Digital sufficiency behaviors to deal with intermittent energy sources in a data center

Jolyne Gatt, Maël Madon*, Georges Da Costa
IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, France
*corresponding author, email: mael.madon@irit.fr

***Abstract*—Data centers have a considerable environmental footprint, mainly arising from their electricity consumption. Co-locating data centers with renewable energy sources allows reducing this footprint, but comes with the challenge of production intermittency. Users of data centers could adopt "digital sufficiency" behaviors and adjust their use of these infrastructures when renewable production is low, by adapting their job submissions.**

In this work, we investigate how effort from users can help to minimize energy consumption in critical periods. We study five behaviors, namely renouncing the submission, degrading it spatially or temporally, reconfiguring it and postponing it to later. These behaviors are combined with a three-state feedback mechanism to provide simple information on the status of renewable production. We propose a validation of our method through a reproducible experimental campaign with a state-of-the-art simulator. We show, for example, that if users accept to apply the above behaviors on 50% of their job submissions at periods of low production, brown energy consumption can be reduced by 8%. Energy savings are linear in the size of the effort made by users.

To the best of our knowledge, this is the first time all these user levers are combined in a context of renewable-energy powered data center. Future works could focus on refining the user model, introducing states and behaviors of anticipation before a drop in production and investigating the willingness of real users to adopt the behaviors.

***Index Terms*—Energy-aware, HPC, simulation, eco-feedback, reproducibility**

I. Introduction

The Information Technology (IT) industry has growing environmental impacts. Estimations of the carbon footprint of the IT industry range from 1.8% to 3.9% of global greenhouse gas emissions in 2020 [1]. About a third of this footprint is attributable to data centers, the server farms that constitute the backbone of the Internet infrastructure. For example, data centers represent 2.4% of the total electricity consumption of a country like France [2].

To reduce this impact, large IT companies have been making 100% renewable energy commitments in the last decade [3]. Most of the time, these commitments are achieved through green tariffs or power purchase agreements [3]. A problem with these contracts is that they do not directly create the new renewable capacity necessary to cover the electricity demand. For this reason, companies have started going one step further, and installing on-site renewable sources on their data centers (e.g., Google in Belgium [4]). Often, the renewable electricity produced is actually sold to the market, leaving the burden of balancing demand with intermittent supply to grid operators. The ultimate step for a truly sustainable data center would be to self-consume this electricity to become fully autonomous in energy supply.

Some papers in the scientific literature study the use and smart management of energy storage systems to overcome the challenge of intermittency [5], [6]. As usual, and like in the majority of works in the field, the levers considered are levers of *efficiency*, i.e., optimizing systems to consume fewer resources, for the same service provided to the user. These techniques have the potential to decrease direct environmental impacts of IT, but fail to address indirect (enabling and systemic) effects [7]. In particular, efficiency in IT often leads to rebound effects [8]. For this reason, an increasing number of international bodies (like IPCC in their 2022 report [9]) have started to acknowledge that efficiency measures must also be accompanied by *sufficiency* measures, i.e., strategies aiming at decreasing the absolute level of resource and energy demand.

The term "digital sufficiency" is relatively new and was theorized recently by Santarius et al. [10]. One of its major dimensions is "user sufficiency", and consists in voluntarily decreasing the demand for digital services. We argue that environmental-aware people are ready to make efforts, as in the example of users of French telecom operator Telecoop restricting their use of mobile data [11].

In previous work, we characterized the impact of so-called "digital sufficiency behaviors" for data center users [12]. In this paper, we propose to study their potential to reduce the load in the data center in periods of low electricity production. We suppose that the users interact with the data center by submitting batch jobs, like in a High Performance Computing (HPC) platform. We consider different types of user flexibility, without excluding the most radical ones: submitting less, shorter or smaller tasks, and submitting later. Our aim is to understand how these behaviors would impact the management of a data center only powered by renewable energy.

Our main contributions are:

- a model of user flexibility for data center users, including delaying, reconfiguring, degrading and renouncing their job submissions;

This work was supported by the French Research Agency under the project DataZero2 (ANR-19-CE25-0016).

- a three-state energy feedback mechanism to inform users on the state of renewable production;
- both their implementation in a state-of-the-art and open-source simulator;
- a set of reproducible experiments to validate the approach.

With these contributions, we investigate the two following research questions:

(1) *How much does user effort impact energy consumption? Is there a threshold where more effort does not result in more energy savings?*

(2) *Does the third state in the energy feedback mechanism have an added value or is it sufficient to only consider nominal and low-renewable states?*

The remainder of this article is organized as follows. In Section II we propose an overview of the works related to the impact of user behavior on data center usage. Section III describes our context and defines the proposed behavior. Then, in Section IV, we detail the experimental environment used for validation purpose. Results are described in Section V and discussed in Section VI. Limitations are presented in Section VII. Finally, Section VIII suggests future works and Section IX concludes the article.

II. Related Work

This section gives an overview of other studies involving users to reduce the carbon footprint of data centers.

Orgerie et al. look at energy savings reachable through accepting temporal delay in the start times of jobs [13]. Their results show that 3% energy can be saved if all users accept delay, and this delay is of 15 hours on average. Guyon et al. look at the impact of spatial reconfiguration of jobs on energy consumption [14]. Users can decide to submit their jobs on a smaller number of computing nodes ('big', 'medium' or 'little' version). Having all users submitting their jobs in 'medium' allows for 20% energy savings compared to 'big', thanks to better bin-packing and machine switch off. In another article, they combine the two levers previously mentioned by proposing their users to accept both delay and reconfiguration in their jobs [15].

Apart from these works on user involvement, a rich literature can be found on the integration of renewable energy in data centers, as discussed in a recent survey [16]. This integration is sometimes combined with so-called "green SLA" (Service Level Agreement), where users ask for a low environmental footprint in their contract with the data center operator [17], [18]. These objectives are typically met through other levers (e.g., self-supply of renewable energy, geo-distributed data centers).

Basmadjian et al. go a step further, and propose in the project All4Green a collaboration between the energy supplier, the data center and customers [19]. The objective is to better match the supply and demand of electricity. Many mechanisms are leveraged: "internal flexibilities" (namely migrations, use of batteries, and adjustment in the cooling temperature) and "external flexibilities" (namely delay and performance degradation for users). Their approach allows 38% energy savings with internal flexibilities only, and a further 5.5% with external flexibilities. It proves particularly useful in the context of demand response: they can reduce the power by 50% during a 2-hour window.

	Delay	Reconfig	Degrad	Renounce	RE?*
Orgerie et al. [13]	✓				
Guyon 2019 [14]		✓			
Guyon 2018 [15]	✓	✓			
All4Green [19]	✓		✓		✓
Madon et al. [12]	✓	✓	✓	✓	

*is the work in the context of Renewable Energy integration?

Table I: Summary of related works and their links to the behaviors studied in this article (Figure 1)

In a previous work, we introduce one additional user flexibility: renouncing the job submission [12]. The three other levers are also characterized: 'delay' (temporal shifting), 'reconfig' (downsizing of requested resources) and 'degrad' (performance degradation). We provide a ranking of their efficiency to reduce the load in the data center in a short window of time (one or four hours), with the help of simulation.

A brief summary of related works is given in Table I. To the best of our knowledge, our study is the first to combine all four user levers together in the context of renewable energy integration. These levers are the center of the study, compared to others where they play only a secondary role. We include the flexibility 'renounce' in a deliberate approach of digital sufficiency [10]. Unlike most studies, we use a state-of-the-art simulator and provide all the software and material to reproduce the experiments as open-source repositories.

III. Model

Renewable energy production is not constant over time. Our approach consists in providing information on production status to users when they want to submit a job, and let them decide how to react. The provided feedback is made simple by adopting the three colors of a traffic light. Users, when submitting jobs, are shown a green, yellow or red light, ranging from high to low renewable production. A user can then choose to react by either renouncing if they consider that this particular job is not critical, reconfiguring the job to reduce its load on the infrastructure, or waiting some later occasion to resubmit. This section describes our model in more detail.

A. Main components

Our data center model is composed of the following components:

- **Users**: they are the humans that make use of the *system*, by submitting *jobs* to it.
- **Jobs**: they are masses of calculations that must be carried out, modelled by an execution time and number of computing resources (later called 'cores') needed to complete.
- **System**: it is the renewable energy powered data center, subdivided into

– **the platform**: group of servers, executing the *jobs*,
– **the scheduler**: program that receives the *job* requests from the *users* and assigns them to specific servers in the *platform* to complete their execution, and
– **the power plant**: renewable energy sources producing the electricity to power the servers.

For the sake of simplicity, memory, network and ancillary equipment such as cooling and lighting are not modelled.

B. Energy state model

The users adapt their job submissions to the current state of renewable energy production. When production is low, they are invited to submit fewer and smaller jobs. To simplify the information that is given to them, we define three “energy states”, named like the colors of a traffic light:

- **Green state**: The system is alright: there is enough energy to power the whole platform.
  We are in this state when the energy production is greater than the max energy consumption of the platform E_{full}.
- **Yellow state**: The system is disturbed: the production is not enough to power the whole platform, but it can power a big part of it.
  We are in this state when the energy production is below E_{full}, but above a certain threshold τ, that we fixed arbitrarily to $0.5E_{full}$.
- **Red state**: The system is critical: the production is low, we must reduce the energy consumption of the platform.
  We are in this state when the energy production is below the threshold τ.

This three-state model can be seen as an eco-feedback design [20], providing simple yet actionable information to the users. Compared to a two-state model, it allows for more expressiveness. In our case, the yellow state indicates that a user effort would be appreciated, but the situation is not critical. It can also be seen as a transition state between green and red, giving the information that the system will soon become critical or is not critical anymore but not yet calm. Finally, this model provides a useful abstraction layer. It is easy to change the way the energy states are defined, by selecting different thresholds or changing the metrics on which they are based. For example, one could define the energy states based on level of battery or instantaneous data center power consumption instead of renewable production like here.

Note that our energy state model is entirely defined, for a given renewable energy production data, by the thresholds on max energy consumption (50% and 100%, in our case). Time is discretized into units of time during which the system is considered to be stable. The time intervals during which one energy state occurred are called “state windows” or simply “windows”. They can last one or several units of time. For example, if energy production is under the red threshold from 6:00 to 8:00, we say that a “red state window” of 2 hours occurred.

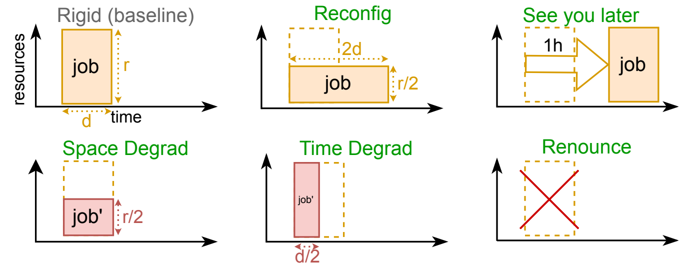


Figure 1: **Graphical representation of the sufficiency behaviors compared to the baseline.**

C. User behavior model

We make the hypothesis that the users will adapt their submission to the state of the system. We consider six submission behaviors: the baseline and five “sufficiency behaviors”. They are illustrated in Figure 1 and described below.

- **Rigid**: submitting the job now and without modification.
- **Reconfig**: submitting the job now, but dividing by two (rounded up) the number of cores requested. The execution time increases proportionally.
- **See you later**: delaying the job submission by one hour. The user will then take a new decision on that job.
- **Space Degrad**: submitting a degraded version of the job now, requesting only half (rounded up) of the original number of cores. The execution time remains the same.
- **Time Degrad**: submitting a degraded version of the job now, which takes half of the original execution time.
- **Renounce**: not submitting the job (and never submitting it in the future).

IV. Experiments

We design a simulation campaign to investigate the potential of the “traffic light” approach to manage efficiently the phases where renewable energy is scarce.

A. Behaviors for each energy state

In the absence of data on the actual behaviors that would be chosen by the users in reaction to eco-feedback, we made the following assumptions.

- **Green state:** there is enough energy, we assume that users submit normally (Rigid behavior).
- **Red state:** we assume that users will make an effort by adopting any of the sufficiency behaviors, depending on their context. Some will accept to reconfigure or degrade the job, if they can. Some will choose to come back later, when the energy state is hopefully better (See You Later). And some will simply Renounce the job, which was maybe not important enough.
- **Yellow state:** we exclude the behavior Renounce, as we assume that the sacrifice would be too big in regard to the criticality of the state. See You Later is also excluded because a yellow state is an intermediate state, so delaying jobs might push them to red states. However,

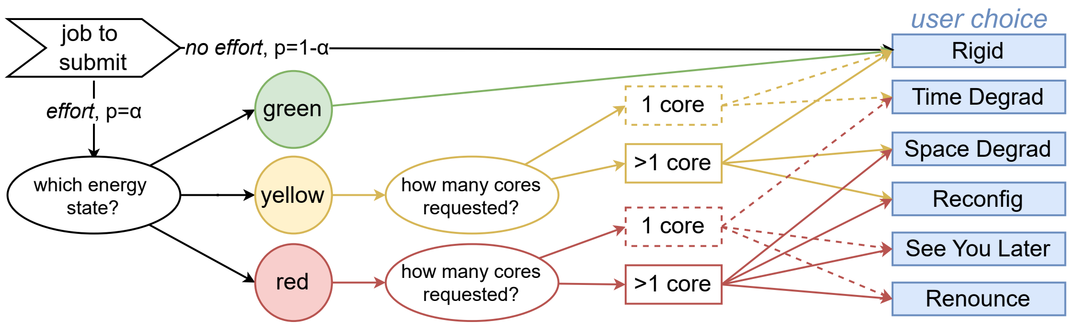


Figure 2: **Decision chart when a user submits a job.** The effort probability α represent the probability for users to consider adopting a sufficiency behavior. For a fixed energy state and number of cores, the available behaviors are equiprobable. For example, multicore job in red state: the four available behaviors have a probability 1/4.

we added behavior Rigid, as some users might choose to ignore the warning and submit normally.

- **Equiprobability of behaviors:** finally, we assume that each time users take a decision, they make a uniform randomized choice.

Distinction had to be made between monocore and multi-core jobs, as space degradations and reconfigurations are not possible on monocore jobs. The modifications applied to the jobs depending on the energy state are illustrated in Figure 2.

B. Experimental plan

To answer research question (1), we simulate an IT platform powered by renewable energy and introduce an additional parameter α. This parameter allows us to vary user involvement, and represents the probability of making a modification to the job when in red or yellow state (see Figure 2). It can take four values, from low effort ($\alpha = 0.25$), medium effort ($\alpha = 0.5$), big effort ($\alpha = 0.75$), up to max effort ($\alpha = 1$).

For each value of α, the simulation is repeated 30 times, to minimize the effect of randomness. As weather and tasks input are always identical, the only difference between each of the 30 experiments is the random choice of effort and then of the exact behavior, following Figure 2. We also add two experiments, for comparison purposes:

- *full rigid*, where all jobs are submitted Rigid all the time (equivalent to $\alpha = 0$). This corresponds to the baseline.
- *full renounce red*, assuming that all users would Renounce submitting jobs in red state. This provides an upper bound on energy savings reachable.

To answer research question (2), the whole experimental campaign is repeated, without taking into account the yellow state. They are instead treated by users as green states.

In the end, our experimental campaign consists of (4 values for alpha) $\times$ (30 repetitions) $\times$ (2 treatments) + (2 exp. for comparison) = **242 simulation runs**.

C. Experimental setup

Software. The experiments were run with Batsim[1], a state-of-the-art infrastructure simulator [21] based on SimGrid[2]. Batsim simulates the resource and job management system including the job manager, resource manager and scheduler, using discrete event simulation. The implementation of the user behaviors described in Section III is available in Batmen[3], our open-source plugin for Batsim enabling the simulation of users.

IT workload. As an input of our simulation, we use a workload adapted from the MetaCentrum trace[4]. This log contains two years worth of record of the Czech National Grid Infrastructure MetaCentrum from January 2013 to April 2015 [22].

Given the high heterogeneity of the workload, we performed the following filtering:

- Step 1: (excludes 87% jobs representing 86% core-hours)
  - Using only the period from June 1, 2014, to November 30, 2014. This part was taken because no cluster was removed or added during this period of time.
  - Removing clusters with GPU, because our simulation only simulate jobs running on CPU.
  - Selecting only the clusters with 12- or 16-core machines, because our simulated platform is composed of 18-core machines.
- Step 2: (excludes a further 5% jobs, 86% core-hours)
  - Removing jobs running on more than 18 cores, because our scheduler does not authorize multi-machine execution.
  - Removing jobs running for more than 15 hours, to mitigate inertia in the system.

After the above filters, our workload is six-months long and contains 693066 jobs and 474 users.

Energy Production Data. We consider electricity production from photovoltaic panels. The energy input data is

[1]https://batsim.org/ version 4.10
[2]https://simgrid.org/ version 3.31
[3]https://gitlab.irit.fr/sepia-pub/mael/batmen version 2.0
[4]file `METACENTRUM-2013-3.swf` available at https://www.cs.huji.ac.il/labs/parallel/workload/l_metacentrum2/index.html

generated from weather (solar irradiation) data provided by the website renewables ninja[5]. The reference point for the weather is 2019, in the city of Toulouse, France. Data is provided with one-hour time steps. Note that the weather and workload traces are not from the same year. However, we took care to align the days in the IT workload with the days in the weather, i.e. the workload of June 24 will be replayed with the energy production of June 24.

For the experiments, we do not take into account energy storage systems. We simply consider that, in periods of low energy production, electricity has to come from other sources, be it batteries, power generator or the electricity grid. However, batteries and fuel cells are considered in the Datazero project, and were taken into account to produce a suitable sizing for both IT and energy platforms (see below).

IT&power platform. Both platforms were created by colleagues, ensuring that the volume of renewable production is sufficient to cover the energy needs from IT, taking into account the efficiency of batteries. The sizing technique used is similar to the one described in this article [25]. The simulated renewable sources consist of $a = 145\text{m}^2$ of solar panels with efficiency $\eta = 0.206$. The power produced at time t is obtained by the formula $\eta \times a \times irr(t)$, with $irr(t)$ the solar irradiation. The simulated IT platform is composed of 42 18-core machines, with idle power consumption 62W and max power consumption 143.4W[6].

Scheduler. To handle the multicore workload and platform, we chose a bin-packing scheduler with idle machine switch off. To be more precise, the jobs are in queue sorted by decreasing requested number of cores and then by increasing submission time. The list of machines is sorted by increasing number of available core, with the switched-off machines at the end. At each scheduling decision, we go through the job queue in order. For each job, the list of machines is traversed to find the machine with the smallest number of available cores that fit the job, if any. If the job is scheduled on a switched-off machine, the machine is switched on (with a delay of 150 seconds). At the end of the bin-packing algorithm, if there are idle machines, they are switched off (delay of 7 seconds).

Time and carbon footprint of the campaign. The experimental campaign ran in 7 hours on a 2x16-core Intel Xeon Gold 6130 and its output took 55 GB storage space. According to a watt meter on the machine used, it consumed 2 kWh of electricity, in the city of Grenoble (France). Assuming a French carbon intensity of electricity of 38 gCO_2e/kWh[7], the carbon footprint of our experiments is estimated to 76 gCO_2e.

Reproducibility. All the material to reproduce our experimental campaign and its analysis are available in a GitLab repository[8]. It contains the Nix file defining the software dependencies, the scripts to launch the experiments and the Notebooks to analyze the output data and produce the graphs included in this article.

[5]open-source weather data repository available at https://www.renewables.ninja/ [23], [24]

[6]we use SimGrid energy model

[7]https://www.rte-france.com/eco2mix

[8]experiment repository available at https://gitlab.irit.fr/sepia-pub/open-science/sufficient-behaviors-with-renewables (use the tag maelPhD)

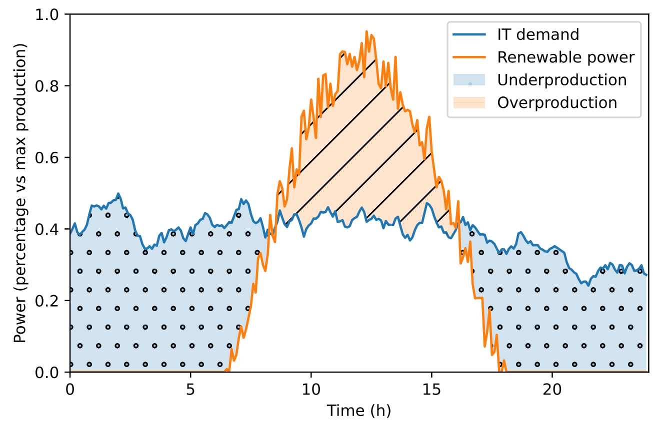


Figure 3: Underproduction occurs when the renewable power does not meet the IT demand. Overproduction is When IT demand is lower than the renewable power.

D. Metrics

From the simulation outputs, we compute both energy- and effort-related metrics. They are defined below and reported later in the Results (Section V).

1) Energy-related metrics: Batsim simulates the energy consumption of the infrastructure thanks to the underlying energy model from SimGrid. These logs are compared with the energy production data, identical for all simulations. In case the renewable production is in excess compared to the energy consumed, we say that we are in a phase of *overproduction*. On the contrary, when the production is insufficient to cover the energy consumption, we talk about *underproduction* (see Figure 3).

We compute on each experiment the following metrics:

- **energy total**: the energy consumed by the IT platform overall
- **energy red / yellow**: the energy consumed during red / yellow state windows
- **overproduction**: the excess renewable energy produced in phases of overproduction, that was not consumed by the IT platform (orange in Figure 3). It is typically sold to the grid or stored in batteries.
- **underproduction** (aka “brown energy”): the excess of energy consumed in phase of underproduction, that could not be produced by the renewable sources (blue in Figure 3). It has to be bought from the grid or drawn from batteries.

2) Effort-related metrics: In addition to energy, we want to report the effort made by the users when they adopt the various behaviors. The effort is a subjective quantity, that will not be experienced the same way by different users. All the same, we will report for each experiment:

- the number of jobs submitted without modification

	Reconfig	See You later	Degrad	Renounce
weight	25	50	75	100

Table II: **Weights for metric *weighted effort*** (without unit).

- the number of jobs that were reconfigured, degraded, renounced and delayed by a See You Later.

Additionally, we propose to aggregate all the above in a metric of *weighted effort*, where we associate to each behavior a weight, supposed to represent the inconvenience for the user. We follow the ranking of behaviors according to their "acceptability" proposed by Madon et al. [12] and choose arbitrary values, given in Table II. With N the total number of job, n_b the number of jobs affected by behavior b and w_b the weight associated to behavior b (Table II):

$$\text{weighted effort} = \sum_{\substack{b \in \{\text{Rigid,Reconfig,} \\ \text{See You,Degrad,Renounce}\}}} \frac{n_b \times w_b}{N} \qquad (1)$$

For example: if a job is delayed one time by a See You Later and then degraded, the effort for this job will be counted as $125/N$. Note that this is more than a single Renounce ($100/N$), since we assume that it is less cumbersome for a user to simply Renounce a job than to connect to the platform a first time, realize that the state is red, decide to come back later, realize that the state is still red and then decide to Degrad.

To give an idea, summed over all jobs of a simulation, a *weighted effort* of 0 corresponds to submitting all the jobs without modification (100% Rigid) and a *weighted effort* of 100 corresponds to renouncing all jobs (100% Renounce).

V. Results

The data presented in this section are obtained following the experimental campaign. They are based on Batsim scheduling and energy outputs in addition to custom user behavior logs, on which we computed the metrics described before.

A. State Window distribution

To get an understanding of the distribution of green, yellow and red state windows in the experiments, Figure 4 shows a typical week (left graph). The window distribution follows day-night cycle with green state during the day, red state during the night and yellow state in-between. This reflects a classic pattern for photovoltaic production. One of the days is yellow due to the lack of sun. It is also possible to have one or more full red days. Also, the length of the yellow windows is small compared to the green and red ones.

On the full experiment of nearly 6 months duration, the distribution of state window is given, aggregated by hour of the day, in the right graph of Figure 4. The day/night pattern is confirmed as green states are always starting after 6:00 and stopping before 18:00, whereas red state are very rare between 19:00 and 15:00. The exact distribution would vary depending on the season and geographic location of solar panels, but the overall shape should be similar.

Red state is the most common state, accounting for 56% of the experiment duration. Green state comes in second (36% of experiment duration). Yellow is the rarest state, appearing less than 8% of the time. Also, yellow states typically last at most one hour (85%), and a yellow state last 6 consecutive hours at maximum. Concerning red states, they typically last between 11 hours and 16 hours (80%). 10% of red state lasts less than 11 hours. Only once did a red state last up to 23 hours.

B. Energy consumption

The results concerning energy and user behavior are presented in Tables III and IV. For each metric, the mean and standard deviation σ are computed on the $n = 30$ replicates of each experiment. The accuracy displayed in the tables is calculated by the formula $3\frac{\sigma}{\sqrt{n}}$ and corresponds to a confidence interval of 99.7%, if we suppose normality of the outputs.

Table III focuses on the energy-related metrics.

Energy savings. First, we see that by adopting sufficiency behaviors, users were successful to cut the energy consumption during critical periods: the energy consumed during red and yellow windows decreases with the effort α. The relationship is linear: a Pearson correlation analysis between α and energy red / yellow gives coefficients of -0.992 and -0.965, respectively.

This is also the case with the underproduction, which is the quantity that we seek to minimize. Here again, the relationship is linear (Pearson coefficient -0.991).

We see that the variability of our results is very small: the confidence intervals are close to the mean. This is due to the size of our workload (almost 700000 jobs) and the 30 replicates for each experiment, which greatly limit the effect of randomness.

Influence of yellow states. Concerning the yellow states, their presence improves the energy-related metrics. In every behavior scenario, the version with yellow state yields better results for every energy-related metrics than the version without yellow states.

Remarks. Firstly, it is important to note that even if *full renounce red* is the best scenario in terms of energy, it does not lead to zero energy consumption in red windows. This is due to the inertia in the system. Already running tasks and tasks already in the queue are not impacted by this behavior, because the users already submitted them. This inertia phenomenon was already discussed in related work [12], where authors called the baseline energy consumption "residual mass" and the consumption caused by jobs submitted inside the window "fluid mass".

Secondly, we note that overproduction of energy is at least twice as big as underproduction, and similar to the energy consumed overall. It means that the electrical infrastructure is oversized and produces more energy than needed by the IT infrastructure. This is linked to the fact the experiments are done in summer and fall, while the electricity sizing was made over the whole year, taking into account battery efficiency.

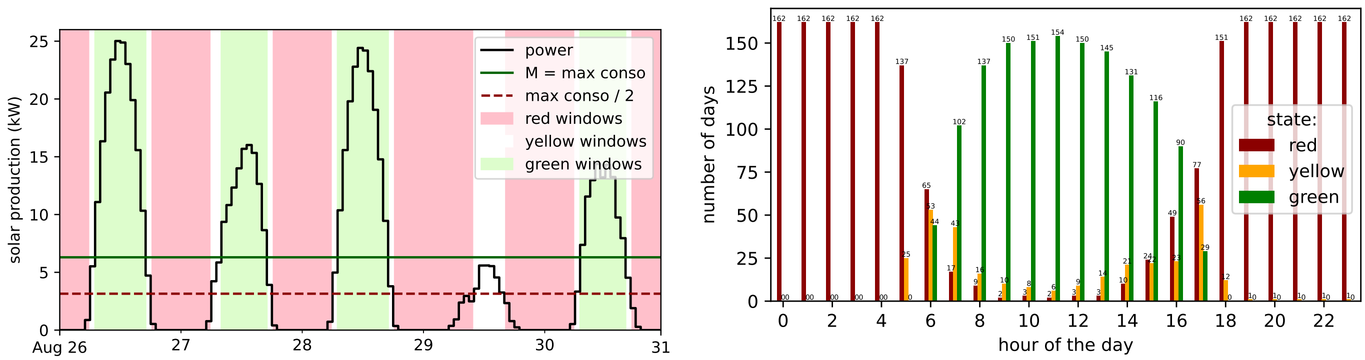


Figure 4: **Window state distribution in the inputs.** *Left graph:* data from 5 days of solar irradiation in Toulouse, August 26-30, 2019, one-hour time step. The two thresholds defining the windows are displayed as horizontal lines. *Right graph:* over the 162 days of the input trace, as a function of the time of the day.

user effort	yellow?	energy red	energy yellow	energy total	overproduction	underproduction
full rigid ($\alpha = 0$)		8458	1145	15143	16532	8098
low ($\alpha = 0.25$)	no	8194 $\pm$ 4	1124 $\pm$ 1	14816 $\pm$ 5	16609 $\pm$ 3	7843 $\pm$ 4
low ($\alpha = 0.25$)	yes	8165 $\pm$ 6	1120 $\pm$ 2	14779 $\pm$ 8	16616 $\pm$ 4	7813 $\pm$ 6
medium ($\alpha = 0.5$)	no	7882 $\pm$ 7	1099 $\pm$ 2	14429 $\pm$ 9	16695 $\pm$ 4	7539 $\pm$ 7
medium ($\alpha = 0.5$)	yes	7827 $\pm$ 7	1089 $\pm$ 2	14361 $\pm$ 8	16705 $\pm$ 4	7480 $\pm$ 6
big ($\alpha = 0.75$)	no	7503 $\pm$ 7	1073 $\pm$ 2	13969 $\pm$ 10	16794 $\pm$ 4	7174 $\pm$ 7
big ($\alpha = 0.75$)	yes	7422 $\pm$ 8	1053 $\pm$ 2	13862 $\pm$ 11	16815 $\pm$ 6	7087 $\pm$ 8
max ($\alpha = 1$)	no	7019 $\pm$ 8	1047 $\pm$ 2	13400 $\pm$ 11	16910 $\pm$ 5	6717 $\pm$ 7
max ($\alpha = 1$)	yes	6915 $\pm$ 8	1020 $\pm$ 2	13253 $\pm$ 11	16943 $\pm$ 5	6602 $\pm$ 7
full renounce red		5620	944	11683	17308	5481

Table III: **Mean energy metrics** in kWh, confidence interval 99.7%

Lastly, we see that overproduction increases with α. This is an unwanted effect, since excess renewable energy production will then have to be stored or sold to the electricity grid. It may also appear counter-intuitive because periods of overproduction are also "green states". In fact, this is again due to the inertia in the system. Degrad and Renounce tend to decrease the "fluid mass", i.e., the load in the infrastructure due to jobs submitted inside red windows. But this fluid mass partly overflows to periods of overproduction, where it gets equally decreased. See You Later and Reconfig should in principle counter-balance this effect by moving load from underproduction to overproduction periods. Overall, we note that overproduction increases less than underproduction decreases, which is a reassuring result.

C. User effort

Table IV focuses on user-effort-related metrics.

Increasing effort. Similarly to energy-related metrics, we observe a strong correlation between the effort probability α and the user effort metrics. This is directly due to the definition of α in our model, which corresponds to the proportion of jobs that will be modified in red and yellow states. The metric *weighted effort*, as a linear combination of the others, is no exception. The Pearson correlation coefficient between α and *weighted effort* is 0.993.

Once again, the results feature a very small variability thanks to the size of our data.

From this table, we also notice that Reconfig is the least used behavior, with only 2.7% reconfigured jobs at most. This is due to the large proportion (77%) of one-core jobs in our log, for which the reconfiguration is not available (see decision chart in Figure 2).

Among all the user effort scenarios, *full renounce red* is the one that requires the most effort, if we look at the number of renounced jobs. However, the scenario *max effort* with yellow states has a higher number of modified jobs, because it leads to the modification of all jobs in the red *and* yellow phases. With the weights of Table II, *max effort* scenarios with and without yellow states result in a bigger weighted effort than *full renounce red*.

Influence of yellow states. Overall, taking into account yellow state increases user effort. For the same effort scenario, the presence of yellow state increases the metrics *degraded jobs* and *reconfigured jobs*, and decreases the number of unmodified jobs. As behaviors Renounce and See You Later are not available in yellow states, the number of *renounced jobs* and *see you later jobs* is not modified.

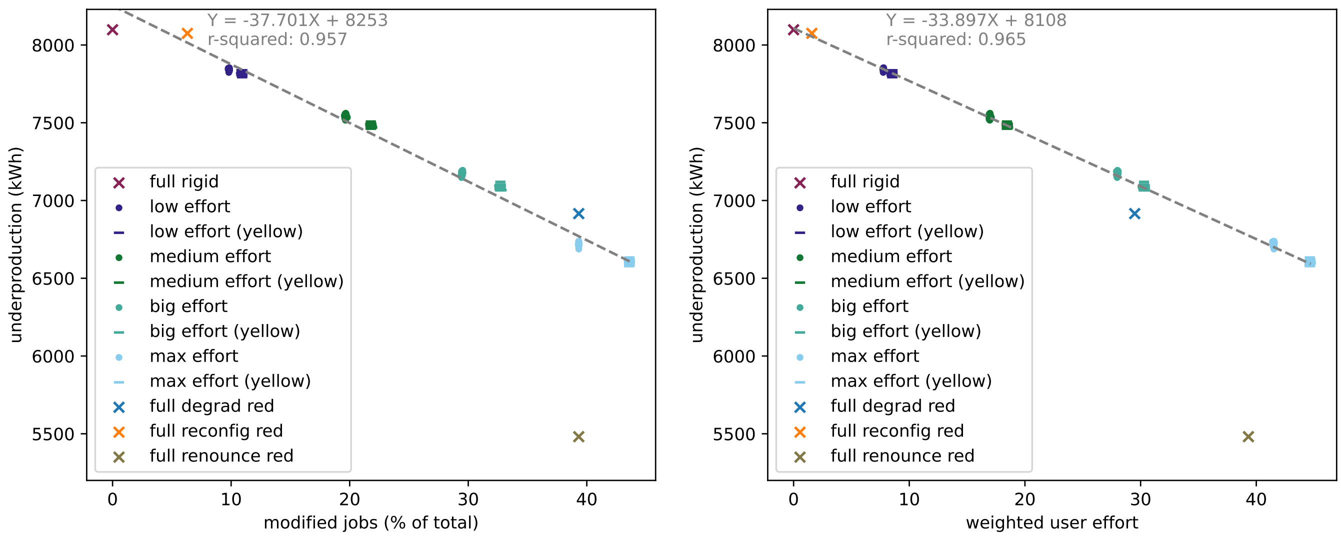
VI. Discussion

A. Trade-off between energy and effort

We plot in Figure 5 the energy gains as a function of user effort. Both graphs use the energy metric *underproduction*, but Figure 5a expresses effort as the number of modified jobs, whereas Figure 5b uses the weighted effort metric. In both

user effort	yellow?	unmodified jobs	renounce jobs	degraded jobs	reconfigured jobs	see you later jobs	weighted effort
full rigid ($\alpha = 0$)		100 %	0 %	0 %	0 %	0 %	0
low ($\alpha = 0.25$)	no	90.2 $\pm$.02 %	3.4 $\pm$.02 %	3.4 $\pm$.02 %	0.4 $\pm$.01 %	3.4 $\pm$.02 %	7.8 $\pm$.02
low ($\alpha = 0.25$)	yes	89.1 $\pm$.02 %	3.4 $\pm$.01 %	4.3 $\pm$.02 %	0.6 $\pm$.01 %	3.4 $\pm$.02 %	8.5 $\pm$.02
medium ($\alpha = 0.5$)	no	80.3 $\pm$.03 %	7.4 $\pm$.02 %	7.4 $\pm$.02 %	0.9 $\pm$.01 %	7.4 $\pm$.02 %	17.0 $\pm$.03
medium ($\alpha = 0.5$)	yes	78.2 $\pm$.02 %	7.5 $\pm$.02 %	9.3 $\pm$.02 %	1.2 $\pm$.01 %	7.5 $\pm$.02 %	18.5 $\pm$.02
big ($\alpha = 0.75$)	no	70.5 $\pm$.03 %	12.3 $\pm$.02 %	12.3 $\pm$.02 %	1.4 $\pm$.01 %	12.3 $\pm$.03 %	28.0 $\pm$.03
big ($\alpha = 0.75$)	yes	67.3 $\pm$.03 %	12.3 $\pm$.02 %	15.3 $\pm$.03 %	2.0 $\pm$.01 %	12.3 $\pm$.04 %	30.3 $\pm$.03
max ($\alpha = 1$)	no	60.7 $\pm$.00 %	18.2 $\pm$.03 %	18.2 $\pm$.03 %	2.1 $\pm$.01 %	18.2 $\pm$.05 %	41.5 $\pm$.03
max ($\alpha = 1$)	yes	56.4 $\pm$.01 %	18.2 $\pm$.03 %	22.2 $\pm$.03 %	2.7 $\pm$.01 %	18.2 $\pm$.04 %	44.6 $\pm$.02
full renounce red		60.7 %	39.3 %	0 %	0 %	0 %	39.3

Table IV: **User effort metrics**, in percentage of the total number of jobs in the workload (except the last column, without unit), confidence interval 99.7%. Note that the sum of columns 3 to 7 is not 100%, because a job that was delayed by a See You Later will be counted both as See You Later and as its final behavior, which can be rigid, Degrad, Reconfig or Renounce.



(a) "effort" in number of jobs modified (Table III, 3rd column)

(b) "effort" expressed with weighted effort (Table IV, last column)

Figure 5: **Underproduction** (Table III, last column) **as a function of user effort.** Each point represents an experiment. Scenarios *full degrad red*, *full reconfig red* and *full renounce red* are given for comparison, and are *not* included in the linear regressions. *Full renounce red* gives a higher bound on energy saving achievable (-32.3% compared to *full rigid*).

cases, we can see that user effort and gains in underproduction counter-balance each other, and the relationships between these two quantities are linear.

In the previous section, we observed that energy consumption in red and yellow phases was linearly correlated to the effort parameter α. This followed almost directly from the parameter's definition. In the present case, linearity between underproduction gains and user effort do not directly derive from the definition of α, but demonstrate that our 3-state energy approach was conclusive to transform efforts in red/yellow phases to brown energy savings.

As a result, we can answer our first research question:

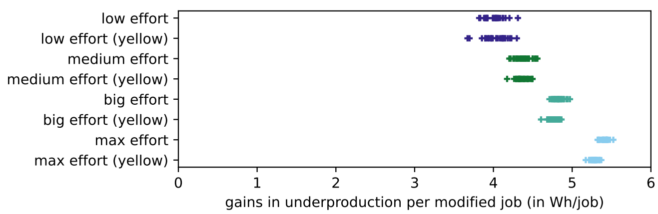
(1) The sufficiency behaviors adopted by the users allow to reduce the underproduction, i.e., the energy consumption that could not be matched by renewable production. The energy savings are linear with the size of the effort, with a maximum effort giving an energy saving of -18.4% compared to no effort.

There is no threshold after which the effort provided does not result in a fair amount of energy savings. To obtain a balanced level between user effort and energy savings, one has to set either a maximum effort acceptable or a minimum energy saving wanted.

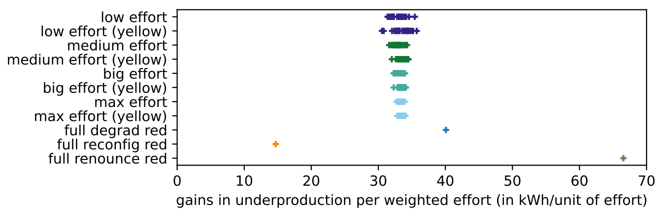
B. Relevance of yellow states

From Figure 5, we observe that adding yellow windows helps further reducing the underproduction, while also increasing the effort. It is not surprising, since scenarios with yellow windows are scenarios with red windows to which extra phases of user effort are added. A higher number of jobs gets modified, lowering their energy consumption (Reconfig and Degrad), hopefully anticipating for red phases to come. The relevant question to ask is: have the yellow states made it possible to reduce underproduction at a lower marginal cost in effort?

To answer this question, we plotted the gains in underproduction per unit of effort in Figure 6. We see that experiments



(a) user effort expressed in number of jobs modified



(b) user effort expressed with the weighted effort metric

Figure 6: **Ratio between gains in underproduction** (compared to *full rigid*) **and user effort.** A high ratio indicates good gains per unit of effort. *Full rigid* is not displayed as its ratio is undefined (0 modified jobs, 0 weighted effort). The other "full" scenarios are given for comparison.

with yellow windows yield lower gains per modified job (6a) than their red-only counterparts. In other words: contrary to our expectations, yellow windows do not succeed to bring some extra gains at the cost of a low effort on this metric. However, with the metric weighted effort (6b), we see no difference between scenarios with and without yellow windows, probably because the behaviors adopted during yellow windows are lower-effort behaviors for this metric.

We can now answer our second research question:

(2) On average, adding yellow states helps further reducing energy consumption in all the metrics considered. But yellow states can be considered as an additional effort, and they result in additional savings of the same scale, if not slightly smaller, as other efforts.

C. User incentives and weighted effort metric

Interestingly, we note in Figure 6a that the marginal gains increase with α (low to max effort). This is an indication that "the more people who make an effort, the greater the impact of a user's additional effort". We will have to see if this observation is confirmed by other data (workload, scheduler) of if it is specific to our inputs.

In any case, since brown energy savings and effort are correlated, it is important to motivate users to adopt the sufficiency behaviors during the critical periods. A possible incentive could be a financial reward for the effort made. For this, the weighted effort metric can prove very useful. In fact, we observe in Figure 6b that this metric is almost perfectly proportional to the gains. The coefficient of proportionality is $\gamma = 33.9$ kWh/unit of effort (slope of the line in Figure 5b). Renouncing a job corresponds to a weighted effort of w_{Renounce}/N (Equation 1). By multiplying by the coefficient of proportionality, it derives that renouncing a job allows reducing the underproduction by $\gamma * w_{\text{Renounce}}/N = 33.9 * 100/69306 = 0.00489$ kWh on average. This is consistent with the average length and size of jobs in the workload, and the power consumption of machines. We could imagine giving to the user a reward equivalent to the cost of the electricity saved that way.

VII. Limitations

Our study has a number of limitations that are discussed in this section.

A. Limits of the model

1) User hypothesis for the modeling: We assumed that users have some technical knowledge and control on the jobs they send. Otherwise, they could not apply the sufficiency behaviors on the jobs they send. Depending on the type of data center, this hypothesis might not hold. For example, if a data center runs mainly automated jobs (e.g., automatic testing, critical services), there is no room for users to decide on the way to run the jobs.

2) Behaviors Degrad and Reconfig: We also assumed that the jobs can always be degraded. However, this is only true for some jobs, e.g. convergence-based algorithms where the convergence criterion can be tuned or video transcoding where the quality can be lowered. Moreover, users might have to make timely changes in the application to modify the number of cores required (e.g., changing some parts of the code or configuration files), which makes Degrad and Reconfig not realistic for real-time response. Improvement of the model is needed to take this into account.

3) Behavior See You Later: The behavior See You Later can create some abnormalities in the submission behavior. First, job submission order is not preserved by this behavior. For example, if a user submits $job1$ at 13:30 and $job2$ at 14:00 in the original workload, a See You Later on $job1$ and a Rigid on $job2$ will lead to a change in the order of the two jobs. Moreover, it could lead to shifting submission time to a time of the day when the user is usually not connected to the platform. For example, we observe simulated users submitting late at midnight when they typically stop submitting after 18:00 in the original workload.

B. Limits of the experiments

1) Equiprobability of every non-rigid behavior: In the experimental campaign, the behaviors are drawn at random, assuming that each non-rigid behavior has the same probability (for a fixed window state and number of cores, see Figure 2). This has a number of drawbacks:

- it might not reflect the real popularity of each behavior,
- different users might have different preferences, and
- the choice of behavior is likely influenced by the nature of the job (criticality, size, difficulty to reconfigure, ...).

However, there is no data available on the popularity of each behavior, and we have no information on the nature of the jobs in the input workload (i.e., which ones are critical or difficult to reconfigure). Consequently, doing a randomized campaign with equiprobable behaviors seemed the most reasonable method to test the potential of our approach experimentally. New studies are needed to explore how the mix of probabilities impacts both the results and the link between effort and gains.

2) Limited experimental conditions: The experimental campaign presented in this paper explores a rather limited set of parameters. First, current input data only include one IT workload (MetaCentrum 2), characteristic of a High Performance Computing infrastructure. It would be interesting to see how the approach can be adapted to other types of data centers, like cloud infrastructures. Similarly, we used only one renewable energy trace, in the period from June 1 to November 30, 2019, which means that two seasons (winter and spring) are not included. Besides, we only looked at solar energy as a renewable source, one sizing for the IT and electrical infrastructure and have not included non-IT energy consumption of the data center in the model. Finally, the simulations are run with only one scheduling policy (bin-packing with greedy machine shut-down), which is rather naive and not state-of-the-art.

However, the focus of this paper is to estimate the potential of *user behaviors*, and does not aim at evaluating the exact gains in all possible configurations. We argue that the results would be similar with other sets of workload/platform/scheduler. Since our code is open-source and our experiments reproducible, it would be easy to verify it in the future.

VIII. Future works

The promising results of our approach together with its current limitations inspire possible extensions for this study.

Thresholds for red/yellow/green states. In our approach, the state window distribution is defined with thresholds on energy production. While this has the advantage of being simple and actionable, it might not be the best suited if the objective is to minimize underproduction. States could rather be defined on instantaneous power consumption, with red states when consumption is above production, green state when production is in excess and yellow states in-between.

Additionally, one could use weather forecasts to *anticipate* energy shortages. For example, yellow states could be used to that end, as a pre-warning before a red state. Users would be encouraged to submit jobs that would finish before the red state, in order to limit energy consumption in that period.

Other user behaviors. To the best of our knowledge, this article is, to date, the one studying the most user levers together. However, more behaviors could be included. For example, See You Later, Degrad and Reconfig could be made parametric on the delay time or scaling factor, respectively. Another behavior of interest is to allow users to checkpoint (stop and resume) their running jobs. This would allow the potential energy savings to go beyond the "fluid mass" [12].

Energy-state scheduler. For now, only the user is informed of the window state. A scheduler which uses different strategies depending on the window state could also be implemented with: a performance-driven strategy on green state; a balance between power-saving and performance in yellow state; and a power-saving strategy in red state (shutdown machines even if jobs are still pending in the queue…).

More realistic replay method. Replaying a recorded workload the way we do in this paper – and in the vast majority of similar works – has been criticized in the literature [26]. Indeed, we limit ourselves to replaying the jobs at their original timestamps of submission as in the recorded log. The problem is that this does not necessarily preserve the logic behind the user submissions, and possible dependencies between them. A feedback system could be implemented for users to wait for their previous jobs to finish before submitting new ones. User model could also include information on work hours. This would altogether solve the issues mentioned with See You Later (see VII-A3).

Impact of eco-feedback. Our approach consists in communicating simple information to users to encourage them to adopt eco-responsible behaviors, an approach sometimes called "eco-feedback". More research is needed to understand the actual effect of this eco-feedback on data center users, as it has been done for example in the software engineering field [27]. This would allow to empirically define the proportion of them who are ready to renounce, degrade, etc.

IX. Conclusion

This paper introduces a new model for renewable-energy-aware user behaviors.

A three-state energy feedback mechanism informs users on the status of renewable production: green when production is abundant, red when production is low, and yellow in-between. Five users behaviors (Reconfig, Space Degrad, Time Degrad, See You Later and Renounce) are considered and evaluated to reduce the energy consumption during critical times by modifying the characteristics of jobs submitted.

Experiments show that the approach is conclusive and allows reducing brown energy consumption. Energy gains are proportional to the efforts made by users. With the current model, yellow states have no particular added value: they increase the gains in exchange for a similar cost in effort.

Future work could focus on refining the feedback mechanism to base it on the difference between production and consumption in real time, and to introduce state forecast to allow anticipation. The actual willingness to adopt the behaviors could also be investigated.

Acknowledgments

Experiments presented in this report were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

CRediT author statement

Conceptualization, M.M. and G.d.C.; Methodology, J.G.; Software, J.G.; Validation, M.M.; Formal analysis, J.G. and

M.M.; Investigation, J.G.; Data Curation, J.G. and M.M.; Writing - Original Draft, J.G.; Writing - Review & Editing, M.M. and G.d.C.; Visualization, J.G. and M.M.; Supervision, M.M. and G.d.C.

References

[1] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday, "The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations," *Patterns*, vol. 2, no. 9, Sep. 2021. [Online]. Available: https://doi.org/10.1016/j.patter.2022.100576

[2] ADEME and ARCEP, "Evaluation de l'impact environnemental du numérique en France et analyse prospective," Ademe et ARCEP, Note de synthèse, Jan. 2022. [Online]. Available: https://librairie.ademe.fr/consommer-autrement/5226-evaluation-de-l-impact-environnemental-du-numerique-en-france-et-analyse-prospective.html

[3] G. Cook, J. Lee, T. Tsai, A. Kong, J. Deans, B. Johnson, and E. Jardim, "Clicking clean: Who is winning the race to build a green internet?" Greenpeace Inc, Tech. Rep., Jan. 2017. [Online]. Available: https://www.greenpeace.de/sites/default/files/publications/20170110_greenpeace_clicking_clean.pdf

[4] Google, "Data Centers: Belgian Site Adds Solar - Google Sustainability," Sep. 2018. [Online]. Available: https://sustainability.google/operating-sustainably/stories/belgium-solar/

[5] Z. Iverson, A. Achuthan, P. Marzocca, and D. Aidun, "Optimal design of hybrid renewable energy systems (HRES) using hydrogen storage technology for data center applications," *Renewable Energy*, vol. 52, pp. 79–87, Apr. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148112006830

[6] 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, Jul. 2019. [Online]. Available: http://doi.org/10.1109/ACCESS.2019.2930368

[7] L. M. Hilty and B. Aebischer, "Ict for sustainability: An emerging research field," in *ICT Innovations for Sustainability*. Springer, 2015, pp. 3–36. [Online]. Available: https://doi.org/10.1007/978-3-319-09228-7_1

[8] V. Coroama and F. Mattern, "Digital Rebound – Why Digitalization Will Not Redeem Us Our Environmental Sins," in *ICT4S 2019: International Conference on ICT for Sustainability*, Lappeenranta, Finland, 2019, p. 10. [Online]. Available: https://ceur-ws.org/Vol-2382/ICT4S2019_paper_31.pdf

[9] IPCC, "Climate Change 2022: Mitigation of Climate Change. Summary for policymakers." IPCC, Tech. Rep. 6, 2022. [Online]. Available: https://www.ipcc.ch/report/ar6/wg3/downloads/report/IPCC_AR6_WGIII_SPM.pdf

[10] T. Santarius, J. C. T. Bieser, V. Frick, M. Höjer, M. Gossen, L. M. Hilty, E. Kern, J. Pohl, F. Rohde, and S. Lange, "Digital sufficiency: Conceptual considerations for ICTs on a finite planet," *Annals of Telecommunications*, May 2022. [Online]. Available: https://doi.org/10.1007/s12243-022-00914-x

[11] C. Decker, "Pourquoi la sobriété numérique est au cœur des offres de TeleCoop," Jan. 2021. [Online]. Available: https://telecoop.fr/blog/pourquoi-placer-la-sobriete-numerique-au-coeur-des-offres-de-telecoop

[12] M. Madon, G. Da Costa, and J.-M. Pierson, "Characterization of Different User Behaviors for Demand Response in Data Centers," in *Euro-Par 2022: Parallel Processing*, ser. Lecture Notes in Computer Science, J. Cano and P. Trinder, Eds. Cham: Springer International Publishing, Aug. 2022, pp. 53–68. [Online]. Available: https://doi.org/10.1007/978-3-031-12597-3_4

[13] A. Orgerie, L. Lefèvre, and J. Gelas, "Save Watts in Your Grid: Green Strategies for Energy-Aware Framework in Large Scale Distributed Systems," in *2008 14th IEEE International Conference on Parallel and Distributed Systems*, Dec. 2008, pp. 171–178. [Online]. Available: https://doi.org/10.1109/ICPADS.2008.97

[14] D. Guyon, A.-C. Orgerie, C. Morin, and D. Agarwal, "Involving users in energy conservation: A case study in scientific clouds," *International Journal of Grid and Utility Computing*, vol. 10, no. 3, pp. 272–282, Jan. 2019. [Online]. Available: https://doi.org/10.1504/IJGUC.2019.099667

[15] D. Guyon, A.-C. Orgerie, and C. Morin, "Energy - Efficient IaaS-PaaS Co-Design for Flexible Cloud Deployment of Scientific Applications," in *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, Sep. 2018, pp. 69–76. [Online]. Available: https://doi.org/10.1109/CAHPC.2018.8645888

[16] G. Rostirolla, L. Grange, T. Minh-Thuyen, P. Stolf, J. M. Pierson, G. Da Costa, G. Baudic, M. Haddad, A. Kassab, J. M. Nicod, L. Philippe, V. Rehn-Sonigo, R. Roche, B. Celik, S. Caux, and J. Lecuivre, "A survey of challenges and solutions for the integration of renewable energy in datacenters," *Renewable and Sustainable Energy Reviews*, vol. 155, p. 111787, Mar. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S136403212101056X

[17] Md. E. Haque, K. Le, Í. Goiri, R. Bianchini, and T. D. Nguyen, "Providing green SLAs in High Performance Computing clouds," in *2013 International Green Computing Conference Proceedings*, Jun. 2013, pp. 1–11. [Online]. Available: https://doi.org/10.1109/IGCC.2013.6604503

[18] M. S. Hasan, Y. Kouki, T. Ledoux, and J. Pazat, "Exploiting Renewable Sources: When Green SLA Becomes a Possible Reality in Cloud Computing," *IEEE Transactions on Cloud Computing*, vol. 5, no. 2, pp. 249–262, Apr. 2017. [Online]. Available: https://doi.org/10.1109/TCC.2015.2459710

[19] R. Basmadjian, J. F. Botero, G. Giuliani, X. Hesselbach, S. Klingert, and H. De Meer, "Making Data Centers Fit for Demand Response: Introducing GreenSDA and GreenSLA Contracts," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3453–3464, Jul. 2018. [Online]. Available: https://doi.org/10.1109/TSG.2016.2632526

[20] J. Froehlich, L. Findlater, and J. Landay, "The design of eco-feedback technology," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Atlanta Georgia USA: ACM, Apr. 2010, pp. 1999–2008. [Online]. Available: https://doi.org/10.1145/1753326.1753629

[21] P.-F. Dutot, M. Mercier, M. Poquet, and O. Richard, "Batsim: A Realistic Language-Independent Resources and Jobs Management Systems Simulator," in *20th Workshop on Job Scheduling Strategies for Parallel Processing*, Chicago, United States, May 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-61756-5_10

[22] D. Klusáček, Š. Tóth, and G. Podolníková, "Real-Life Experience with Major Reconfiguration of Job Scheduling System," in *Job Scheduling Strategies for Parallel Processing*, ser. Lecture Notes in Computer Science, N. Desai and W. Cirne, Eds. Cham: Springer International Publishing, 2017, pp. 83–101. [Online]. Available: https://doi.org/10.1007/978-3-319-61756-5_5

[23] I. Staffell and S. Pfenninger, "Using bias-corrected reanalysis to simulate current and future wind power output," *Energy*, vol. 114, pp. 1224–1239, Nov. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544216311811

[24] S. Pfenninger and I. Staffell, "Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data," *Energy*, vol. 114, pp. 1251–1265, Nov. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544216311744

[25] M. Benaissa, G. D. Costa, and J.-M. Nicod, "Standalone Data-Center Sizing Combating the Over-Provisioning of the IT and Electrical Parts," in *2022 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW)*, Nov. 2022, pp. 57–62. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9978502

[26] D. G. Feitelson, "Resampling with Feedback: A New Paradigm of Using Workload Data for Performance Evaluation," in *Job Scheduling Strategies for Parallel Processing*, ser. Lecture Notes in Computer Science, D. Klusáček, W. Cirne, and G. P. Rodrigo, Eds. Cham: Springer International Publishing, 2021, pp. 3–32. [Online]. Available: https://doi.org/10.1007/978-3-030-88224-2_1

[27] A. Noureddine, M. Dieguez Lodeiro, N. Bru, and R. Chbeir, "The Impact of Green Feedback on Users' Software Usage," *IEEE Transactions on Sustainable Computing*, pp. 1–14, 2022. [Online]. Available: https://doi.org/10.1109/TSUSC.2022.3222631