Analysis of a Self-Organizing Algorithm for Energy Saving in Data Centers

Carlo Mastroianni, Giuseppe Papuzzo

Institute for High Performance Computing and Networks, Italy

Spin-off from Italian CNR
http://www.eco4cloud.com

Michela Meo

Politecnico di Torino, Italy

Cloud and data centers

- Clouds are hosted on data centers
- Size ranges from tens to tens of thousands of physical servers
- Inefficiencies cause:
  - ✓ high electricity costs (also for cooling)
  - ✓ huge carbon emissions
  - ✓ server overload and low QoS
- Data center efficiency is a huge issue!

Facebook data center in Sweden

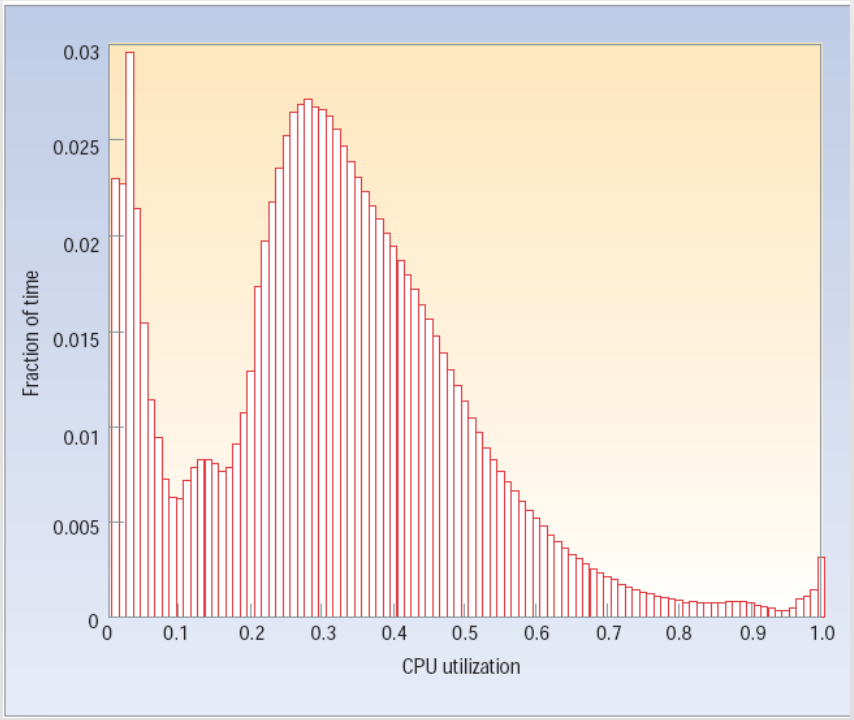
Inefficiency of servers

Two sources of inefficiency

1. **On average only 30% of server capacity is exploited**
2. **Active but low-utilized servers consume more than 50% of the energy consumed when fully utilized**

This means that it's generally possible to consolidate the load on fewer and better utilized servers!

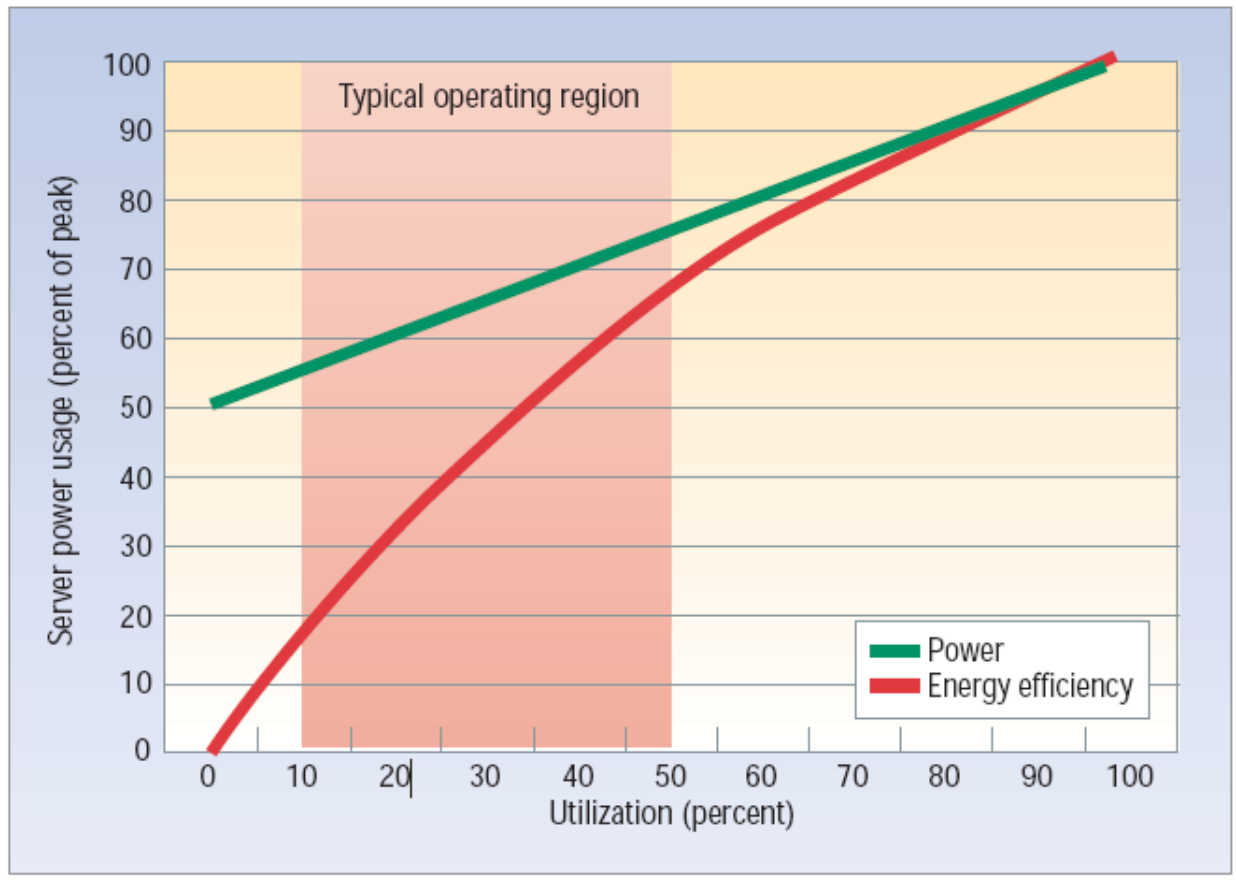
Typical utilization of servers



most servers are in 20% to 40% region of CPU utilization

Source: L.Barroso, U.Holzle, The case of energy proportional computing, ACM Computer Journal, Volume 40 Issue 12, December 2007.

Typical energy efficiency behavior



Power consumption is 50% or more when server is idle

Energy efficiency is utilization divided by power consumption

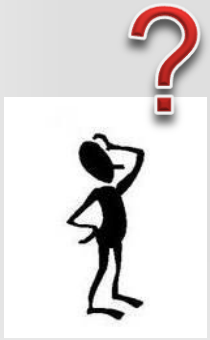
Energy efficiency is <u>low</u> in the typical operating region

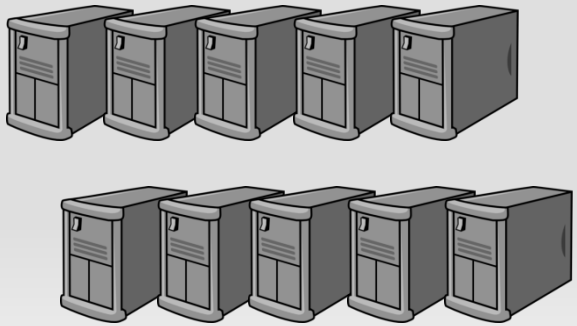
Current solutions for data centers

- More efficient cooling
  - this helps to improve the PUE index (Power Usage Effectiveness), not to increase computational efficiency
- Adopt "energy-efficient" servers
  - e.g., voltage and frequency scaling
  - good for CPU, partially for RAM, not for other components
  - several steps ahead in this direction, but now progress is slower
- Consolidate VMs on fewer servers
  - unneeded servers can be hibernated or used to accommodate more load

Consolidation of VMs in data centers

- Assign VMs on the smallest number of servers, so as to hibernate the remaining servers, and save energy
- An NP-hard problem (online bin packing problem)
- Solutions available today are often complex, not scalable and may require a massive reassignment of VMs





Known solutions for consolidation

- **Best Fit: each VM is assigned to the server whose load is the closest to a target (e.g. 90%)**

  This only guarantees a performance ratio of 17/10: at most 17 servers are used when the minimum is 10

- **Best Fit Decreasing: VMs are sorted in decreasing order, then assigned with Best Fit**

  Performance ratio is 11/9, but sorting VMs may not be easy in large data centers, and many concurrent migrations are needed

- **DPM of VMWare adopts a greedy algorithm**

  Servers are sorted according to numerous parameters (capacity, power consumption, etc.). DPM scans the list and checks if servers can be unloaded and hibernated

eco4cloud solution

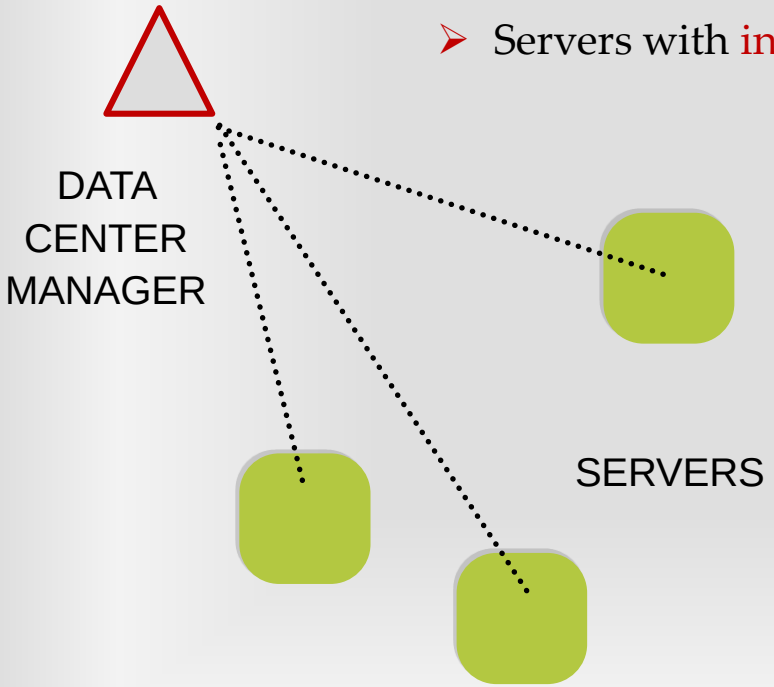
www.eco4cloud.com

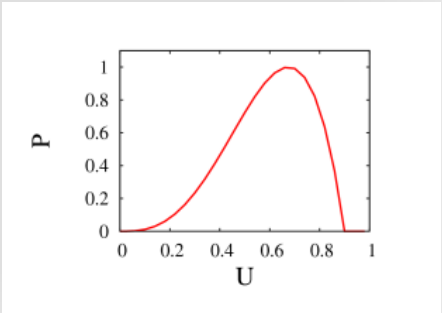
- C. Mastroianni, M. Meo, G. Papuzzo, "Self-Economy in Cloud Data Centers: Statistical Assignment and Migration of Virtual Machines", Euro-Par 2011, September 2011.
- C. Mastroianni, M. Meo, G. Papuzzo, "Analysis of a Self-Organizing Algorithm for Energy Saving in data Centers", HPPAC 2013, May 2013.
- PCT Patent "SYSTEM FOR ENERGY SAVING IN COMPANY DATA CENTERS"

eco4cloud in a nutshell

The data center manager assigns and migrates VMs to servers based on local probabilistic trials:

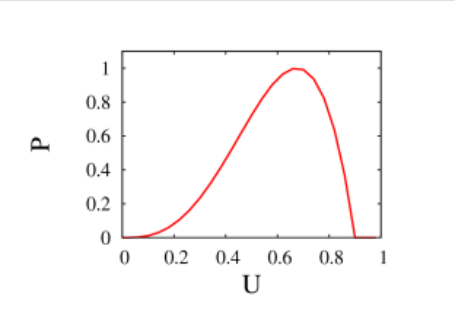
- Lightly loaded servers tend to reject VMs
- Highly loaded servers tend to reject VMs
- Servers with intermediate load tend to accept VMs





VM assignment procedure

1. The manager sends an invitation to a subset of servers
2. Each server evaluates the assignment probability function based on the utilization of local resources (e.g. CPU, RAM…)
3. The server performs a Bernoulli trial to decide whether or not to be available: if available, the server sends a positive ack to the manager
4. The data center manager collects the positive replies and selects the server that will execute the VM



assignment probability function

- The assignment probability is a function of the CPU utilization **u** (with values between **0** and **1**) and of the threshold **Ta**, defined as the maximum allowed utilization (e.g., **Ta = 0.9**)

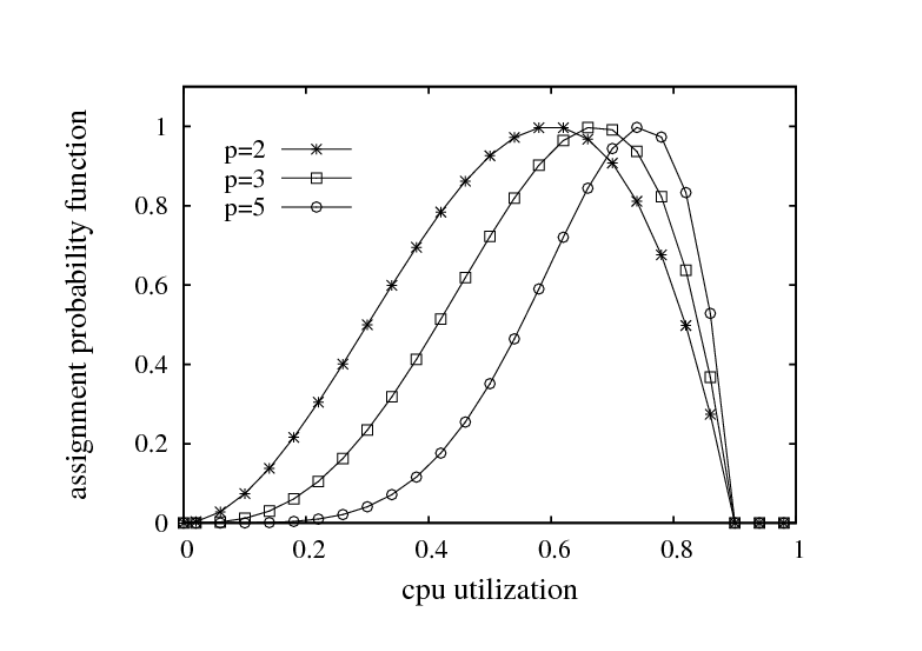
$$f_{assign}(u) = 1/M_p \cdot u^p \cdot (T_a - u)$$

The factor Mp is used to set the maximum value to 1.

- The function assumes a value between **0** and **1**, which is used as the success probability of the Bernoulli trial

assignment probability function

- The graph shows that servers with medium or moderately high load are more likely to accept new VMs
- The parameter **p** can be used to modulate the function shape: the function reaches its maximum value (**=1**) when **u=p/(p+1)*Ta**



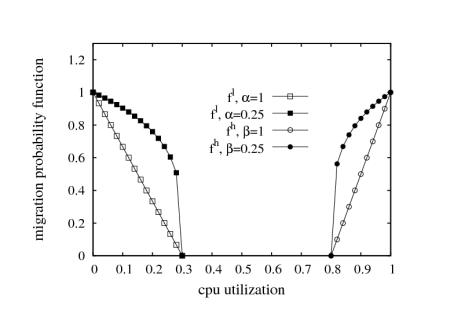
VM migration procedure

1. A server checks if its load is in the range between a low and a high threshold

2. When the utilization is too low, the server should try to get rid of the running VMs. When the utilization is too high, an overload event may occur in a near future

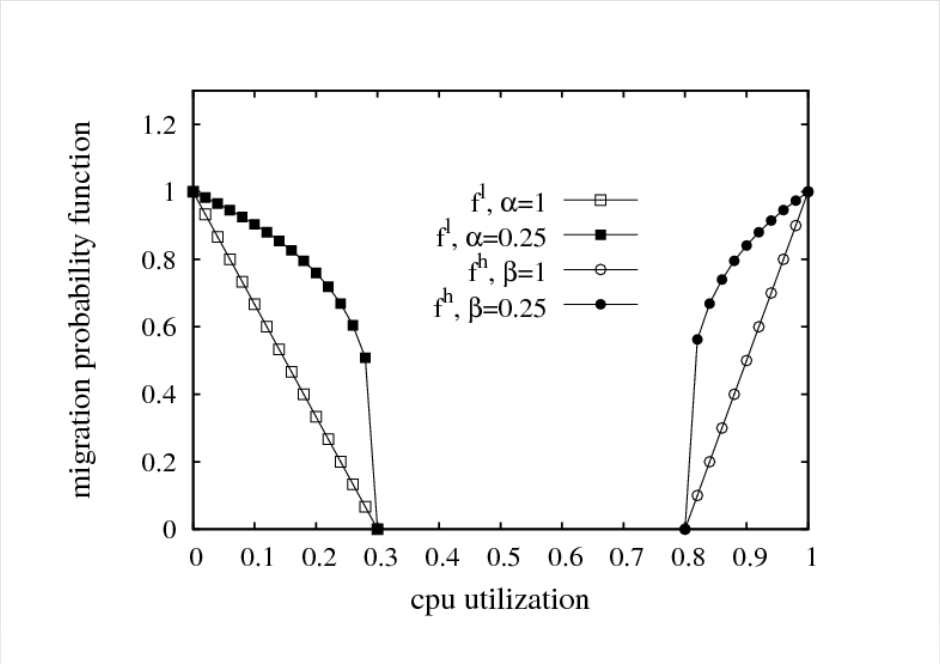
3. In these two cases, the server performs a Bernoulli trial based on the migration probability function

4. If the trial is positive, one or more VMs are migrated



migration probability function

- The function is not null only when **u** < **Tl** (under-utilization) and when **u** > **Th** (over-utilization)
- The function shape can be tuned using parameters **α** and **β**

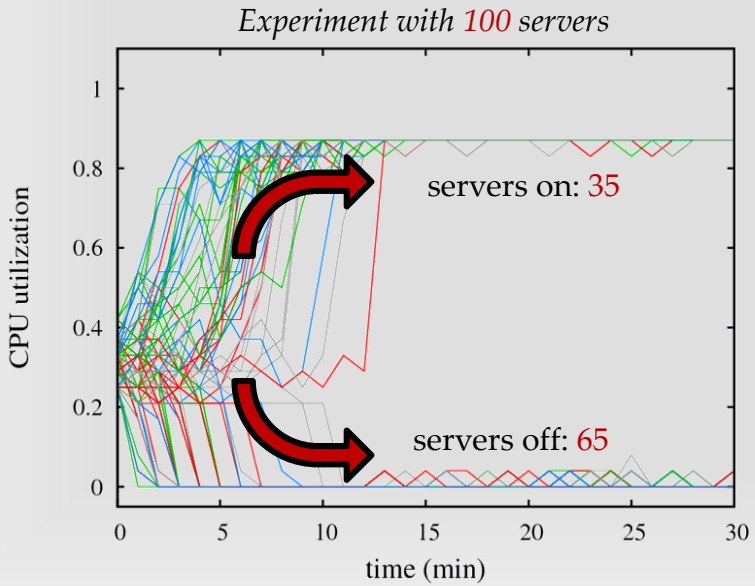


main features of eco4cloud

1. No complex deterministic algorithm: decisions are based on local information
2. Scalable behavior, thanks to the probabilistic and self-organizing approach
3. Migrations are gradual and asynchronous
4. Overload events are prevented with timely migrations
5. Same algorithm and software for all virtualization environments: VMWare, HyperV, KVM

consolidation with eco4cloud

- Servers are not allowed to stay in a low utilization range
- They either get hibernated or are utilized efficiently



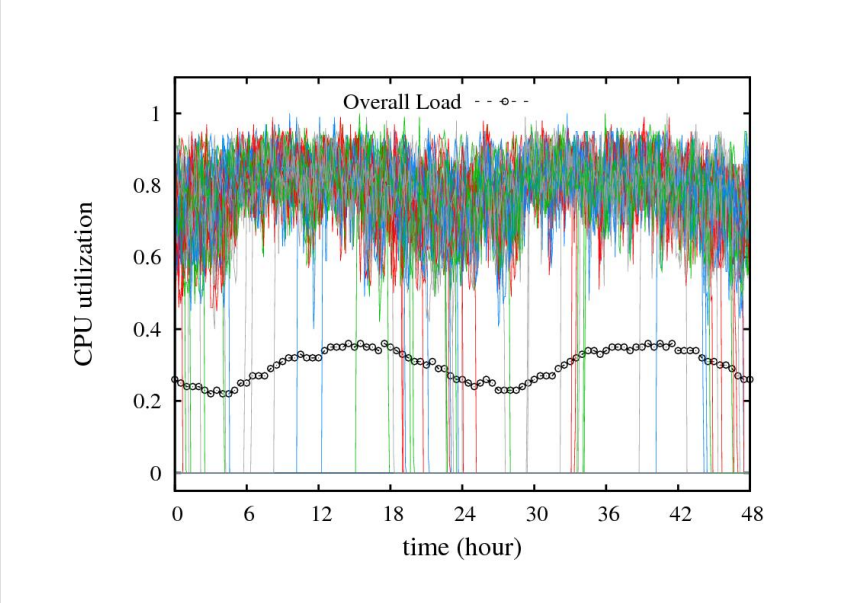
With eco4cloud:

35 servers take all the load

65 servers are hibernated

test with 400 servers and 6000 VMs

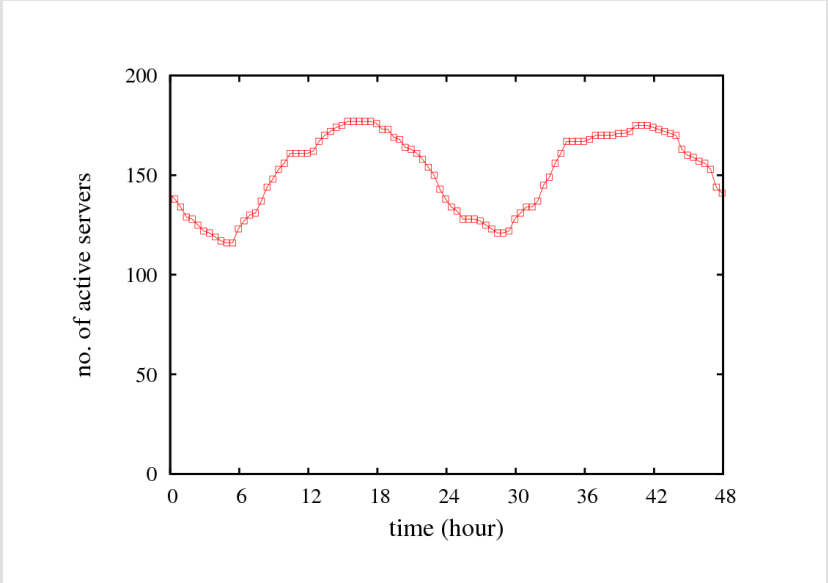
CPU utilization of the servers in a 48-hours interval
(*overall load shown as a reference*)



- CPU utilization of active servers is always between 0.5 and 0.9
- Many servers are hibernated
- Vertical lines correspond to server switches, in ascending and descending phases of the workload

test with 400 servers and 6000 VMs

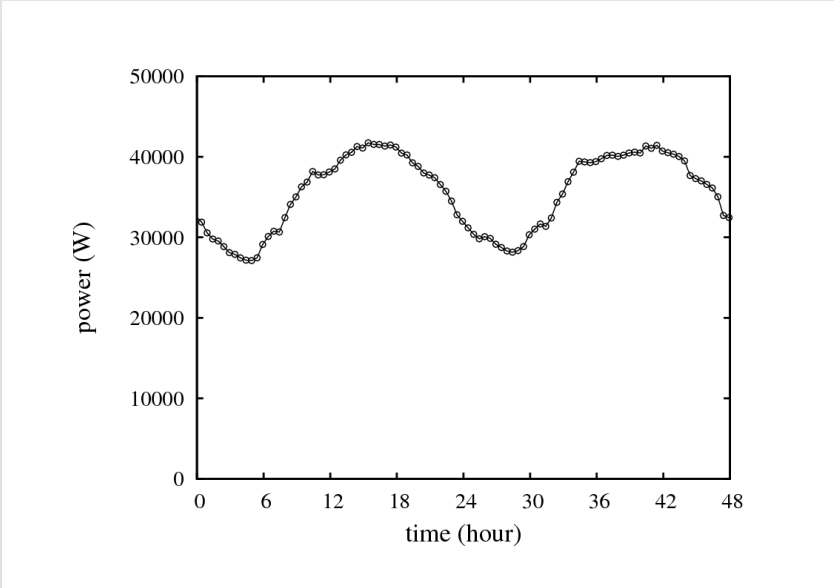
Number of active servers



- Servers are used efficiently, so only a fraction of them are needed
- The number of active servers follows the overall workload
- Many servers are never activated: they can be safely devoted to other applications

test with 400 servers and 6000 VMs

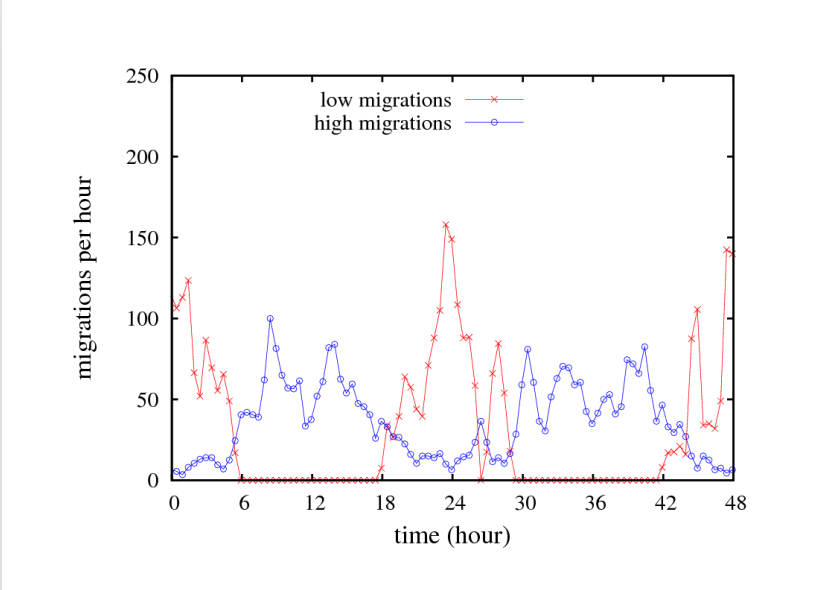
Consumed power



- The consumed power follows the workload
- More savings are obtained thanks to decreased cooling needs

test with 400 servers and 6000 VMs

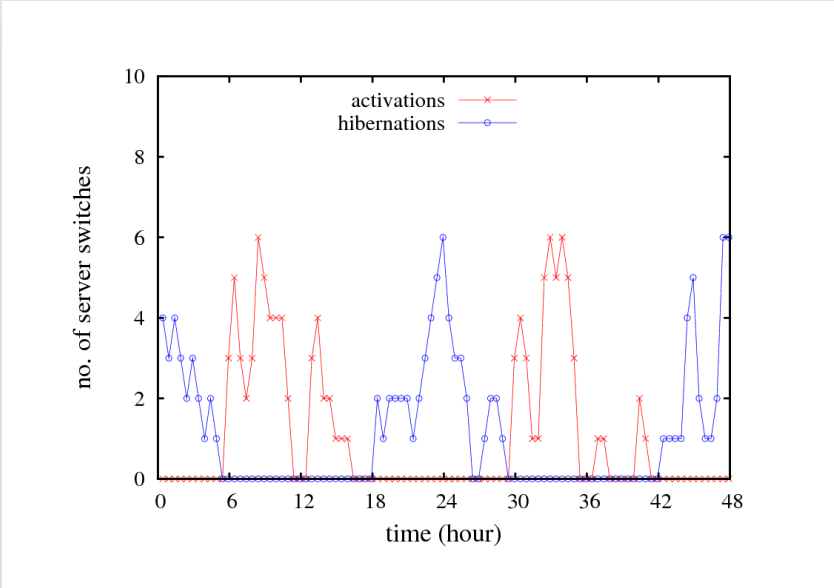
Frequency of migrations



- "High migrations" when the load increases, "low migrations" in descending phases
- Less than one migration every 4 days per VM
- Migrations are asynchronous, with other algorithms they are often simultaneous

test with 400 servers and 6000 VMs

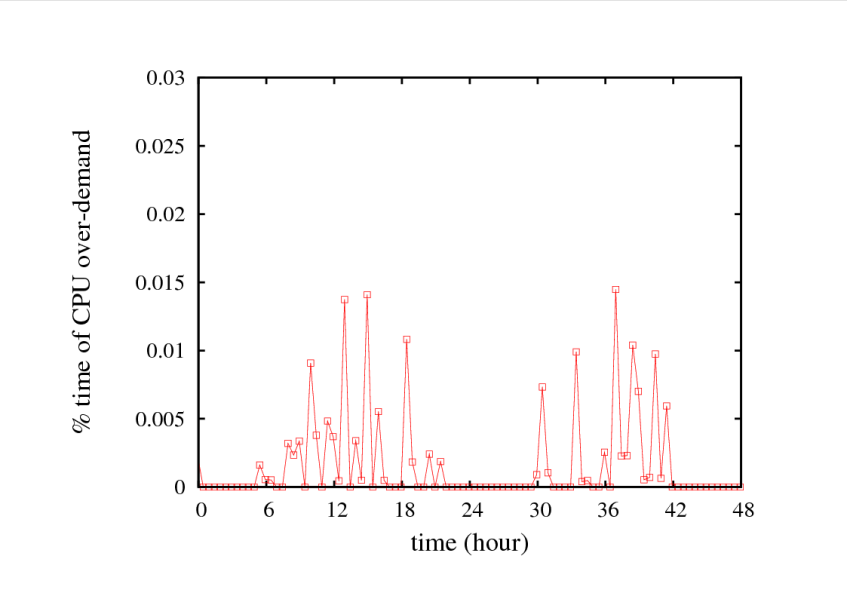
Frequency of server switches



- Some activations when the load increases, some hibernations when the load decreases
- Mechanisms are used to prevent consecutive on/off switches of the same server

test with 400 servers and 6000 VMs

Fraction of time of CPU over-demand



- Time in which the VMs running on a server demand more CPU than the server capacity
- Always lower than 0.02% → high quality of service
- High migrations are triggered when the load exceeds the high threshold

Mathematical Analysis

The assignment process (no migrations) can be modeled with fluid-like differential equations:

$$\frac{\partial u_s(t)}{\partial t} = -\mu(t)u_s(t) + \lambda(t)A_s(t)$$

$$s = 0, \cdots, N_s - 1$$

- **$u_s(t)$** is the CPU utilization of server **s**
- **$\lambda(t)$** is the rate of VM arrivals in the entire data center
- **$\mu(t)$** is the service rate at each server
- **$A_s(t)$** is portion of VMs that are assigned to server **s** (to be computed, depends on **f_a**)

The exact computation of **$A_s(t)$** is costly, but the model can be simplified

Mathematical Analysis (simplified)

The portion of VMs assigned to **s** - **$A_s(t)$** - is assumed to be proportional to the assignment probability evaluated on **s** - **$f_a(u_s(t))$**

$$\frac{\partial u_s(t)}{\partial t} = -\mu(t)u_s(t) + \lambda(t)\frac{f_a(u_s(t))}{\sum_{i=0}^{N_s-1}(f_a(u_i(t)))}$$

$$s = 0, \cdots, N_s - 1$$

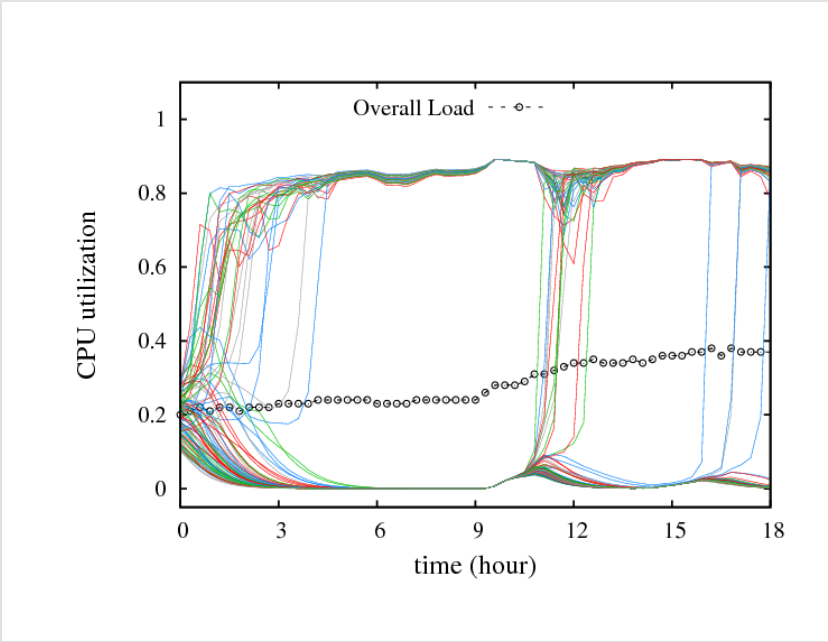
The rate of incoming VMs is normalized

The equations are useful to:

- better understand the system dynamics
- do parameter sweep analysis
- validate results obtained with simulations and real testbeds

Analytical results

CPU utilization of 100 servers
Values of λ(t) and μ(t) are taken from real traces



- Initial conditions: utilization between 20% and 40% for all the 100 servers
- 43 servers take all the load, 57 are hibernated

Conclusions

- Eco4Cloud: a new method for workload consolidation on data centers
- Founded on distribution of the intelligence (to single servers), probabilistic trials, self-organization
- Scalable, adaptive, hypervisor-agnostic

Future work

- Extension of the algorithm to consider more parameters (CPU, RAM, bandwidth)
- Extension of the analytical model to capture VM migrations

THANK YOU

Carlo Mastroianni

ICAR-CNR & eco4cloud srl
Rende (CS) Italy
--
www.eco4cloud.com
mastroianni@eco4cloud.com
fb: www.facebook.com/eco4cloud