Université Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

préparée dans le cadre d'une cotutelle entre l'Université de Grenoble et l'Universidade de São Paulo

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Miguel Felipe SILVA VASCONCELOS

Thèse dirigée par **Fanny DUFOSSÉ**
et codirigée par **Daniel CORDEIRO**

préparée au sein du **Laboratoire d'Informatique de Grenoble** et **Escola de Artes Ciências e Humanidades**
dans les Écoles Doctorales **Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII)** et **Programa de Pós-Graduação em Sistemas de Informação (PPgSI EACH-USP)**

Strategies for operating and sizing low-carbon cloud data centers

Thèse soutenue publiquement le **13 décembre 2023**,
devant le jury composé de :

Rapporteur RAPPORTEUR
Professeur d'Université, Université, Pays, Rapporteur
President PRESIDENT
Professeur d'Université, Université, Pays, Président
Examinatrice EXAMINATRICE
Professeur d'Université, Université, Pays, Examinatrice
Examinateur EXAMINATEUR
Professeur d'Université, Université, Pays, Examinateur
Daniel CORDEIRO
Professeur d'Université, Université, Pays, Co-Directeur de thèse
Fanny DUFOSSÉ
Professeur d'Université, Université, Pays, Directeur de thèse

Aos meus pais, Rosilene e Jose.

Frase de epigrafo

— Autor do epigrafo

Remerciements (Acknowledgments)

Agradecer aos meus pais e irmao

meus amigos do lab e da each

meus orientadores

colaboradores do proejto datazero

tb a each e a univ de grenoble, fapesp, labex, pessoal do datazero e td mais

Abstract / Résumé

Abstract

Falar do consumo de energia

impacto ambiental dos dcs

emissao de carbono

tecnicas carbon aware

carbon responsive

sizing e contention

follow the renewables

Résumé

Falar do consumo de energia

impacto ambiental dos dcs

emissao de carbono

tecnicas carbon aware

carbon responsive

sizing e contention

follow the renewables

List of Figures

List of Tables

Contents

1 Introduction

Cloud computing has revolutionized the industrial and academic world of Information Technology with its ability to provide large amounts of computing resources on demand. Most of the applications and services we use every day that feature more and more users, such as social networking, e-mail, video games, internet banking, health applications, and video streaming, are supported by cloud computing platforms.

Despite all its benefits, cloud data centers have a significant power demand. The International Energy Agency (IEA) estimates that the data centers (DCs) that host cloud computing platforms consumed between 240 to 320 TWh — approximately 1% of the global final electricity generated in 2022 [28]. To put this energy consumption into perspective, it's enough to supply the anual energy demand of entire countires: in 2022 seven countries had an energy consumption similar to the DCs energy demand: Greece (316 TWh), Israel (304 TWh), Belarus (297 TWh), Switzerland (292 TWh), Hungary (266 TWh), Portugal (258 TWh) and Morocco (257 TWh) [47].

Efforts from both academy and industry alike are trying to reduce this country-like energy consumption. An analysis of the global energy use by data centers performed by [34] showed that during the period between 2010 and 2018, the DC's internet protocol traffic increased more than 10-fold, it is estimated that the DC's storage capacity increased by 25-fold, and the DCs workload and compute instances increased by 6-fold. On the other hand, the energy consumption only increased by 6% thanks to improvements in: i) virtualization, which allowed a 5-fold increase in the average number of compute instances by physical server; ii) more energy-efficient servers (a decrease with a factor of 4 regarding the power used for

computation and a factor of 9 in terms of watts per terabyte); and iii) energy-efficient infrastructures, in particular for the cooling needs and power supply.

It is uncertain if the improvements in efficiency in cloud computing will mitigate the power increase by the growing demand for computational resources in the long term. In the work of [30], the authors developed a model for forecasting the energy consumption of data centers using the system dynamic models approach. Considering the period from 2016 to 2030, and the scenario with the end of Moore's law — that predicts that the number of components in the integrated circuits doubles every year [32]— and the rise of industrial Internet of Things applications, the energy demand of the data centers could reach 2.3% of the global electricity generation — it will double considering what was reported by the IEA for the year 2022 [28].

The electricity consumption of the DCs results in environmental impact in terms of greenhouse gas emissions (GHG). The extent of this impact depends on the type of source used to produce the electricity — Table 1.1 presents the emissions for the different types of power sources. In 2019, approximately 70% of the internet's data traffic was processed by the DCs in the "Data Center Alley", a region located in Virginia (United States of America), which was only supplied by less than 2% of renewable energy (green energy) [19]. In order to reduce their environmental impact, major cloud providers — such as Amazon AWS, Apple, Google, Meta (former Facebook) and Microsoft — made commitments to integrate low-carbon intensive electricity produced by renewable sources in the operations of their data centers, and some already deployed their projects to reach the goal of minimizing the carbon footprint. In fact, Google reported that in 2022 64% of the power used to supply its DCs an hourly basis came from renewable sources [17], with some DCs reaching up to 97% of green energy usage.

The cloud platforms are integrating renewable power in their operations in the following ways. One possibility is to buy green energy from the local electricity grid,

Table 1.1 CO_2 emissions from different power sources. Source [45].

Generation technology	**Emissions ($g\,\mathbf{CO_2}e/kWh$)**
Coal	1 010
Oil	840
Natural gas	486
Biomass	52
Photovoltaic	43
Hydropower	21
Nuclear	13
Wind	13

although it makes them dependent on the grid's energy mix, and not all locations in the world are supplied by low-carbon sources. Another possibility is to manufacture their own on-site renewable infrastructure. A critical aspect of producing electricity using renewable energy that must be considered for both possibilities, is their intermittent nature — their production fluctuates over time and is influenced by variables such as the time of the day, climate, season, and geographic location. For example, solar power is only produced when the sun is shining, and during winter, the solar irradiation is lower than in the summer, and for hydropower, long periods without rain will affect power production.

The adoption of energy storage devices (ESDs) in DCs to store renewable energy [62] is an approach to deal with intermittency. These devices are already used in DCs as a temporary backup for power outages. The green energy from the ESDs could be used when there is no production of green power, or the demand is greater than the production. Nevertheless, this approach also presents some challenges for deciding when to use or to recharge, given the characteristics of the ESDs such as the self-discharge rate (they lose stored energy over time when not used), aging (their maximum capacity decreases over time), and limited charge and discharge rates (the maximum power that can be extracted or recharged).

In this thesis, we study how to reduce the environmental impact of cloud computing data centers, specifically in terms of CO_2 emissions. The usage of Carbon-Responsive

strategies — approaches that are aware of their carbon footprint and use this information to make decisions [37] — is explored to reduce the CO_2 emissions in both operating and sizing the data centers — determining the required on-site renewable infrastructure (area of solar panels, number of wind turbines, capacity of ESDs). In terms of DC operations, we leverage the workload scheduling using "follow-the-renewables" approaches — strategies that migrate and allocate the workload to the locations with higher renewable energy availability [52] — and study their impact on both energy consumption and network congestion. Regarding sizing, we model our problem using a Linear Program formulation that considers the characteristics of each DC location in terms of cooling needs and local grid energy mix, as well as the fact that the renewable infrastructure generates an environmental impact. More details about the structure and contributions of this thesis are presented in the next section.

1.1 Structure of the thesis and contributions

This thesis is structured into six chapters, which include this introductory chapter. An overview of the content of each chapter and their respective contributions is described in what follows.

In Chapter 2, the reader is introduced to the concepts required to comprehend the research done in this thesis. In particular, the chapter contains a description of the general scenario of cloud computing, scheduling algorithms for cloud computing, the definition of carbon-responsive strategies — in particular, the approaches explored in this thesis: "follow-the-renewables" and sizing, and how one can measure the environmental impact of cloud platforms. In addition, the related works used as inspiration or baseline for our proposed solutions are presented in this chapter as well.

Chapter 3 presents an analysis of the adoption of the carbon-aware strategy "follow-the-renewables" and its impact on energy consumption and network congestion. The proposed solution consists of an algorithm that considers the network topology and usage for scheduling the workload migration. Through computational experiments with real-world data for the workload, cloud infrastructure, and a simulator that uses realistic models validated by the scientific community for both the energy consumption and network usage, the results show that the proposed solution outperforms the baselines considering the energy consumption, non-renewable energy consumption and network congestion.

Chapter 4 introduces another Carbon-Responsive strategy to reduce the carbon footprint of operating cloud computing platforms: sizing the renewable infrastructure. This strategy involves determining the required area of on-site solar panels and the capacity of the batteries to supply the DCs power demand. The proposed solution consists of a Linear Program formulation that addresses both the workload scheduling using follow-the-renewables and the sizing of the on-site renewable infrastructure. By modeling these two sub-problems as one problem it is possible to evaluate whether it is more effective to execute the workload in other geographic locations or to increase the area of solar panels and capacity of the batteries to achieve the goal of reducing the CO_2 emissions. The model only uses linear variables, enabling it to be optimally solved in polynomial time — essential for the scale of cloud computing platforms that have millions of servers and executes hundred of millions of computational tasks daily. Other important aspects of the modeling are that it considers the characteristic of each DC geographic location in terms of cooling needs and the energy mix of the local electricity grid (that may have a share of renewable energy sources), and the fact that manufacturing solar panels and batteries have an environmental impact. The results demonstrate that the hybrid configuration that combines both the on-site renewable infrastructure and the regular grid outperforms in terms of CO_2 emissions the strategies of having a DC only supplied by energy produced from its local renewable infrastructure — a

reduction of approximately 30% — and only using power from the regular electrical grid — a reduction of approximately 85%.

In Chapter 5, the model introduced in Chapter 4 is extended to evaluate how to minimize the carbon emissions of the cloud data center operations considering the long-term. The following modifications and analysis were performed: the environmental impact of whole life cycle of the renewable infrastructure (from manufacturing, operation and recycling) is taken into account to make the model closer to the real-world; an analysis of the benefits and challenges of including wind turbines in the on-site renewable infrastructure; an assessment of the model's sensitivity to the inputs, in particular climate condiitons and grid emissions data; an anaylisis of how much the CO_2 emissions can be reduced by adopting the flexibility of delaying a part of the workload, which could be used to mitiate the impacts of over-sizing the renewable infrastructure; a discussion of the monetary costs (in dolars) associated with on-site renewable infrastructure, as well as the trade-offs between minimizing the carbon footprint and reducing the costs for the DCs; and deciding to manufacture new servers over the years that may be more energy-efficient, considering that manufacturing them also emmits carbon. This modeling may be used to guide decision-makers in their efforts to reduce the carbon footprint of cloud data centers.

Finally, Chapter 6 presents a discussion of the main contributions of this thesis as well as possible future research trajectories.

Background and Related Work

2

This chapter lays the foundational knowledge required for the reader to comprehend the subsequent chapters of this thesis. First, the general scenario of cloud computing will be presented in Section 2.1. Then, an introduction to scheduling in the context of clouds will be shown in Section 2.2. The existing approaches explored in this thesis to reduce the environmental impact of cloud computing platforms will be discussed in Section 2.3, and state-of-the-art studies used in this thesis as inspiration and baselines will be presented as well. Finally, section 2.4 will present how one can measure the environmental impact of cloud computing platforms.

2.1 Cloud computing

Cloud computing was introduced by the industry to address the most significant problems of e-commerce found at the time. Before the advent of cloud computing and its on-demand access to computational resources, users and companies had to purchase their computational infrastructure to deploy their services or applications. In the event of a sudden surge in requests, it was often challenging to scale the computational infrastructure in time to handle the computational load demand. Despite its revolutionary aspects of information technology, cloud computing is not classified as a new paradigm in computer science research. As a matter of fact, it is the evolution of research into different fields of computer science, such as clusters, grids, autonomous computing, and ubiquitous computing.

One of the critical elements that enable cloud computing is virtualization technology: the computational resources of a physical machine are converted to virtual resources that can be shared among many users and applications — also known as multi-tenancy. The main ideas from virtualization originate from the late 1950's and beginning of 1960's with the multiprogramming paradigm, and today, the research on virtual machines (VMs) and containers continues to investigate the compromises outlined in the literature of multiprogramming in terms of portability — allowing the virtual resource to execute in any hardware, performance — the impact of the additional virtual layer in the application execution in comparison to the gains regarding the multitenancy, and security — how to isolate the interaction between different applications and the applications and the computational resources [42].

Cloud computing platforms generally support services in three distinct levels: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) [14]. In the first level, Infrastructure as a Service (IaaS), the platform grants users access to hardware resources (such as processing and storage) and charges them for their usage. Services such as Amazon EC2 (Elastic Cloud Computing) Service and Amazon S3 (Simple Storage Service) are examples of IaaS clouds. At the second level, Platform as a Service (PaaS), the provider supports complete development, testing, and deployment environment for the application developer, which usually means that the developer will have to follow a specific fixed development model and accept restrictions on how to model the software in exchange for the scalability provided. An example is the Google App Engine. Finally, at the third level, Software as a Service (SaaS), specific applications are offered to users via the Internet, and the rate is proportional to the use of the application. We can cite the Google Docs office applications as examples.

Modern cloud computing platforms are composed of multiple data centers geographically distributed over the world — also denominated cloud federations or multi-clouds. Figure 2.1 illustrates the locations of the cloud data centers from

Microsoft's Azure that, in total, consist of millions of physical servers [48]. The need for such geographically distributed infrastructure comes from meeting the demand of the huge number of users and reducing the response time for their applications, as well as for security and redundancy reasons. For example, if a data center in a region goes down because of a power problem or hacking attack, another region could temporarily receive the computational load. Finally, this distributed infrastructure enables exploring different approaches to reduce the environmental impact of operating the data centers, and more details of these strategies will be given in Section 2.3.

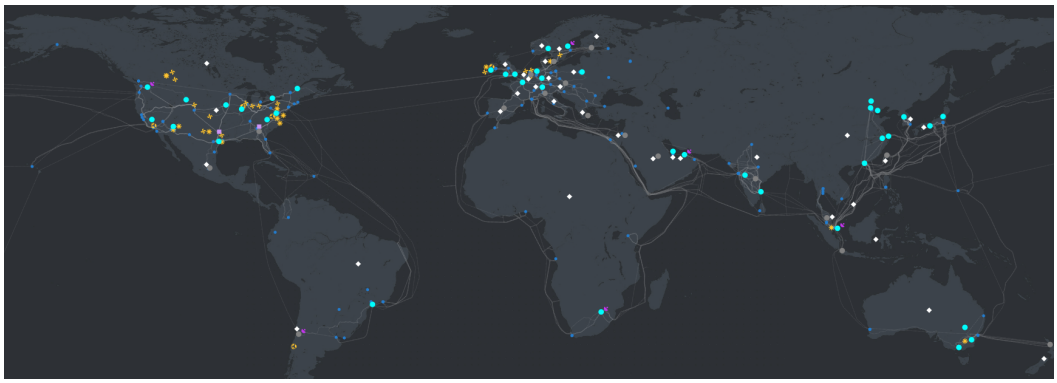

Figure 2.1 Geographic locations of Microsoft's Azure DCs. Extracted from [36].

2.2 Scheduling for cloud computing platforms

Scheduling problems are combinatorial optimization problems where, given the description of the characteristics of a computational resource set (α) and a task set (β), the objective is to find an allocation (in time) of resources to tasks that minimize some optimization criteria (γ). These problems are denoted, generally, using the $\alpha \mid \beta \mid \gamma$ notation, introduced by Graham [18].

The most common criterion studied in problems of high-performance computing is called makespan (also denoted by $C_{\max}$), which indicates the time when the last task that makes up an application finishes its execution. When the available

computational resources are identical and known previously, and we are interested in minimizing makespan—typically the case in scheduling problems in cluster clusters and computational grids—the problem is strongly NP-complete [15]. This problem is denoted by $P||C_{\max}$ in Graham's notation.

For the research problem explored in this thesis, α represents the computational resources (memory and CPU) from the servers of the data centers as well as their information about the availability of low-carbon intensive power sources, β represents the VM requirements in terms of execution time, deadline, memory and CPU, and γ represents the reduction of carbon-intensive energy consumption.

It is known—from the work of Graham [18] and Garey and Johnson [15]—that the class of greedy algorithms known as list algorithms provides fast and efficient heuristics for scheduling tasks on parallel computers. These algorithms have a $2-1/m$ approximation guarantee for the worst case, but are remarkably effective in practice, especially when the ratio of the number of tasks to the number of available computational resources is large. Among the most commonly used list algorithms are the Longest Processing Time first (LPT) and Shortest Processing Time first (SPT), for homogeneous platforms, and the Heterogeneous Earliest Finish Time (HEFT) algorithm, for heterogeneous platforms.

The VM consolidation strategy illustrates an example of a scheduling problem on cloud computing enabled by virtualization. It consists of using live-migrations — migrating the VM between different machines while in execution, that is, without preemption — to reallocate the VMs in order to achieve the minimum number of physical machines used, and turning off the machines that became idle to save power. The reader can check a systematic literature review on such techniques at [13].

The greedy algorithms were adopted in some of the solutions proposed in this thesis, as well as most of the related works used as baseline and inspiration. The

justification for such a decision is that despite not providing the optimal solution, they can provide an acceptable solution in a reasonable amount of time, which is important considering the workload size with millions of tasks and restrictions such as deadline, response time, and others. The following section presents these strategies to the reader.

2.3 Carbon-Responsive Computing

Many efforts from both academia and industry are being made to reduce the environmental impact of information technology. The term Carbon-Responsive Computing was created to describe the strategies that explore moving the computational workload both in the time and spatial dimensions aiming to reduce the carbon footprint of the operation [37].

There are three levels for Carbon-Responsive computing. In the first, Carbon-aware computing, the system knows the carbon intensity of the electricity used by its workloads or IT equipment. In the second, Carbon-responsive computing, the information about the carbon intensity is taken into account to make decisions. In the final stage, Carbon-resilient computing studied how to manage and integrate carbon-responsive elements and what modifications are needed in IT and power supply infrastructure.

In this thesis, it is presented a study of the Carbon-Aware strategy follow-the-renewables — presented in Section 2.3.1, and the Carbon-Reslient computing strategy of sizing the renewable infrastructure — discussed in Section 2.3.2.

2.3.1 Follow-the-renewables

As we saw in Section 2.1, modern cloud computing platforms are geographically distributed across the globe, having data centers on many combinations of latitudes, longitudes, and time zones. Considering that each location has a different potential for generating renewable power, and that renewable energy is intermittent — the sun only shines during the day, and the wind does not blow all the time at speeds that are sufficient to turn the wind turbine propeller — the scheduling algorithms can use this characteristic to plan the execution of the workload in the locations that have more availability of green energy. In the literature, the term "follow-the-renewables" [52] describes this strategy.

The work from Xu and Buyya [63] provides a comprehensive overview of the classical strategies utilized to reduce data centers' power consumption. Additionally, they present their scheduling algorithm that incorporates the "follow-the-renewables" to allocate the workload between the data centers located in different time zones. The optimization algorithm has the objective of minimizing the total carbon emissions of the operation while ensuring the average response time of the requests. The workload will be executed on the data centers they were allocated to, and during execution it is not possible to migrate to other DCs that may have a higher availability of green energy — no VM live-migration, nor VM consolidation to reduce the number of servers being used.

Reducing power consumption, the environmental impact generated by using carbon-intensive electricity, and monetary costs while not degrading the workload performance were studied by Ali et al. [3]. They proposed a distributed scheduling algorithm for geographically distributed DCs with heterogeneous hardware. The algorithm has two main steps that adopt greed heuristics. The first consists of allocating the workload to the servers given a policy (lower energy price or carbon intensity). In the second step, it is possible either to apply server consolidation on

the workload in execution to reduce the number of servers in utilization (by using intra-DC migrations) or the tasks can be shifted to other data centers (inter-DC migrations) in order to use cheaper or greener electricity. The algorithm also considers that the migration's destination server might be less powerful than the server where the tasks were originally in execution, which would degrade the performance of the workload.

Ruiz Duarte and Fan [49] studied the problem of operating data centers geographically distributed across the United States in terms of scheduling the workload and the power supply — that comes from the regular electrical grid, batteries, or on-site renewables. The proposed solution consisted of a Mixed Integer Linear Program (MILP) formulation that adopts the follow-the-renewables for allocating and migrating the workload, and manages the decision of selecting which power source will be used. The objectives of the MILP are to minimize both costs — that originate from purchasing electricity from the grid, energy storage devices, selling or buying carbon credits, and the CO_2 emissions — that originate from using carbon-intensive electricity from the regular electrical grid.

In Khodayarseresht et al. [29], the authors study the workload scheduling problem to minimize energy consumption while avoiding violating the Service Level Agreements. The proposed solution used greedy heuristics to find the server that could run the tasks with the minimum increase in total energy consumption. If the algorithm found more than one server available, the criteria of the one that uses less carbon-intensive electricity was used to break the tie. Therefore, minimizing the emissions was the secondary objective of the algorithm, and the "follow-the-renewables" was only applied at this step. It is important to remind the reader that minimizing energy consumption and carbon emissions are different problems. For example, suppose DC A uses power from a grid supplied by coal power plants. In that case, it will have a carbon footprint more than 20 times higher than a DC B that consumes the same

amount of power and uses photovoltaic energy — $1001g\,CO_2 - eq.kWh^{-1}$ for coal power and 43 $g\,CO_2 - eq.kWh^{-1}$ for solar power [45].

The studies Camus et al. [9] and Camus et al. [8, 7] are from the same research group and aim to optimize low-carbon intensity energy usage in the context of data centers that have a power supply for both the regular electrical grid and locally installed solar panels. Also, these studies consider data centers that are geographically distributed, which allows for exploring "follow-the-renewables" strategies. The first study proposed, SAGITTA[9], uses a stochastic approach to estimate renewable energy production and a greedy heuristic to allocate resources to meet the demand of virtual machines. The greedy heuristic chooses which servers of the data centers will receive the virtual machine workload considering their green energy availability and turns off those not selected to reduce the loss of renewable energy. After determining the servers to turn on or off, the algorithm tries to migrate the workload between servers of distinct data centers to use those with the most green energy availability. NEMESIS[8] is an evolution of SAGITTA, and also uses a stochastic approach to estimate renewable energy production and greedy heuristics for VM allocation that select the servers from the data centers with the most available green energy. The main differences are that the proposed solution uses a consolidation algorithm to reduce the number of powered servers and a VM migration algorithm considering network costs. SCORPIOUS [7] is an extension of NEMESIS with a new approach: the sharing of excess renewable energy generated by geographically distributed data centers that have solar panels installed. This approach uses SmartGrid's collective self-consumption, and the excess of green energy can be used by data centers that are not generating solar power, or when the demand is greater than the production.

Finally, "follow-the-renewables" approaches are feasible candidates for mitigating the intrinsic intermittent nature of renewable power, but one must not neglect its limitations. Chapter 3 presents a discussion of the impact of adopting such strategies, in particular on the network and energy consumption, a proposed solution that

extends the work from Camus et al. [8] and the works of Xu and Buyya [63] and Ali et al. [3] were used as baselines to evaluate the solution.

2.3.2 Sizing the renewable infrastructure

There are two main options for supplying the cloud federation power demand with low-carbon intensive electricity. The data center owner can purchase green power from the local electricity grid that has the integration of renewables — which makes them dependent on the grid, and not all locations provide this possibility — or it can install a local renewable infrastructure in the DCs. For the latter option, the main problem is deciding the dimensions of the renewable infrastructure needed, for example, the required area of solar panels, number of wind turbines, and capacity of the batteries (to store green power and use when opportune) to supply the data center and minimize its carbon footprint. This problem is known as sizing, dimensioning, or capacity planning in the literature. Many factors need to be considered in the sizing decision: the costs of manufacturing and operating these devices, the potential of generating power given the climate conditions and the geographic location, and the fact that these devices also present an environmental impact during their lifetime cannot be neglected as well.

Most of the scientific literature on the sizing problem for cloud data centers has focused on the scenario of a single DC. Two main approaches are investigated: i) the data center may be supplied by the regular electrical grid as a backup when the renewable power production is not sufficient, or ii) the DC is fully supplied by green energy from its on-site renewable infrastructure — which may require the adoption of energy storage devices.

The first approach — using the regular electrical grid as a backup — is adopted by the majority of works on sizing for clouds. To supply the power demand of fog DCs in a rural area in India, Priya et al. [40] proposed a solution that adopts a

Particle Swarm Optimization strategy for sizing a smart microgrid. The optimation problem's objective is to minimize the monetary costs of buying solar panels, wind turbines, diesel generators, and batteries. Additionally, the authors propose a scheduling algorithm to maximize low-carbon-intensive energy utilization from its local renewable infrastructure.

The possibility of using curtailed green energy to supply the DCs power demand and provide hydrogen to hydrogen refueling stations was evaluated by Niaz et al. [38]. The proposed solution consisted of an optimization problem that used Mixed Integer Linear Programming (MILP) with the objective of minimizing the capital costs. The modeling included natural-gas–powered combined cooling, heating, power systems, electrolyzers, hydrogen fuel cells, heat pumps, hydrogen tanks, and battery energy storage systems. The study provided three interesting results: i) the worst scenario regarding both economic and environmental aspects was only to consume power from the regular electrical grid; ii) The scenario with the minimal monetary costs was the combination of using curtailed renewable energy and power from the grid; and iii) the best solution for the environment was to consume only renewable power, however, it resulted in the highest costs.

The second main approach — using only power from the on-site renewable infrastructure without accessing the grid — is becoming more and more explored in the literature, and it presents some promising results. A planning methodology for a net-zero energy system was discussed by Richter et al. [46], in which it was accessed the potential for a net-zero energy DC in Germany through a qualitative study. The research concluded that the DC presented a large potential to operate as a net-zero energy system, taking into account the importance of selecting proper technologies for producing electricity, increasing energy efficiency, and optimal sizing of energy storage devices. Other secondary benefits are that the marketing image and economic value can increase when the DC operates as a net-zero energy system.

The DATAZERO [39] project explores the possibility of a single data center being operated exclusively from renewable energy and uses energy storage devices to try to smooth the short and long-term intermittency. The work from Haddad et al. [24] — part of DATAZERO project — evaluates the possibility of using the combination of solar panels and wind turbines for supplying the DC power demand and the usage of batteries and hydrogen storage — for short and long-term storage, respectively. The study discussed the impact of the geographic location and the computational workload that must be executed in the resulting sizing — number of servers, dimensions of the renewable power sources, and capacity of the energy storage devices. Another work from the DATAZERO project evaluated how to reduce oversizing, that is, manufacturing more renewable infrastructure than needed, in DCs powered by green energy [6]. The classical sizing approaches are very sensitive to variations of the inputs, as there are few days where there is a peak in computational demand for the workload or low renewable production. The proposed solution evaluated how to minimize the over-sizing and the impact in terms of Quality of Service and on the dimensions of the renewable infrastructure. A binary search strategy was adopted to find the best relevant sizing, in contrast to the previous studies that used MILP approaches.

Other research area explored in the scientific literature analyzes the sizing of particular elements, such as the electrical infrastructure. The usage of batteries as an energy storage device and their size to supply 50% of the DC's total energy demand from renewables was studied by Sheme et al. [50]. For the experiments, the authors developed a simulator that receives the information on the PV area and capacity of the batteries as input, and different geographic locations were considered: Finland, Crete, and Nigeria. The results highlight the different potential of the geographic locations: Nigeria presented the smallest area of solar panels required: 17% less than Crete and 45% less than Finland. Additionally, Finland requires a battery with 39 times more capacity than Nigeria — even if it receives 15% less solar energy, and Crete needs a battery with a capacity 27% higher than Nigeria.

Most studies in the literature focus on dimensioning the renewable infrastructure for a single data center. A similar trend can be observed in the context of scheduling problems for DCs powered by green energy: a literature review on recent publications on the subject of DCs that are supplied by renewable energy was made by Song et al. [53], and the results show that over 100 publications, only around 25% considered geographically distributed data centers partially powered by renewable energy. Additionally, the authors reported that most research on integrating renewable sources in cloud data centers focuses on the scheduling of the workload, and few articles analyze the dimensioning of the renewable infrastructure.

As an example of a research effort that explores the scenario of sizing for multiple geographically distributed DCs, we have the Carbon Explorer framework[1]. The study's objective is to supply DCs located in the United States exclusively with renewable power, and three strategies were used: i) only use green power; ii) use renewable power and energy storage devices; and iii) use green energy and workload scheduling. It is assumed that the DCs have access to low-carbon intensive power from the regular electrical grid — solar, wind, or both — and additional renewable infrastructure will be built — PV panels, wind turbines, and batteries, taking into account the emissions of manufacturing. The study performed an exhaustive search to evaluate the possible scenarios. In conclusion, the authors report that when taking into account the scenario of geographically distributed data centers and the manufacturing carbon emissions of the renewable infrastructure, there may be better solutions than powering the cloud federation exclusively with green power. Finally, deciding the optimal solution is still an open research question for future work.

This thesis explored the dimensioning for geographically distributed data centers worldwide. The proposed solution combined both "follow-the-renewables" for scheduling the workload and sizing the renewable infrastructure to reduce the carbon footprint of a global cloud federation. Compared to the Carbon Explorer framework, our proposed solution considers the usage of the local electrical grid

when viable, given it may have the presence of low-carbon intensive sources. Finally, our model uses a linear program formulation, in contrast to most of the works from the literature that use MILP.

In Chapter 4, the reader is introduced to our initial modeling of the problem and the evaluation of the short-term scenario (1 year) for reducing the carbon footprint of operating a cloud federation. In Chapter 5, an extension of the modeling is presented, focusing on the long-term operation, taking into account the life-cycle of the renewable infrastructure, another source for renewable power, scheduling strategies to reduce the carbon footprint further, and manufacturing servers taking into account their carbon footprint. Additionally, one new dimension of the problem is explored: the monetary costs of reducing the environmental impact with the on-site renewable infrastructure.

2.4 Measuring the environmental impact of cloud computing platforms

The GHG protocol [55] was developed as a way to standardize the measurement and the report of the companies' environmental impact. It considers six types of Green Houses Gases of the Kyoto protocol: carbon dioxide (CO_2), methane (CH_4), sulfur hexafluoride SF_6, nitrous oxide (N_2O), hydrofluorocarbons (HFCs), and perfluorocarbons (PFCs).

The protocol has three scopes to quantify the direct and indirect emissions:

- **Scope 1 - Direct GHG emissions**: Takes into account the emissions that are generated by sources that the company owns or has control over. In the context of cloud computing, it could come from the electricity generated when there is a power surge, and generators should be used as a backup; transportation of

products, materials, employees using the company vehicles; and the cooling of the data center infrastructure [20].

- **Scope 2 - Electricity indirect GHG emissions**: Takes into account the emissions generated from the electricity consumed.

- **Scope 3 - Other indirect GHG emissions**: This one is optional, and takes into account emissions from the production or extraction of materials and fuels bought, for example, IT equipment (servers, routers, and so on) as well as constructing the data centers facilities. Emissions from transportation are also included in this scope if they come from vehicles not owned by the company.

Major cloud players already provide sustainability reports and use the GHG Protocol. In the 2023 sustainability report from Meta[35], it is estimated that 1% of the emissions came from Scope 1, less than 1% from Scope 2, and 99% from Scope 3. Additionally, it is stated that the operational emissions have been reduced by 94% in comparison to 2017 thanks to the renewable energy commitments. Another remarkable example is the Google environmental report from 2023[17], where it is estimated that 1% of the emissions come from Scope 1, 24% from Scope 2, and 74% from Scope 3. Google also informs that most of the Scope 3 emissions come from manufacturing hardware and capital goods they bought and constructing the data center facilities.

Regarding the carbon-responsive strategies explored in this thesis, the follow-the-renewables focuses in Scope 2 — schedule the workload to the location that uses less carbon-intensive power from renewable sources, and the sizing of the renewable infrastructure considers the 1, 2, and 3 — the direct carbon emissions from generating electricity in the DC, emissions from consuming power from the regular electrical grid, and other indirect GHG emissions that come from the manufacturing and life cycle of the renewable infrastructure and IT equipment. Finally, in this thesis, the environmental impact is only evaluated in terms of carbon footprint —

using the metric CO_2 equivalent, or CO_2-eq. This decision was taken due to the lack of data for measuring other environmental impacts, such as water usage, and impacts caused by extracting raw minerals to fabricate the integrated circuits.

2.5 Summary

This chapter presented an overview of the context of cloud computing, scheduling, and the concept of carbon-responsive computing to describe the strategies for reducing its environmental impact, in particular the follow-the-renewables and the sizing that are explored in this thesis, and how it is possible to measure the environmental impact of the cloud computing platforms using the GHG protocol. The content presented in this chapter should provide the reader with adequate information to comprehend the succeeding chapters of this thesis.

3 Impact of the follow-the-renewables approaches in energy consumption

3.1 Introduction

The “follow-the-renewables” approach (the reader may consult Chapter 2 for more details) is an interesting strategy for mitigating the intermittency of renewable energy availability without the need to use energy storage devices, for example, when is night in the location where a DC A resides, the workload could be migrated to another DC B where the sun is still shining to use solar power. However, this strategy also has some limitations. First, the process of migrating a job between different DCs consumes energy itself: it uses network devices (such as switches and routers) and a computational task for the live-migration process. The scheduling algorithm must consider this energy consumption before deciding if the migration is advantageous. Second, the network communications links that connect the servers inside the DC and the different DCs can suffer from congestion, which may increase migration duration. This results in unnecessary computation and energy consumption on both the origin and the target server: the server that is sending the job needs to wait for the migration process to finish to free its resources and receive additional jobs, or it could be turned off as well to save energy; and the server that is receiving the job will only start executing the new job after the end of the migration process. An

efficient scheduling algorithm must consider those factors to decrease the carbon footprint of the DCs operations.

The work by [9, 8] studied the scheduling of the cloud workload, in the form of VMS, on geographically distributed DCs to minimize non-renewable energy consumption. It proposed different stochastic models to estimate renewable energy production and greedy heuristic algorithms to allocate tasks to servers. VMs are allowed to be migrated during the execution to a computer within the same DC (intra-DC server consolidation) or to a computer in DC located in another geographic location ("follow-the-renewables"). The scheduling algorithm considers the network cost of the migration.

This chapter presents an extension of this work with an analysis of the indirect impact of using the follow-the-renewables approach in energy consumption. This impact can be divided into two: direct and indirect. The network devices' energy consumption causes the direct impact, and the power consumption is considered static since it does not vary significantly based on the device usage [26]. For the indirect impact, it is generated by the live-migration of the jobs, which uses the network to transfer all the data related to the task to the destination machine, as well as a computational task required to perform this live-migration. More specifically, this chapter presents the following contributions:

- an analysis of the impact of not considering the network (bandwidth, latency and topology) in both the energy consumption and network congestion
- an accurate estimation algorithm for the time it takes to migrate a virtual machine that takes into account the network topology, links bandwidth, and latency
- a scheduling algorithm for the live-migration of the jobs that uses the estimation algorithm and has the same or lower brown energy consumption than the baselines, with no network congestion

The work of this chapter resulted in the following publication: ***Vasconcelos, M. F. S.**, Cordeiro, D., and Dufossé, F. (2022). Indirect network impact on the energy consumption in multi-clouds for follow-the-renewables approaches. In Proceedings of the 11th International Conference on Smart Cities and Green ICT Systems — SMARTGREENS, pages 44–55. INSTICC, SciTePress*. The text of the present chapter is an adaptation of this publication.

This chapter is organized as follows. In Section 3.2 the model of [8], the fundation of this work is summarized. Section 3.3 details the new scheduling method for the migrations, while Section 3.4 is devoted to simulations parameters. Results are detailed in Section 3.5. Finally, Section 3.6 summarize the chapter.

3.2 The NEMESIS modeling

The resource management framework NEMESIS [8] manages a multi-cloud composed of DCs geographically distributed across a country. The DCs power demand is supplied from the regular electrical grid and locally installed PVs. Given the intermittency of renewable energy, the NEMESIS algorithm uses the stochastic modeling of SAGITTA [9] for obtaining the expected value of the renewable energy available at a given time to be used as input for the scheduling algorithms.

NEMESIS has a central controller and the workload execution is scheduled at time slots of 5 minutes. The workload consists of heterogeneous VMs in terms of the number of CPU cores, RAM size, and requested execution time. The scheduling decision uses greedy heuristics inspired by the Best-Fit algorithm. While it may not result in the optimal solution, greedy heuristics can provide an acceptable result in a reasonable amount of time, and these characteristics are relevant for the scenario of cloud computing with millions of jobs submitted per hour and a significant portion

of these tasks have strict quality of service requirements and cannot be delayed. The scheduling algorithm has four main steps detailed as follows.

In the first step of the algorithm (pre-allocation of the incoming VMs), the controller will search for servers to allocate the VMS received during the time slot. There are two restrictions for this scheduling algorithm: i) the server has available computational resources to host the VM; and ii) executing the VM in this server would result in the minimum increase in the expected brown energy consumption. The algorithm first sorts the VMs by their volume (a product of the number of CPU cores and the amount of RAM) in decreasing order and then performs the search for each of the VMs. If a server is found that respects the two restrictions, the algorithm makes a reservation (pre-allocation) for the VM being analyzed and goes to the next VM. On the other hand, if no server is found, the VM will be delayed to be processed in the next time slot.

The second step of the algorithm (revision of the pre-allocations) performs a revision of the reservations made by the previous step, given that the greedy heuristic used may not provide the optimal solution. The strategy is to move the reservation from the DCs expected to consume more brown energy to the DCs expected to have the most availability of green energy. There are two constraints for this algorithm: i) there exists a server in the DC being evaluated that can host the VM; and ii) the expected brown energy is reduced.

The availability of green energy may change during the execution of the VMs, and the third step of the algorithm (migration of the running VMs) aims to reduce brown energy consumption by migrating the running VMs. It uses the same strategy as the second step of the algorithm (moves the workload from DCs using more brown power to DCs that have more green power available) and with the following restrictions: i) the migration needs to finish before the beginning of the next time slot; ii) the remaining execution time of the VM needs to be greater than the duration of the migration process; iii) one DC can only migrate to another 2 DCs during a

time slot; and iv) the migrations from one DC are planned to execute one after another, that is, they cannot happen simultaneously in parallel. Restrictions (iii) and (iv) are simple heuristics to avoid network congestion with the load generated by the VM live migrations.

The availability of green energy may change during the execution of the VMs, and the third step of the algorithm (migration of the running VMs) aims to reduce brown energy consumption by migrating the running VMs. It uses the same strategy as the second step of the algorithm (moves the workload from DCs using more brown power to DCs that have more green power available) and with the following restrictions: i) the migration needs to finish before the beginning of the next time slot; ii) the remaining execution time of the VM needs to be greater than the duration of the migration process; iii) one DC can only migrate to another 2 DCs during a time slot; and iv) the migrations from one DC are planned to execute one after another, that is, they cannot happen simultaneously in parallel. Restrictions (iii) and (iv) are simple heuristics to avoid network congestion with the load generated by the VM live migrations.

3.2.1 Cloud Modeling

NEMESIS cloud modeling is based on a real cloud infrastructure: the Grid'5000 testbed[1]. It is considered a subset of the original platform with 1035 servers distributed among 9 data centers in France and Luxembourg: 116 servers in Grenoble; 74 servers in Lille; 38 servers in Luxembourg; 103 servers in Lyon; 240 servers in Nancy; 44 servers in Reims; 129 servers in Rennes; 151 servers in Sophia Antipolis; and 140 servers in Toulouse. The servers are considered homogeneous in terms of memory, CPU, and energy consumption, and are based in the Taurus node of the

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

Grid'5000, equipped with two Intel Xeon E5-2630 CPUs (6 cores per processor), and 32 GB RAM.

Regarding the network, it is modeled both the connection of the servers inside the DCs (network links with 1 Gbps of bandwidth) as well as the connection between the different DCs (network links with 10 Gbps bandwidth). Figure 3.1 presents the cloud platform's network topology (the DCs' placement was not based on their geographic location but to visualize the network links better). One can observe that multiple DCs share some network links, thus the migration planning needs to consider this information to avoid generating network congestion and the resulting waste of resources.

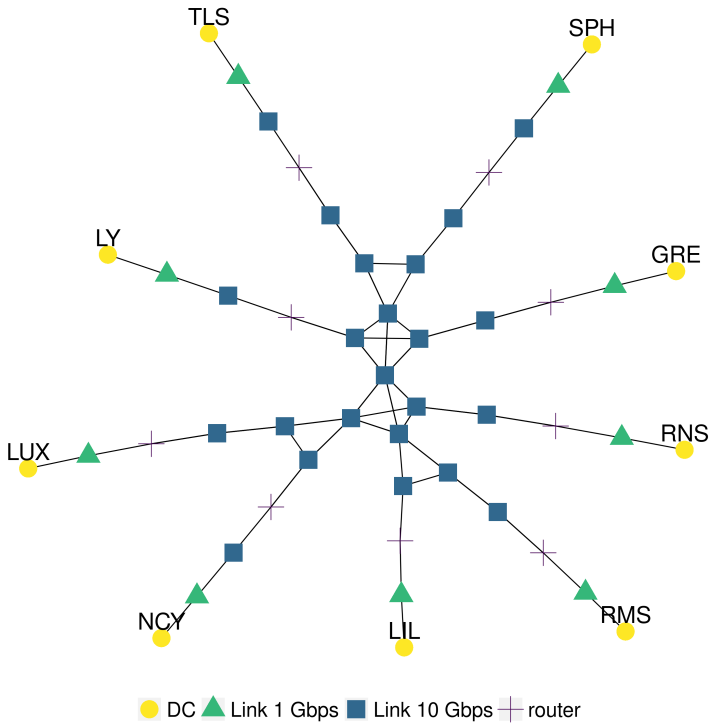

Figure 3.1 Topology of the Cloud platform, where "GRE" is Grenoble, "LIL" is Lille, "LUX" is Luxembourg,"LY" is Lyon, "NCY" is Nancy, "RMS" is Reims, "RNS" is Rennes, "SPH" is Sophia, and "TLS" is Toulouse.

Regarding the energy consumption of the servers, it is considered a linear model based on CPU usage, and the VMs always use 100% of the requested number of CPU cores. The server presents a fixed consumption for its IDLE state (97 W),

and the power consumption based on core usage is as follows: 128 W for 1 core; 146.4 W for 2 cores; 164.8 W for 3 cores; 183.2 W for 4 cores; 201.6 W for 5 cores; and 220 W with 6 cores. Furthermore, the energy consumption of turning on a server (127 W during 150 s), turning off a server (200 W for 10 s), and when the server is off (8 W) are modeled as well. Finally, only the power consumption of the servers is considered to model the power consumption of the DCs (Power Usage Effectiveness, or P.U.E., equals to 1), as we are focusing on the energy consumption of the computing part — the major consumer. Choosing a P.U.E. different than 1 would not affect the scheduling decisions made by the migration planning algorithm, since only the total power consumption would be increased by a constant factor, and the ordering of the DCs in terms of green energy availability would be the same. Furthermore, we consider a homogeneous P.U.E for all the DCs, given that the DCs are inside the same country.

3.2.2 VM live migration model

The VM live migration model of NEMESIS has 3 phases. In the first phase, all the memory pages of the VM are transferred to the destination server. In the second, after completing the copy of all the memory pages, a message is sent to notify the end of the stop-and-copy step. Finally, in the last step, the commitment message is sent to the new host to inform the end of the migration process, and the VM will be destroyed in the server of origin and resumed in the destination host.

The duration of the migration is essential information for the scheduling decision, as it can help avoid oversaturating the network links and generating network congestion. Algorithm 1 is an extension of the NEMESIS algorithm, and it executes this estimation, where: *linkLatency* is the latency of the link; *routeSize* is the number of links that interconnects the host where the VM is originally running to the target host of the migration; *windowSize* = 4,194,304 Bytes, is the TCP maximum window

size; *bandWidthRatio* = 0.97, represents the additional load caused by the headers of TCP/IP; *bandwidth* = the minimum bandwidth among the links that interconnect the host of origin with the target host; $\alpha = 13.01$, simulates the TCP slow start factor, that is, not all the bandwidth is instantly available for the communication; and γ is used to represent the Bottleneck effect of the TCP protocol. The parameters used in Algorithm 1 were based on [58]. The difference between Algorithm 1 and NEMESIS is the *routeSize* variable: NEMESIS used fixed values (5 for live migrations intra-DC, and 11 for live migrations inter-DC), and now is assumed that the cloud operator has information about the network topology of his DCs, thus the real number of links that interconnect the two hosts is used.

Algorithm 1 Estimation of the migration duration.

$theLatency \leftarrow linkLatency \cdot routeSize$
$transferLat \leftarrow theLatency + \frac{\gamma}{bandwidth}$
$throughputL \leftarrow \frac{windowSize}{2 \cdot transferLatency}$
$throughputB \leftarrow bandwidth$
$throughput \leftarrow min(throughputL, throughputB)$
$throughput \leftarrow throughput \cdot bandWidthRatio$
$timeToMigrate \leftarrow 3 \cdot \alpha \cdot transferLat + \frac{vmRamSize}{throughput}$

3.3 Planning the migrations

Algorithms 2 and 3 plan the VM migrations considering the network topology and the bandwidth usage. They are inspired in the migration planning of NEMESIS, with the following modifications: i) the available bandwidth of the links is considered (it changes based on the number of migrations that use the same network link); ii) migrations can be performed in parallel between different DCs respecting the links bandwidth constraint; iii) the intra-DC migrations (for the server consolidation) are sequentially distributed in time (do not execute simultaneously) ; iv) the real number of links that interconnects the origin and the target server is used as input for the estimation algorithm of the migration duration.

Algorithm 2 General planning of the migrations.

DCs ▷ Sorted by increasing ERGE
$plannedTime \leftarrow 0$
$timeSlotDuration \leftarrow 300$
$linksHistory \leftarrow \emptyset$
while $plannedTime < timeSlotDuration$ **do**
  $dc\_tx \leftarrow$ first item of DCs
  while $dc\_tx \neq$ last item of DCs **do**
    $dc\_rx \leftarrow$ last item of DCs
    if dc_tx has VMs that can be migrated **then**
      while $dc\_tx \neq dc\_rx$ **do**
        evaluate if it is worth to migrate VMs from dc_tx to dc_rx using Alg. 3
        $dc\_rx \leftarrow$ previous DC of DCs
      end while
    end if
    $dc\_tx \leftarrow$ next DC of DCs
  end while
  if no VM migration was planed and no migration is in execution **then**
    exit
  end if
  $plannedTime \leftarrow$ instant after the expected end of next migration
end while

Algorithm 3 Detailed migration planning between two DCs.

Require: dc_tx, dc_rx
$VMs \leftarrow$ list of VMs of dc_tx
for vm in VMs **do**
  $origin \leftarrow$ server where the vm is running
  $target \leftarrow$ server from dc_rx being evaluated
  $worth \leftarrow brownMig < brownNotMig$
  $e\_time \leftarrow$ estimate the migration time using Alg. 1
  $band\_ok \leftarrow$ links between $source$ and $target$ can receive the additional load of the migration
  if $worth$ and $band\_ok$ **then**
    registers the planning of the migration and updates the link history
  end if
end for

The algorithms work as follows. Initially, the DCs are sorted by increasing order of expected remaining green energy (ERGE). The DC with the most available energy (initially at the beginning of the list) is denoted by M, and N is the DC with the least green energy (initially at the end of the list). The main idea of the algorithms is to migrate VMs from the DCs that are expected to use brown energy to the DCs that are expected to have green energy. The algorithm receives as input the information of the running VMs (grouped by the servers) of the DC M. The planning starts at the beginning of the time slot. For each VM that can be migrated, the algorithm tries to find a server in the destination DC N with the following restrictions: i) it has available computational resources to host the VM (CPU and memory); ii) the links that interconnect the server of origin (where the VM is running) and the target server can receive the additional load of the migration without violating their bandwidth constraint; iii) the VM migration finishes during the current time slot (if the destination server is off, the time to turn the server on is taken into account); and iv) performing the migration reduces the expected brown energy consumption. If all these restrictions are respected, the VM is planned to migrate, and the algorithm registers in the *linkHistory* that the links that connect these two servers will be used until the instant when the VM migration is expected to finish in order to compute the available bandwidth of the network links. If there are still remaining VMs that could be migrated from DC M, the algorithm will try to migrate to the DC N-1, and so on, until all the VMs from DC M are planned to migrate, or all the DCs are processed. After finishing processing for the DC M, the algorithm repeats the same process for the DC M+1 until all the DCs are processed.

After evaluating all the possible migrations for the instant at the beginning of the time slot, the algorithm will use the link usage history to obtain information about when there will be available links in terms of bandwidth, that is, at what instant of time `t` the next migration is expected to finish. If there are still VMs that could be migrated, the migration planning algorithm will be executed again using the

availability of the bandwidth at instant *t*. Then, the process repeats until there are no more VMs to migrate or the evaluation time reaches the end of the time slot.

Regarding the server consolidation, there are two differences to the original NEMESIS algorithm: i) it will only be applied to the DCs that didn't have inter-DC migrations planned to avoid generating network congestion — given that the intra-DC migrations could use the same network links as the inter-DC migrations planned in the previous step; and ii) the migrations are distributed in time using the estimation computed with Algorithm 1 to avoid overlapping them.

The new algorithm is called c-NEMESIS, where the "c" stands for congestion and its full name is "Congestion and Network-aware Energy-efficient Management framework for distributEd cloudS Infrastructures with on-Site photovoltaic production", an extension of the NEMESIS algorithm with modifications in the steps "migration of the running VMs" and "server consolidation".

The computational complexity of the Algorithm 3 is $O(n_{VMS} \times n_{servers} \times n_{links})$, where n_{VMS} is the number of running VMs on the DC that is sending the VM, $n_{servers}$ is the number of candidate servers that have the least possible amount of free cores to run the VM in the destination host, and n_{links} represents the number of links that interconnects the VM's host of origin and the target host. For Algorithm 2, the computational complexity is given by $O(n_{DCs} \log n_{DCs} + {n_{DCs}}^2 \times n_{VMS} \times n_{servers} \times n_{links})$, where n_{DCs} is the number of DCs.

3.3.1 Energy cost of migrations

The energy consumption of a live VM migration is modeled in NEMESIS by a computation task that is executed in the target host. This task uses 100% of a single CPU core during the migration process. Suppose the migration is impacted by network congestion. In that case, its duration will increase, resulting in a wastage of energy from the server where the VM was initially running as in the target server.

The wasted energy is proportional to the extra time migrating, that is, the difference between the duration of the migration process compared to the duration that it would take if there were no network congestion. A lower bound for the wasted energy can be computed with Algorithm 4 using as input this extra time migrating. Given that the migration planning uses a "follow-the-renewables" approach to move the workload from the DCs that are using brown power at that instant to the DCs that have more available green energy, the algorithm computes individually the wasted energy in the server that is sending the VM and in the server that is receiving the VM. For the server that is sending the VM, this extra energy consumed is brown(er), increasing the overall brown energy consumption of the cloud, since this server could be turned off to save energy. For the server that will receive the VM, this extra energy is green(er), and it is wasted because it was only available at that instant of time (no energy storage devices) and could have been used to execute the workload.

Algorithm 4 Extra energy consumption of migrating.

$pCore \leftarrow 20.5$
$wastedOrigin \leftarrow 0$
$wastedTarget \leftarrow 0$
for mig in $Migrations$ **do**
    $vmCores \leftarrow$ amount of cores of the VM being migrated
    $extraTime \leftarrow mig.Time - mig.TimeNoCong$
    if $extraTime > 0$ **then**
        $wastedOrigin += extraTime \cdot pCore \cdot vmCores$
        $wastedTarget += extraTime \cdot pCore$
    end if
end for

The value of *pCore* is an estimate for the additional cost of executing a core, and is obtained as follows: the server consumes $220\,\mathrm{W}$ at full load and subtracting the power consumption of the IDLE state ($97\,\mathrm{W}$) it results in $123\,\mathrm{W}$. Finally, this value is divided by the total amount of cores of the server (6), resulting in $20.5\,\mathrm{W}$. Notice that *pCore* is only multiplied by the number of cores of the VM for the server of

origin, since it remains executing the VM until the end of the migration, and in the destination server the computational task uses only a single CPU core.

3.4 Experiments

The experiments were performed using computational simulations. The Simgrid [10] (version 3.28) framework was used to develop the simulations because it allows modeling distributed computing experiments, such as cloud platforms, in particular: energy consumption of executing the tasks, turning on and off the servers, network usage by the live-migration process, network topology and the network congestion. Another relevant reason for choosing the Simgrid framework is the fact that it is well-validated by the scientific community with over 20 years of usage. For the network, the default flow-level TCP modeling of Simgrid produces precise results for large distributed computing scenarios (as in our case with thousands of servers) in a reasonable amount of time. It is possible to use packet-level simulation, however, despite being more precise, it would result in huge execution time for the simulations [58]. Regarding the cloud infrastructure, it is considered the adapted version of the Grid'5000 (same as NEMESIS, see Section 3.2.1). Finally, it was simulated one week of the multi-cloud operation.

3.4.1 Workload

Traces from real cloud providers were used to generate the workload for the experiments. The data extracted from the traces were: the number of CPU cores requested, the instant of time when the task was submitted, and its duration. Regarding the requested RAM per VM, it is considered that each VM will consume 2 GB per core requested, similar to the `t2.small` instance of Amazon EC2[2]. It is also considered

[2] `https://aws.amazon.com/ec2/instance-types/`

that the VMs execute with a full CPU usage of the requested cores, the worst scenario for energy consumption. The workloads are scaled to use at a maximum of 80% of the computational resources of the cloud platform at any given simulated time. This decision ensures that the tasks will always be allocated to the servers, and that there will be some room for the servers to receive VMs from live-migrations, thus allowing analyzing the different scheduling approaches. Furthermore, from the cloud traces, it is possible to observe that the DCs are not used 100% of their computational capacity. Figure 3.2 illustrates the number of VMs submitted during the week (in yellow), and the cumulative demand of CPU cores requested at a given time (in purple), that is, the total number of CPU cores used by the VMs in execution at that instant of time.

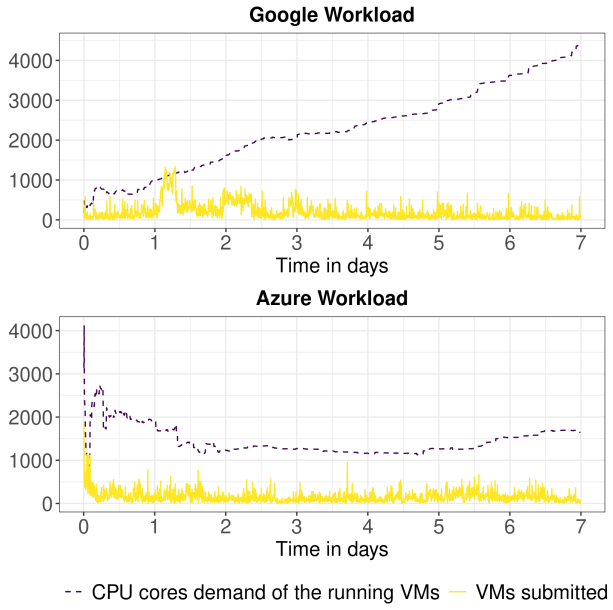

Figure 3.2 Workloads used as input for our simulations.

The first workload was generated using as a base the 2011 Google Cluster Workload traces [44], and it consists of over 380,000 VMs. In this workload, the VMs have a long execution time, as can be seen in Figure 3.2 that the value of the running VMs' CPU cores demand keeps increasing throughout the week. The second workload was based on traces from Microsoft Azure [22], more specifically, the Azure Trace

for Packing 2020, and contains over 304,000 VMs. This generated workload has a different behavior than the previous one: there is a peak in the VM submissions at the beginning of the week, and the VMs have a shorter execution time, as seen that the CPU core demand does not keep increasing during the week.

These traces do not provide data about the network usage of the tasks and it was not modeled in the simulations. The additional load in the network is generated only by the live migration of the VMs. Therefore, this work can be seen as a lower bound for the real world scenario.

3.4.2 Green energy traces

The data for the energy produced by the solar panels were obtained from the Photovolta[3] project by *Université de Nantes*. The data represents the power production of two arrays of 4 PVs each (Sanio HIP-240-HDE4) with a total nominal power production of 1920 Wc at intervals of 5 minutes. In order to simulate the different production of the geographically distributed DCs, a different week of the production trace was used for each DC. Furthermore, each DC had 3 PV panels per server. The PVs installed in the DCs generated the following amount of energy during the simulated week: i) Grenoble: 1.58 MWh; ii) Lille: 0.07 MWh; iii) Luxembourg: 0.15 MWh; iv) Lyon: 1.19 MWh; v) Nancy: 2.16 MWh; vi) Reims: 0.38 MWh; vii) Rennes: 1.63 MWh; viii) Sophia: 1.75 MWh; and ix) Toulouse: 1.53 MWh. In total, around 10.5 MWh of green power was produced in the simulated week. Figure 3.3 shows the green power production per DC during the simulated week.

[3] http://photovolta.univ-nantes.fr/

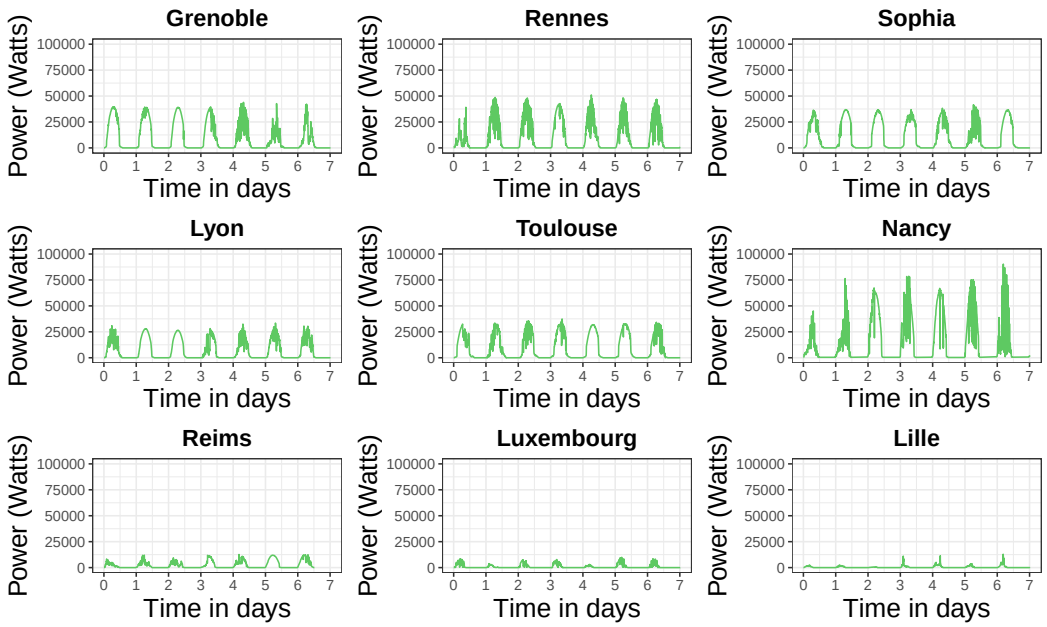

Figure 3.3 Green power production (in Watts) produced by DC during the simulated week.

3.5 Results

In this section, the results of the simulations are presented. First, it is shown an analysis of the accuracy of an essential component of NEMESIS and c-NEMESIS algorithms: the algorithm for the duration of migration (Algorithm 1. Then, it is presented an analysis of the live migrations performed and their impact on network congestion, wasted energy, and the total and brown energy consumption. To further evaluate the effectiveness of using “follow-the-renewables approaches”, results from two other state-of-the-art works (at the time the article was written, november of 2021) are presented that incorporate this approach in different ways.

The first is the Workload shifting non brownout (WSNB) algorithm [63]. The algorithm works as follows. Initially, the VM is allocated to the nearest DC (aiming to reduce the response time) and to the server that will increase the energy consumption by the least (a server that is already on and its available computational resources are equal to or slightly greater than the requirements of the requested VM). Then, suppose this initial DC does not have available renewable power to execute the

VM. In that case, the algorithm will evaluate another DC (the DCs are sorted by available green energy) that matches this demand and does not exceed a threshold for the response time. If another DC is found, the VM will be reallocated to it. The algorithm does not perform server consolidation with live-migrations, it tries to use the minimum number of servers during this allocation. To adapt this baseline for the experiments, the response time restriction was removed from the algorithm, that is, we consider that all the DCs are homogeneous in terms of latency for the VM request, since the used workloads do not have this data, and this modification does not change the behavior of the algorithm.

The second work from state of the art is the FollowMe@Source (or FollowMe@S) algorithm [3] that has two variants: i) FollowMe@S Intra, that only perform VM migrations inside the same DC (intra-DC migrations); and ii) FollowMe@S Inter, that only perform VM migrations between different DCs (inter-DC migrations). Both algorithms have the same general steps, described as follows.

The FollowMe@S algorithm has two main steps: allocation and migrations. The allocation step begins by sorting the DCs by the availability of green energy. Then, the algorithm will search for a server that can host the VM, respecting the sorted list of DCs. If no server was found after searching through all DCs, the VM will be processed in the next scheduling round. The process will be executed again for all the other VMs to be scheduled. In the migration step, either only intra-DC or inter-DC migrations are performed. First, the algorithm obtains the list of all the VMs that are in execution at that instant of time. Then, it evaluates the usage of the servers to obtain the underutilized hosts (less than 20% of CPU usage) and marks them to be turned off. The running VMs of these underutilized hosts will be removed using live-migrations in order to turn them off and save energy. In the intra-DC case (FollowMe@S Intra), the algorithm will run in each DC separately. For each VM, it will try to find a server with available computational resources to host it, and the migration is performed if a server is found. In the inter-DC case (FollowMe@S

Inter), for each VM that can be migrated, the algorithm will evaluate the DCs sorted by the availability of green energy, and migrate the VM to the server of the first DC that can host the VM. One important point to notice is that FollowMe@S does not consider the network to schedule its migrations. The following modifications were performed in the algorithm to adapt it for the experiments: i) it is not considered the network costs of a distributed algorithm, since NEMESIS, c-NEMESIS, and WSNB are centralized algorithms; ii) the workload degradation performance by migrating the VM — for example migrating to a server with a less efficient CPU — is not modeled as a homogeneous platform is used.

The baselines adopt the "follow-the-renewables" approaches in different ways. The algorithms WSNB and FollowMe@S Intra only use "follow-the-renewables" for the initial scheduling, and the VMs are not migrated to other DCs during their execution. On the other hand, the algorithm FollowMe@S Inter performs VM migrations to other DCs — the same strategy adopted by NEMESIS and c-NEMESIS.

A public Git repository hosts all the simulation code, the traces for the workloads and green power production, and the instructions to run and extract the results[4]. Finally, the results from the experiments using these algorithms and workloads are deterministic. Therefore, only results for a single execution of the simulations are presented.

3.5.1 Accuracy of the estimation algorithm

We define an underestimated migration as a VM live migration process with a longer duration than estimated. Table 3.1 presents the number of live migrations that were underestimated by NEMESIS and c-NEMESIS, and the percentage value that is based on the total number of migrations (that can be found in Table 3.3). For

[4]https://gitlab.com/migvasc/c-nemesis

both workloads, c-NEMESIS presented almost no underestimation for the migrations compared to NEMESIS. Furthermore, it is interesting to notice that virtually all the intra-DC migrations were underestimated in NEMESIS because they start simultaneously, resulting in network congestion.

Table 3.1 Number of underestimated live migrations and the ratio of the overestimation, where "W" stands for "Workload", "G" for Google, and "A" for Azure.

Algorithm	W	Inter	Intra
NEMESIS	G	245 (4.4%)	3 393 (95.7%)
c-NEMESIS	G	0 (0%)	61 (2.9%)
NEMESIS	A	140 (3.6%)	3 324 (94.1%)
c-NEMESIS	A	24 (0.1%)	49 (3.8%)

In order to evaluate the estimation algorithm for the duration of the migrations, two metrics are used to assess its accuracy: the Mean Absolute Percentage Error (M.A.P.E.) and the Root Mean Square Error (R.M.S.E.). The M.A.P.E. is defined by: $\frac{1}{n}\sum_{i=1}^{n}\frac{|R_i-F_i|}{R_i}$, where n represents the amount of values being considered, i the index of the value being considered, R_i the real duration of migration, and F_i the estimated duration. The results of the M.A.P.E. is a percentage value, and it represents the relative value of the estimation errors compared to the original value, thus it is an easy-to-understand metric. The R.M.S.E. is defined by: $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(R_i-F_i)^2}$. The R.M.S.E. is a metric similar to the standard deviation, and it allows us to validate how far the estimation was from the original value.

Table 3.2 Accuracy measurements, where "W" stands for "Workload", "G" for Google, and "A" for Azure. The M.A.P.E. value is in percentage (%), and the R.M.S.E. in seconds.

Algorithm	W	M.A.P.E.	R.M.S.E.
c-NEMESIS	G	0.70	0.395
NEMESIS	G	32.895	18.56
c-NEMESIS	A	0.649	0.432
NEMESIS	A	34.01	20.03

Table 3.2 presents the results for the accuracy measurements. It is possible to observe that c-NEMESIS is accurate with low error values. Two points justify the improvement in the precision of the estimation algorithm: i) the new algorithm has full information on the network topology and uses the actual number of links

that interconnect the servers involved in the migration process; ii) since it is more precise, the migration planning results in less network congestion, which affects the duration of the migration.

3.5.2 Analysis of the VM live migrations performed

Table 3.3 presents the number of live migrations performed during the simulated week. NEMESIS performed the lowest number of inter-DC migrations for both workloads, since its migration planning does not allow for migrations in parallel leaving or arriving at the same DC. The c-NEMESIS algorithm executed more inter-DC migrations than NEMESIS, as it takes into account the network topology and allows for migrations in parallel. c-NEMESIS had the lowest number of intra-DC migrations for two factors: i) the server consolidation step is only executed for the DCs that do not have inter-DC migrations planned in a time slot; and ii) intra-DC migrations are distributed in time. The FollowME@S algorithm had the highest number of inter and intra-DC migrations because the planning does not consider network usage.

Table 3.3 Number of VM live migrations performed, where "W" stands for "Workload", "G" for Google, and "A" for Azure.

Algorithm	W	Inter	Intra
NEMESIS	G	5 617	3 545
c-NEMESIS	G	18 056	2 074
FollowMe@S Intra	G	0	560 862
FollowMe@S Inter	G	96 464	0
NEMESIS	A	3 863	3 532
c-NEMESIS	A	17 479	1 300
FollowMe@S Intra	A	0	177 086
FollowMe@S Inter	A	93 388	0

3.5.2.1 Total and brown energy consumption of the cloud platform

Table 3.4 presents the cloud platform's total and brown energy consumption for the simulated week. The c-NEMESIS performed better in terms of brown energy usage, having the lowest consumption among the other evaluated algorithms — except for NEMESIS using the Azure workload, in which the consumption was the same. Regarding the total energy, c-NEMESIS consumed more than NEMESIS because it performed more migrations. On the other hand, more green energy was harnessed, since the brown energy consumed was the same or lower. Regarding the FollowME@S algorithm, both inter and intra versions had similar brown energy consumption, but the inter-DC approach had a marginally lower brown energy consumption than the intra-DC (around 0.3% for the Google workload and 0.04% for the Azure workload). The WSNB algorithm had the highest consumption of both total and brown energy.

Table 3.4 Comparison of energy consumption (MWh), where "W" stands for "Workload", "G" for Google, and "A" for Azure.

Algorithm	W	Total	Brown
NEMESIS	G	25.46	17.23
c-NEMESIS	G	25.56	17.18
FollowMe@S Intra	G	27.56	19.13
FollowMe@S Inter	G	27.59	19.07
WSNB	G	29.49	20.89
NEMESIS	A	30.43	21.21
c-NEMESIS	A	30.55	21.20
FollowMe@S Intra	A	31.69	22.41
FollowMe@S Inter	A	31.69	22.40
WSNB	A	33.56	24.23

The difference between the total and brown energy consumed with the green energy generated in the simulated week (10.5 MWh) is compared to obtain the green energy usage of the algorithms. For the Google workload, the usage of green energy was: c-NEMESIS = 80%, NEMESIS = 79%, FollowMe@S Intra = 80%, FollowMe@S Inter = 81%, and WSNB = 82%. Regarding the Azure workload, the usage was:

c-NEMESIS = 89%, NEMESIS = 88%, FollowMe@S Intra = 89%, FollowMe@S Inter = 89%, and WSNB = 89%.

The scheduling policies and how the algorithms adopt the "follow-the-renewables" strategy justify the difference between the total and brown energy consumption, and the relative value of renewable energy used. The algorithms WSNB and FollowMe@S Intra that presented the highest brown energy consumption applied "follow-the-renewables" at the initial scheduling of the workload, and didn't migrate the VMs in execution to other DCs as the green energy availability changed over time — as did by the algorithms FollowMe@S Inter, NEMESIS and c-NEMESIS. To further understand the difference in these results, the next section will analyze the live migrations' impact on the total and brown energy consumption.

3.5.2.2 Wasted resources in the migrations

To evaluate the wastage of resources in terms of network and energy, all the live migrations performed in the algorithms are compared with a perfect scenario: all the migrations are performed individually and isolated, having full access to the network.

Table 3.5 presents statistics about the extra time (in seconds) it took to migrate the VMs in the simulations in comparison with the perfect scenario. The absolute value is the absolute difference in seconds. For example, on average, the live migrations performed by the NEMESIS algorithm were 10.11 seconds longer compared to the perfect scenario for the Google workload. The relative value is the ratio of the difference. For example, in the FollowMe@S Inter with the Google workload, the migration duration was more than 10 times longer in comparison with the perfect scenario.

The c-NEMESIS algorithm had the best performance in terms of extra seconds spent migrating for both workloads, with values close to the perfect scenario: the average

Table 3.5 Extra seconds during migrations compared to the case when there is no congestion, where "W" stands for "Workload", "G" for Google and, "A" for Azure. "avg." for the average of the observations, "max." for the maximum value, "abs." for the absolute value, and "rel." for the relative value.

Algorithm	W	avg. abs.	max. abs.	avg. rel.	max. rel.	Total seconds
NEMESIS	G	10.11	56.98	1.53	3.98	92915.7
c-NEMESIS	G	0.22	0.88	1.0	1.05	4331.93
FollowMe@S Intra	G	113.94	1 028.51	6.37	40.82	64 525 098.3
FollowMe@S Inter	G	215.18	4 875.8	10.67	155.5	22 058 949.9
NEMESIS	A	11.62	62.96	1.59	3.98	86 235.46
c-NEMESIS	A	0.23	6.14	1.0	1.32	4 224.43
FollowMe@S Intra	A	91.08	938.48	4.39	25.56	16 384 188.8
FollowMe@S Inter	A	186.56	8 047.89	7.8	157.24	18 531 893.3

relative difference (avg. rel.) was approximately 1. The NEMESIS algorithm had a good performance as well with low extra seconds values, and the duration of the migrations took, on average, about 1.5 more than in the perfect scenario. The migrations performed by FollowME@S, both intra and inter-DC cases, had the worst performance: the duration took, on average, from 4 to 10 times more than the perfect scenario. These results highlight the importance of considering the network for the migration scheduling, since c-NEMESIS and NEMESIS presented the lowest wastage of resources.

Using the extra seconds spent in the migrations, a lower bound for the wasted energy is computed using Algorithm 4 for both the servers that sent the VM (origin) — that represents the brown(er) energy consumption — and the servers that received the VMs (target) — that represents the green(er) energy waste. Table 3.6 presents the results. The algorithm FollowMe@S (both inter and intra-DC versions) was the algorithm that wasted the most energy. The c-NEMESIS algorithm had the lowest wastage of energy overall, despite performing more migrations than NEMESIS (the second best in wastage of energy). These values are justified by the fact that wasted energy is directly proportional to the duration of the migrations, thus bad planning will congest the network and, as consequence, increase the total energy consumption. Finally, it is important to notice that the green wasted energy is not negligible and

can be used to power the cloud. For example, in the FollowMe@S intra-DC with the Google workload, the wasted green energy could have powered the Luxembourg DC for approximately 44 hours (at 100% computational performance).

Table 3.6 Wasted energy in the migrations (Wh) in comparison to the perfect scenario where the migration process have full access to the network links. In the table "W" stands for "Workload", "G" for Google, and "A" for Azure.

Algorithm	W	Origin	Target
NEMESIS	G	545.1	529.1
c-NEMESIS	G	35.42	24.67
FollowMe@S Intra	G	473 971.6	367 434.6
FollowMe@S Inter	G	153 829.96	125 613.5
NEMESIS	A	539.59	491.06
c-NEMESIS	A	39.31	24.06
FollowMe@S Intra	A	163 128.14	93 298.9
FollowMe@S Inter	A	175 086.3	105 528.8

3.6 Summary

Reducing the environmental impact of the operation of cloud computing platforms, particularly the carbon footprint generated from the country-like the energy consumption, is a complex and challenging problem currently being addressed from multiple angles. In this chapter, we focus on the strategy "follow-the-renewables", and studied the indirect impact on the energy consumption caused by the additional load generated in the network. The experiments were based on real-world data for the cloud infrastructure, workloads, and photovoltaic power production.

This chapter demonstrates that the indirect network impact on the energy consumption in multi-clouds for "follow-the-renewables" approaches is generated by bad scheduling policies for the migrations, which results in network congestion — as the live migrations compete for the available bandwidth of the network links, affecting not only the applications that are running inside the VMs, but as well the duration of the migration process. Given that migrating a VM also consumes energy — propor-

tional to its duration — and that the workload will be sent to the DCs using green(er) energy ("folllow-the-renewables"), the extra energy consumption is in reality wasted energy that could be used to power the cloud platform. Furthermore, the adoption of the "follow-the-renewables" needs to consider the whole execution of the workload: the state-of-the-art algorithms that only used the green energy information for the initial scheduling and didn't migrate the workload as the availability of renewable energy changed had the highest brown energy consumption.

The proposed estimation algorithm for the duration of the live migrations that uses as input information about the communication network of the data centers (network links latency, bandwidth and topology) is accurate. By using this estimation algorithm and using as input the network characteristics of the cloud, the migration planning of c-NEMESIS was able to increase the number of migrations by at least 3-fold without network congestion, while maintaining or reducing the brown energy consumption in comparison to other state-of-the-art works.

As future research directions, the network usage by the workload and how it will compete for network resources with the live migrations needs further investigation. It is also possible to extend the proposed solutions for other virtualization techniques as containers by updating the estimation algorithm with a model for container live migration. Finally, some approaches explore turning off the network devices to save energy. This technique reduces the available network links in the cloud platform, and the network traffic will be re-routed, and it is necessary to analyze if the energy savings are more significant than the impacts caused by the network congestion.

4 Sizing the renewable infrastructure to greener the DC operation

4.1 Introduction

In the previous chapter, we saw the adoption of the carbon-aware technique follow-the-renewables applied to a multi-cloud distributed over a country, and the drawbacks it presents if not used properly, such as network congestion and extra energy consumption. Now, the reader will be presented with another strategy: sizing (or dimensioning) the renewable infrastructure to reduce the carbon footprint of the cloud operation. This strategy consists of defining how much investment needs to be made, for example, defining the area of solar panels (when considering solar power), the number of wind turbines, and the capacity of the energy storage devices (such as lithium-ion batteries) that need to be manufactured.

The scenario considered has some differences to better represent modern cloud providers. First, it is considered that the DCs are geographically distributed over the world, which represents real cloud providers such as Google, Microsoft, Facebook, and so on, as can be seen in Figure 2.1. Many reasons justifies the need for geographically distributed DCs: i) meet the demand for the ever-increasing number of users; ii) redundancy, for example if there is a problem in some region the workload can be shifted to another location; iii) reduce the latency or response time for the users. This geographic distribution also allows to better harvest the renewables sources, since there is a more variable climate conditions, solar irradiation levels is different

between the countries, as well as the wind received. Second, many locations in the world have the presence of low carbon intensive sources in their local electricity grid, therefore in reality there is not only green and brown classification (as seen in the previous chapter) but many shades of the colors. Classifying between brown and green made sense in the previous scenario since it was considered only a single country, and the renewable energy was less carbon intensive than the local electricity grid. Finally, energy storage devices can be used to store the overproduction of renewable energy and use it when opportune.

One point that cannot be neglected is that manufacturing the renewable infrastructure also presents a environmental impact: batteries have an ideal level of charge that can improve their lifetime, which can reduce the replacement frequency, but on the other hand, causes them to be oversized [5], and their recycles rates still need to improve [41]; and considering the current state-of-the-art PVs, if they produce 40% of the global electricity by 2050, they will consume about 5% of today's CO_2 budget [60].

In this chapter, we explore the adoption of both strategies, sizing the PVs and batteries and scheduling with follow-the-renewables to reduce the carbon footprint of operating existing cloud platforms. More specifically, this chapter presents the following contributions:

- the two sub-problems—PVs and batteries sizing, and workload scheduling—are modeled as a single problem, which allows evaluating scenarios such as: should the battery capacity or the PV area be increased, or should the workload be scheduled in a data center located in another part of the world?

- we propose a model that uses a linear programming approach (LP) with real variables, allowing us to optimally solve the problem we address in polynomial time using classical LP solvers. This allows a large number of scenarios to be evaluated over broad time horizons (i.e., one year) to take

the seasonal behavior of renewable energy production into account. This model can be extended to multiple scenarios, and it may help decision-makers evaluate which regions need more investment to reduce the cloud operation's environmental impact.

The work of this chapter was done in collaboration with the following members of the DATAZERO[39] research team: Georges da Costa, Jean-Marc Nicod and Veronika Rehn-Sonigo, and it resulted in the following publication: ***Vasconcelos, M.**, Cordeiro, D., Costa, G. D., Dufossé, F., Nicod, J.-M., and Rehn-Sonigo, V. (2023). Optimal sizing of a globally distributed low carbon cloud federation. In The 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing*. The text of the present chapter is an adaptation of this publication.

The remainder of the chapter is organized as follows: Section 4.2 defines the problem addressed, the assumptions, the models, and the objective function. Details about the problem constraints and how to optimally solve the problem are given in Section 4.3. Comprehensive experiments, presented in Section 4.4, are discussed in Section 4.5 before we conclude in Section 4.6.

4.2 Problem statement

In this section, the reader will find a description of the addressed problem and hypothesis. The next two sections further detail the modeling, notations, and the optimal approach to solve the decision problem that we tackle within the chapter.

4.2.1 Addressed problem

As mentioned earlier, the goal is to reduce the carbon footprint of an existing cloud platform both in its operation and sizing the renewable infrastructure. The cloud

platform consists of several data centers spread worldwide, in both hemispheres and on all continents. The solution that will be proposed aims to design an additional solar-based power supply infrastructure to the classical power grid connection and to define an optimal way to operate the global IT cloud platform by scheduling its workload. To green the cloud infrastructure and reduce its carbon footprint, we must reduce the use of high-carbon-intensive energy to operate the data centers. Using renewable energy is a promising option, however, the carbon footprint of manufacturing needs to be taken into account.

Another problem is that the location where the DC is installed determines how sustainable it can be. First, the carbon intensity of the energy consumed from the local electricity grid depends on how it is produced: natural gas, coal combustion, hydraulic or nuclear energy. Second, each location has different climate conditions, and renewable power production will also have different efficiency. For example, solar power will be higher in locations that receive more solar irradiation, depending on whether it is near or far from the tropics. Therefore, powering the cloud federation is a balance or mix between using low-carbon electricity from the local grid and using its solar panels, with the understanding that batteries are mandatory to mitigate intrinsic solar power intermittency.

The current decision problem aims to define the additional renewable power supply architecture from solar energy to reduce the carbon footprint of the global cloud infrastructure. It is assumed that: i) the data centers are already in operation, and the sizing will only define the area of PV panels and the capacity of the batteries; ii) the DCs can be supplied from the power of the local electricity grid, power from the PV panels or power stored in the batteries; iii) job submission is centralized; iv) 100% of the jobs must be completed in time (no delay); v) no migration of jobs; vi) jobs can be executed in any of the data centers; vii) cloud platform is homogeneous regarding the IT part (number of servers, CPU cores, and model, network equipment), but the total power consumption of the DCs is different because of the power used

for cooling the DCs at each geographic location. Finally, the following inputs are considered:

- specifications of the cloud infrastructure
  - number of servers, CPU cores per server, network switches
  - power consumption of the servers (static and dynamic) and network equipment
  - Power Usage Effectiveness (PUE) to represent cooling needs
- specifications of the renewable infrastructure
  - manufacturing carbon footprint
  - technical parameters: batteries charge/discharge ratio and Max Depth of Discharge, PV efficiency to convert solar irradiation to power
- weather conditions (solar irradiation) in areas where each data center operates for the federation (time series of 1 year with one value for every hour)
- carbon intensity of the local electricity grid in g CO2 eq per kWh for each DC
- the workload computing demand from clients (time series of 1 year with one value for every hour)

Now, the models and notations will be introduced before the objective function to optimize.

4.2.2 Models and notations

To propose a solution for job operations, a decision horizon $\mathcal{H}$ is defined where job scheduling decisions can be taken. To do so, we propose discretizing $\mathcal{H}$ into K indivisible time slots with a duration of Δt such that $\mathcal{H} = K \times \Delta t$. To simplify the

Table 4.1 Main notations for the IT model for each DC^d $(1 \leq d \leq D)$ during time slot k $(0 \leq k < K)$

Δt	time slot's time duration in the unit of time $[u.t]$
$\mathcal{H}$	decision horizon $\mathcal{H} = K\Delta t$
K	number of time slots $\Delta t = 1\,\text{h} = 1\,u.t.$
k	time slot between dates $k\Delta t$ and $(k+1)\Delta t$ excluded
DC^d	a specific data center d of the cloud federation
$\mathcal{DC}$	the set of all data centers $\{DC^d \mid d = 1,\ldots,D\}$
C^d	number of CPU cores within DC^d
$Pcore$	dynamic power consumption of one CPU core at 100% of utilization
$Pidle^d$	static power consumption of all the servers of DC^d
$Pintranet^d$	power consumption of network devices of the DC^d
P_k^d	the power demand to perform tasks on DC^d during time slot k
PUE^d	Power Usage Effectiveness of data center d
$\mathcal{T}$	the workload to execute $(= \{T_i \mid 1 \leq i \leq N\})$
T_i	task i of the workload $\mathcal{T}$ $(1 \leq i \leq N)$
r_i	release date of tasks T_i
p_i	processing time of tasks T_i
c_i	number of CPU cores needed to execute task T_i
w_k	number of required CPU cores during the kth time slot
w_k^d	number of CPU cores needed during the kth time slot on DC^d

notations, we consider $\Delta t = 1u.t.$ (unit of time). In our experiments, we will assume that $\Delta t = 1h$ such that $K = 8760\,h$ with $\mathcal{H} = 1$ year. Let k be the time slot index that addresses any instant t such that $k\Delta t \leq t < (k+1)\Delta t$ with $0 \leq k < K$.

4.2.2.1 IT part model

Let $\mathcal{DC} = \{DC^d \mid d = 1, \ldots, D\}$ be the set of data centers in the cloud federation. Considering a given data center DC^d, let C^d be its number of CPU cores and $Pcore$ the energy used to power one core.

The power consumption of a cloud data center can be classified as static or dynamic [2]. For the static part, the current model considers the idle power consumption $Pidle^d$ of the servers, the Power Usage Effectiveness (PUE) to represent the power consumption used to cool the DC infrastructure, and the power consumption of the network switches $Pintranet^d$ that interconnect the servers in each data center DC^d. Regarding the latter, cloud data centers usually adopt the fat-tree topology to interconnect servers in the DC [2]. In this topology, one can compute the number of network switches needed to match the number of servers. The power consumption of the network switches is considered to be static based on actual measurements, which have shown that the consumption does not change significantly with the device usage [27]. Moreover, each geographic location has different cooling needs, therefore each DC has a specific PUE^d value.

Finally, the dynamic part of the power consumption is represented by the additional power demand generated by using the CPU cores in each data center — to execute the workload w_k^d assigned to the DC d at time slot k . Equation (4.1) represents the power consumption of each DC for each step k $(0 \leq k < K)$:

$$P_k^d = PUE^d \times (Pidle^d + Pintranet^d + Pcore \times w_k^d) \tag{4.1}$$

4.2.2.2 Workload model

Considering the workload that needs to be executed, let $\mathcal{T} = \{T_i \mid i = 1, \ldots, N\}$ be the set of N tasks that have to finish in time during the time horizon $\mathcal{H}$. Each task

Table 4.2 Main notations for the Electrical part model of each DC^d $(1 \leq d \leq D)$ during each time step k $(0 \leq k < K)$

I_k^d	solar irradiation at time slot k $[Wh/m^2]$
Apv^d	surface area of photovoltaic panels of DC DC^d $[m^2]$
η_{pv}	PV efficiency (in %) in converting solar irradiation to power
$Pgrid_k^d$	power consumed from the grid at time slot k $[W]$
Pre_k^d	power generated from the PVs at time slot k $[W]$
BAT^d	battery capacity installed in DC^d $[Wh]$
B_k^k	battery level of energy at the time $k \times \Delta t$ $[Wh]$
Pch_k^d	power charged in the batteries during time slot k $[W]$
$Pdch_k^d$	power discharged from the batteries during time slot k $[W]$
η_{ch}	efficiency of the charging process
η_{dch}	efficiency of the discharging process

T_i has the following properties: i) release date r_i; ii) processing time p_i; and iii) needs c_i CPU cores at 100% of usage to be executed. Let w_k be the total number of CPU cores needed to compute tasks during the time slot k in order to complete the workload in time. At each time step, w_k is the sum over all cores required by the tasks executed in time slot k, and given that the tasks will be scheduled to the DCs, w_k^d is the sum of the number of cores used to execute the allocated workload in $\mathcal{DC}$ (with $1 \leq d \leq D$) and $0 \leq k < K$) as shown by Equation (4.2):

$$w_k = \sum_{T_i | r_i \leq k\Delta t < r_i + p_i} c_i = \sum_d w_k^d \tag{4.2}$$

4.2.2.3 Electrical part model

The power supply of the whole cloud platform $\mathcal{DC}$ has three sources: i) the renewable energy generated by the solar panels (PVs) installed on each DC^d site; ii) the classical electrical grid $Pgrid_k^d$ of each country on which $DC^d \in \mathcal{D}$ is hosted — used as backup when the power from the renewable infrastructure is not enough; and iii) energy discharged from the energy storage devices. The storage devices are mandatory to mitigate the intermittency of solar power by storing the overproduction when the sun shines to provide energy during the night when the solar panels do not produce power. Lithium-Ion batteries have been chosen to play this role because of their good efficiency in terms of costs, power and energy density, charge and discharge rates, and self-discharge [62]. To model the fact that the cloud platform will not sell back energy to the grid, there is the constraint that the power from the grid ($Pgrid_k^d$) is always positive.

Equation (4.3) models the on-site renewable power production, it depends on the solar irradiation I_k^d received at the location of DC^d during the time slot k, on the surface area of the solar panels Apv^d and on the efficiency η_{pv} of the PVs.

$$Pre_k^d = I_k^d \times Apv^d \times \eta_{pv} \tag{4.3}$$

Batteries are systematically installed next to the PVs for the reason mentioned above. Let B_k^d be the battery level of energy, that is, the amount of energy (in Wh) at time kt stored in batteries of capacity BAT^d installed in DC^d ($B_0^d = Binit^d$ being the amount of energy at the beginning the time horizon $\mathcal{H}$). The variable Pch_k^d represents the power charged in the batteries of DC d during the time slot k (and $Pdch_k^d$ is the power discharged from the batteries). It is not possible to charge and discharge simultaneously: if Pch_k^d is greater than zero, $Pdch_k^d$ equals zero and vice versa. Regarding the batteries modeling, the charging and discharging process has an efficiency η_{ch} and η_{dch} less than 1. Lithium-Ion batteries have a low value for daily

self-discharge (0.5 % per day) [62], thus this property was not modeled. Finally, to increase the lifetime, the batteries cannot be discharged more than its Maximum Depth of Discharge.

Equation (4.4) models the battery in terms of the level of energy:

$$B_k^d = B_{k-1}^d + Pch_{k-1}^d \times \eta_{ch} \times \Delta t - \frac{Pdch_{k-1}^d}{\eta_{dch}} \times \Delta t \tag{4.4}$$

with $0.2 \times BAT^d \leq B_k^d \leq 0.8 \times BAT^d$ for any time slot k and DC^d ($0 \leq k < K$ and $1 \leq d \leq D$) — this last restriction models the Max Depth of Discharge property to increase the battery lifespan. The modeling of the batteries and PVs are based on [23].

4.2.2.4 Footprint model

In the current model, carbon emissions from operating the cloud platform originate from three sources: (i) consuming power from the regular electrical grid and manufacturing of both (ii) the photovoltaic panels, and (iii) the batteries. Equation (4.5) models the carbon footprint of the regular local electrical grid, defined by the carbon intensity of the power grid $gridCO2^d$ in the region of data center DC^d times the amount of energy used during the time slot k.

$$FPgrid_k^d = Pgrid_k^d \times \Delta t \times gridCO2^d \tag{4.5}$$

It is considered that the power from the local electricity grid may originate from multiple sources, and the $gridCO2^d$ is an input that represents its average carbon intensity during the year: the value will be low if it is supplied by solar, wind power, hydroelectric or nuclear power. On the other hand, the value will be higher if it is supplied by coal, oil, biomass, or natural gas.

For the carbon footprint of the PVs, to account for the different climate conditions at each region, in particular the solar irradiation received, one must also consider the expected power output that PVs can produce over their lifetime, as using a single year may cause over or under-sizing of the PV infrastructure. Therefore, the carbon footprint of PVs is also related to the location of each data center. Equation (4.6) models the carbon footprint for PVs:

$$pvCO2^{d} = \frac{FPpv_{1m2}}{expectedEpv^{d}} \tag{4.6}$$

where $FPpv_{1m2}$ is the emissions of manufacturing $1\,m^2$ PV in $g\,\mathrm{CO}_2 - eq$, and $expectedEpv^{d}$ is the expected energy production in Wh that $1\,m^2$ of the PV during its lifetime at the location of DC^{d}. As a result, the unit of this metric is expressed in $g\,\mathrm{CO}_2 - eq.Wh^{-1}$, and so, the total emissions from the PVs are related to its power production, as shown in Equation (4.7).

$$FPpv_{k}^{d} = pvCO2^{d} \times Pre_{k}^{d} \times \Delta t \tag{4.7}$$

Regarding the batteries' carbon footprint $FPbat^{d}$ of the DC DC^{d}, it is computed based in their capacity BAT^{d} (kWh) and the carbon emissions of the manufacturing process $batCOS$ in $g\,\mathrm{CO}_2 - eq.kWh^{-1}$, as seen in Equation (4.8). To be consistent with the calculation of $FPpv_{k}^{d}$, $batCO2$ is the share of the carbon footprint of the battery type chosen for a capacity of $1\,kWh$ over the time horizon of $\mathcal{H}$, assuming a battery has a lifetime of 10 years. Thus, given that we are considering one year of cloud operation, $batCO2$ is the tenth of the total carbon footprint of the considered battery. It is considered that the battery carbon footprint is homogeneous for all the locations, as the climate conditions do not influence it.

$$FPbat^{d} = BAT^{d} \times batCO2 \tag{4.8}$$

These modifications regarding the lifetime of PVs and batteries were necessary because we are considering only one year of cloud operation. Suppose we use the total carbon emissions for manufacturing the PVs and batteries. In that case, the solver will find a solution where there is little to no PV or batteries, because using the regular electrical grid would be less carbon-intensive.

4.2.3 Objective function

Now that the reader has been introduced to the modeling of the problem, the objective function can be defined (see Equation (4.9)). It consists of minimizing the carbon footprint of the globally distributed cloud federation to reduce as much as possible the carbon emissions, which come from both the consumption of electricity from the power grid, as well as from the manufacturing of photovoltaic panels and batteries with k and d defined as follows: $0 \leq k < K$ and $1 \leq d \leq D$.

$$\text{minimize} \sum_{k=0}^{K-1} \sum_{d=1}^{D} (FPgrid_k^d + FPpv_k^d) + \sum_{d=1}^{D} FPbat^d \tag{4.9}$$

4.3 Optimal resolution

In this section, we show that linear expressions can be used to express the constraints governing the operation of the globally distributed cloud platform. This is a mandatory step, since all the models presented in the previous section are linear equations. New real variables and constraints will be introduced to finish building the linear program that needs to be solved to achieve the targeted objective (see Equation (4.9)). The solution obtained after solving the linear program is optimal in nature and computed in polynomial time, as the model uses only real variables. Polynomial time is mandatory if we consider the number of variables needed for a time horizon $\mathcal{H}$ as long as one year (with time slots of 1 hour). We assume to choose

real positive values for all variables even if variables denote discrete objects like cores. Indeed, w_k^d is the number of cores needed to run tasks on DC^d during time slot k. Considering the size of modern cloud data centers with thousands of servers and millions of cores, the decimal part of each w_k^d can be neglected. Having the solution with more or less than a core on a given DC keeps the order of magnitude for the PV and battery sizing process the same. Finally, some constraints are explicit, and others are implicit to avoid the addition of integer variables, which would transform this LP into a MILP whose solving process would not scale at all.

4.3.1 Constraints to address the workload

The workload to be executed must respect each data center's computational capabilities C^d (number of available CPU cores). Equation (4.10) expresses that the number of cores that will be allocated to execute the tasks of the workload (w_k^d) does not exceed the existing number of cores of DC^d:

$$w_k^d \leq C^d \tag{4.10}$$

4.3.2 Constraints to reach the power demand

The electric part of each DC has to supply the DC power demand using local renewable energy from its PV panels (Pre_k^d), from batteries ($Pdch_k^d$ and Pch_k^d) and/or from the classical electrical grid ($Pgrid_k^d$). Equation (4.11) presents the restriction for the power consumption.

$$P_k^d \leq Pre_k^d + Pgrid_k^d + Pdch_k^d - Pch_k^d \tag{4.11}$$

4.3.3 Constraints on batteries

The batteries are defined by their capacity (BAT^d in kWh), and each data center can have a different size of batteries depending on how the intermittency of the renewable energies is managed on each site, for example, if the local electricity grid is high-carbon intensive, the battery will need a higher capacity in order to store solar power during the day to use it during night, otherwise if the local electricity grid has the presence of low-carbon intensive sources, the battery will have a smaller capacity. Equation (4.12) models the Max Depth of Discharge property, that is, the range that the battery needs to operate in order to increase its lifetime. In addition, the power to charge or discharge a battery is also limited by the level of energy remaining in the associated battery, so it is not possible to reach a forbidden energy level. Equations (4.13) and (4.14) express these constraints.

$$0.2 \times BAT^d \leq B_k^d \leq 0.8 \times BAT^d \tag{4.12}$$

$$Pch_k^d \times \Delta t \times \eta_{ch} \leq 0.8 \times BAT^d - B_{k-1}^d \tag{4.13}$$

$$Pdch_k^d \times \Delta t \ / \ \eta_{dch} \leq B_{k-1}^d - 0.2 \times BAT^d \tag{4.14}$$

The reader may notice the absence of restrictions for charging and discharging simultaneously in the modeling. Such restrictions would impose the usage of binary variables that would significantly increase the required computational time to find the optimal solution to the problem. We performed experiments with a shorter duration (around one month), and the sizing results were the same between both versions: using and not using binary variables. Furthermore, given a solution for the linear program, it is possible to calculate an alternative solution program where there would be no charge and discharge at the same time slot and with the same

sizing results — the differences between the solutions would be the values of the variables Pch_k^d and $Pdch_k^d$.

4.3.4 Linear program

The following linear program (LP) summarises what has been described in the current section concerning the model to solve the tackled problem. After solving the LP, the values of the variables are used to define both the renewable power supply part completely — variables BAT^d for the battery capacity and Apv^d for the area of the solar panels at each DC^d — and the core operating process of the distributed low carbon cloud federation with details of the way each DC is used time slot by time slot for one year on the considered weather conditions — variables w_k^d that represents the workload scheduled and Pch_k^d (and $Pdch_k^d$) for the power charged into (and discharged from) the batteries of each DC^d at each time slot k. Comprehensive experiments have been led to highlight the pertinence of the approach. These experiments are shown in the next section, and a discussion is proposed in Section 4.5.

$$\text{(LP)} \begin{cases} \text{minimize } \sum_{k=0}^{K-1} \sum_{d=1}^{D} (FPgrid_k^d + FPpv_k^d) + \sum_{d=1}^{D} FPbat^d \\ \\ \text{s.t.} \quad \text{(4.1) (4.3) (4.4) (4.5) (4.7) (4.8) (4.10) (4.11) (4.12) (4.13) (4.14)} \end{cases}$$

where all variables are positive real variables.

4.4 Experiments

In this section, we present the settings and the results of the experiments. More details for reproducing the experiments can be found in the public available software artifact [57].

4.4.1 Settings

4.4.1.1 Cloud infrastructure

The servers are homogeneous and based on equipment of real cloud infrastructure: the Taurus server of the Grid'5000 testbed[1]. The servers are equipped with two Intel Xeon E2630 CPUs, with a total of 12 CPU cores. For modeling the power consumption of the servers, real measurements conducted by [2] were considered: in the idle state, each server consumes 97 W, and their maximum power consumption (when using 100% of the 12 cores) is 220 W. The value of Pcore is 10.25 W, and it was obtained by linear interpolation between the power consumption of the idle and the fully used state.

Each data center is equipped with 23,200 servers (and a total of 278,400 cores). This number represents what can be seen in production data centers of major cloud players: Microsoft operates over 4 million servers distributed over 200 DCs [48]. The total power consumption from the servers of each data center when idle is 2.25 MW, and their maximum power consumption is 5.1 MW.

For the current experiments, we considered a network with a 48-ary fat-tree topology for the interconnection of the servers inside the DCs, that is, the network switches have 48 ports, and the total number of switches is 2,880. The power consumption of the switches was based on real measurements by Hlavacs et al. [27]: the HP

[1] https://www.grid5000.fr/w/Lyon:Hardware#taurus

ProCurve 2810-48G was selected, with 48 ports and approximately 52W per device. In total, the network equipment power consumption is 149.76 kW for each DC.

The locations of the data centers were based on the real cloud infrastructure of Microsoft Azure[2], and different regions in different continents, hemispheres, and time zones were selected. Figure 4.1 presents the details of the locations.

Figure 4.1 Selected locations for the data centers.

In order to represent the fact that each geographic region has different cooling needs, we used values for the PUE inspired by real data from Microsoft Azure for each site: for the Americas, the PUE is 1.17 (DCs São Paulo and Virginia), Asia Pacific has a PUE of 1.405 (DCs Pune, Canberra, Singapore, and Seoul), and for the Europe region, Middle East, Africa the PUE is 1.185 (DCs Johannesburg, Dubai, and Paris) [61].

[2]Azure global infrastructure: `https://infrastructuremap.microsoft.com/`.

4.4.1.2 Workload

The workload used was created using the Grog generator[3], a workload generator tool based on analysis of properties of the execution trace made available by Google in 2011 [11]. For reproducibility purposes, the parameters regarding the number of tasks were set to 350,000, and the duration was 30 days. The workload generator was run 12 times — 1 per month. Finally, the tasks have a duration of one hour.

4.4.1.3 Photovoltaic power production

The data for simulating the solar power production — Global Solar Horizontal Irradiation (in Wh/m^2) — was collected from the MERRA-2 project [16], since it provides information for anywhere on earth. Figure 4.2 illustrates the average solar irradiation of each location throughout the year 2021.

4.4.1.4 Carbon footprint

For PV panels, it is considered a lifetime of 30 years, and manufacturing 1 m^2 emits $250kg\,CO_2 - eq$, inspired from real measurements [64]. To compute the emissions in the form of $g\,CO_2 - eq.kWh^{-1}$ as stated in Section 4.2.2.4, we considered the total solar irradiation that was produced during the year 2021 multiplied by 30 (to account for the PV module lifetime of 30 years). For the electrical grid, we also considered the real-world data of the carbon footprint ($g\,CO_2 - eq.kWh^{-1}$). Table 4.3 lists the carbon emission values for each region.

Regarding the batteries, the emissions are only considered for the manufacturing step—$59kg\,CO_2 - eq$ per kWh and are the same for all the DCs locations. In our experiments, the considered lifetime of the batteries is ten years. Therefore, the input used is equal to $5.9kg\,CO_2 - eq$ per kWh, given that we simulated one year.

[3] `https://pypi.org/project/grog/`

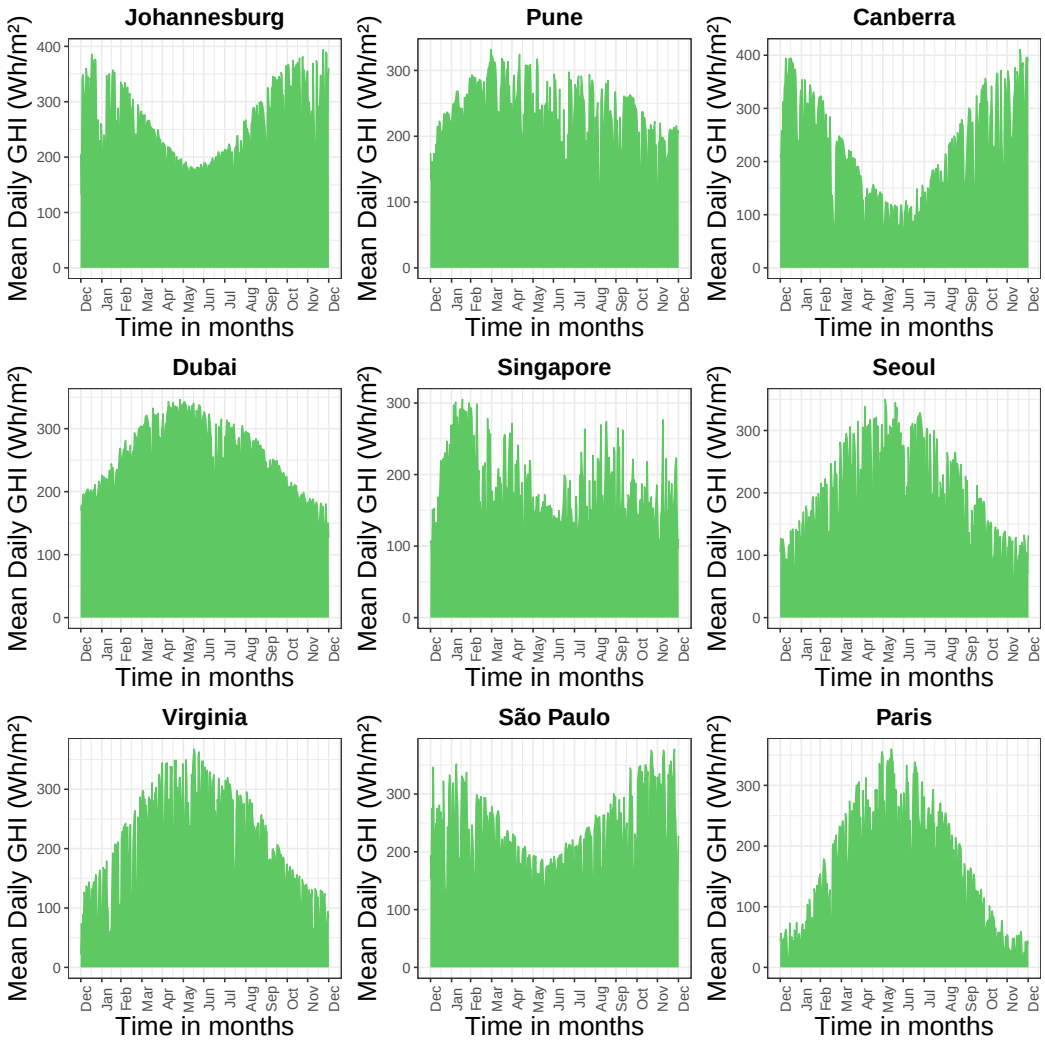

Figure 4.2 Average daily solar irradiation per location throughout the year 2021.

Table 4.3 Emissions (in $g\,CO_2 - eq.kWh^{-1}$) for both PV usage and using the regular grid. Source for grid emissions: electricityMap, climate-transparency.org.

Location	Grid	PV
Johannesburg	900.6	24.90
Pune	702.8	27.96
Canberra	667.0	29.71
Dubai	530.0	24.84
Singapore	495.0	36.19
Seoul	415.6	34.00
Virginia	342.8	31.71
São Paulo	61.7	27.99
Paris	52.6	39.93

4.4.1.5 Execution environment

The experiments were executed on a machine with the following configurations: Intel i9-11950H CPU, and 32 GB of RAM. The solver program used for solving the LP was the Gurobi Optimizer (version 9.5.2). The execution time for solving the LP with the inputs listed in the previous sections—which resulted in a total of 394,263 variables—was in the order of 30 seconds.

4.4.2 Results

In this section, we present the results in terms of the computed optimal area of the PVs and capacity of the batteries, the origin of energy used to supply the DCs operation (from the electrical grid, batteries, or PV panels), and the total emissions of the cloud operation, generated from both manufacturing PVs and batteries, and power consumption of the regular electrical grid. Furthermore, to assess the solution computed by the LP, we compare it with two other scenarios: i) DCs can only be supplied using power from the regular electrical grid — representing most of the DCs in operation, and ii) only power generated from the PV panels, and stored and discharged from the batteries are used to supply the DCs — represent green DCs that fully autonomous from the power grid. Finally, we present an evaluation using metrics to assess the environmental impact of the results.

Figure 4.3 illustrates the optimal area of the photovoltaic panels and the capacity of the batteries computed from the LP using the inputs described in Section 4.4.

To analyze the sources of energy that supplied the DCs operation, we present in Figure 4.4 the percentage that each source (grid, renewable, and batteries) was used to daily supply the DCs throughout the year. Figure 4.5 is a fine-grain visualization of the DC operation regarding the power consumed or produced for one day (January 1st, 2021): it illustrates hour-by-hour the DC total power demand, how much power

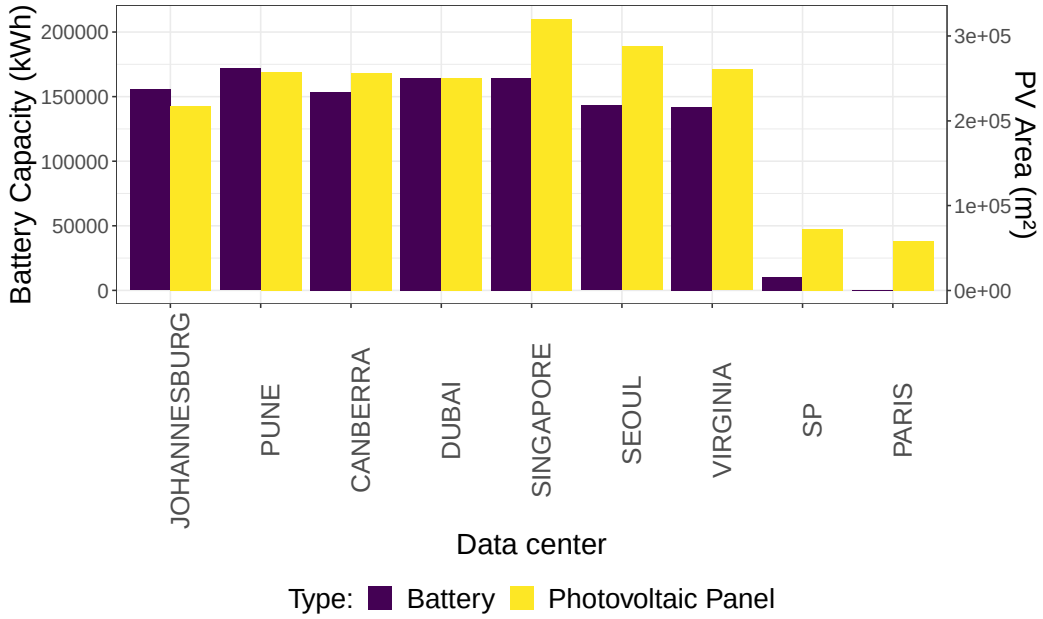

Figure 4.3 Optimal result for the area of PV panels and capacity of the batteries.

was consumed from the grid, discharged from the batteries, and produced by the PV panels.

In order to assess the optimal solution of the LP, we compared it with two other scenarios in terms of total carbon emissions ($t\,CO_2 - eq$): i) the DCs are only supplied by power from the regular electrical grid, and ii) the DCs are only supplied by renewable power from the photovoltaic panels and batteries. Table 4.4 presents the results. In comparison with the first scenario (only grid power), the reduction in the CO_2 emissions was approximately 85%, and it was approximately 30% for the second scenario (only renewable power).

Table 4.4 Total emissions for the different scenarios.

Scenarios	**Emissions ($t\,CO_2 - eq$)**
Electrical grid	201 211.3
PV and batteries	42 370.6
PV, batteries, and grid	29 600.6

To further evaluate these scenarios, we present in Table 4.5 results in terms of the average load each DC executed throughout the year. Equation (4.15) represents

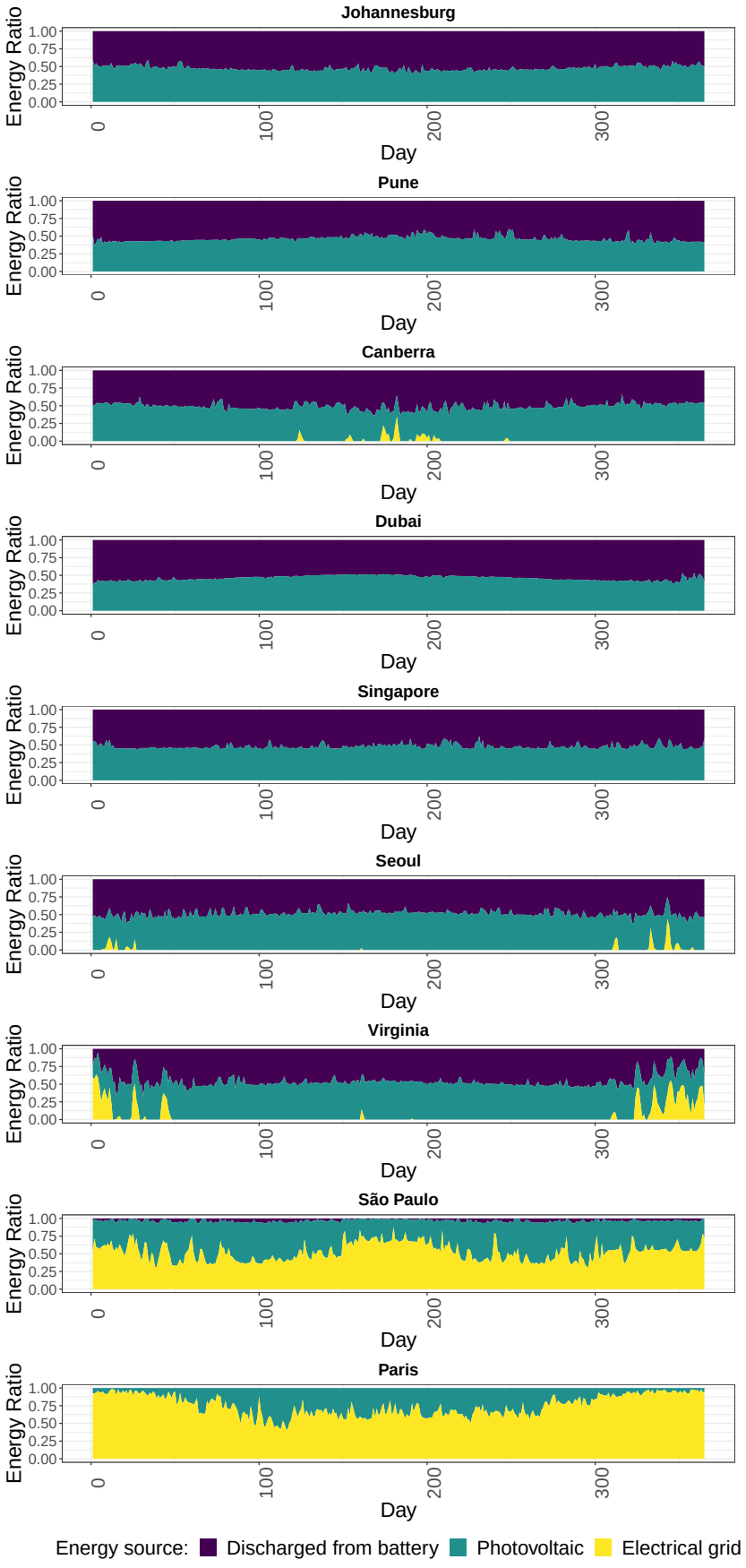

Figure 4.4 Composition of the DCs' daily energy consumption throughout the year considering the different sources of energy, where 1.0 is the DC's total energy consumption.

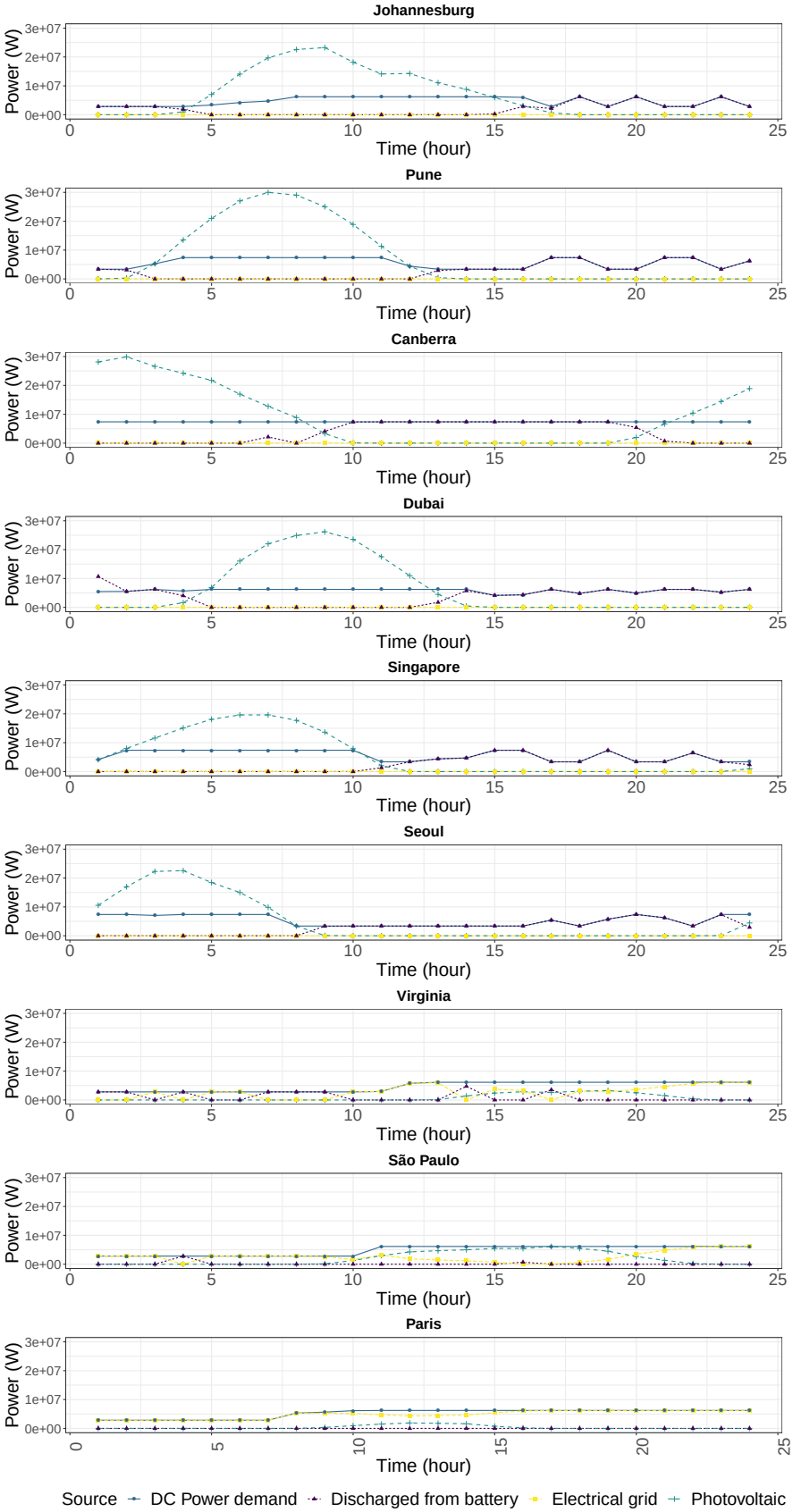

Figure 4.5 Composition of the DCs' hourly power consumption throughout the first day of the year. Time follows the Universal Time (UT) standard.

how the metric was computed for each DC d. One may notice that the Johannesburg DC received no load for the scenario where only grid power is used. This is justified by two reasons: i) there is no restriction to the amount of workload the data centers need to receive in our initial modeling; and ii) Johannesburg has the most carbon-intensive electrical grid, so the LP decided to use all the DCs in all the others locations to minimize the carbon footprint of the global cloud operation.

$$\frac{\sum_k w_k^d}{C^d \times K} \tag{4.15}$$

Table 4.5 Average DC load throughout the year

Location	Grid	PV + Bat	PV + Bat + Grid
Johannesburg	0	79.31	86.20
Pune	10.25	82.07	89.34
Canberra	99.72	66.62	67.95
Dubai	99.97	93.93	95.11
Singapore	99.93	72.6	85.18
Seoul	99.99	81.87	65.39
Virginia	100.0	88.54	75.51
São Paulo	100.0	63.67	59.06
Paris	100.0	81.24	86.11

To analyze the environmental impact of the solution, we used metrics extracted from [43]. The first metric, the Green Energy Coefficient (or GEC), is the ratio between the total renewable energy generated and the DC total energy consumption, and it can illustrate the oversizing of the green power supply infrastructure, that is, values above 1 indicate that more green energy is being produced than what the DCs are consuming. The second metric is the CO_2 savings, which represents the emissions reduction after DC equipment upgrade or flexibility mechanisms. CO_2 savings is computed as seen in Equation 4.16, where: $CO2_{current}$ represents the system studied after the modifications — the DCs with PV area and batteries capacity values computed from the LP — and $CO2_{baseline}$ the system in its original state. Here, it was considered that $CO2_{baseline}$ has the same workload allocation of $CO2_{current}$; the difference between the two is that $CO2_{baseline}$ does not have PVs

and batteries, and thus only consumes power from the local grid. Table 4.6 shows the computed values for both metrics.

$$CO2_{savings} = \left(1 - \frac{CO2_{current}}{CO2_{baseline}}\right) \times 100 \tag{4.16}$$

Table 4.6 Results of the sustainability metrics for the experiments

Location	GEC	CO_2 savings (%)
Johannesburg	1.47	93.93
Pune	1.45	91.5
Canberra	1.57	89.59
Dubai	1.59	89.1
Singapore	1.42	85.75
Seoul	1.53	82.51
Virginia	1.46	75.99
São Paulo	0.5	20.05
Paris	0.24	5.25

In order to assess the robustness of the sizing process for the area of PV panels and the capacity of the batteries, it is necessary to take into account the variability of meteorological conditions, given that the DCs will operate for decades and not only for one year. The metric selected is the Mean Absolute Percentage Error (MAPE) defined by: $\frac{1}{n}\sum_{i=1}^{n}\frac{|R_i - F_i|}{R_i}$, where n represents the number of values being considered, i the index of the value being considered, R_i the real value for the year, and F_i the estimated value (in this case, the computed sizing for the year 2021 that was used in the experiments). Table 4.7 presents the results of the MAPE for both the area of PV and capacity of the batteries when we solve the LP using as input the solar irradiation for the years 2018, 2019, and 2020. Results indicate a variation of less than 10% in the different DCs over the years.

Table 4.7 Evaluating sizing for different years using the MAPE metric (values are in %)

Location	PV Area	Battery Capacity
Johannesburg	1.72	1.64
Pune	3.72	0.76
Canberra	8.62	4.25
Dubai	2.31	2.88
Singapore	7.22	0.34
Seoul	3.15	1.11
Virginia	2.2	0.87
São Paulo	5.81	8.05
Paris	2.76	0

4.5 Analysis and Discussion

The results presented in the previous section permit the evaluation of the carbon footprint impact of different electricity supply policies for operating cloud data centers. On the one hand, as shown in Table 4.4, there is a significant reduction in carbon emissions — 5-fold decrease in our experiments — to obtain by including renewable energy in the electricity sources of DCs. Many Cloud providers have committed to using 100% renewable energy supplies for their DCs in the following years, and some already started this process by buying renewable energy generated elsewhere or installing renewable infrastructures. On the other hand, this objective of 100% renewable is, in our opinion, more ideological than pragmatic, and there is more benefit to obtain by combining grid and renewable electricity. We observe in our experiments a further reduction of a fourth in the optimal solution compared to the 100% renewable scenario. This study thus gives further insight into the debate of energy sources in modern clouds.

The data centers locations used allow us to benefit from the diversity of latitudes, hemispheres, and climates. As shown in Figure 4.1, the model includes 3 DCs in the southern hemisphere, 5 in the northern one, and Singapore almost on the equator. Considering the longitudes, two DCs are on the American continent, two on the African and European longitudes, while the four last ones are geographically

distributed on the Asian and Australian continents. This variety of longitudes and hemispheres permits mitigation of the impact of seasonal and daily variations of solar irradiation on electricity production and always has at least some DCs with good PV production, as shown in Figure 4.2. The diversity of climates is highlighted by the case of Singapore's solar production, which is the second lowest with Paris, while its location close to the equator could permit better irradiation.

As indicated in Table 4.3, we observe significant heterogeneity in the carbon footprint of grid electricity of the different DCs: Paris and São Paulo have the lowest electricity footprint, close to 50 g CO_2-eq/kWh, four DCs with a grid footprint between 300 and 600 g CO_2-eq/kWh (Virginia, Seoul, Singapore, and Dubai), Pune just over 700 g CO_2-eq/kWh and the most carbon-intensive grid electricity is in Johannesburg with more than 900 g CO_2-eq/kWh. This heterogeneity results in two categories for the optimal solution: i) Paris and São Paulo, DCs with a reduced number of PVs and batteries (no battery in Paris), and ii) the other locations have quite similar sizes of PV and batteries. For the first category, we see the carbon footprint of PVs and ESDs compared to the one of grid electricity: as the grid is low-carbon intensive for both locations, it can be used to supply the DCs. This justifies the low area for the PVs, and since there will be virtually no solar power overproduction, the battery capacity will be low as well. In the second category, the larger PV area is mainly associated with low solar irradiation. It might appear counterintuitive to allocate more PVs to locations with lower solar production, but this is more comprehensive considering the static part of the power consumption of DCs — from the idle consumption of servers and the interconnection network, as referred to in Equation (4.1). This static electricity consumption implies either using the carbon-intensive grid or sizing of PV and batteries that match the demand, even during winter days of low PV production. This results in a large PV and battery sizing — the PVs are producing up to 1.6 times the total DC energy consumption as seen in Table 4.6 — and, as shown in Figure 4.4, the grid energy consumption of these DCs is very low.

The detail of hourly electricity consumption and renewable production is highlighted in Figure 4.5. The workload is allocated in DCs with solar power production. If all this power production is used, or the corresponding DCs are full (in terms of available CPU cores), then the allocation is driven by the battery level of energy, and when none of these possibilities are available, the allocation is for the DC with the lowest grid electricity footprint. For example, in the last hours, the electricity consumption of the different DCs is supplied by power discharged from the batteries, and the remaining is allocated in the DCs of Paris, São Paulo, and Virginia. Thus, the DC of Virginia consumes grid electricity in two cases: either when Paris and São Paulo DC are full and cannot receive more work (from hours 10 to 24), or when the DC is empty and only local electricity can be used (hours 3, 5, and 6 in Figure 4.5). The follow-the-sun approach can be partially observed between hours 7 and 8, when Seoul PV power production decreases, and the workload is transferred to Paris, with grid consumption. Then, at hour 10, the same happens between PV production in Singapore and grid electricity in São Paulo, and at hour 11 between Pune and Virginia. The figure also shows the impact of location, season, and PV sizing on the solar production between Pune and Canberra, large PV production in the best hours — summer in the southern hemisphere, and the tiny production in Paris — winter in the northern hemisphere.

Table 4.5 presents the impact of the different scenarios for energy sources on the computational load of the different DCs using the DCU metric, that is, the average usage of the DCs total computational capacity in CPU cores being used during the year, which can also be used to infer how much workload is scheduled to the DCs. In the first scenario (electrical grid only), the single important information available for the allocation decision of the workload is the grid electricity footprint. Thus, one could expect the DCU values of the DCs to be sorted in the same order as the grid footprint order per kWh. However, the energy consumption does not only depend on the workload execution but also on the PUE of the different DCs — the power needed for cooling. We can thus observe a higher workload in Dubai compared to

Singapore, considering that Dubai has electricity with a slightly higher footprint but the lowest PUE. The second scenario considers a model supplied only by the local renewable infrastructure. The allocation is surprisingly distinct from the solar irradiation of the different DCs. For example, the DC of Paris has the lowest yearly irradiation but the median DCU in this scenario. Its DCU is higher than the one in Johannesburg, which has the second-highest yearly irradiation and is on a similar longitude. The workload allocation is thus not only driven by yearly irradiation. The extremely low PV production in Paris during winter, associated with the static part of the electricity consumption in each data center (network devices and idle consumption of the servers), implies a high sizing of PV and battery, which leads to a high solar power production during the other seasons that make it an interesting candidate for receiving the workload. On Johannesburg, the seasonal variation is lower, so the static power consumption does not influence the PV sizing. Another surprising result is that the DCs with the lowest DCU in this scenario are the 4 in the southern hemisphere (including Singapore). This contradicts the intuition of "follow-the-summer" allocation. The case of São Paulo and Canberra could be similar to the one of Johannesburg and Paris, considering the minimal daily production in Virginia and Seoul during winter. The largest DCU concerns DCs with the more stable production (Dubai and Pune) and the lowest minimum daily production (Virginia, Paris, and Seoul). Finally, for the last scenario — hybrid configuration with power from the electrical grid and DC's renewable infrastructure, the DCs with the largest DCU are the three that receive the largest solar irradiation during the year (Dubai, Pune, and Johannesburg), followed by Paris with the lowest grid electricity footprint. The two surprises are: i) the DCU of São Paulo, which is low considering its low grid electricity footprint and high solar irradiation; ii) and the DCU of Singapore, which is high considering its low PV production. Concerning São Paulo, this could be explained by the fact that it has the second-lowest grid footprint. This implies a low PV sizing (and a low battery sizing as a consequence), and finally, it mainly receives workload only when no more DC can provide electricity from PV or

battery discharge, and when the DC of Paris is full, which results in many constraints. Considering Singapore, it is probably due to its position close to the equator, which implies no "winter" season, and its large PV sizing. Finally, the reduction of the carbon footprint of each DC between the hybrid scenario (PV + bat + grid) and the scenario with only grid electricity is evaluated using the CO_2 savings metric, and the results are shown in Table 4.6. It is possible to observe a small decrease in Paris and São Paulo — since there are already low-carbon sources in the local electricity grid, and a large decrease in the other locations, correlated to the electricity footprint. This information could be used by the decision-maker to prioritize investments in the locations that would result in the highest reduction of the carbon footprint.

4.6 Summary

In this chapter, we studied the problem of greening the operation of a distributed cloud data center (DC) federation to lower its carbon footprint. The IT part of the cloud platform already exists — servers and network devices, and the idea is to add renewable infrastructure equipment on site to introduce low-carbon intensive energy in the DC power supply, as the local electrical grid may be high carbon-intensive. Given that the sun is shining everywhere on earth, we have proposed photovoltaic panels (PVs) to produce renewable energy and batteries as energy storage devices to mitigate the intrinsic intermittency of this energy during the day and for night computations. The question is how to size the PV array area (m^2) and associated battery capacity (kWh), given an existing federation of DCs distributed around the earth and not neglecting the fact that manufacturing PVs and batteries also presents an environmental impact. We have provided a formulation of the problem as a linear program. The particularity of our formulation is that the modeling uses only real variables given our objective function and the context of the problem. As a result, the linear program allows to optimally solve large problem sizes in polynomial time,

e.g., minimizing the carbon footprint of a nine-site federation, each with its own weather conditions, upon a one-year horizon, hour by hour. We have demonstrated that our program is able to calculate the optimal sizing for PVs and batteries in just a few minutes. Numerous experiments have brought forward results that we have analyzed and discussed to explain what these results express. As an example, an interesting result, depending on the DC locations considered, is that the optimal solution to reduce the carbon footprint is to maintain an energy mix through a hybrid configuration including both PVs and a classical grid where the production is low carbon instead of proposing an all renewable platform. Moreover, batteries are not always mandatory in each location (as in the case of Paris DC). Finally, our model has the flexibility to be extended to assess other scenarios (more DCs, other locations, values for carbon emissions, or workloads) and it may help decision-makers build their strategy to reduce the environmental impact of the cloud operation.

In future work, we plan to propose a sizing process that also includes the IT part, for example, the new generation of servers with more powerful and energy-efficient hardware, and also their associated carbon footprint from the manufacturing process. Since this investment has been made for years, another perspective is to introduce uncertainty into this sizing process to obtain a more robust distributed DC platform that can provide satisfying service to clients even if the weather conditions change and the submitted workload evolves. The goal always being to remain as virtuous as possible.

Long-term evaluation of sizing the DC operation

5

5.1 Introduction

In the previous chapter, we presented an initial model to reduce the carbon emissions of operating a cloud federation for the short term (one year) by sizing its renewable infrastructure, that is, defining the area of solar panels and the capacity of the batteries for each data center. Given that the cloud federation will operate for the long term, the decision-maker needs to be sure of the investments that will be made to avoid wasting money and generating environmental impact with a bad strategy. Furthermore, a planning process is subject to many uncertainties that, if not taken into account, will decrease the validity and credibility of the solution.

This chapter presents an extension of the model focusing on the long-term operation of the cloud DCs and the uncertainties that this sizing process is subject to. In our context, the uncertainty comes mainly from the intermittency of renewable sources. Therefore, the reader will find in the next sections an analysis of the sensitivity of the modeling to the climate conditions. Another input that may affect the sizing process is the value of the local electricity grid emissions. In our initial modeling, we used the average value of the year, given that not all locations provide this value with shorter intervals — hour by hour variation, for example. The reader will also find an analysis of the impact of using the fine-grain information of the grid emissions in comparison to the year's average on the sizing process and the resulting total carbon footprint.

Given that the focus is now on long-term operation, it is essential to consider the entire life cycle of the renewable infrastructure. In the previous chapter, we made the modeling considering only the environmental impact of manufacturing solar panels and lithium-ion batteries. This decision was made because it was the available data that we found at the time of studying the problem. However, the environmental impact is also present in the other phases of the lifetime of these devices — operation, discarding, and recycling. Therefore, the modeling will be updated to account for the emissions of the life cycle of the renewable infrastructure.

The reduction of carbon emissions by using solar panels is limited by the fact that they only generate power when the sun is shining, therefore they need to supply the operations of the cloud during the day and charge the batteries for night operations. In this chapter, we will evaluate if using wind turbines could further reduce the carbon emissions of the cloud operation, and the sizing of both the PV panels and batteries — considering that the wind may also be available at night.

One must also take into account that once built, the renewable infrastructure cannot be reduced — this would imply destroying/discarding PV panels, batteries, and wind turbines. Therefore, another strategy is necessary to deal with the oversizing that the intermittency of renewable sources might cause. Thankfully, another part can be managed to reduce the impact of bad-sizing: scheduling of the workload. In the cloud platform, a significant part of the workload are tasks that do not have a high priority and whose execution can be delayed over time, the so-called batch tasks: at Meta, 7.5% of the workload are offline tasks [1] and low priority tasks consume in average 20% of the computational resources (CPU and memory) of Google's clusters [56]. We will show an analysis of the viability of delaying α % of the tasks up to β time slots (each time slot has 1 hour of duration) to reduce the carbon footprint.

Another important aspect to consider is that the demand for cloud computing resources grows year by year — to deal with the increasing number of users and requests of applications, and new server generations are launched that have more

computationally powerful hardware that may be more energy efficient. As shown by Masanet et al. [34], from 2010 to 2018 the DC workload increased 6 times, however, the increase in energy consumption was only 6% thanks to efficiency improvements in both hardware and software. On the other hand, manufacturing these serves also emits carbon, which cannot be neglected. Furthermore, given the increasing integration of renewable infrastructures in the cloud data centers, most of the carbon emissions are shifting from the power consumption of the DC operation to the manufacturing of IT equipment [20].

Until now, we only explored the environmental impact aspect — in terms of carbon emissions — of the cloud federation operation. Another dimension of high importance for the decision maker is the monetary costs, since the business will only survive if it generates profit. The reader is presented to an analysis of how expensive it is to reduce the carbon footprint of the multi-cloud operation by using a local renewable infrastructure. Similar to the environmental impact analysis, we also consider the costs of the whole life of the solar panels, wind turbines and batteries

More specifically, the chapter presents the following contributions:

- we extend the modeling to account for all the life cycle emissions of the renewable infrastructure — from manufacturing to discarding/recycling
- we present an evaluation of how much the carbon footprint could be reduced if wind turbines are also included in the renewable infrastructure
- we will show an evaluation regarding the sensitivity of the LP to the inputs: i) uncertainties caused by the intermittency of solar irradiation, and wind speed, and what needs to be considered in the modeling to avoid over or under-sizing of the renewable infrastructure; ii) some locations have data sources with grid carbon footprint at time intervals of one hour, and we will show an analysis if this fine-grain value would affect the renewable infrastructure sizing

- we will how much we can reduce the carbon footprint using the flexibility of the scheduling — delaying batch tasks
- we will show an analysis of how expensive (in dollars) is to reduce the cloud federation footprint and the gains in both monetary and emissions savings
- we will present an analysis of when is viable to add new servers (that may replace servers from older generations) in terms of carbon footprint

The work of this chapter is a direct extension of the work of the Chapter 4, and it was also done in collaboration with the following members of the DATAZERO[39] research team: Georges da Costa, Jean-Marc Nicod and Veronika Rehn-Sonigo. Partial results were presented in the form of a short presentation: ***Vasconcelos, M.**, Cordeiro, D., Costa, G. D., Dufossé, F., Nicod, J.-M., and Rehn-Sonigo, V. (2023). Long-term evaluation for sizing low-carbon cloud data centers. In ComPAS 2023 Conférence francophone en informatique, Jul 2023, Annecy, France*. The results will be published in a scientifical journal — the publicaton was at the writing phase at the time this thesis was written.

The remainder of the chapter is organized as follows. Considering that we are extending the model of the previous chapter to evaluate different scenarios, each scenario will have its own Section — Sections 5.2, 5.3, 5.4, 5.5, 5.6, and 5.7, and in each section, the reader will find the necessary modifications in the modeling, the experiments performed and the results of the experiments. After presenting the evaluated scenarios, we will show the discussion at Section 5.8. Finally, Section 5.9 concludes the chapter.

5.2 Life-cycle of the renewable infrastructure

In this section we will present how we can model the carbon footprint from the whole life cycle — from manufacturing to recycling/discarding — of the renewable

infrastructure, a mandatory step to make the model more realistic, in particular when focusing on the long-term operation of the cloud federation.

5.2.1 Updates in the model

To model the carbon footprint taking into account the whole life cycle of the renewable infrastructure, we will use the emissions of the energy produced (or delivered in the case of batteries) in $g\text{CO}_2 - eq.Wh^{-1}$. Equation (5.1) models the new footprint for the PV panels, and the main difference is the $pvCO2LC$ input that represents the life-cycle emissions.

$$FPpv_k^d = pvCO2LC \times Pre_k^d \times \Delta t \tag{5.1}$$

For the batteries, the initial modeling considered the emissions of manufacturing related to its capacity. Now to account for all the life-cycle, the emissions relate to the energy delivered (discharged) by the batteries. Equation (5.2) models the new footprint for batteries, where $batCO2LC$ represents the life-cycle emissions ($g\text{CO}_2 - eq.Wh^{-1}$).

$$FPbat_k^d = Pdch_k^d \times batCO2LC \times \delta \tag{5.2}$$

Considering that now the information used for computing the batteries' emissions is the power discharged, we need to add a restriction for the batteries being used only for night computations, otherwise the solver may compute a solution where the batteries will be oversized to store all the excess of renewable energy during summer to supply during winter, resulting in GWh of capacity — which is a problem considering the other environmental impact of batteries as their recycle rates needs yet to improve [5]. Equation (5.3) models this restriction, where $SUNRISE^d$ is the

set with all the instants of times where the sun starts shining throughout the year in each location considering its time zone.

$$\forall t \in SUNRISE^d : B_t^d = 0.2 \times Bat^d \tag{5.3}$$

Finally, the objective function also needs to be modified in order to include the new life cycle emissions of the batteries. Equation 5.4 models the new objective function.

$$\text{minimize} \sum_{k=0}^{K-1} \sum_{d=1}^{D} (FPgrid_k^d + FPpv_k^d + FPbat_k^d) \tag{5.4}$$

5.2.2 Experiments

To evaluate the impact of taking into account the emissions from the whole life cycle in terms of carbon footprint and dimensions of the renewable infrastructure, we will perform the sizing and use the previous model — which considered only the emissions from manufacturing — as baseline.

5.2.2.1 Settings

The cloud infrastructure, the workload, grid emissions, and the execution environment are the same as in Section 4.4.1. For the photovoltaic power production, we will use the average of the irradiation from the years 1980 to 2019, and Figure 5.1 illustrates the values for the different DCs locations. This modification was to consider the variations between the years, as using a single year is not enough: it may be the case of a year with the lowest or highest solar irradiation. More details for choosing the average irradiation value will be given in the next section.

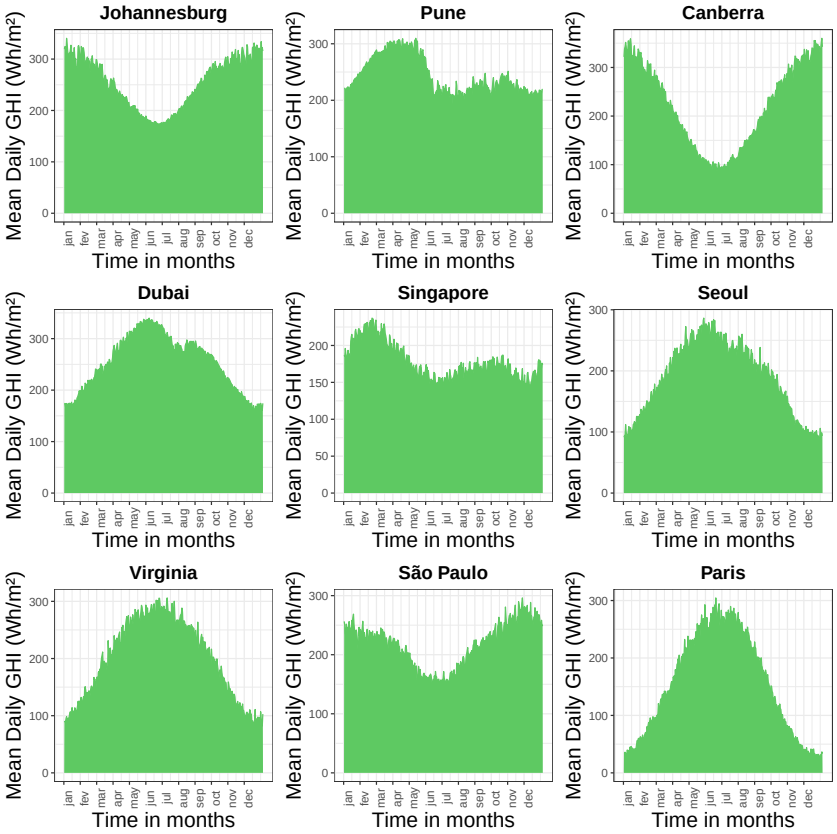

Figure 5.1 Average solar irradiation from 1980 to 2019 per location.

For the life-cycle CO_2 emissions of the batteries and PVs, we used values from the National Laboratory of Renewable Energy (NREL) of the United States that performed an analysis with over 3000 published studies on the life-cycle of renewable infrastructures [45]: $43g\,CO_2 - eq.Wh^{-1}$ for the PVs and $33g\,CO_2 - eq.Wh^{-1}$ for the Lithium-Ion batteries.

5.2.2.2 Results

Table 5.1 presents the CO_2 emissions considering the whole life cycle compared to considering only the one from the manufacturing. As expected, the total CO_2 emissions are higher (around 8% of increase) since now the whole life cycle of the renewable infrastructure is taken into account.

Table 5.1 Total emissions for the different scenarios.

Scenarios	**Emissions ($t\,CO_2 - eq$)**
CO_2 from manufacturing	34 559.03
CO_2 from the whole life cycle	37 828.49

Finally, to better assess the impact of considering the emissions from the whole life cycle in the sizing process, we compare its results with the sizing process that only considers the emissions from the manufacturing phase — Figures 5.2 and 5.3.

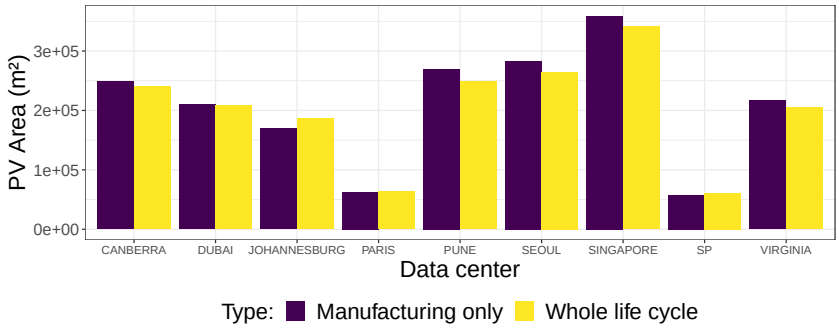

Figure 5.2 PV Area sizing when considering carbon footprint of manufacturing vs whole life cycle.

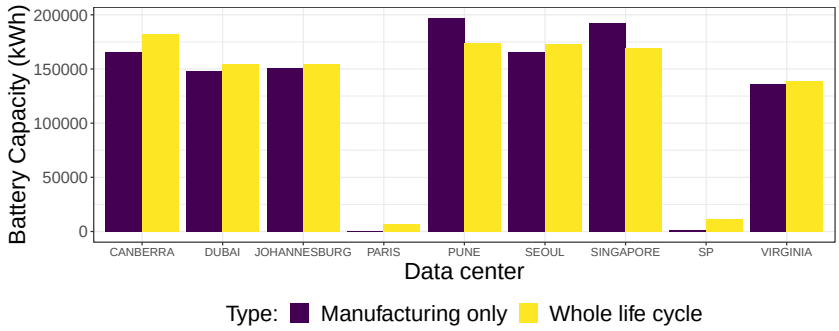

Figure 5.3 Battery capacity sizing when considering carbon footprint of manufacturing vs whole life cycle.

5.3 Including wind turbines to the renewable infrastructure

In this section, we will evaluate the impact of adding another renewable power source, wind turbines (WT), and to what extent they complement the photovoltaic

power production to reduce the total emissions of the cloud federation operation, in particular during the night and seasons with lower solar irradiation as the winter.

5.3.1 Updates in the model

We consider a new variable WT^d representing how many wind turbines will be built at each DC location. The number of wind turbines (WT) is an integer number (we cannot produce 1.2 WT for example). However, to avoid increasing the computational complexity of the LP by including integer variables, we made a relaxation in our modeling to allow the WT variable to be linear.

The power production of the WT^d wind turbines at location of data center d at instant k (Pwt_k^d) is modeled by Equation 5.5, where V_k^d is the wind speed at location of data center d at instant k; $Vici$ is the *cut in* wind speed, that is, the wind turbines start producing power when the wind speed is greater or equal to $Vici$; Vco is the *cut out* wind speed, that is, the wind turbines stop producing power when the wind speed is greater than Vco; Pr is the WT rated power production in W; and Vr is the speed where the WT starts producing Pr power. This model is based on [24].

$$Pwt_k^d = WT^d \times \begin{cases} 0 & \text{if } V_k^d \leq Vici \\ Pr \times \frac{V_k^d - Vici}{Vr - Vici} & \text{if } Vici < V_k^d \leq Vr \\ Pr & \text{if } Vr < V_k^d \leq Vco \\ 0 & \text{if } Vco < V_k^d \end{cases} \tag{5.5}$$

Given that we are using an additional renewable power infrastructure, it is also necessary to update the model of the renewable power production on DC d at instant

k. Equation (5.6) models that renewable power can originate from both the PVs and WT.

$$Pre_k^d = Ppv_k^d + Pwt_k^d \tag{5.6}$$

To model the carbon footprint of the wind turbines, their whole life cycle also needs to be taken into account. Equation (5.7) models the footprint of using the WT at the location of data center d at instant k, where $wtCO2LC$ is the input that represents its emissions for the life cycle (in $g\,\mathrm{CO}_2 - eq.Wh^{-1}$).

$$FPwt_k^d = wtCO2LC^d \times Pwt_k^d \times \Delta t \tag{5.7}$$

Finally, the objective function also needs to be modified in order to include the carbon emissions of using WTs. Equation 5.8 models the new objective function.

$$\text{minimize} \sum_{k=0}^{K-1} \sum_{d=1}^{D} (FPgrid_k^d + FPpv_k^d + FPwt_k^d + FPbat_k^d) \tag{5.8}$$

5.3.2 Experiments

In order to analyse the impact of including wind turbines in the renewable infrastructure in terms of carbon footprint and dimensions of the solar panels and batteries, we will perform the sizing and use the previous model — which considered only the PV and batteries — as baseline.

5.3.2.1 Settings

The cloud infrastructure, the workload, grid emissions, and the execution environment are the same as in Section 4.4.1. The solar irradiation values, and emissions

from PV panels and batteries are the same as Section 5.2.2. For the CO_2 emissions of the wind turbines, it is considered that it emits $13g\,CO_2 - eq.Wh^{-1}$ taking into account its life cycle. The source for the values was the same as in Section 5.2.2, the analysis from NREL that evaluated more than 3000 studies.

For the wind speed data, we used values from the ERA5 data source [25] — average of the years 1980 to 2019. The data source contains values for the 100m v-component of wind (air horizontal speed towards the east) and 100m u-component of wind (air horizontal speed towards the north) — both values are at 100 m of height. To transform these values into wind speed, we need to execute the following computation using Pitagora's theorem: $w_s = \sqrt{v^2 + u^2}$, where w_s is the wind speed, u the value of the u-component and v the value of the v-component.

5.3.2.2 Results

To evaluate the impact of including WT, we compare the sizing for using only PVs and batteries and using PV, batteries and WT. Table 5.2 presents the results in terms of total emissions. When also considering the wind as a renewable power source, we observe a reduction of approximately 34% in comparison with the scenario where no WT is used and green energy is only produced by the PVs.

Table 5.2 Total emissions ($t\,CO_2 - eq$) for different scenario.

Scenario	**Total emissions ($t\,CO_2 - eq$)**
PV + Bat + WT + Grid	24 977.89
PV + Bat + Grid	37 828.49

Table 5.3 presents the number of WTs manufactured at each data center location, as said in Section 5.3.1, we use a linear number for the wind turbines to avoid increasing the computational complexity of the problem. The decision-maker could decide whether to round the value up or down based on its budget constraints.

Table 5.3 Computed number of WT for each location.

Location	Number of WT
Johannesburg	58.11
Pune	25.77
Canberra	66.56
Dubai	79.09
Singapore	36.93
Seoul	109.56
Virginia	39.16
São Paulo	86.53
Paris	22.15

The Capacity Factor metric can be used to better analyse the potential of a location to generate renewable power. This metrics measures a ratio of the real energy produced by the device in comparison to its maximum energy output [33]. In our experiments, we use the rated power of 150 W for 1 square meter solar panel, and we used climate conditions from the years 1980 to 2019 to estimate the Capacity Factor. Table 5.4 presents the results for both the PV and wind turbines

Table 5.4 Capacity Factor (in %) for solar panels and wind turbines at each location.

Location	PV	WT
Johannesburg	25.55	12.96
Pune	24.26	10.04
Canberra	22.08	12.97
Dubai	25.28	13.98
Singapore	17.68	8.58
Seoul	18.81	9.41
Virginia	19.83	14.68
São Paulo	21.74	10.06
Paris	15.37	23.51

Finally, to assess the impact of including the WT regarding the sizing of both PV and batteries, we present in Figure 5.4 and Figure 5.5 the dimensioning for both equipment considering and not considering manufacturing WT. It is possible to observe that the inclusion of wind turbines has a significative reduction in the total solar panels area, and for some locations — such as São Paulo and Paris — the optimal is to use only wind turbines. The batteries are also reduced, given that the

wind can blow in the night as well, so no extra energy needs to be charged in the batteries from the solar panels to be used for night computations.

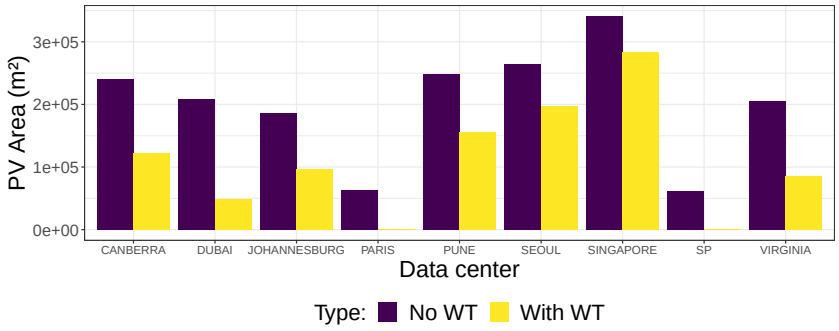

Figure 5.4 PV Area sizing when the WT are and are not included.

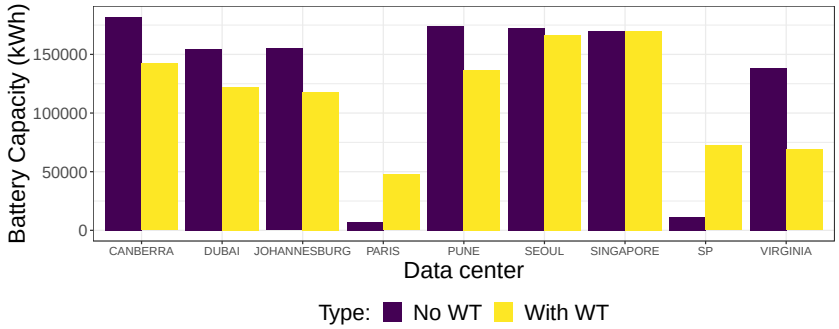

Figure 5.5 Battery capacity sizing when the WT are and are not included.

5.4 Sensibility of the LP to uncertainties

To assess the sensibility of the linear program to the inputs, we will evaluate it considering different scenarios in terms of solar irradiation, wind speed, and grid emissions data. The renewable sources are intermittent by nature, and this variation can impact the sizing process, resulting in an over or under-sizing of the renewable infrastructure. To illustrate the intermittency of both wind speed and solar irradiation, we present in Figure 5.6 boxplots of the total energy one solar panel produced per year during the years 1980 to 2019, and in Figure 5.7 for a single wind turbine. Those figures illustrate the intermittency of renewable power production over the

years, and one may also observe the potential for using one source or another: for example, Paris has the lowest solar power production considering one solar panel, and on the other hand, it has the highest wind power production.

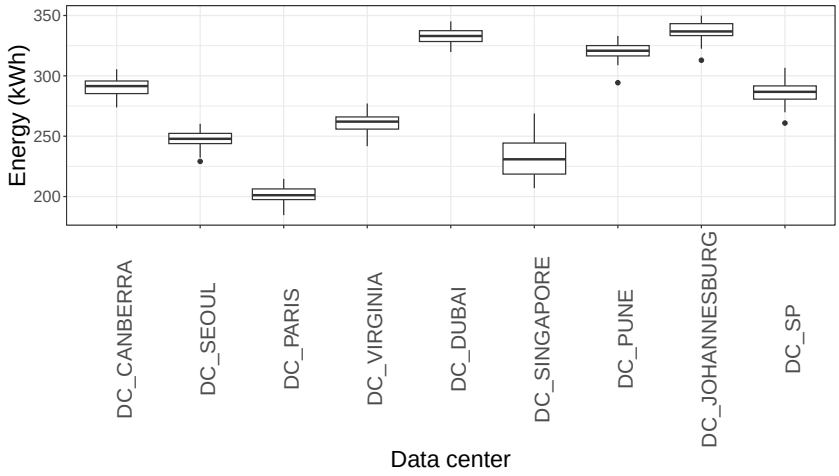

Figure 5.6 Different energy production for 1 PV panel at each location for the years 1980 to 2019.

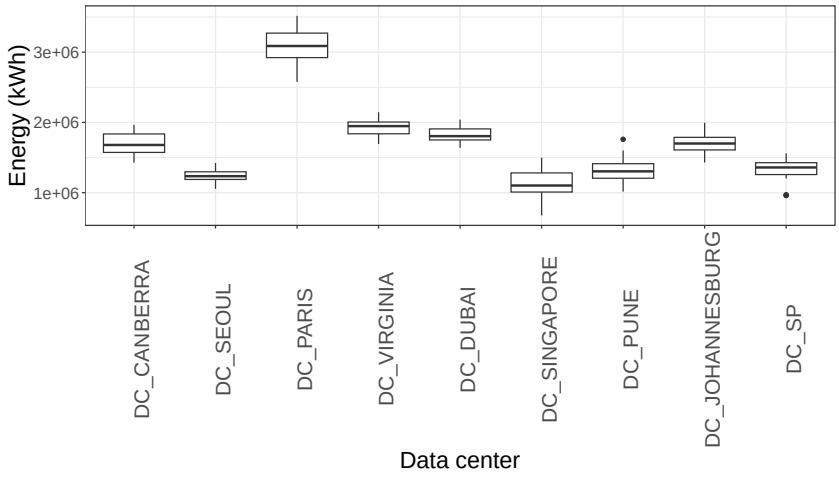

Figure 5.7 Different energy production for 1 wind turbine at each location for the years 1980 to 2019.

Given the presence of low-carbon intensive sources in some countries, the emissions of the local electricity grid are not static all over the year. For example, regions with access to solar energy have a lower grid footprint during the day. Some countries provide access to fine-grain information about their electricity mix in real-time

or historical data in the form of time series. However, obtaining data for all the locations is not an easy process: not all the locations provide this information, or they provide different time granularity (day, hour, month), historical data with different durations (for example only the last six months), or only real-time values. For modeling the variation in grid emissions, it is necessary to update the model, and in the next section, we present the modifications.

5.4.1 Updates in the model

5.4.1.1 Intermittency of renewable sources

Regarding the intermittency of solar irradiation and wind speed, there are no necessary modifications to be made in the modeling itself, as the impact is generated by which input data will be used for the sizing process.

5.4.1.2 Fine-grain values of grid emission data (hour by hour)

To model the change in the electricity grid emissions over time, we need to modify the input $gridCO2^d$ to be a time series. For the locations where data is not available hour by hour, a pre-processing of the data is needed. For example, if the region only provides daily data, one can assume that for the 24 hours of the day the value of grid emissions will be the same. Equation (5.9) presents the modification where $gridCO2_k^d$ is the new input for the grid emissions with one value for each time slot k at the location of data center d.

$$FPgrid_k^d = Pgrid_k^d \times \Delta t \times gridCO2_k^d \tag{5.9}$$

5.4.2 Experiments

To assess the sensitivity of the model to the inputs of climate conditions and grid emissions, we will conduct a series of experiments. For the climate conditions, we will evaluate if using simple forecasts provide precise results and what is the impact in carbon footprint considering the future operation — 10 years, and the baseline will be the optimal scenario where the climate conditions are know in advance. For the grid emissions, we will evaluate the impact in carbon footprint of using the fine-grain information — hour by hour — and the average value of the year.

5.4.2.1 Settings

For both scenarios, the cloud infrastructure, the workload, and the execution environment are the same as in Section 4.4.1.

5.4.2.1.1 Intermittency of renewable sources To evaluate the sensibility of the sizing process considering the uncertainty of the renewable sources production, we will compare — in terms of CO_2 emissions — the operation of the cloud federation using the optimal renewable infrastructure (where all the climate conditions are known in advance) with a renewable infrastructure dimensioned using simple forecasting techniques (average and median from the past). The scenarios are described as follows. First, the average and the median values of solar irradiation and wind speed per time slot were collected from 1980 to 2009. Second, the sizing process was executed for these two data sets. Then, the cloud was operated for ten years (2010 to 2019) using the dimensions computed in the previous step, but with the climate condition of each respective year. Finally, the results in terms of CO_2 emissions were compared to the optimal renewable infrastructure for the period of 2010 to 2019.

We will also evaluate the adoption of PV only and WT only in the sizing, to assess which has higher impact on the sizing process. The evaluation will follow the same steps described in the previous paragraph.

The data for the solar irradiation was collected from the MERRA-2 project [16], and for the wind speed from ERA5 data-source [25].

5.4.2.1.2 Fine-grain values of grid emission data (hour by hour) In order to evaluate what is the impact of using a fixed value for the grid emissions — an average of the year — or a fine grain value – hour by hour time series — we will perform the sizing with these two scenarios and compare them in terms of CO_2 emissions and dimensions of the renewable infrastructure for the years 2018, 2019, 2020 and 2021.

For the scenario of hour-by-hour grid emissions, we used data sources provided by Electricity Map. For the locations: Pune, Canberra, Seoul, Virginia, São Paulo, and Paris, the data source provides CO_2 intensity of the grid hour by hour for the duration of 1 year. For Johannesburg and Singapore regions, the information of the CO_2 emissions was in the granularity of months. Therefore, we generated a time series with a total duration of one year with the same value of CO_2 per hour for each corresponding month. For Dubai, we only found data for the year average grid CO_2 emissions, and we generated a time series in which the grid emissions is fixed for every hour of the year.

Table 5.5 presents the sources of the data, the time span, and the granularity for the grid emissions of each data center location, and Table 5.6 presents the average CO_2 emissions of the year for these data sources — the value used is the average of the time series of each location for the evaluated years.

To illustrate the variation of the grid emission over time, we will present two scenarios to the reader: one with a significative presence of renewables in the local

Table 5.5 Comparison of local electricity grid information.

Location	Granularity	Span	Source
Johannesburg	month	year	ElectricityMaps
Pune	hour	year	ElectricityMaps
Canberra	hour	year	ElectricityMaps
Dubai	year	year	https://1p5ndc-pathways.climateanalytics.org/
Singapore	month	year	ElectricityMaps
Seoul	hour	year	ElectricityMaps
Virginia	hour	year	ElectricityMaps
São Paulo	hour	year	ElectricityMaps
Paris	hour	year	ElectricityMaps

Table 5.6 Average emissions (in $g\text{CO}_2 - eq.kWh^{-1}$) from using the regular grid at the different years.

Location	2021	2020	2019	2018
Johannesburg	700.66	700.66	700.66	700.66
Pune	728.15	724.04	726.43	723.83
Canberra	655.36	692.23	712.43	728.21
Dubai	530.00	530.00	530.00	530.00
Singapore	491.01	491.01	491.01	491.01
Seoul	490.60	490.15	490.73	490.90
Virginia	435.25	415.14	447.98	453.40
São Paulo	172.54	103.47	108.95	105.21
Paris	63.48	62.99	61.62	60.00

electricity grid (São Paulo) and another with a low presence (Seoul). It will be possible to observe the variance between day and night, seasons, and years.

Figure 5.8 illustrates the local electricity grid emissions over time for São Paulo, and each subplot represents one bimester. The differences between day and night might be justified by using solar energy, as well as the difference between seasons. However, it is also possible to observe a difference among the years — 2021 is the year with the highest carbon footprint of the grid. The higher emissions of the year 2021 are justified by the hydric crisis that occurred during that period — the greatest of the previous 91 years, since a large share of São Paulo's electricity demand is supplied by hydroelectric power, and coal power plants were used to meet the power demand [31].

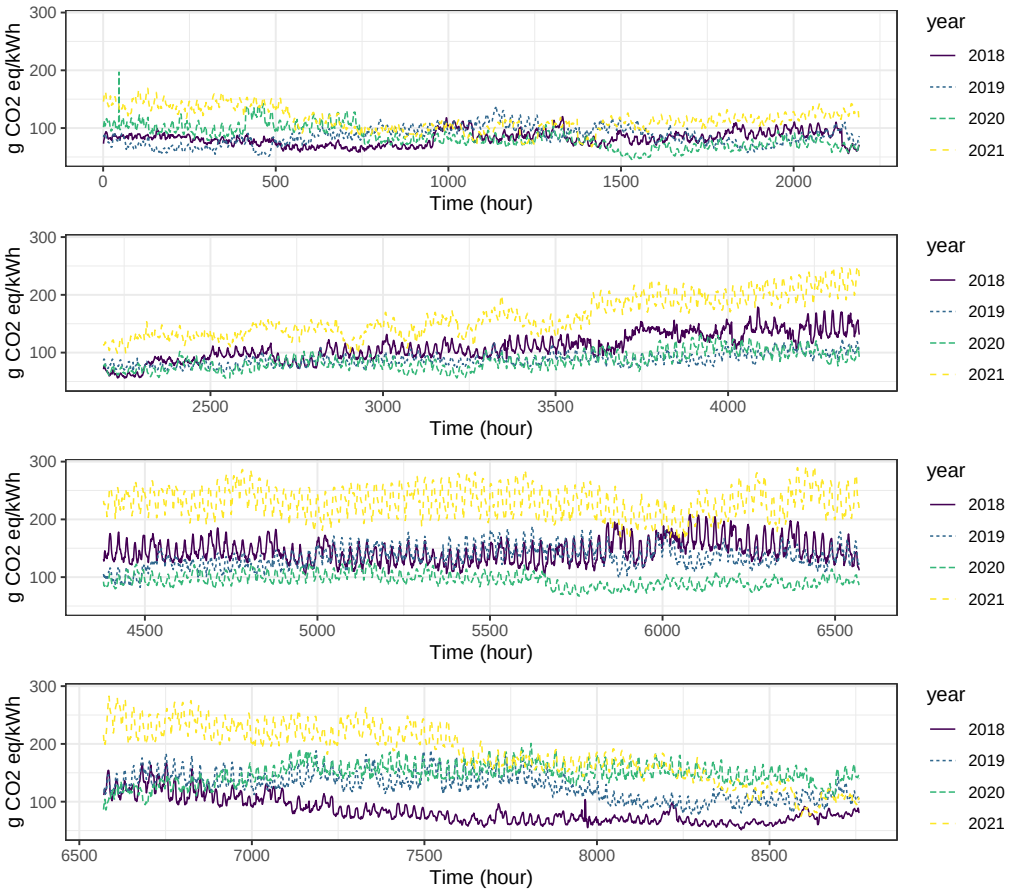

Figure 5.8 São Paulo's hour by hour grid emissions for the years 2018, 2019, 2020 and 2021.

The scenario of Seoul — illustrated in Figure 5.9 — represents a location with a low presence of renewables: most of the electricity demand is supplied by carbon-intensive sources such as coal and gas. The usage of PV panels can justify the small variation between the hours, as when the sun starts shining, they produce power and the emissions are reduced.

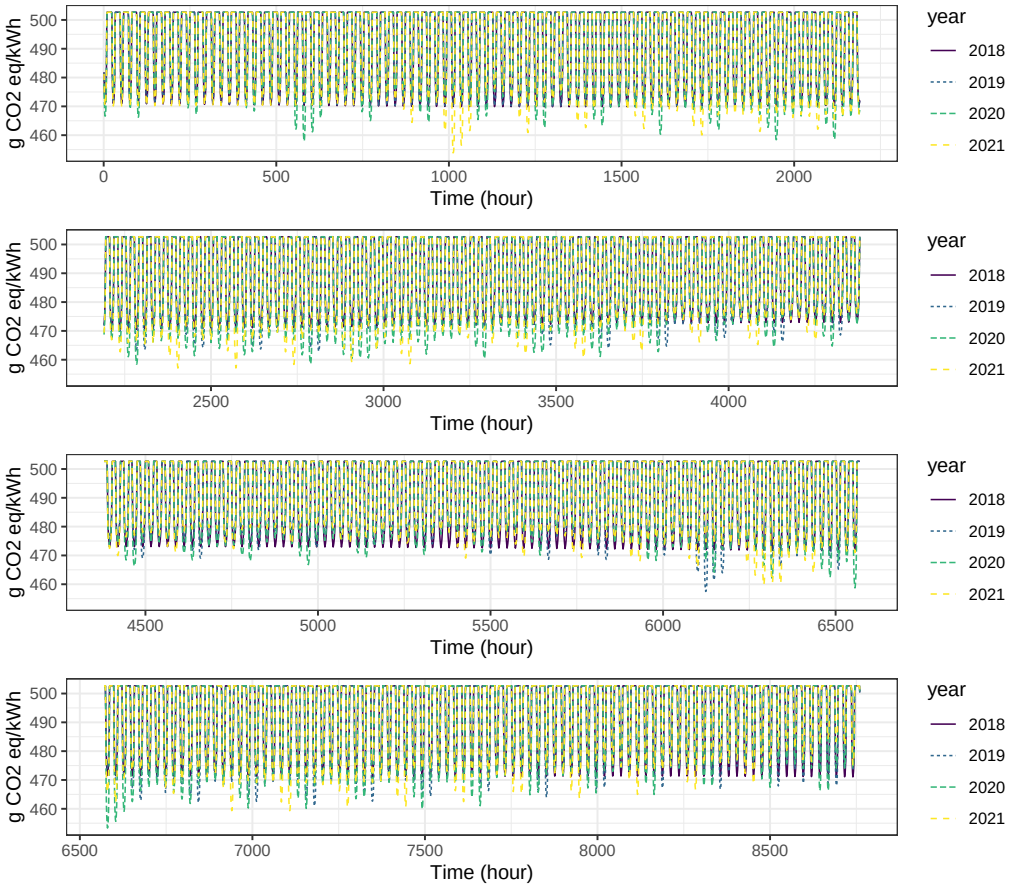

Figure 5.9 Seoul's hour by hour grid emissions for the years 2018, 2019, 2020 and 2021.

5.4.2.2 Results

5.4.2.2.1 Intermittency of renewable sources Table 5.9 presents the results in terms of CO_2 emissions for the renewable infrastructure that included both PV panels and wind turbines, and in Table 5.7 only PV panels are considered.

Table 5.7 Emissions (in kt CO_2-eq) for the different scenarios — renewable infra only have PV panels and batteries.

Sizing Scenario	Emissions	Diff. to baseline (%)
Optimal sizing 2010 - 2019	408.767	-
Average irradiation 1980 - 2009	414.956	1.49
Median irradiation 1980 - 2009	422.928	3.35

Table 5.8 Emissions (in kt CO_2-eq) for the different scenarios — renewable infra only have wind turbines and batteries.

Sizing Scenario	Emissions	Diff. to baseline (%)
Optimal sizing 2010 - 2019	710.351	-
Average wind speed 1980 - 2009	777.957	9.5
Median wind speed 1980 - 2009	770.891	8.5

Table 5.9 Emissions (in kt CO_2-eq) for the different scenarios — renewable infra has PV panels, wind turbines and batteries.

Sizing Scenario	Emissions	Diff. to baseline (%)
Optimal sizing 2010 - 2019	301.931	-
Average irrad. and wind speed 1980 - 2009	342.584	11.9
Median irrad. and wind speed 1980 - 2009	340.784	11.4

5.4.2.2.2 Fine-grain values of grid emission data (hour by hour)

Table 5.10 presents the increase in CO_2 emissions (relative value in percent) of not having access to the fine-grain value of CO_2 emissions (as hour-by-hour values).

Table 5.10 Difference in total emissions (%) between the scenario where the sizing used the grid year's average value in comparison to using hour by hour information.

Year	Difference (%) in CO_2 emissions
2018	0.34
2019	0.59
2020	0.18
2021	0.74

5.5 Scheduling flexibility to reduce CO_2 emissions

One of the challenges in the sizing process is that once built, the renewable infrastructure dimensions cannot be reduced — this would imply destroying them resulting in environmental impact and wastage of money. This over-sizing of the infrastructure could be generated by the intermittency of the renewable sources. In this section, we will evaluate how the workload scheduling may mitigate the effects of a bad-sizing process, in particular what are the gains in terms of reduction of carbon emissions if we delay the execution of the workload to handle renewable power fluctuation.

5.5.1 Updates in the model

Suppose that we have the flexibility to delay α percent of workload up to β time slots. We need two new variables to model this new feature. Let $alloc_k^d$ represent the required number of CPU cores to execute the workload allocated to the data center d at the time slot k and $delay_{k,i}^d$ be the required number of CPU cores to execute the workload of data center d that can be delayed from time slot k up to β time slots (i is the index of list of size β for the delayed workload).

This feature also requires new restrictions. Equation (5.10) models the flexibility of allowing α percent of the workload to be delayed, where W_k is the input that represents the total CPU cores demand to execute the workload at time slot k.

$$\sum_{d=1}^{D}\sum_{i=1}^{\beta} delay_{k,i}^d \leq \alpha \times W_k \tag{5.10}$$

Equation (5.11) models the flexibility to delay the workload up to the next β time slots and ensure that all the workload will be executed:

$$\sum_{d=1}^{D}(alloc_k^d + \sum_{i=1}^{\beta} delay_{k,i}^d) = W_k \tag{5.11}$$

The allocated workload at a time slot k, and the delayed workloads from the previous k - β time slots need to respect the data center servers CPU core capacity. Equation (5.12) models this restriction.

$$\sum_{d=1}^{D}(alloc_k^d + \sum_{i=k-\beta}^{k-1} delay_{i,k-i}^d) \leq C_d \tag{5.12}$$

5.5.2 Experiments

To assess the reductions in carbon footprint we can achieve by delaying the workload, we will perform a series of experiments with different combinations for the values of α — how much of the workload it is possible to delay — and $beta$ — how long it is possible to delay the workload — and compare it with the scenario where it is not possible to delay the workload.

5.5.2.1 Settings

The cloud infrastructure, the workload, grid emissions, and the execution environment are the same as in Section 4.4.1. In order to evaluate only the impact of the flexibility of the scheduling in reducing carbon emissions, all the other parameters are fixed: workload, solar irradiation (average from 1980 to 2019), wind speed (average from 1989 to 2019), area of PV panels and the capacity of the batteries, number of wind turbines, power consumption of servers and network devices, and so on. We will present an evaluation for delaying 10% up to 50% of the workload —

α values of 0.1, 0.2, 0.3, 0.4, and 0.5 — for the period of one hour up to one week — β values of 1 h, 12 h, 24h, 48 h, 96 h, 120 h, 144 h and 168 h.

5.5.2.2 Results

The results of the experiments in terms of reduction of CO_2 emissions are presented in Table 5.11.

Table 5.11 Reductions of total carbon emissions (%) in comparison to the scenario where is not possible to delay the workload.

α \ β	**1 h**	**12 h**	**24 h**	**48 h**	**72 h**	**96 h**	**120 h**	**144 h**	**168 h**
10%	0.46	2.59	3.14	3.48	3.66	3.76	3.81	3.85	3.85
20%	0.84	3.4	3.85	4.11	4.21	4.21	4.21	4.22	4.22
30%	1.15	3.71	4.07	4.25	4.25	4.26	4.26	4.27	4.27
40%	1.42	3.89	4.15	4.25	4.26	4.27	4.28	4.28	4.29
50%	1.65	4.01	4.22	4.26	4.27	4.28	4.29	4.3	4.3

5.6 Monetary costs of reducing the carbon emissions

In the previous sections, we showed the results only in terms of the environmental impact of the cloud operation, now we will present the results in terms of monetary costs (in dollars) of operating the cloud platform and how much it would cost for the DC operator to reduce their environmental impact. This is also an important aspect, as the decision-makers want their business to be lucrative.

5.6.1 Updates in the model

To perform this new analysis, we need some modifications in our modeling. The first modification regards buying electricity from the regular grid, and it is modeled

by Equation (5.13), where $costGrid_k^d$ is an input that represents the costs of buying electricity at each location (in dollars per kWh) at time slot k.

$$PriceGrid_k^d = Pgrid_k^d \times costGrid_k^d \tag{5.13}$$

To measure the price of electricity from renewable sources, we can use the metric Levelized Cost of Energy (LCOE) [51]. This metric compares the total costs of manufacturing and operating the electricity infrastructure (for example the PV panel) with the total energy it can deliver in its lifetime, and the metric is in the form of costs per unit of electricity generated — $ per kWh.

Similar to the CO_2 emissions of renewable sources, the price will also differ per location, as the total amount of irradiation and wind speed differ.

Equation (5.14) models the costs of consuming energy from the solar panels, where $PVLCOE^d$ represents the specific LCOE at the location of data center d (in dollars per kWh).

$$PricePV_k^d = PPV_k^d \times PVLCOE^d \tag{5.14}$$

For the wind turbines, we also use the LCOE to compute its price, and Equation (5.15) models the costs of using energy from the WT, where $WTLCOE^d$ represents the specific LCOE at the location of data center d (in dollars per kWh).

$$PriceWT_k^d = PWT_k^d \times WTLCOE^d \tag{5.15}$$

For the batteries, the metric Levelized Cost of Storage (LCOS) is similar to the LCOE, and it represents the cost per unit of energy discharged (dollars per kWh or MWh) that takes into account its whole life cycle: calendar and cycle life, Max Depth of Discharge limitations, and costs from operation and maintenance [59].

Equation (5.16) models the costs of the energy used from the batteries using its LCOS as input ($BatLCOS$ in dollars per kWh of electricity discharged). To avoid the oversizing of the batteries, the restriction of Equation (5.3) that makes the battery starts empty on the beggining of the day is also used in this scenario.

$$PriceBat_k^d = Pdch_k^d \times BatLCOS \tag{5.16}$$

We can also modify our objective function to minimize the price of operating the cloud platform instead of its environmental impact, and Equation (5.17) model this new objective.

$$\text{minimize} \sum_{k=0}^{K-1} \sum_{d=1}^{D} PriceGrid_k^d + PriceBat_k^d + PricePV^d + PriceWT_k^d \tag{5.17}$$

5.6.2 Experiments

To analyse the monetary costs of operating the cloud federation, we will present results in terms of price (millions of dollars) and carbon footprint (tons of CO_2-eq) for a series of experiments with both the objective of minimizing the monetary costs and the carbon emissions. Additionally, the reader will find a comparison of the renewable infrastructure dimensions between the solution of minimal monetary costs and minimal carbon footprint.

5.6.2.1 Settings

The cloud infrastructure, the workload, and the execution environment are the same as in Section 4.4.1. In this scenario, since the objective is minimize the costs, the

carbon footprint from the grid and renewable infrastructure is not considered for the experiments.

For the price of solar panel electricity, we use the tool "Comparative Photovoltaic Levelized Cost of Energy Calculator" from the National Renewable Energy Laboratory of the USA [4] — version 2.0.0 of August 2021. This tool provides a detailed cost model for many parts of the PV panel, with adequate default values. For reproducibility purposes, Table 5.12 lists the values of the parameters used, and the only differences between the default values were for the parameters Service life, Performance Efficiency, and the Energy yield (that depends on the location where the PV module is installed). For the latter, the following values where considered: 1934.03 for Canberra; 1648.08 for Seoul; 1346.32 for Paris;1737.38 for Virginia; 2215.17 for Dubai; 1549.23 for Singapore; 2125.62 for Pune; 2238.52 for Johannesburg; and 1904.66 for São Paulo.

Table 5.12 Parameters used in the LCOE PV calculator.

Parameter name	**Value**
Cell technology	mono-SI
Package type	glass-polymer backsheet
System type	fixed tilt, utility scale
Inverter loading ration	1.5
Front layer cost (USD/m^2)	3.5
Cell cost (USD/m^2)	22.20
Back layer cost (USD/m^2)	2.40
Non-cell module cost (USD/m^2)	13.60
Extra component cost (USD/m^2)	0
Operations and Maintenance cost (USD/kWDC/year)	3.5
Balance of system cost,power-scaling (USD/W)	0.2
Balance of system cost,area-scaling (USD/m^2)	53.38
Performance Efficiency (%)	15.0
Energy yield (kWh/kWDC)	specific per location
System degradation rate (%/year)	1.0
Service life (years)	30
Discount rate	6.3

For computing wind turbines' LCOE (dollars per MWh), we also used the modeling from NREL [54]. Equations (5.18) represents how to calculate the LCOE for WT,

where CapEx is the capital expenditures (dollars per kilowatt) — represents the initial investment cost in the infrastructure, OpEx is the operation expenditures (dollars per kilowatt per year), FCR is the fixed charge rate (%) — the amount of revenue per dollar of investment that must be collected annually from customers to pay the carrying charges on that investment, such as return on debt and equity, income and property tax, book depreciation and insurance [51]. We used the default values for a Distributed Single-Turbine (Large): Capex = 3540; Fcr = 5.42; OpEx = 35. For the net annual energy production (in MWh/MW/year), it also depends on the location where the wind turbine is installed: 1138.5 for Canberra; 829.57 for Seoul ; 2065.38 for Paris; 1289.12 for Virginia; 1217.64 for Dubai; 754.0 for Singapore; 869.19 for Pune ; 1143.32 for Johannesburg; and 881.95 for São Paulo.

$$LCOE_{wt} = \frac{(CapEx \times FCR) + OpEx}{\frac{AEP_{net}}{1000}} \tag{5.18}$$

For the batteries, we used values from a technical report of the US Department of Energy [59], where each kilowatt hour of energy discharged (LCOS) has a price of 20 cents.

Finally, Table 5.13 presents the cost in dollars per kWh of using the electrical grid, PV panels and wind turbines. For the grid, to simulate the variation in price, it is considered that during off peak times — from 22h to 8h — the value is half of the total, as in the work of Khodayarseresht et al. [29].

5.6.2.2 Results

Table 5.14 illustrates the cloud operation's total costs (in dollars), environmental impact (CO_2 emissions), and how much energy from the grid (relation to the total energy consumed by the cloud federation) was used to supply the cloud

Table 5.13 Price of different sources of energy (USD per kWh) at each location. Source for the grid electricity prices: Petrolprices.org.

Location	Grid	PV	WT
Johannesburg	0.074	0.0385	0.1984
Pune	0.104	0.0406	0.2610
Canberra	0.331	0.0445	0.1993
Dubai	0.101	0.0390	0.1863
Singapore	0.272	0.0557	0.3009
Seoul	0.092	0.0525	0.2735
Virginia	0.150	0.0498	0.1760
São Paulo	0.144	0.0453	0.2572
Paris	0.340	0.0643	0.1098

federation operation considering multiple scenarios. The *Minimum Costs* scenario refers to solving the LP using the objective of Equation (5.17) — minimizing the monetary costs; the *Minimum CO_2* refers to solving the LP using the objective of Equation (4.9) — minimize the carbon footprint, and for both cases the reader will find in parentheses the configuration of the renewable infrastructure used; and *Only grid* represents the case where there is no on-site renewable infrastructure in the DCs and they only use power from the local electricity grid.

Table 5.14 Total costs (millions of $) and emissions ($t\,CO_2 - eq$) for different scenarios.

Scenario	Millions of $	$t\,CO_2 - eq$	Grid usage (%)
Min. cost (PV + WT + Bat + grid)	36.76	141 001.67	55.5
Min. cost (PV + Bat + grid)	38.34	145 193.64	59.69
Min. cost (WT + Bat + grid)	51.08	204 568.07	87.79
Min. CO_2 (PV + Bat + grid)	72.8	37 828.49	16.90
Min. CO_2 (WT + Bat + grid)	277.52	58 263.38	22.43
Min. CO_2 (PV +WT+ Bat + grid)	117.31	24 977.89	5.83
Only grid	56.66	222 876.62	100.00

Finally, Figures 5.10, 5.11, and 5.12, show the difference in the renewable infrastructure sizing when the objective function is to minimize the costs and the carbon emissions of the cloud federation operation.

Table 5.15 Total costs (millions of $) and emissions ($t\,CO_2 - eq$) for different scenarios.

Scenario	Millions of $	$t\,CO_2 - eq$	Carbon tax	CO_2 red. (%)
PV + WT + Bat + grid	52.62	126 944.19	119	9.9
PV + WT + Bat + grid	43.53	136 048.94	49	3.5
PV + WT + Bat + grid	37.74	140 382.56	7	0.4
PV + WT + Bat + grid	36.76	141 001.67	0	-
Grid only	82.62	216 855.76	119	2.7
Grid only	67.44	216 866.90	49	2.7
Grid only	58.22	222 876.62	7	0.0
Grid only	56.66	222 876.62	0.0	-

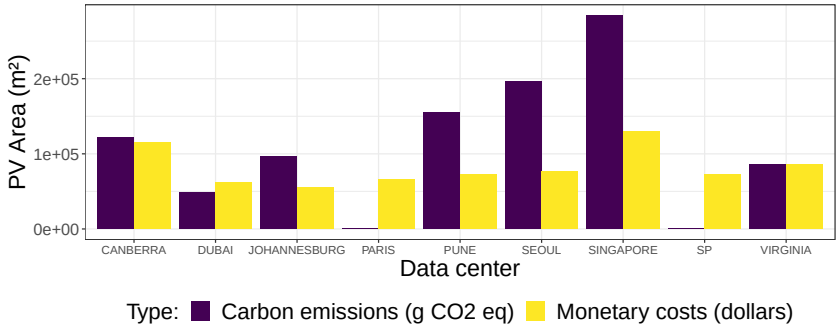

Figure 5.10 PV Area to minimize costs in comparison to minimizing carbon emissions.

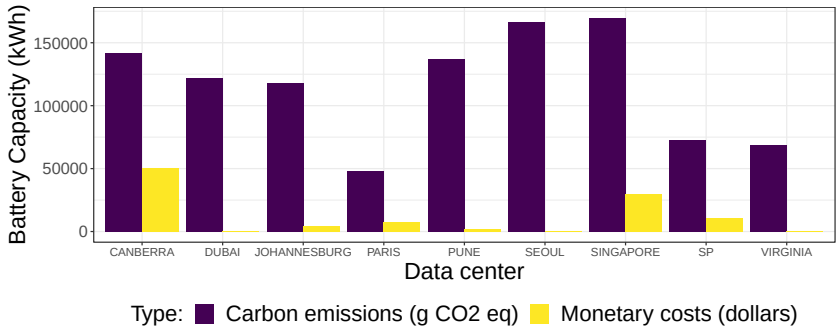

Figure 5.11 Battery capacity to minimize costs in comparison to minimizing carbon emissions.

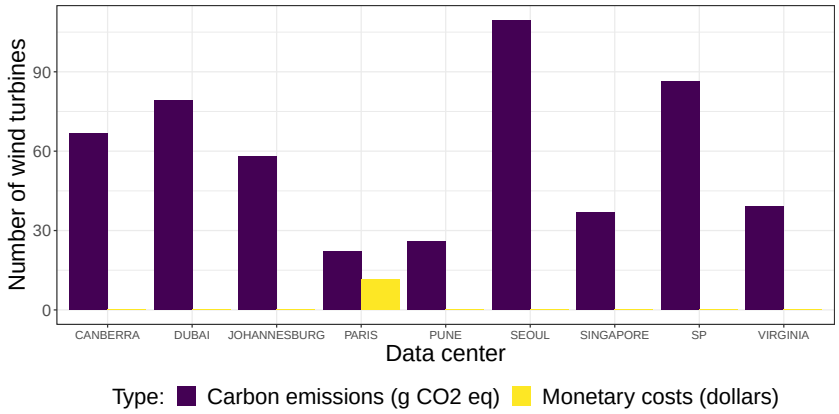

Figure 5.12 Number of wind turbines to minimize costs in comparison to minimizing carbon emissions.

5.7 Adding or replacing servers considering new generations

In this section we show the necessary modifications in our modeling to evaluate the decision of manufacturing servers from new generations to match the cloud workload demand, considering that the servers also emit CO_2 during their lifetime. It is assumed a long-term operation of the cloud federation (10 years for example) and that new servers will be launched during the lifetime of the cloud data centers. The decision-maker of the cloud federation will determine whether to add new servers to operate with the older ones from previous generations or to replace them.

5.7.1 Updates in the model

To model this approach, two new variables were created: NS^d, which represents the number of new servers to manufacture at data center d; and REP^d_{gen}, the number of servers of the generation gen to replace at data center d. For replacing the servers from previous generations, it is considered that the data center computational capacity in the number of CPU cores cannot decrease, that is, the total number of

CPU cores of the new servers needs to be greater or equal to the total number of CPU cores from the servers replaced. This constraint is modeled by Equation (5.19).

$$NS^d \times cpucores_{newgen} \geq \sum_{oldergen} REP^d_{oldergen} \times cpucores_{oldergen} \tag{5.19}$$

The number of servers replaced by a generation cannot be greater than the number of servers of this generation that exist in the DC (ES^d_{gen}). The value of ES^d_{gen} is extracted from the solution of the previous year. Equation (5.20) models this constraint.

$$REP^d_{gen} \leq ES^d_{gen} \tag{5.20}$$

Given that a data center may have servers of different generations, it is also necessary to update the models regarding workload execution and power consumption. Now, we have a variable $w^d_{k,gen}$ that represents the workload that is executed on the servers of a generation gen at the data center d during time slot k. We have two constraints for this variable: one for the servers that already exist in the DC, and another for the new servers that will be manufactured in the DC. Equation (5.21) models this constraint.

$$w^d_{k,gen} \leq \begin{cases} (ES^d_{gen} - REP^d_{gen}) \times cpucores_{gen} & \text{if server from older generation} \\ NS^d \times cpucores_{gen} & \text{if server from new generation} \end{cases} \tag{5.21}$$

Equation (5.22) models the constraint that all the workload must be executed, and that it can be run in any generation of the servers.

$$w_k = \sum_{T_i | r_i \leq k\Delta t < r_i + p_i} c_i = \sum_d \sum_{gen} w^d_{k,gen} \tag{5.22}$$

We also assume that the workload is classified into two categories: services that are mandatory tasks that execute in the DCs, and batch tasks that can execute in any of the DCs. This assumption represents the modern workload of cloud platforms and also avoids allocating all the workload to the DC with the lowest-carbon intensity grid and only adding and replacing servers on this DC. To model the classification of the workload in two categories, we need a new parameter γ that represents the ratio of tasks that are of the service type (and 1 - γ tasks are of the batch type) executed at each data center d. Equation (5.23) models this restriction.

$$\sum_{gen} w_{k,gen}^{d} \geq w_k \times \gamma^d \tag{5.23}$$

Equation (5.24) presents the new modeling of the power consumption for all the servers of different generations at a DC d, and Equation (5.25) models the power consumption of the DC (including the costs of the intra-network equipment and the cooling devices).

$$PServers^d = (\sum_{gen} ES_{gen}^d \times Pidle_{gen}^d + w_{k,gen}^d \times Pcore_{gen}) \tag{5.24}$$

$$P_k^d = PUE^d \times (Pintranet^d + PServers^d) \tag{5.25}$$

Regarding the carbon footprint of the servers, it is assumed that it is amortized over the server lifetime and the LP considers the annual value. For example, if manufacturing a server emits 800 kg of CO_2-eq and its expected lifetime of 4 years, 25% (200 kg of CO_2-eq) of its manufacturing emissions will be accounted on the 1st year of operation, other 25% at the second year and so on. After the fourth year, the server can still remain in the cloud platform, and its carbon emissions will come only from the power consumption — all the carbon emissions from manufacturing have been amortized. On the other hand, if the server is replaced in the second year of

operation, the 75% remaining of its emissions from manufacturing will also be taken into account. Equation (5.26) illustrates the footprint of building additional servers at a given data center d, where $serversCO2$ is an input with the total CO_2 emitted from manufacturing the server (in $g\,CO_2 - eq$), and ϵ the share of the emissions by year — ϵ is equal to 0.25 on the previous example.

$$FPns^d = NS^d \times (serversCO2 \times \epsilon) \tag{5.26}$$

Equation (5.27) illustrates the footprint of replacing servers from previous generations at a given data center d, where $repCO2_{gen}$ is the emissions (in kg of CO_2 eq) from replacing the old servers that depend on the server age (age_{gen}) in years. The function that computes the value of $repCO2_{gen}$ is defined in Equation (5.28).

$$FPrep^d = \sum_{gen} REP^d_{gen} \times repCO2_{gen} \tag{5.27}$$

$$repCO2_{gen} = \begin{cases} (1 - (age_{gen} * \epsilon)) \times serverCO2 & \text{if } age_{gen} < \text{expected lifetime} \\ 0 & \text{otherwise} \end{cases} \tag{5.28}$$

We also have the carbon footprint of the servers that were manufactured in previous years. Equation (5.29) illustrates the emissions of the **e**xisting **s**ervers where $esCO2_{gen}$ represents the emissions of operating the server of the generation gen during that year (in $g\,CO_2 - eq$) and its value is computed by the function in Equation (5.30).

$$FPes^d = \sum_{gen} (ES^d_{gen} - REP^d_{gen}) \times esCO2_{gen} \tag{5.29}$$

$$esCO2_{gen} = \begin{cases} \epsilon \times serverCO2 & \text{if } age_{gen} < \text{expected lifetime} \\ 0 & \text{otherwise} \end{cases} \tag{5.30}$$

It is assumed that the renewable infrastructure cannot change from one year to another. Equations (5.31), (5.32) and (5.33) model these constraints, where $lowboundbat^d$, $lowboundpv^d$ and $lowboundwt^d$ are the capacity of batteries, area of PVs and number of wind turbines, respectively, obtained from the sizing process.

$$Bat^d = lowboundbat^d \tag{5.31}$$

$$A^d = lowboundpv^d \tag{5.32}$$

$$WT^d = lowboundwt^d \tag{5.33}$$

Finally, Equation (5.34) presents the new objective function of operating the cloud platform considering that new servers can be manufactured, and servers from previous generations can be replaced.

$$\text{minimize } \sum_{k=0}^{K-1} \sum_{d=1}^{D} (FPgrid_k^d + FPpv_k^d + FPbat_k^d) + \sum_{d=1}^{D} FPns^d + FPrep^d + FPes^d \tag{5.34}$$

5.7.2 Experiments

Two scenarios will be considered in the experiments to assess the impact of the decision to manufacture new servers in terms of CO_2 emissions. In the first, the LP is

solved only by having information on the server of the current year and will generate a short-term plan for the duration of the expected lifetime of the servers (4 years) to consider all the carbon emissions from manufacturing. Suppose a new, more computationally powerful and energy-efficient server is launched in the next year. In that case, the initial decision of the previous year can be modified to consider the new server. This greedy approach is then compared to the optimal solution, in which all the server specifications, workload, and climate conditions are known for a longer period — 10 years, and we use the metric total carbon emissions to compare both approaches.

5.7.2.1 Settings

We consider ten years of operation and seven generations of servers. Table 5.16 lists the difference in terms of hardware characteristics, power consumption, and the carbon footprint of manufacturing each generation. For the CO_2 of manufacturing the servers, the methodology from the work of [21] is used, which presents how to compute the carbon footprint of servers considering the integrated circuits of RAM, storage (HD and SSDs) and CPU.

Table 5.16 Servers specifications for different generations.

Year(s)	CPU	CPU Cores	P idle	P Core	Nodes	$kg\,CO_2 - eq$
1	Intel Xeon E5-2630	20	52	7.5	1	-
2, 3	Intel Xeon E5-2699 v3	36	40.1	6.3	1	577.2
4	Intel Xeon E5-2698 V4	40	43.5	7.1375	1	569.4
5, 6	Intel Xeon Platinum 8180	56	48.9	6.68	1	578.6
7, 8	AMD EPYC 7742	64	66.1	2.71	1	587.2
9	AMD EPYC 7763	128	75.6	3	1	590.3
10	AMD EPYC 9654	192	126	3.65	1	614.9

All the servers are based on the Dell R740 for the HD and SSD values (HD with 3.84 TB and SSD with 400GB), with the same values for CO_2 as in the work from [21], and they have 256 GB of RAM (16 modules of 16 GB DDR4 10 nm). For the CPU, the CO_2 depends on the die area size and the processor size (in nanometers), and the respective values can be seen in Table 5.17. For the first generation, it is

considered that the servers have been used more than their expected lifetime, and all their CO_2 footprint have been paid. Finally, it is considered that the servers have an expected lifetime of 4 years.

Table 5.17 CPU specifications in terms of die area and processor size. Source: x86guide.net.

Model	CPU die area (cm^2)	Processor size (nm)
Intel Xeon E5-2699 v3	6.62	20
Intel Xeon E5-2698 V4	4.56	14
Intel Xeon Platinum 8180	6.94	14
AMD EPYC 7742	5.92	7
AMD EPYC 7763	6.45	7
AMD EPYC 9654	8.66	5

To obtain the area of solar panels, capacity of batteries and the number of wind turbines, the sizing process was realized using the average irradiation and wind speed from 1980 to 2019, as in Section 5.3.2.1, and the servers configurations of the year 1. For the remaining 9 years of operation, the wind speed data and solar irradiation used are from the years 2010 to 2018. Finally, we assume that for each year there is an increase of 10% in the workload, that is, the number of the needed number of CPU cores to process the tasks at each time slot increases 10% year after year.

Two scenarios are considered in the experiments. In the first, the LP is solved year by year, and it only has access to the current year's server specification. This will generate a plan for the expected server lifetime of 4 years, which can be updated when new servers are launched in the subsequent years. In other words, the LP is solved for four years – starting from the year x — considering that it is possible to manufacture the server of the current year that will be used together with the existing computational infrastructure or to replace older servers. If a new server is launched in the year $x+1$, the initial plan is reviewed to access manufacturing the new server and it can generate a plan for the next 4 years (until year $x+5$) taking into account the servers that were manufactured in the year x. After the LP is solved for the year x, the results in terms of new servers manufactured and older ones replaced will be used as input to the year $x+1$. In the second scenario, the

nine years of operation are solved at once, therefore the LP has access to all the information about future workload, climate conditions, and server configurations. The first scenario represents an approach more realistic that has the potential to be taken by the decision maker, considering that it is hard to have accurate values for workload, climate conditions and server configurations in the long term, and the second scenario is only used as baseline — it represents the optimal solution.

5.7.2.2 Results

Figure 5.13 presents the result of the DCs evolution when the decision is made year by year, that is, at each year the cloud-federation operator has information about the configurations of the current generation's server, and the decision is made to manufacture these new servers to add or to replace older ones. Figure 5.14 presents the optimal solution, when all the information about the workload, servers specifications, and climate conditions, is known in advance for the ten years. In both figures, each bar represents the computational capacity of the data center (in number of CPU cores) over the years, and the colors represent which server's generation the cores belong to (more recent generations have darker colors). One can also observe when new servers are added — when the bar's height of the corresponding generation increases — and when servers from older generations are replaced for newer ones — when the bar's height of the corresponding generation decreases.

We observe that the optimal solution make very few changes in the servers, because it knows in advance when is the right moment to add/replace the servers. However, in real-life the first approach where the decision is made year by year is more close to what the cloud-federation operator would do. Finally, in terms of total CO_2 emissions, the optimal solution is 11.5% better than the solution that only has information about the current year, as can be seen in Table 5.18.

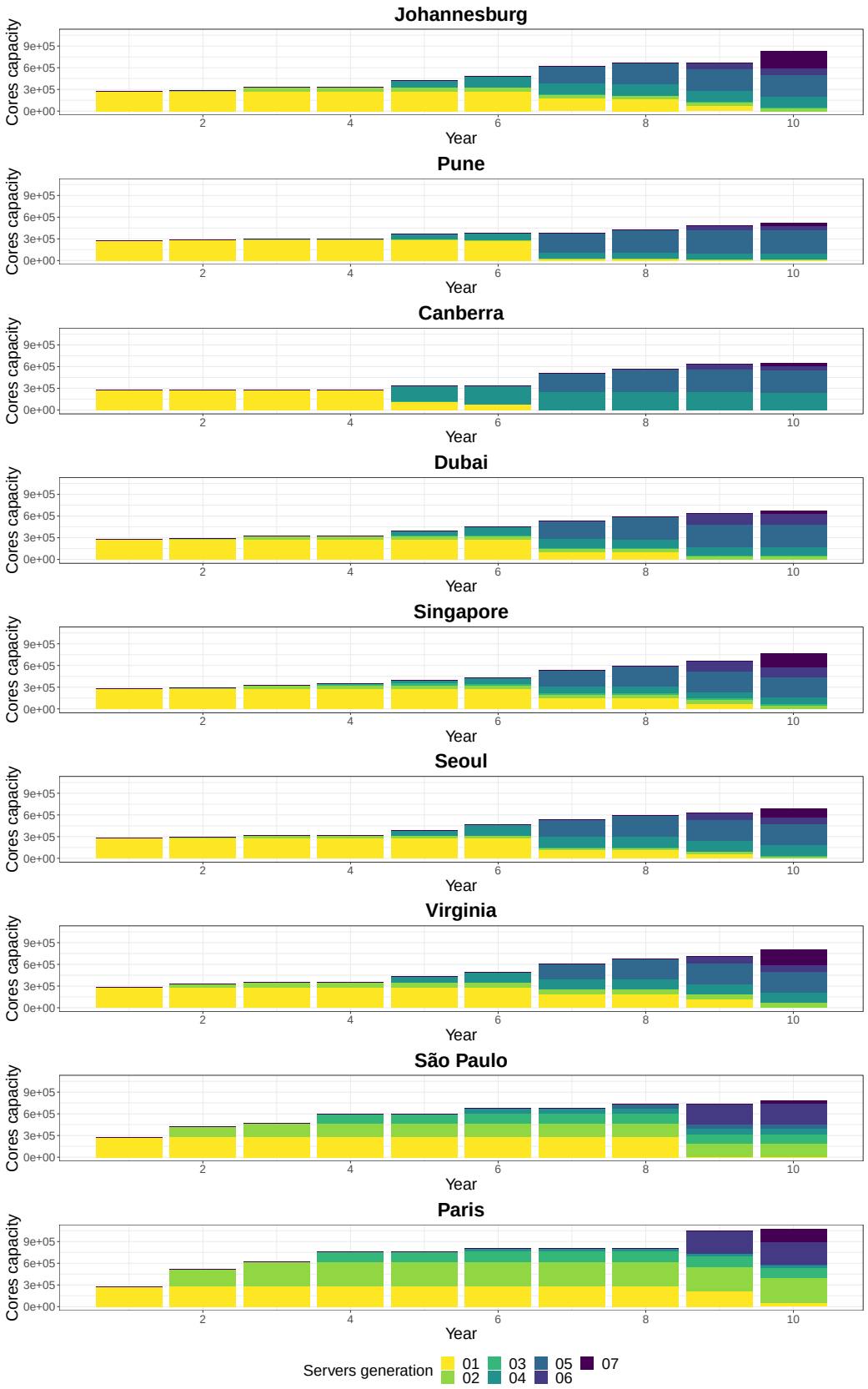

Figure 5.13 Cloud federation evolution knowing only the information of the current year.

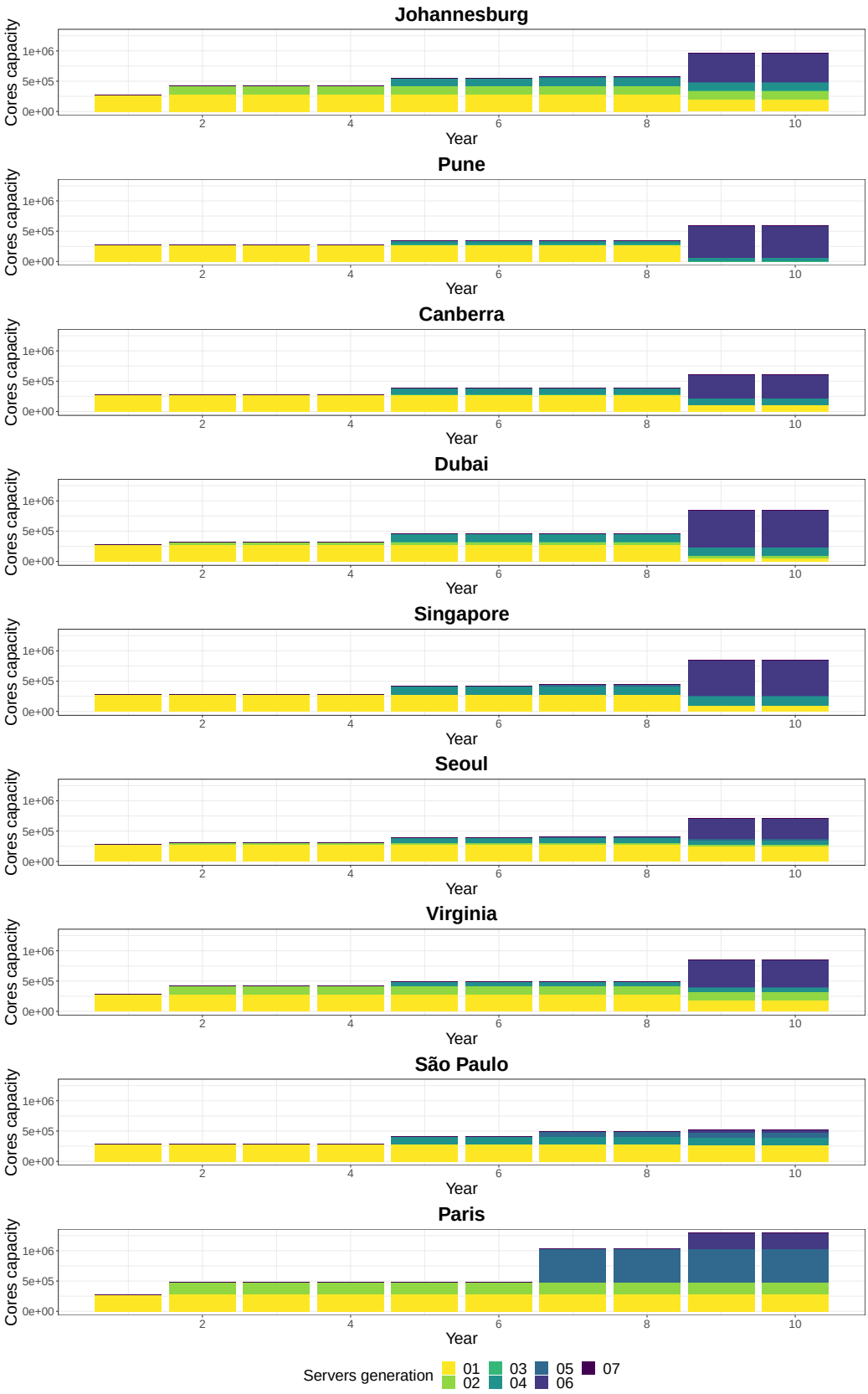

Figure 5.14 Cloud federation evolution knowing all the information of the 10 years — optimal solution.

Table 5.18 Total emissions for the different scenarios.

Scenarios	Emissions ($t\,CO_2 - eq$)
Sizing with all the information of the 10 (optimal solution)	302 214.38
Sizing with information of the current year	267 391.20

5.8 Discussion

In this section, the reader will be presented to an analysis of the experiments and their results. The scope of the study is how to reduce the environmental impact that comes from the long-term operation of a cloud federation — in terms of CO_2 emissions — by adding locally installed renewable infrastructure: solar panels, wind turbines, and batteries (to deal with intermittency). One must not neglect that the renewable infrastructure also emits carbon during its whole lifetime. We made the modifications in our modeling — that initially only considered emissions from manufacturing — and, as expected, there was an increase in the total emissions — around 8%. It is also possible to observe that the area of PVs was reduced in most of the locations — Figure 5.2, with the exception of Johannesburg. This could be explained by the fact that Johannesburg is the location that receives the most solar irradiation throughout the year, so it could compensate for the decrease of PV area in the other locations. Regarding the batteries, the reduction in Pune, Seoul, and Singapore could be justified by the smaller PV area, as there will be less energy to store.

On the other hand, we see an increase in the battery capacity for the other locations. This could be justified by having a smaller PV area so the renewable energy cannot be wasted. Finally, including the life cycle aspect makes our modeling more realistic, and the emissions are still better than the scenario where only power from the local electricity grid is used in terms of carbon footprint — 81% in the scenario with PV and batteries.

Regarding the wind turbines, we saw that including them in the renewable infrastructure has a significative positive impact on the carbon footprint of operating the cloud federation: a 34% reduction in CO_2 emissions in comparison to only using solar panels and batteries. This reduction could be justified by the fact that renewable power is produced even when the sun is not shining and throghout the whole year, which results in a reduction in the required PV area and batteries capacity that can be observed in Figures 5.4 and 5.5 — in the previous chapter we saw the dimensions of PV and batteries needed to deal with the low solar irradiation during winter.

Nevertheless, the number of necessary wind turbines to manufacture is quite high, and this is a problem considering the land area requirements of wind turbines. One study from NREL that evaluated the land area requirements in the United States reports that the average capacity density (relation between wind power production and required land area) is 3.0 ± 1.7 MW/km² [12]. Taking into account that in our experiments we considered wind turbines with a rated power of 1.5MW, this implies that for each square kilometer it is possible to install from one up to three wind turbines. This would not be possible for data centers located in urban centers that are highly populated. For example, in the case of Paris — the one with the lowest number of wind turbines — we would need from 7 to 23 km², and in the case of Seoul — the one with the highest number of wind turbines — it would require 36 to 110 km². In comparison to solar panels, we observe that for all the evaluated locations — with the exception of Paris — the wind turbines have a lower potential for power generation — as seen in the Capacity Factor values of Table 5.4. Taking into account these points, it may be more interesting for the cloud federation operator to buy power from wind farms located near its data centers than to install wind turbines on them.

As said in Chapter 4, solar irradiation has a lower variance than wind speed, impacting the sizing process's quality. We saw that in the comparison of the optimal scenario

for the renewable infrastructure that only considered PV panels and batteries, the sizing process that used simple estimation techniques (average and median values) had precise results, being up to 3% worse than the optimal scenario (Table 5.7). On the other hand, the scenario that only consider wind turbines and batteries as the green power infrastructure (Table 5.8) was significatively worse when comparing with the optimal solution — around 11% — and presented more than 2 times the carbon emissions of only using solar panels — which can be justified by the lower potential to generate wind power on the selected locations as seen in Table 5.4. For the scenario with a hybrid configuration of solar panels and batteries, it is possible to observe in Table 5.7) that it became less precise — up to 12% worse than the optimal — as it is affected by the uncertainties of solar irradiation and wind speed. Nevertheless, even in the worst case, the reduction in total CO_2 emissions by using the hybrid configuration with PVs and WTs is significant: the worst scenario for PV + WT + Bat has a 17% lower carbon footprint than the scenario with PV + Bat.

In what concerns the sensitivity of the model to the grid emissions input, we saw that the initial modeling that uses the average value of the year is robust: using the hour-by-hour value reduces the total CO_2 emissions by less than 1% — as seen on Table 5.10. This result presents a positive outcome, considering that not every location provides a fine-grain value of the carbon intensity of its electricity grid.

On the topic of exploring the flexibility in the scheduling of the workload to reduce carbon emissions, it is possible to extract the following information from the results. The first is that the reduction of the carbon footprint seems to have a limit (around 4%). This can be justified by the fact that there is little difference in the climate conditions during the interval of one week, or that the DC, which would reduce the carbon emissions, is already at peak capacity and cannot receive more workload. The second interesting observation is that by delaying a small portion of the workload (for example 10% to 20%) the reduction is already really close to the maximum possible reduction value. Finally, this approach can be used to reduce the impact

of renewable sources' intermittency in the sizing process — as we saw differences from 1% to 12% in comparison to the optimal scenario on Section 5.4 — and in the context of cloud federations that emits from tens to hundreds of kilotons of $CO_2 - eq$, a reduction of 4% is not negligible.

Concerning the costs of reducing the environmental impact of the cloud operation — one of the most important variables for the decision-makers — we see in Table **??** that the hybrid configuration that used solar panels and wind turbiness is the cheapest — around 35% cheaper and emits 37% less CO_2 in comparison with using only the grid. On the other hand, the solution that results in the lowest environmental impact (*Minimum CO_2 PV + WT + Bat + grid*) is the second most expensive: it presents a reduction of a factor of 9 in CO_2 emissions than the grid only scenario but it costs the double. The results allows comparing the benefiits of wether only using wind turbines or solar panels as well. We observe that only using wind power as renewable source (*Min. CO_2 (WT + Bat + grid)*) results in the most expensive scenario and the third best in terms of environmental impact — in comparison with only using power from the grid it is almost 5 times more expensive and emits 73% less. The scenario where the renewable infrastructure is only composed of PV panels (*Min. CO 2 (PV + Bat + grid)*) is the best in terms of costs and environmental impact: it has a price close to the grid-only scenario — 22% more expensive — and has a reduction of 81% in CO_2 emissions. One can also observe that in terms of minimizing the operation monetary costs, the cloud federation will be less independent from the grid: using from 55% to 88% of grid power to supply its energy demand. In contrast, when the objective is to minimize the environmental impact, the cloud federation will be more autonomous: consuming from 6% to 22% of grid power.

When considering the long-term operation of cloud data centers, the evolution of the technology hardware and increase in computational workload is of significant concern as well. Furthermore, these two variables increase the uncertainty of the problem: how power-efficient will the CPUs be, or how much will the workload

increase in the next decade? The approach presented in this chapter, where the choice of manufacturing new servers is made year by year, is closer to what could be adopted by the decision-makers of cloud federations, and it presented good results: in comparison to the optimal scenario — unlikely to be achieved as it needs all the information of future climate conditions, hardware characteristics and workload of 10 years — it is 11.5% worse. Furthermore, we see that in the optimal solution, the servers are being used way more than the expected lifetime of 4 years: except for the Pune DC, the first-generation servers are still being used nine years after their expected lifetime. This is interesting to motivate extending the lifetime of servers, as they are computationally powerful, and discarding them presents other environmental impacts than carbon emissions.

Finally, all the modifications in the modeling does not change its computational complexity, we still uses only linear variables, and it can be solved optimally in polynomial time.

5.9 Summary

In this chapter, we studied the problem of reducing the long-term environmental impact of cloud federations by integrating local renewable infrastructure in the sites where the data centers reside. We extended our initial modeling, presented in Chapter 4 in the following points: considered the whole life cycle of the renewable infrastructure; integrated wind turbines in the model to further reduce the environmental impact, in particular when the sun is not shining; evaluated how sensible the model is to the inputs of solar irradiation, wind speed and grid emission values; showed that allowing the workload to be delayed can reduce the carbon footprint of the cloud operation, and partially mitigate the erros of the sizing process caused by the intermittency of renewables, which is important given that once built it is not feasible to reduce the dimensions of the renewable infrastructure; conducted

an analysis of the monetary costs of minimizing the cloud's environmental impact, and illustrated that the renewable infrastructure indeed can reduce costs for the cloud operators; and, presented one approach for deciding when to manufacture new servers taking into account their carbon footprint, to deal with the workload increase over the years, where we showed that servers could be extended way longer than their expected lifetime. These modifications were carefully made to use linear variables and restrictions, therefore the problem remains optimally solved in polynomial time.

All these scenarios may further help the decision-makers in the quest to reduce the environmental impact of their cloud federation. For example, we saw that the renewable infrastructure that included wind turbines had the lowest environmental impact, however, in comparison to PV panels, wind turbines require a large land area — PVs could be installed on the rooftops — and are more expensive, therefore their adoption needs to be thoroughly examined — in particular if the data centers reside on large urban areas.

As future work, many research trajectories can be explored, for example: i) including degradation of the renewable infrastructure; ii) the monetary costs of servers for the decision of manufacturing new generations to replace older ones; iii) degradation and failure of specific hardware components of the servers, and the monetary costs and environmental impact of fixing and replacing them; and iv) flexibility in the scheduling could reduce the number of servers needed.

General Discussion

6

6.1 General Remarks and Future Research Directions

In this thesis we aimed to ...

contar a historinha dos capitulos,

no fim falar das possiveis work directions para trabalhos futuros ...

hardwares especificos

degradacao

6.2 Work Dissemination

Falar das contribuicoes ...

listar papers:

smart greens, ccgrid, erad, compas, roadef, disseminacao ...

Bibliography

[1] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, et al. “Carbon Explorer: A Holistic Approach for Designing Carbon Aware Datacenters”. In: *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (2023) (cit. on pp. 18, 82).

[2] Ehsan Ahvar, Anne-Cécile Orgerie, and Adrien Lebre. “Estimating Energy Consumption of Cloud, Fog, and Edge Computing Infrastructures”. In: *IEEE Transactions on Sustainable Computing* 7.2 (2022), pp. 277–288 (cit. on pp. 55, 64).

[3] Hashim Ali, Muhammad Zakarya, Izaz Ur Rahman, Ayaz Ali Khan, and Rajkumar Buyya. “FollowMe@LS: Electricity price and source aware resource management in geographically distributed heterogeneous datacenters”. In: *Journal of Systems and Software* 175 (2021), p. 110907 (cit. on pp. 12, 15, 39).

[4] SJ Andrews, B Smith, MG Deceglie, KA Horowitz, and TJ Silverman. *NREL Comparative PV LCOE Calculator*. Version 2.0.1. Aug. 9, 2021 (cit. on p. 107).

[5] M. Baumann, J. F. Peters, M. Weil, and A. Grunwald. “ CO_2 Footprint and Life-Cycle Costs of Electrochemical Energy Storage for Stationary Grid Applications”. In: *Energy Technology* 5.7 (2017), pp. 1071–1083. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/ente.201600622` (cit. on pp. 50, 85).

[6] Manal Benaissa, Jean-Marc Nicod, and Georges Da Costa. “Standalone Data-Center Sizing Combating the Over-Provisioning of the IT and Electrical Parts”. In: *Proceedings of the Workshop on Cloud Computing (WCC)*. 2022 (cit. on p. 17).

[7] B. Camus, A. Blavette, F. Dufossé, and A. Orgerie. “Self-Consumption Optimization of Renewable Energy Production in Distributed Clouds”. In: *2018 IEEE International Conference on Cluster Computing (CLUSTER)*. Belfast, UK: IEEE, 2018, pp. 370–380 (cit. on p. 14).

[8] B. Camus, F. Dufossé, A. Blavette, M. Quinson, and A. Orgerie. “Network-Aware Energy-Efficient Virtual Machine Management in Distributed Cloud Infrastructures with On-Site Photovoltaic Production”. In: *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. Lyon, France: IEEE, 2018, pp. 86–92 (cit. on pp. 14, 15, 24, 25).

[9] Benjamin Camus, Fanny Dufossé, and AnneCécile Orgerie. “A stochastic approach for optimizing green energy consumption in distributed clouds”. In: *SMARTGREENS 2017 - International Conference on Smart Cities and Green ICT Systems*. Porto, Portugal: SMARTGREENS, 2017 (cit. on pp. 14, 24, 25).

[10] Henri Casanova, Arnaud Giersch, Arnaud Legrand, Martin Quinson, and Frédéric Suter. "Versatile, scalable, and accurate simulation of distributed applications and platforms". In: *Journal of Parallel and Distributed Computing* 74.10 (2014), pp. 2899–2917 (cit. on p. 35).

[11] Georges Da Costa, Léo Grange, and Inès de Courchelle. "Modeling, classifying and generating large-scale Google-like workload". In: *Sustainable Computing: Informatics and Systems* 19 (2018), pp. 305–314 (cit. on p. 66).

[12] P Denholm, M Hand, M Jackson, and S Ong. *Land use requirements of modern wind power plants in the United States. NREL/TP-6A2-45834*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2009 (cit. on p. 122).

[13] Alexandre H. T. Dias, Luiz. H. A. Correia, and Neumar Malheiros. "A Systematic Literature Review on Virtual Machine Consolidation". In: *ACM Comput. Surv.* 54.8 (Oct. 2021) (cit. on p. 10).

[14] I. Foster, Y. Zhao, I. Raicu, and S. Lu. "Cloud Computing and Grid Computing 360-Degree Compared". In: *2008 Grid Computing Environments Workshop*. Austin, TX, USA: IEEE, 2008, pp. 1–10 (cit. on p. 8).

[15] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979 (cit. on p. 10).

[16] Ronald Gelaro, Will McCarty, Max J. Suárez, et al. "The Modern-Era Retrospective Analysis for Research and Applications, Version 2 (MERRA-2)". In: *Journal of Climate* 30.14 (2017), pp. 5419–5454 (cit. on pp. 66, 97).

[17] Google. *Environmental Report 2023*. Google, 2023 (cit. on pp. 2, 20).

[18] R. L. Graham. "Bounds for Certain Multiprocessing Anomalies". In: *Bell System Technical Journal* 45.9 (1966), pp. 1563–1581 (cit. on pp. 9, 10).

[19] Greenpeace. *Clicking Clean Virginia*. Greenpeace, 2019 (cit. on p. 2).

[20] U. Gupta, Y. Kim, S. Lee, et al. "Chasing Carbon: The Elusive Environmental Footprint of Computing". In: *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. Los Alamitos, CA, USA: IEEE Computer Society, 2021, pp. 854–867 (cit. on pp. 20, 83).

[21] Udit Gupta, Mariam Elgamal, Gage Hills, et al. "ACT: Designing Sustainable Computer Systems with an Architectural Carbon Modeling Tool". In: *Proceedings of the 49th Annual International Symposium on Computer Architecture*. ISCA '22. New York, New York: Association for Computing Machinery, 2022, pp. 784–799 (cit. on p. 116).

[22] Ori Hadary, Luke Marshall, Ishai Menache, et al. "Protean: VM Allocation Service at Scale". In: *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*. 2020, pp. 845–861 (cit. on p. 36).

[23] Maroua Haddad, Jean-Marc Nicod, Marie-Cécile Péra, and Christophe Varnier. "Stand-alone renewable power system scheduling for a green data center using integer linear programming". In: *Journal of Scheduling* 24 (2021), pp. 523–541 (cit. on p. 58).

[24] Marwa Haddad, Georges Da Costa, Jean-Marc Nicod, et al. “Combined IT and power supply infrastructure sizing for standalone green data centers”. In: *Sustainable Computing: Informatics and Systems* 30 (2021), p. 100505 (cit. on pp. 17, 89).

[25] H. Hersbach, B. Bell, P. Berrisford, et al. *ERA5 hourly data on single levels from 1940 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS)*. DOI:10.24381/cds.adbb2d47 (Accessed on 03-08-2023). 2023 (cit. on pp. 91, 97).

[26] Helmut Hlavacs, Georges Da Costa, and Jean-Marc Pierson. “Energy Consumption of Residential and Professional Switches”. In: *2009 International Conference on Computational Science and Engineering*. Vol. 1. 2009, pp. 240–246 (cit. on p. 24).

[27] Helmut Hlavacs, Georges Da Costa, and Jean-Marc Pierson. “Energy Consumption of Residential and Professional Switches”. In: *2009 International Conference on Computational Science and Engineering*. Vol. 1. 2009, pp. 240–246 (cit. on pp. 55, 64).

[28] IEA. *Data Centres and Data Transmission Networks*. Tech. rep. IEA, Paris, 2022 (cit. on pp. 1, 2).

[29] Ehsan Khodayarseresht, Alireza Shameli-Sendi, Quentin Fournier, and Michel Dagenais. “Energy and carbon-aware initial VM placement in geographically distributed cloud data centers”. In: *Sustainable Computing: Informatics and Systems* 39 (2023), p. 100888 (cit. on pp. 13, 108).

[30] Martijn Koot and Fons Wijnhoven. “Usage impact on data center electricity needs: A system dynamic forecasting model”. In: *Applied Energy* 291 (2021), p. 116798 (cit. on p. 2).

[31] Raquel Landim. “Inflação sofre impacto da crise hídrica, a pior dos últimos 91 anos no Brasil”. In: *CNN Brasil* (Aug. 24, 2021) (cit. on p. 98).

[32] Chris A. Mack. “Fifty Years of Moore’s Law”. In: *IEEE Transactions on Semiconductor Manufacturing* 24.2 (2011), pp. 202–207 (cit. on p. 2).

[33] Seyed Hossein Madaeni, Ramteen Sioshansi, and Paul Denholm. *Comparison of Capacity Value Methods for Photovoltaics in the Western United States. NREL/TP-6A20-54704*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States) (cit. on p. 92).

[34] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. “Recalibrating global data center energy-use estimates”. In: *Science* 367.6481 (2020), pp. 984–986 (cit. on pp. 1, 83).

[35] Meta. *2023 Meta Sustainability Report*. Meta, 2023 (cit. on p. 20).

[36] *Microsoft Datacenters*. `https://datacenters.microsoft.com/globe/explore`. Accessed: 2023-09-25 (cit. on p. 9).

[37] Dawn Nafus, Eve M. Schooler, and Karly Ann Burch. “Carbon-Responsive Computing: Changing the Nexus between Energy and Computing”. In: *Energies* 14.21 (2021) (cit. on pp. 4, 11).

[38] Haider Niaz, Mohammad H. Shams, Mohammadamin Zarei, and J. Jay Liu. “Leveraging renewable oversupply using a chance-constrained optimization approach for a sustainable datacenter and hydrogen refueling station: Case study of California”. In: *Journal of Power Sources* 540 (2022) (cit. on p. 16).

[39] Jean-Marc Pierson, Gwilherm Baudic, Stéphane Caux, et al. “DATAZERO: Datacenter with zero emission and robust management using renewable energy”. In: *IEEE Access* 7 (2019), pp. 103209–103230 (cit. on pp. 17, 51, 84).

[40] Padma Priya, R., and D. Rekha. “Sustainability modelling and green energy optimisation in microgrid powered distributed FogMicroDataCenters in rural area”. In: *Wireless Networks* 27.8 (2021), pp. 5519–5532 (cit. on p. 15).

[41] Md Mustafizur Rahman, Abayomi Olufemi Oni, Eskinder Gemechu, and Amit Kumar. “Assessment of energy storage technologies: A review”. In: *Energy Conversion and Management* 223 (2020), p. 113295 (cit. on p. 50).

[42] Allison Randal. “The Ideal Versus the Real: Revisiting the History of Virtual Machines and Containers”. In: *ACM Comput. Surv.* 53.1 (Feb. 2020) (cit. on p. 8).

[43] V. Dinesh Reddy, Brian Setz, G. Subrahmanya V. R. K. Rao, G. R. Gangadharan, and Marco Aiello. “Metrics for Sustainable Data Centers”. In: *IEEE Transactions on Sustainable Computing* 2.3 (2017), pp. 290–303 (cit. on p. 72).

[44] Charles Reiss, John Wilkes, and Joseph L Hellerstein. “Google cluster-usage traces: format+ schema”. In: *Google Inc., White Paper* (2011) (cit. on p. 36).

[45] National Laboratory of Renewable Energy (NREL). “Life Cycle Greenhouse Gas Emissions from Electricity Generation: Update. NREL/FS-6A50-80580”. In: (Sept. 2021) (cit. on pp. 3, 14, 87).

[46] Marc Richter, Pio Lombardi, Bartlomiej Arendarski, et al. “A vision for energy decarbonization: Planning sustainable tertiary sites as net-zero energy systems”. In: *Energies* 14.17 (2021) (cit. on p. 16).

[47] Hannah Ritchie, Max Roser, and Pablo Rosado. “Energy”. In: *Our World in Data* (2022). `https://ourworldindata.org/energy`. Accessed: 2023-09-26 (cit. on p. 1).

[48] John Roach. “Microsoft’s virtual datacenter grounds “the cloud” in reality”. In: *Microsoft Innovation Stories* (Apr. 20, 2021) (cit. on pp. 9, 64).

[49] José Luis Ruiz Duarte and Neng Fan. “Operations of data centers with onsite renewables considering greenhouse gas emissions”. In: *Sustainable Computing: Informatics and Systems* 40 (2023), p. 100903 (cit. on p. 13).

[50] Enida Sheme, Sébastien Lafond, Dorian Minarolli, Elinda Kajo Meçe, and Simon Holmbacka. “Battery Size Impact in Green Coverage of Datacenters Powered by Renewable Energy: A Latitude Comparison”. In: *Advances in Internet, Data & Web Technologies*. Ed. by Leonard Barolli, Fatos Xhafa, Nadeem Javaid, Evjola Spaho, and Vladi Kolici. Cham: Springer International Publishing, 2018, pp. 548–559 (cit. on p. 17).

[51] W. Short, Daniel Packey, and T. Holt. “A Manual for the Economic Evaluation of Energy Efficiency and Renewable Energy Technologies NREL/TP-462-5173 ON: DE95000211; TRN: 95:003272”. In: *NASA STI/Recon Technical Report N* 95 (Mar. 1995) (cit. on pp. 105, 108).

[52] Junaid Shuja, Abdullah Gani, Shahaboddin Shamshirband, Raja Wasim Ahmad, and Kashif Bilal. “Sustainable cloud data centers: a survey of enabling techniques and technologies”. In: *Renewable and Sustainable Energy Reviews* 62 (2016), pp. 195–214 (cit. on pp. 4, 12).

[53] Jie Song, Peimeng Zhu, Yanfeng Zhang, and Ge Yu. “Versatility or validity: A comprehensive review on simulation of Datacenters powered by Renewable Energy mix”. In: *Future Generation Computer Systems* 136 (2022), pp. 326–341 (cit. on p. 18).

[54] Tyler Stehly and Patrick Duffy. “2021 Cost of Wind Energy Review [Slides]”. In: (Jan. 2023) (cit. on p. 107).

[55] World Business Council for Sustainable Development and World Resources Institute. *The Greenhouse Gas Protocol: A Corporate Accounting and Reporting Standard*. World Business Council for Sustainable Development, 2004 (cit. on p. 19).

[56] Muhammad Tirmazi, Adam Barker, Nan Deng, et al. “Borg: the Next Generation”. In: *EuroSys’20*. Heraklion, Crete, 2020 (cit. on p. 82).

[57] Miguel Vasconcelos, Daniel Cordeiro, Georges Da Costa, et al. *Artifact: Optimal sizing of a globally distributed low carbon cloud federation. doi.org/10.25666/DATAUBFC-2023-02-03*. 2023 (cit. on p. 64).

[58] Pedro Velho, Lucas Schnorr, Henri Casanova, and Arnaud Legrand. “On the Validity of Flow-level TCP Network Models for Grid and Cloud Simulations”. In: *ACM Transactions on Modeling and Computer Simulation* 23.4 (Oct. 2013) (cit. on pp. 30, 35).

[59] Vilayanur Viswanathan, Kendall Mongird, Ryan Franks, et al. *2022 Grid Energy Storage Technology Cost and Performance Assessment*. Tech. rep. PNNL-33283. Pacific Northwest National Laboratory, 2022 (cit. on pp. 105, 108).

[60] Lukas Wagner, Simone Mastroianni, and Andreas Hinsch. “Reverse Manufacturing Enables Perovskite Photovoltaics to Reach the Carbon Footprint Limit of a Glass Substrate”. In: *Joule* 4.4 (2020), pp. 882–901 (cit. on p. 50).

[61] Noelle Walsh. “How Microsoft measures datacenter water and energy use to improve Azure Cloud sustainability”. In: *Microsoft Azure Blog* (Apr. 22, 2022) (cit. on p. 65).

[62] Di Wang, Chuangang Ren, Anand Sivasubramaniam, Bhuvan Urgaonkar, and Hosam Fathy. “Energy Storage in Datacenters: What, Where, and How Much?” In: *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*. SIGMETRICS ’12. London, England, UK, 2012, pp. 187–198 (cit. on pp. 3, 57, 58).

[63] Minxian Xu and Rajkumar Buyya. “Managing renewable energy and carbon footprint in multi-cloud computing environments”. In: *Journal of Parallel and Distributed Computing* 135 (2020), pp. 191–202 (cit. on pp. 12, 15, 38).

[64] Dajun Yue, Fengqi You, and Seth B. Darling. “Domestic and overseas manufacturing scenarios of silicon-based photovoltaics: Life cycle energy and environmental comparative analysis”. In: *Solar Energy* 105 (2014), pp. 669–678 (cit. on p. 66).

Abstract

Falar do consumo de energia
impacto ambiental dos dcs
emissao de carbono
tecnicas carbon aware
carbon responsive
sizing e contention
follow the renewables

Résumé

Falar do consumo de energia
impacto ambiental dos dcs
emissao de carbono
tecnicas carbon aware
carbon responsive
sizing e contention
follow the renewables