HAL
open science

Mixing Offline and Online Electrical Decisions in Data Centers Powered by Renewable Sources

Igor Fontana de Nardin, Patricia Stolf, Stéphane Caux

To cite this version:

Igor Fontana de Nardin, Patricia Stolf, Stéphane Caux. Mixing Offline and Online Electrical Decisions in Data Centers Powered by Renewable Sources. 48th Annual Conference of the IEEE Industrial Electronics Society (IECON 2022), IEEE Industrial Electronics Society (IES), Oct 2022, Brussels, Belgium. pp.1-6, 10.1109/IECON49645.2022.9968999 . hal-03841611

HAL Id: hal-03841611

https://hal.science/hal-03841611v1

Submitted on 14 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Mixing Offline and Online Electrical Decisions in Data Centers Powered by Renewable Sources

Igor Fontana de Nardin
Laplace UMR5213, IRIT,
Université de Toulouse, CNRS, Toulouse INP
Toulouse, France
igor.fontana@irit.fr

Patricia Stolf
IRIT, Université de Toulouse,
CNRS, Toulouse INP, UT3
Toulouse, France
patricia.stolf@irit.fr

Stephane Caux
Laplace UMR5213,
Université de Toulouse
Toulouse, France
stephane.caux@laplace.univ-tlse.fr

***Abstract*—Since the Paris Agreement, academics and industry have dedicated efforts to reducing Information and Communications Technology (ICT) greenhouse gas (GHG) emissions. Data centers play a key role in ICT electricity consumption since they are built to run 24/7. One way to reduce these emissions is to switch from brown energy to green energy from renewable sources (RES). However, RES introduce several uncertainties due to their intermittence. This work is part of the Datazero2 Project. This project aims to design a data center powered only by RES production and storage elements. Datazero2's clean-by-design data center requires several decisions at different levels of management. Usually, research works focus on offline decisions (e.g., decisions for the next three days) or online decisions (e.g., decisions in real-time). This work proposes a mix of offline and online decisions producing a reliable and change-aware solution. While the offline focuses on the long-term renewable usage plan, online adapts it according to the power fluctuations. The results demonstrate that this mixed management approximates the planned and real storage levels with a low impact on the Quality of Service (QoS).**

***Index Terms*—Power decisions, Data center, Renewable sources**

I. Introduction

Information and Communications Technology (ICT) produce around 1.8-2.8% of the world's total greenhouse gas (GHG) [1]. The data centers sector is one of the most electricity-expensive ICT actors due to its uninterrupted operation [2]. A report revealed that Google data centers consumed the same amount of energy as the entire city of San Francisco in 2015 [3]. In addition, the situation tends to get even worst due to the predicted expansion of internet usage [4] and the reduction of the improvements in processor technologies [1]. Aiming to reduce the impact of data center providers' emissions, the IT community has started investigating the replacement of brown energy with renewable energy [5]. Renewable sources (RES), such as wind and sunlight, can provide clean energy. However, they introduce uncertainties due to weather conditions. Big cloud providers, such as Google and Amazon, smooth these uncertainties by inserting grid connections [6]. Therefore, they are not completely clean data centers. This work is part of the Datazero2 Project [7]. Datazero2 Project designed a data center operated only by RES without any link to the grid. This project aims to provide a feasible architecture to maintain data centers 100% clean.

A clean-by-design data center must introduce several elements to provide energy to the IT servers, such as Wind turbines, Solar panels, Batteries, and Hydrogen tanks. Consequently, a manager must predict the weather conditions and workload. These predictions allow the manager to plan long-term actions (offline), such as using more storage, recharging the batteries, discharging hydrogen, turning on servers, etc. Then, the manager must apply the plan (online). However, the predictions are not exactly the real values, so the manager must react to the online events. For example, workload prediction uses stochastic models to estimate the mass of the load. However, in online mode, it receives the precise jobs to execute. This paper proposes a way to mix online and offline decisions for data centers powered by only RES. The article's main contributions are twofold. First, it presents the model to integrate the two levels with the expected inputs and outputs. The objective is to maintain the storages' stability in real-time while using the offline plan as a guide. Second, this work details experiments showing the resilience of the proposed model. For a better understanding of the models, a table of the notation used is presented in Table I.

This paper is organized as follows. Section II presents the related work, highlighting the gap in state-of-the-art. Then Section III shows an overview of the problem. Section IV addresses the proposed model. Section V presents the experiments and the discussion about the results. Finally, Section VI concludes the article.

II. Related work

Several works seek ways to deal with RES uncertainties but, in many cases, with grid (brown) connections or just one level of management (online or offline) [8], [9], [10], [11], [12], [13]. The authors in [8] propose an offline data service price mechanism to vary the price according to the grid and RES price. They implemented a Stackelberg game to stimulate users to participate in the Demand-Response's load shift. In work [12], the authors modeled an offline optimization framework using a model to capture the randomness of the RES. [13] proposes two online job schedulers to maximize renewable energy usage. Their objective is to maximize the

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

Table I
NOTATIONS.

Notation	Description
S	Number of servers
s	Server index
T	Number of steps inside the time window
L	Step length $[s]$
t	Step index
D	Number of states
d	State index
j	Job index
SoC_t	State of charge of battery at step t $[kWh]$
$SoCmax$	Maximal state of charge $[kWh]$
LoH_t	Level of hydrogen at step t $[kg]$
$Pprod_t$	Power delivered by power commitment $[kW]$
$Pmin$	Power to maintain all servers sleeping $[kW]$
$Pmax$	Power to maintain all servers at maximum speed $[kW]$
Ppv_t	Photovoltaic Panel estimated power at time step t $[kW]$
Pwt_t	Wind Turbines estimated power at time step t $[kW]$
r	Relax factor
$Pload_t$	Estimated IT load at step t $[kW]$
σ	Self discharge rate
Pch_t	Power to charge the batteries at step t $[kW]$
$Pdch_t$	Power to discharge the batteries at step t $[kW]$
$Pchmax$	Maximum charge $[kW]$
$Pdchmax$	Maximum discharge $[kW]$
Pfc_t	Power from fuel cell at step t, discharging from Hydrogen $[kW]$
Pez_t	Power to electrolyzer at t, charging the Hydrogen $[kW]$
Pr_t	Estimated renewable power at step t $[kW]$
ηch	Battery charge efficiency
ηdch	Battery discharge efficiency
$E_{s,t}$	Energy usage by server s at step t $[J]$
$D_{s,d,t}$	Is the server s at state d at step t? $[bool]$
$l_{s,d,t}$	Duration that server s stays at state d at step t $[s]$
$P_{s,d}$	Power usage of server s at state d $[W]$
$F_{s,d}$	Speed of server s at state d $[flop]$
$wall_j$	Job j walltime $[s]$
$elapTime_j$	Job j elapsed time $[s]$
$jobFlop_j$	(Estimated) total work to do in job j $[flop]$
$elapFlop_j$	Job j elapsed work $[flop]$
ϵ_u	Calculated walltime error by the user u

number of jobs running when there is more solar irradiation, using the grid and batteries to deal with the intermittence. The authors in [11] describe two offline DVFS-aware scheduling algorithms: a Genetic Algorithm (GA) and a Similar Mathematical Morphology (SMM). SMM is quite similar to GA, but it uses frames to compare and improve the best solution. In [9], the authors create an online heuristic to efficiently assign the user requests to the data centers, reducing the makespan, energy consumption, and overall cost.

Finally, the authors of [10] proposed a Mixed Integer Linear Programming (MILP) to optimize the commitment of a data center powered by only wind turbines, solar panels, batteries, and hydrogen storage systems. The work is part of the Datazero2 Project. Their approach uses weather forecasts to find the optimal long-term decisions to supply a data center demand. However, the work does not consider real-time events, such as scheduling and fluctuations in power usage. Therefore, to the best of our knowledge, no work mixes offline and online decisions for data centers powered by only renewable energy.

III. PROBLEM STATEMENT

Figure 1 illustrates the problem, showing all the connections between the modules. There are four main groups: Offline, online, power plan, and IT plan. The offline group 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. Given each time step t with length $L = 300s$, the offline generates $T = 864$ steps. It defines different power and IT decisions for each step. First, the offline power commitment finds the optimal power decisions using the predictions and constraints. The output of this module is the power plan. Also, it indicates to the server configuration the power available to maintain the servers. The offline server configuration's main objective is to find the data center's fastest speed (in flops [14]) for the available power. The server configuration can change the processor's speed of each server due to Dynamic Voltage-Frequency Scaling (DVFS) technique. This technique allows reducing the processor's frequency to use less energy. The CPU frequency range is discrete, although some works define it to be continuous [15]. Table II shows a discretized example of the relationship between power and speed from Grid5000 [1]. The output of the server configuration is each server state at each time step for the next time window.

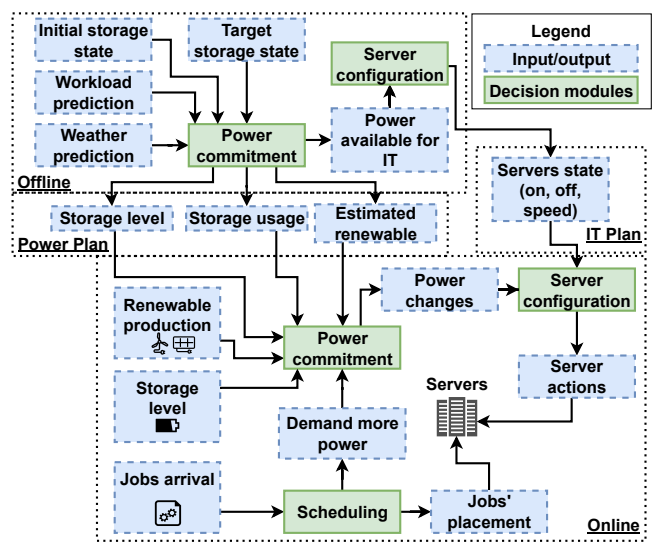

Figure 1. Online and offline integration with the input/output of each decision module.

Online follows the offline IT and power plans as much as possible. However, the online's main objective is to react and improve in real-time. So, the scheduler module receives the actual jobs submitted by the users and defines the job placement in the servers. After that, it evaluates the Quality of Service (QoS), asking for more power if it is not good enough. The power commitment verifies if it is possible to deliver more power without violation in the storage levels. If so, the power commitment changes the power usage and demands the server configuration to change the speed of the servers. The server configuration adapts the IT plan for the new power available, using Table II definition. It is important to notice that all online decisions must be fast. All this online process occurs at the beginning of each step. So, these online modules can not find the optimal solution for the time window.

[1] www.grid5000.fr

The following sections will detail the modules, presenting the proposed solution for each one.

Table II
SERVER DEFINITIONS. STATE 6 IS THE SLEEP STATE.

State (d)	Paravance		Taurus	
	$P_{s,d}$ (W)	$F_{s,d}$ (Gflops)	$P_{s,d}$ (W)	$F_{s,d}$ (Gflops)
0	200.5	38.4	223.7	18.4
1	165.1	34.56	189.03	16.56
2	136.76	30.72	161.28	14.72
3	114.69	26.88	139.67	12.88
4	98.10	23.04	123.43	11.04
5	86.22	19.20	111.79	9.20
6	4.5	0	8.5	0

IV. PROPOSED MODEL

A. Offline Model

1) Power Commitment Model: First of all, the management must create a power plan for the next time window. This section does not describe the sizing inputs (such as battery size, solar panel area, etc.) for ease of understanding. However, the experiments took them into account. As mentioned before, several works have proposed solutions for offline power commitment. This work uses the same model as [10]. This section highlights only some important aspects. The Article [10] can be consulted for more detailed information. This model has the following input:

- SoC_0: The State in Charge at the beginning of the time window;
- LoH_0: The Level of Hydrogen at the beginning of the time window;
- SoC_T: The desired State in Charge at the end of the time window;
- LoH_T: The desired Level of Hydrogen at the end of the time window;
- $Pload_t$: The estimated power load demanded by the data center at each step t;
- Ppv_t: Given estimated solar irradiation, the commitment applies the mathematical model from [16];
- Pwt_t: Given an estimated wind, the commitment applies the mathematical model from [17];

The sum between Ppv_t and Pwt_t is the total estimated renewable energy produced Pr_t at each step t. Using the targets of SoC_T and LoH_T as constraints, the model must finish the time window with both levels equal or greater than the target. Equation 1 demonstrates the SoC_t calculation using the Pch_{t-1} and $Pdch_{t-1}$. There are losses for battery charging/discharging (ηch and ηdch), and also a self-discharge (σ). $Pprod_t$ is the power delivered to IT servers by all sources (wind turbines, solar panels, batteries, and hydrogen). Equations 2 and 3 demonstrate the objective function. Given the previous constraints, the optimal solution is the closest value between $Pprod_t$ and $Pload_t$, matching the storage targets.

$$SoC_t = (SoC_{t-1} \times (1-\sigma)) + (Pch_{t-1} \times \eta ch \times L) - (\frac{Pdch_{t-1}}{\eta dch} \times L) \quad (1)$$

$$Pprod_t \geqslant (1-r) \times Pload_t \quad (2)$$

$$\textbf{minimize } r \quad (3)$$

The output of this model is:

- Pr_t: Estimate power from RES at step t. It is given by $Ppv_t + Pwt_t$;
- $Pdch_t$: Power to discharge from batteries;
- Pch_t: Power to charge the batteries;
- Pfc_t: Power to discharge from Hydrogen;
- Pez_t: Power to charge the Hydrogen;
- $Pprod_t$: Power for IT servers at step t. It is $Pprod_t = Pr_t - Pch_t - Pez_t + Pdch_t + Pfc_t$;
- SoC_t: State of charge at step t;
- LoH_t: Level of Hydrogen at step t;

2) Server Configuration Model: After creating a power plan, the server configuration translates $Pprod_t$ into server states. Equation 4 defines the objective function of the model. $D_{s,d,t}$ describes the state d of the server s at step t. So, Equation 4 finds the highest possible speed delivered by the IT. Equation 5 introduces the power constraint $Pprod_t$ where $E_{s,t}$ is the energy usage of server s at step t. Equation 6 calculates this energy usage. $P_{s,d}$ indicates the server s power usage at state d. Table II shows an example of $P_{s,d}$ and $F_{s,d}$ for each d. Due to size limitations, the article will not present all equations. Besides the presented formulas, the offline server configuration also considers the transitions between running and sleeping. These transitions use energy and take time. During the transition, the server is unavailable to execute jobs, so they do not increase the flops in Equation 4. The only output of this model is $D_{s,d,t}$.

$$\textbf{maximize} \sum_{t=0}^{T} \sum_{s=0}^{S} \sum_{d=0}^{D} D_{s,d,t} \times F_{s,d} \times l_{s,d,t} \quad (4)$$

$$\sum_{s=0}^{S} E_{s,t} \leqslant Pprod_t \times L, \forall t \quad (5)$$

$$E_{s,t} = \sum_{d=0}^{D} P_{s,d} \times l_{s,d,t}, \forall s, t \quad (6)$$

B. Online Model

1) Scheduling: The main objective of a data center is to run jobs coming from users. The scheduler component defines the action of running a job j in a server s. The scheduling model is applied to scientific High-Performance Computing (HPC) data centers. The user of these data centers submits an expected job duration (named walltime) with the job. If the job takes more time than the walltime, the scheduler kills the job. Usually, the scheduler is driven by some QoS metric. Our QoS metrics are twofold: Number of jobs killed and slowdown. The former indicates how many jobs have started but, for some reason, did not finish their execution. Equation 7 presents the latter QoS metric [18]. The slowdown indicates the relationship between the time that the job spends in the system and its size. The best slowdown value is 1, where the scheduler puts the job to

run as soon as it arrives. It increases according to the waiting time.

$$Slowdown = \frac{Waiting\ time + Execution\ time}{Execution\ time} \tag{7}$$

The model deals with both metrics in two ways. First, the scheduler decides the placement in the available servers given by $D_{s,d,t}$ plan. We chose as the scheduling algorithm the heuristic EASY-Backfilling [19]. This heuristic naturally improves the slowdown due to its backfilling behavior, but we also sort the job waiting list by the slowdown. It uses the walltime as execution time. Secondly, the scheduler maintains the servers at the minimum speed to avoid killing the jobs using Equation 8. Equation 9 estimates the job size (in floating-point operations) using the walltime $wall_j$, a fixed server's speed $F_{s',d'}$, and an error ϵ_u. In [20], the authors introduced the error ϵ_u calculated by each user. They claimed that the users usually overestimate the execution time by up to 5 times. Therefore, Equation 8 estimates how many flops the server must deliver to finish the job before the deadline. The scheduler will modify the power plan to avoid killing jobs. However, this modification respects the rules of the next section.

$$(wall_j - elapTime_j) \times D_{s,d,t} \times F_{s,d} \geqslant jobFlop_j - elapFlop_j \tag{8}$$

$$jobFlop_j = wall_j \times F_{s',d'} \times \epsilon_u \tag{9}$$

2) Power commitment: The power commitment has one objective: to ensure that SoC and LoH end close to the plan. Hydrogen is harder to work in online mode since it needs time to turn on/off. So, this model focuses on batteries level decisions. The battery power usage can vary from the plan. This variation happens because:

1) Server idleness: The offline estimates that the servers will spend maximum usage during the time step. However, the server can stay idle for some time;
2) Renewable production: Renewable production can be different from the estimated. The batteries will maintain the servers running at the defined speed;
3) Scheduling changes: The scheduler can ask for more power;

The power commitment compensates for these variations in future steps. Figure 2 illustrates the three compensation policies: Peak, Next, and Last. In the figure example, if the scheduler demands more power at step 1, it could compensate for it at step 3 (Next), step 12 (Peak), or step 15 (Last). The same happens for server idleness and renewable production. However, since both are unpredictable, their compensation is reactive (after finishing the step). When the power needed is less than the plan, the commitment also modifies a future step. The Next and Last policies are similar independently of the type of compensation. In Peak policy, in case of less usage, it will increase a step with the lesser power usage. The power compensation must observe the rules from Equations 10, 11, and 12. It is important to notice that Equation 11 uses Pr_t to define the maximum possible battery charge. This value is

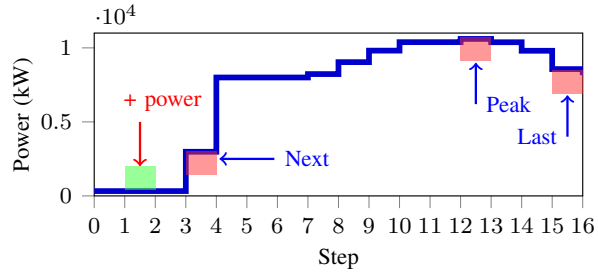

Figure 2. Compensation policies.

an estimation given by the offline. So, the power commitment uses offline prediction to find when to compensate.

$$Pmin \leq Pprod_t \leq Pmax, \forall t \tag{10}$$

$$0 \leq Pch_t \leq \min(Pr_t, Pchmax), \forall t \tag{11}$$

$$0 \leq Pdch_t \leq Pdchmax, \forall t \tag{12}$$

Besides this compensation, power commitment also evaluates the impact of the power usage variations on the state of charge. So, it recalculates the state of charge recursively of the future steps using Equation 1. If this deviation violates a future SoC, the heuristic changes the usage in the steps before the violation. A violation is given by Equation 13. For example, if the algorithm estimates that the SoC will be above 95% in the future, it will increase the usage backward from this violation. The heuristic also compensates for this modification using the same three previous policies. Finally, the last commitment responsibility is answering the scheduler if it is possible to modify a usage. The commitment verifies if it is possible to compensate for the change and if it will not violate the SoC.

$$0.05 \times SoCmax \leq SoC_t \leq 0.95 \times SoCmax, \forall t \tag{13}$$

3) Server configuration: The last part is transforming the power change in server configuration. Since the offline server configuration model is an NP-Complete problem, it is not possible to run it to find the best solution quickly. So, this model is a simple heuristic that uses the $D_{s,d,t}$ given by the offline plan but changes it according to the power available. This heuristic uses Equations 5 and 6 as constraints. It creates a list of power and flop differences from one state to another (using the data from Table II). So, it can fastly find the possible speed increases below the power available. When the step has more power than planned, the heuristic executes the following:

1) Find the highest improvement possible for the servers running some job;
2) Find the highest improvement possible for the idle servers.

For example, if there are two Paravance servers: one with $d = 5$ and another with $d = 2$. If the system has $30W$ to increase, it will increase first the server with $d = 5$, because it will increase 7.68 Gflops (against 3.84 Gflops). When the step has less power than planned, it runs the following:

1) Reduce the speed of idle servers;

2) Reduce the speed of the servers running jobs with higher $(wall_j - elapTime_j)$.

The second step will reduce servers' speed with more time to compensate for this reduction in the future. It is better to maintain jobs closer to finish with the maximum speed, granting that they will complete.

V. Simulation Experiment

A. Tested Scenarios

The main experiment's idea is to compare the three compensation policies (Peak, Next, and Last) with a baseline implementation. The baseline implementation only follows the offline IT and Power plans with no modifications. The simulated data center consists of 50 servers from Grid5000: 25 Paravance and 25 Taurus. This setup adds heterogeneity to the simulation. This configuration has a power demand range from $Pmin = 0.325kW$ to $Pmax = 10.605kW$. We have generated 20 different IT workloads using the Google Trace Generator from [21] with an arrival rate of 50 jobs per hour for three days (3600 jobs in total). Also, we have created two power profiles using different days of irradiation and wind from the Los Angeles data set [22]. Figure 3 demonstrates both profiles.

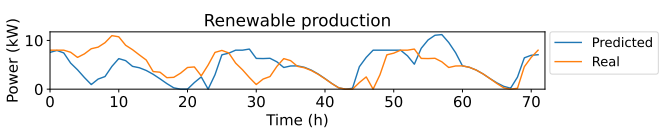

Figure 3. The renewable produced by the Los Angeles data set.

The first power profile (Predicted) emulates a power prediction. The second (Real) is the power profile used for renewable production in the experiments. The power system consists of two wind turbines and 30 m² of photovoltaic panels. The battery $SoCmax = 64kWh$. We have taken one of the 20 workloads to generate the input $Pload_t$ for the offline. We have defined $SoC_0 = 50\%$, $SoC_T = 50\%$, $LoH_0 = 300kg$, $LoH_T = 300kg$. Then, we have run two scenarios for all workloads:

S1 Predicted power profile for offline and the Real power profile for online. This scenario simulates a poor weather prediction;

S2 Both offline and online with predicted power profile. This scenario simulates an excellent weather prediction;

Table III shows the results of the experiments. The best storage levels are closer to 50% for the battery and 300 kg for hydrogen (both target levels). We have highlighted the best results in green and the worst in red. The sum of completed and killed jobs is not 100% because the simulation rejects all waiting jobs at the end of the time window.

B. Discussion

Table III shows that the baseline has the best QoS results in general. This execution stays close to the plan independently of the power fluctuations. Therefore, it maintains the server speed, being less likely to kill jobs. However, its hydrogen and battery levels are the worst in both scenarios due to its lack of power awareness. In scenario S1 (poor weather prediction), the baseline execution finished with a low battery level and high hydrogen level. The power conversion in hydrogen has a high loss, so it is crucial to use the hydrogen only in (planned) extreme cases. On the other hand, it ended with the right hydrogen level but a higher battery level in scenario S2 (good weather prediction). This variance shows that the baseline is highly dependent on both renewable production and workload estimations. Figure 4 highlights this baseline behavior from the battery level point of view. In S1, the baseline always finished at around 30%, but in S2, it ended at approximately 70%. Since the baseline does not adapt the power plan, it does not reintroduce the power not used due to server idleness. So, both $PLoad_t$ and Pr_t estimation must be almost perfect to have good results in the baseline approach.

Table III
Results from all experiments with the average (outside parentheses) and standard deviation (inside parentheses).

Exp		Slowdown	Completed jobs (%)	Killed jobs (%)	Battery level (%)	Hydrogen level (kg)
S1	Baseline	7.0 (0.8)	95.85 (0.74)	0.37 (0.07)	30.1 (0.7)	300.4722 (0.0067)
	Peak	8.7 (0.7)	94.07 (0.70)	0.36 (0.06)	58.9 (2.3)	300.0003 (0.0013)
	Next	10.9 (1.1)	91.77 (0.73)	0.79 (0.15)	36.1 (5.7)	300.0002 (0.0008)
	Last	8.5 (0.8)	94.03 (0.77)	0.35 (0.06)	53.8 (1.9)	300.0004 (0.0013)
S2	Baseline	7.0 (0.8)	95.85 (0.74)	0.37 (0.07)	71.4 (1.0)	300 (0)
	Peak	9.1 (0.8)	94.95 (0.75)	0.38 (0.07)	52.2 (0.4)	300 (0)
	Next	7.9 (0.8)	95.17 (0.73)	0.42 (0.06)	49.8 (0.3)	300 (0)
	Last	9.3 (0.8)	94.28 (0.69)	0.41 (0.07)	66.0 (0.5)	300 (0)

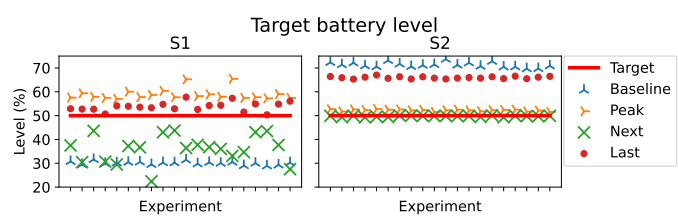

Figure 4. Battery target distance. The x-axis shows each different workload execution.

For the policies, the best results are related to the level of prediction. Under a scenario of high uncertainty (S1), it is better to compensate in the end. For example, if the scheduler requires more power at any step, the Last policy will reduce the usage at the last possible moment. Any other server idleness from this step will also be compensated at the end. So, idleness could smooth this change, reaching the planned power from the beginning. On the other hand, the Next policy will reduce the usage in the first moment after the scheduler requirement. Since the scheduler demand is likely at a step with a higher difference between estimated and real, this behavior could fastly degrade the storage level. Figure 4 well illustrates the Next and Last behaviors, with Last outperforming Next in almost every metric in Scenario S1 from Table III. There is a

small difference between the Next and Last hydrogen levels that can be ignored. In a scenario of an excellent weather estimation (S2), it is better to stay close to the plan. Next outperforms the other policies in almost every metric because it compensates for the power increasing as soon as possible. Also, Next is nearly perfect in the battery level, staying very close to the target level in Figure 4. Putting all compensations at the end may be too late to use the power. So, the Last policy finished with higher storage than the target.

Finally, the Peak policy is the more stable execution. Independently of the scenario, it delivered a well-balanced solution, resulting in good QoS and storage level values. It has a slightly higher storage level on average for scenario S1. For scenario S2, it is very close to the target level. The Peak policy will flat the curve of battery usage. In uncertain environments, it is a reasonable way to smooth the modifications. With good estimation, it will "steal" from the step where the power is more abundant. So, this future step could degrade the jobs. Therefore, the best policy depends on how good are the weather estimation:

1) Last policy with a poor prediction;
2) Next policy with a good prediction;
3) Peak policy with undefined prediction accuracy.

VI. Conclusion

Datazero2 is a project that aims to model a green by-design data center powered by only renewable sources. Due to the several elements that compose a renewable-only data center, it is vital to mix offline and online decisions. This work presented a model for online management to approximate the storage level from the offline plan. The results demonstrated that simply following the offline plan is not enough due to the variance in workload and renewable production. The experiments also revealed that the presented policies could approximate the target storage level without a significant impact on QoS. Future works will include other policies linked to the workload estimation (e.g., increase power when there will be more jobs arriving) and more complex QoS improvements (e.g., improve future slowdown).

References

[1] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. Blair, and A. Friday, "The climate impact of ict: A review of estimates, trends and regulations," 2021.

[2] Q. Zhou, M. Xu, S. S. Gill, C. Gao, W. Tian, C. Xu, and R. Buyya, "Energy efficient algorithms based on vm consolidation for cloud computing: comparisons and evaluations," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. IEEE, 2020, pp. 489–498.

[3] M. A. Khan, A. P. Paplinski, A. M. Khan, M. Murshed, and R. Buyya, "Exploiting user provided information in dynamic consolidation of virtual machines to minimize energy consumption of cloud data centers," in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2018, pp. 105–114.

[4] U. Cisco, "Cisco annual internet report (2018–2023) white paper," 2020.

[5] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020. [Online]. Available: https://science.sciencemag.org/content/367/6481/984

[6] S. Kwon, "Ensuring renewable energy utilization with quality of service guarantee for energy-efficient data center operations," *Applied Energy*, vol. 276, p. 115424, 2020.

[7] J.-M. Pierson, G. Baudic, S. Caux, B. Celik, G. Da Costa, L. Grange, M. Haddad, J. Lecuivre, J.-M. Nicod, L. Philippe, V. Rehn-Sonigo, R. Roche, G. Rostirolla, A. Sayah, P. Stolf, M.-T. Thi, and C. Varnier, "DATAZERO: DATAcenter with Zero Emission and RObust management using renewable energy," *IEEE Access*, vol. 7, p. (on line), juillet 2019. [Online]. Available: http://doi.org/10.1109/ACCESS.2019.2930368

[8] C. Jiang, C.-L. Tseng, Y. Wang, Z. Lan, F. Wen, F. Chen, and L. Liang, "Optimal Pricing Strategy for Data Center Considering Demand Response and Renewable Energy Source Accommodation," *Journal of Modern Power Systems and Clean Energy*, pp. 1–9, 2021, conference Name: Journal of Modern Power Systems and Clean Energy.

[9] S. K. Nayak, S. K. Panda, S. Das, and S. K. Pande, "An efficient renewable energy-based scheduling algorithm for cloud computing," in *International Conference on Distributed Computing and Internet Technology*. Springer, 2021, pp. 81–97.

[10] M. Haddad, J. M. Nicod, C. Varnier, and M.-C. Peéra, "Mixed integer linear programming approach to optimize the hybrid renewable energy system management for supplying a stand-alone data center," in *2019 Tenth international green and sustainable computing conference (IGSC)*. IEEE, 2019, pp. 1–8.

[11] X. Liu, P. Liu, H. Li, Z. Li, C. Zou, H. Zhou, X. Yan, and R. Xia, "Energy-aware task scheduling strategies with QoS constraint for green computing in cloud data centers," in *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, ser. RACS '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 260–267. [Online]. Available: http://doi.org/10.1145/3264746.3264792

[12] Y. Lu, R. Wang, P. Wang, Y. Cao, J. Hao, and K. Zhu, "Energy-Efficient Task Scheduling for Data Centers with Unstable Renewable Energy: A Robust Optimization Approach," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Jul. 2018, pp. 455–462.

[13] Y. Li, A.-C. Orgerie, and J.-M. Menaud, "Balancing the Use of Batteries and Opportunistic Scheduling Policies for Maximizing Renewable Energy Consumption in a Cloud Data Center," in *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, Mar. 2017, pp. 408–415, iSSN: 2377-5750.

[14] R. Hunger, *Floating point operations in matrix-vector calculus*. Munich University of Technology, Inst. for Circuit Theory and Signal . . . , 2005.

[15] S. Saha and B. Ravindran, "An experimental evaluation of real-time dvfs scheduling algorithms," in *Proceedings of the 5th Annual International Systems and Storage Conference*, 2012, pp. 1–12.

[16] W. Dong, Y. Li, and J. Xiang, "Optimal sizing of a stand-alone hybrid power system based on battery/hydrogen with an improved ant colony optimization," *Energies*, vol. 9, no. 10, p. 785, 2016.

[17] S. Sinha and S. Chandel, "Review of recent trends in optimization techniques for solar photovoltaic–wind based hybrid energy systems," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 755–769, 2015.

[18] S. Srinivasan, R. Kettimuthu, V. Subramani, and P. Sadayappan, "Characterization of backfilling strategies for parallel job scheduling," in *Proceedings. International Conference on Parallel Processing Workshop*. IEEE, 2002, pp. 514–519.

[19] D. A. Lifka, "The anl/ibm sp scheduling system," in *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer, 1995, pp. 295–303.

[20] S. Takizawa and R. Takano, "Effect of an incentive implementation for specifying accurate walltime in job scheduling," in *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*, 2020, pp. 169–178.

[21] G. Da Costa, L. Grange, and I. De Courchelle, "Modeling, classifying and generating large-scale google-like workload," *Sustainable Computing: Informatics and Systems*, vol. 19, pp. 305–314, 2018.

[22] M. Sengupta, Y. Xie, A. Lopez, A. Habte, G. Maclaurin, and J. Shelby, "The national solar radiation data base (nsrdb)," *Renewable and sustainable energy reviews*, vol. 89, pp. 51–60, 2018.