An Energy-aware Multi-start Local Search Metaheuristic for Scheduling VMs within the OpenNebula Cloud Distribution

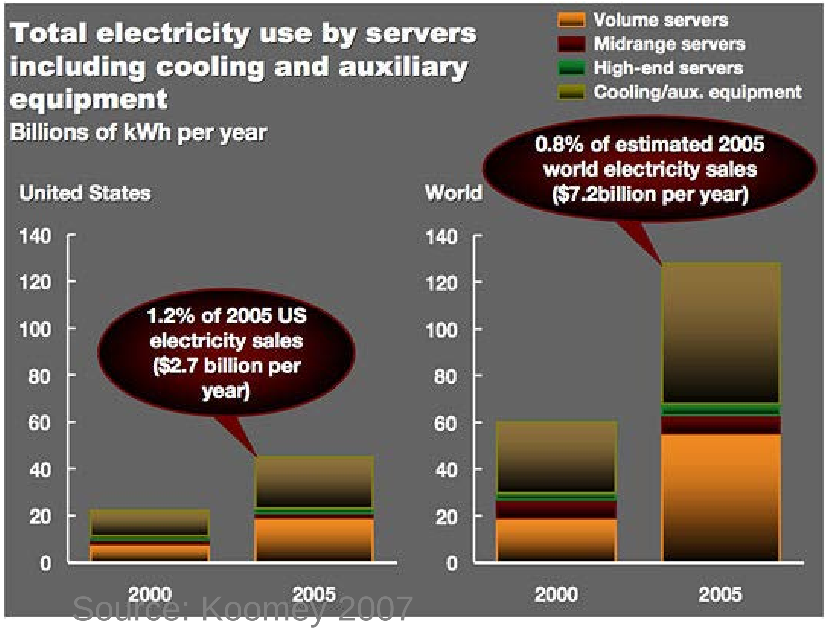
Y. Kessaci, N. Melab et E-G. Talbi

Dolphin Project Team, Université Lille 1, LIFL-CNRS, INRIA Lille-Nord Europe

Outline

- Motivation
- Cloud Models
- Problem formulation
- Problem resolution
- Experiments and Results
- Conclusion and Perspectives

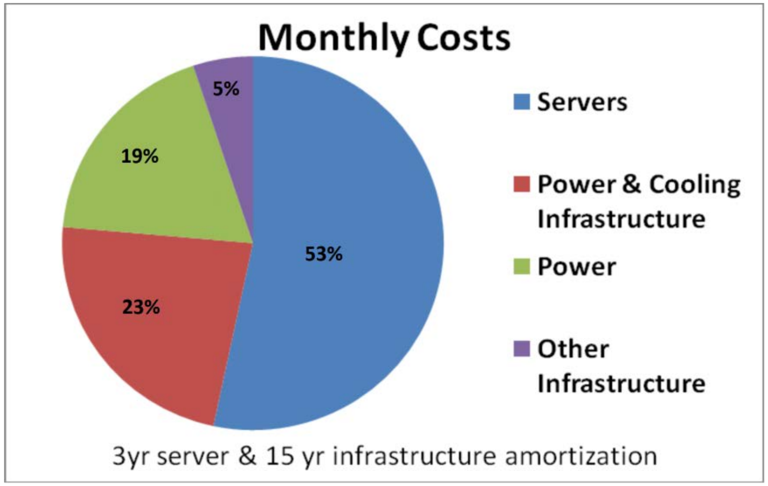
Motivation



Source: Koomey 2007

➢ The total electricity consumed by servers doubled over the period 2005 to 2010 in worldwide.

➢ Information and Communications Technology (ICT) industry accounts for approximately **2%** of CO_2 emissions.

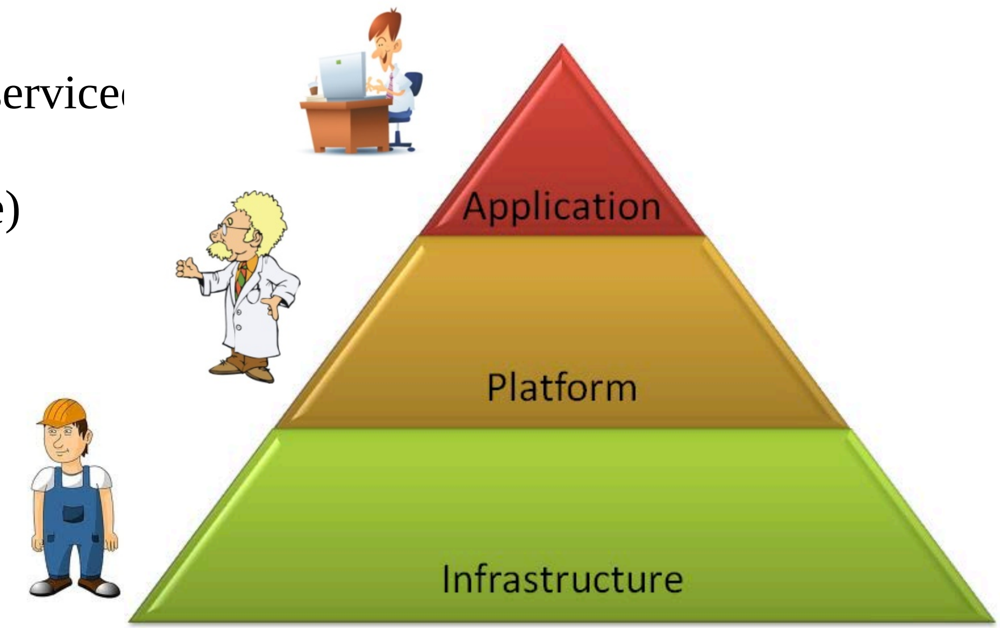


Source: Hamilton 2009

➢ Energy consumption represents more than 42% of the total data center budget.

Cloud Models

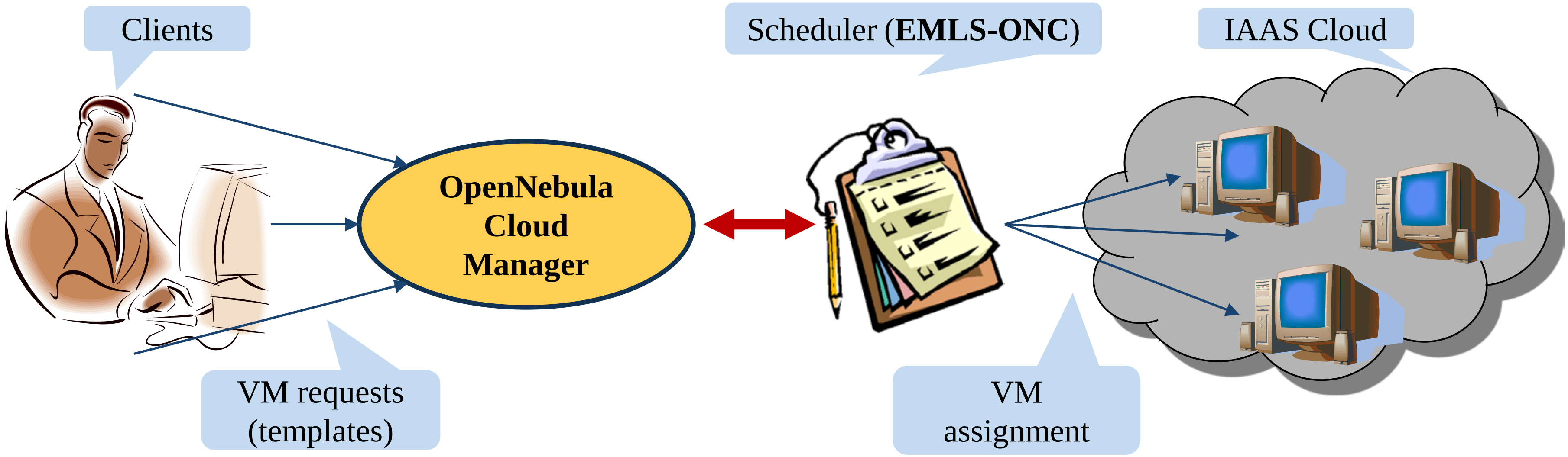
- Different according to the provided service
  - ✓ IaaS (Infrastructure as a Service)
  - ✓ PaaS (Platform as a Service)
  - ✓ SaaS (Software as a Service)



- The cloud provider deals with: the virtualization, server hardware, storage and networks.

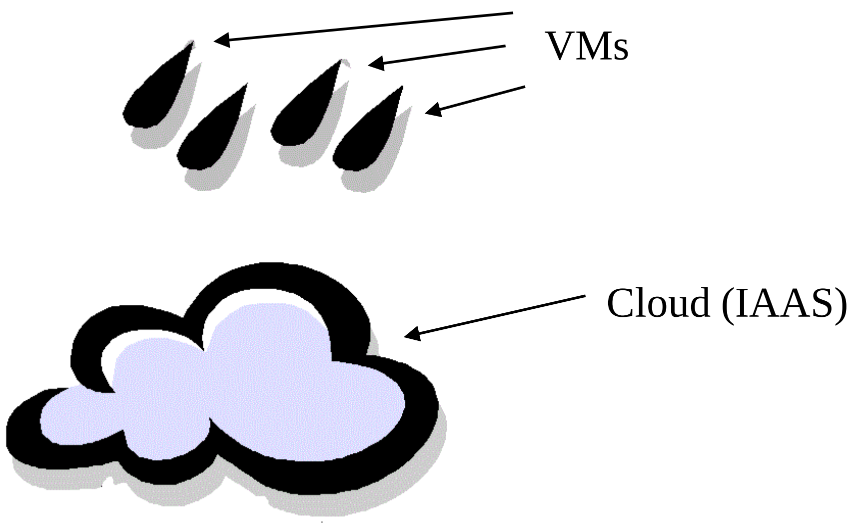
- The client deals with: applications, runtimes, SOA integration, databases and server software.
  - ✓ Two-tier model (Client, Provider).
  - ✓ Payment on demand (pay as you go).

Overview (two-tier architecture)



Virtual Machines for High Performance Computing Usage

- The VM template file (in OpenNebula) contains information about the number of CPU, memory, disk capacity, the CPU clock frequency, the time reservation …

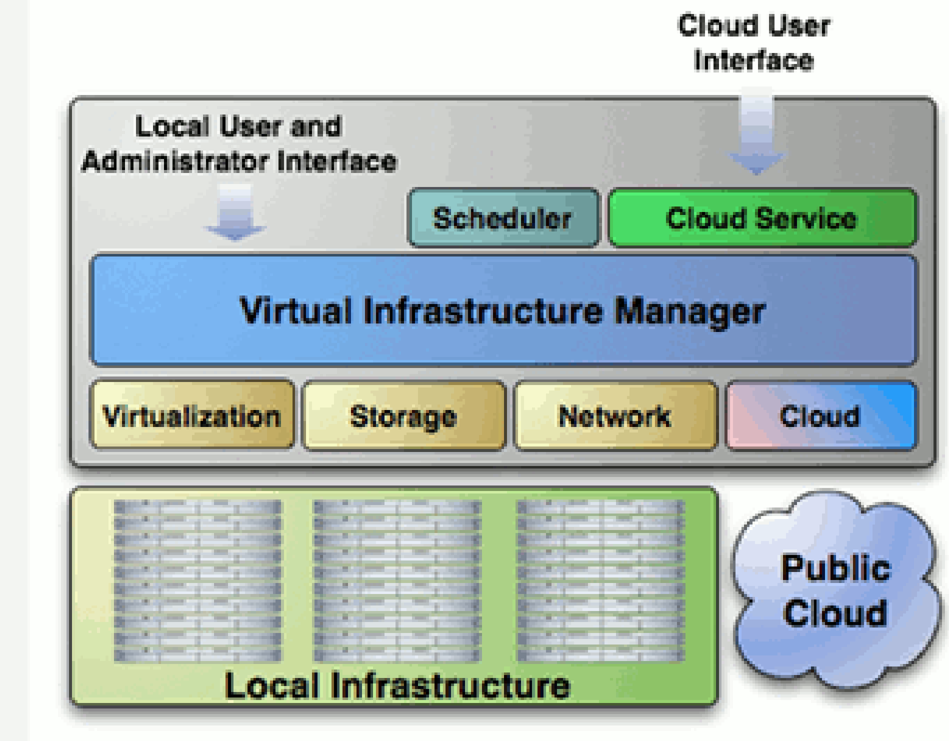


- The information used by our scheduler are given by the triplet:

$$(e_j, n_j, m_j)$$

e_j the execution time of the application (booking) – reservation time of VMs
n_j the number of reserved processors
m_j the required memory

OpenNebula

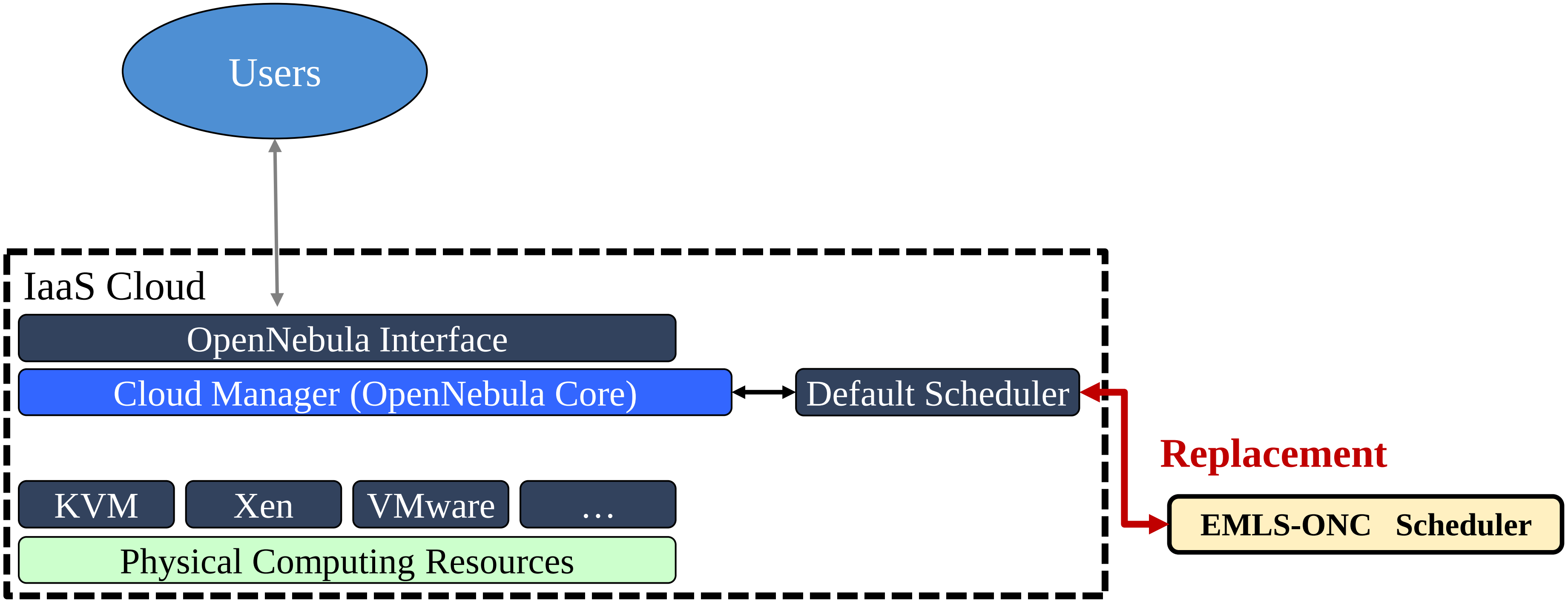


- ✓ 3 layers
- ✓Cloud Manager
- ✓ Distributed Cloud

- Infrastructure information (CPU, memory, disk, clock frequency …)➔ Hypervisor (KVM, XEN,…)
- Client requests (VM features)➔ VM templates
- Information exchange process ➔ XML-RPC

Scheduler

- Tool
  - Metaheuristic : Multi-start Local Search MetaHeuristic (EMLS-ONC).



- Solution
  - Optimal application assignment based on **objective functions** and **constraints**.

Problem formulation

- Input

  - ✓ Scheduling J VMs requests on N geographically distributed hosts (**NP hard**) respecting the constraints…

    - …Each VM can be affected to one and only one host.

    - …The host have to provide the total number of requirements requested (memory, CPU) by the VM to host the VM.

- Objectives

  - ✓ Meet maximum client's requests …

    - … while optimizing the energy consumption

Metrics (CMOS architectures)

➢Energy

- Using always the combination of host that **minimize** the energy consumption
  - ✓ Energy needed for the computation (E^c)
  - ✓ Energy required for cooling the host

$$E_{ij}^{c} = (\beta_i + \alpha_i f_{ij}^{3}) \times n_j \times e_{ji}$$

Assumption (The higher the CPU frequency is, the more cooling energy is required)

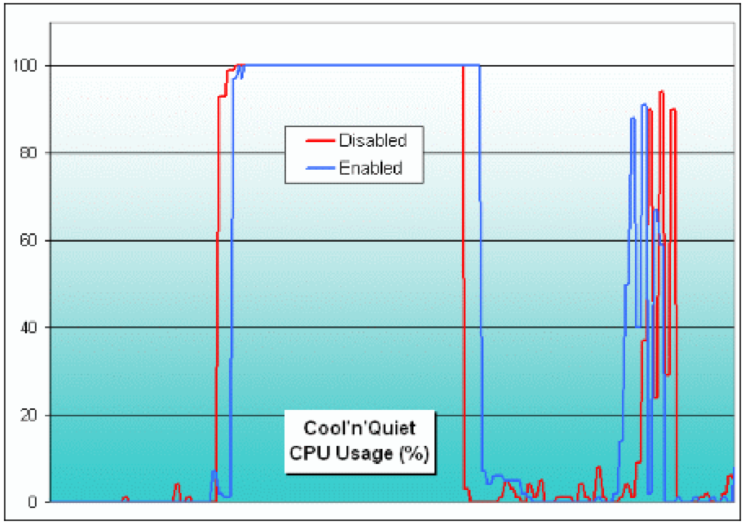
$$\textit{Minimizing the energy consumption} = \sum_{i}^{N} \sum_{j}^{J} (E)_{ij}$$

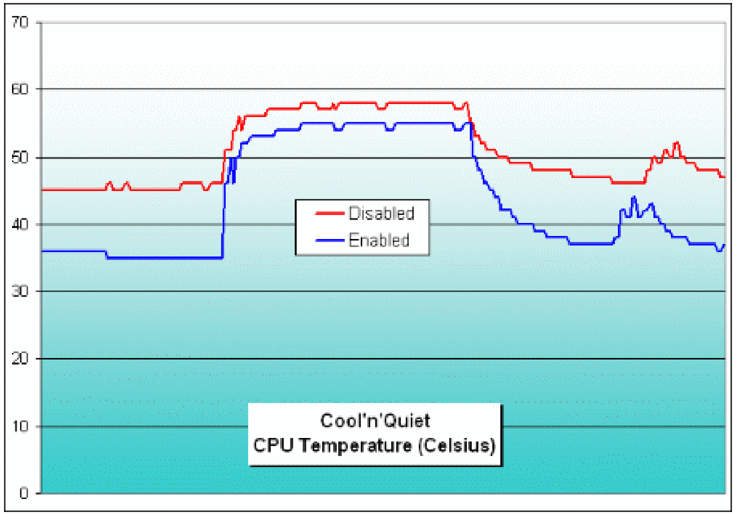
➢CO2

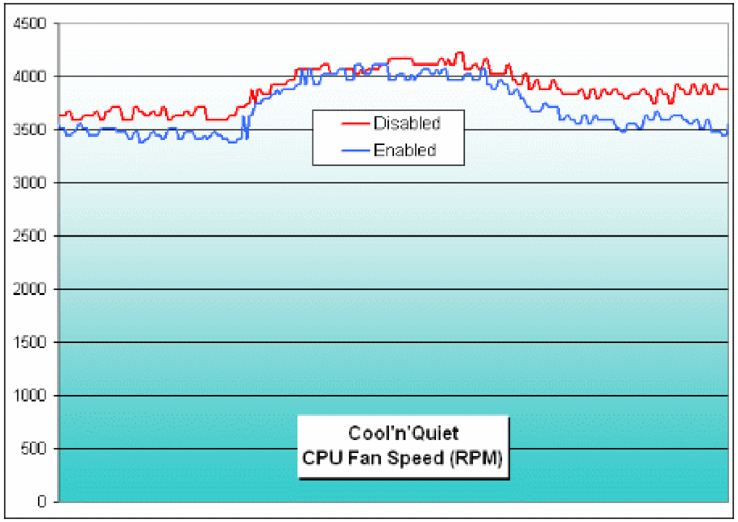
- Deducing the amount of CO2 emissions of each host of the cloud.

$$CO_2 = E_{total} \times Taux\, CO_2$$

CPU usage Vs Fan speed (cooling)
Ex: AMD Cool'n'Quiet technology



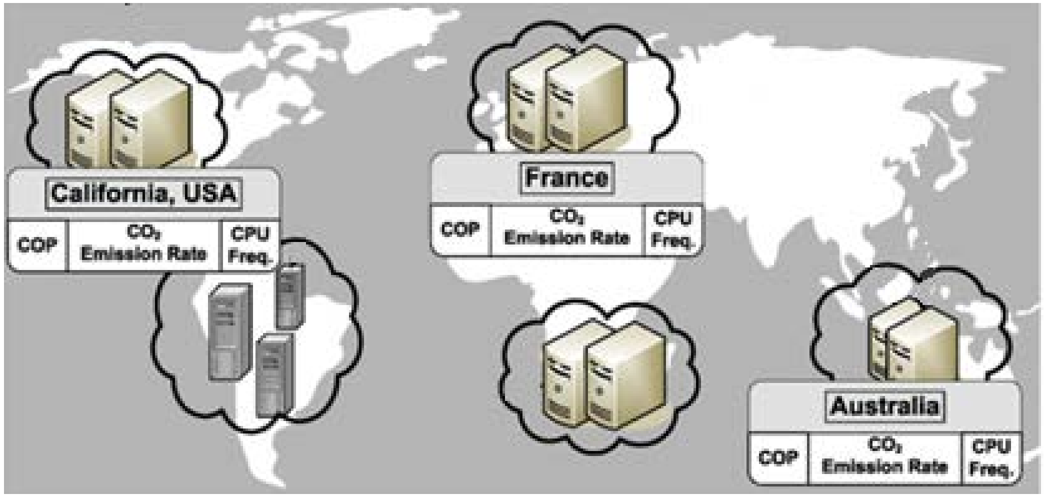




Clouds Federation

✓ Geographically distributed cloud (OpenCirrus)

✓ 3 continents (America, Europe et Oceania)



Location	Co2 Rate (kg/kW h)	Electricity price ($/kW h)	Cpu Alpha	Cpu Beta	Max Frequency (Ghz)	Opt Frequency (Ghz)	Nb processor
New York, USA	0.389	0.15	65	7.5	1.8	1.630324	2050
Pennsylvania, USA	0.574	0.09	75	5	1.8	1.8	2600
California, USA	0.275	0.13	60	60	2.4	0.793701	650
Ohio, USA	0.817	0.09	75	5.2	2.4	1.93201	540
North Carolina, USA	0.563	0.07	90	4.5	3.0	2.154435	600
Texas, USA	0.664	0.1	105	6.5	3.0	2.00639	350
France	0.083	0.17	90	4.0	3.2	2.240702	200
Australia	0.924	0.11	105	4.4	3.2	2.285084	250

CO2 rate ➔ Department Of Energy (DOE)

Electricity price ➔ Energy Information Administration (EIA)

High-level template of S-metaheuristics

Main common concept of metaheuristics

- Representation
- Objective function
- Constraint satisfaction

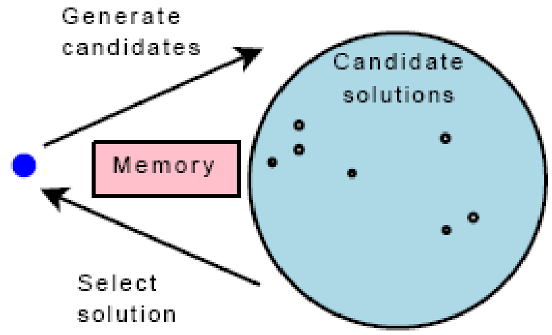
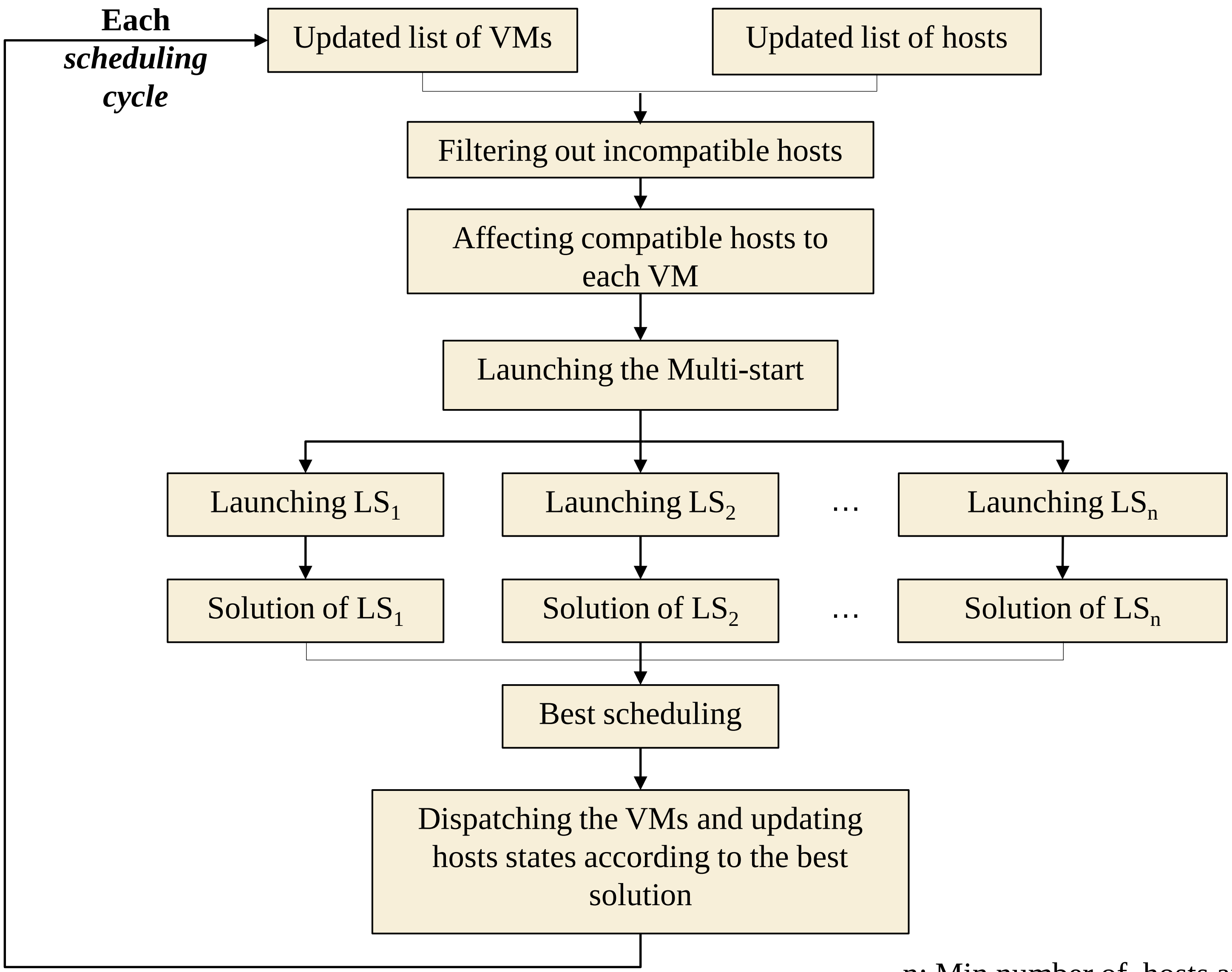


Fig. 2.1 Main principles of single-based metaheuristics.

Common concepts for S-metaheuristics

- Neighborhood, very large neighborhoods
- Initial solution
- Incremental evaluation of neighbors

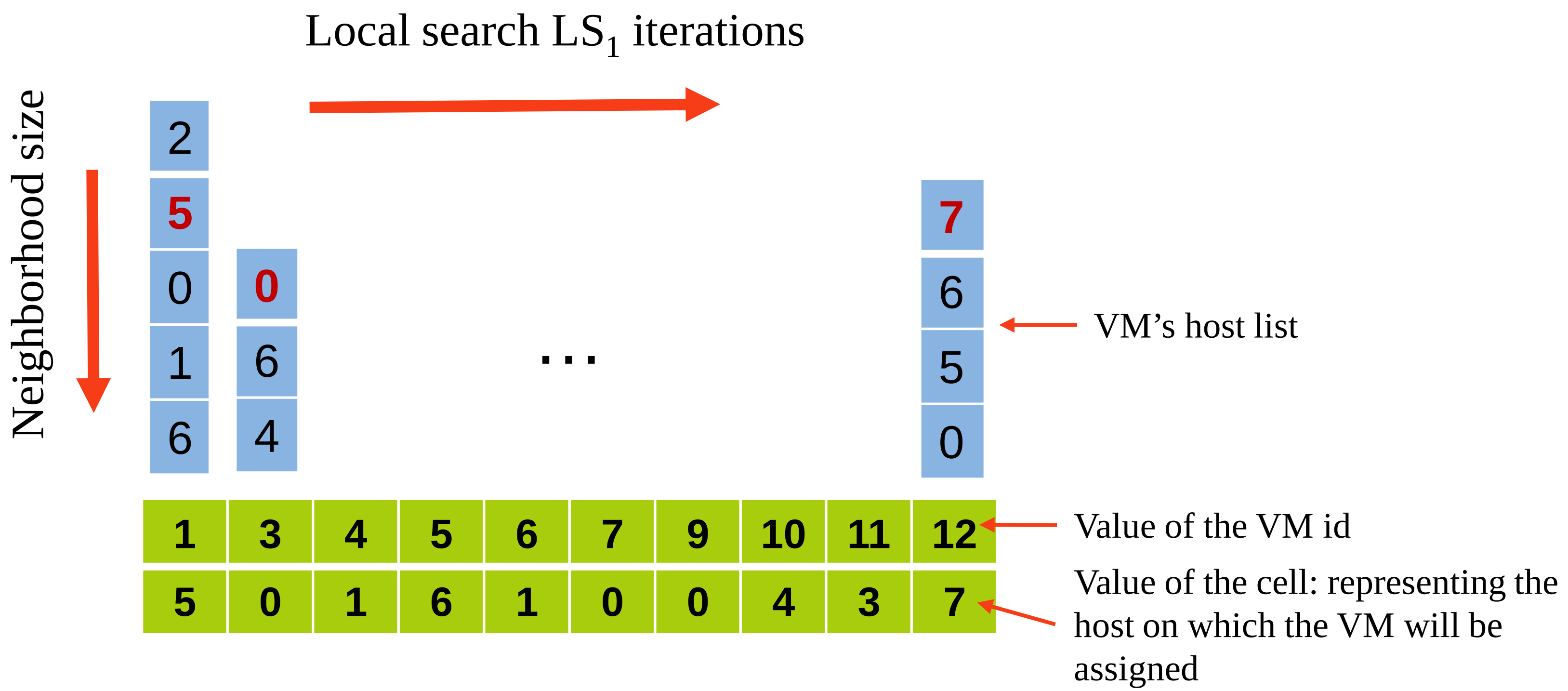
Scheduling Approach



n: Min number of hosts and 20

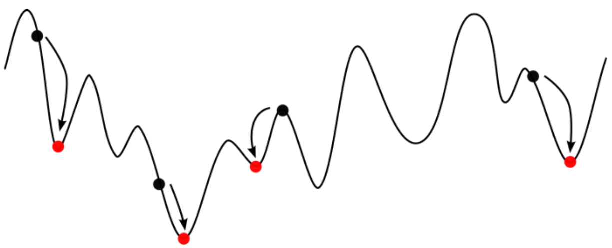
Local Search

- Initial solution

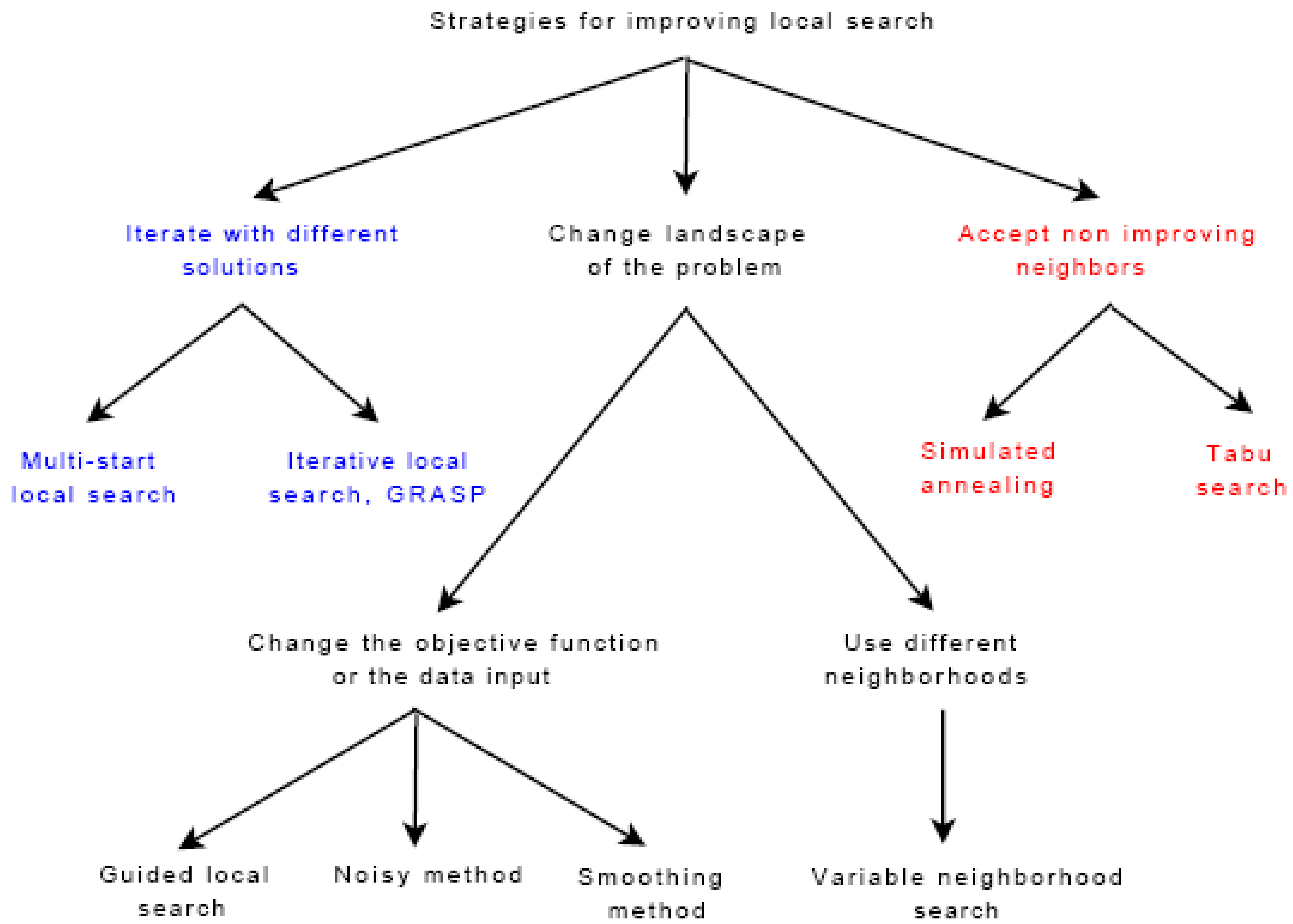


- The encoding of a solution

- The neighborhood operator is a switch of the hosts of each VM (exhaustive, random)



Escaping from local optima



Experiments

Experiment parameters

- VM and host features generation, to fit the OpenNebula parser (XML format).

- Random generation of the VM requirements:
  1- The execution time in the interval [1,10] hours.
  2- The processor requirement from [0.5,9].
  3- The memory needs from [1,3] GBs.

- Different types of hosts, changing each time their features randomly.
  1- The number of cores generated between [1,24] cores.
  2- The memory capacity from [2,24] GB.
  3- The CPU speed from [1,3] Ghz.
  4- The number of VMs already running on it [0,10].

- One scheduling cycle run (30sec), to compare the ability of both algorithms to handle a big number of VMs requests at the same time and their processing time.

- 4 VM arrival rates (1, 10, 40, 100) VMs.

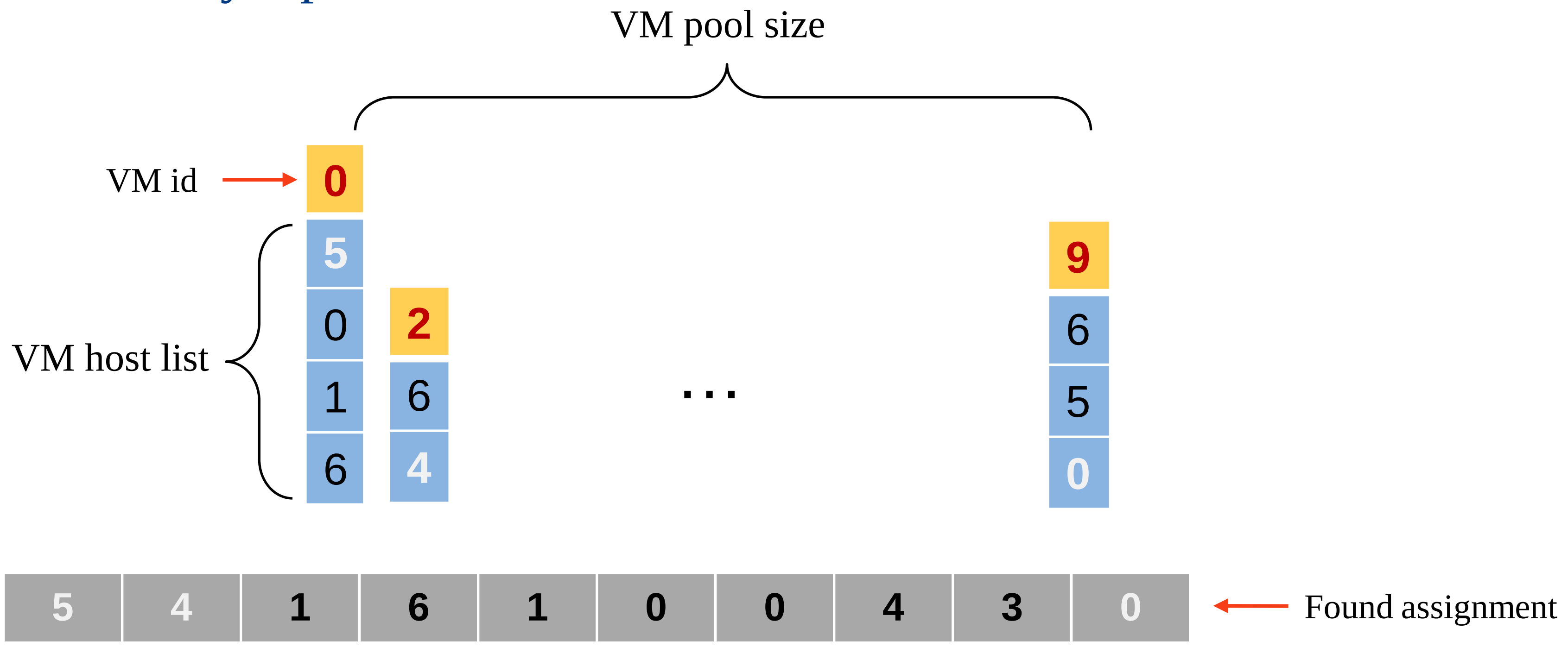
- 4 cloud sizes (5, 80, 320, 1280) machines.

Comparison

- No previous energy aware scheduler on OpenNebula Cloud Manger to compare with.

- Comparison done with the default OpenNebula scheduler.

- Greedy OpenNebula scheduler



- For each VM, takes the first host that satisfies the VM requirements.

Results (1)

 # VMs **Energy** **Time**

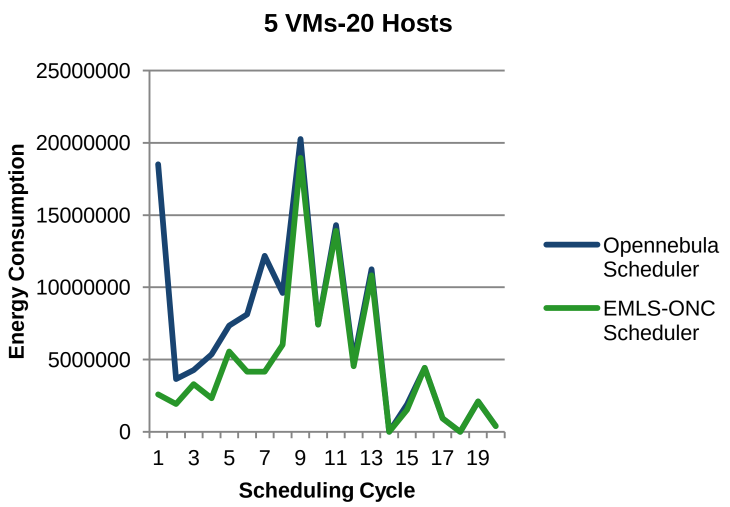
Number of VMs	1					
Number of hosts	Number of scheduled VMs		Consumed Energy (Energy Unit)		Time (s)	
	EMLS-ONC	OpenNebula scheduler	EMLS-ONC	OpenNebula scheduler	EMLS-ONC	OpenNebula scheduler
5	**1**	1	**453035**	453035	**10e-04**	10e-06
80	**1**	1	**2505407.9**	3973660	**10-e03**	10e-06
320	**1**	1	**2841140.6**	3973660	**10-e03**	10e-06
1280	**1**	1	**181747.4**	226517	**10-e02**	10e-06

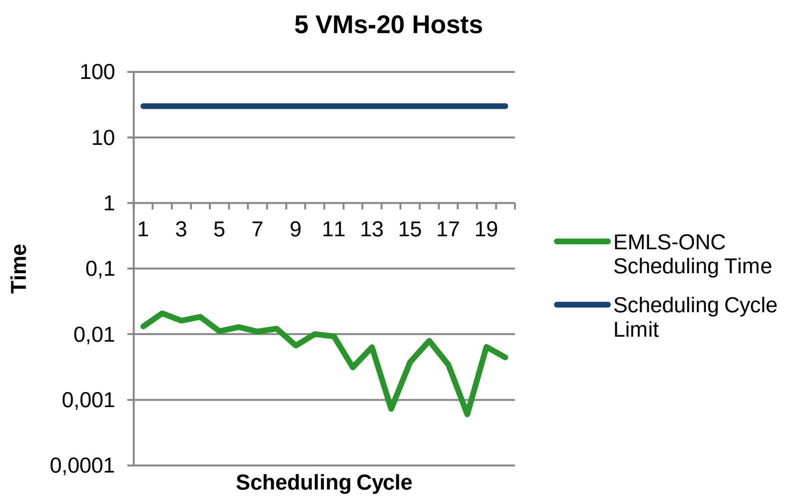
Number of VMs	10					
Number of hosts	Number of scheduled VMs		Consumed Energy (Energy Unit)		Time (s)	
	EMLS-ONC	OpenNebula scheduler	EMLS-ONC	OpenNebula scheduler	EMLS-ONC	OpenNebula scheduler
5	**6**	7	**3328294.6**	3624280	**10e-03**	10e-06
80	**10**	10	**22162200**	19340900	**10e-02**	10e-05
320	**10**	10	**22108773.3**	34809800	**10e-02**	10e-02
1280	**10**	10	**12144173.3**	13509200	**0.2**	10e-04

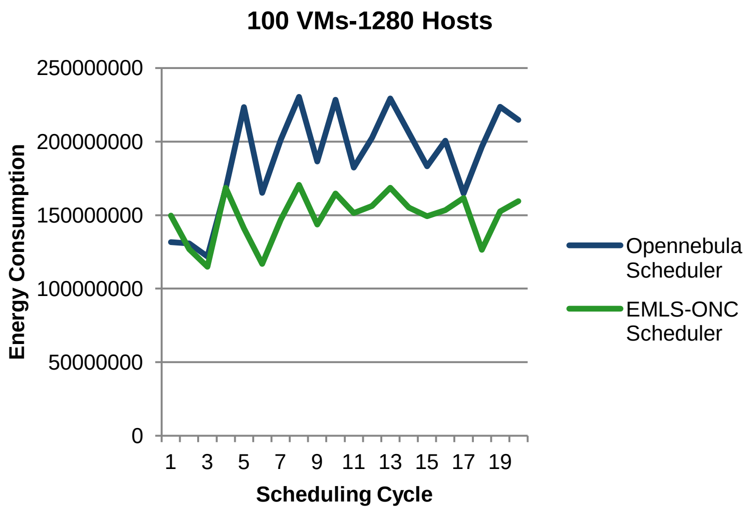
Number of VMs	40					
Number of hosts	Number of scheduled VMs		Consumed Energy (Energy Unit)		Time (s)	
	EMLS-ONC	OpenNebula scheduler	EMLS-ONC	OpenNebula scheduler	EMLS-ONC	OpenNebula scheduler
5	**11.6**	12	**25837606.7**	18146400	**10e-02**	10e-04
80	**40**	40	**56091493.3**	73667200	**0.1**	10e-04
320	**40**	40	**68386173.3**	84563000	**0.2**	10e-04
1280	**40**	40	**59823060**	65303900	**0.8**	10e-04

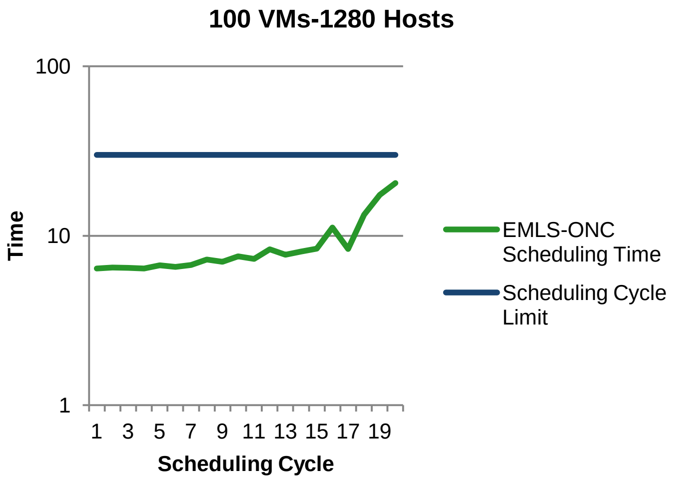
Number of VMs	100					
Number of hosts	Number of scheduled VMs		Consumed Energy (Energy Unit)		Time (s)	
	EMLS-ONC	OpenNebula scheduler	EMLS-ONC	OpenNebula scheduler	EMLS-ONC	OpenNebula scheduler
5	**13.9**	14	**3100997.3**	41853300	**10e-02**	10e-04
80	**91.7**	84	**175029600**	181780000	**0.7**	10e-03
320	**100**	100	**132239933**	149678000	**1**	10e-03
1280	**100**	100	**135882200**	161002000	**2.4**	10e-03

Results (2) – 20 cycles – 10 mn









Results (3)

- **Scheduling Success Rate Improvement**
  - The improvement compared to the default OpenNebula's scheduler concerning the number of scheduled VMs is **2.75%**.
- **Energy Improvement**
  - The total energy consumption of the distributed cloud have been reduced on average by **28.6%.**

- **Scheduler's time processing**
  - Scheduling cycle **30** sec.
  - EMLS-ONC maximum scheduling time **24.3** sec (5 s in average).

Conclusion

- We presented a new scheduler for the cloud manager OpenNebula using a multi-start local search algorithm to minimize the energy consumption.

- The energy saving of our approach exploits the disparity and the differences in the features of the hosts that compose the distributed cloud.

- Our approach satisfies the clients QoS by assigning the maximum VMs.

- EMLS-ONC gives a higher rate of scheduled VMs compared with the default OpenNebula scheduler.

- EMLS-ONC allows an improvement of the energy consumption.

- EMLS-ONC never override the scheduling cycle time.

Perspectives

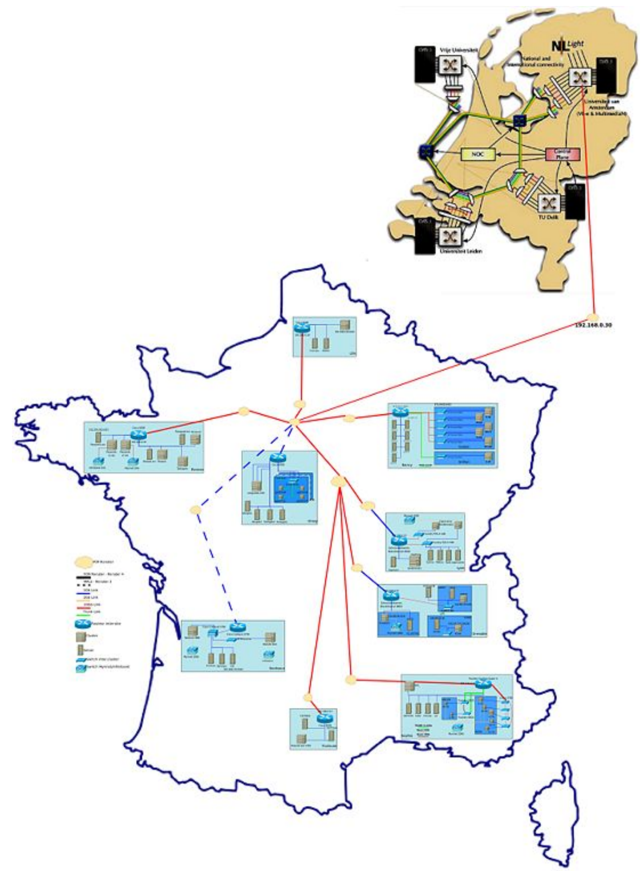
- Use Grid5000 as a test bed cloud for our next experiments. (Done, interesting preliminary results) → Validate the simulation

- Propose a better energy reduction by including other energy consumption resources like memory and hard drives.

- Integrate dynamicity in the scheduler for a real time VM's Reallocation (VM migration).

- Deploy our approach on the EGI production grid and benefit from the geographical dispersion offered by **Europe** to reduce the energy consumption.

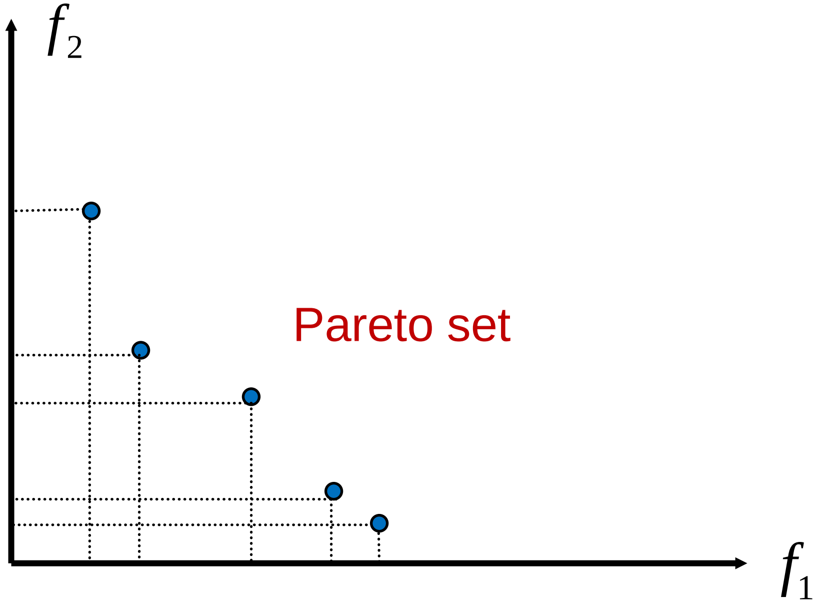
Source: EGI



Source: G5K

Perspectives (2)

➢ A multi-objective scheduler to generate non-dominated Pareto solutions (Cost of energy, CO2 emission, …).



➢ Where f_1 is the energy criterion and f_2 the VM performance.
➢ Energy criterion can be decomposed in : Cost of energy, Green aspects – CO2 emission, cooling, …

Thank you for your attention

Questions ?