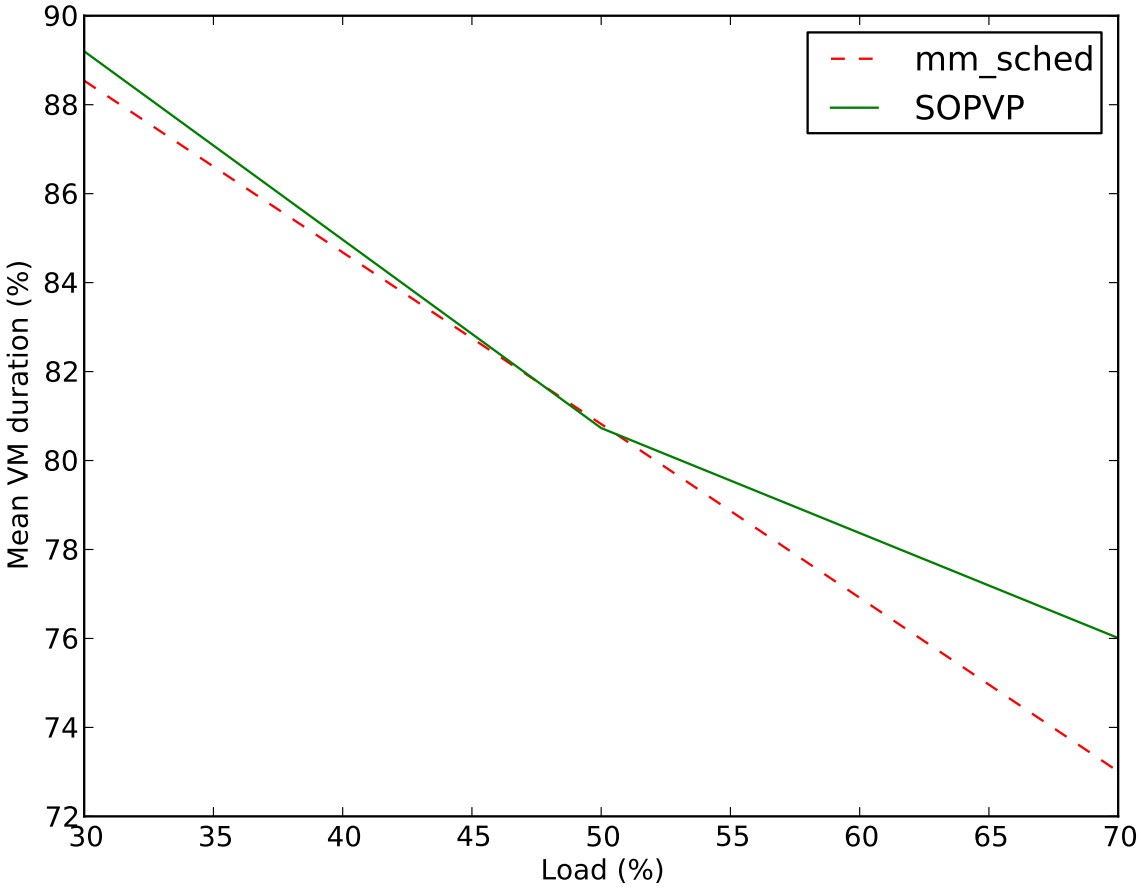


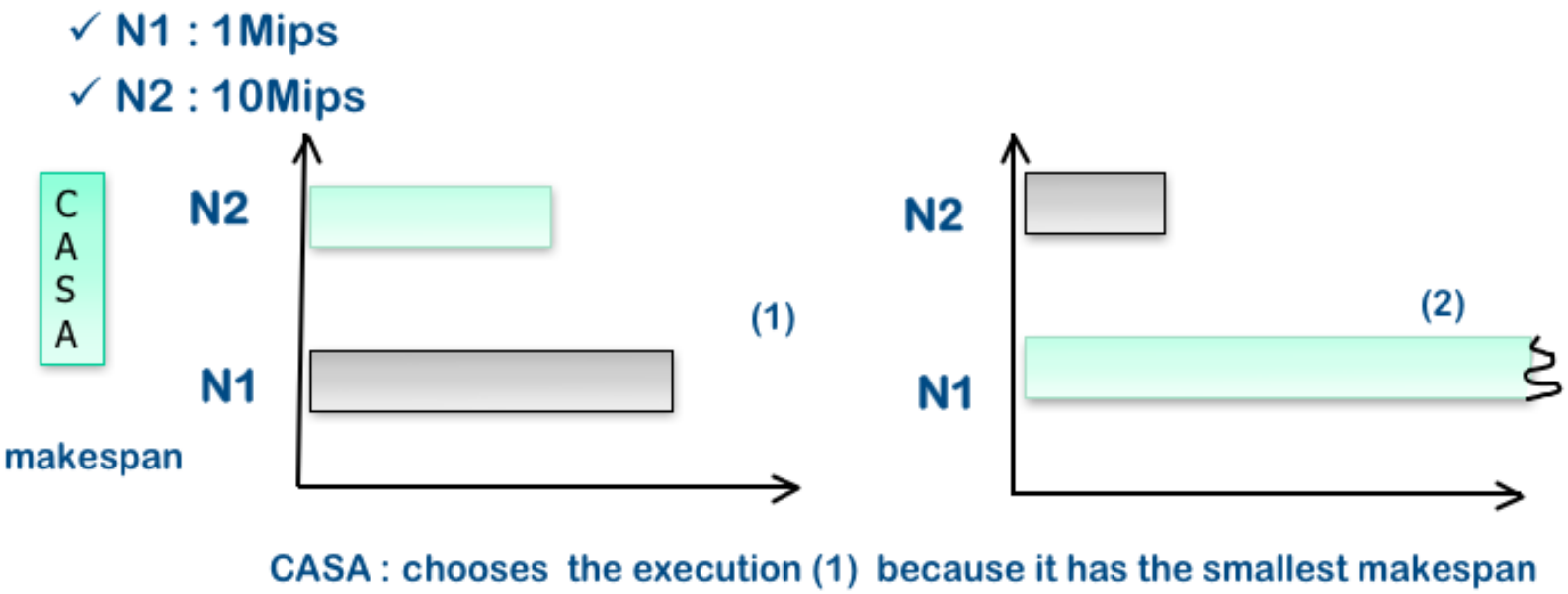
Figure: VM durations[2] vs System Load

[2] Higher is better

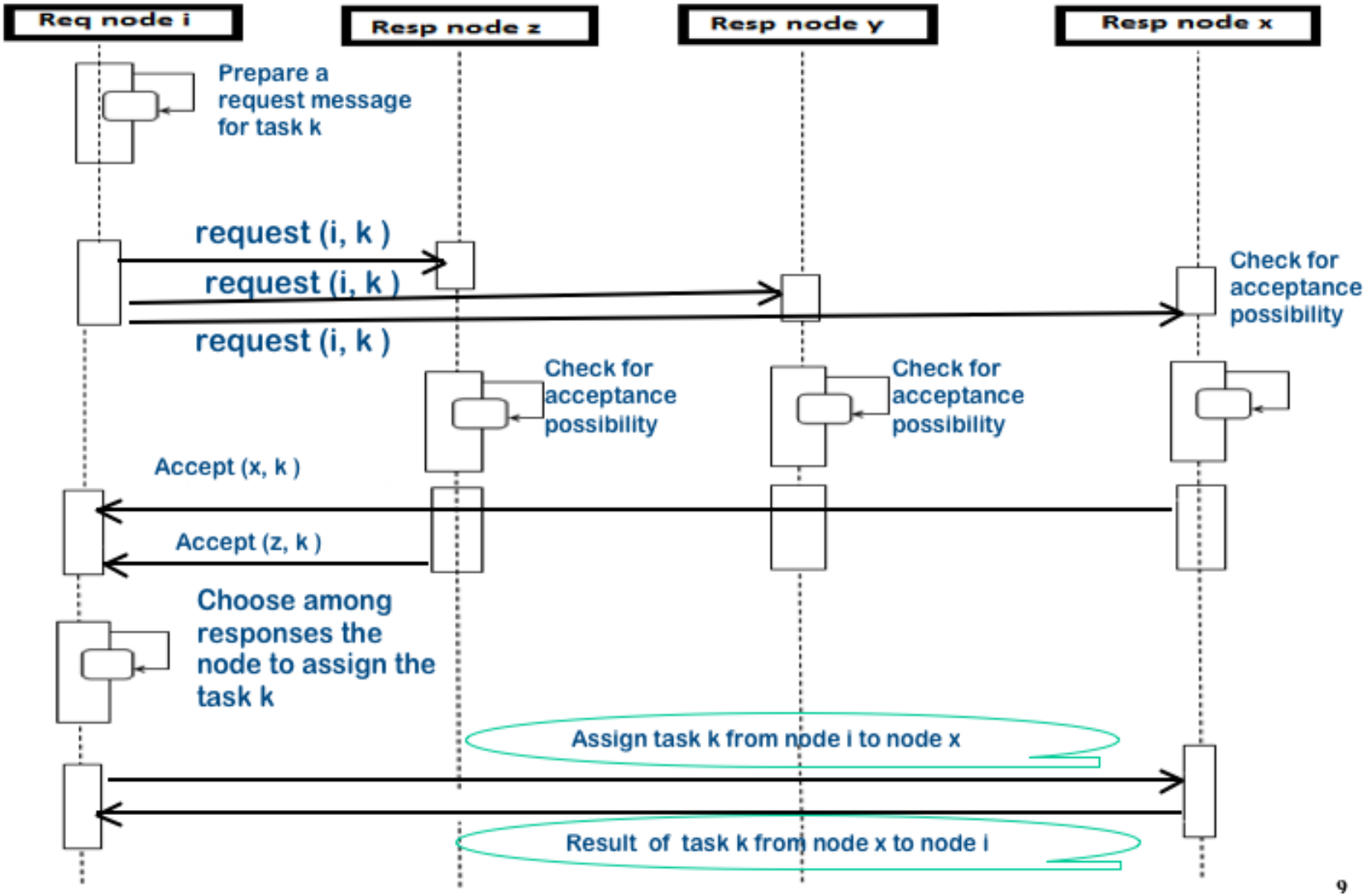
Cooperative Scheduling

Cooperative scheduling Anti-load balancing Algorithm for cloud (CSAAC)

- Based on CASA
  - CASA is a cooperative algorithm which minimize response time
  - A participating node calculates estimated response time if a job is on it
  - Makes job allocation decisions based on contacted nodes real-time responses
  - Allows for rescheduling jobs through time

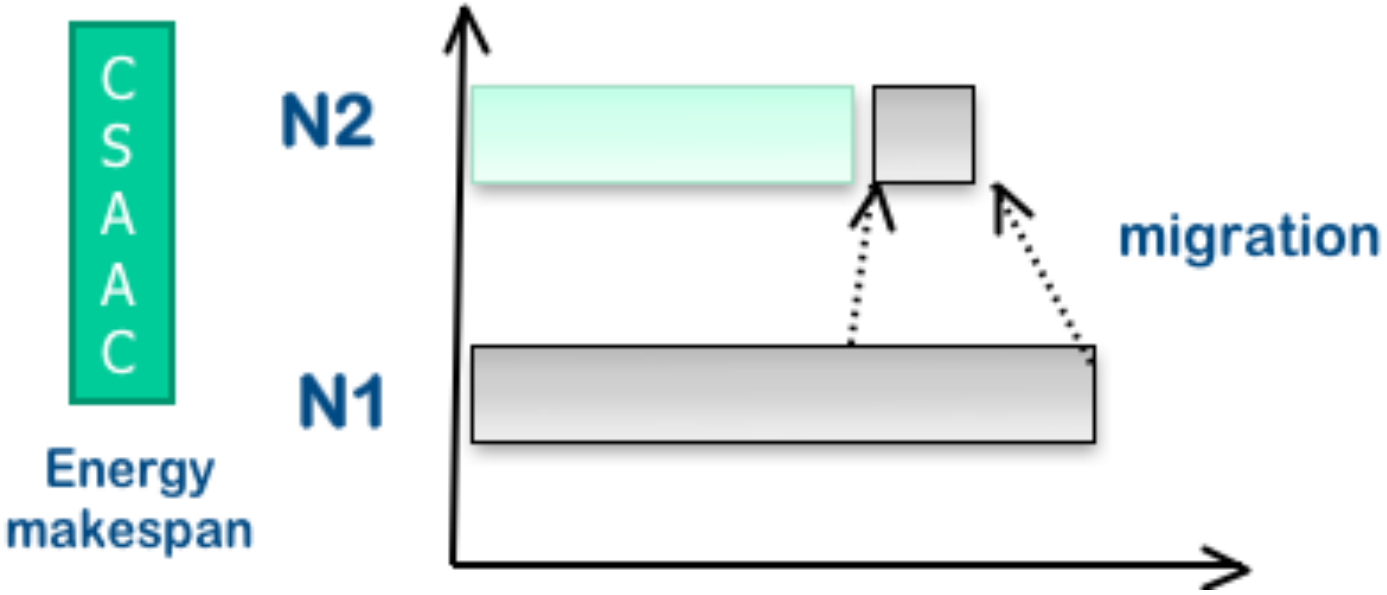


CASA: Global view of algorithm

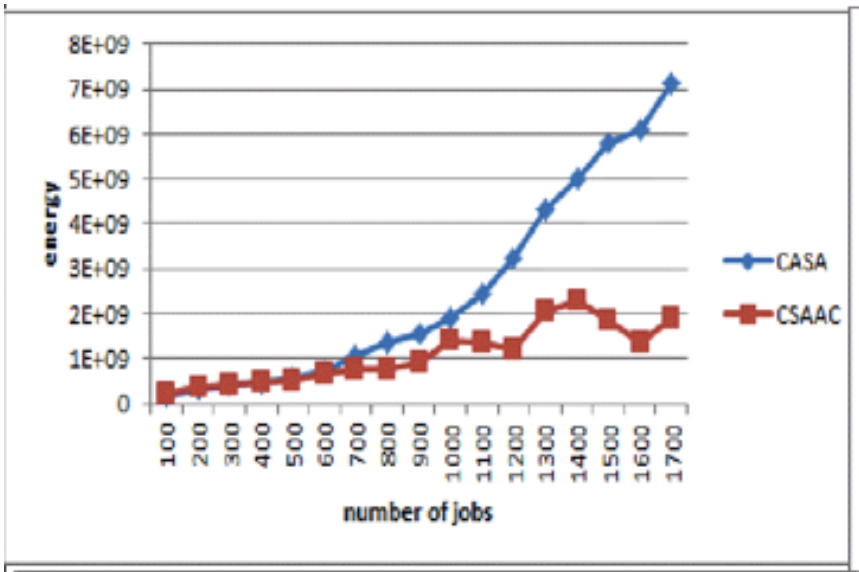


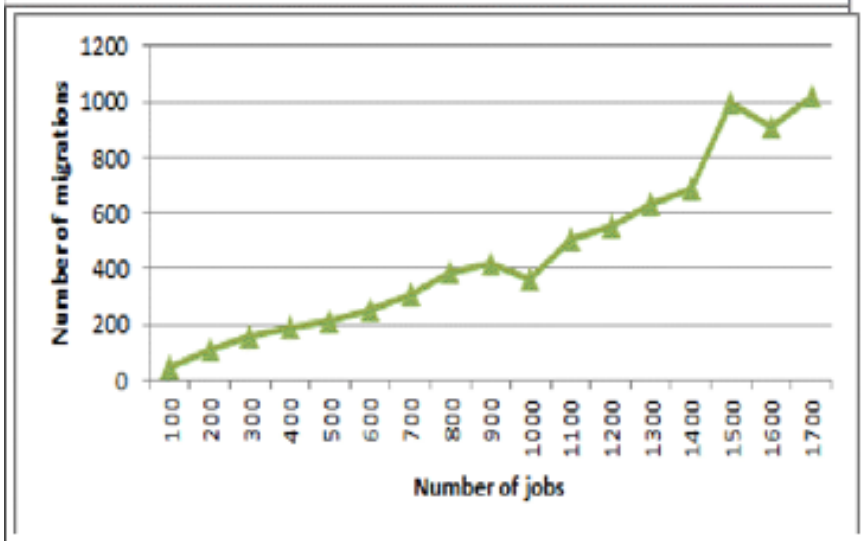
Adaptation to Cloud and Energy: CSAAC

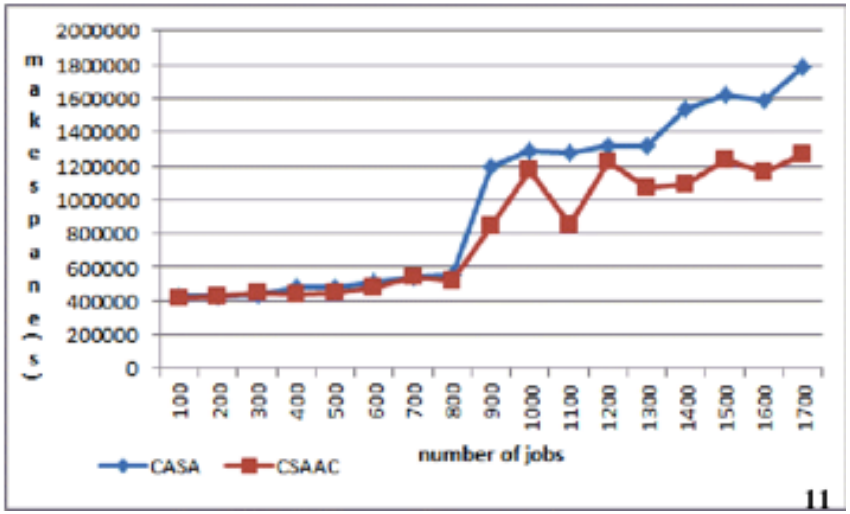
- A participating node calculates estimated response time and energy if a VM is on it
- Makes VM allocation decisions based on contacted nodes estimated energy to finish, then response time
- Allows for VM migration



CSAAC: results







Conclusion

Conclusion

- Energy consumption of VM: measuring at platform level (and where even at platform level) or inside VM?
- Acting to reduce energy: at system level, at hypervisor level, at middleware level, or at application level?
- Centralized decision centers vs decentralized ones?

Credits:

- Platforms: Francois Thiebolt
- Energy Consumption Modeling: Leandro Fontoura Cupertino, Georges Da Costa, Amal Sayah
- System Adaptation: Landry Ghislain Tsfack Chetsa, Georges Da Costa, Laurent Lefevre, Patricia Stolf
- Power proportionality: Violaine Villebonnet, Georges Da Costa, Laurent Lefevre, Patricia Stolf
- Scheduling, Placement: Damien Borgetto, Thiam Chekhou, Georges Da Costa, Patricia Stolf

Questions?

Jean-Marc.Pierson@irit.fr
www.irit.fr/~Jean-Marc Pierson