Towards Game-Theoretic Approaches to Attributing Carbon in Cloud Data Centers

Leo Han
lxh4@cornell.edu
Cornell Tech
New York, New York, USA

Jash Kakadia
jkakadia@seas.upenn.edu
University of Pennsylvania
Philadelphia, Pennsylvania, USA

Benjamin C. Lee
leebcc@seas.upenn.edu
University of Pennsylvania
Philadelphia, Pennsylvania, USA

Udit Gupta
ugupta@cornell.edu
Cornell Tech
New York, New York, USA

ABSTRACT

Data centers are becoming an ever greater threat to our climate: their energy usage alone constituted 0.6% of global greenhouse gas emissions (GHG) emissions in 2020. Recent studies have shown that embodied GHG emissions of data centers are comparable to that from their energy usage. As cloud customers increasingly seek to better understand their carbon footprint, public cloud providers have begun providing tools to attribute the carbon costs of data centers to users. Many open-source carbon attribution and accounting tools have also emerged in the last few years to help users measure the carbon footprint of their workloads. However, existing attribution methodologies lack fairness guarantees and are overly simple and coarse-grained. This paper presents a game theoretic framework for fair and comprehensive operational and embodied carbon attribution within the scope of a single node. We demonstrate the attribution framework on a real cloud server.

CCS CONCEPTS

• **Computer systems organization** → **Cloud computing**; • **Hardware** → **Impact on the environment**.

KEYWORDS

Sustainable Computing, Cloud, Data centers, Carbon Attribution

ACM Reference Format:
Leo Han, Jash Kakadia, Benjamin C. Lee, and Udit Gupta. 2024. Towards Game-Theoretic Approaches to Attributing Carbon in Cloud Data Centers. In *Proceedings of 3rd Workshop on Sustainable Computer Systems (HotCarbon'24)*. ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

The information and computing technology (ICT) industry accounted for a significant 2.1% to 3.9% of global greenhouse gas (GHG) emissions in 2021 with emissions growing annually since [15]. A significant portion of the ICT industry's carbon footprint comes from data centers, whose energy usage accounted for 0.6% of global GHG emissions in 2020 [19] and embodied carbon footprint accounts for a similar portion [24]. Driven by the rapid growth of cloud services and large machine learning models, data centers are expected to grow roughly 10% year-on-year until 2030 [30].

As computing's impact on energy usage and carbon emissions grows ever larger, data center operators seek to better understand the carbon footprint of individual users and jobs. This understanding could drive sustainability strategies for the data center provider's internal operations as well as permit better carbon accounting and mitigation for cloud customers. The largest cloud operators – Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) – have developed accounting tools with which users can estimate the carbon impact of their cloud computation [6, 33, 41]. Open-source tools, such as Cloud Carbon Footprint [48] and CodeCarbon [40], have been created to help users understand the climate impact of their software applications [27, 28, 37].

Existing carbon accounting models do not fully capture nuances in colocated data center jobs. For instance, some workloads may use particular hardware resources more intensely than others such that time on the server does not precisely reflect carbon costs. Some workloads may interfere and lengthen job completion times for others such that carbon costs depend on colocation decisions. When jobs colocate, the carbon accounting framework must fairly attribute the server's fixed and variable costs to individual jobs. Fixed costs include carbon associated with shared idle power as well as shared hardware components (e.g., the printed circuit board, chassis, etc.). Variable costs include carbon associated with a job's specific usage of individual hardware components. Note that fixed and variable costs exist for both operational carbon, which depends on the server's electricity usage, and embodied carbon, which depends on the server's construction and life cycle.

We explore the use of the Shapley value [43], a game-theoretic concept, for fairly attributing data centers' carbon to users' jobs. The Shapley value guarantees several fairness properties and has been proven useful in many applications in economics [23, 25, 35] and computer systems [11, 14, 21, 26]. Past work has looked at using the Shapley value for attributing power amongst colocated workloads [11, 21], but no existing work looks at using the Shapley value for fair attribution of both operational and embodied carbon.

The key contributions of this work include:

HotCarbon'24, July 9, 2024, Santa Cruz, CA

- A Shapley value based model for fairly attributing a server's operational and embodied carbon using power and resource utilization telemetry.
- A set of resource-specific embodied carbon attribution models for the CPU, DRAM, mainboard, peripherals, storage, and the power supply unit.
- A demonstration of the fair attribution model on a Dell R650 dual-socket server on CloudLab [12] using CloudSuite 4.0 [13] workloads, showing up to a 43% difference in workload carbon attribution compared to a baseline energy-based attribution method.
- An exploration of challenges and a roadmap for future directions in fairly attributing data center carbon using game theory.

2 LIMITATIONS OF EXISTING CLOUD CARBON ATTRIBUTION MODELS

The major public cloud providers and hyperscalers — Microsoft Azure, Google Cloud Platform (GCP), and Amazon Web Services (AWS) — have developed their own carbon attribution tools so that users can estimate their cloud carbon footprint [6, 33, 41]. There also exist many open-source tools [40, 44, 45] and academic works [1, 5, 16, 18, 39, 46, 49] that support a mix of operational carbon attribution, embodied carbon attribution, and energy attribution.

2.1 Cloud carbon attribution models

Azure attributes operational carbon by attributing each server's energy to each customer based on their resource utilization [31, 34] and then converting that to carbon footprint via the grid's carbon intensity. Azure attributes embodied carbon to each customer proportional to the billing cost of services rendered to that customer [32, 34].

GCP attributes operational carbon emissions by attributing each internal service's energy use to customers based on billing cost and then converting that to carbon footprint via the grid carbon intensity [7]. GCP attributes embodied carbon emissions to each customer proportional to the energy use attributed to that customer [7].

AWS attributes operational carbon emissions to customers using region-specific carbon intensity [42]. Unfortunately, AWS does not publicize further details on its energy attribution and carbon attribution methodology and does not provide embodied carbon attribution to its customers.

Open-source tools such as Cloud Carbon Footprint [48], CodeCarbon [40], Green Metrics Tool [44] and models such as carbond [39], and Westerhof et al. [49] attribute operational carbon on a per-node granularity. CodeCarbon and Green Metrics Tool do not attribute embodied carbon. Cloud Carbon Footprint and carbond attributes a server's embodied carbon footprint proportional to the customer's resource utilization quantity and time. Westerhof et al. attributes embodied carbon proportional to a user's energy attribution.

2.2 Limitations in existing methods

Existing tools and frameworks attribute operational carbon at workload-granularity with detailed energy accounting and attribution via hardware power and resource utilization telemetry. However, the tools either lack embodied carbon attribution entirely [40, 41, 44] or attribute simply based on billing cost, energy usage, or resource utilization quantity and time [6, 33, 39, 48, 49]. Moreover, existing open-source [40, 44, 48] and academic [39, 49] models do not attribute operational carbon between colocated workloads on the same node. We list below several additional shortcomings of existing frameworks that we seek to address via our Shapley value based carbon attribution method.

Resource-dependent carbon footprint. Embodied carbon attribution methods that group together the embodied carbon of all components and then attribute to customers via billing cost or energy use ignore the varying embodied carbon footprints of different components. Carbon accounting should be treated separately from monetary accounting and energy accounting. Billing cost is based on economic cost and pricing policies and is not representative of embodied carbon. The CPUs in our case study system cost 10766 USD and 13.60 $kgCO_2e$ (791.62 USD/$kgCO_2e$) whereas the RAM costs 763.68 USD and 178.00 $kgCO_2e$ (4.29 USD/$kgCO_2e$) [4, 20]. If data center providers set billing costs for resource usage roughly proportional to resource monetary costs, then a billing cost based approach for embodied carbon attribution will over-attribute the embodied costs of CPU usage by more than two orders of magnitude compared to RAM usage. Energy use is also not representative of embodied carbon since different components can have drastically different power to embodied carbon ratios. At an estimated 5 W TDP per module [3], the DRAM in our case study system (table 1) has a TDP to embodied carbon ratio of 1 W:2.225 $kgCO_2e$. The two CPUs in the same system have a ratio of 1 W: 0.0272 $kgCO_2e$, a difference of two orders of magnitude versus DRAM. Embodied carbon attribution should thus be done on a per-component granularity based on per-component carbon profiles.

Colocation effects are ignored. Colocated workloads on a single node can interfere with each other in complex, non-linear ways [26, 29]. Interference from colocated workloads may cause longer execution times and thus greater resource utilization. Thus, attribution methodologies that only use per workload resource utilization ignore such interference effects and can unfairly attribute workload carbon emissions without accounting for external influences from colocated workloads.

Dynamic resource demand is ignored. Resources are provisioned to accommodate peak demand in data centers; this worst-case provisioning directly impacts the embodied carbon of data centers. Data center resource utilization can exhibit strong diurnal and other patterns [10], with low demand periods requiring much fewer resources than peak demand periods. Intuitively, embodied carbon attribution should account for the dynamics of resource demand: workloads that use resources during peak demand contribute more to the overall embodied carbon footprint. All existing embodied carbon attribution models fail to address this relationship between resource demand and aggregate embodied carbon footprint.

3 SHAPLEY VALUE CARBON ATTRIBUTION

The cloud data center is a system setting in which resources are inherently shared. The carbon impact of a data center depends on of its population of users and data center operator's decisions. For this complex system, we seek to fairly divide and attribute the data center's carbon costs to individual users and the data center operator. Fairness matters because users are strategic and they can use another cloud service provider if they feel they are not being attributed the correct amount of carbon. The Shapley value [43] is a well-known game theoretic solution to complex, fair division problems [11, 14, 21, 23, 25, 26, 35].

3.1 Properties

Using the Shapley value to attribute the data center's shared carbon costs has four key desirable properties.

Null Player. The Shapley value is zero for a workload that has no effect on carbon.

Symmetry. Workloads in the same equivalence class (*e.g.*, with the same computational intensities and resource utilization profiles) are attributed the same amount of carbon.

Efficiency. The carbon footprint is fully attributed across all workloads and no carbon remains unattributed. Carbon is neither over- nor under-counted during attribution.

Linearity. Shapley values when attributing carbon for smaller sub-populations of workloads sum to the Shapley value when attributing carbon for the overall population of workloads. Linearity allows us to break down the problem of attributing data center carbon to each cloud user into smaller attribution sub-problems (*e.g.*, at rack or cluster scale). In this paper, we start with the smallest sub-problem: attributing carbon within a single server node.

3.2 Formula

The Shapley value examines all the possible ways to construct a set of colocated values by adding one workload at a time. In other words, it examines all workload permutations. In each permutation, a workload makes a marginal contribution to carbon costs by increasing the use of server hardware and power. A workload's Shapley value is the average of these marginal contributions across all permutations. Given the set N of n workloads with the carbon footprint function v, the formula for the Shapley value φ for workload $i \in N$ is:

$$\varphi_i(v) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S)) \tag{1}$$

Although the Shapley value calculation scales with the square of the number of workloads, it can be a viable solution for carbon attribution at the single-node scope. Within a single node, the physical resources limit how many workloads can colocate, thus limiting the set size of possible workloads. Moreover, this assumption can be more reasonably made for smaller data centers with a limited number of internal users and workloads.

3.3 Operational Carbon Attribution

We first attribute power using the Shapley value formula. We then find operational carbon from attributed power usage and the grid's carbon intensity. We directly apply Shapley value attribution to power in a single server node. Here, $p(N)$ is a function that maps the set of colocated workloads N to system power P. With a small set of possible workloads, it is possible to profile all possible workload combinations, finding $p(S)$ for all $S \subseteq N$. Using these profiles, we calculate the Shapley value for power i-th workload at each moment in time:

$$P_i(t) = \varphi_i(p, t) = \frac{1}{n} \sum_{S \subseteq N(t) \setminus \{i\}} \binom{n-1}{|S|}^{-1} (p(S \cup \{i\}) - p(S)) \tag{2}$$

where $N(t)$ is the set of all workloads running at time t.

Finally, we calculate the i-th workload's operational carbon footprint CF_i based on the grid's carbon intensity $ci(t)$ in gCO_2e/J, which varies across time:

$$CF_{op_i} = \int P_i(t) \times ci(t) dt \tag{3}$$

3.4 Embodied Carbon Attribution

We frame the problem of embodied carbon attribution as the problem of demand-driven supply provisioning for hardware. At provisioning time, the decision is made as to what hardware is needed to meet future (expected peak) resource demands. For example, a 48-core CPU may be chosen because the cloud provider thinks that the workload will require no more than 48 cores simultaneously.

If the workload only uses 42 cores from the 48-core CPU, we could have provisioned fewer cores without affecting performance and the original 48-core CPU was *over-provisioned.* The over-provisioned hardware incurs additional carbon costs:

$$CF_{overprovisioning} = CF_{CPU}(48) - CF_{CPU}(42),$$

which could be attributed to the data center operator, which made the decision to over-provision the hardware. The carbon attribution problem is no longer attributing the carbon for the entire CPU $CF_{CPU}(48)$ to users but rather attributing $CF_{CPU}(48) - CF_{overprovisioining} = CF_{CPU}(42)$ to users. We can decompose the aggregate time-varying resource demand $Q_{demand}(t)$ as the sum of each individual user's resource demand:

$$Q_{demand}(t) = \sum_i Q_i(t) \tag{4}$$

We can determine peak demand $Q_{peak} = max(Q_{demand}(t))$ and thus embodied carbon footprint CF as a function of the set S of workloads. Finally, we can apply the Shapley value calculation to attribute embodied carbon to a set N of n colocated workloads.

$$CF_{emb_i} = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|\ S\ |}^{-1} (CF_{emb}(S \cup \{i\}) - CF_{emb}(S)) \tag{5}$$

Applying this embodied carbon attribution methodology requires modeling each hardware type's embodied carbon footprint as a function of its quantity. For example, we model a CPU's embodied carbon footprint as a function of the number of cores and DRAM's embodied carbon footprint as a function of memory capacity.

Our approach attributes only the resources demanded by the user to the user. For example, a workload that uses no storage will not be attributed any of the embodied carbon from SSDs or HDDs in the node. Conventional methods, however, attribute the server's embodied carbon footprint as a whole and can unfairly over-attribute carbon to workloads for resources that they did not request.

CPU Embodied Carbon. We define the resource quantity Q as the number of CPU cores reserved by active users/workloads. We use a linear model to model the embodied carbon footprint of a CPU as a function of the number of cores.

First, we model a single CPU's embodied carbon as a linear function of chip area [17]. A fixed carbon footprint $CF_{packaging}$ is associated with the packaging of a single CPU. The embodied carbon of manufacturing the silicon chip is equal to the carbon per unit area of silicon, λ, multiplied by the silicon area, A.

$$CF_{CPU}(A) = CF_{packaging} + \lambda \times A \quad (6)$$

We express CPU embodied carbon as a function of the number of cores by separating the chip area into a constant non-core area, A_{const}, and a variable core area. A_{const} includes I/O, memory, system management, etc. The variable core area is equal to the area of each core, A_{core}, multiplied by the number of cores, Q.

$$\begin{aligned} CF_{CPU}(Q) &= (CF_{packaging} + \lambda \times A_{const}) + (\lambda \times A_{core}) \times Q \\ &= \alpha + \beta \times Q \end{aligned} \quad (7)$$

α represents the portion of embodied carbon that does not scale with the number of CPU cores. This includes embodied carbon costs associated with packaging and with non-core chip area. β is the embodied carbon cost of adding each core, modeled as the embodied carbon cost of the associated silicon area.

For multi-socket systems, we need to consider having multiple CPUs. When modeling embodied carbon as a function of the number of cores, we assume that the number of CPUs is the minimum needed to supply the number of cores. If each CPU has Q_{perCPU} cores, the number of CPUs needed is $\lceil \frac{Q}{Q_{perCPU}} \rceil$.

$$CF_{CPU}(Q) = \alpha \times \lceil \frac{Q}{Q_{perCPU}} \rceil + \beta \times Q \quad (8)$$

DRAM Embodied Carbon. We break down total system memory into individual modules. As with multi-socket CPUs, we view the combined memory capacity from all modules as a pool of resources. When modeling DRAM's embodied carbon as a function of memory capacity, we assume only the minimum number of DRAM modules is provisioned for that capacity. If each module has $M_{perModule}$ amount of memory, then the number of DRAM modules needed is $\lceil \frac{M}{M_{perModule}} \rceil$.

We separate out the carbon for each DRAM module into two parts. The first part comprises the DRAM chips. The carbon of a DRAM chip includes carbon for packaging $CF_{packaging}$ and for the silicon die CF_{die}. For the carbon of the DRAM chips, we assume carbon is directly proportional to memory capacity at a rate of μ. The second part includes everything else associated with the DRAM module, including the PCB, the RCD chip, any other components. We assume the carbon footprint of these components, κ, is constant per module.

$$CF_{MEM}(M) = \kappa \times \lceil \frac{M}{M_{perModule}} \rceil + \mu \times M \quad (9)$$

Mainboard Embodied Carbon. We split the embodied carbon footprint of the mainboard (also known as the motherboard or baseboard) into a fixed component and a component that scales proportionally with power. The portion that scales proportionally with power is the mainboard's power delivery network. Proportional scaling is a reasonable assumption since most server mainboard power delivery networks are multi-phase, switched-mode supplies

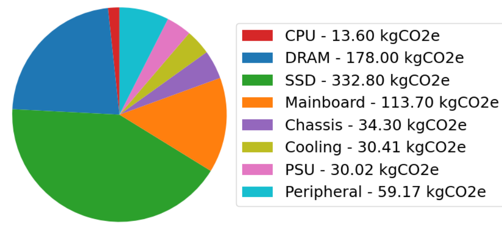


Figure 1: The Dell R650 server's embodied carbon footprint is dominated by SSD storage, DRAM, and the mainboard. Other components combined make up around 21% of its overall embodied carbon footprint.

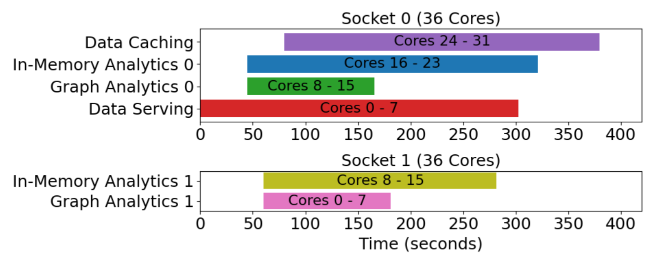


Figure 2: Six workloads are run on the R650 node during the 7-minute test period. Each workload is allocated eight cores when running.

where both the number of components and the peak current capacity are roughly proportional to the number of phases.

We model the embodied carbon footprint of the mainboard as a linear function of peak power capacity required, P_{peak}. We define ϕ as the embodied carbon footprint of the PDU per watt of power capacity.

$$CF_{MB}(P_{peak}) = CF_{fixed} + \phi \times P_{peak} \quad (10)$$

Other Carbon. We model the embodied carbon footprint of storage to be proportional to storage capacity. Depending on the storage technology, we apply a technology-specific rate of carbon per GB of storage. We assume the embodied carbon footprint of peripheral devices (e.g. NICs, HDD controllers, riser cards, etc.) to be fixed. We leave the development of variable embodied carbon models for peripheral devices to future work. We model the embodied carbon footprint of a power supply unit (PSU) and cooling as proportional to power capacity. We leave the development of more nuanced PSU and cooling embodied carbon models to future work.

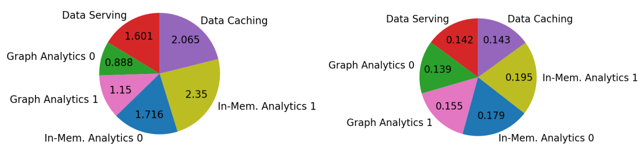
4 CASE STUDY: CLOUDSUITE ON DELL R650

We demonstrate our Shapley value-based carbon attribution framework on a CloudLab [12] Dell R650 server, described in table 1. We run a schedule of 6 workloads from CloudSuite 4.0 [13], shown in [12], on the node over a test period of 7 minutes and attribute carbon per workload. We use Intel RAPL to measure power for each CPU package and for DRAM, and we use Linux `top` to measure memory utilization.

We use publicly available information [38, 50] to estimate the CPU die and core areas and we assume a packaging carbon footprint

Table 1: CloudLab R650 Server Hardware Configuration and Embodied Carbon Model

Component	Specifications [8]	Embodied carbon model
CPU	Two 36-core Intel Xeon Platinum 8360Y at 2.4GHz	$CF_{CPU}(Q) = (2.804\ \texttt{kgCO}_2\texttt{e/CPU})\lceil \frac{Q}{36\ \texttt{cores/CPU}} \rceil + (0.1110\ \texttt{kgCO}_2\texttt{e/core})Q$
DRAM	256GB ECC Memory (16x 16 GB 3200MHz DDR4)	$CF_{MEM}(M) = (2.885\ \texttt{kgCO}_2\texttt{e/module})\lceil \frac{M}{16\ \texttt{GB/module}} \rceil + (0.5151\ \texttt{kgCO}_2\texttt{e/GB})M$
SSD	One 1.6TB NVMe SSD (PCIe v4.0) One 480GB SATA SSD	$CF_{SSD}(D) = (0.16\ \texttt{kgCO}_2\texttt{e/GB})D$
Mainboard	Dell R650 two-socket mainboard	$CF_{MB}(P_{peak}) = 100.1\ \texttt{kgCO}_2\texttt{e} + (0.02725\ \texttt{kgCO}_2\texttt{e/W})P_{peak}$
Chassis	Dell R650 1U, 2-socket	$CF_{chassis} = 34.30\ \texttt{kgCO}_2\texttt{e}$
Cooling	Air cooling (assumed)	$CF_{cooling}(P_{peak}) = (0.06082\ \texttt{kgCO}_2\texttt{e/W})P_{peak}$
PSU	1100W rated (assumed)	$CF_{PSU}(P_{peak}) = (0.06003\ \texttt{kgCO}_2\texttt{e/W})P_{peak}$
Peripherals	Two Mellanox PCIe 4.0 NICs	$CF_{peripheral}(P_{peak}) = 59.17\ \texttt{kgCO}_2\texttt{e}$



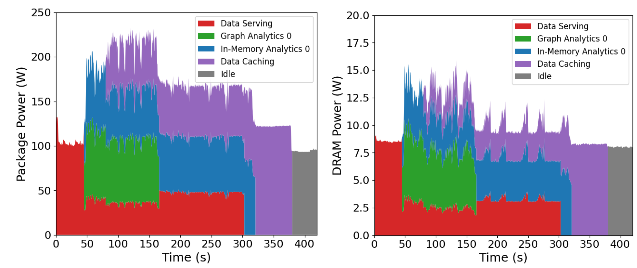
(a) Operational Carbon (gCO_2e) **(b) Embodied Carbon (gCO_2e)**

Figure 3: Carbon footprint per workload using Shapley value based attribution.

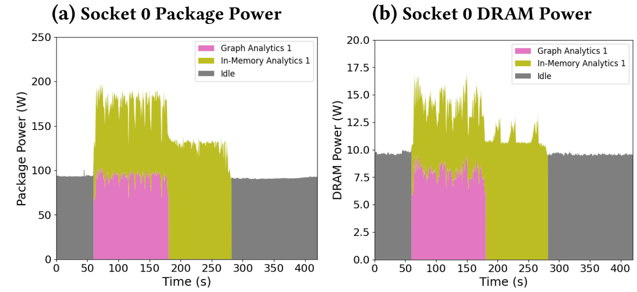
of 0.150 $kgCO_2e$ [9], in line with the ACT model [17]. We estimate the DRAM carbon footprint using wafer manufacturing and bit density data from [22]. Using DRAM module carbon footprint estimates from [36], we estimate the non-DRAM chip portion (PCB, RCD, miscellaneous components) of carbon footprint. We assume that SSDs have a embodied carbon footprint of 0.16 $kgCO_2e$/GB, based on [47]. For the chassis, mainboard PCB and connectors, PSU, and peripherals we assume the same carbon footprint as that of the Dell R740 [36]. For the power delivery network and cooling, we scale the Dell R740's carbon footprint by the R650's TDP.

4.1 Carbon Attribution Results

Operational and embodied carbon attribution results per workload are shown in figure 3 with per-component breakdowns shown in 5. Operational carbon attributions are derived from Shapley value based power attribution results, shown in figure 4, and using live grid carbon intensity data from Electricity Maps [2]. As expected, longer running workloads, like Data Caching, have higher operational carbon attribution. The shortest running workloads: Graph Analytics 0 and Graph Analytics 1, have the lowest operational carbon attributions. Moreover, as seen in figure 4, when more workloads are running concurrently, each workload's individual power attribution reduces as idle power is divided among a greater number of workloads. As a result, we see workloads running on socket 1 (Graph Analytics 1 and In-Memory Analytics 1) incurring greater power and operational carbon attributions than their counterparts on socket 0 (Graph Analytics 0 and In-Memory Analytics 0). Moreover, as seen in figure 5b, the embodied carbon per workload varies based on resource utilization and power. For example, In-Memory Analytics workloads use much more DRAM than other workloads



(a) Socket 0 Package Power **(b) Socket 0 DRAM Power**

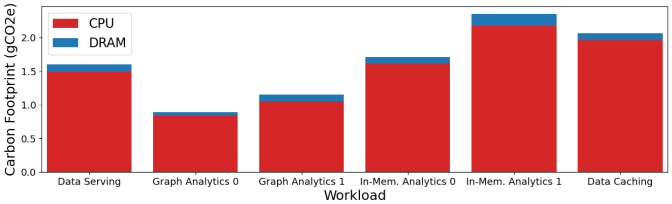


(c) Socket 1 Package Power **(d) Socket 1 DRAM Power**

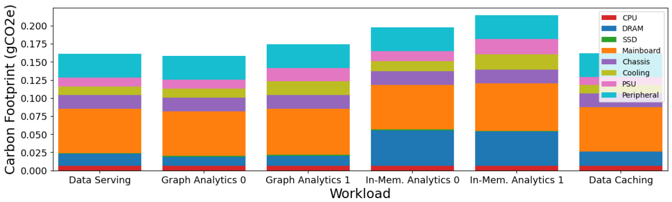
Figure 4: Shapley value based power attribution fairly divides system power among concurrent workloads based on each workload's contribution to overall power, capturing the non-linear effects of colocation.

and thus are attributed much more of the DRAM's embodied carbon.

4.1.1 Comparison with Energy-Proportional Attribution. In figure 6, we compare embodied carbon attribution results from our fair Shapley value method with results from a baseline energy-proportional method, showing that the baseline method can under-attribute by up to 43% and over-attribute by up to 37%. The baseline energy-proportional method attributes server embodied carbon to each workload proportional to the workload's energy attribution. The same proportion of embodied carbon from each resource is attributed regardless of resource utilization; for example, a workload that used 20% of total energy will be attributed 20% of the server's DRAM embodied carbon even if it used only negligible amounts



(a) Per Component Operational Carbon Breakdown



(b) Per Component Embodied Carbon Breakdown

Figure 5: The Shapley value based model attributes carbon on a component granularity. CPU dominates operational carbon. In contrast, CPU makes up only a small share of embodied carbon and other components (i.e., mainboard, DRAM, chassis, PSU, cooling) make up the bulk.

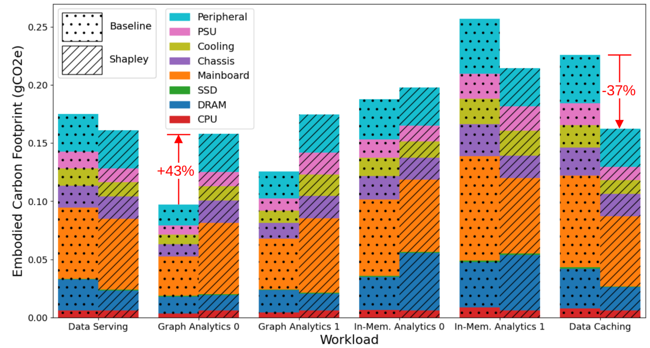


Figure 6: Energy-proportional baseline versus Shapley attribution. The baseline model attributes carbon to workloads proportional to the workload's energy attribution. The baseline attributes fixed proportions of each resource's carbon footprint to each workload regardless of actual per resource utilization.

of DRAM. GCP and Microsoft Azure use energy-proportional attribution along with billing-cost-proportional attribution to attribute embodied carbon to users. As discussed in section 2.2, energy and billing cost are poor proxies for embodied carbon. In contrast, our Shapley value method will attribute each resource's embodied carbon separately based on that specific resource's utilization while providing additional fairness guarantees via the Shapley value properties listed in section 3.1.

Looking at Graph Analytics 0 results in figure 6, we see that the baseline method attributes much less mainboard and peripheral embodied carbon even though they are embodied carbon costs that are largely fixed. The baseline method also attributes little CPU embodied carbon to Graph Analytics 0 compared to other workloads even though all the workloads use the same number of cores. The baseline method under-attributes embodied carbon to those workloads that consume less power relative to the amount of resources they use. On the other hand, the baseline over-attributes embodied carbon to workloads such as Data Caching because they consume more power relative to the resources they use.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose a model to fairly attribute carbon footprint to different workloads running on a data center server using a game-theoretic approach. Fine-grained carbon attribution can drive sustainability strategies for data center operators as well as carbon accounting and mitigation for cloud users. We formulate a Shapley value based attribution model for both operational and embodied carbon, and demonstrate the corresponding fair attribution on a dual socket server running CloudSuite 4.0 workloads. While the paper aims to address the need for fair fine-grained carbon attribution, we identify some key areas of future work.

Fine-grained feedback for improving software sustainability. The Shapley value based model empirically captures nuances of colocation on power and resource utilization. Future work should bridge the gap between the underlying software and hardware mechanisms that drive power consumption and the Shapley value to provide users insight into how to reduce their carbon emissions.

Scaling to many workloads and users. Given the number of diverse workloads running in hyperscaler data centers, it is intractable to compute the exact Shapley value at scale. Moreover, our approach of offline profiling of workload colocations for power measurements may not be practical for scaling reasons and also because workloads can be dynamic.

Incorporating data center scale hardware and software overheads. The experiments in this paper are run on a single-node, dual-socket server. However, data centers comprise tens of thousands of nodes that are supported by an extensive hardware and software infrastructure. The corresponding operational and embodied carbon from the supporting infrastructure should be attributed to workloads in future work.

Expanding model resource scope and colocation model. Our current model considers power, CPU cores, memory utilization, and storage utilization as resources metrics for inputs. In the future, we plan to expand our model to include network utilization, memory bandwidth utilization, and a more complex model for interference and resource contention effects.

ACKNOWLEDGMENTS

We sincerely thank Michael Shen, Zhanqiu Hu, Yueying Li, and the anonymous reviewers for their valuable feedback. We would also like to acknowledge the CloudLab infrastructure support (NSF 1419199) for providing resources to conduct these experiments. This work is also in part supported by NSF Grant CNS-2335795.

REFERENCES

[1] Marcelo Amaral, Huamin Chen, Tatsuhiro Chiba, Rina Nakazawa, Sunyanan Choochotkaew, Eun Kyung Lee, and Tamar Eilam. 2023. Kepler: A Framework to Calculate the Energy Consumption of Containerized Applications. In *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*. 69–71. https://doi.org/10.1109/CLOUD60044.2023.00017

[2] Electricity Maps ApS. 2024. Electricity Maps - South Carolina Electric Gas Company - May 6, 12:00 pm. Website. Retrieved May 6, 2024 from https://app.electricitymaps.com/zone/US-CAR-SCEG

[3] Crucial by Micron. 2024. How Much Power Does Memory Use? Retrieved May 2, 2024 from https://www.crucial.com/support/articles-faq-memory/how-much-power-does-memory-use

[4] Crucial by Micron. 2024. Micron 16GB DDR4-3200 VLP ECC UDIMM 2Rx8 CL22. Retrieved June 30, 2024 from https://www.crucial.com/memory/server-ddr4/mta18adf2g72az-3g2r

[5] Sunyanan Choochotkaew, Chen Wang, Huamin Chen, Tatsuhiro Chiba, Marcelo Amaral, Eun Kyung Lee, and Tamar Eilam. 2023. Advancing Cloud Sustainability: A Versatile Framework for Container Power Model Training. In *2023 31st International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 1–4. https://doi.org/10.1109/MASCOTS59514.2023.10387542

[6] Google Cloud. 2023. Carbon Footpint. Retrieved May 2, 2024 from https://cloud.google.com/carbon-footprint

[7] Google Cloud. 2023. Carbon Footprint reporting methodology. Retrieved May 2, 2024 from https://cloud.google.com/carbon-footprint/docs/methodology

[8] CloudLab. 2024. The CloudLab manual - 12. Hardware - 12.3 CloudLab Clemson. Retrieved May 2, 2024 from https://docs.cloudlab.us/hardware.html

[9] Ltd Siliconware Precision Industries Co. 2019. SPIL 2019 Corporate Social Responsibility. http://csr.spil.com.tw/uploads/2018-SPIL-CSR-EN.pdf

[10] Eli Cortez, Anand Bonde, Alexandre Muzio, Mark Russinovich, Marcus Fontoura, and Ricardo Bianchini. 2017. Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms. In *Proceedings of the 26th Symposium on Operating Systems Principles* (Shanghai, China) *(SOSP '17)*. Association for Computing Machinery, New York, NY, USA, 153–167. https://doi.org/10.1145/3132747.3132772

[11] Mian Dong, Tian Lan, and Lin Zhong. 2014. Rethink energy accounting with cooperative game theory. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking* (Maui, Hawaii, USA) *(MobiCom '14)*. Association for Computing Machinery, New York, NY, USA, 531–542. https://doi.org/10.1145/2639108.2639128

[12] Dmitry Duplyakin, Robert Ricci, Aleksander Maricq, Gary Wong, Jonathon Duerig, Eric Eide, Leigh Stoller, Mike Hibler, David Johnson, Kirk Webb, Aditya Akella, Kuangching Wang, Glenn Ricart, Larry Landweber, Chip Elliott, Michael Zink, Emmanuel Cecchet, Snigdhaswin Kar, and Prabodh Mishra. 2019. The Design and Operation of CloudLab. In *Proceedings of the USENIX Annual Technical Conference (ATC)*. 1–14. https://www.flux.utah.edu/paper/duplyakin-atc19

[13] PARSA @ EPFL. 2023. CloudSuite 4.0. Github repository. Retrieved May 1, 2024 from https://github.com/parsa-epfl/cloudsuite

[14] Joan Feigenbaum, Christos H. Papadimitriou, and Scott Shenker. 2001. Sharing the Cost of Multicast Transmissions. *J. Comput. System Sci.* 63, 1 (2001), 21–41. https://doi.org/10.1006/jcss.2001.1754

[15] Charlotte Freitag, Mike Berners-Lee, Kelly Widdicks, Bran Knowles, Gordon S. Blair, and Adrian Friday. 2021. The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations. *Patterns* 2, 9 (2021), 100340. https://doi.org/10.1016/j.patter.2021.100340

[16] Xiaoding Guan, Noman Bashir, David Irwin, and Prashant Shenoy. 2024. WattScope: Non-intrusive Application-level Power Disaggregation in Datacenters. *SIGMETRICS Perform. Eval. Rev.* 51, 4 (feb 2024), 24–25. https://doi.org/10.1145/3649477.3649491

[17] Udit Gupta, Mariam Elgamal, Gage Hills, Gu-Yeon Wei, Hsien-Hsin S. Lee, David Brooks, and Carole-Jean Wu. 2022. ACT: Designing Sustainable Computer Systems with an Architectural Carbon Modeling Tool. In *Proceedings of the 49th Annual International Symposium on Computer Architecture* (New York, New York) *(ISCA '22)*. Association for Computing Machinery, New York, NY, USA, 784–799. https://doi.org/10.1145/3470496.3527408

[18] Hongyu Hè, Michal Friedman, and Theodoros Rekatsinas. 2023. EnergAt: Fine-Grained Energy Attribution for Multi-Tenancy. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems* (Boston, MA, USA) *(HotCarbon '23)*. Association for Computing Machinery, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3604930.3605716

[19] IEA. 2023. *Tracking Clean Energy Progress 2023*. Technical Report.

[20] Intel. 2024. Intel® Xeon® Platinum 8360Y Processor. Retrieved June 30, 2024 from https://www.intel.com/content/www/us/en/products/sku/212459/intel-xeon-platinum-8360y-processor-54m-cache-2-40-ghz/ordering.html

[21] Mohammad A. Islam and Shaolei Ren. 2016. A New Perspective on Energy Accounting in Multi-Tenant Data Centers. In *USENIX Workshop on Cool Topics on Sustainable Data Centers (CoolDC 16)*. USENIX Association, Santa Clara, CA. https://www.usenix.org/conference/cooldc16/workshop-program/presentation/islam

[22] Scotten W Jones. 2023. *Modeling 300mm Wafer Fab Carbon Emissions*. Technical Report.

[23] Yadong Li, Dimitri Offengenden, and Jan Burgy. 2019. *Reduced Form Capital Optimization*. Papers 1905.05911. arXiv.org. https://ideas.repec.org/p/arx/papers/1905.05911.html

[24] Paul Lin, Robert Bunger, and Victor Avelar. 2023. *Quantifying Data Center Scope 3 GHG Emissions to Prioritize Reduction Efforts*. Technical Report.

[25] S. C. Littlechild and G. F. Thompson. 1977. Aircraft Landing Fees: A Game Theory Approach. *The Bell Journal of Economics* 8, 1 (1977), 186–204. http://www.jstor.org/stable/3003493

[26] Qiuyun Llull, Songchun Fan, Seyed Majid Zahedi, and Benjamin C. Lee. 2017. Cooper: Task Colocation with Cooperative Games. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 421–432. https://doi.org/10.1109/HPCA.2017.22

[27] Alexandra Sasha Luccioni, Yacine Jernite, and Emma Strubell. 2023. Power Hungry Processing: Watts Driving the Cost of AI Deployment? arXiv:2311.16863 [cs.LG]

[28] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2023. Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. 24 (2023), 1–15.

[29] Jason Mars, Lingjia Tang, Robert Hundt, Kevin Skadron, and Mary Lou Soffa. 2011. Bubble-Up: increasing utilization in modern warehouse scale computers via sensible co-locations. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture* (Porto Alegre, Brazil) *(MICRO-44)*. Association for Computing Machinery, New York, NY, USA, 248–259. https://doi.org/10.1145/2155620.2155650

[30] McKinsey and Company. 2023. *Investing in the rising data center economy*. Technical Report.

[31] Microsoft. 2020. *The carbon benefits of cloud computing - A study on the Microsoft Cloud in partnership with WSP*. Technical Report.

[32] Microsoft. 2021. *A new approach for Scope3 emissions transparency*. Technical Report.

[33] Microsoft. 2023. Emissions Impact Dashboard. Retrieved December 15, 2023 from https://www.microsoft.com/en-us/sustainability/emissions-impact-dashboard

[34] Microsoft. 2024. Microsoft Cloud for Sustainability API calculation methodology. Retrieved May 2, 2024 from https://learn.microsoft.com/en-us/industry/sustainability/api-calculation-method#calculation-methodology

[35] Hervé Moulin. 2003. *Fair Division and Collective Welfare*. The MIT Press. https://doi.org/10.7551/mitpress/2954.001.0001

[36] Thinkstep on behalf of Dell Technologies. 2019. *Life Cycle Assessment of Dell R740*. Technical Report.

[37] Himanshu Pandey. 2023. *Software tactics for sustainable cloud : an empirical study*. Ph. D. Dissertation. https://lutpub.lut.fi/handle/10024/166174 Accepted: 2023-08-08T06:45:04Z.

[38] Irma Esmer Papazian. 2020. New 3rd Gen Intel® Xeon® Scalable Processor (Codename: Ice Lake-SP). In *2020 IEEE Hot Chips 32 Symposium (HCS)*. 1–22. https://doi.org/10.1109/HCS49909.2020.9220434

[39] Andreas Schmidt, Gregory Stock, Robin Ohs, Luis Gerhorst, Benedict Herzog, and Timo Hönig. 2023. carbond: An Operating-System Daemon for Carbon Awareness. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems* (Boston, MA, USA) *(HotCarbon '23)*. Association for Computing Machinery, New York, NY, USA, Article 2, 6 pages. https://doi.org/10.1145/3604930.3605707

[40] Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni. 2021. CodeCarbon: Estimate and track carbon emissions from machine learning computing. https://doi.org/10.5281/zenodo.11097062

[41] Amazon Web Services. 2023. AWS Customer Carbon Footprint Tool Overview. Retrieved May 2, 2024 from https://aws.amazon.com/aws-cost-management/aws-customer-carbon-footprint-tool/

[42] Amazon Web Services. 2024. Understanding your carbon emission estimations. Retrieved May 2, 2024 from https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/ccft-estimation.html

[43] Lloyd S. Shapley. 1951. *Notes on the N-Person Game mdash; II: The Value of an N-Person Game*. RAND Corporation, Santa Monica, CA. https://doi.org/10.7249/RM0670

[44] Green Coding Solutions. 2024. Green Metrics Tool. Github repository. Retrieved May 2, 2024 from https://github.com/green-coding-solutions/green-metrics-tool

[45] Emily Sommer, Mike Adler, John Perkins, Joshua Thiel, Hilary Young, Chelsea Mozen, Dany Daya, and Katherine Sundstrom. 2020. Cloud Jewels: Estimating kWh in the Cloud. https://www.etsy.com/codeascraft/cloud-jewels-estimating-kwh-in-the-cloud?utm_source=OpenGraph&utm_medium=PageTools&utm_campaign=Share

[46] Abel Souza, Noman Bashir, Jorge Murillo, Walid Hanafy, Qianlin Liang, David Irwin, and Prashant Shenoy. 2023. Ecovisor: A Virtual Energy System for Carbon-Efficient Applications. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* (Vancouver, BC, Canada) *(ASPLOS 2023)*. Association for Computing Machinery,

New York, NY, USA, 252–265. https://doi.org/10.1145/3575693.3575709
[47] Swamit Tannu and Prashant J. Nair. 2023. The Dirty Secret of SSDs: Embodied Carbon. *SIGENERGY Energy Inform. Rev.* 3, 3 (oct 2023), 4–9. https://doi.org/10.1145/3630614.3630616
[48] Thoughtworks. 2023. Cloud Carbon Footprint. Retrieved May 2, 2024 from https://www.cloudcarbonfootprint.org/
[49] Richard Westerhof, Richard Atherton, and Vasilios Andrikopoulos. 2023. An Allocation Model for Attributing Emissions in Multi-tenant Cloud Data Centers. http://arxiv.org/abs/2305.10439 arXiv:2305.10439 [cs].
[50] Wikichips. 2023. Sunny Cove - Microarchitectures - Intel. Retrieved May 2, 2024 from https://en.wikichip.org/wiki/intel/microarchitectures/sunny_cove