Latency, Energy and Carbon Aware Collaborative Resource Allocation with Consolidation and QoS Degradation Strategies in Edge Computing

Wedan Emmanuel Gnibga, Anne Blavette, Anne-Cécile Orgerie

Latency, Energy and Carbon Aware Collaborative Resource Allocation with Consolidation and QoS Degradation Strategies in Edge Computing

Wedan Emmanuel Gnibga*, Anne Blavette†, Anne-Cécile Orgerie*
*Univ. Rennes, Inria, CNRS, IRISA, Rennes, France, Email:{wedan-emmanuel.gnibga, anne-cecile.orgerie}@irisa.fr
†Univ. Rennes, ENS Rennes, CNRS, SATIE lab, Rennes, France, Email: anne.blavette@ens-rennes.fr

***Abstract*—Edge Computing has emerged from the Cloud to tackle the increasingly stringent latency, reliability and scalability imperatives of modern applications, mainly in the Internet of Things arena. To this end, the data centers are pushed to the edge of the network to diversify and bring the services closer to the users. This spatial distribution offer a wide range of opportunities for allowing self-consumption from local renewable energy sources with regard to the local weather conditions. However, scheduling the users' tasks so as to meet the service restrictions while consuming the most renewable energy and reducing the carbon footprint remains a challenge. In this paper, we design a nationwide Edge infrastructure, and study its behavior under three typical electrical configurations including solar power plant, batteries and the grid. Then, we study a set of techniques that collaboratively allocates resources on the edge data centers to harvest renewable energy and reduce the environmental impact. These strategies also includes energy efficiency optimization by means of reasonable quality of service degradation and consolidation techniques at each data center in order to reduce the need for brown energy. The simulation results show that combining these techniques allows to increase the self-consumption of the platform by 7.83% and to reduce the carbon footprint by 35.7% compared to the baseline algorithm. The optimizations also outperform classical energy-aware resource management algorithms from the literature. Yet, these techniques do not equally contribute to these performances, consolidation being the most efficient.**

***Index Terms*—Resources Scheduling, Renewable energy, self-consumption, consumption scaling, consolidation**

I. Introduction

The advent of Cloud Computing proved to be a great technological evolution, as it allowed machines to delegate their computational and storage load to remote data centers (DCs). From the first connected appliance experimented in 1990 by John Romkey, one could estimate the number of internet-connected objects in 2020 at 20 billion worldwide [1]. While they tend to be miniaturized for portability and discretion, they run more sophisticated and resource-demanding applications such as augmented reality, gaming, video streaming and artificial intelligence-based applications. The Cloud has become a bottleneck in terms of timeliness of the service, inducing a latency usually higher than 100ms which is not supportable in use cases like real-time video analytics [2]. Therefore, the Edge Computing paradigm has been designed to push the service provider to the edge of the network, close to the connected objects. It is materialized by the massive deployment of a few nodes to micro-DCs at various locations, near the end users. One of the greatest challenges is to build a scalable latency-efficient infrastructure with locality and application-type awareness, where the sites can make collaborative decisions while keeping a high level of independence.

The Edge concept also raises electrical and environmental concerns. In fact, the global DCs electricity use in 2020 was estimated to 200-250TWh, or approximately 1% of the world electricity consumption [3], and this can be accelerated by expanding them to the network edges. Fossil (such as natural gas, oil) energies are currently the main sources used to supply them at a global scale [3]. Consequently, they represent a source of greenhouse gas (GHG) emissions. Some efforts are being made to improve DCs energy efficiency and to progressively power them from renewable energies, of which one major constraint is intermittency. Given that most DCs powered from renewables use photovoltaic energy which may be variable during the day and is not generated during the night, a secondary source of power is required for continuous power supply. This source may be either a battery or the national/regional electrical grid. In this context, the question is how to allocate workload onto computing resources in order to reduce the footprint of the edge infrastructure.

Several techniques have been explored in the literature to decrease either the overall energy consumption or the carbon footprint of Edge infrastructures [4]. The geographical distribution of edge data centers and their solar panels can be leveraged to increase the renewable part of energy consumption through follow-the-sun techniques or location-aware resource allocation [5], [6]. Consolidation techniques combine energy-efficient scheduling algorithms with shut down strategies to consolidate the workload on the fewest number of servers and switch off unused ones [7], [8]. On a given server, dynamic voltage and frequency scaling (DVFS) techniques can also be employed to reduce the power consumption [9]. Yet, this latter technique may increase the runtime of CPU-intensive applications and can thus be contraindicated for latency-sensitive edge applications. While these various popular techniques have been extensively studied in the energy-efficient edge context, we propose to study another promising solution which has yet receive little attention: application performance degradation. Contrarily to DVFS, the runtime of the application is not

affected, but the quality of service (QoS) is. For instance, for a video streaming service, it means reducing temporarily the video resolution in order to save energy and to reduce the carbon footprint.

In this paper, we design a large-scale Edge Computing infrastructure powered from renewable energy sources associated with energy storage devices and the main electrical grid. We explore three techniques to optimize its power consumption: 1) reducing the power consumption by reasonably reducing the applications' QoS when necessary, 2) consolidating the workload to switch off unused servers, and 3) negotiation between nearby data centers to move jobs while limiting latency degradation. In particular, we examine whether performance degradation is useful in practice (without always opting for the lowest QoS) and whether it can be combined with the two other techniques. In order to provide a quantitative analysis of the gains reachable through each technique independently and all together, we chose to focus on an emblematic application of Edge computing infrastructures (and previously Content Delivery Networks): video streaming, video traffic representing nearly four-fifths of global mobile data traffic in 2022 [10].

Our main contributions are as follows:

- measuring the impact of performance degradation on the power consumption of video streaming servers through real measurements
- investigating the level of power consumption reduction available through performance degradation, consolidation and location-aware allocation, independently and jointly.
- quantifying the carbon footprint reduction and self-consumption increase available through performance degradation, consolidation and location-aware allocation, independently and jointly.
- investigating the impact of the electrical infrastructure (i.e. battery sizing) on these techniques' performance.

The rest of this paper is organized as follows. Section II presents recent work optimizing performance and energy management in Edge infrastructures. Section III describes the computing and electrical infrastructures considered in this study. Section IV presents the models of power consumption used to realize the study. Section V presents the methodology and algorithms and Section VI presents the algorithms performance.

II. Related Work

A. Edge infrastructures and performances optimization

Edge Computing is presented as gathering the concepts of Fog Computing, Cloudlets and Multi-Access Edge Computing [11], all distributed. Many contributions have been made with partially distributed infrastructures [12], [13] composed of two levels: an Edge layer that performs full processing or prepossessing of requests, and a Cloud layer to which the requests are forwarded when the Edge resources are insufficient. Others implement fully distributed infrastructures, made of edge clusters of few nodes [13]. Edge Computing is meant to address concerns with Cloud Computing such as cost, latency, bandwidth congestion, scalability, and privacy [4]. These features are necessary in critical applications such as autonomous driving, real-time video analytics, surveillance, virtual reality, real-time traffic monitoring [11]. Their deployment is mainly based on virtualization technologies [14], containers (e.g. Docker, unikernel) [15], and Software-Defined Network based solutions [16] in order to hide the physical complexities of the infrastructure to the end-users and applications. In large-scale networks, many candidate locations may host edge nodes. However, they may not lead to an optimal operation. Thus, several authors proposed to deploy edge servers in strategic locations (which may rely on existing Internet Service Providers' infrastructures) that optimize the access delay between the end users and edge servers [5], [17], [18]. Considering the small size of the edges sites and the multiplicity of applications to be run on them, it is beneficial to adequately distribute the instances of each application across the nodes through service placement optimization [19]. One approach is to use a centralized controller with a global knowledge of the infrastructure to balance the load on the suitable resources, yet providing limited scalability and introducing bandwidth congestion due to the volume of data exchanges. The second approach is a distributed resource scheduling that provides each site with a controller capable of coordinating with the others through peer-to-peer communications [14], at the inconvenience of losing the global knowledge of the system and adding communication overheads. Sahn *et al.* [4] wrote a comprehensible survey on these strategies. The controller can be designed to support and optimize the application requirements such as latency [20] and cost [21].

In this work, we rely on a fully distributed infrastructure and manage the task allocation in a fully distributed manner, thus ensuring scalability. We design a large-scale infrastructure in which the edge clusters are able to collaboratively exchange tasks without increasing significantly their latency.

B. Energy aware optimizations

Service providers expect a high energy efficiency and an economical attractiveness of their facilities, hence the necessity to achieve a dual computer-electrical Edge design. Jiang *et al.* [7] published a survey on energy-aware Edge Computing in which they show a set of operating systems, hardware and software level technologies allowing to reach high computing performances with a low power consumption. Over the system layer, one can enumerate energy-aware server [22] and virtual machine [23] placement scheme. Several techniques have also emerged to dynamically migrate tasks to the sites with the lowest energy cost and/or power consumption while maintaining a high-level QoS. In this direction, Suryadevara *et al.* [24] explored a range of machine learning algorithms and classifiers for managing and balancing load deployed on a Fog platform, in the sense of reducing latency and energy consumption in an IoT context. However, this work solely relies on the electrical grid, which, despite being permanent, is generally a source of pollution (with electricity mixes based on fossils).

Another variety of work aims to maximize the renewable energy consumption. These projects are part of broader decarbonization programs including data centers, such as the Green Deal [25]. Toor *et al.* [9] proposed a framework that dynamically adjusts the frequency of the compute nodes in a partially distributed infrastructure, following the on-site solar and battery energy availability. Karimiafshar *et al.* [6] used Lyapunov optimization techniques to dispatch the users' requests among the nearby fog nodes and remote DCs, increasing the on-site solar energy consumption. This type of multi-participant oriented renewable energy consumption is known as collaborative self-consumption [26].

In this work, we propose a two-phase methodology to increase the on-site power consumption and reduce the carbon footprint. We first use the heterogeneity and distribution of the Edge DCs' electrical configurations to balance the load in order to increase the collective self-consumption (i.e. to reduce exchanges with the electrical grid), while ensuring an acceptable latency for the running applications. In a second step, we reduce or differ the power consumption of the compute nodes by dynamically adapting the jobs' QoS and consolidating them on a minimal set of nodes.

III. System description

A. Edge infrastructure

The infrastructure is based on several edge DCs, referred to as Edge-DCs, linked together by a telecommunication network. We considered the national Internet service provider's network topology presented in [27]. It is a hierarchical network consisting of four layers of routers located in Points-of-Presence (PoP) distributed throughout the national territory. From the core of network to its edges, their are 8 core routers of 50Gbps with a 30ms latency, 52 backbone routers of 20Gbps with 20ms latency, 52 metro routers of 10Gbps with 15ms latency and and 260 feeder routers of 10Gbps with 5ms.In particular, the feeder routers receive and transmit data flows at the scale of a city or a set of districts. So the PoPs that host them represent an opportunity exploited in this work to place micro data centers in order to provide local computing services. We refer to them as edge data centers (Edge-DCs).

An edge data center contains a few racks of homogeneous compute nodes (servers) communicating through a wired intra-DC network. For sake of simplicity, all the Edge-DCs are assumed homogeneous, i.e they have the same number of compute nodes and the same internal network topology. We consider location awareness in order to reduce the transmission delays. Just like an ISP client has access to the Internet via the nearest PoP, the users are linked to the nearest Edge-DC to submit jobs and/or requests. The response is also received from the Edge-DC to which the user is linked.

Each Edge-DC has a local task manager which is a containerized process in charge of allocating resources to the submitted jobs, controlling the servers parameters and state, optimizing their power consumption, and coordinating with the neighboring sites to harvest renewable energy. In this study, we do not consider over-commitment, i.e the jobs allocated on a given server cannot exceed its physical capabilities. Inside Edge-DCs, we employ a redundant tree network topology using identical switches. As shown in Figure 1, the topology contains three layers of switches: core, aggregated and edge. This topology is suitable for the considered DCs size of this study and ensures a high redundancy, a key parameter in Edge-DCs to overcome bandwidth bottlenecks. In the example of Figure 1, each edge switch of 12 ports can handle 10 servers.

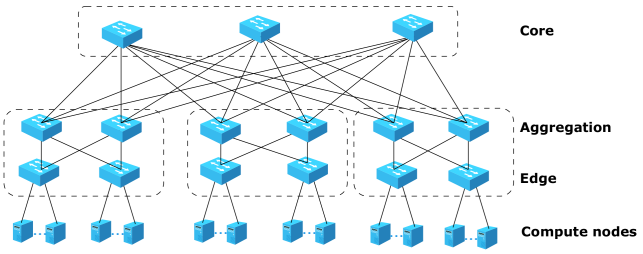


Figure 1: Intra-Data center network topology, each Edge router links 10 servers.

We consider a real-time applications in this work, precisely video-streaming. This type of applications is characteristic of the trend towards Edge Computing [10] and features several flexibilities for reducing the hosts power consumption.

B. Electrical infrastructure and Energy management policy

The Edge-DCs are connected to the electrical grid with which they can transact electrical power. The energy management strategy comprises two modes: normal operation and degraded operation. In this section, we only describe the normal operation and explain the degraded mode in Section V-C. The Edge-DCs are divided into three categories with respect to their electrical system, following a round robin policy. One third of them contain only on-site photovoltaic (PV) power plant, a configuration based on some service providers like Apple [28] that associate the electrical grid with renewables to ensure electrical security in much of their data centers. In the normal operating mode, these Edge-DCs first consume power from the PV plant and inject the rest to the grid if any. In the reverse, they import power from the grid to fill the shortfall. The second third of Edge-DCs is designed without any additional energy source (no PV nor battery). In fact, this is the configuration of many of today's data centers. In a normal mode, they permanently purchase electrical power from the grid to meet their load. The Edge-DCs of the last category are equipped with a battery and a PV plant. These Edge-DCs consume from the PV plant, then from the battery when the PV generation is insufficient and lastly from the electrical grid when the battery is too low. In the reverse, they use the surplus of PV generation (if any) to charge the battery and injects the rest to the grid if the battery gets fully charged.

Figure 2 summarizes the electrical infrastructure with three Edge-DCs, each representing one category. For an Edge-DC i, $P_{g,p,i}(t)$ and $P_{g,s,i}(t)$ represent the power imported and exported from/to the grid at time t respectively. However, an

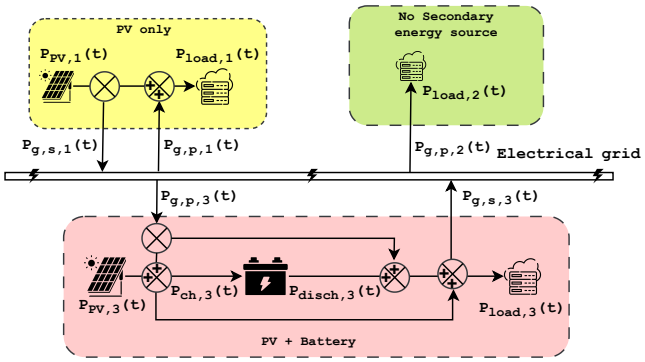


Figure 2: Electrical infrastructure with three categories of Edge-DCs.

edge-DC cannot import and export electrical power simultaneously. Moreover, the DCs are not collective self-consumption agents as per the French law considered in this paper, because their mutual distances exceed the maximum allowed limit of 2 km [29], since we focus on a nation-wide edge infrastructure. $P_{ch,i}(t)$ and $P_{disch,i}(t)$ stand respectively for the power of charge and discharge of the battery (when it exists on Edge-DC i). $P_{load,i}(t)$ is the power consumed by compute nodes and the intra-DC network. $P_{PV,i}(t)$ represents the power generated by the on-site PV plant (when it exists on Edge-DC i). The batteries and the PV plants are respectively identical in all the edge-DCs where they are present.

IV. Modeling power consumption of Edge-DCs

This section presents the models of the energy sources and the load. We adopt a classical time-slotted approach in which the time is divided into timestamps of duration Δt.

A. Edge-data centers power consumption

Three main components contribute to the energy consumption in the Edge-DCs: the compute nodes, the switches and the cooling system. The power consumption of a compute node is made up of two terms: a static consumption that accounts for the power consumed when the server is powered on but not running any load or running background activities, and a dynamic consumption that depends linearly on the CPU frequency and the nature of the compute load [30]. Adelin *et al.* [31] showed that a switch consumes more than 80% of its nominal power when it is switched on, including when there is no traffic (static state). Hence, we assume that the switches operate at their rated power that can be estimated using the Cisco Power Calculator [32]. Thus, the power consumed by the intra-DC network is assumed to be the power rating of a single switch multiplied by the number of switches required in the topology. This may represent a minor overestimation of the intra-DC network power consumption. In addition, we consider the power consumption of other equipment of the data center, such as the cooling system, facility lighting, etc. To do so, we employ the Power Usage Effectiveness (PUE). It represents the ratio between the total facility power consumption and the IT power consumption (i.e. server and switch power consumption). Thus, an Edge-DC overall power consumption can be estimated by multiplying the cumulative power consumption of the compute nodes and switches by its PUE value.

B. Workload power consumption analysis

A client job k submitted to the platform is modeled as a tuple $(t_{0,k}, dt_k, c_k, u_k, ram_k)$ where $t_{0,k}$ is the arrival date, dt_k is the service duration assumed to be known in advance, c_k is the number of cores required, u_k is the percentage of the requested cores usage which is assumed to be constant during the computation time, ram_k is the size of memory in Bytes required to load and run the application accessed by the job. We consider that the resource controllers are 4 core and 4GB processes using 100% of the resources. The number of jobs arriving at each Edge-DC follows Poisson distribution with a constant arrival rate $\lambda_{ar} = 5/mn$. The service duration dt_k follows an exponential distribution with a constant mean time $\lambda_{dt} = 30mn$. c_k, u_k and ram_k are random integer values respectively selected from the ranges [1 - 4], [5 - 100]%, $[10^9 - 4.10^9]$ Bytes following a normal law. By choosing randomly CPU and memory, we avoid jobs to be proportional in terms of CPU/RAM request and make the computing and energy patterns more realistic.

The workload power consumption can be considered as linearly dependent on its CPU usage [30]. For a video-streaming application, the CPU usage is a function of the images size streamed. Let $P_{\min}$ and $P_{\max}$ be a server's respective static and dynamic power consumption. Let also n be the number of cores in the server and $u_k(px, t)$ the CPU usage of the job k streaming images of size px (the default value is px_0 = 1980x1080) pixels. Equation 1 where $P_k(px, t)$ is the power of k can be derived from [30].

$$\frac{u_k(px,t)}{u_k(px_0,t)} = \frac{P_k(px,t) - P_{\min}}{P_k(px_0,t) - P_{\min}} \quad (1)$$

C. Electrical components models

A PV panel converts sunlight into electrical power through PV cells arrayed on the panel. The source generates power which value is the product of its surface, its conversion efficiency and the solar irradiance [33]. We consider an homogeneous solar irradiance on all the PV panels of the same plant. We also neglect the aging effect on the performance of the panels. Thus, the power generated by the PV plant is obtained by multiplying the number of panels by the power production of a single panel.

We consider a Lithium-ion battery located at Edge-DC i whose operation, degradation and economical models are further described in previous work [33]. The energy stored in it relative to its capacity is called its state-of-charge $SoC_i(t)$. The power of charge and that of discharge are respectively named $P_{ch,i}(t)$ and $P_{disch,i}(t)$.

V. Resource allocation and workload adaptation

The resource and energy management are fully decentralized: each Edge-DC has a controller that manages its local

resources. The controllers collaborate in order to increase the overall renewable energy consumption. The resource management strategy combining all the energy-efficient techniques contains 4 steps. First, each site allocates computing resources to the jobs submitted during the time slot (Algorithm 1), then negotiation between nearby Edge-DCs is initiated (Algorithm 2) in order to relocate some jobs to consume more on-site energy while limiting the response delays. Then the sites with a green energy deficit switch to a power saving mode by degrading the running and allocated containers' activity to reduce their power consumption (Algorithm 3). In the last step, the controllers consolidate the containers on the minimum number of compute nodes and switch off the empty ones if any (Algorithm 4).

A. Resource allocation

During time slot t, the controller of each data center i receives a set of jobs from its clients (assigned by their geographical proximity), allocates the required resources to them and evaluates the power consumption $P_{load,i}(t)$ of the Edge-DC for that time slot. The resource allocation follows a best fit policy as shown in Algorithm 1. The algorithm looks sequentially in the active nodes queue $L_{on,i}$, for the most loaded nodes with enough computing resources that can host each job k. When none is found among the active nodes, an idle node is switched on from the inactive nodes queue $L_{Off,i}$, to host the incoming job. In the same order, when no node is found in the Edge-DC i, the resource allocation is a failure, so the job k is kept in a pending queue $L_{pend,i}$, to be forwarded to another location in the context of collaboration.

Algorithm 1: Resources allocation phase

Input: i,k
Output: Allocation status
`Allocation(i, k)`
Sort $L_{on,i}$ by increasing free CPU and RAM
for $S \in L_{on,i}$ **do**
  if *free CPU of* $S \leq c_k$ & *free RAM of* $S \leq ram_k$ **then**
    Reserve resources on S and return $Success$
  end
  if *S is tail of* $L_{on,i}$ **then**
    if $L_{off,i}$ *not empty* **then**
      remove the first machine from $L_{off,i}$, add it to $L_{on,i}$. Switch on the machine, reserve the resources on it and return $Success$
    else
      Add the job to $L_{pend,i}$ and return $Failure$
    end
  end
end

B. Inter Edge-DCs Negotiation

An Edge-DC i negotiates in peer-to-peer manner with its neighborhood $\mathcal{N}_i$ (by increasing distance) to exchange jobs. Regarding the performance requirements, we limited the perimeter of job exchange in order to ensure that a forwarded job latency is contained within a predefined threshold L_0 (Eq. 2f). In this study, $\mathcal{N}_i$ is the set of Edge-DCs sharing the same tuple of metro router.

The negotiation consists in synchronously finding in $\mathcal{N}_i$ a set of Edge-DCs with available computing resources and renewable energy to host some jobs of i. Thus, the Edge-DCs collaboratively minimize their dependence on the electrical grid (Eq 2a) –imports and exports that are mutually exclusive (Eq. 2c) – as shown in Equation 2 and increases the self-consumption. Let's consider i with a deficit of on-site power. For i equipped with only a PV plant (first category), that corresponds to when the PV plant does not generate sufficiently to meet the load. When i has no PV nor battery (second category), the deficit is permanent. Lastly, for i having a PV and a battery (third category), the deficit happens when both sources cannot meet the load demand. The negotiation between i and $j \in \mathcal{N}_i$ follows this process : i provides a list of its pending and allocated jobs to j sorted by increasing power consumption. Then j attempts to allocate resources to each job (Algorithm 1) within the physical limits (Eq. 2b). Hence, j sends back a list of the non allocated jobs to i that releases the resources of jobs successfully deployed on j and continue the negotiation with other sites $\in \mathcal{N}_i$ if some jobs remain non forwarded. No simultaneous negotiation with several remote sites is allowed by the same Edge-DC.

$$\min_{P_{load,i,j}(t), P_{load,j,i}(t)} (P_{g,p,i}(t) - P_{g,s,i}(t)) \quad \forall j \in \mathcal{N}_i \tag{2a}$$

$$s.t$$

$$P_{PV,i}(t) + P_{disch,i}(t) + P_{g,p,i}(t) = P_{ch,i}(t) + P_{g,s,i}(t) + PUE.(P_{load,i}(t) + \sum_{j \in \mathcal{N}_i} (P_{load,j,i}(t) - P_{load,i,j}(t))) \tag{2b}$$

$$P_{g,p,i}(t).P_{g,s,i}(t) = 0 \tag{2c}$$

$$P_{load,i,j}(t).P_{load,j,i}(t) = 0 \tag{2d}$$

$$P_{load,i,j}(t) = 0 \text{ if } P_{PV,j}(t) - PUE.P_{load,j}(t) < 0 \tag{2e}$$

$$L_{k,i,j} \leq L_0 \tag{2f}$$

Where $P_{load,i,j}(t)$ and $P_{load,j,i}(t)$ are respectively the power load forwarded from site i to j and from site j to i, $P_{g,p,i}(t)$ and $P_{g,s,i}(t)$ are respectively the amount of power imported and exported from/to the grid, $L_{k,i,j}$ is the latency of job k submitted to site i and forwarded to site j.

Algorithm 2: Negotiation algorithm

Input: i , $\mathcal{N}_i$
`Negociation(i , N_i)`
SavePendJobs ← false
for $j \in \mathcal{N}_i$ *with priority on the categories* **do**
  $L_{pend,i,j} \cup L_{alloc,i,j}$ ← Send $L_{pend,i} \cup$ Allocated Jobs
  for $k \in L_{in,i,j}$ **do**
    if *Allocation(j, k) with success* **then**
      if $P_{PV,j}(t) > PUE.P_{load,j}(t)$ *OR* $SoC_j(t) > SoC_L$ *OR SavePendJobs=true* **then**
        Remove k from $L_{pend,i,j} \cup L_{alloc,i,j}$ and continue with next job
      else
        Release the Resources allocated on j
      end
    end
  end
  Send back $L_{pend,i,j} \cup L_{alloc,i,j}$ to i
end
if $L_{pend,i,j}$ *not empty* **then**
  SavePendJobs ← true
  Repeat $Negociation(i, \mathcal{N}_i)$
end
Release the resources on i previously allocated to the forwarded jobs.

During day time (8am to 6pm), all the negotiations are first held with the sites equipped with only a PV plant. At night,

this step is skipped as the weather conditions do not allow PV power generation. In a second instance, i may forward jobs to the sites of third category with a surplus of PV generation or a battery state of charge above a predefined threshold SoC_L (Eq. 2e) that represents a security margin. If after the negotiations some pending jobs remain non scheduled, the controllers proceed another round and resources are allocated on the Edge-DCs with enough computing resources, regardless the on-site energy status. However, the search is still carried out in the following order: 1st (PV only), 3rd (PV and battery) and 2nd (no onsite source) categories of DCs, with a perspective of increasing on-site consumption later on. In order to minimize the traffic in the telecommunication network, jobs are only sent in one direction, i.e from the deficient Edge-DC (in our example, from site i to j) as presented in Equation 2d. The negotiation process is summarized in Algorithm 2.

C. Performance degradation

Performance degradation allows Edge-DCs with energy production deficit (or no production), to reasonably lower their QoS and to reduce their consumption of brown energy. One can use hardware level service degradation called Dynamic Voltage and Frequency Scaling [30]. However, it extends the duration of the jobs and consequently increases the probability for the servers to operate for longer, which could offset the energy savings. Here, we dynamically adjust the images size to degrade the quality of the video in order to reduce the energy consumption. The degraded mode is used in the following situations according to the edge-DC category: 1) with a PV plant and a battery, the degradation intervenes when there is a deficit of PV generation and the battery is below the security margin ($P_{PV,i}(t) < P_{load,i}(t)$ and $SoC_i(t) < SoC_L$), 2) with only a PV plant, the degraded mode is used when the PV plant does not generate sufficiently ($P_{PV,i}(t) < P_{load,i}(t)$) and 3) with no additional energy source, the degraded mode is always active. In future works, more moderate requirements will be explored (arbitration for instance between cost and on-site generation).

To guarantee that the performance degradation does not cause too much discomfort to the end-users, the service provider concludes a Service Level Agreement (SLA) with them to define the expected QoS and the efforts they are willing to make to enjoy a decarbonized service (green-SLA). For real-time applications such as video-streaming, the green-SLA defines for a given job, the minimum resolution ($px_{\min}$) bearable by the user. To simplify, we consider that all the jobs are subject to the same green-SLA. For sake of generality, let's consider an Edge-DC with a battery and a PV plant. The degraded mode is represented as an optimization problem which aims to simultaneously minimize the battery use (it degrades), the power importations and the performance degradation ($P_{shed,i}(t)$) as represented in Equation 3a. Equation 3b represents the energy balance and Equation 3c stands for the QoS requirement of each job.

$$\min_{P_{disch}(t),P_{g,p}(t),P_{shed,i}(t)} \alpha P_{disch}(t) + \beta P_{g,p}(t) + \gamma.P_{shed,i}(t) \quad (3a)$$

$$s.t$$

$$PUE.P_{load,i}(t) - P_{shed,i}(t) = P_{disch,i}(t) + P_{PV,i}(t) + P_{g,p,i}(t) \quad (3b)$$

$$px_k(t) \geq px_{\min} \quad (3c)$$

Where $\alpha, \beta, \gamma \leq 1$ are variable coefficients with $\alpha + \beta + \gamma = 1$. We consider $\beta = \frac{CI(t)}{CI_0}$ where $CI(t)$ (kgCO$_2$/kwh) is the carbon intensity of the grid electricity, CI_0 a carbon intensity value above which no energy may be imported from the grid as it is heavily carbonized.

We propose heuristics to solve the optimization problem in Equation 3. The trivial solution for edge-DCs without on-site generation and those with only a PV sources at night (6am to 8pm) is when all the containers operate with the minimum resolution $px_{\min}$. In the sites without a battery, $\alpha = 0$ and $P_{disch,i}(t) = 0$. Otherwise, $\alpha = \gamma = \frac{1-\beta}{2}$. In the later configuration, when the carbon intensity is high ($CI(t) \geq 80\%CI_0$), no power importation from the grid is allowed, unless a minimum QoS cannot be met. Hence, $P_{g,p}(t) = 0$ and $\beta = 0$. The controller in each Edge-DC first estimates a theoretical envelope of power $P^{th}_{shed,i}$ to degrade (Eq. 4a), with regard to the PV generation, the battery state and the grid carbon intensity. Then it evaluates the average ratio of CPU usage for each server in the site (Eq. 4b) and determines the theoretical resolution px^{th} matching that CPU usages. However, as the resolution is a standard, the controller chooses the normalized px so that $px \geq \max(px^{th}, px_{\min})$ for a given server. Lastly, the contribution of the grid to the Edge-DC can be derived from Equation 3b if applicable.

$$P_{shed,i}(t) \leq P^{th}_{shed,i} = (1-\gamma).(PUE.P_{load,i}(t) - P_{PV,i}(t)) \quad (4a)$$

$$\frac{u(px^{th},t)}{u(px_0,t)} = \frac{PUE.P_{load,i}(t) - P^{th}_{shed,i} - N_{s,i}.P_{\min}}{PUE.P_{load,i}(t) - N_{s,i}.P_{\min}} \quad (4b)$$

$$P_{disch,i}(t) = \min\{P^{\max}_{Bat}\ ;\ \frac{C_{Bat,i}(t-\Delta t).\eta_{disch}.(SoC_i(t)-SoC_{\min})}{\Delta t}\ ;\ (1-\alpha).(PUE.P_{load,i}(t) - P_{shed,i}(t) - P_{PV,i}(t))\} \quad (4c)$$

Where $N_{s,i}$ is the number of active servers in i.

Algorithm 3 summarizes resolution-based performance degradation strategy. As shown in Equation 4c, the battery outputs the minimum value among: its nominal power (first term), the power available in it (second term) and the theoretical optimum (last term).

Algorithm 3: Degradation phase

Input: i
`Degradation (i)`
if $SoC_i(t) < SoC_L$ *and* $P_{PV,i}(t) < P_{load,i}(t)$ **then**
  if $CI(t) > 0.8CI_0$ **then**
    $\beta = 0$; $P_{g,p}(t) = 0$
  else
    $\beta = CI(t)/CI_0$
  end
  $\alpha = \gamma = (1-\beta)/2$
  Evaluate the theoretical power to shed $P^{th}_{shed,i}$ (Eq. 4a)
  Determine theoretical px^{th} 4b) and identify the feasible px
  Calculate the actual P_{shed} and the contribution of the battery $P_{disch,i}(t)$.
  Assess $P_{g,p}$(t) using Equation 3b).
end

D. Consolidation

The consolidation strategy consists in migrating the running containers to the smallest number of physical machines and shutting down idle ones. Not only does this make better use of the computing resources, it also reduces the energy demand. The container live migration considered in this work is an abstraction of the mechanism proposed by Pickartz *et al.* [34] using the Checkpoint/Restore In Userspace (CRIU) technology. As summarized in Algorithm 4, the controllers seek to reserve resources on the most loaded servers in order to relocate containers from the least loaded ones. Jobs on a given server are therefore migrated when all its containers are assigned a new node.

Algorithm 4: Consolidation phase

```
Input: i
Consolidation(i)
Sort L_{on,i} in increasing free CPU and RAM
H = head of L_{on,i}    T = tail of L_{on,i}
while H ≠ T do
    while H ≠ T do
        if the containers of H can be moved to the servers [H, T[ then
            move them and switch off T
            L_{on,i} = L_{on,i} − T    and    L_{off,i} = L_{of,i} + T
        end
        if H is fully loaded then
            H= server of position |H| +1
        end
    end
    T = server of position |T| -1
end
End Function
```

VI. Evaluation

A. Experimental environment

This study is conducted with the simulation toolkit SimGrid [35]. We used a timestep of 5 minutes to perform simulations for a full month and we consider here the last week only (i.e. the steady state). The compute nodes are based on the Nova cluster of the Grid'5000 experimental platform [36]. They are equipped with 32 Intel Xeon E5-2620 v4 with 8 cores each, 64GB memory, 598GB HDD, 2.1GHz frequency and an Intel Ethernet 10G Ethernet card. We measure $P_{\min}$ = 78W and $P_{\max}$ = 151W. Each Edge-DC has 45 nodes. The intra-DC network is designed with 13 C8500-12X Cisco switches of 12 ports and their nominal power consumption is 200W according to the Cisco Power Calculator [32]. We consider a PUE value of 1.2 which is slightly more than the Google current data centers having a PUE ranging from 1.09 to 1.12 [37]. Yet Google DCs are much larger than Edge-DCs. We consider a minimum acceptable resolution $px_{\min}$ = 1080x720.

To instantiate a model of the CPU usage of a real video-streaming application, we measure the power consumption of a VLC media server[1], streaming a 10min video on all its CPUs with a bit rate of 7GBps. We performed this operation on the Grid'5000 testbed [36] for several standard image size px. Figure 3 shows the power consumption of the VLC server for each video size. Then, we used Equation 1 to compute the ratio $\frac{u_k(px,t)}{u_k(px_0,t)}$. For sake of generality, we modeled that ratio as a third-order polynomial $a(px)^3 + b(px)^2 + c(px) + d$ where a, b, c, d are constants. We obtain through polynomial regression, a = 5.16 10^{-3}, b = - 7.36 10^{-2}, c = 0.41, d = 5.18 10^{-2} with an error rate of less than 2%.

[1]https://www.videolan.org/vlc/index.en.html

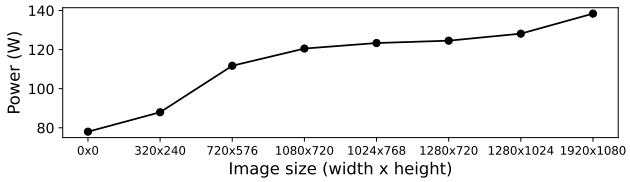


Figure 3: Average node power consumption as a function of the frames size (0×0 represents the idle power consumption of the VLC server).

We use real rooftop solar power trace collected in 2018, in Austin, Texas, and available on the Pecan street website [38]. To simulate the diversity of PV production in our large-scale platform, we split the data by week and periodically (every week), each Edge-DC provided with a PV plant composed of 250 solar panels randomly selects from the split data. The carbon intensity data is provided by the British national grid Electricity System Operator's carbon intensity API [39]. We use here the month of August 2022 which corresponds to the most polluting period of that year. The carbon intensity profile for the last week of August 2022 is shown in Figure 4. The peak value is $CI_0 = 308gCO_2/kWh$. We consider a battery consisting of unit modules of 1kWh capacity ($C_{Bat}(0)$) and 500kW nominal power ($P^{Bat}_{\max}$).

B. Results

In this section, we evaluate the performance of the proposed strategies together first and then separately in order to evaluate their individual gains.

1) A single Edge data center analysis:
First, let's consider one randomly selected Edge-DC equipped successively with three sizes of battery that are 400kWh, 600kWh and 800kWh. The objective is to analyze both the performance of the algorithms and the impact of the electrical system infrastructure on the QoS. Figure 5a presents the power generation of the PV plant and the load trace when the Edge-DC functions with the medium battery (600kWh). Figure 5b represents the SoC of the batteries during the week where the dotted bar is the margin threshold set to 35%. Figure 5c presents the average power consumption of the Edge-DC over three time periods of the day: 0AM to 8AM, 8AM to 6PM and 6PM to 0AM. These time zones correspond to the potential power generation patterns. Figures 5d shows the average image resolution on the site.

On the first day, the PV generation and the batteries charge/discharge power are all low, making all the jobs stream at the minimal resolution most of the time, with only sporadic improvement of images quality when the generation meets the demand. The second day is sunnier, allowing the batteries to be charged. The 400kWh battery leaves the safety margin,

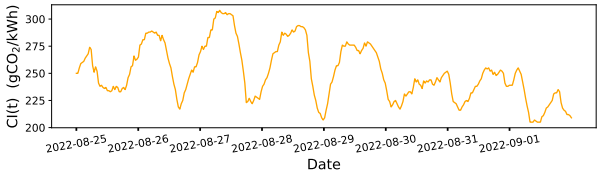


Figure 4: Carbon intensity collected from British national grid Electricity System Operator's carbon intensity API [39]

followed by the 600kWh. In both cases, the images are streams with the maximum resolution and the controller receives new jobs in the negotiation process (more jobs are accepted for 400kWh), which increases the average consumption. From the fourth to the last day, the smallest battery leaves the safety zone, followed by the medium battery. Thus, the power consumption in the two configurations is similar as the same jobs can be accepted from the neighborhood and the images resolution is maximal. However, using the largest battery (800kWh), the image sizes fluctuate between the maximum and minimum resolution, reducing the power consumption in the site. In the middle of the last day, the gap between server demand and PV production does not necessitate a maximum degradation, hence 1280x1024 is chosen. The configurations with a 600kWh and 800kWh batteries allowed energy savings of 0.27% and 0.51% respectively compared to the 400kWh battery which is negligible, especially when considering the battery financial cost. In addition, they require more degradation of the QoS to achieve these savings.

2) Collective Power and carbon profiles analysis:
We define the instantaneous collective self-consumption rate as the proportion of the aggregated edge-DCs power consumption originating from on-site sources at a given time t (Equation 5:

$$1 - \left(\sum_{i=0}^{N_{DC}-1} P_{g,p,i}(t)\right) \Big/ \left(\sum_{i=0}^{N_{DC}} PUE.P_{load,i}(t)\right) \quad (5)$$

where N_{DC} is the number of Edge-DCs.

Using 600kWh batteries, we compute this rate at each time step over a one-week horizon in the following scenarios: 1) all the optimization are performed (CDN) – i.e the inter-site negotiation, the performance degradation and the consolidation are active –, 2) the controllers perform degradation and negotiation (DN) 3) none of the proposed optimizations is performed (None). We compare these scenarios with the state-of-the-art Modified Best Fit Decreasing (MBFD) [8] energy-aware algorithm that combines energy-based resource allocation and consolidation strategies. The results are presented in Figure 6a. (None) has a self-consumption rate ranging from 1.24% to 63.3%, (DN) from 1.35% to 63.6%, MBFD from 18.9% to 64.7% and (CDN) from 20.1% to 65.1%. The strategies of combining spacial collaborative load balancing and QoS degradation which is an interesting opportunity to harvest renewable energy, increases the self consumption by only 0.4% compared to (None). In the opposite, (CDN) allows to consume more on-site energy and outperforms the MBFD algorithm. In fact, (CDN) reaches up to 39.6% self consumption in average during the week, representing an improvement of 7.83% compared to None. However, the benefit is restricted to the extent that the switches in the non-heavily loaded Edge-DCs remain operational, as does the cooling system, and are mainly powered from the grid.

Figure 6b shows the equivalent CO_2 attributable to the platform in the scenarios described above. It shows that (CDN) accounts for less CO_2 than the others, in particular, it exhibits a 35.7% smaller footprint than the (None) scenario. That is due to the fact that (CDN) reduces considerably the need for importing electrical power from the grid. It represents in average 5.2% of carbon savings compared to the MDFD. As for the self-consumption, (DN) leads to a slight reduction of the carbon footprint by 4.2kgCO_2/h which, extrapolated in an annual basis represents 252,526km of car driving.

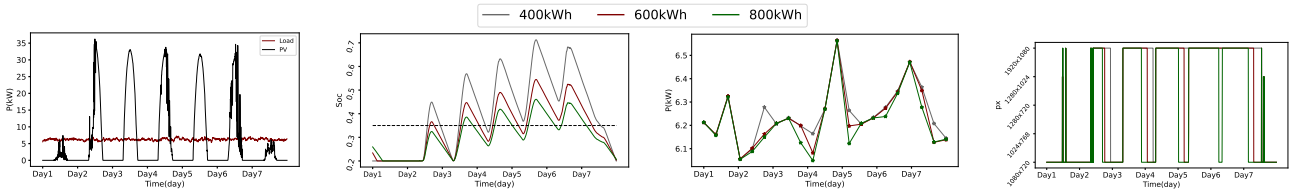
To figure out an upper bound of the savings brought by the degradation in the platform, let's introduce a scenario (Dmax) in which only degradation is performed with images streamed at the lowest resolution (320x240 pixel). For a given server, running at the lowest resolution represents a power consumption reduction of 37% compared to the highest resolution as shown in Figure 3. The average power consumption over one week in the four scenarios is presented in Table I. In terms of energy savings, (D-max) outperforms the (DN) that uses a reasonable resolution. However, it still remains less energy efficient than MBFD and (CDN). Moreover, this scenario is not realistic as this quality of image might not be acceptable by the user despite their will to enjoy a low carbon service. Thus, the degradation and negotiation are more beneficial when associated with consolidation. Indeed, consolidating reduces considerably the static power consumption and reorganizes the resources, freeing some space for new jobs to be exchanged.

Table I: Average power consumption over one week for several scenarios

Scenario	CDN	MBFD	Dmax	DN	None
Power consumption (MW)	1.602	1.66	2.06	2.17	2.205

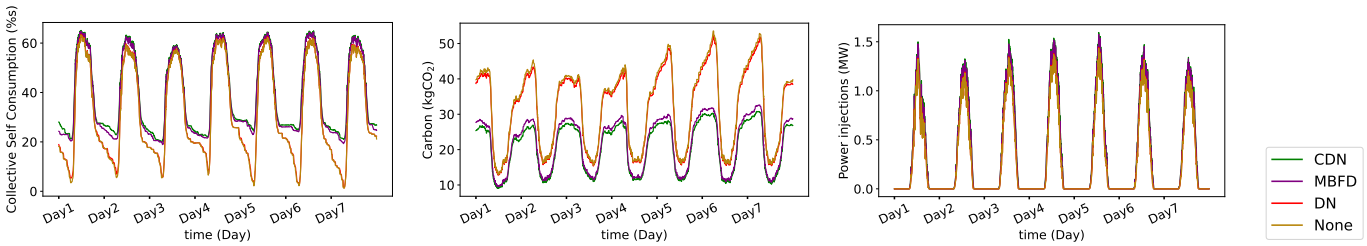
The last analysis of the entire platform concerns the profile of power injected into the grid in the strategies (CDN, MBFD, DN and None). As shown in Figure 6c, peaks of up to 1.6MW are injected in all four cases. The scenarios including consolidation inject the most power into the grid. In fact, by switching off the idle servers, the load is reduced leading to a high surplus that, in the Edge DCs with no battery, is directly injected into the grid. Such power peaks may be difficult to handle for the electrical network, as they may not be synchronized with the consumption at a large scale. So, consolidation is the best option from an edge computing provider to reduce both its energy consumption and its carbon footprint, yet these high peak injections may be costly in terms of infrastructure for the electricity provider [33] and consequently, they can induce a higher electricity bill for the edge provider despite its energy injection into the grid.

In this study, we proposed a fully distributed energy, location and carbon-aware resource allocation strategy that



(a) PV generation and load trace. (b) Batteries State of Charge (c) Day-night average load (d) Video-streaming resolution

Figure 5: Impact of streaming on the infrastructure, example on one Edge-DC with respectively 400kWh, 600kWh and 800kWh batteries.



(a) Collective self-Consumption (b) Carbon generation (c) Peak injection

Figure 6: Impact of the strategies on the edge infrastructure: no optimization (None), consolidation, degradation and negotiation (CDN), degradation and negotiation (DN), Modified Best Fit Decreasing (MBFD).

show promising performances. However, it leaves openings for enrichment in order to incorporate the critical aspects of system security and reliability. In fact, in distributed systems, node failures are commonplace. Establishing fault detection and recovery mechanisms is crucial to ensure the reliability and stability of task allocation and when migrating containers. The fault tolerance and recovery mechanism should have the ability to handle cases where container migration failures or the shutdown of physical machines lead to issues.

VII. Conclusion

In this paper, we have designed a large-scale edge computing infrastructure where the data centers are brought close to the end user while being grouped according to their closeness in order to guarantee latency tolerance when jobs are moved among them. We presented a strategy that allow the infrastructure to consume less energy by consolidating the load on the smallest number of servers and degrading the application quality of service when their is shortage of on-site power generation, while respecting service level agreements. This strategy also increases the self-consumption within the edge infrastructure by forwarding the incoming jobs to the nearest data center with enough on-site power available. The results show that the combination of these three techniques increases the collective self-consumption and reduces significantly the carbon footprint of the infrastructure compared to the literature. Although the degradation and negotiation strategies achieve energy savings, they remain less efficient than consolidation and are even greatly improved by combination with consolidation. Also, we found that using large batteries may lead to very frequent degradation, which may be counterproductive, as one invests in large storage to increase the efficiency. In future work, we aim to investigate the sizing of the battery and the PV plant in order to optimize the electrical infrastructure. In the Edge-DCs with no onsite-source, the network switches continuously consume from the grid, even with less load. Hence, we will investigate the flexibility to optimize the network in order to reduce its consumption, thus the imports from the electrical grid.

Acknowledgments

This project has received financial support from the CNRS through the MITI interdisciplinary programs. Experiments presented in this paper 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.

References

[1] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of things market analysis forecasts, 2020–2030," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, pp. 449–453.

[2] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *Computer*, vol. 50, no. 10, pp. 58–67, 2017.

[3] IEA. (2021) Data Centres and Data Transmission Networks. [Online]. Available: https://www.iea.org/reports/data-centres-and-data-transmission-networks

[4] Y. Sahni, J. Cao, L. Yang, and S. Wang, "Distributed resource scheduling in edge computing: Problems, solutions, and opportunities," *Computer Networks*, vol. 219, 2022.

[5] T. Lähderanta, T. Leppänen, L. Ruha, L. Lovén, E. Harjula, M. Ylianttila, J. Riekki, and M. J. Sillanpää, "Edge computing server placement with capacitated location allocation," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 130–149, 2021.

[6] A. Karimiafshar, M. R. Hashemi, M. R. Heidarpour, and A. N. Toosi, "A request dispatching method for efficient use of renewable energy in fog computing environments," *Future Generation Computer Systems*, vol. 114, pp. 631–646, 2021.

[7] C. Jiang, T. Fan, H. Gao, W. Shi, L. Liu, C. Cérin, and J. Wan, "Energy aware edge computing: A survey," *Computer Communications*, vol. 151, pp. 556–580, 2020.

[8] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[9] A. Toor, S. ul Islam, N. Sohail, A. Akhunzada, J. Boudjadar, H. A. Khattak, I. U. Din, and J. J. Rodrigues, "Energy and performance aware fog computing: A case of DVFS and green renewable energy," *Future Generation Computer Systems*, vol. 101, pp. 1112–1121, 2019.

[10] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "A Survey on Multi-Access Edge Computing Applied to Video Streaming: Some Research Issues and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 871–903, 2021.

[11] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

[12] M. Alkhalaileh, R. N. Calheiros, Q. V. Nguyen, and B. Javadi, "Data-intensive application scheduling on mobile edge cloud computing," *J. of Network and Computer Applications*, vol. 167, 2020.

[13] E. Ahvar, A.-C. Orgerie, and A. Lebre, "Estimating Energy Consumption of Cloud, Fog, and Edge Computing Infrastructures," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 2, pp. 277–288, 2019.

[14] M. P. Alves, F. C. Delicato, I. L. Santos, and P. F. Pires, "LW-CoEdge: a lightweight virtualization model and collaboration process for edge computing," *World Wide Web*, vol. 23, pp. 1127–1175, 2020.

[15] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate IoT edge computing with lightweight virtualization," *IEEE network*, vol. 32, no. 1, pp. 102–111, 2018.

[16] J. Li, J. Cai, F. Khan, A. U. Rehman, V. Balasubramaniam, J. Sun, and P. Venu, "A Secured Framework for SDN-Based Edge Computing in IoT-Enabled Healthcare System," *IEEE Access*, pp. 479–490, 2020.

[17] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.

[18] D. Lu, Y. Qu, F. Wu, H. Dai, C. Dong, and G. Chen, "Robust server placement for edge computing," in *IEEE Int. Parallel and Distributed Processing Symposium (IPDPS)*, 2020, pp. 285–294.

[19] A. M. Maia, Y. Ghamri-Doudane, D. Vieira, and M. F. de Castro, "Optimized placement of scalable IoT services in edge computing," in *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 189–197.

[20] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5506–5519, 2018.

[21] Y. Zhang, X. Chen, Y. Chen, Z. Li, and J. Huang, "Cost Efficient Scheduling for Delay-Sensitive Tasks in Edge Computing System," in *IEEE Int. Conference on Services Computing (SCC)*, 2018, pp. 73–80.

[22] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *IEEE Int. Conference on Edge Computing (EDGE)*, 2018, pp. 66–73.

[23] Q. Fan and N. Ansari, "Cost aware cloudlet placement for big data processing at the edge," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[24] N. K. Suryadevara, "Energy and latency reductions at the fog gateway using a machine learning classifier," *Sustainable Computing: Informatics and Systems*, vol. 31, 2021.

[25] European Commission, "A Clean Planet for all - a European strategic long-term vision for a prosperous, modern, competitive and climate neutral economy," https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52018DC0773, 2018.

[26] A. D. Mustika, R. Rigo-Mariani, V. Debusschere, and A. Pachurka, "A two-stage management strategy for the optimal operation and billing in an energy community with collective self-consumption," *Applied Energy*, vol. 310, p. 118484, 2022.

[27] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: A case study," in *IEEE International Conference on Communications Workshops*, 2009, pp. 1–5.

[28] Apple, "Apple Environmental Progress Report," https://www.apple.com/environment/pdf/Apple_Environmental_Progress_Report_2021.pdf, 2021.

[29] "Arrêté du 21 novembre 2019 fixant le critère de proximité géographique de l'autoconsommation collective étendue," https://www.legifrance.gouv.fr/loda/id/JORFTEXT000039417566/, 2020.

[30] R. F. da Silva, H. Casanova, A.-C. Orgerie, R. Tanaka, E. Deelman, and F. Suter, "Characterizing, modeling, and accurately simulating power and energy consumption of i/o-intensive scientific workflows," *Journal of computational science*, vol. 44, p. 101157, 2020.

[31] A. Adelin, P. Owezarski, and T. Gayraud, "On the impact of monitoring router energy consumption for greening the Internet," in *IEEE/ACM International Conference on Grid Computing*, 2010, pp. 298–304.

[32] CISCO, "Cisco's power consumption calculator." https://cpc.cloudapps.cisco.com/cpc/, accessed April 2023.

[33] W. E. Gnibga, A. Blavette, and A.-C. Orgerie, "Renewable Energy in Data Centers: the Dilemma of Electrical Grid Dependency and Autonomy Costs," *IEEE Transactions on Sustainable Computing*, pp. 1–13, 2023.

[34] S. Pickartz, N. Eiling, S. Lankes, L. Razik, and A. Monti, "Migrating LinuX containers using CRIU," in *ISC High Performance*, 2016, pp. 674–684.

[35] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, "Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2899–2917, Jun. 2014.

[36] D. Balouek *et al.*, "Adding virtualization capabilities to the Grid'5000 testbed," in *Cloud Computing and Services Science*. Springer, 2013, vol. 367, pp. 3–20.

[37] Google, "Google data centers efficiency," https://www.google.com/about/datacenters/efficiency/, accessed February 2023.

[38] Pecan Street, "Pecan street dataport," https://www.pecanstreet.org/dataport/, 2018.

[39] Google, "Britannic National ESO Carbon Intensity API," https://carbonintensity.org.uk/, accessed June 2023.