Towards Transparent and Trustworthy Cloud Carbon Accounting

Tamar Eilam
IBM T.J.Watson Research Center
NY, USA

ABSTRACT

Climate Change is arguably the biggest challenge that humanity faces today. Multiple trends such as the exponential explosion of data transfer, the emergence and popularity of power intensive workloads such as AI, and the flattening of Moore's law contribute to a rising concern over the increasing carbon footprint cost of digital computation. Any effective strategy to reduce the energy consumption and associated carbon footprint of computations must begin with an accurate and transparent quantification method. However, while most businesses today run a significant portion of their workloads on third party cloud environments, transparent carbon quantification of tenant workloads in cloud environments is lacking. This regretful situation inhibits reliable reporting of Scope 3 Green House Gas (GHG) by cloud users, meaningful comparison of cloud carbon efficiencies, and measurable reduction strategies. In this extended abstract we explain the unique challenges that arise in multi-tenant cloud environments, and propose and discuss an approach, consistent with the GHG Protocol, for cloud carbon footprint quantification. The quantification is a first step towards sustainable cloud environments, that employ dynamic controllers to quantify and reduce the carbon footprint at every layer of the cloud stack.

KEYWORDS

Climate Change, Carbon Footprint, cloud, greenhouse gas, renewable energy

ACM Reference Format:

Tamar Eilam. 2018. Towards Transparent and Trustworthy Cloud Carbon Accounting. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https: //doi.org/10.1145/3501255.3501408

1 INTRODUCTION

Science determined that humanity must reach net zero green house gas emission by the year 2050 in order to keep global warming below 1.5 degrees Celsius relative to the pre-industrial era. [7]

Pressure on enterprises is drastically increasing from consumers, regulators, governments, and employees to reduce the carbon emission associated with the (direct or indirect) operation of their businesses. Investors demand to see bold action from businesses to reduce their carbon footprint. In the current climate (pun intended), green washing, a term coined to describe a behavior of conveying misleading, or inaccurate information about the carbon emission associated with company's products, processes, or services, may lead to dire business consequences.

Compute data centres account for 200 TWh yr [19], or around 1% of total global electricity demand. While data center energy usage has been stable in recent years, multiple trends such as the explosion of data and data transfer, the proliferation of AI (see e.g., [27]), and the flattening of Moore's law (see e.g., [17]) have caused some to raise the alarm on the expected increase in the carbon cost of computing. In particular, while Artificial Intelligence(AI) has revolutionized how businesses address data for rapid actionable insight, it is also associated with a huge carbon cost. AI accuracy continued to improve in the last decade, however training energy cost have increased by a staggering 300,000x [27]. In addition, it has been shown that some large AI training jobs consume the equivalent of 5 cars carbon footprint throughout their life time [28]. Unfortunately, we have also reached the physical limits in semiconductor miniaturization as a viable way to increase efficiency of general purpose computer chips (CMOS) (see e.g., [17]) leading to a golden era of special purpose systems (see e.g., [12]). Due to these trends, it is anticipated by some that under no action data center energy consumption will increase to 12% or more by 2030 [8]. Energy is the main cost factor in running a data center, therefore efficiency has always been a goal. However, due to increased attention to climate change, reducing the *carbon footprint* associated with data centers, also factoring in the source of the energy and its intensity, attempting to maximize use of clean energy such as solar and wind is a new frontier of research.

Cloud computing offers a promise of greater efficiencies due to the multiplexing of a diverse set of workloads on a shared pool of resources, automation, and in particular, auto-scaling, efficient hardware, and bold acquisition of renewable energy. Unfortunately, the inherent nature of cloud computing works against transparency. Businesses running their workloads on most commercial clouds have no visibility necessary to determine the electric power consumed by the resources they use. This discrepancy has been noted in [25] which further claims that cloud providers ought to calculate and make available the energy and carbon footprint associated with every cloud tenant. While Azure [1] and Google Cloud Platform [2] recently provided some tenant carbon footprint calculation tools [3, 4] there are no sufficient details on the methods being used and the boundary conditions. Consequently, it is impossible to use these tools to compare with confidence the efficiency of these cloud environments for different workloads.

In the scientific literature multiple works focus on estimating the energy consumed by servers (e.g., [15, 23]), Virtual Machines (e.g. [18, 31]), containers (e.g. [13, 14, 31]), and applications (e.g., [20]), however, while these works provide valuable insight, methods

Middleware '21 Extended Abstracts, December 6–10, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-9192-4/21/12.
https://doi.org/10.1145/3501255.3501408

and tools, they do not address the complexity of a cloud environment, and they are inconsistent with the GHG Protocol Guidance Document for Cloud Services ([5]).

Challenges that are unique to a cloud environment include multi-tenancy - the fact that multiple customer workloads are sharing a set of physical resources, such as servers and routers. In addition, different granularity of sharing, and levels of abstraction may exist in the same cloud environment. For example, customers may or may not share physical servers or Virtual Machines in multi-tenant services performing computation or data management. The style and units of computation may vary dramatically across services (e.g., transactions, stateless functions, or ML training jobs). Cloud dynamicity such as auto-scaling, and advanced optimization methods, such as dynamic power management complicate the calculation even further.

In this Extended Abstract we discuss the principles described in the GHG protocol as they pertain to cloud services [5, 6] and offer an approach to quantify the carbon footprint of cloud tenants. Many challenges and open questions remain and are discussed here as well.

2 MODEL

A simple model for calculating the carbon footprint of a data center is given by

$$CFP_{DC} = E_{TOTAL} \times CI \tag{1}$$

Where E_{total} is the total amount of energy that the data center consumes (a number that is easily obtained) and CI is the *carbon intensity*, namely the amount of carbon dioxide (or equivalent) released to the atmosphere per a unit of energy (expressed in grams of carbon dioxide equivalent per megajoule (gCO2e/MJ)). Carbon intensity factors in the source of the energy, i.e., the method of production, where renewable energy, such as energy produced from solar or wind power, enjoy a carbon intensity that is close to 0. The carbon intensity associated with a particular data center depends on the electric grid being used among other factors.

The second aspect to consider is non-IT energy. The total data center energy E_{total} equals $E_{IT} + E_{nonIT}$, where,E_{IT} is the energy consumed by the servers and network devices, and E_{nonIT} is the energy used for everything else, including cooling, power distribution, and lighting.

Power usage effectiveness (PUE) is a measure of the efficiency of energy use. It is given by $PUE = E_{total}/E_{IT}$. Obviously, PUE equals 1 is optimal, however, in most traditional data centers we can expect PUE values of 1.5 or even higher. Encouragingly, some hyperscalers claim to have reached near optimal PUE.

Equation 1 can now be written as

$$CFP_{DC} = E_{IT} \times PUE \times CI \tag{2}$$

We posit that in a cloud environment we can and should further break E_{IT} to *payload energy* ($E_{payload}$), used for tenant workloads, and *management energy* (E_{mng}), used for cloud management software such as Software Defined Network (SDN), storage, schedulers, monitoring, metering, and billing. These platform services are typically not used by end-users, nevertheless they consume energy that must be factored in. Additionally, we define $E_{reserve}$ as the energy consumed by all servers that are not currently running any tenant workload, or platform services. Idle power, i.e., the power used by physical hosts when there is no load, can reach 50% of the maximum power draw. This is a significant amount of energy, and obviously must be factored in by any proper accounting method.

As we will discuss later, the GHG Protocol mandates that all data center energy must be split among the tenants for Scope 3 accounting purposes. We therefore propose a new term *cloud power usage effectiveness*, or $CPUE$ that we define as follows:

$$CPUE = E_{total}/E_{payload} = \tag{3}$$

$$(E_{mng} + E_{payload} + E_{reserve} + E_{nonIT})/E_{payload} \tag{4}$$

Clearly, $CPUE \geq PUE$. We argue that $CPUE$ is a valuable measure of a cloud energy efficiency since it factors in not only the overhead of cooling and power distribution, but also the overhead of management services and reserved capacity in running tenant workloads.

Equation 2 can be generalized as follows:

$$CFP_{DC} = E_{payload} \times CPUE \times CI \tag{5}$$

The model above is too coarse grained to account for individual tenant workloads.

Lets formally denote the set of tenants associated with a data center as $T_{DC} = T_1, ..., T_n$. Lets assume for simplicity that this set is static. CFP_i denotes the carbon footprint associated with tenant T_i.

At any time interval, a tenant T_i runs a number of workloads in a datacenter given by $W_i = \{w_i^1, ...w_i^k\}$. Every workload is a distributed application that may run on multiple VMs and servers. Different components of the application may communicate, consuming additional energy from routers and switches along the communication path. The workload's resource consumption, is the summation of energy consumed by resources used by the workload, including, CPU, Memory, Disk, IO, of all devices used by the workload. This is a dynamic set. Formally, Let $R = \{r_1, ...r_t\}$ be the set of IT resources in the data center including servers, routers, and switches. Let $R_w \subset R$ be the subset of resources used by workload w. We assume workloads are not shared across tenants, namely $W_i \cap W_j = \emptyset$ iff $i \neq j$. However, workloads may, and typically will, share resources, such as servers and network devices. Namely, $R_{w_i} \cap R_{w_k} \neq \emptyset$. If a resource is shared then energy consumed by this resource must be split among its users. Before we proceed to formally define the *cloud carbon accounting problem* lets spend some time to discuss the accounting principles defined by the GHG Protocol.

2.1 The GHG Protocol

The Greenhouse Gas (GHG) Protocol [5] establishes comprehensive global standardised frameworks to measure green house gas emissions. It is the most wildly used GHG accounting standard. Either the GHG Protocol or other organizations develop additional guidance that build upon it. In particular, Carbon Trust, Global e-Sustainability Initiative (GeSI) [6] developed ICT Sector Guidance for the calculation of life-cycle GHG emissions for ICT. Chapter 4 in this document focuses on Cloud Computing and Data Center Services with the proclaimed goal of defining how to allocate the GHG emissions of a data center to its various services and clients. Our approach is consistent with this guidance.

In simple terms, the guidance mandates that *the totality of energy of the data center must be completely split across the tenants of the data center.* Note that for these tenants this energy is considered Scope 3, i.e., indirect use of energy[1]. We term this principle *the completeness property.* In addition, the guidance defines several principles governing the method to split the energy across the users of a shared resource or service. In particular, the guidance distinguishes between *fixed* and *dynamic energy.* Lastly, the guidance encourages energy calculation methods specific to the actual service and style of computation.

We summarize the principles as follows.

(1) All the data center emission must be allocated to the individual services (Completeness Principle).
(2) All IT devices must be allocated across the services.
    (a) If a device supports more than one service, divide its allocation using a consistent method
    (b) IT equipment that is used to manage workloads for a service should also be allocated to that service and its users.
    (c) IT equipment that serves as reserve capacity for a service, should also be allocated to that service.
    (d) Every device is fully allocated either directly or indirectly to users.
(3) The selected allocation method should seek to separate the fixed and variable emissions. The intent is to allocate the fixed emissions based on the provisioned capacity and the variable emissions based on the energy consumption of each platform, customer, or device.
(4) Every service defines functional unit(s) which are being used to partition the energy. Examples include: Transactions, Number and size of volumes of storage, Number and size of VMs and CPU utilization

The idea of 'fixed' vs. 'dynamic' energy is intended to separate the portion of the energy that does not directly correlate with user usage from the portion that correlates directly with user usage. For example, take a single server that runs multiple VMs on behalf of multiple tenants. Server idle power (where no workload runs) can be as high as 50% of the total power draw of the server. An allocation method of energy across the tenant VMs must treat differently the server idle energy from the usage-based energy. Let idle energy be denoted as E_{IDLE}. Note this is a fixed number for a given server. $E_t - E_{IDLE}$ is the usage-based energy during a time period t. A common fair allocation method of energy to VMs will be to split E_{IDLE} across the VMs in proportion to their size, and split $E_t - E_{IDLE}$ based on usage (weighted by size). Obviously, such an allocation method is necessary since there is no way to *directly* measure power used by each VM. For the usage based portion, Machine Learning is often used to identify a set of counters (such as vCPU utilization) and an energy prediction method based on the counters. In the simplest case, vCPU is the only counter and linear regression determines the linear fitting function. The guidance fall short in considering different power states of servers. We refine and generalize the method to account for those (deferred to future publications).

[1]For definitions of Scope 1, 2 and 3 the reader is referred to [5].

We are now ready to describe our approach to address the cloud carbon accounting challenge.

3 APPROACH

The *Cloud Carbon Accounting (CCA) Problem* is concerned with attributing energy and carbon footprint to each one of the tenants in a cloud environment. Note that here we restrict the discussion to just the operations phase. Embodied emission of servers is outside the scope of this work. For consistency with the guidance, the completeness property must be satisfied, i.e., $CFP_{DC} = \sum_1^n CFP_i$, where CFP_{DC} is the total carbon footprint associated with the data center, and CFP_i is the carbon footprint attributed to a single tenant.

Our approach is based on the concept of *an energy attribution graph.* The energy attribution graph is direct and dynamic. A snapshot is depicted in Figure 1. The leaves of the direct graph are the IT devices. An IT device is defined as a unit that is associated with direct power meter (example: server, network device, storage device). Each such device is associated with an allocation functions to split the power it consumes among its users (where a user can be yet another device, a service, or an end user). In some advanced cases more than one allocation function is defined, factoring in, e.g., the characteristics of the workload. Other logical constructs, may exist and participate in the allocation process, such as a VM, a container, a function, a transaction, a storage volume, and an Object (in an object store), a volume, a block, etc. Such logical constructs are assigned a *calculated* energy allocation. Every IT device is allocated to a service, a user, or a set of users and/or services. This is represented in the figure as an edge directed from the device out. Services are associated with a set of in-edges, and out-edges. There is an in-edge for every device or service that *this* service uses. For example, a server that is used in an IaaS service (See example in Section 3.1) will have an edge that is directed from it to the IaaS Service. Out-edges connect a service to all its users (other services or tenants). Like with devices, every service defines an allocation function to split its own energy. Note that a service is a composite logical construct; it does not consume energy directly. Example of services include a VM provisioning service, a Functions-as-a-Service, or an Object Storage service. Note that in many cloud environments the later two services will use the VM provisioning service. The energy consumed by a service is calculated as the sum of all allocations to it by the devices or other services it uses. The energy that is consumed by a service (calculated) must be allocated *in its entirety* across the users of the service (Completeness Principle). In other words, assume labels on edges correspond to energy units. For each node in the graph, the sum of energy units on its in-edges must be equal to the sum of units on its out-edges. The sum of labels on in-going edges for a tenant i represent the E^i_{IT} attributed to that tenant. To get the total energy per tenant one still have to multiply E^i_{IT} by the PUE (which is a global number for the entire data center). CFP_i is given by $E^i_{IT} \times PUE \times CI$. In advanced cases, CI is a modeled as a time series. The method can be easily generalized to account for a cloud comprising multiple data centers.

The service (or device) energy allocation method must be designed carefully. For example, service reserved capacity (which is not used by any user at a given time period) must be factored in. The overhead of management nodes must be factored in. Tenant

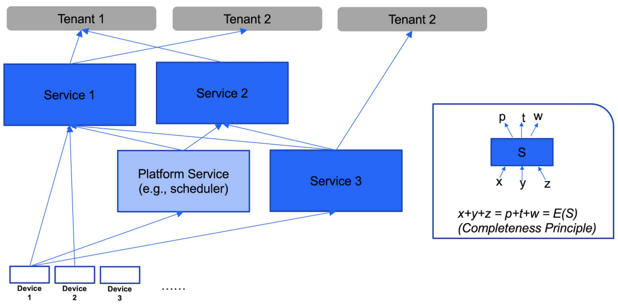

Figure 1: Energy Attribution Graph

Service Level Agreements (SLA) and different key usage metrics will play role in how the energy allocation is calculated.

3.1 Example: Infrastructure-as-Service (IaaS)

Consider an IaaS service S that provisions Virtual Machines (VMs) on a pool of shared servers on behalf of a set of users (tenants). For simplicity, let's assume that VM types only vary by the number of cores assigned to them. Let $S(V)$ be the number of cores assigned to a VM V. Further, the service maintains two server pools: Active pool, where any server in the pool hosts at least one VM, and StandBy pool, where servers are not hosting any VMs, but are ready to serve new requests. Further, the service management software, such as a VM scheduler, runs on a separate set of servers. The following classes of energy can be defined:

- E_{mng} = The total energy consumed by the pool of servers running the service management software.
- $E_{standby}$ = The total energy consumed by all servers in the standby pool (note that this is all idle power).
- E_{idle} = The total idle power of all servers in the active pool (note, the E_{idle} may differ across servers based on, e.g., their type, but it is a fixed number for a given server).
- $E^M_{use} = E^M - E^M_{idle}$, where for a server M, E^M is the measured energy that M consumed, and E^M_{idle} is the energy that M consumes when in idle state (constant per server)

Now for usage metrics let us define

- K_s = size of the VM (i.e., number of vCPUs in this example)
- K_u = the average utilization of the VM (averaged over all its vCPUs).

The allocation method $f(V_i)$ is defined as follows for a VM V_i running on a server M, with usage energy of E^M_{use}.

$$f(V_i) = \tag{6}$$

$$\frac{K_s(V_i)}{\sum_{all\ VMs} K_s(V_j)} \times (E_{mng} + E_{standby} + E_{idle}) + \tag{7}$$

$$\frac{K_s(V_i)K_u(V_i)}{\sum_{VM \in M} K_s(V_j)K_u(V_j)} \times E^M_{use} \tag{8}$$

Note that the utilization of the VM relative to other VMs running on the same server M is used in order to split the usage portion of the energy E^M_{use} of that server across the VMs running on it ($VM \in M$), but the rest of the energy of the service ($E_{mng} + E_{standby} + E_{idle}$) is split across all VMs based on their size. Considering this IaaS service only, the completeness property is satisfied, as it is easily verifiable that $\sum_{all\ VMs} f(V_i) = E_{IT}(S)$.

To calculate a tenant energy we add up the energy associated with all VMs provisioned on its behalf. I.e., $E^i_{IT} = \sum_{V \in T_i} f(V)$, and $CFP_i = E^i_{IT} \times PUE \times CI$. An alternative calculation will leave E_{mng} and $E_{standby}$ out of the calculation of $f(V_i)$, to calculate a strict $E^i_{payload}$ and then multiple by the Cloud Power Usage Effectiveness (CPUE) to essentially get the same result $CFP_i = E^i_{payload} \times CPUE \times CI$.

Note that in more realistic use cases, we may want to allocate the reserve capacity based on tenant SLAs, specifically, the level of redundancy they require. In general, SLAs play a key role here. There exists a trade off between performance, availability and carbon footprint. Carbon efficiency should be estimated with respect to the non-functional requirements expected.

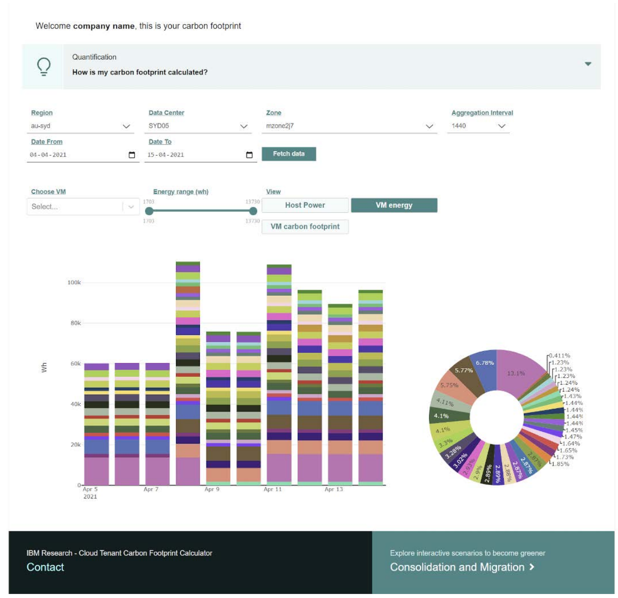

Figure 2: UI of CARE

3.1.1 *Implementation.*

To date, we have implemented the method focusing first on an IaaS service to dynamically provision VMs on behalf of tenants on a shared pool of resources. The implementation collects the following time series: VM provisioning requests including their sizes, placement of VMs on servers, respective utilization, and server power consumption. The implementation calculates for each VM the energy and carbon footprint with choice of sampling frequency and averaging intervals. We are actively engaging with a selected set of users to evaluate the method. A screenshot of the tool showing the calculations per VM is shown below.

4 DISCUSSION

In this extended abstract, we have discussed the importance of a transparent method to quantify carbon footprint per tenant in modern cloud environments. Such a method can be used for reporting of carbon footprint by business users, for comparison of the efficiencies obtained by different cloud environments, and as a baseline for further optimization. We proposed an approach that is consistent with the GHG protocol ICT guidance document [6]. Our approach allows for varying degrees of resources sharing, and abstraction layers (e.g., IaaS, PaaS). A lot more research is needed to design accurate methods to fairly split the energy of a shared resource or service. Related scientific works, focused on the energy consumption of Servers (e.g., [23]), VMs (e.g., [18]), containers (e.g., [13, 14, 31]), or applications (e.g., [13, 20]), but did not factor in consistency with the GHG protocol and the special characteristics of modern dynamic and multi-tenant cloud environments.

A large body of scientific work is also focused on reduction of energy and carbon footprint of computation. This work can be roughly divided into the following categories: (1) design of specialized hardware (e.g., [12]); (2) energy aware cloud resource management (e.g., [9–11, 16, 24, 26, 26, 29, 30, 30]); (3) AI efficiency (e.g., [21]); and, (4) data center efficiency focusing on cooling and power distribution optimization (e.g., [22, 32]).

We posit that to get to the vision of a carbon performant (sustainable) cloud one must employ quantification techniques first to establish a baseline, and a set of *controllers* to optimize the carbon footprint at every level of the cloud stack: facility, systems, software, and management systems. Our team is working on such as architecture and controllers for power management, scheduling and dispatching, and vertical scaling. As more workloads move to the cloud and with the looming threat of climate change this goal is more important than ever.

5 ACKNOWLEDGEMENTS

The author is in debt to numerous colleagues that contributed to this work, including Umamaheswari Devi, Eun K Lee, Arun Vishnawath, Ram Kolluri, Kalyan Dasgupta, Aanchal Goyal, Kedar Kulkarni, Jagabondhu Hazra (Jago), Bruno Michel, Asser Tantawi, and Alaa Youssef.

REFERENCES

[1] [n. d.]. https://azure.microsoft.com/en-us/
[2] [n. d.]. https://cloud.google.com/gcp
[3] [n. d.]. https://azure.microsoft.com/en-us/
[4] [n. d.]. https://techcrunch.com/2021/10/12/google-cloud-will-now-show-its-users-their-carbon-footprint-in-the-cloud/
[5] [n. d.]. ICT Sector Guidance built on the GHG Protocol Product Life Cycle Accounting and Reporting Standard. *Global e-sustainability Institute* ([n. d.]). https://ghgprotocol.org/sites/default/files/GHGP-ICTSG%20-%20ALL%20Chapters.pdf
[6] [n. d.]. ICT Sector Guidance for the calculation of life-cycle GHG emissions for ICT. https://www.gesi.org/public/index.php/research/ict-sector-guidance-built-on-the-ghg-protocol-product-life-cycle-accounting-and-reporting-standard
[7] [n. d.]. Net Zero by 2050 - A Roadmap for the Global Energy Secto. *International Energy Agency (IEA)* ([n. d.]). https://iea.blob.core.windows.net/assets/4719e321-6d3d-41a2-bd6b-461ad2f850a8/NetZeroby2050-ARoadmapfortheGlobalEnergySector.pdf
[8] Anders S. G. Andrae and Tomas Edler. 2015. On Global Electricity Usage of Communication Technology: Trends to 2030. *Challenges* 6, 1 (2015), 117–157. https://doi.org/10.3390/challe6010117
[9] S. Azizi, Zandsalimi, M., and D. Li. 2020. An energy-efficient algorithm for virtual machine placement optimization in cloud data centers.
[10] Josep L. Berral, Chen Wang, and Alaa Youssef. 2020. AI4DL: Mining Behaviors of Deep Learning Workloads for Resource Management. In *12th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 20)*. USENIX Association. https://www.usenix.org/conference/hotcloud20/presentation/berral
[11] David Buchaca, Josep LLuis Berral, Chen Wang, and Alaa Youssef. 2020. Proactive Container Auto-scaling for Cloud Native Machine Learning Services. In *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*. 475–479. https://doi.org/10.1109/CLOUD49709.2020.00070
[12] Luca P. Carloni. 2020. Scalable Open-Source System-on-Chip Design. In *Proceedings of the IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*.
[13] Silva de Souza W, Iranfar A, Bráulio A, Zapater M, Xavier de Souza S, Olcoz K, and Atienza D. Containergy. [n. d.]. A Container-Based Energy and Performance Profiling Tool for Next Generation Workloads.
[14] Guillaume Fieni, Romain Rouvoy, and Lionel Seinturier. 2020. SmartWatts: Self-Calibrating Software-Defined Power Meter for Containers.
[15] R. Rouvoy G. Fieni and L. Seiturier. [n. d.]. SELFWATTS: On-the-fly Selection of Performance Events to Optimize Software-defined Power Meters.
[16] Niloofar Gholipour, Niloufar Shoeibi, and Ehsan Arianyan. 2021. An Energy-Aware Dynamic Resource Management Technique Using Deep Q-Learning Algorithm and Joint VM and Container Consolidation Approach for Green Computing in Cloud Data Centers. In *Distributed Computing and Artificial Intelligence, Special Sessions, 17th International Conference*, Sara Rodríguez González, Alfonso González-Briones, Arkadiusz Gola, George Katranas, Michela Ricca, Roussanka Loukanova, and Javier Prieto (Eds.). Springer International Publishing, Cham, 227–233.
[17] M. Horowitz. [n. d.]. computing's energy problem.
[18] Weixiang Jiang, Fangming Liu, Guoming Tang, Kui Wu, and Hai Jin. 2017. Virtual Machine Power Accounting with Shapley Value. *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (2017), 1683–1693.
[19] N. Jones. [n. d.]. How to stop data centres from gobbling up the world's electricity.
[20] Sandro Kreten, Achim Guldner, and Stefan Naumann. 2018. An Analysis of the Energy Consumption Behavior of Scaled, Containerized Web Apps. *Sustainability* 10, 8 (2018). https://doi.org/10.3390/su10082710
[21] Nicola Lenherr, René Pawlitzek, and Bruno Michel. 2021. New universal sustainability metrics to assess edge intelligence. *Sustainable Computing: Informatics and Systems* 31 (2021), 100580. https://doi.org/10.1016/j.suscom.2021.100580
[22] Y. Li, Y. Wen, D. Tao, and K. Guan. 2020. Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning.
[23] W. Lin, Zhang Y., L. Wu W.and Fong, S.and He, and J. Chang. [n. d.]. An adaptive workload-aware power consumption measuring method for servers in cloud data centers.
[24] Rajat Mehrotra, Abhishek Dubey, Sherif Abdelwahed, and Asser N Tantawi. 2012. Power-Aware Modeling and Autonomic Management Framework for Distributed Computing Systems. *Handbook of energy-aware and green computing* 2 (2012), 621–648.
[25] D. Mytton. [n. d.]. Hiding greenhouse gas emissions in the cloud. *Nature Climate Change 10, 701 (2020)* ([n. d.]). https://doi.org/10.1038/s41558-020-0837-6
[26] T. Renugadevi, K. Geetha, K. Muthukumar, and Zong Woo Geem. 2020. Optimized Energy Cost and Carbon Emission-Aware Virtual Machine Allocation in Sustainable Data Centers. *Sustainability* 12, 16 (2020). https://www.mdpi.com/2071-1050/12/16/6383
[27] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM* 63, 12 (Nov. 2020), 54–63. https://doi.org/10.1145/3381831
[28] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. arXiv:1906.02243 [cs.CL]
[29] Marc Taubenblatt, Pavlos Maniotis, and Asser Tantawi. 2022. Optics enabled networks and architectures for data center cost and power efficiency. *J. Opt. Commun. Netw.* 14, 1 (Jan 2022), A41–A49. https://doi.org/10.1364/JOCN.440205
[30] M. Xu, A. N. Toosi, and R. Buyya. [n. d.]. A Self-adaptive Approach for Managing Applications and Harnessing Renewable Energy for Sustainable Cloud Computing.
[31] X. Zhang, Z. Shen, B. Xia, Z. Liu, and Y. Li. [n. d.]. Estimating Power Consumption of Containers and Virtual Machines in Data Centers.
[32] Severin Zimmermann, Ingmar Meijer, Manish K. Tiwari, Stephan Paredes, Bruno Michel, and Dimos Poulikakos. 2012. Aquasar: A hot water cooled data center with direct energy reuse. *Energy* 43, 1 (2012), 237–245. https://doi.org/10.1016/j.energy.2012.04.037 2nd International Meeting on Cleaner Combustion (CM0901-Detailed Chemical Models for Cleaner Combustion).