Power Efficient Resources Consolidation in Virtualized Environments

Antonio Marotta,

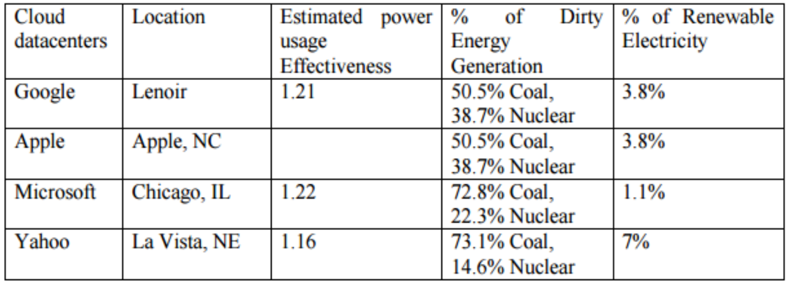
PhD, Computer Science

COMPUTER SCIENCE
DATAVETENSKAP

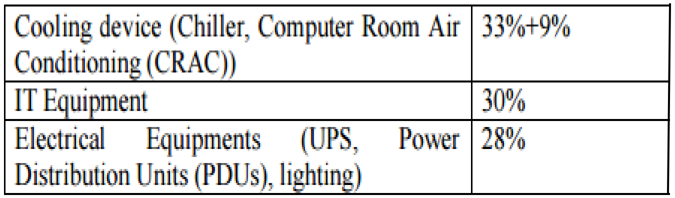
Karlstad, 17th February 2016

Energy Efficiency

- Cloud Computing and energy-efficiency
  - **The cost-efficiency of a cloud provider depends on its ability to over-subscribe capacity by leveraging the smoothing effect without degrading the quality of service**
  - Objectives:
    1. *Increase the cost-efficiency of the underlying infrastructure*
    2. *Find the right compromise between the energy consumption and the perceived QoS of the applications*
- Energy consumption is a double-faced issue: from one hand, there is the economic concern and, on the other, the environmental one



Cloud datacenters	Location	Estimated power usage Effectiveness	% of Dirty Energy Generation	% of Renewable Electricity
Google	Lenoir	1.21	50.5% Coal, 38.7% Nuclear	3.8%
Apple	Apple, NC		50.5% Coal, 38.7% Nuclear	3.8%
Microsoft	Chicago, IL	1.22	72.8% Coal, 22.3% Nuclear	1.1%
Yahoo	La Vista, NE	1.16	73.1% Coal, 14.6% Nuclear	7%



Cooling device (Chiller, Computer Room Air Conditioning (CRAC))	33%+9%
IT Equipment	30%
Electrical Equipments (UPS, Power Distribution Units (PDUs), lighting)	28%

Techniques for Energy Efficient

- ***Energy Efficient*** techniques have been adopted in relation to different scopes:
  - ***Hardware**:* for instance the Dynamic Voltage Frequency Scaling, offered by all modern CPUs
  - ***Application***: task scheduling problems are faced with algorithms which aim at achieving energy reduction by examining the workload characteristics
  - ***Network***: new techniques are implemented in the networking devices in order to optimize consumption, by guarateeing a dependable communication and energy savings

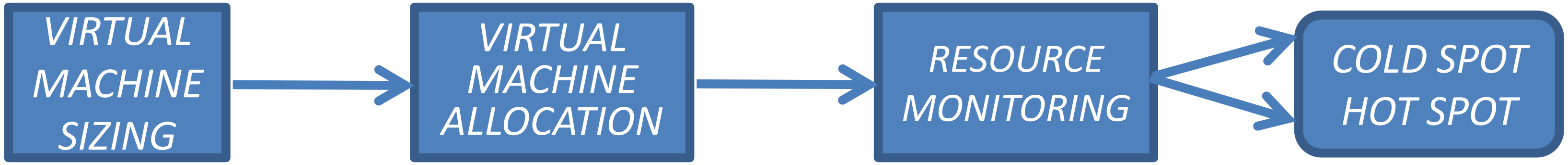
- The literature stresses out that the principal sources of energy consumption are servers and storage, which account for 50% of the total:
  - *This is the reason why, with the advent of virtualization and Cloud Computing, one of the main goals to pursue has become the assurance of a high resource utilization*

VMs Allocation and Consolidation

- New cloud-based techniques have been introduced to save energy and to improve resources utilization

- One of the most famous problem which has been massively studied from the energy efficient view-point is the **VMs Allocation**

- It derives from a classical problem,the ***"Bin Packing Problem"***
    - ***Multi-dimensional, Multiple choice, Multiple constraints***



- *The allocation can be not very efficient from the power consumption view-point at any time*
- The ***VM Live Migration*** enables a new version of the problem, namely the ***VMs Consolidation***

A Novel VMs Consolidation Problem

- The classical ***VMs Consolidation Problem*** was extended and a novel version was proposed

- The new problem is defined as ***Power Efficient VMs Consolidation***
  - Considered
    - *a set of m VMs*
    - *the current allocation of each VM to a physical node*
    - *the set of active nodes*
    - *the amount of each resource available at each server*
    - *the amount of resources requested by the VMs*
    - *the power consumption model of the servers*

the objective is to find the set of migrations defining a new allocation scheme that minimizes the linear combination of the overall power consumption of the servers and the number of migrations according to proper weights

$$f = \alpha \cdot \frac{\sum_{j \in J_{act}^{N}} P_{act}^{N}}{\sum_{j \in J_{act}^{o}} P_{act}^{o}} + \beta \cdot \frac{|Mig|}{m}$$

The Model's Parameters

Input Parameters

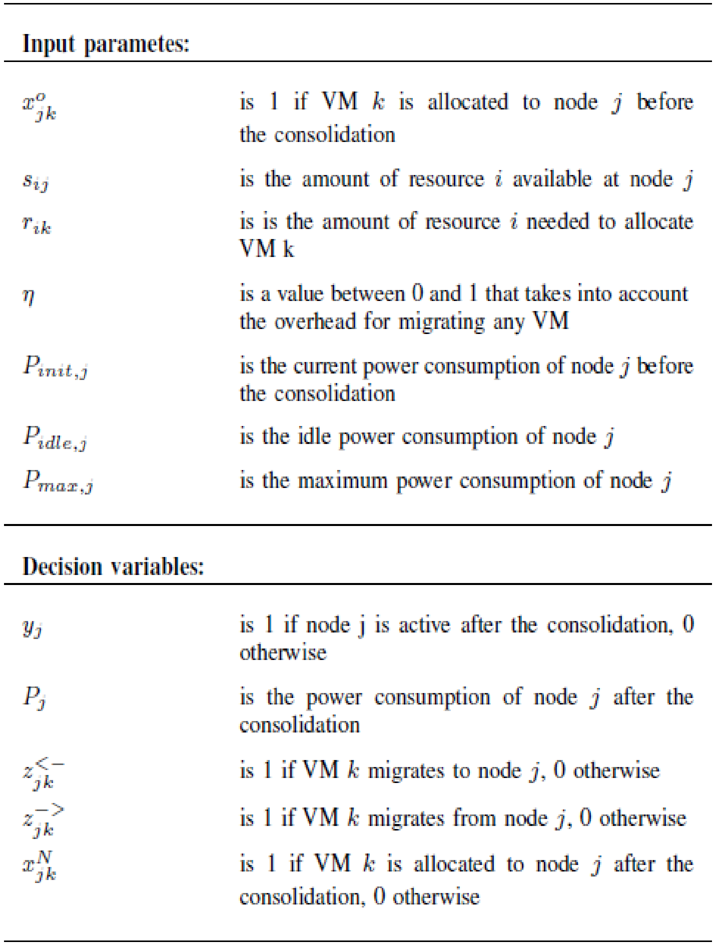
Decision Variables

Mixed integer/non integer parameters

The model takes into account the servers' power profile which, for the sake of simplicity, is linear

$$P_j(t) = P_{idle,j} + (P_{max,j} - P_{idle,j}) \cdot U_{cpu,j}(t)$$

- *Other power profiles could also be used in the model*

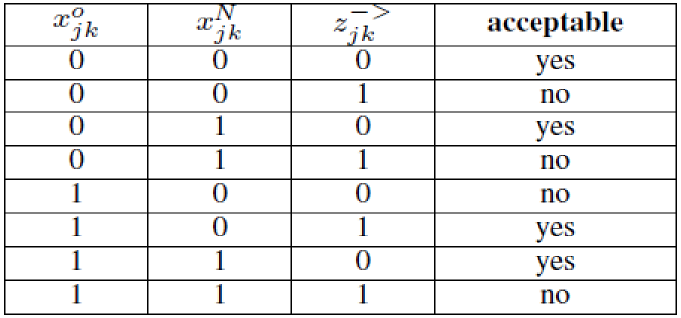


Input parametes:	
x^o_{jk}	is 1 if VM k is allocated to node j before the consolidation
s_{ij}	is the amount of resource i available at node j
r_{ik}	is is the amount of resource i needed to allocate VM k
η	is a value between 0 and 1 that takes into account the overhead for migrating any VM
$P_{init,j}$	is the current power consumption of node j before the consolidation
$P_{idle,j}$	is the idle power consumption of node j
$P_{max,j}$	is the maximum power consumption of node j
Decision variables:	
y_j	is 1 if node j is active after the consolidation, 0 otherwise
P_j	is the power consumption of node j after the consolidation
$z^{<-}_{jk}$	is 1 if VM k migrates to node j, 0 otherwise
$z^{->}_{jk}$	is 1 if VM k migrates from node j, 0 otherwise
x^N_{jk}	is 1 if VM k is allocated to node j after the consolidation, 0 otherwise

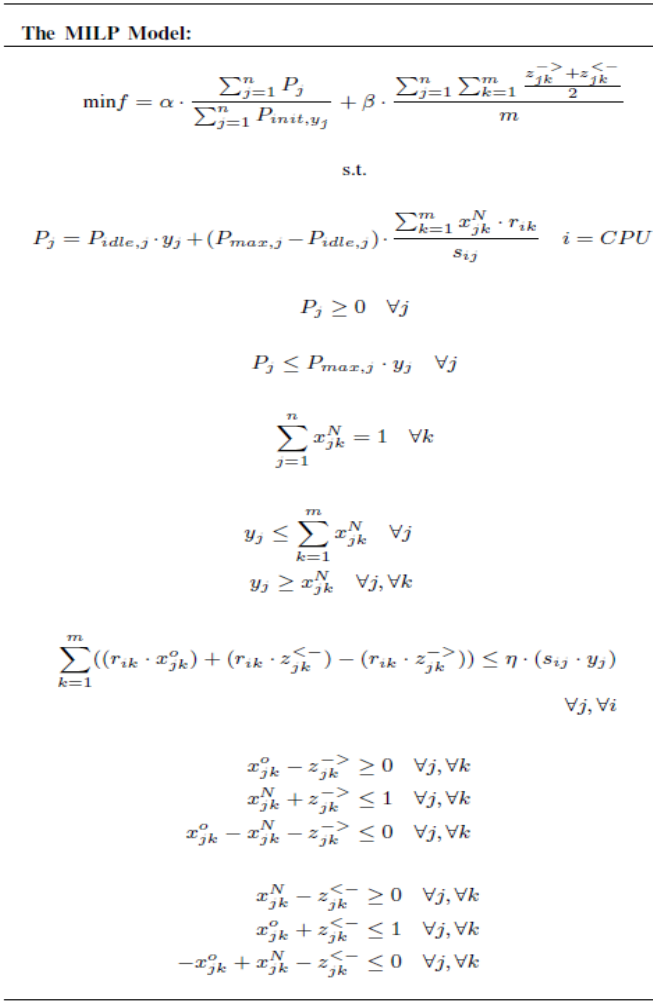
A Mixed Integer Linear Programming Model

- The objective function is the linear combination of the total power consumption after the consolidation normalized to the initial one and the number of migrations over the number of VMs

  - *Power consumption constraints*
  - *Active servers constraints*
  - *"Budget" constraints*
  - *Migration constraints*



x_{jk}^{o}	x_{jk}^{N}	$z_{jk}^{->}$	**acceptable**
0	0	0	yes
0	0	1	no
0	1	0	yes
0	1	1	no
1	0	0	no
1	0	1	yes
1	1	0	yes
1	1	1	no



The MILP Model:

$$\min f = \alpha \cdot \frac{\sum_{j=1}^{n} P_j}{\sum_{j=1}^{n} P_{init,y_j}} + \beta \cdot \frac{\sum_{j=1}^{n} \sum_{k=1}^{m} \frac{z_{jk}^{->} + z_{jk}^{<-}}{2}}{m}$$

s.t.

$$P_j = P_{idle,j} \cdot y_j + (P_{max,j} - P_{idle,j}) \cdot \frac{\sum_{k=1}^{m} x_{jk}^{N} \cdot r_{ik}}{s_{ij}} \quad i = CPU$$

$$P_j \geq 0 \quad \forall j$$

$$P_j \leq P_{max,j} \cdot y_j \quad \forall j$$

$$\sum_{j=1}^{n} x_{jk}^{N} = 1 \quad \forall k$$

$$y_j \leq \sum_{k=1}^{m} x_{jk}^{N} \quad \forall j$$

$$y_j \geq x_{jk}^{N} \quad \forall j, \forall k$$

$$\sum_{k=1}^{m} ((r_{ik} \cdot x_{jk}^{o}) + (r_{ik} \cdot z_{jk}^{<-}) - (r_{ik} \cdot z_{jk}^{->})) \leq \eta \cdot (s_{ij} \cdot y_j) \quad \forall j, \forall i$$

$$x_{jk}^{o} - z_{jk}^{->} \geq 0 \quad \forall j, \forall k$$

$$x_{jk}^{N} + z_{jk}^{->} \leq 1 \quad \forall j, \forall k$$

$$x_{jk}^{o} - x_{jk}^{N} - z_{jk}^{->} \leq 0 \quad \forall j, \forall k$$

$$x_{jk}^{N} - z_{jk}^{<-} \geq 0 \quad \forall j, \forall k$$

$$x_{jk}^{o} + z_{jk}^{<-} \leq 1 \quad \forall j, \forall k$$

$$-x_{jk}^{o} + x_{jk}^{N} - z_{jk}^{<-} \leq 0 \quad \forall j, \forall k$$

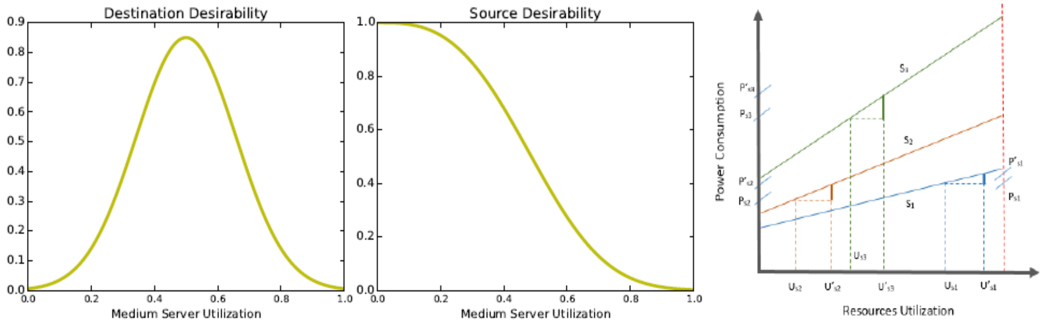
A Simulated Annealing-based Algorithm

- The algorithm follows the same approach of the ***Simulated Annealing***
  - *Perturbation phase: the current solution is perturabed by selecting a new VM migration to update the current allocation*
    - ***The perturbation phase is not random like in the original Simulated Annealing***
    - ***In order to guide the search towards efficient migrations, the concept of "desirability" is introduced***
    - ***The "desirability" of migrating a VM from a source node to a destination is defined as:***

$$D_{jh}^{k} = \frac{D_j(u_j) + D_h(u_h)}{2} + \Delta p$$

A Simulated Annealing-based Algorithm (2)

The parameters are obtained from three functions measuring how much the migration is efficient for the resource consolidation and the minimization of the overall energy consumption



(a) Destination Desirability (b) Source Desirability (c) Power Deltas

- ***A VM migration is selected on the basis of the "desirability" values of the source and destination***
  - ➢ *If they are higher than a given threshold, the total migration desirability is computed*
  - ➢ *The allocation is updated according to the migration with the highest value of desirability*

A Simulated Annealing-based Algorithm (3)

- ***Acceptance Criteria***

  - ***The difference between the current value of the objective function and the new one is calculated:***

    - *if the difference is negative the VM re-allocation is unconditionally accepted*
    - *otherwise we can accept new migrations because they could lead to consolidate new servers but with a certain probability which depends on the temperature*

$$temp = temp \cdot cooling \cdot (1 - \frac{acc\_by\_prob}{num\_vms}) \qquad prob = e^{(-\frac{delta}{temp})}$$

  - *The temperature also is dependent on the number of accepted migrations by probability*

Extended Problem

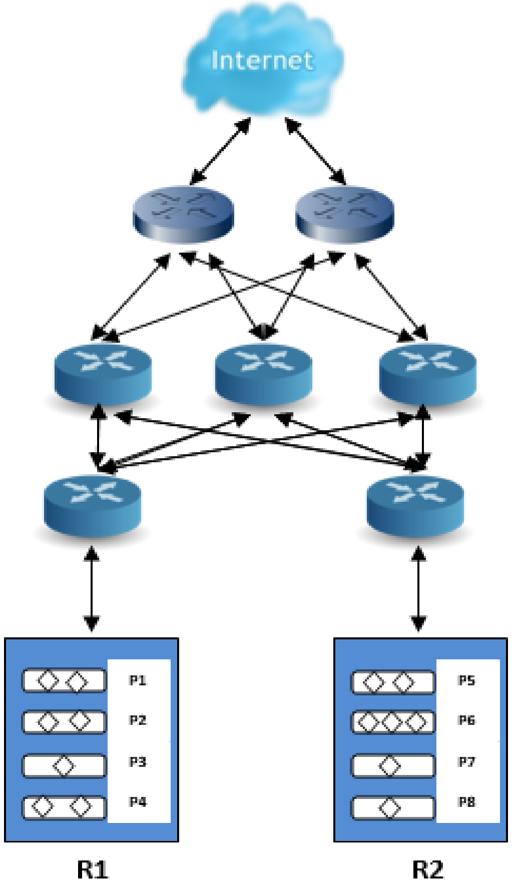
- The problem was extended in order to also take into account the network power consumption

- The new objective of the problem takes into account:

  - the power efficiency of the servers where VMs can be replaced;
  - the power efficiency of the switches sending and receiving the VMs traffic demands;
  - the number of VMs re-allocations

- The topology consists of:

  - A group of racks;
  - A hierarchy of switches;
  - A set of gateways
  - A sink (Internet)



Switch Power Consumption

- In order to compute the network power consumption we need to model the switches' power just as we did for the physical servers

- The switch power consumption can be modelled as:

$$P = P_{ch} + n_{cards} \cdot P_{cards} + \sum_{r=1}^{max} n_p^r \cdot P_p^r \cdot u_p$$

- where
  - P_{ch} is the power consumption (fixed) of the chassis;
  - N_{cards} is the number of line cards;
  - P_{cards} is their nominal power consumption
  - U_p is their the instantaneous utilization of the port p

- In the optimization model a simplified switch power profile is taken into consideration

$$P = P^{tot} + \sum_{port \in Q_{act}} P_{port}$$

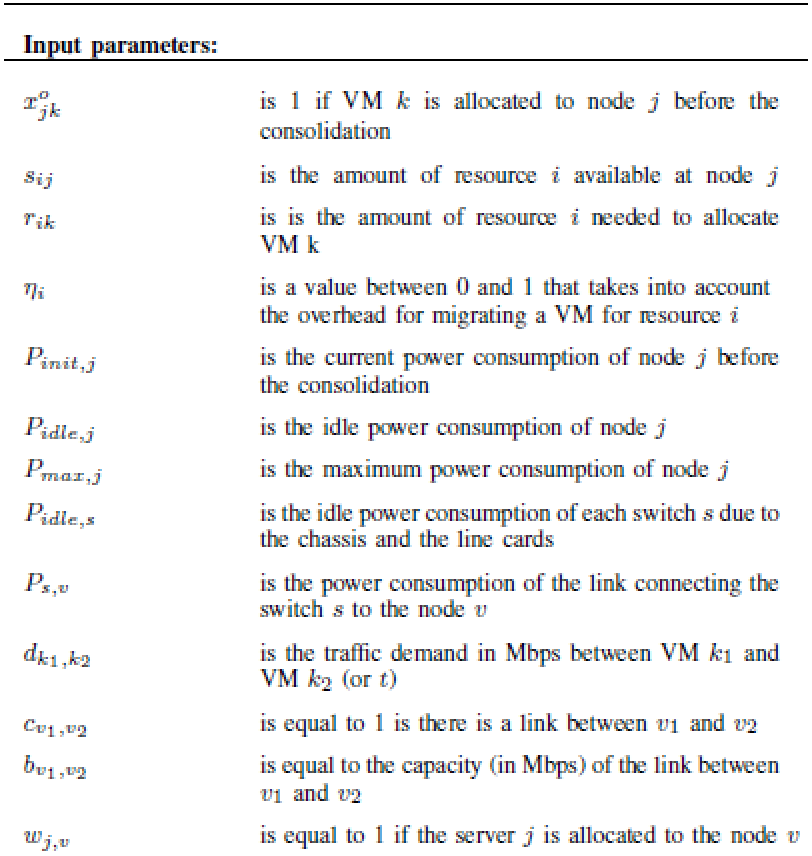
New Problem Definition: a Cross-Layer Consolidation Problem

- Considered
  - *a set of active servers, each one packed in a specific rack*
  - *their initial power consumption due to the current allocation each VM to a physical node*
  - *A set of switches along with their initial power consumption*
  - *A set of links connecting the switches each one with a maximum capacity*
  - *A set of traffic demands among the VMs*

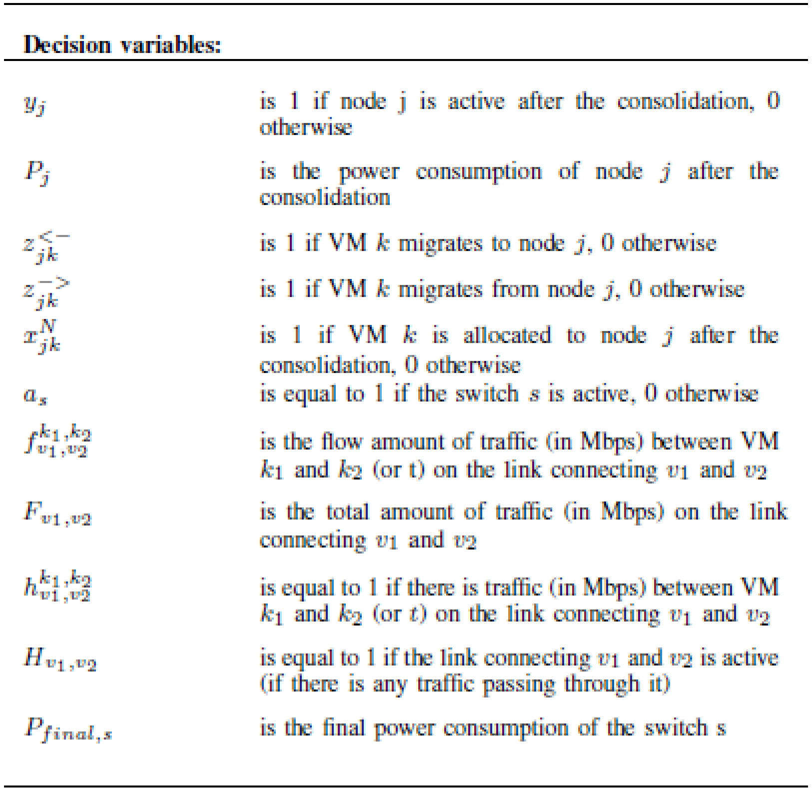
the objective is to find the set of migrations defining a new allocation scheme that minimizes the linear combination of the overall power consumption of the servers, the power consumption of the switches and the number of migrations so that all the traffic demands are also satisfied.

$$f = \alpha \cdot \frac{\sum_{j \in J^{N}_{act}} P^{N}_{act}}{\sum_{j \in J^{o}_{act}} P^{o}_{act}} + \beta \cdot \frac{|Mig|}{m} + \gamma \cdot \frac{\sum_{s \in S^{N}_{act}} Ps^{N}_{act}}{\sum_{s \in S^{o}_{act}} Ps^{o}_{act}}$$

The Problem's Parameters



Input parameters:	
x_{jk}^{o}	is 1 if VM k is allocated to node j before the consolidation
s_{ij}	is the amount of resource i available at node j
r_{ik}	is is the amount of resource i needed to allocate VM k
η_i	is a value between 0 and 1 that takes into account the overhead for migrating a VM for resource i
$P_{init,j}$	is the current power consumption of node j before the consolidation
$P_{idle,j}$	is the idle power consumption of node j
$P_{max,j}$	is the maximum power consumption of node j
$P_{idle,s}$	is the idle power consumption of each switch s due to the chassis and the line cards
$P_{s,v}$	is the power consumption of the link connecting the switch s to the node v
d_{k_1,k_2}	is the traffic demand in Mbps between VM k_1 and VM k_2 (or t)
c_{v_1,v_2}	is equal to 1 is there is a link between v_1 and v_2
b_{v_1,v_2}	is equal to the capacity (in Mbps) of the link between v_1 and v_2
$w_{j,v}$	is equal to 1 if the server j is allocated to the node v



Decision variables:	
y_j	is 1 if node j is active after the consolidation, 0 otherwise
P_j	is the power consumption of node j after the consolidation
$z_{jk}^{<-}$	is 1 if VM k migrates to node j, 0 otherwise
$z_{jk}^{->}$	is 1 if VM k migrates from node j, 0 otherwise
x_{jk}^{N}	is 1 if VM k is allocated to node j after the consolidation, 0 otherwise
a_s	is equal to 1 if the switch s is active, 0 otherwise
$f_{v_1,v_2}^{k_1,k_2}$	is the flow amount of traffic (in Mbps) between VM k_1 and k_2 (or t) on the link connecting v_1 and v_2
F_{v_1,v_2}	is the total amount of traffic (in Mbps) on the link connecting v_1 and v_2
$h_{v_1,v_2}^{k_1,k_2}$	is equal to 1 if there is traffic (in Mbps) between VM k_1 and k_2 (or t) on the link connecting v_1 and v_2
H_{v_1,v_2}	is equal to 1 if the link connecting v_1 and v_2 is active (if there is any traffic passing through it)
$P_{final,s}$	is the final power consumption of the switch s

The MILP Model
for the Combined problem

$$\min \quad f = \alpha \cdot \frac{\sum_{j=1}^{n} P_{y_j}}{\sum_{j=1}^{n} P_{init,y_j}} + \beta \cdot \frac{\sum_{j=1}^{n} \sum_{k=1}^{m} \frac{z_{jk}^{->}+z}{2}}{m} + \gamma \cdot \frac{\sum_{s=1}^{S} P_{final,s}}{\sum_{s=1}^{S} P_{init,s}}$$

s.t.

....

Switch pw consumption

$$P_{final,s} = P_{idle,s} \cdot a_s + \sum_v (H_{s,v} \cdot P_{s,v}) \quad \forall s$$

Flow Conservation Principle

$$\sum_{v_2} f_{v_2,v_1}^{k_1,k_2} \cdot c_{v_2,v_1} - \sum_{v_2} f_{v_1,v_2}^{k_1,k_2} \cdot c_{v_1,v_2} = d_{k_1,k_2} \cdot (\sum_j x_{j,k_2}^{N} \cdot w_{j,v_1} - \sum_j x_{j,k_1}^{N} \cdot w_{j,v_1})$$

$\forall v_1 \quad \forall k_1, k_2$ *not allocated to the same rack*

$$f_{v_1,v_2}^{k_1,k_2} = h_{v_1,v_2}^{k_1,k_2} \cdot d_{k_1,k_2} \quad \forall v_1, v_2 \quad \forall k_1, k_2$$

$$F_{v_1,v_2} = \sum_{k_1,k_2} (f_{v_1,v_2}^{k_1,k_2} + f_{v_2,v_1}^{k_1,k_2}) \quad \forall v_1, v_2$$

Budget Constraint

$$F_{v_1,v_2} \leq b_{v_1,v_2} \quad \forall v_1, v_2$$

Links Activation

$$H_{v_1,v_2} \leq \sum_{k_1,k_2} h_{v_1,v_2}^{k_1,k_2} \quad \forall v_1, v_2$$

$$H_{v_1,v_2} \geq h_{v_1,v_2}^{k_1,k_2} \quad \forall v_1, v_2 \quad \forall k_1, k_2$$

Switches Activation

$$a_s \leq \sum_{v2} H_{s,v2} + \sum_{v1} H_{v1,s} \quad \forall s$$

$$a_s \geq H_{v1,s} + H_{s,v2} \quad \forall s \quad \forall v1$$

The Extended Algorithm

- Given the current allocation, the initial solution to the flow routing problem is obtained through a modified version of the Dijkstra algorithm

- Every link is characterized by a cost, expressed as:

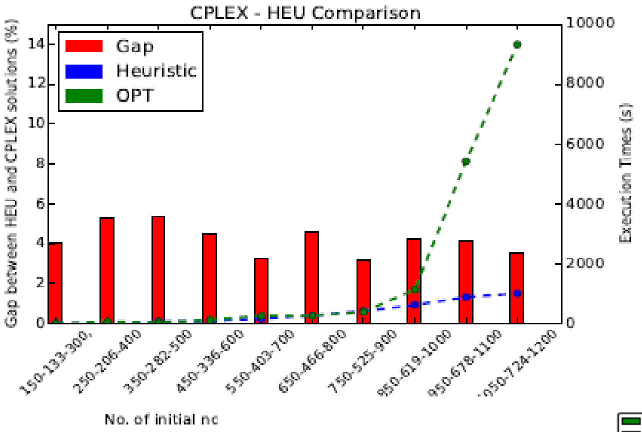
$$Cost\ (s,v) = P(s) + P(v) + P(s,v)$$

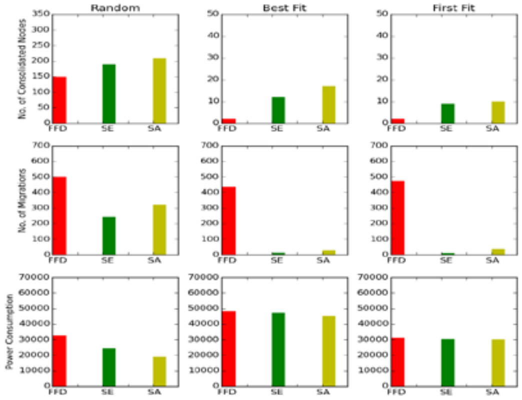
- When computing the desirability, the algorithm tries to promote migrations which do not result in a high reconfiguration cost for the established routes

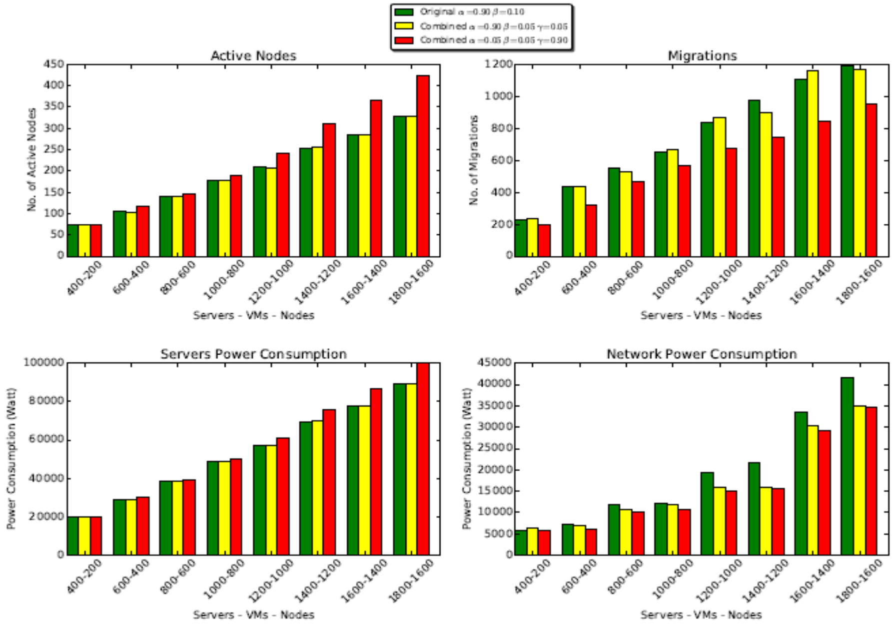
  - *A penalty is introduced in the desirability if the migration causes several path-reconfigurations*

- A migration requiring path-reconfigurations can be triggered

Some Results







Bibliography (extract)

[1] C. Ghribi, M. Hadji, and D. Zeghlache, "***Energy Efficient VM Scheduling for Cloud Data Centers: Exact Allocation and Migration Algorithms***," 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, pp. 671–678, May 2013.

[2] T. Setzer and a. Wolke, "***Virtual machine re-assignment considering migration overhead***," 2012 IEEE Network Operations and Management Symposium, pp. 631–634, Apr. 2012.

[3] D. Kapil, E. Pilli, and R. Joshi, "***Live virtual machine migration techniques: Survey and research challenges***," in Advance Computing Conference (IACC), 2013 IEEE 3rd International, 2013, pp. 963–969.

[4] C. Xian, Y.-H. Lu, and Z. Li, ***"Dynamic voltage scaling for multitasking real-time systems with uncertain execution time***", Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 27, no. 8, pp. 1467–1478, 2008

[5] L. T. Kou and G. Markowsky, "***Multidimensional bin packing algorithms***", IBM Journal of Research and Development, vol. 21, no. 5, pp. 443–448, 1977

[6] A. Murtazaev and S. Oh, "***Sercon: Server Consolidation Algorithm using Live Migration of Virtual Machines for Green Computing***", IETE Technical Review, vol. 28, no. 3, p. 212, 2011

[7] A. Beloglazov and R. Buyya, "***Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers***", Concurr. Comput. Pract. Exper., vol. 24, no. 13, pp. 1397–1420, Sep. 2012