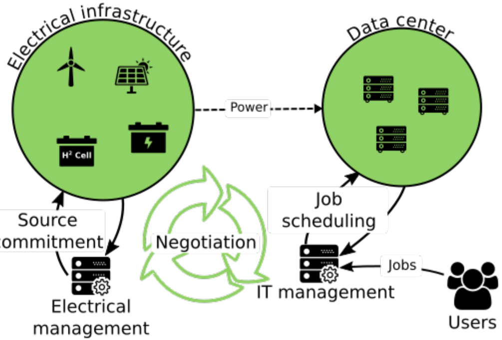
Assessing power needs to run a workload with quality of service constraint on green datacenters

Louis-Claude Canon [1], Damien Landré [1 2], Laurent Philippe [1], Jean-Marc Pierson [2], Paul Renaud-Goud [2]

[1]FEMTO-ST institute, Besançon
[2]IRIT, Toulouse

DATAZERO2: the big picture



Partners:

Funded by

Table of contents

Table of contents

Objectives

Minimize the power demand over a time interval to provide a power profile

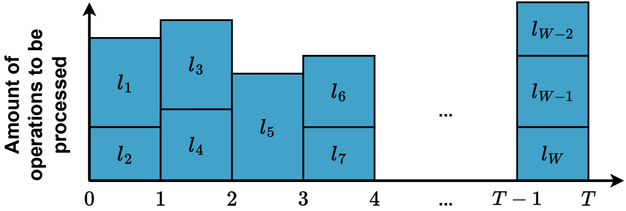
- Constant power value over a time interval
- Power value needs to run the workload with quality of service constraint and to save energy

Model - Workload

Assumption: Workload forecast known

- T time steps $t \in \{1, ..., T\}$
- W load parts l_k. $\forall k \in= \{1, ..., W\}$:
  - Release time r_k
  - Deadline d_k
  - Amount of operations to be processed p_k
- Load part: grouping of loads with the same deadline

Load parts and amount of operations to be processed depending on time



Time steps over a time interval

Model - Machines

- M heterogeneous[1,2] machines $i \in \mathcal{M} = \{1, ..., M\}$
- Idle power $static_i$
- Each type of machine i can have different DVFS states[3] $j \in \mathcal{S}^{(i)}$.
  $\forall i \in \mathcal{M}, \forall j \in \mathcal{S}^{(i)}$:
  - $g^{(i)}_{max_j}$: maximum computing power (operations per second)
  - $0 \leq g^{(i)} \leq g^{(i)}_{max_j}$: computing power (operations per second)
  - $power^{(i)}_j$: consumed power per operations per second
- $P_i = static_i + g^{(i)} \times power^{(i)}_j$

---

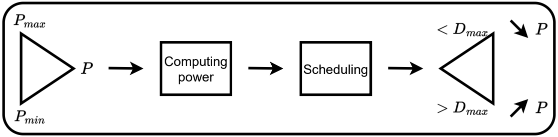
[1] Wang, Li, and Liang, "Dominant resource fairness in cloud computing systems with heterogeneous servers".

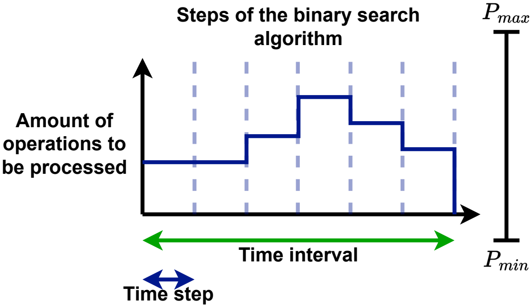
[2] Reiss et al., "Heterogeneity and dynamicity of clouds at scale: Google trace analysis".

[3] Arjona Aroca et al., "A measurement-based analysis of the energy consumption of data center servers".

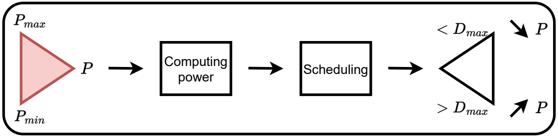
Table of contents

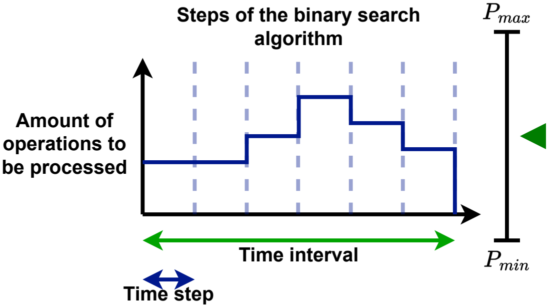
Binary Search Algorithm



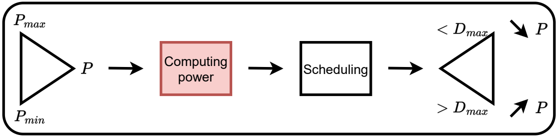


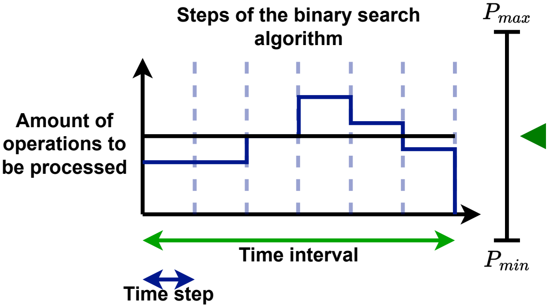
Binary Search Algorithm



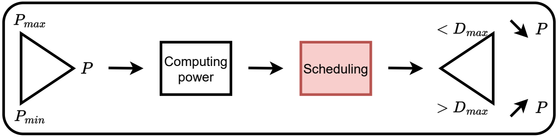


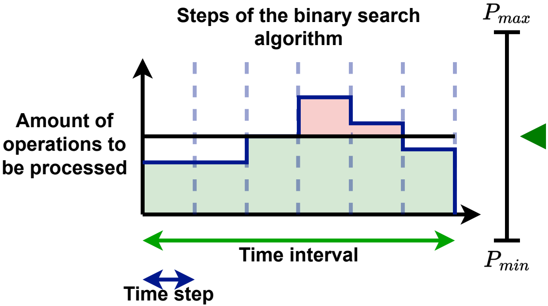
Binary Search Algorithm



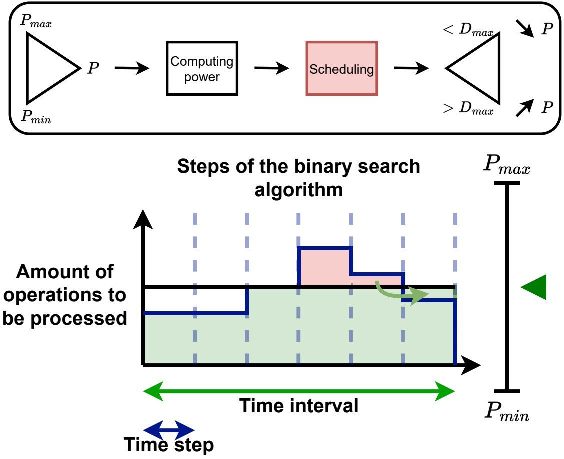


Binary Search Algorithm

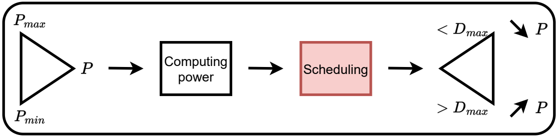


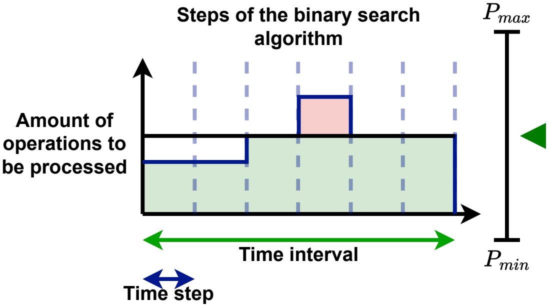


Binary Search Algorithm

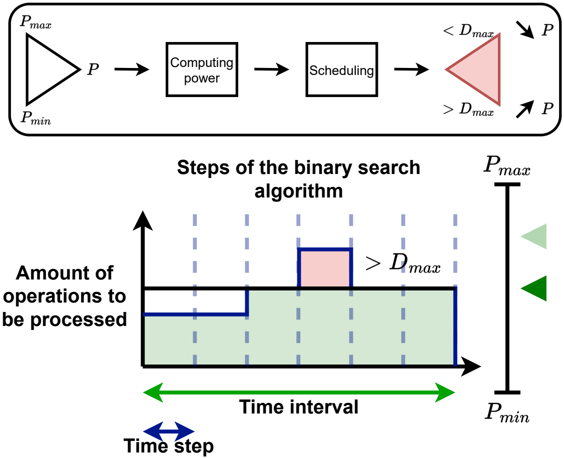


Binary Search Algorithm

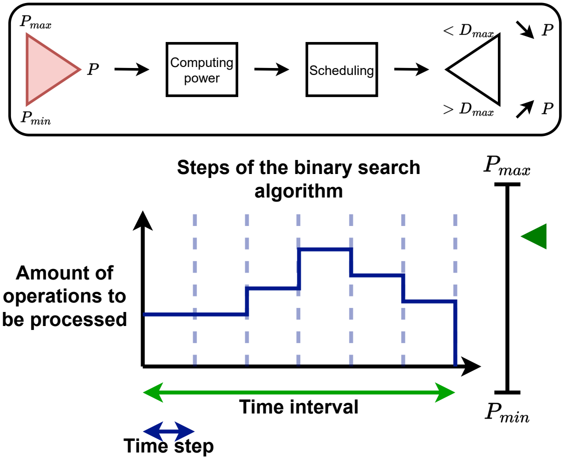




Binary Search Algorithm



Binary Search Algorithm



Binary Search Algorithm

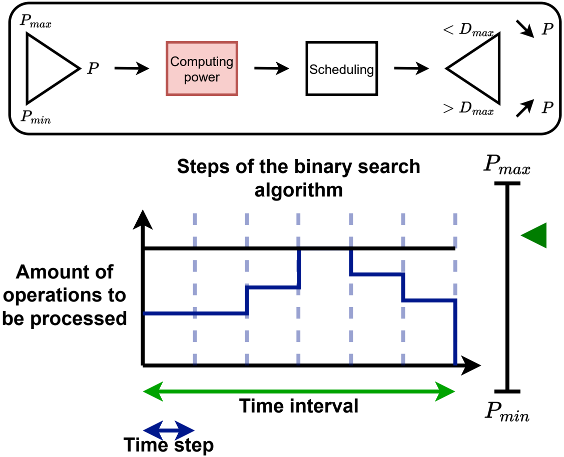


Table of contents

MILP

MILP: Maximize computing power under the constraint of a power value

- Objective : Maximize $\sum_{i=1}^{M} g^{(i)}$
- $\forall i \in \mathcal{M}$: $\sum_{j=0}^{S^{(i)}} x_{i,j} = 1$ with $x_{i,j} \in \{0, 1\}$: For each machine, determine a DVFS state
- $\forall i \in \mathcal{M}$: $0 \leq g^{(i)} \leq \sum_{j=0}^{S^{(i)}} (x_{i,j} \times g^{(i)}_{max_j})$: For each machine, determine the computing power
- $\forall i \in \mathcal{M}$: $P_i = x_{i,0} \times 0 + \sum_{j=1}^{S^{(i)}} (idle_i \times x_{i,j} + g^{(i)} \times power_j^{(i)})$: For each machine, determine the power consumption
- $\sum_{i=1}^{M} P_i \leq p$: The total power consumption of the machines must not exceed P

Simulation data

Table: Machine types[4] and their characteristics[5]

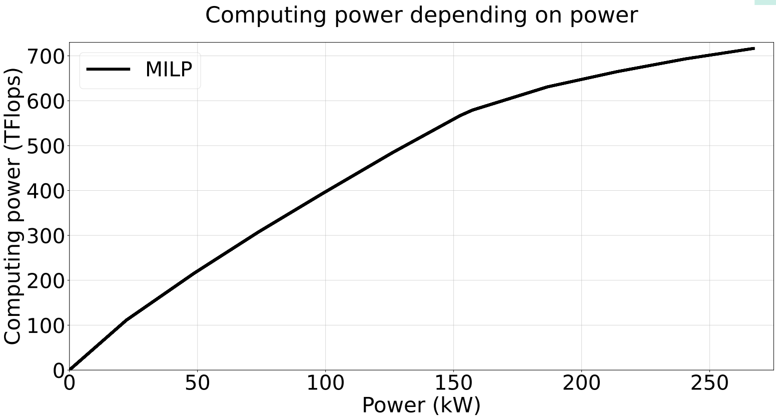
Machine name	No.	No. of states	Static power (W)	$\max(g_{max_j})$ (GFlops)	$\max(power_j)$ (W/GFlops)
Taurus	150	13	93.0	220.80	0.38
Parasilo	169	12	94.1	614.40	0.12
Graoully	145	14	98.2	614.40	0.15
Gros	195	14	62.9	633.60	0.12
Grimoire	134	14	121.2	614.40	0.13
Chifflet	89	14	198.7	1075.20	0.09
Grele	106	12	163.2	844.80	0.11
Gemini	34	12	740.8	1408.00	0.09
Graphite	91	10	226	256.00	0.26
Orion	128	13	121.2	220.80	0.39

[4] *Clusters Grid5000.*

[5] *Energy saving in large scale distributed platforms – Energumen.*

MILP: results

Average runtime: 2.83 s

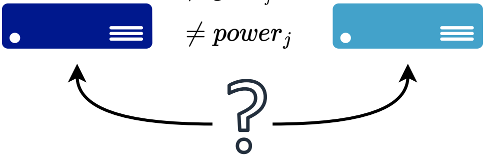


RC heuristic

Random Choice (RC):

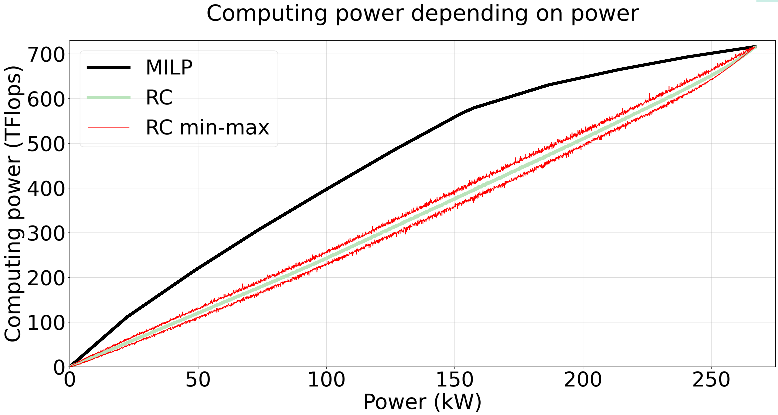
- Randomly choose the type of machine to switch-on
- Allocate the power needed to provide the maximum computing power

$$\neq static$$
$$\neq g_{max_j}$$
$$\neq power_j$$



RC heuristic: results

Average runtime: 1.15 ms (∼2400x)



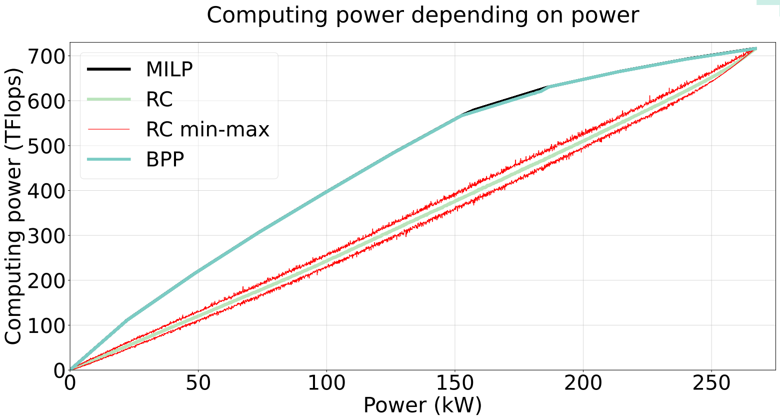
BPP heuristic

Balance Power-Performance (BPP):

- Score computed for each machine and each DVFS state depending on $0 \leq \alpha \leq 1$
- $g^{(i)}$: Computing power of the machine i, $\max(g^{(i)})$ max computing power computed to normalize : $0 \leq \frac{g^{(i)}}{\max(g^{(i)})} \leq 1$
- $ratio_{(i,j)}$: Ratio $(P_i/g^{(i)})$. $\min(ratio_{i,j})$: Ratio min computed to normalize : $0 \leq \frac{\min(ratio_{i,j})}{ratio_{i,j}} \leq 1$
- $score_{i,j} = \alpha \times (\frac{g^{(i)}}{\max(g^{(i)})}) + (1 - \alpha) \times (\frac{\min(ratio_{i,j})}{ratio_{i,j}})$ with $0 \leq score_{i,j} \leq 1$
- The highest score determines the type of machine to be switched-on and its DVFS state
- Several alpha values are checked in order to produce different configurations
- Keep the best configuration

BPP heuristic: results

Average runtime: 9.04 ms (∼300x)



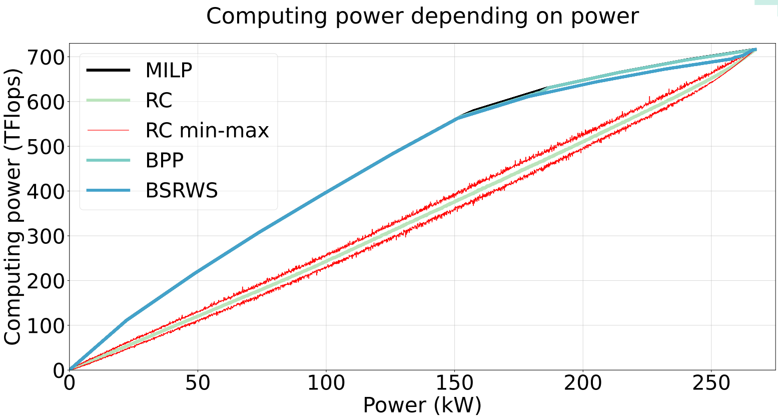
BSRWS heuristic

Best State Redistribute Without Static (BSRWS):

- Ratio performance computed (with static power) ($P_i/g^{(i)}$) for each type of machine and each DVFS state
- Machine with the best ratio performance is switched-on
- Remaining power is redistributed to the machines switched-on

BSRWS heuristic: results

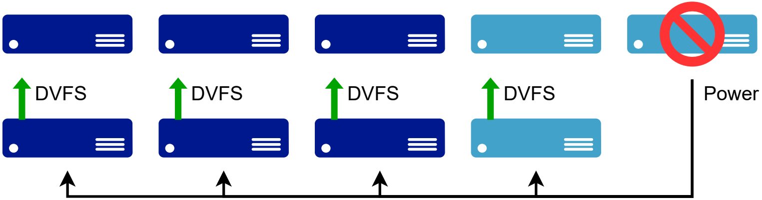
Average runtime: 0.98 ms (∼2900x)



BSRWSAR heuristic

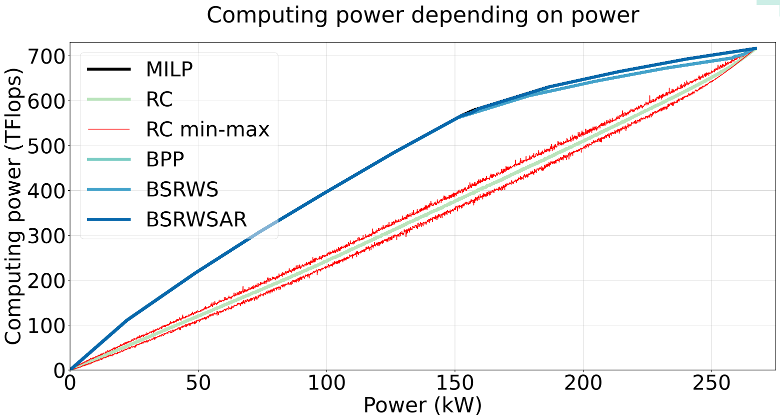
Best State Redistribute Without Static And Removing (BSRWSAR):

- BSRWS + Re-running the heuristic with one less machine available, then 2 less, then 3 less, etc...
- Keep the best configuration



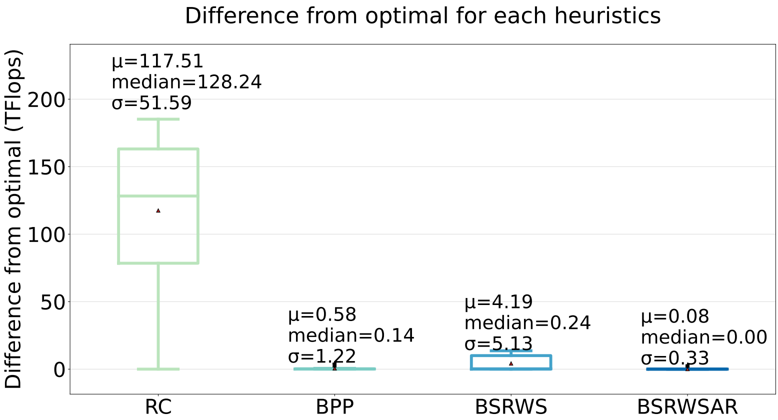
BSRWSAR heuristic: results

Average runtime: 0.25 s (∼10x)

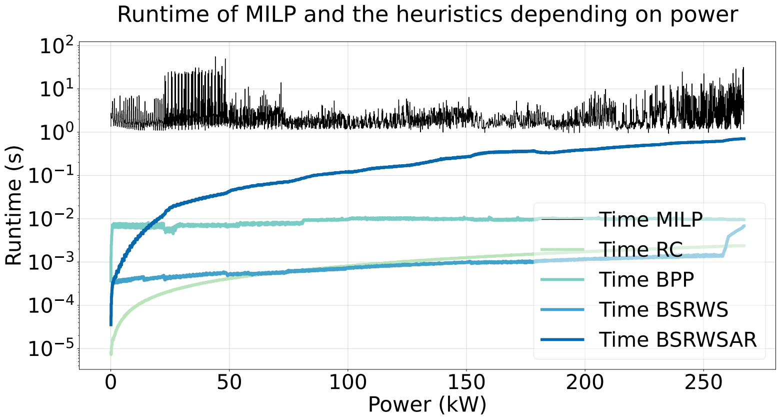


Heuristics: results

max. deviation: RC 52.43% BPP 3.15% BSRWS 3.15% BSRWSAR 2.04%
avg. deviation: RC 31.84% BPP 0.12% BSRWS 0.65% BSRWSAR 0.03%



Runtimes



Runtime of MILP and the heuristics depending on power
Runtime (s)
Power (kW)
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
0
50
100
150
200
250
Time MILP
Time RC
Time BPP
Time BSRWS
Time BSRWSAR

Table of contents

Conclusion & Perspectives

Conclusion

- Binary search algorithm
- MILP and 3 non-trivial heuristics that computes the maximum processing power in the heterogeneous case.
- Heuristics: average deviation from optimal solution of 0.12% (BPP), 0.65% (BSRWS) and 0.03% (BSRWSAR).

Perspectives

- Improve the execution time of the MILP from a solution given by a heuristic.
- Consider different server energy profiles depending on the workload.
- Consider other energy costs (memory, networks...).

Thank you for your attention

Arjona Aroca, Jordi et al. “A measurement-based analysis of the energy consumption of data center servers”. In: *Proceedings of the 5th international conference on Future energy systems*. 2014, pp. 63–74.

Clusters Grid5000. https://www.grid5000.fr/. [Online; accessed 21-October-2022]. 2022.

Energy saving in large scale distributed platforms – Energumen. https://anr.fr/Project-ANR-18-CE25-0008. [Online; accessed 20-November-2022]. 2018.

Reiss, Charles et al. “Heterogeneity and dynamicity of clouds at scale: Google trace analysis”. In: *Proceedings of the third ACM symposium on cloud computing*. 2012, pp. 1–13.

Wang, Wei, Baochun Li, and Ben Liang. “Dominant resource fairness in cloud computing systems with heterogeneous servers”. In: *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE. 2014, pp. 583–591.