

# Energy-Aware Scheduling in Virtualized Datacenters

**Íñigo Goiri**, Ferran Julià, Ramón Nou, Josep Ll. Berral,  
Jordi Guitart and Jordi Torres

Universitat Politècnica de Catalunya and Barcelona Supercomputing Center (BSC)

- Energy consumption is a large cost in datacenters
  - Server energy consumption: 11%
  - PUE overhead: 11%
- Virtualization is used to
  - Consolidate tasks in the same server
  - Save energy and reduce management complexity
- Execute HPC tasks on top
  - SLAs based on deadlines

- Virtualization adds overheads
  - Creation time
  - Migration
  - Disk management
- Aggressive consolidation for saving energy
  - May incur in performance loss
- Delay finish time of HPC tasks
  - SLA violation → pay penalty

- New scheduling policy: score-based
  - Focused on running HPC jobs
  - Reduce energy consumption
  - Manage virtualization overheads
  - Reduce management complexity
  - Reduce SLA violations

# Score-based scheduling

## Scheduling algorithm

- Decide where to run a VM dynamically
  - Evaluate every VM allocation in every server
- Give a *score* to each allocation
  - Aggregation of parameters
  - Lower score is better
- Model datacenter scheduling as a matrix:  $vm \times host$ 
  - Each cell represents the cost of allocating a *vm* in a *host*
  - N: number of virtual machines
  - M: number of active hosts
  - 1 special host: virtual queue
  - Matrix size:  $N \times (M + 1)$
- Find scheduling with global lower score

# Score-based scheduling

## Scheduling algorithm

- M servers
- $VM_1$  submitted to the datacenter
  - Scheduled to  $H_1$

	$VM_1$
$H_1$	<b>S(1,1)</b>
$\vdots$	
$H_M$	S(M,1)
$H_V$	-

# Score-based scheduling

## Scheduling algorithm

- $VM_1$  running on  $H_1$
- $VM_2$  submitted to the datacenter
  - Scheduled to  $H_M$

	$VM_1$	$VM_2$
$H_1$	<b>S(1,1)</b>	S(1,2)
$\vdots$		
$H_M$	S(M,1)	<b>S(M,2)</b>
$H_V$	-	-

# Score-based scheduling

## Scheduling algorithm

- After  $N$  VMs submitted

	$VM_1$	$VM_2$	$VM_3$	$VM_4$	...	$VM_N$
$H_1$	<b>S(1,1)</b>	S(1,2)	S(1,3)	S(1,4)		S(1,N)
$\vdots$					$\ddots$	
$H_M$	S(M,1)	<b>S(M,2)</b>	S(M,3)	S(M,4)		S(M,N)
$H_V$	-	-	-	-		-

# Score-based scheduling

## Calculate score

- Score of a tentative allocation of virtual machine  $VM$  in server  $H$
- Aggregation of parameters
  - Requirements (booleans)
  - Resources (booleans)
  - Virtualization overhead (time)
  - Power (watts)
  - ...
- $Score = weight_1 \cdot requirements + weight_2 \cdot resources + \dots$
- Lower score, better allocation
  - Impossible allocations,  $\infty$  score

# Score-based scheduling

Score calculation: Hardware, software, and resource requirements

- If the host cannot fulfill the VM requirements:
  - Lacks required hardware: number of CPUs, disk. . .
  - Lacks required software
  - Lacks required hypervisor
  - $\infty$  score
  
- If the host does not have enough free resources:
  - Not enough CPU, memory. . .
  - $\infty$  score

# Score-based scheduling

Score calculation: Virtualization overhead

- Overhead introduced by virtualization management
  - Time to create the VM
  - Time to migrate the VM
- Avoid operating on VMs undergoing migrations
- Minimize concurrent operations
  - Interfere with other actions and make actions take longer
  - Creating two VMs at the same time is slower

# Score-based scheduling

Score calculation: Energy efficiency and others

- Estimate energy usage
  - Reward almost full servers: low score
  - Penalize empty servers: high score
- Other parameters which can be added:
  - Fault tolerance
  - SLA enforcement

# Matrix solving

System modeled as a matrix

- Put all the scores in the matrix:
  - Bold indicates current allocation
  - $VM_4$  cannot be executed and it is in the queue

	$VM_1$	$VM_2$	$VM_3$	$VM_4$	...	$VM_N$
$H_1$	15.2	15.2	$\infty$	15.2		<b>10.0</b>
$H_2$	$\infty$	<b>7.8</b>	7.8	7.8		$\infty$
$H_3$	<b>10.3</b>	10.3	$\infty$	10.3		10.5
$\vdots$					$\ddots$	
$H_M$	11.0	$\infty$	<b>11.0</b>	11.0		$\infty$
$H_V$	$\infty$	$\infty$	$\infty$	$\infty$		$\infty$

# Matrix solving

## Solving scheduling

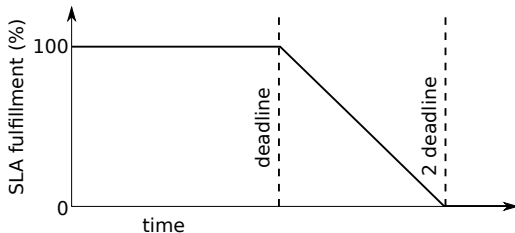
- Calculate scores of the current allocation
- Optimize matrix to get lower scores for VM allocations
  - Hill climbing
- Apply changes to the system
  - Create VM
  - Migrate VM between nodes
  - Keep VMs that cannot be executed in the queue
  - Apply turn on/off policy

# Energy savings

## Turn on/off policy

- Turn on/off approach: save energy
  - Use two thresholds
  - Turn off idle servers
    - Turn off servers as soon as they are not used
  - Turn on new machines if they are required
    - Wait until machines are required
- More consolidation
  - More energy savings
  - Lower performance

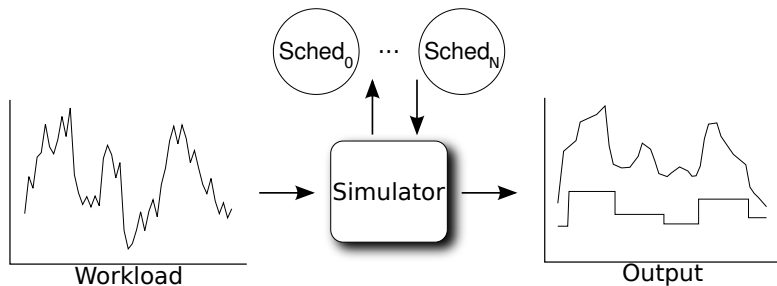
- Simulated environment
- Workload
  - One week of Grid 5000 workload
  - $\sim 2000$  tasks with an average of  $\sim 5000$  seconds per task
- 100 virtualized hosts
- Metrics: energy consumption, client satisfaction



# Evaluation

## Power simulator

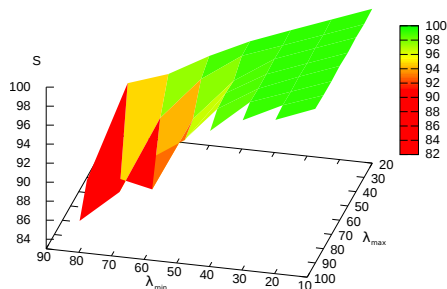
- Simulate nodes with different features
  - Fast and reproducible results
- Schedulers
  - Random
  - Round robin
  - Backfilling
  - Dynamic backfilling
  - Score-based



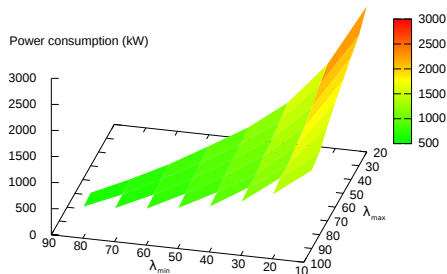
# Evaluation

## Energy consumption vs. SLA fulfillment trade-off

- More aggressive consolidation (left part)
  - More energy savings
  - Fulfill fewer SLAs
- Less aggressive consolidation (right part)
  - Less energy savings
  - Fulfill more SLAs



(a) Client satisfaction



(b) Power consumption

# Evaluation

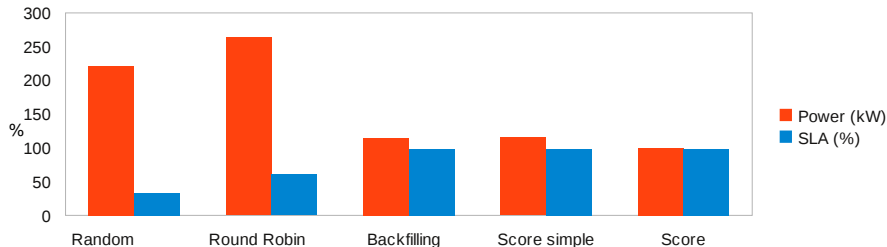
## Static allocation

### ● Metrics

- Normalized average power to the best policy
- Client satisfaction

### ● Static scheduling policies

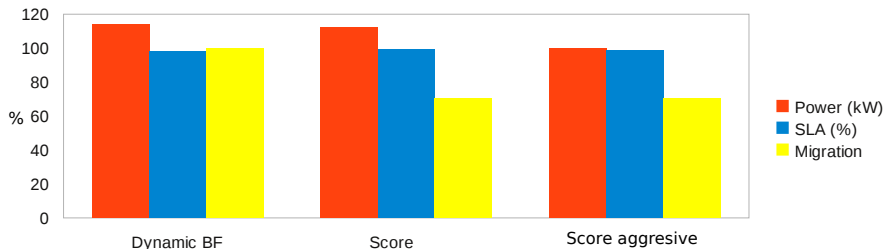
- Do not reschedule: no migration



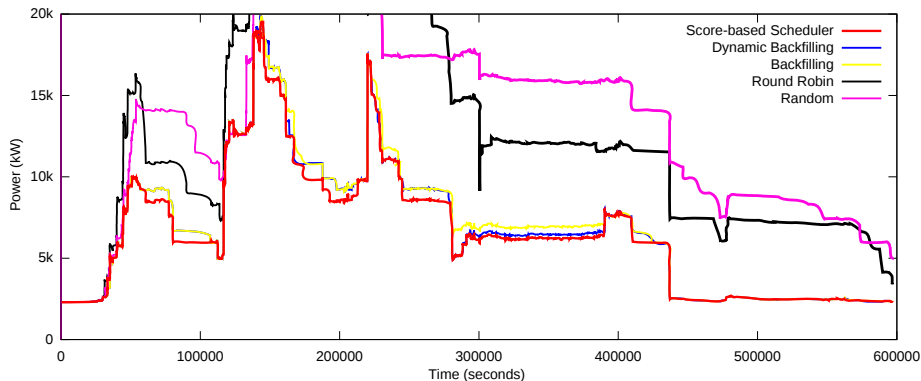
# Evaluation

## Impact of migration

- Add migration capability to scheduling
  - Consolidation is increased
  - Score-based gets better results



- Comparing energy usage for different scheduling policies



# Conclusions and future work

- Improve energy efficiency
- Deal with virtualization overheads
  - More SLAs are fulfilled
- Intuitive formulation
- Easy to extend
  
- Future work
  - Add economic model
    - Move from score to economic units
  - Evaluate reliability
  - Extend to other applications with different SLAs: services, transactional

# Energy-Aware Scheduling in Virtualized Datacenters

**Íñigo Goiri**, Ferran Julià, Ramón Nou, Josep Ll. Berral,  
Jordi Guitart and Jordi Torres

Universitat Politècnica de Catalunya and Barcelona Supercomputing Center (BSC)