Analyzing Power Decisions in Data Center Powered by Renewable Sources

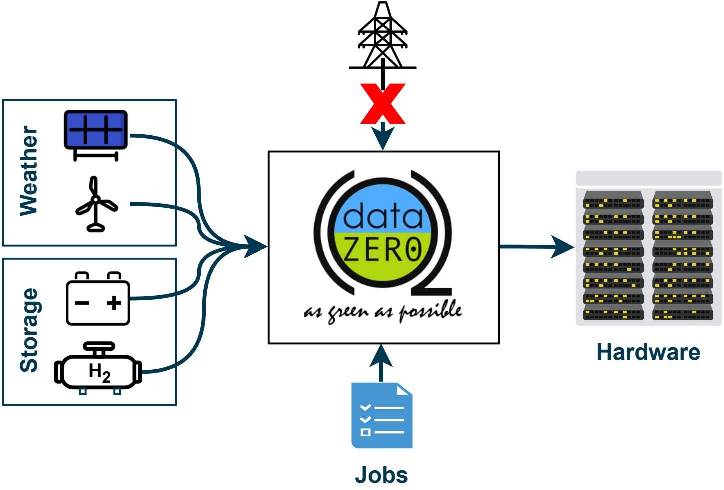
Igor Fontana de Nardin, Patricia Stolf, and Stephane Caux
LAPLACE, IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3

Green Days 2023 @ Lyon

Contents

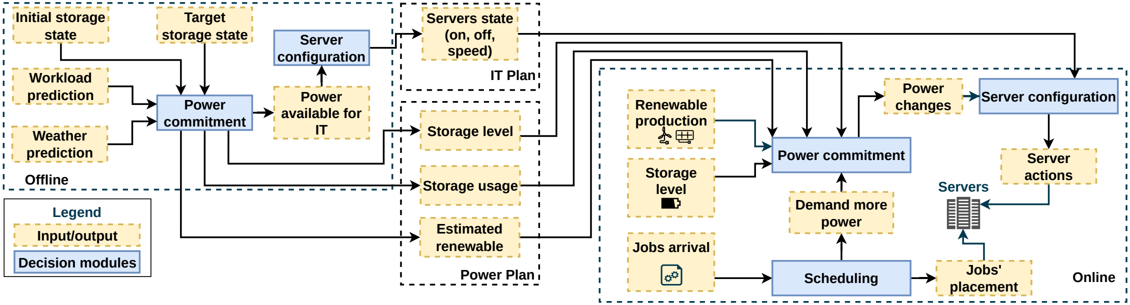
Introduction

Introduction

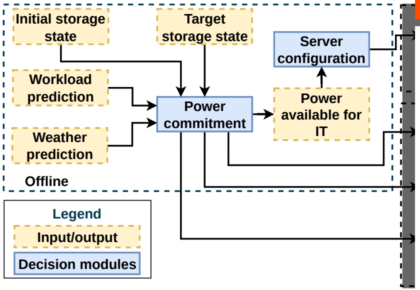


Problem statement

Problem statement

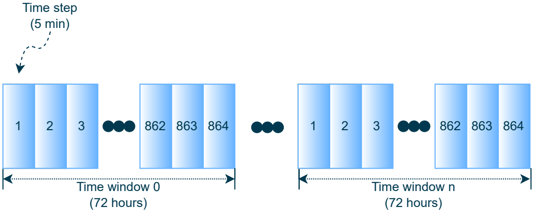


Problem statement

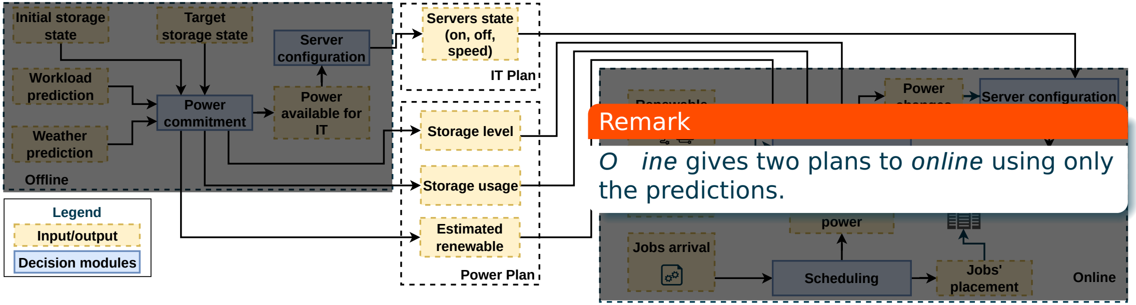


Remark

The *o ine* uses two predictions (workload and weather) and two constraints (initial and target storage level) to decide the actions for the next 3-day time window.



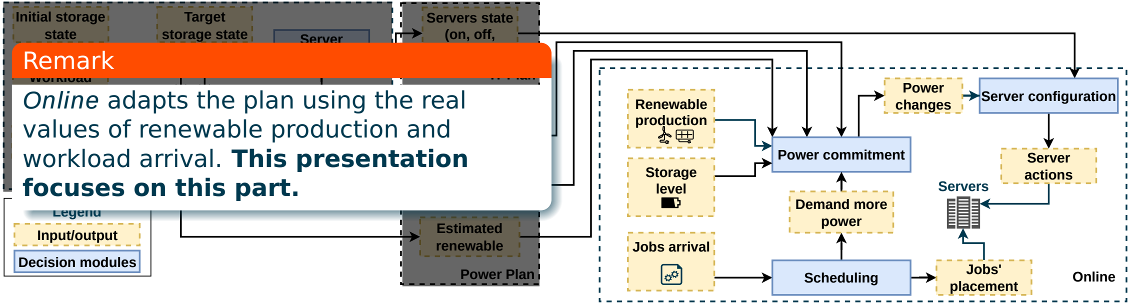
Problem statement



Remark

Offline gives two plans to *online* using only the predictions.

Problem statement



Remark

Online adapts the plan using the real values of renewable production and workload arrival. **This presentation focuses on this part.**

Model

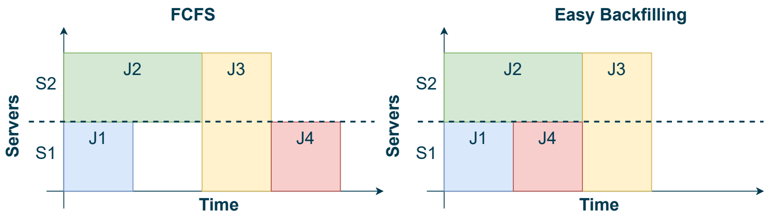
Online scheduling

The main goals of online scheduling are to **place the jobs** on servers and **avoid killing jobs**.

Online scheduling

The main goals of online scheduling are to ***place the jobs*** *on servers and avoid killing jobs.*

We have implemented a well-known algorithm named Easy-Backfilling.



Online scheduling

The main goals of online scheduling are to ***place the jobs*** *on servers and avoid killing jobs.*

We have sorted the waiting queue by the slowdown.

$$slow_j = \frac{wait_j + wall_j}{wall_j} \quad (1)$$

Where:

- $slow_j$ is the slowdown;
- $wait_j$ is the waiting time;
- $wall_j$ is the walltime. Walltime is the maximum execution time of a job given by the user;

Online scheduling

The main goals of online scheduling are to place the jobs on servers and ***avoid killing jobs****.*

A job is killed in one of below situations:

1. Server goes to sleep. In this case, we try to use more battery than planned to maintain the server running;
2. Walltime is reached. Trying to avoid this, we estimate the total work to do and try to maintain the speed $Flops_{s;d}$ from the equation:

$$(wall_j - elapTime_j) \times Flops_{s;d} \geqslant jobFlop_j - elapFlop_j \tag{2}$$

Where:

- $elapTime_j$ is elapsed time;
- $Flops_{s;d}$ is speed of the server in flop per second;
- $jobFlop_j$ is an estimate of the FLOP to run;
- $elapFlop_j$ is how much the jobs have run already;

Online power commitment

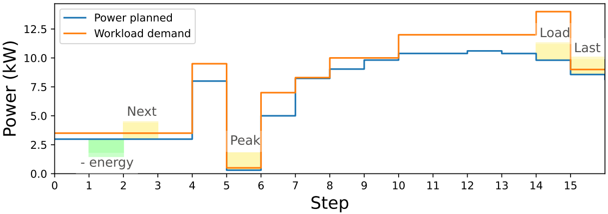
Workload and weather real values can vary from the estimations.

We consider three types of variations:

1. Renewable production;
2. Server idleness;
3. Scheduling changes.

Online power commitment

So, we propose four policies of compensation to deal with these variations.



The main goal is to have the storage level as close to the plan as possible at the end of the time window.

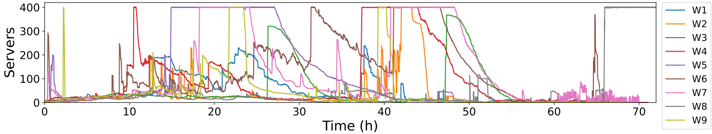
Experiments

Experimental Environment

Laplace

Workload

We have taken nine different workloads (3 days each) from Metracentrum trace[a].



[a] Dalibor Klusáček, Šimon Tóth, and Gabriela Podolníková. "Real-life experience with major reconfiguration of job scheduling system". In: *Job scheduling strategies for parallel processing*. Springer. 2015, pp. 83–101.

Experimental Environment

Data center

We have simulated a platform from Grid5000 [a] using 400 servers with eight different types:

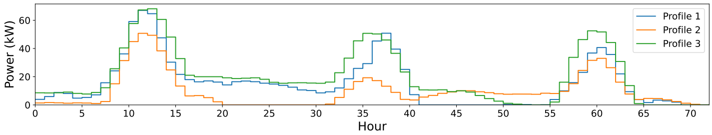
- Dahu;
- Grvingt;
- Parasilo;
- Chifflet;
- Grisou;
- Chetemi;
- Gros;
- Graffiti.

[a] https://www.grid5000.fr

Experimental Environment

Renewable production

We have used three power profiles collected from the Renewable ninja website [a].



We have created two scenarios:

- Offline uses profile 1 and online uses profile 2 (less energy).
- Offline uses profile 1 and online uses profile 3 (more energy).

[a] https://www.renewables.ninja/

Experimental Environment

Simulator

We have simulated this environment in the BATSIM simulator[a], which operates using the SIMGRID framework[b].

[a] Pierre-François Dutot et al. "Batsim: a realistic language-independent resources and jobs management systems simulator". In: *Job Scheduling Strategies for Parallel Processing*. Springer. 2015, pp. 178–197.

[b] Henri Casanova. "Simgrid: A toolkit for the simulation of application scheduling". In: *Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid*. IEEE. 2001, pp. 430–437.

Experimental Environment

We have compared our four policies (*Peak*, *Next*, *Last*, and *Load*) with:

- Baseline: This execution applies the offline plan with no changes;
- Power reactive: This execution configures the servers according to the renewable power available;
- Workload reactive: This execution places any incoming job on a server. It uses a Dynamic power management (DPM) technique[1] to define the moment to turn off the servers.

[1] Issam Raïs et al. "Quantifying the impact of shutdown techniques for energy-efficient data centers". In: *Concurrency and Computation: Practice and Experience* 30.17 (2018), e4471.

Results

Pro le 2 (less energy)

Hydrogen (Target 300 kg)

Battery (Target 50%)

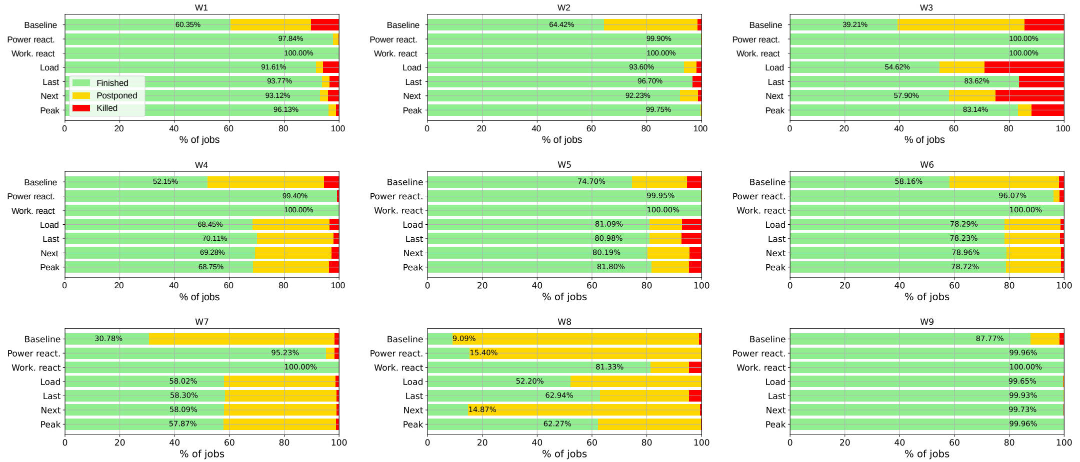
Pro le 2 (less energy)

Pro le 3 (more energy)

Hydrogen (Target 300 kg)

Battery (Target 50%)

Profile 3 (more energy)



Conclusion

Conclusion

Just following an offline plan may not be sufficient to deal with all elements of a renewable-only data center.

This work presented a model for online adaptations to change an offline plan, aiming to improve jobs finished and deal with power fluctuations.

Future works:

1. Create a reinforcement learning algorithm to learn which policy to use in each case. **Well, we have tried, and it is not so good**;
2. Create a heuristic mixing prediction and scheduling.

Merci ! Thank you! Obrigado!

https://www.irit.fr/datazero/
igor.fdn@gmail.com

Acknowledgment

This work was partly supported by the French Research Agency under the project Datazero 2 (ANR-19-CE25-0016)