Ensenada Center for Scientific Research and Higher Education

Energy consumption and quality of service optimization in containerized cloud computing

Rewer Canosa

CICESE Research Center,
Ensenada, Baja California, México
rewer@cicese.edu.mx

ISPRAS OPEN 2018

Ivannikov ISPRAS OPEN Conference

Moscow, Russia, November 22, 2018

Virtualization technologies

Server virtualization technologies

VM
hardware−level virtualization
(Xen, KVM, VMware ESX)

Container (OS+Application)
operating system−level virtualization
(LXC, Docker, BSD Jails, Windows Containers)

Containers have:

- low-overhead virtualization
- improved performance when compared to VMs
- management is much faster than VMs
- more efficient hardware performance than virtual machines
- reduced startup time

1. P. Sharma, L. Chaufournier, P. Shenoy, and Y. C. Tay, “Containers and Virtual Machines at Scale,” Proc. 17th Int. Middlew. Conf. - Middlew. ’16, pp. 1–13, 2016.

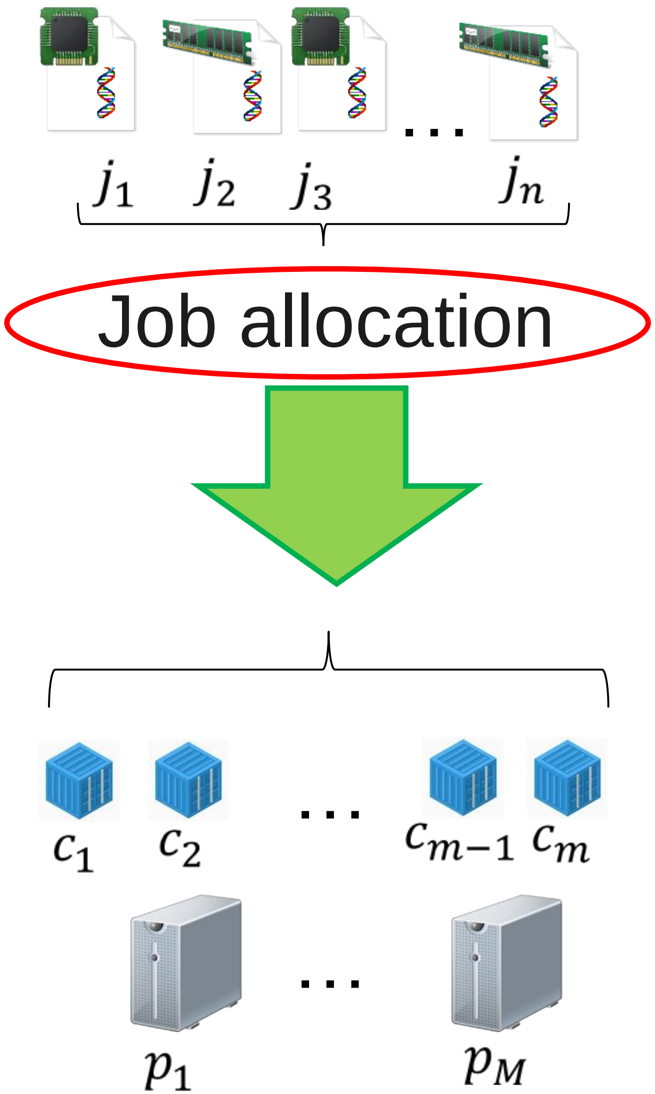
Applications of containers

Containerization technology has been implemented on large scale by cloud companies such as Google and Facebook[1]

Google launches more than 2 billion containers per week considering all of its data centers[2]

1. S. F. Piraghaj, "Energy-Efficient Management of Resources in Container-based Clouds," no. March, p. 220, 2016
2. https://opensource.googleblog.com/2014/06/an-update-on-container-support-on.html

Problem statement



Let $J = \{j_1, j_2, .., j_n\}$ - a set of jobs

Each j_j job is characterized by:

- r_j: release time (sec.)
- s_j: minimum processing capacity (MIPS)
- w_j: amount of work (Instructions)
- ρ_j: type of work.
  - CI: CPU-intensive
  - MI: Memory-Intensive

$C = \{c_1, c_2, .., c_m\}$ - set of containers

q_i - maximum processing capacity of container c_i(MIPS)

$P = \{p_1, p_2, .., p_M\}$ - set of servers (processors)

Q_k - maximum processing capacity of server p_k(MIPS)

Pandit, D., Chattopadhyay, S., Chattopadhyay, M., Chaki, N. 2014. Resource allocation in cloud using simulated annealing. In 2014 Applications and Innovations in Mobile Computing (AIMoC), 21-27

Criteria

Satisfy the minimum CPU capacity requirement for each job

$$SLA = \sum_{i=1}^{m} \sum_{\forall_{j \in J(c_i)}} \propto_{i,j} \forall_{c_i} \in \boldsymbol{C}, \forall_j \; work\ assigned\ to\ each\ c_i$$

Energy consumption

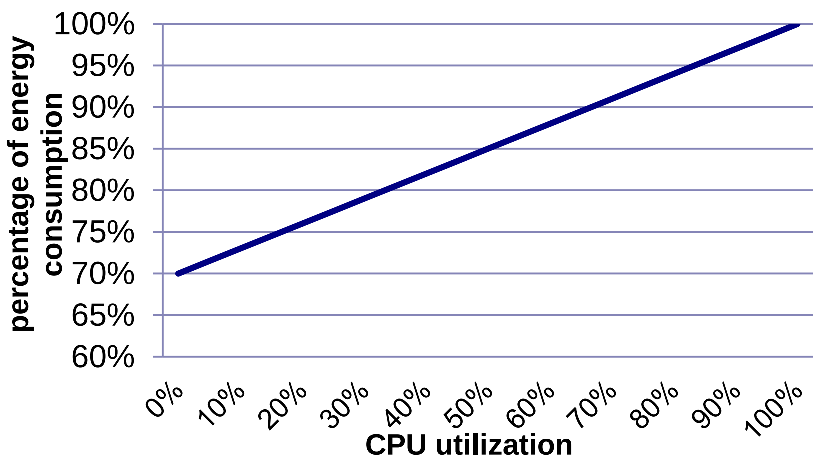
Energy consumption of the infrastructure

$$E = \int_{t=1}^{C_{max}} E^{op}(t)dt$$

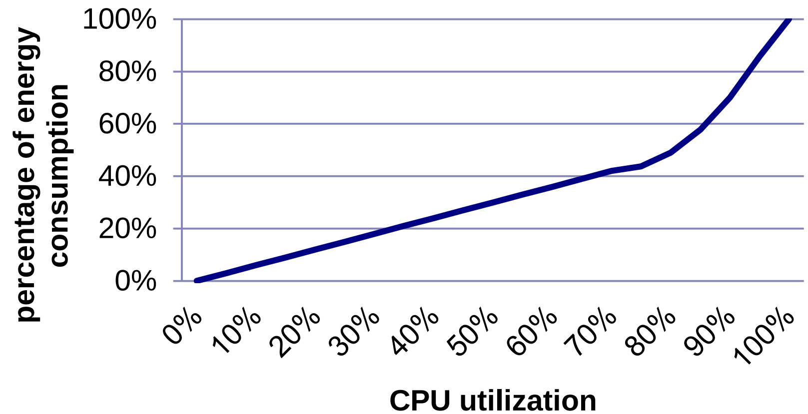
$$\begin{cases} E^{op}(t) = \sum_{k=1}^{M} e_k^{proc}(t) \\ e_k^{proc}(t) = o(t)\left(e_k^{idle} + e_k^{used}(t)\right) \\ e_k^{used}(t) = \left(e_k^{max} - e_k^{idle}\right) * F(t) * g\left(\alpha_{CI}(t)\right) \end{cases}$$

F. Armenta, A. Tchernykh et al. Min Concentration policy for Scheduling Applications with Resource Contention

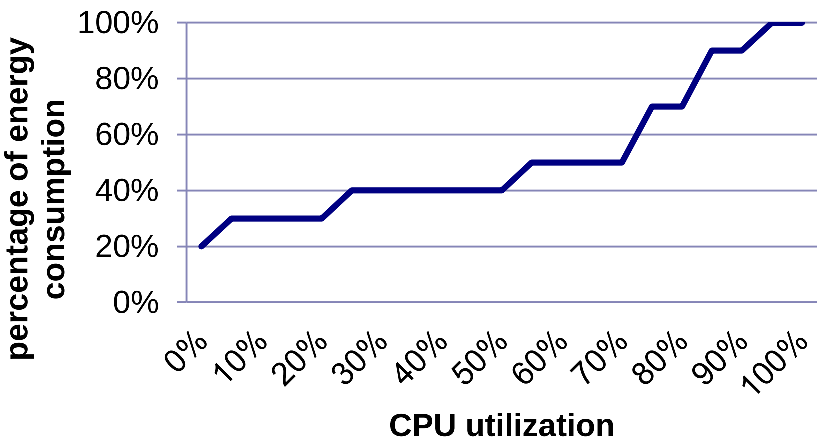
Energy models



A. Beloglazov, et.al "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing" 2012.



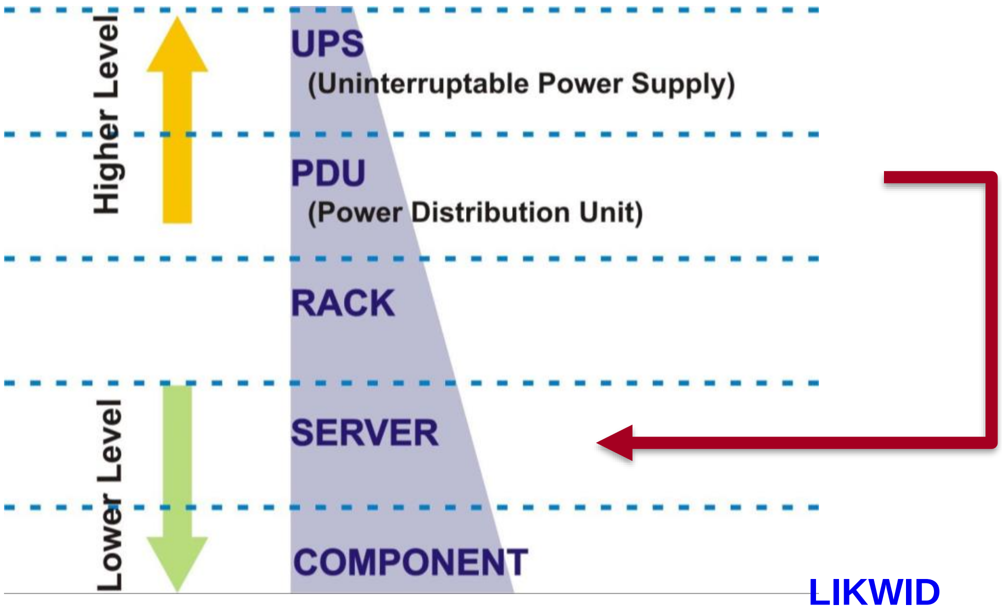
Y. Gao, et. al "An Energy and Deadline Aware Resource Provisioning, Scheduling and Optimization Framework for Cloud Systems," 2013.



C.-H. Hsu, et. al, "Optimizing Energy Consumption with Task Consolidation in Clouds," 2014.

Power distribution

Benchmark: SysBench



likwid-pin

likwid-powermeter

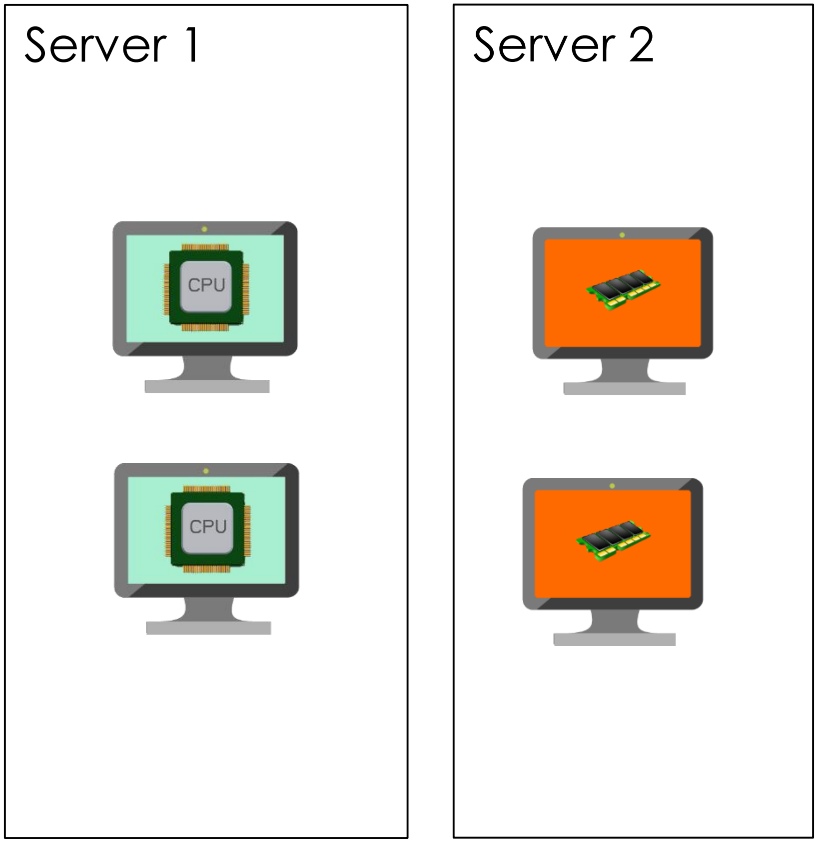
Power Distribution Unit (PDU)

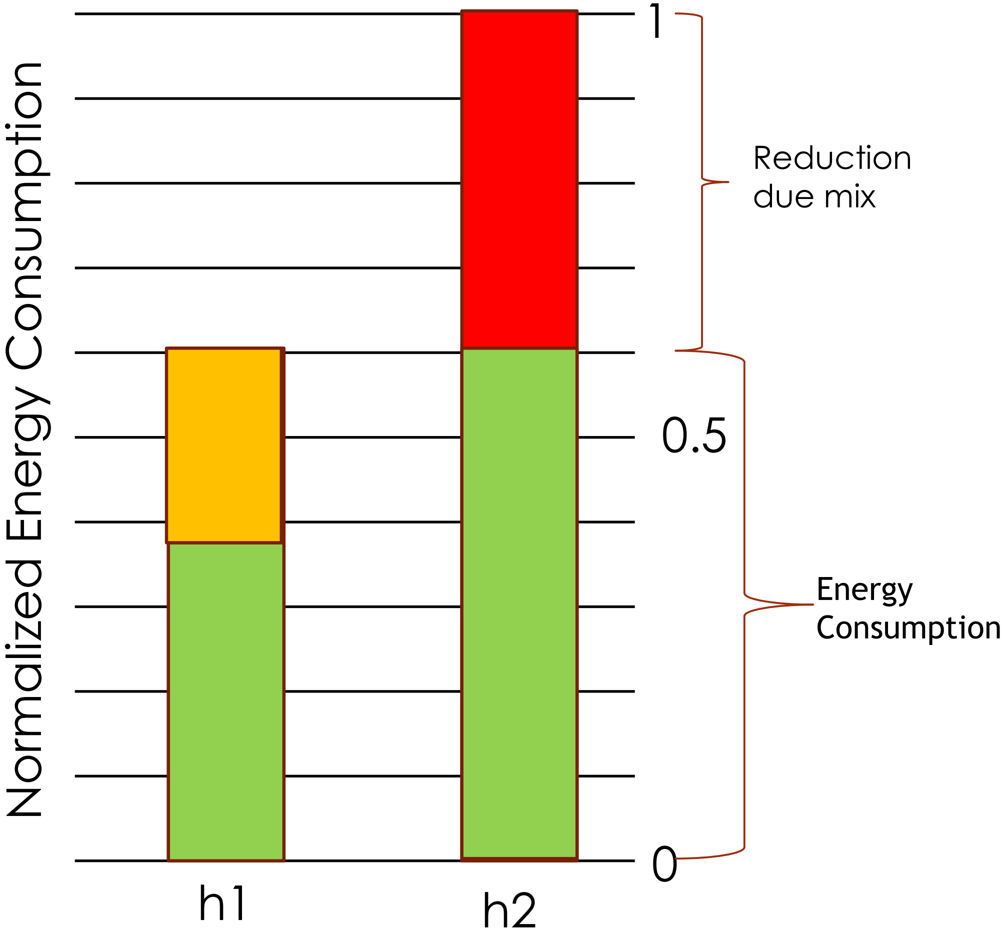
VMR-8HD20-1 Outlet Metered PDU Dual 20A 120V (8)5-15R

Benchmarks

Benchmark	CI	MI	NI	DI
LINPACK	●			
STREAM		●		
SysBench	●	●		●
iperf			●	
IOR				●
IOzone				●
NPB	●	●		●
Netperf			●	
SPEC	●	●		

Concentration

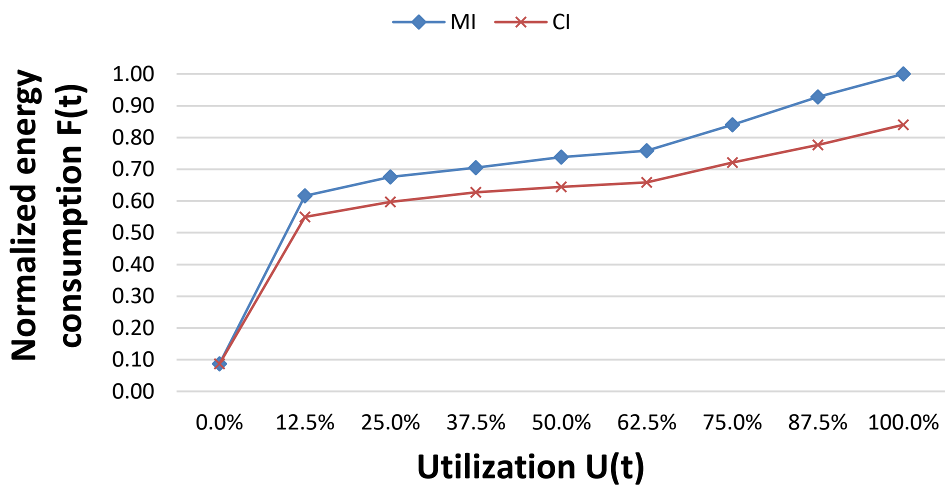




Utilization function $F(t)$

$f_d(U_d(t))$ - fraction of power consumption when a CI or MI application is executed

$$F(t) = \sum_{\forall d} f_d(\,U_d(t))\ , 0 \le F(t) \le 1, d \in \{CI, MI\}$$



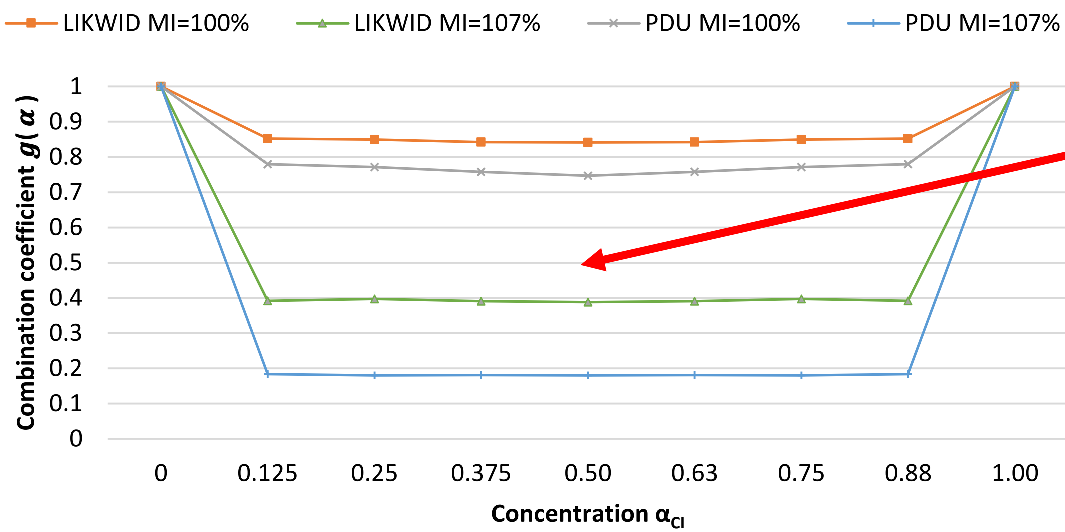
Normalized energy consumption versus CPU utilization

$U_T(t)$ - the total CPU utilization at time t:

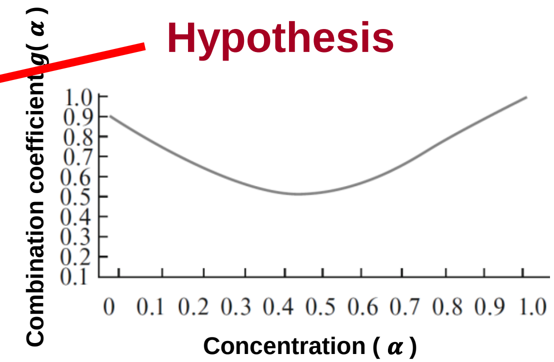
$$U_T(t) = U_{CI}(t) + U_{MI}(t)$$

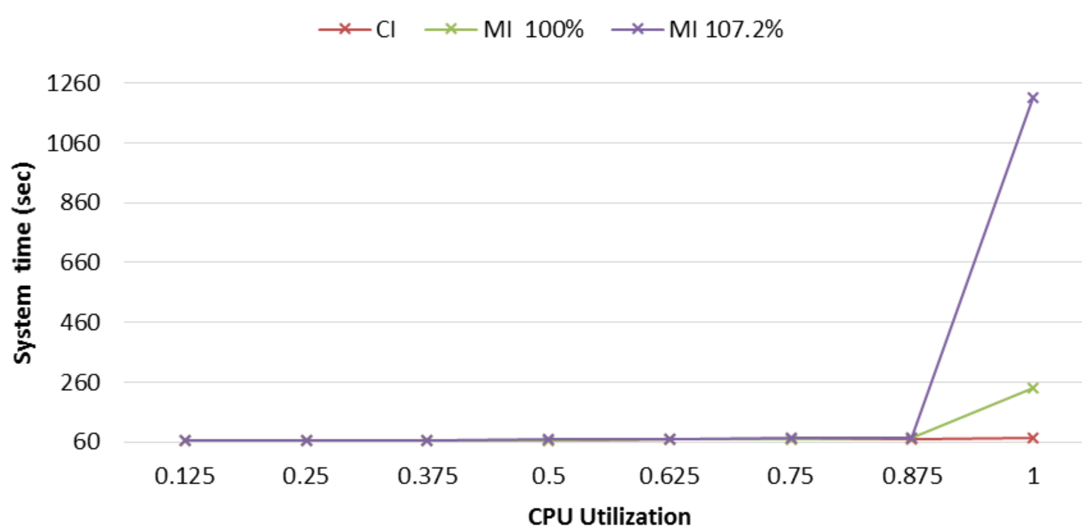
F. Armenta, A. Tchernykh et al. Min Concentration policy for Scheduling Applications with Resource Contention

Combination coefficient $g(\alpha_{CI}(t))$

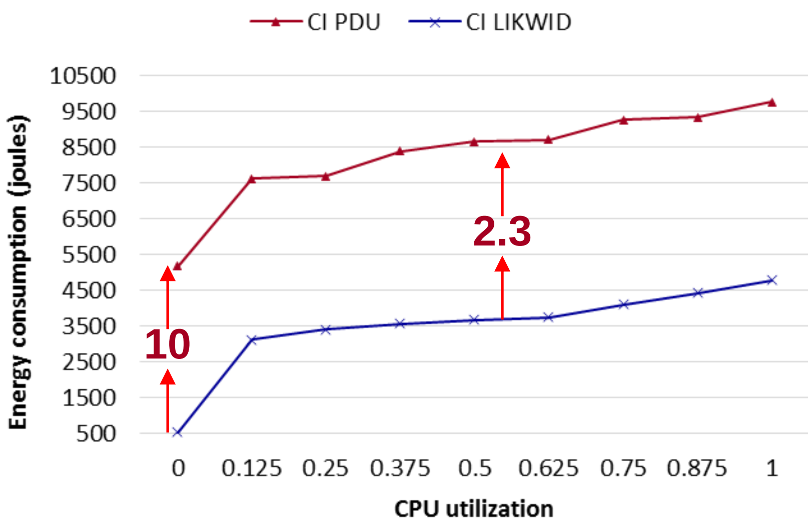


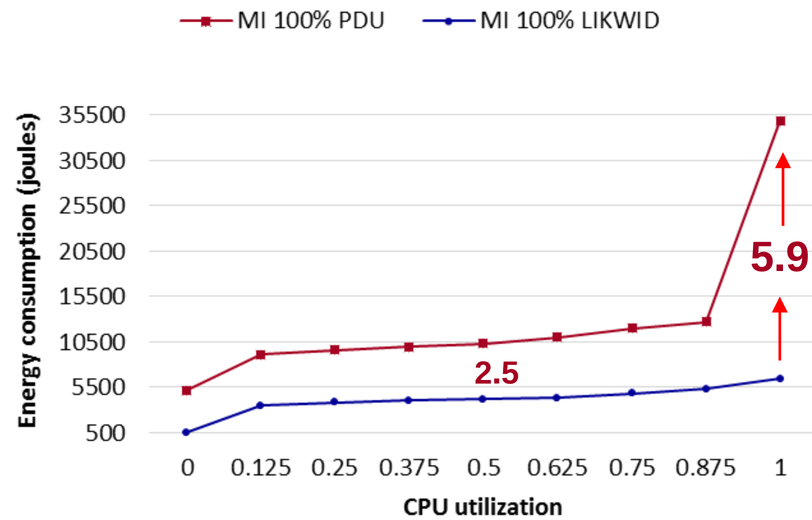
Hypothesis

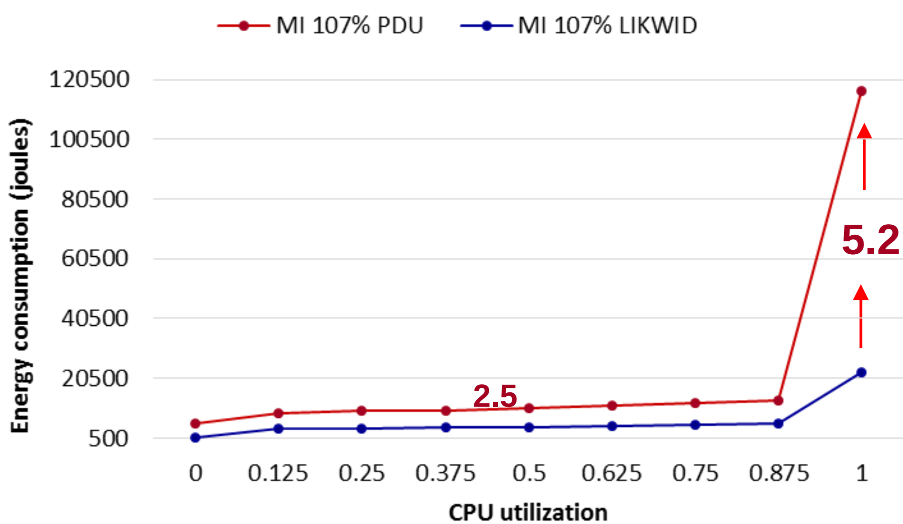




Energy consumption $F(t)$







Strategies and scenarios

Scenarios

Required Capacity (RC) → Three levels

First level: Select containers that has enough CPU capacity to process the arriving job.

Second level: One container, at least, has free CPU capacity

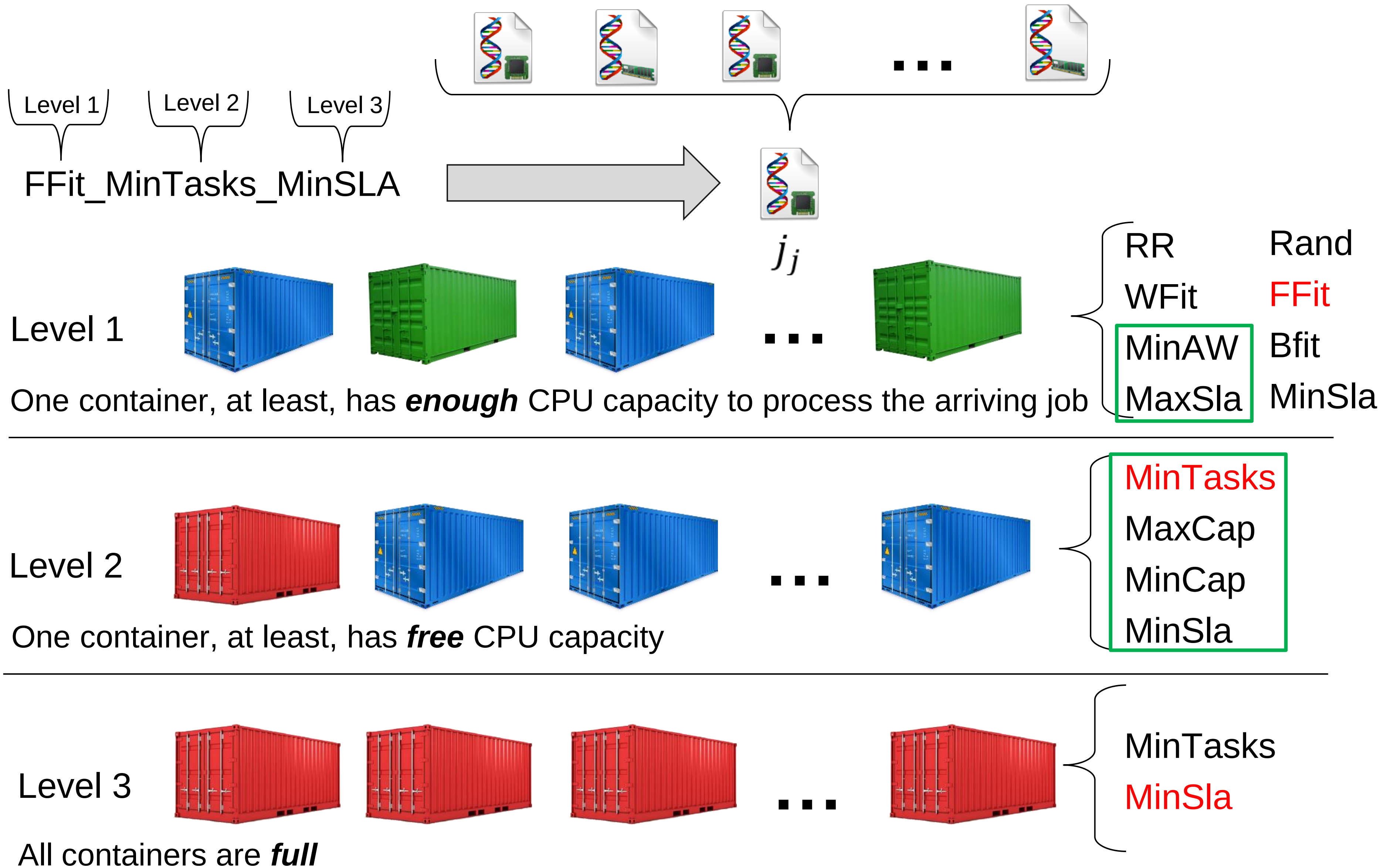
Third level: Any container has no free CPU capacity

Full Capacity (FC) → Two levels

First level: Job assignment does not generate SLA violations

Second level: Job assignment generates SLA violations

Possible assignation levels



Level 1 allocation strategies

Strategy	Description
Random (Rand)	Allocates job j to a random container using a uniform distribution in the range $[1..k]$ $k \leq m$; k is the number of jobs that are running in the container prior to the arrival of the job j
Round Robin (RR)	Allocates job j to the container by Round Robin strategy.
First Fit (FFit)	Assign job j to the first available container and capable of executing it.
Best Fit (BFit)	Allocates job j to the container that minimizes the difference between the container capacity q_i and the sum of previous speed jobs assigned, plus the required speed of the j job $Min\left(q_i - \left(\sum s_d + s_j\right)\right) \geq 0\ \forall_d\ assigned\ to\ c_i.$
Worst Fit (WFit)	Allocates job j to the container that maximizes the difference between the container capacity q_i and the sum of previous speed jobs assigned, plus the required speed of the j job $Max\left(q_i - \left(\sum s_d + s_j\right)\right) \geq 0\ \forall_d\ assigned\ to\ c_i.$
Maximum number of SLA violations (MaxSla)	Assign job j to the container where more SLA violations have occurred.
Container with Minimum Amount Work (MinAW)	Allocates job j to the container with minimum amount work of previously allocated jobs running at time r_j: $\min\left(\sum w_d\right)\ \forall_d\ assigned\ to\ c_i.$

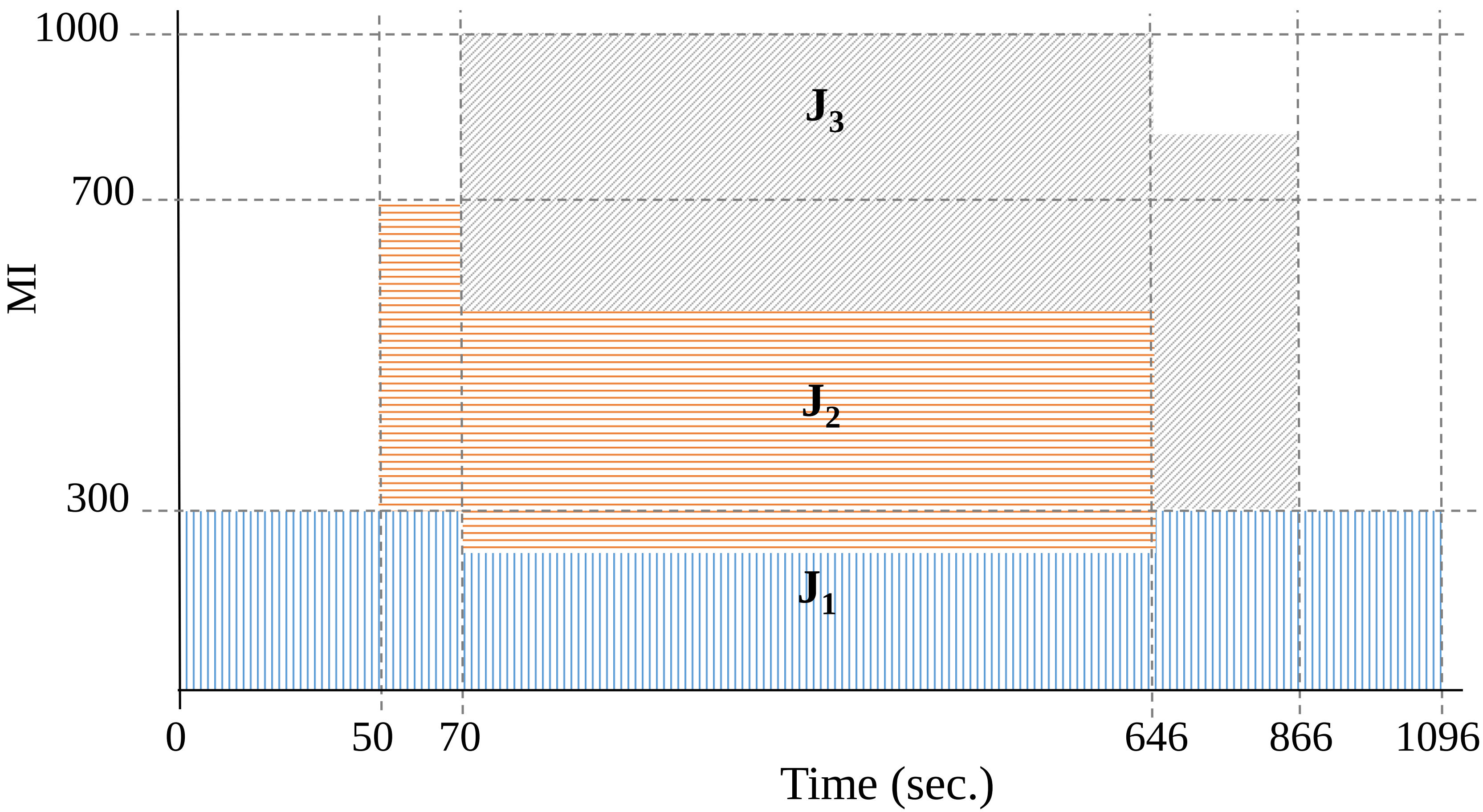
Level 2 allocation strategies

Strategy	Description
MinTasks	Assign job j to the container with the least number of jobs in execution among all those that have available processing capacity.
MaxCap	Assign job j to the container with the largest available processing capacity.
MinCap	Assign job j to the container with minimum available processing capacity.
MinSLA	Assign job j to the container where fewer SLA violations have occurred.

MinTasks and MinSLA are the strategies for the third level.

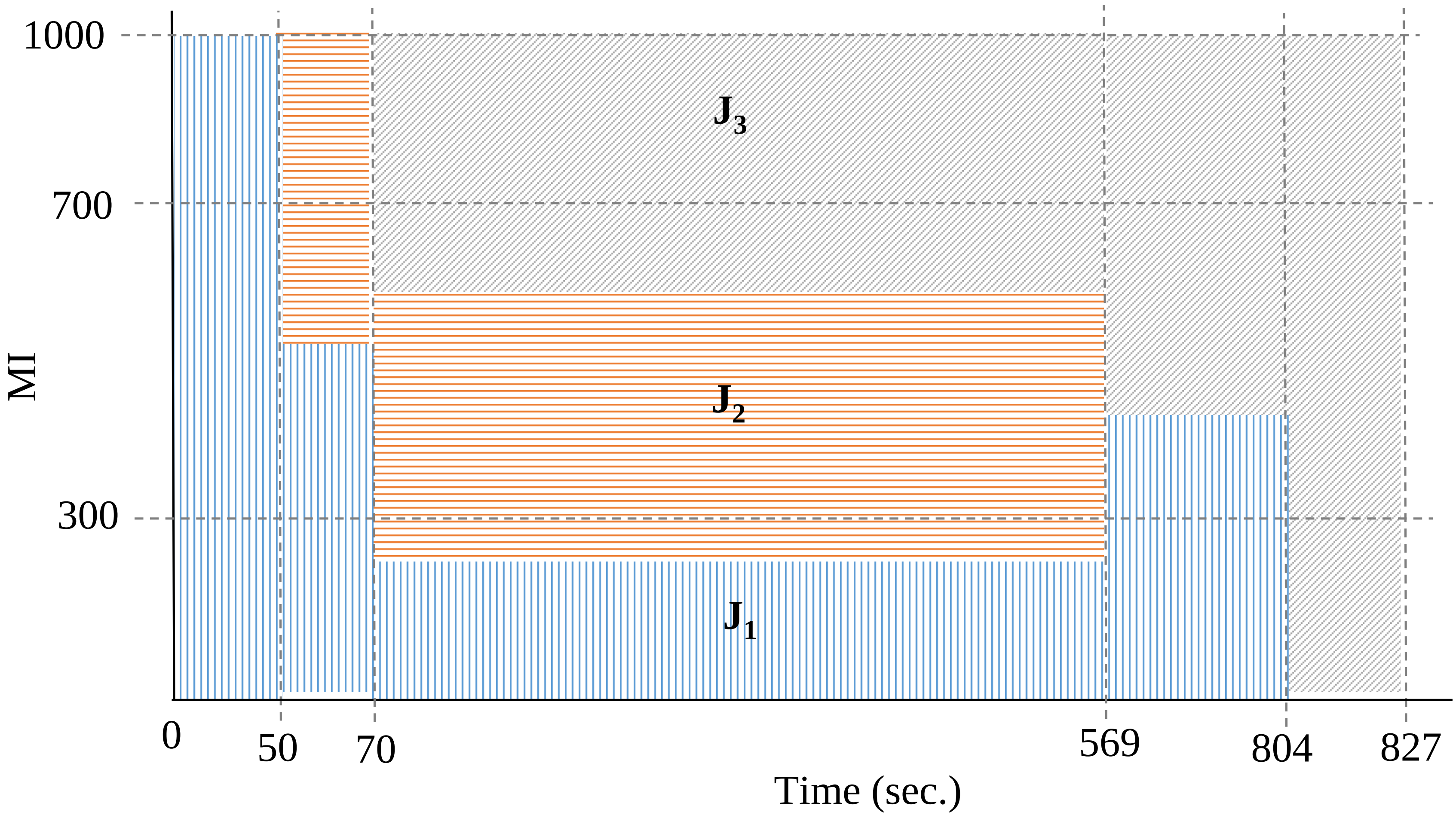
Required Capacity model

$j_1(0, 300, 3 * 10^{14}, CI)$ $j_2(50, 400, 2 * 10^{14}, MI)$ $j_3(70, 500, 3.5 * 10^{14}, CI)$



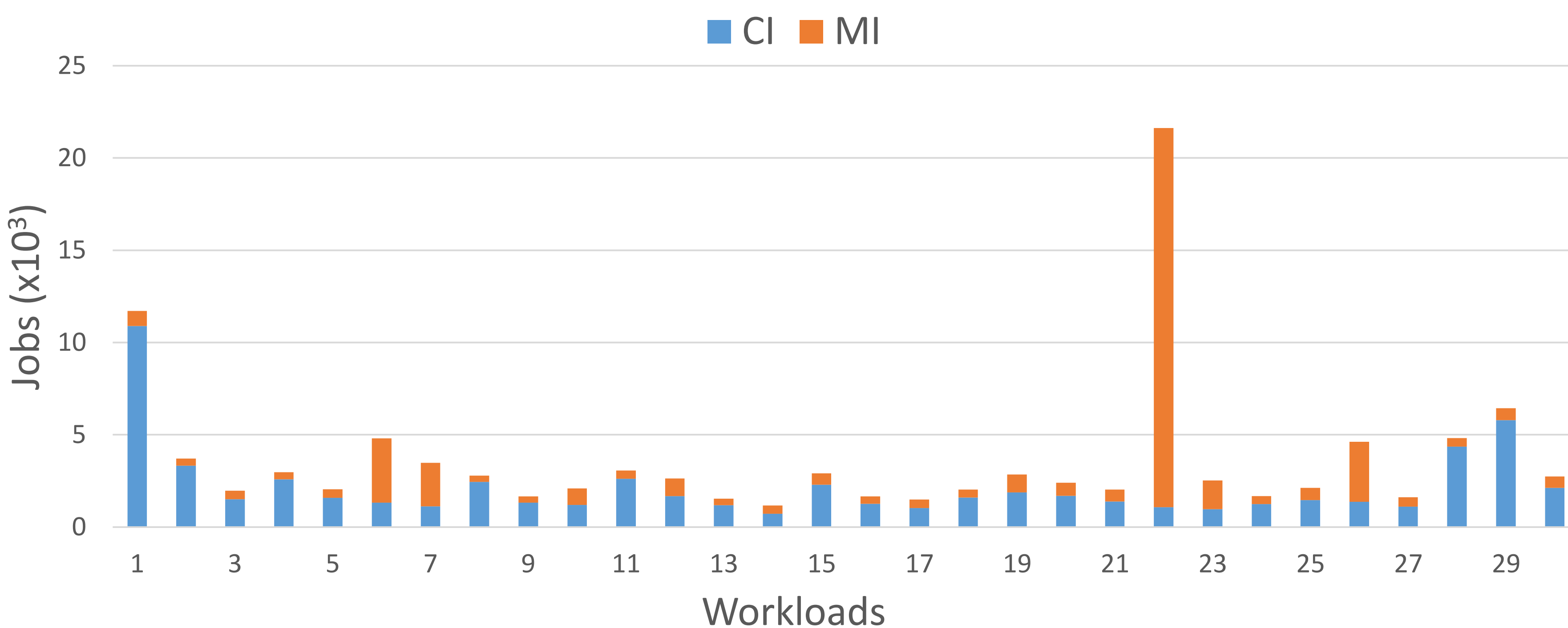
Full Capacity model

$j_1(0, 300, 3 * 10^{14}, CI)$ $j_2(50, 400, 2 * 10^{14}, MI)$ $j_3(70, 500, 3.5 * 10^{14}, CI)$



Workload

- from Parallel Workloads Archive
- Filters remove inconsistent information
- Standard Workload Format (SWF) with two additional fields.
  - Job type (CI, and MI).
  - Minimum speed of job.



Feitelson, D.(2008). Parallel Workloads Archive. from: http://www.cs.huji.ac.il/labs/parallel/workload/

Methodology of analysis

Degradation in performance

The evaluation is done relative to the best performing strategy for the metric:

$$(\gamma - 1) \text{ with } \gamma = \frac{strategy\ metric\ value}{best\ found\ metric\ value}$$

$$strategy\ metric\ value > 0$$

Performance profile

$\delta(\tau)$ is a non-decreasing function that presents the probability that a ratio γ is within a factor τ of the best ratio

$$\delta_{metric} = \frac{|\{i|result_i \leq \sigma\}|}{n_{experiment}}$$

Strategies with large probability $\delta(\tau)$ for small τ are to be preferred.

$result_i$ - degradation of the performance of each solution found.

Tsafrir, D., Etsion, Y., Feitelson, D. G. 2007.Backfilling using system-generated predictions rather than user runtime estimates. IEEE transactions on parallel and distributed systems 18(6): 789-803

Setup

CloudSim

- Modeling and simulation of large-scale Cloud computing environments.
- Extended by our algorithms:
  - ✓ supporting dynamic jobs arrival.
  - ✓ containers deployment.
  - ✓ statistical analysis.
  - ✓ workload processing (jobs with CPU processing variable).

Strategies & workloads

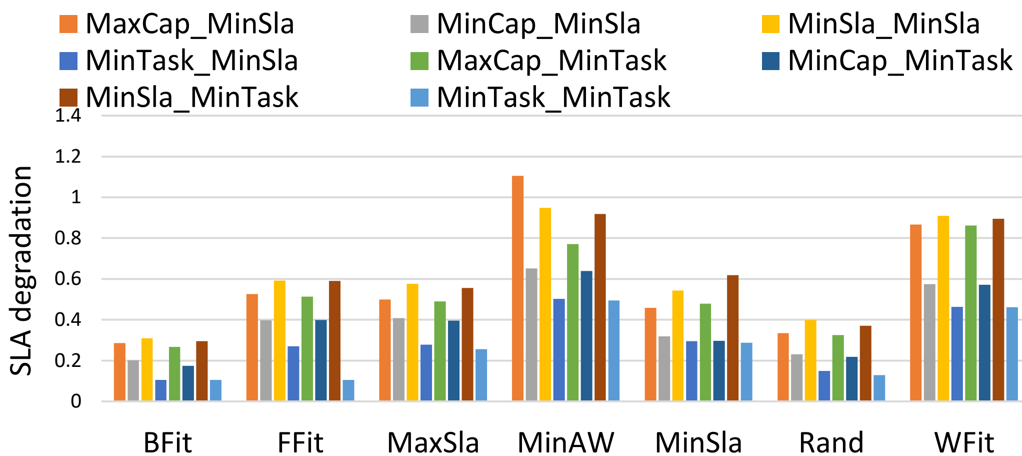
Name	Scenario	
	RC	FC
Strategies:	58	14
1st level	7	7
2nd level	4	2
3rd level	2	-
Workload	30	
Workload length	24 hours	
Number of jobs (n)	109,345	

Resources

Name	Server	CT
Processor (M)	25	50
PES	1	1
MIPS	1,000	500
RAM	1,000	500
e^{idle}	86 W	-
e^{max}	180 W	-
Type	-	OS

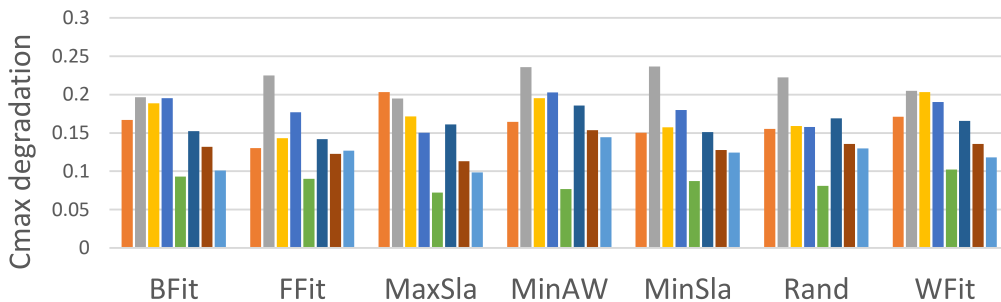
Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., Buyya, R. 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software-practice & experience 41(1): 23-50

RC – Average performance degradations



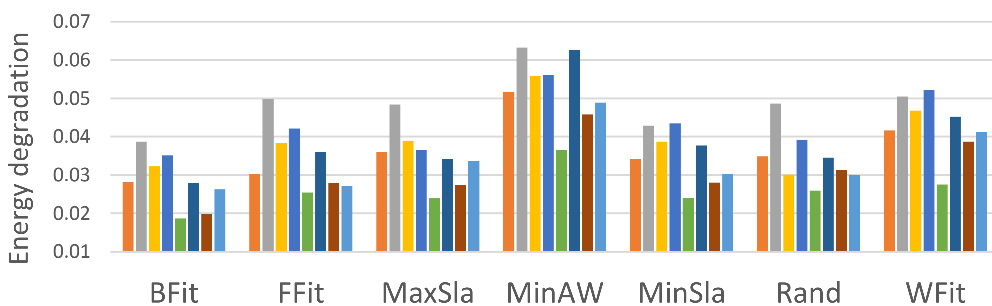
SLA violations

BFit_MinTask_MinSla



C_{max}

MaxSla_MaxCap_MinTask

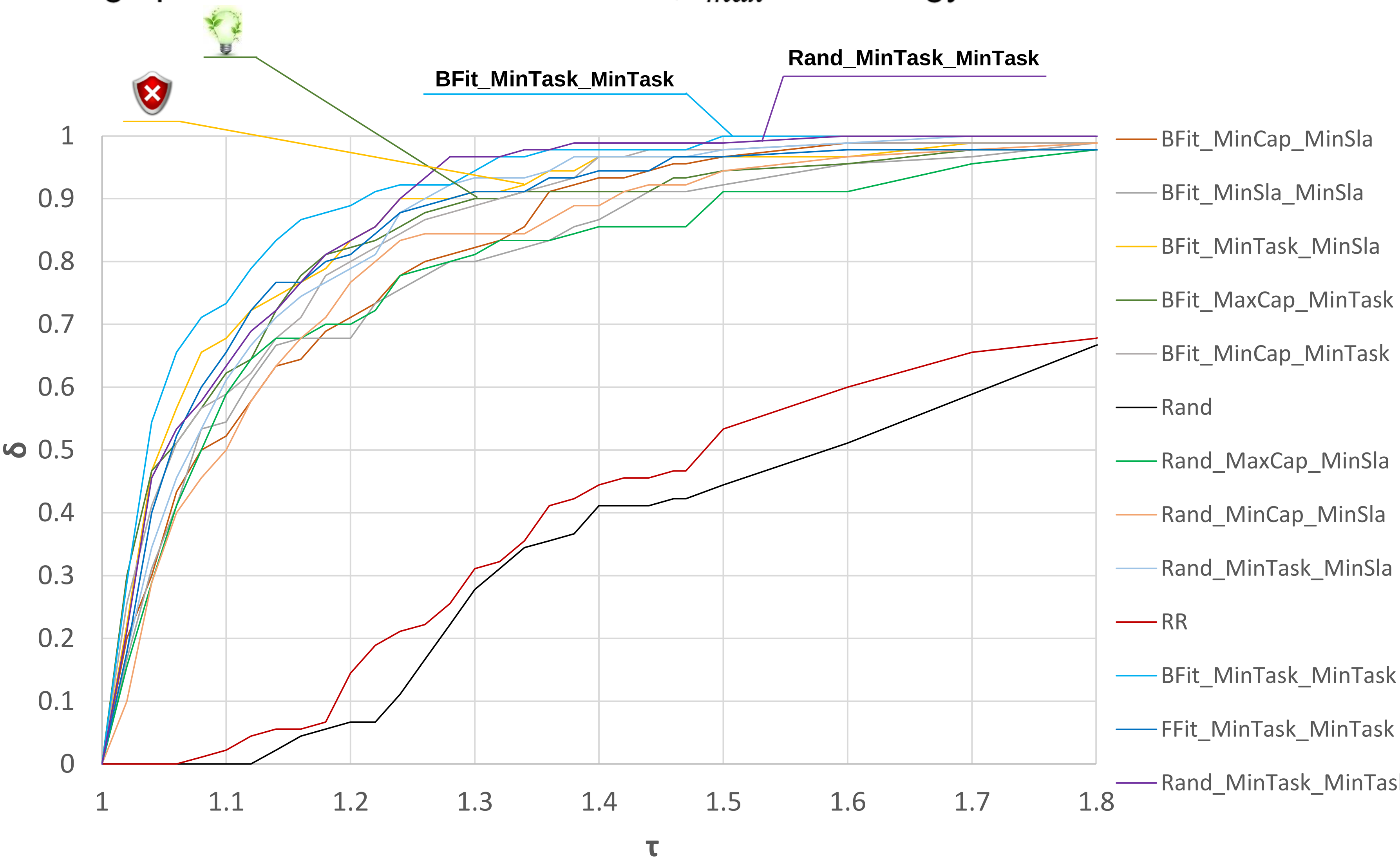


Energy

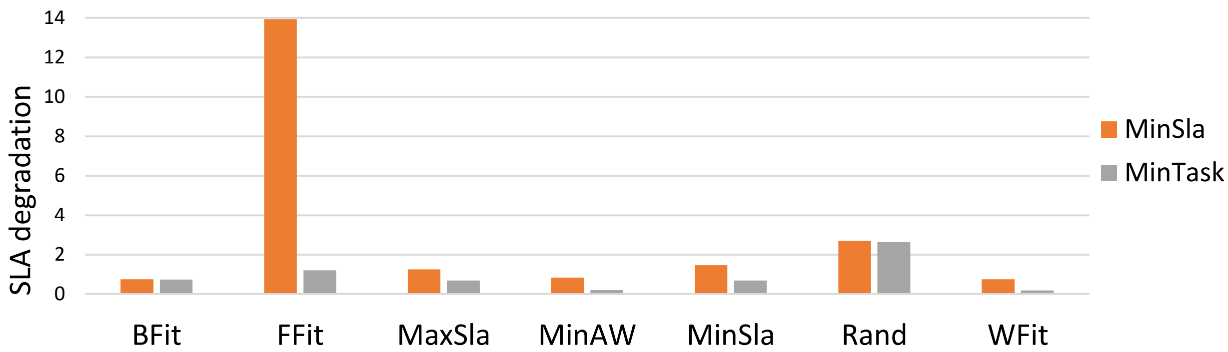
BFit_MaxCap_MinTask

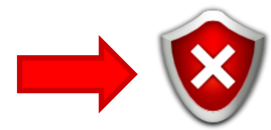
RC - Performance profile

Average performance for SLA violations, C_{max} and Energy



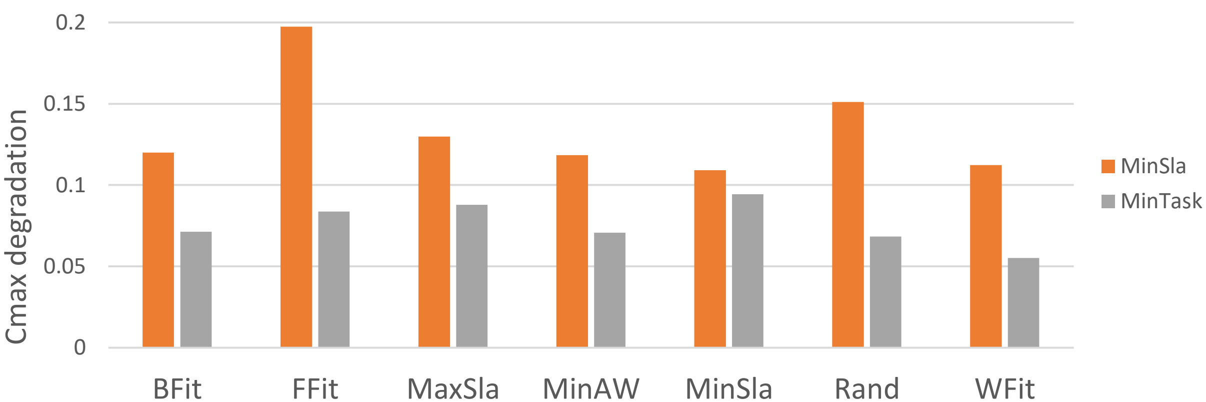
FC - Average Performance degradations

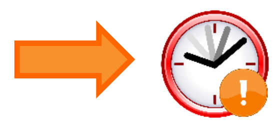




SLA violations

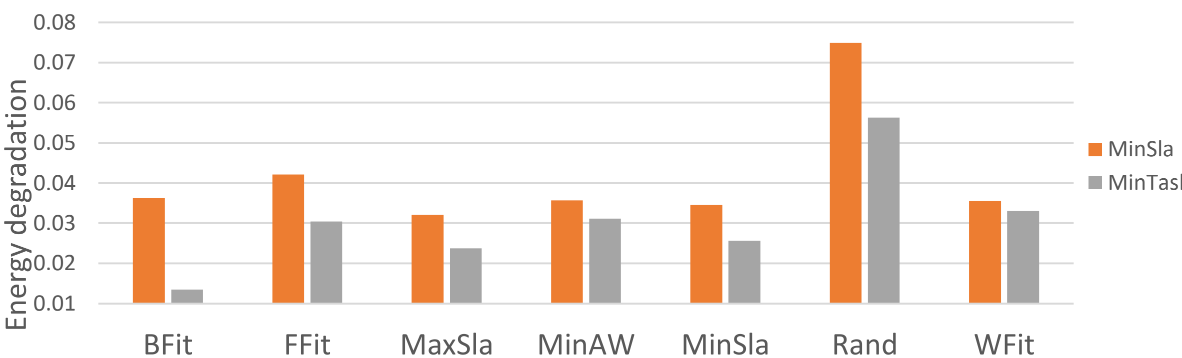
WFit_MinTask





C_{max}

WFit_MinTask

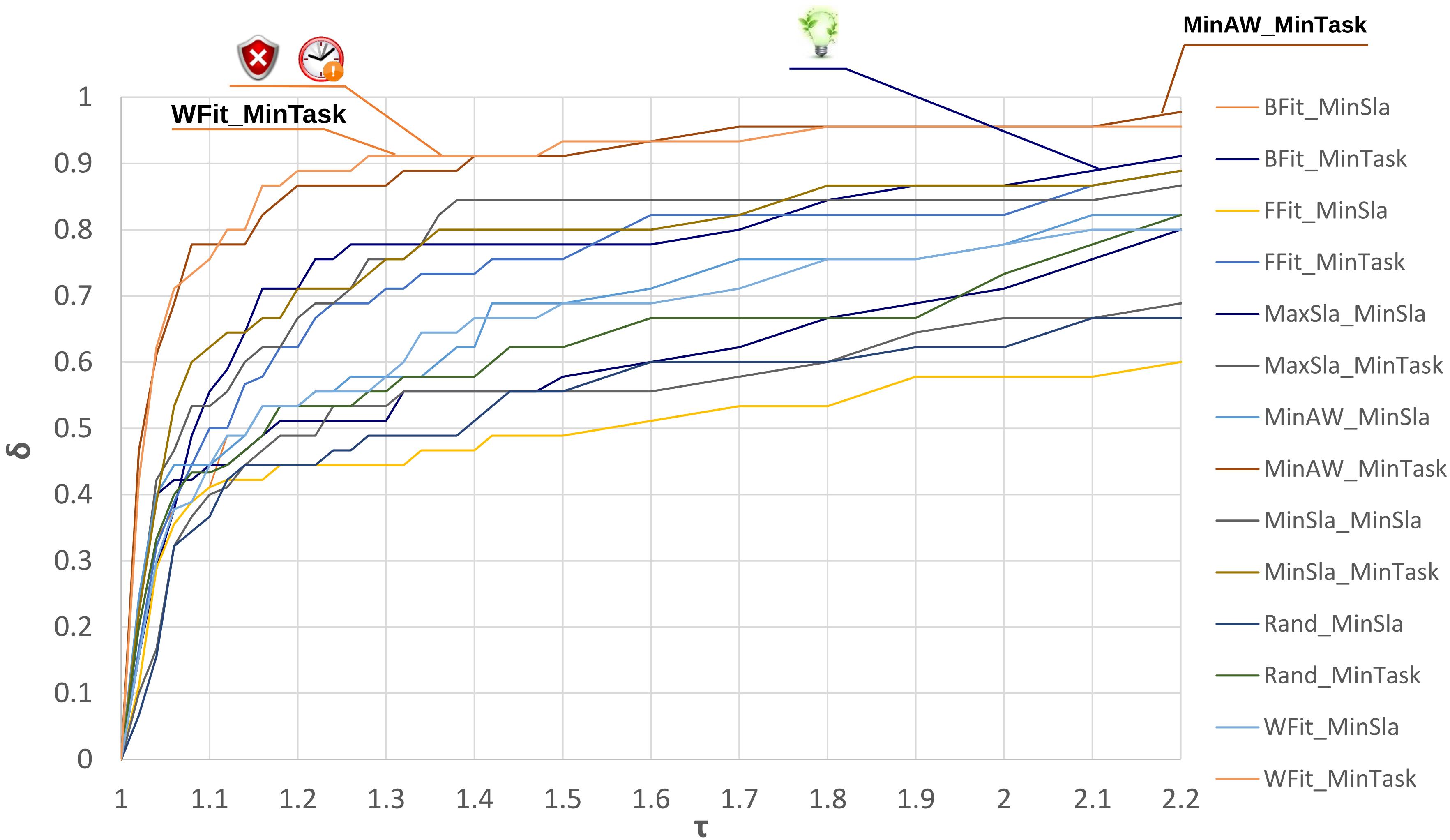


Energy

BFit_MinTask

FC - Performance profile

Average performance for SLA violations, C_{max} and Energy

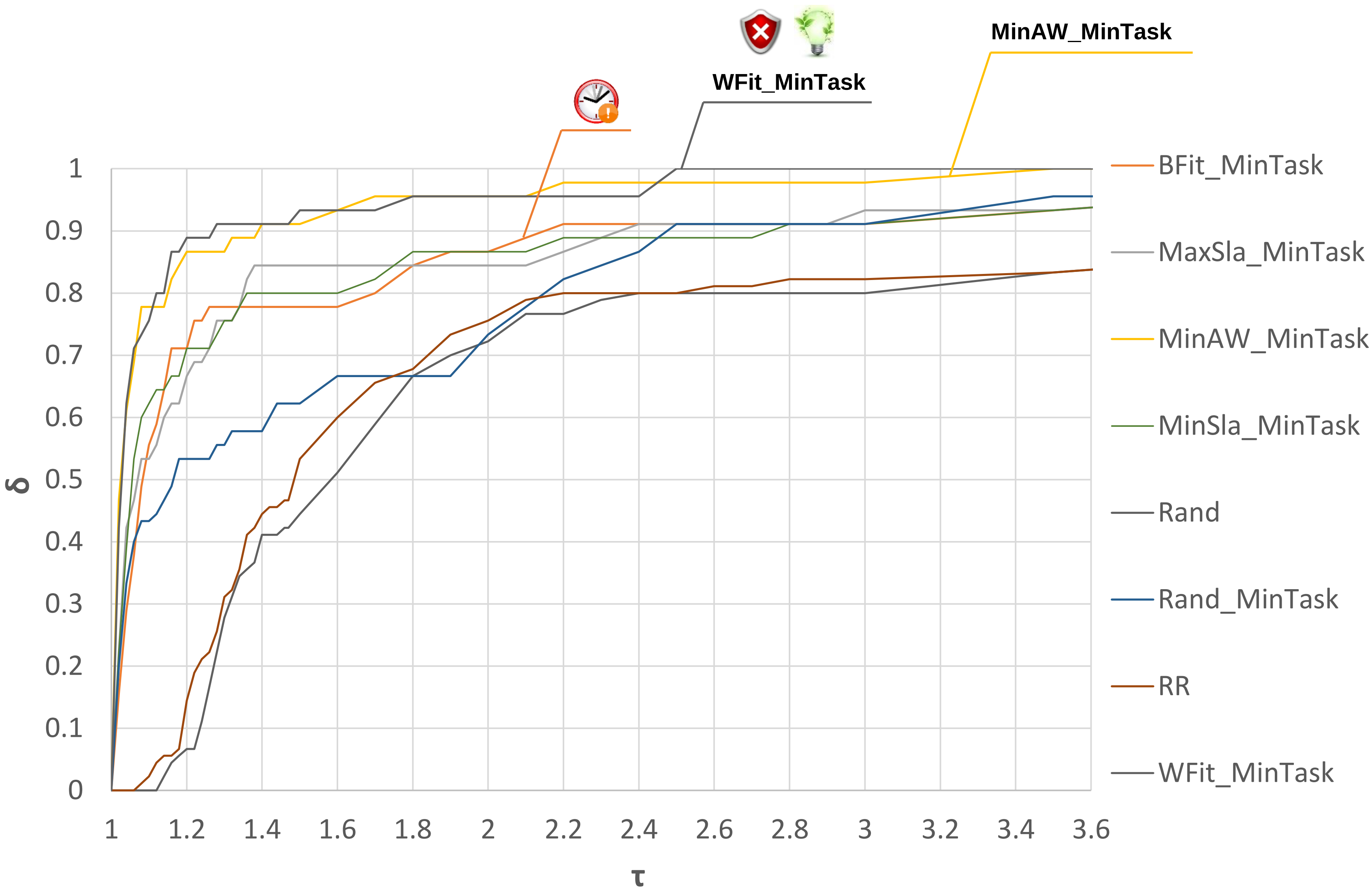


Required Capacity(RC) vs Full Capacity(FC)

6 of the 7 best strategies of the two scenarios belong to Full Capacity

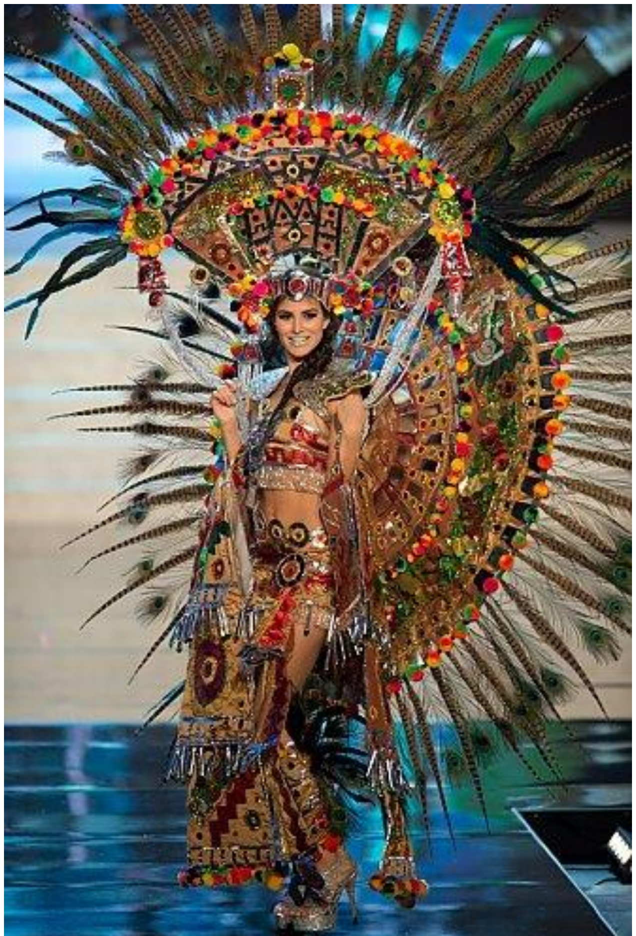
Strategy	Scenario	SLA	c_{max}	Energy	Average	Rank SLA	Rank c_{max}	Rank Energy	Rank Average
BFit_MinTask	FC	0.738	0.025	0.097	0.287	5	**1**	6	5
MaxSla_MaxCap_MinTask	RC	11.752	0.050	0.090	3.965	41	15	**2**	39
MaxSla_MinTask	FC	0.683	0.035	0.114	0.278	**3**	**2**	12	**3**
MinAW_MinTask	FC	0.208	0.043	0.097	0.116	**2**	5	5	**2**
MinSla_MinTask	FC	0.685	0.034	0.121	0.281	4	**3**	15	4
Rand	RC	129.48	0.295	0.827	43.53	72	72	72	72
Rand_MinTask	FC	2.632	0.068	0.094	0.932	12	50	**3**	12
RR	RC	122.29	0.274	0.715	41.09	71	71	71	71
WFit_MinTask	FC	0.185	0.044	0.081	0.104	**1**	7	**1**	**1**

Average performance profile (2 scenarios)



Conclusions

Thanks for your attention!




Orden de la Merced