The Sunk Carbon Fallacy: Rethinking Carbon Footprint Metrics for Effective Carbon-Aware Scheduling

Noman Bashir
Massachusetts Institute of Technology

Varun Gohil
Massachusetts Institute of Technology

Anagha Belavadi
Massachusetts Institute of Technology

Mohammad Shahrad
University of British Columbia

David Irwin
University of Massachusetts Amherst

Elsa Olivetti
Massachusetts Institute of Technology

Christina Delimitrou
Massachusetts Institute of Technology

ABSTRACT

The rapid increase in computing demand and its corresponding energy consumption have focused attention on computing's impact on the climate and sustainability. Prior work proposes metrics that quantify computing's carbon footprint across several lifecycle phases, including its supply chain, operation, and end-of-life. Industry uses these metrics to optimize the carbon footprint of manufacturing hardware and running computing applications. Unfortunately, prior work on optimizing datacenters' carbon footprint often succumbs to the *sunk cost fallacy* by considering embodied carbon emissions (a sunk cost) when making operational decisions (i.e., job scheduling and placement), which leads to operational decisions that do not always reduce the total carbon footprint.

In this paper, we evaluate carbon-aware job scheduling and placement on a given set of servers for a number of carbon accounting metrics. Our analysis reveals state-of-the-art carbon accounting metrics that include embodied carbon emissions when making operational decisions can actually increase the total carbon footprint of executing a set of jobs. We study the factors that affect the added carbon cost of such suboptimal decision-making. We then use a real-world case study from a datacenter to demonstrate how the sunk carbon fallacy manifests itself in practice. Finally, we discuss the implications of our findings in better guiding effective carbon-aware scheduling in on-premise and cloud datacenters.

CCS CONCEPTS

• **Hardware** → **Impact on the environment**; **Emerging tools and methodologies**; • **General and reference** → **Metrics**; **Measurement**; **Evaluation**; **Empirical studies**.

KEYWORDS

Sustainable computing, operational carbon emissions, embodied carbon emissions, carbon footprint, sustainability, metrics, software carbon intensity, lifecycle carbon footprint, datacenters, server lifetime, scheduling, job placement, performance.

SoCC '24, November 20–22, 2024, Redmond, WA, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1286-9/24/11.
https://doi.org/10.1145/3698038.3698542

ACM Reference Format:
Noman Bashir, Varun Gohil, Anagha Belavadi, Mohammad Shahrad, David Irwin, Elsa Olivetti, and Christina Delimitrou. 2024. The Sunk Carbon Fallacy: Rethinking Carbon Footprint Metrics for Effective Carbon-Aware Scheduling. In *ACM Symposium on Cloud Computing (SoCC '24), November 20–22, 2024, Redmond, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3698038.3698542

1 INTRODUCTION

Computing's demand has experienced a meteoric rise over the last few decades with no signs of slowing down [18]. Indeed, computing's demand is likely accelerating due to the recent emergence of generative artificial intelligence (AI) tools, such as ChatGPT [13] and GitHub Copilot [45], that are computationally-intensive. New AI tools promise to unlock a broad spectrum of innovative and useful applications. However, as marginal improvements in computing's energy efficiency shrink due to the deceleration of process scaling [27, 54], the ever-growing demand for computing power is poised to drive a proportional increase in its energy consumption. Computing's growing energy footprint has raised significant climate and sustainability concerns. Fortunately, the importance of improving computing's sustainability is gaining awareness [8, 46, 60], with coordinated efforts from both industry and academia aimed at mitigating its climate impact [9, 41, 55, 56, 60].

Recent efforts to improve computing's sustainability have focused on quantifying and optimizing its carbon footprint across all stages of the computing lifecycle, from chip design and manufacturing [1, 24] to the operation of computer systems [25, 28, 43] and the management of e-waste at the end of life [49]. The Greenhouse Gas (GHG) Protocol [59] distinguishes between different types of emissions: Scope 2 covers the GHG emissions related to electricity consumption in datacenters (often referred to as operational emissions), while Scope 3 includes emissions arising from chip manufacturing, the supply chain, and e-waste management (often referred to as embodied emissions). Prior work on quantifying computing's carbon footprint employs various metrics based on either operational emissions alone or a weighted combination of both operational and embodied emissions. A common approach in the literature is to aggregate the operational emissions from executing a job with a portion of the embodied emissions of the server running that job, where the server's embodied emissions are distributed across jobs based on the time and resources allocated to them. Notable examples include the Software Carbon Intensity

(SCI), introduced by the Green Software Foundation [20], Computational Carbon Intensity [49], and Sustainability Cost Rate [21]. While these metrics might use different terminology, the underlying principle remains consistent: a job's carbon footprint is the sum of its share of the hardware's embodied carbon emissions and the operational emissions generated during its execution.

In this paper, we focus on carbon-aware workload scheduling and job placement on datacenter servers. While embodied carbon metrics like SCI are often proposed to guide operational decisions, such as scheduling and job placement, we argue that scheduling and procurement are orthogonal processes that operate on different timescales and should be optimized independently. Scheduling decisions determine which servers handle specific jobs and should target reducing the operational carbon footprint generated by actively running servers. In contrast, procurement decisions – such as which servers to buy and when to replace them – can affect the embodied carbon footprint associated with the hardware's manufacturing, which cannot be influenced when a job is being scheduled. These processes are separate: scheduling occurs continuously as jobs are assigned to servers, while procurement decisions are made at longer intervals based on hardware lifecycles.

Importantly, metrics like SCI, which incorporate lifecycle emissions, typically account for only the emissions of the servers running the jobs, and ignore the embodied carbon of idle or unused servers. This oversight can lead to unintended consequences when optimizing for SCI-like metrics in job scheduling by, paradoxically, increasing rather than decreasing a datacenter's overall carbon footprint, as it neglects the broader carbon impact of the entire server fleet, including unused hardware. Therefore, we demonstrate that optimizing SCI-like metrics alone when scheduling may actually undermine the goal of minimizing a datacenter's total carbon footprint, highlighting the need for separate, independent optimization of scheduling and procurement.

The suboptimal outcomes of carbon-aware scheduling decisions based on SCI-like metrics stem from a cognitive bias known as the *sunk cost fallacy*. According to the *principle of bygones*, a concept rooted in the principle of separability in standard economic theory, operators should base decisions solely on future possibilities, without being influenced by past expenditures or events that cannot be changed [17]. Applied to datacenter operations, this means that scheduling and job placement decisions should be made by focusing only on the current operational context by, in this case, ignoring the embodied carbon emissions that have already occurred. Since embodied emissions are fixed at procurement time, they cannot be altered by later operational choices. Thus, operators should aim to minimize operational carbon emissions from running jobs on the existing hardware rather than factoring in sunk embodied carbon.

Ignoring sunk costs is intuitive and supported by prior research [22, 39, 48, 57]. However, recent attempts to develop metrics that optimize computing's lifecycle carbon footprint inadvertently fall into the *sunk carbon fallacy*, a variant of the sunk cost fallacy applied to carbon. By incorporating embodied emissions into real-time scheduling decisions, these metrics conflate two separate processes: procurement and operation. Our illustrative example in Section 3.3 highlights how using SCI as a scheduling metric can counterproductively increase a datacenter's overall carbon footprint, underscoring the importance of independently optimizing scheduling and procurement to optimize carbon efficiency.

The degree to which minimizing the total carbon footprint of a datacenter diverges from minimizing the sum of individual job-level lifecycle carbon using metrics like SCI depends on several characteristics of datacenter infrastructure. One significant factor is the heterogeneity in servers' performance relative to their operational and embodied carbon footprints. If all servers in the datacenter are homogeneous – delivering similar performance for a given application – then incorporating embodied carbon into a scheduling metric like SCI would not affect the overall system-level carbon footprint. However, real-world datacenters often consist of heterogeneous servers, with differences arising from factors such as hardware age (new vs. old) and type (CPU vs. GPU). For instance, older servers typically have a smaller embodied carbon footprint due to their earlier manufacture but often exhibit a higher operational carbon footprint than newer, more energy-efficient servers, as shown in Figure 1. Additionally, while GPUs may excel in compute-intensive tasks, CPUs can sometimes deliver better performance per unit energy or carbon for specific workloads [2].

Our work focuses on heterogeneity in CPUs, which is significant enough to demonstrate that applying a one-size-fits-all metric like SCI, which includes embodied carbon, can distort scheduling decisions in ways that increase the total carbon footprint. A second critical factor is datacenter utilization. If utilization is either very high or very low – where either all servers are in use, or none are – then the choice of scheduling metric will have little impact on the overall carbon footprint. However, at intermediate utilization levels, common in many datacenters, metrics like SCI can lead to inefficient scheduling, increasing the total carbon footprint. In Section 3.3, we further explore this discrepancy and evaluate the influence of these infrastructure factors on a datacenter's carbon footprint through concrete examples.

In illustrating how the *sunk cost fallacy* manifests in carbon-aware scheduling, this paper makes the following contributions:

1 – We demonstrate that metrics incorporating both embodied and operational carbon emissions, while seemingly comprehensive, can result in sub-optimal scheduling decisions. These metrics may paradoxically increase a datacenter's total carbon footprint, contrary to their intended purpose. We also examine the key factors, such as datacenter utilization levels, operational carbon intensity, and embodied carbon amortization approaches that exacerbate these sub-optimal outcomes.

2 – We evaluate three metrics, including those that prioritize operational carbon emissions or account for infrastructure-wide embodied carbon emissions more appropriately than SCI. Through a real-world case study of an on-premise datacenter, we show that, under realistic workload assumptions, focusing on operational carbon emissions leads to more carbon-efficient scheduling outcomes, effectively reducing the total carbon footprint.

3 – We provide practical guidelines for datacenter operators and users, detailing how to avoid the sunk carbon fallacy. Our recommendations include selecting metrics that accurately reflect the carbon costs relevant to operational decision-making, thereby optimizing for a lower total carbon footprint.

2 BACKGROUND AND MOTIVATION

This section provides an overview of efforts to improve computing's sustainability, the metrics used in carbon-aware optimizations, and the research gaps in understanding sustainability metrics.

Prior work on sustainable computing. There has been significant work highlighting the environmental impact of computing [46] and understanding what it means for computing to be sustainable [11, 60]. Prior work has also analyzed various carbon accounting paradigms in the context of computing and highlighted the challenges in accounting for the carbon footprint of computing [10, 30]. Specifically, it examines how the values for embodied carbon [10] and operational carbon intensity can be error-prone [30]. Recent work has also focused on quantifying operational and embodied carbon, as well as their tradeoff. These efforts guide the architectural design process to reduce servers' overall lifecycle carbon footprint. There is also prior work on understanding the benefits and limitations of spatiotemporal workload scheduling for reducing carbon [47]. An orthogonal body of work has focused on devising algorithms for carbon-aware workload shifting and building systems that enable the deployment of such carbon-aware algorithms [23, 25, 43, 52]. Unfortunately, the real-world use of carbon-aware optimizations is limited, with a single example of carbon-aware workload shifting in hyperscalers [41].

Metrics for sustainable computing. Recent work has looked at various metrics that should be used to quantify and optimize computing's carbon footprint. Gandhi et al. [21] propose sustainability metrics for datacenters, such as the amortized sustainability cost metric that attributes operational and embodied carbon to a job. Switzer et al. [49] look at the end-of-life problem for computing hardware and propose a metric, called computational carbon intensity (CCI), for making component replacement and end-of-life decisions for computing hardware. The software industry has also focused on promoting and facilitating the development of green software, such as prior work done by the Green Software Foundation (GSF) [20]. GSF has proposed a metric, Software Carbon Intensity (SCI), that can quantify software's carbon footprint and enable software practitioners to make decisions that reduce it.

Limitations and research gaps. Prior work on carbon accounting has proposed various metrics to reduce computing's carbon footprint, which have spurred a debate on the usefulness and efficacy of the proposed metrics [14, 15, 19, 42]. However, despite this being a critical problem, very little work has been done on analyzing the incentives each metric offers and the outcomes it yields. Recent work argues that devising a single metric that is simple, accurate, precise, and provides desirable incentives for optimizing decision-making across computing's entire lifecycle may not be possible [50]. Simultaneously, it is challenging to evaluate all potential combinations. The total lifecycle carbon footprint includes the embodied carbon of all servers, operational carbon of idle servers, and carbon emissions of active servers running workloads. Procurement decisions and job scheduling choices matter, but they operate on different timescales: seconds to days for scheduling and months to years for procurement. This work focuses on carbon-aware workload scheduling in the public cloud and enterprise datacenters, aiming to reduce the carbon footprint added in this life stage.

3 THE SUNK CARBON FALLACY

This section illustrates how the state-of-the-art carbon accounting metrics fall victim to the *sunk carbon fallacy*, outlines factors that determine the impact of suboptimal decision-making on the total carbon footprint, and analyzes alternative metrics that yield better carbon-aware scheduling outcomes.

3.1 Analysis Setup

The task of carbon-aware scheduling and job placement is to schedule a given set of jobs on a set of available servers to minimize the total carbon footprint of executing the jobs. In our illustrative example, we assume the following setup.

– **Scheduler.** The goal of the scheduler is to place a set of jobs onto a given set of servers to minimize the total carbon footprint of running all the jobs. The scheduler does not assume any information about the future arrival of jobs and their characteristics, and makes instantaneous decisions on job placement – a setup commonly used by in-production schedulers, such as Borg [7, 53].

– **Jobs.** The performance characteristics and energy consumption of jobs on the available servers are known. These characteristics can be obtained through profiling or increasingly common benchmark databases, such as MLPerf [32] and OpenBenchmarking Suite [38].

– **Servers.** The servers are not power proportional, i.e., they consume significant power at 0% utilization [6, 29], often more than 30% of their peak power usage. However, for the processing component, the idle power is significantly lower. Finally, while individual servers may be fully utilized, the datacenter-level utilization ranges from 30% to 60% even for the state-of-the-art datacenters [53].

– **Energy and Carbon Footprint Estimates.** The energy consumption and carbon footprint at the server level also depends on the power supplies used, the number of hard drives, the memory size, and the chassis, among other components. We use the inventory information from MIT's academic clusters, including the Bates Research and Engineering Center [33] and the hydro-powered Massachusetts Green High Performance Computing Center (MGHPCC) [34]. We only have information on the processor component of the server. As a result, our current embodied and operational carbon estimates only account for the processing component.

The embodied carbon for the processor is estimated using a research version of the integrated circuit module of PAIA [37], which uses information on the technology node (such as 7nm or 28nm), CPU package area, die size, and the fabrication location. The official Intel and AMD websites often provide data on the technology node and CPU package area but do not provide information on the die size. We obtained the die sizes using data from the TechPowerUp [51], CPU-World [16], X86 CPU's Guide [35], and WikiChip [58] websites. Since these are not official websites, we ensured that the die sizes across websites were consistent; we only used processors with consistent information across at least two websites. For AMD processors, we used carbon intensity values for Taiwan, which is $495\texttt{g.CO}_2\texttt{/kWh}$ [36]. For Intel, we assume fabrication in Hillsboro, Oregon, with a carbon intensity of $357\texttt{g.CO}_2\texttt{/kWh}$ [31].

For the operational carbon estimates, we assume a server consumes its rated Thermal Design Power (TDP) at 100% utilization, with a linear increase in power between the extremes. For the carbon intensity, unless otherwise specified, we assume the datacenter is situated in Sweden and has a carbon intensity of $\mathtt{20g.CO_2/kWh}$ [31]. In Section 3.3, we vary the carbon intensity for analysis in embodied- and operational-dominant regions.

– **Performance Benchmarks.** We use three benchmarks to get performance scores for the processors: Multithread Ratings for CPUs by PassMark [40], HEPScore [26], and SPEC CPU2017 Floating Point Speed [44]. However, not all the benchmarks profile each processor, narrowing the set of processors used in our analysis.

3.2 Carbon-Aware Scheduling Metrics

This section defines three different metrics that can be used to evaluate carbon-aware scheduling and job placement.

1 – Software Carbon Intensity (SCI) was introduced by the Green Software Foundation [20]; it quantifies the rate of total carbon emissions per functional unit **R**. The functional unit here can be an API call, ML training, or large language model (LLM) inference.

The carbon emissions for a given job include both the operational carbon emissions (denoted by O) for running the job on the server and the embodied carbon emissions (denoted by M) for the functional unit representing the job. SCI is expressed as,

$$\mathtt{SCI} = (\mathtt{O} + \mathtt{M}) \text{ per } \mathtt{R}.$$

$$\mathtt{SCI} = ((\mathtt{E} * \mathtt{I}) + \mathtt{M}) \text{ per } \mathtt{R}.$$

Here, E is the energy consumption in kilowatt-hours of the job over a given time window. This includes a portion of the idle power for the server assigned to the job and dynamic power due to the job's resource usage. I is the carbon intensity of electricity in grams of carbon dioxide equivalent per kilowatt-hour ($\mathtt{g.CO_2/kWh}$) for the region where the server consumes electricity.

SCI only accounts for the embodied carbon (M) of the active server running the job with its value is computed as,

$$\mathtt{M} = \mathtt{TE} \times \frac{\mathtt{T} \times \mathtt{RR}}{\mathtt{EL} \times \mathtt{TR}}. \qquad (1)$$

Here, TE is the total embodied emissions, EL is the expected lifespan, and TR is the server's total resources for the server running the job. T is the time duration and RR is the resource reserved for the job. Note that SCI ignores the embodied and operational emissions of the idle servers in the datacenter, as it focuses on accounting for a given job's carbon footprint (see SCI specifications for details [20]).

2 – Total Software Carbon Intensity (tSCI) extends **SCI** by considering the embodied carbon emissions at the infrastructure scale, a potential solution to accurately account for total emissions. This metric assigns a portion of the total embodied emissions for the infrastructure to each running job instead of just considering the embodied emissions for the server that runs the job.

In extending SCI, which already accounts for the embodied carbon for the server that runs the job, we must add a fraction of the infrastructure-level embodied carbon emissions proportional to the resources reserved and allotted time for the job. This can be simplified by taking the datacenter's total embodied carbon and assigning a portion of that to the job. In this case, the total embodied carbon for the job (tM) is computed as,

$$\mathtt{tM} = \mathtt{M} + \mathtt{M}_{\text{idle-infra}}.$$

The value of $\mathtt{M}_{\text{idle-infra}}$ is calculated as the sum of M for all the idle servers using Equation 1, i.e., each idle server's embodied carbon is also proportionally assigned to the job. Similarly, the operational carbon from the base power consumption of idle servers also contributes to tO, the total operational carbon footprint,

$$\mathtt{tO} = \mathtt{O} + \mathtt{O}_{\text{idle-infra}}.$$

Finally, the total software carbon intensity can be computed as,

$$\mathtt{tSCI} = (\mathtt{tO} + \mathtt{tM}) \text{ per } \mathtt{R}.$$

To show how to calculate this value, consider an example where there is a datacenter with two servers A and B, with embodied carbon values of $\mathtt{400g.CO_2}$ and $\mathtt{50g.CO_2}$ and an expected lifetime of 10 years and 5 years, respectively. Assume server A has 40 cores, and server B has 10 cores. Suppose a job $\mathtt{J_1}$ that runs for one year and uses 10 cores is scheduled on server B. Another job $\mathtt{J_2}$ runs on server A and uses 10 cores; the value of embodied carbon attributed to $\mathtt{J_1}$ will be computed as,

$$\mathtt{tM} = 10\mathtt{g.CO_2} + \underbrace{\frac{400\mathtt{g.CO_2} \times 1yr}{10yrs}}_{\text{time fraction}} \times \underbrace{\frac{30cores}{40cores}}_{\text{idle fraction}} \times \underbrace{\frac{10cores}{20cores}}_{\text{usage fraction}},$$
$$= 25\mathtt{g.CO_2}.$$

In the fraction above, the time, idle, and usage fractions are the amortization terms that amortize the embodied carbon of the idle infrastructure over time (1 out of 10 years), idle resources (30 out of the 40 cores in the remaining infrastructure are idle), and usage (job uses 10 of the total 20 cores used). Since operational carbon emission rates are instantaneous, the value of tO can also be calculated using the same method, except for the time fraction component.

3 – Operational Software Carbon Intensity (oSCI) metric ignores the embodied carbon emissions for all the servers. It makes scheduling decisions based on the operational carbon emissions of running a given job. oSCI is expressed as,

$$\mathtt{oSCI} = (\mathtt{E} * \mathtt{I}) \text{ per } \mathtt{R}.$$

This metric can include a portion of the base power from the idle servers to incentivize turning off servers when they are not needed. However, for the current purpose, we keep it simple and only account for the energy used by the server running the job.

Computing SCI, tSCI, and oSCI in Practice involves different degrees of challenges. First, oSCI is a subset of the other two metrics and is the simplest to calculate, as the operating power of a job can be estimated through offline profiling. SCI requires embodied carbon estimates for all the servers in a datacenter, which can be difficult to obtain in practice. Note that embodied carbon estimates also tend to have a high degree of uncertainty [3, 12, 37]. As a result, the uncertainty in the embodied carbon estimates can propagate and affect the scheduling outcomes unpredictably.

Finally, calculating tSCI and tracking it over time is complex and necessitates comprehensive datacenter-level information, encompassing all hardware components and active jobs, including their resource reservations and expected runtime. Also, as the jobs arrive and leave, the idle fraction of the infrastructure will change, resulting in a time-varying value of tSCI. While cloud operators have access to this data, calculating tSCI demands sophisticated data collection infrastructure and precise online attribution, which entail considerable cost and carbon overheads. Such detailed information is generally not accessible to end users in many contexts,

Table 1: Specifications of servers in our illustrative example.

	S_A	S_B
Processor	`Xeon E-2286G`	`Xeon Gold 6538N`
Release Date	05/29/2019	12/14/2023
PassMark Score	14020	44895
TDP (W)	90	205
Technology Node	14nm	10nm
Embodied Carbon (Kg.CO2)	8.04	101.89

Table 2: Values of `SCI`, `tSCI`, `oSCI` for S_A and S_B for job placement in `g.CO`$_2$ per `Score-Yr`. We also report the total cluster carbon footprint for each metric.

Metric	Scheduling/Placement		Accounting
	S_A	S_B	Cluster Carbon Footprint
`SCI`	0.11 + 0.83 = **0.94**	0.45 + 0.56 = 1.01	(0.11 + 0.45) + 0.83 = 1.39
`tSCI`	0.94 + 0.45 = 1.39	1.01 + 0.11 = **1.12**	(0.11 + 0.45) + 0.56 = 1.12
`oSCI`	0.83	**0.56**	(0.11 + 0.45) + 0.56 = 1.12

such as public cloud environments, prohibiting them from computing their carbon footprint if the cloud providers do not share this information. Therefore, we do not envision this metric being used in practice; instead, we include it for completeness of the metrics and show that a more straightforward metric of `oSCI` can achieve the same scheduling outcomes.

Finally, incorporating information on operational and embodied carbon estimates into scheduling decisions depends on the scheduler being used. For example, in Slurm, nodes can be assigned arbitrary weights that determine their priority for scheduling; these weights can be set to the value of the metric of choice, such as `oSCI`. To compute `oSCI` values, Slurm's or any other resource manager's energy monitoring tool can be easily augmented to report operational emissions with little overhead.

3.3 An Illustrative Example

We first use an illustrative example to demonstrate the *sunk carbon fallacy*. Consider a small datacenter with two servers powered by two processors from Intel: `Xeon E-2286G` and `Xeon Gold 6538N`. We refer to these two servers as S_A and S_B, respectively. Table 1 provides the detailed specifications for the two servers, including processor model, their release dates, PassMark scores, embodied carbon estimates, and TDP values.

Figure 1 shows the operational and embodied carbon emissions normalized to the PassMark score and the server's expected lifetime for the two servers in our illustrative datacenter. Intuitively, an operational carbon value of 0.56 means that getting a performance of 1 score for one year using S_A will incur operational emissions of 0.56`g.CO`$_2$. Note that our illustrative example maps to an increasingly common real-world scenario, where a newer server (S_B) manufactured using recent technology, 10nm in this case, has 4.09× higher embodied carbon footprint than an older server (S_A) manufactured with 14nm technology. However, the energy efficiency gains over the last few years mean that S_B is significantly more energy-efficient (consumes 32.5% less energy) than S_A.

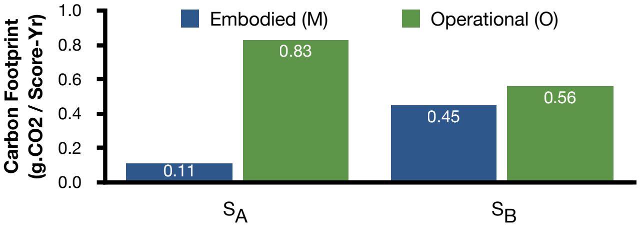

Figure 1: The normalized embodied and operational carbon footprint (`g.CO`$_2$) per `Score-Yr` assuming the datacenter is located in Sweden and the electricity has a carbon intensity of 14`g.CO`$_2$`/kWh` [31]. The servers have a lifetime of 5 years.

Table 3: Additional example scenarios that may lead to sunk carbon fallacy, i.e., an inefficient server with low `SCI` value is used before an efficient server with high `SCI` value. Values of carbon emissions are in `g.CO`$_2$ per `Score-Yr`.

Server Pairs		Additional Details
Xeon E-2486 0.08 + 0.47 = **0.55**	EPYC 9334 0.23 + 0.39 = 0.62	New Xeon server (12/14/2023, 10nm) vs. old EPYC server (11/10/2022, 5nm).
Ryzen 5965WX 0.15 + 0.51 = **0.66**	Xeon W9-3495 0.25 + 0.46 = 0.71	Older Ryzen server (03/08/2022, 5nm) vs. New Xeon server (02/15/2023, 10nm).

1 – Analyzing Scheduling Outcomes. Table 2 shows the carbon footprint values used to choose one of the servers for job placement. We also report the total lifecycle emissions of the datacenter for the duration of the job, including the embodied carbon footprint of all the servers and the operational carbon footprint of the active servers. The values for the server with the lowest metric are highlighted in bold; the server with the lowest value is chosen to run the job. We compute the datacenter-level carbon footprint as the sum of embodied carbon for all the servers (the sunk cost) and operational carbon for the server running the job (the marginal or additional cost). As shown, in prioritizing the sum of embodied and operational, `SCI` chooses a highly energy-inefficient server with a low `SCI` value due to a small embodied carbon value. While this placement is preferable based on the `SCI` metric, it leads to a 24.10% higher carbon footprint for the cluster. On the other hand, the placement choices of `tSCI` and `oSCI` align and lead to the minimum value of the cluster-level carbon footprint as both minimize the additional emissions to obtain the desired performance.

In this example, we used the classic case of a new efficient server with high embodied carbon against an old energy-inefficient server with low embodied carbon, primarily due to the technology node difference. However, this discrepancy of an energy-inefficient server having a lower `SCI` value than an energy-efficient server can also occur in other scenarios. For example, as shown in Table 3, the new `Xeon E-2486` server uses a 10nm technology node and has a smaller embodied carbon than its counterpart `EPYC 9334` server. The energy efficiency gains and performance improvement for `EPYC 9334` from advanced manufacturing are not enough to outweigh the increase in embodied carbon, leading to its higher `SCI` value. Surprisingly, a similar discrepancy occurs between `Ryzen Threadripper 5965WX` and `Xeon W9-3495` as the former has a lower embodied carbon footprint due to a smaller die despite using a 5nm technology node as compared to 10nm for the latter. We hand-picked these examples to demonstrate the existence of the *sunk carbon fallacy* beyond the classic old vs. new example. While these servers are not like-for-like replacements for each other, they can still be available in a given datacenter or a cloud platform, leading to the choice of an inefficient server over an energy-efficient one.

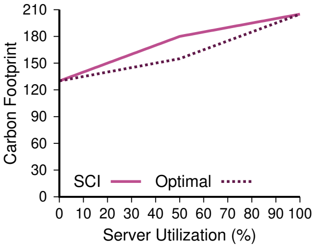

Figure 2: Utilization Impact

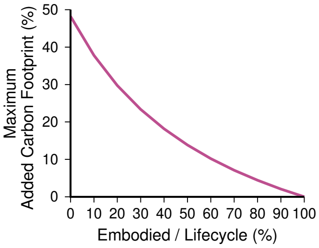

Figure 3: Op. Carbon Impact

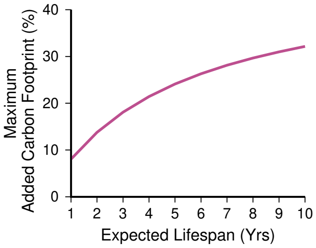

Figure 4: Lifespan Impact

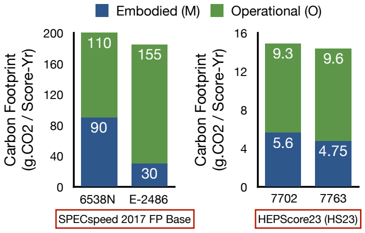

Figure 5: Benchmark Impact

2 – Effect of Datacenter Utilization. Our illustrative example shows how differences in server characteristics lead to suboptimal scheduling outcomes. We next analyze the effect of utilization on the increase in the system-level carbon footprint when using SCI. The server S_A has 12 logical cores (6 physical cores and 2 threads per core), where each logical core provides a performance score of 1168. The server S_B has 64 logical cores (32 physical cores and 2 threads per core), where each logical core provides a performance score of 701. Each job uses one logical core on S_A. On S_B, each job uses 2 logical cores to get a 1402 performance score (closer to 1168 score for S_A). We have a total of 44 cores of similar performance.

Figure 2 shows the increase in the system-level carbon footprint when jobs are scheduled using SCI compared to scheduling based on tSCI or oSCI. If no server in the datacenter is being used or all servers are being used, all the metrics yield the same outcome. However, when the utilization levels are between 0 and 100%, the set of servers selected to run the job matters. The peak discrepancy happens when only 12 cores are needed to run the jobs (at 27.3% utilization) and decreases afterwards. The exact magnitude of the peak and the utilization at which this manifests will change based on the set of servers, their base power values, and the granularity at which the jobs can be scheduled. In Section 4, we provide the same results for the academic datacenter we study.

3 – Effect of Operational Carbon Intensity. In our current setup, embodied carbon accounts for 11.7% and 44.5% of the lifecycle emissions for S_A and S_B, respectively. The average value across servers is 28.1%. We scale our normalized operational carbon footprint to generate values such that embodied carbon accounts for 10% to 90% of the lifecycle emissions and study its impact on the increase in system-level footprint due to SCI. Figure 3 shows the maximum value of the added carbon footprint due to the sunk carbon fallacy as the embodied carbon footprint accounts for an increasing fraction of the lifecycle emissions. At 0%, only the operational efficiency matters and the use of S_A results in a 48% increase in the system-level carbon footprint. At the other extreme of 100%, the operational carbon is 0 and the choice of server does not matter.

We note that operational carbon emissions dominate despite our use of Sweden for the datacenter location, one of the greenest regions in the world with only $20\texttt{g.CO}_2\texttt{/kWh}$. This is because our embodied carbon estimates only include the processor component that contributes a small fraction of the overall server-level carbon footprint. However, the TDP value of the processor component accounts for most of the server-level power and operational carbon footprint. If server-level embodied carbon values are used, the carbon intensity values at which embodied carbon accounts for a given percentage of lifecycle emissions will be higher.

4 – Effect of Server's Expected Lifetime. The expected lifespan of servers has a similar impact on the added carbon footprint at the system level. Figure 4 shows the maximum added carbon footprint at the system-level as the server's embodied carbon is amortized over a longer period. As the expected lifespan increases, the amortized embodied carbon per year decreases, and its fraction of the lifecycle carbon footprint decreases. As shown in Figure 3, lower embodied values result in a higher system-level carbon footprint under SCI, magnifying the impact of the *sunk carbon fallacy*.

5 – Effect of Performance Metric. Our results thus far have used PassMark scores. However, our observation is agnostic to any particular benchmarking method. Figure 5 shows that the conditions required for the *sunk carbon fallacy*, i.e., a server with low SCI is inefficient, manifest across different benchmarks. The servers we use in our examples changed, as we did not have SPEC and HS26 scores for the servers in the illustrative example. While the combination of servers that manifest the *sunk carbon fallacy* may change, the effect should be present in all performance benchmarks.

3.4 Generalization of Outcomes

We next review if our observations hold for all scenarios of hardware choices (concerning their embodied and operational carbon ratios). Let us assume there are N servers in a datacenter, and we need k servers at a time. Let M_i, O_i be the embodied and operational carbon costs of server i. Let $Z_i = M_i + O_i$ be the total carbon emissions for a functional unit or over the server's lifetime.

The SCI and oSCI strategies can be written as:

$$\texttt{SCI} = \{i \mid Z_i \text{ are the } k \text{ smallest values of } Z\},$$
$$\texttt{oSCI} = \{i \mid O_i \text{ are the } k \text{ smallest values of } O\}.$$

If k is zero or equals N, both strategies yield the same set of servers. oSCI directly minimizes $\sum_{i \in \texttt{oSCI}} O_i$, the operational carbon, which is the only cost that can be reduced post-purchase. Since SCI might include servers with a lower lifecycle cost Z_i but potentially higher O_i, oSCI can yield a lower total carbon when both embodied and operating carbon are considered together. Therefore:

$$\sum_{i \in \texttt{oSCI}} O_i \leq \sum_{i \in \texttt{SCI}} O_i.$$

Given the above inequality and considering the total carbon footprint $Z_i = M_i + O_i$ across embodied and operation al phases across all the servers in the datacenter, the choice of oSCI ensures the minimum total carbon across purchase and operation.

Extending our example to show that operational carbon emissions yield the lowest carbon footprint even when jobs arrive over time is straightforward. However, doing so is outside the scope of this vision paper and a subject of future work.

Table 4: List of servers and their specifications for the case study datacenter. The embodied carbon values only account for the processor. The server life is assumed to be 5 years if it is less than 5 years old; otherwise, its embodied carbon is amortized over all the years since purchase. The operational carbon values are for five years and computed with a carbon intensity of 10 $g.CO_2/kWh$ (chosen such that embodied carbon accounts for 20% of the lifecycle emissions).

Processor	Purchase	Server	Technology	Embodied Carbon	Performance & Power				Operational Carbon	Carbon ($g.CO_2$ /Score-Yr)		
	Year	Count	Node	($KgCO_2$)	PassMark	TDP (W)	Cores	Threads	($KgCO_2$)	M	O	SCI
Xeon-Silver-4216	2020	59	14	24.15	20613	100	16	32	43.80	0.234	0.425	0.659
Xeon-Silver-4116	2019	109	14	21.18	14660	85	12	24	37.23	0.289	0.508	0.797
Xeon-E5-2640v4	2016	54	14	19.08	12472	90	10	20	39.42	0.194	0.632	0.826
Xeon-E5-2640v3	2015	65	22	19.36	11118	90	8	16	39.42	0.183	0.709	0.892
Xeon-E5-2650v2	2014	36	22	09.44	9866	95	8	16	41.61	0.096	0.844	0.939
Xeon-E5-2620-v4	2017	30	14	13.47	9193	85	8	16	37.23	0.209	0.810	1.019
Xeon-Gold-6326	2021	68	10	101.0	35270	185	16	32	81.03	0.573	0.459	1.032
Xeon-E5640	2012	47	32	11.39	3782	80	4	8	35.04	0.251	1.853	2.104
Xeon-E5620	2010	52	32	12.71	3590	80	4	8	35.04	0.253	1.952	2.205
Xeon-E5-2609-v2	2014	22	22	10.49	3369	80	4	4	35.04	0.312	2.080	2.392
Xeon-X5647	2012	82	32	13.45	4441	130	4	8	56.94	0.253	2.564	2.818
Xeon-E5520	2010	25	45	12.12	2524	80	4	8	35.04	0.343	2.777	3.120
Xeon-E5410	2008	43	65	11.75	2007	80	4	4	35.04	0.365	3.492	3.857
Xeon-E5335	2007	28	65	13.45	1549	80	4	4	35.04	0.542	4.524	5.066
Xeon-E5310	2007	20	65	14.19	1306	80	4	4	35.04	0.639	5.366	6.005
Total	–	**740**	–	**17632.71**	**8261198**	**74045**	**6204**	**11956**	–	–	–	–

4 CASE STUDY: AN ACADEMIC DATACENTER

In the previous section, we used two simple servers to illustrate the effect of different metrics and how various server specifications, datacenter characteristics, and accounting considerations influence the *sunk carbon fallacy*. As a case study, we use an MIT academic datacenter that runs scientific computing workloads [33, 34].

Our case study demonstrates that the *sunk carbon fallacy* is not an artifact of our illustrative example; it manifests itself in real-world datacenters with an arbitrary set of servers. Our findings and analysis predicate the assumption that carbon-aware scheduling aims to reduce the total cluster-level carbon footprint (embodied and operational) of running a set of jobs on a given set of servers.

1 – Case Study Setup. We use the setup described in Section 3.1, except where noted below. Table 4 shows the detailed specifications of the servers in our real-world case study datacenter. Our datacenter inventory contains 15 different processor types across 740 servers. The average age of a server is 9.5 years: the oldest servers (E5310 and E5335) are 17 years old, the newest servers (Gold-6326) are just 3 years old, and only 236 out of 740 servers (31.9%) less than five years old. All processors are Intel-manufactured using technology nodes ranging from 64nm to 10nm. The embodied carbon of processors in servers ranges from 9.44 $KgCO_2$ to 101.0 $KgCO_2$ with a total of 17,633 $KgCO_2$ embodied carbon from processing component of servers. The PassMark score (multi-threaded rating) for the processors has a wide range: 1306 for E5310 (the oldest processor) to 35,270 for Gold-6326 (the newest processor). The TDP values are also at the extreme ends for the two processors, with 80W for E5310 and 185W for the Gold-6326.

We take a server's expected lifespan to be 5 years, which is generally true for modern datacenters. However, servers are typically kept operational in academic datacenters as purchasing new hardware necessitates considerations beyond performance and operating costs. We use two accounting approaches to amortize embodied carbon over a server's lifespan: first, the embodied carbon of servers older than five years is set to 0, and second, embodied carbon is amortized over the years they have been operational. In Table 4, the normalized values of operational carbon (O), embodied carbon (M), and software carbon intensity (SCI) use the latter approach. We choose the latter as setting embodied carbon to 0 for older servers artificially inflates the *sunk carbon fallacy*. We later show the impact of both approaches on the added carbon footprint.

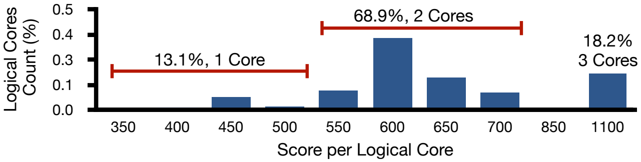

Figure 6: Normalized logical cores of similar performance.

We use job traces from a year-long trace from 2016 consisting of 14M jobs from the MGHPCC cluster used in prior work [4, 5]. The trace provides information on the job submission time, job end time, the requested number of cores, and the amount of memory requested (not used in our analysis). To get core allocations that provide roughly similar performance across these heterogeneous machines, we normalize the machines with thread count and categorize threads into three virtual core categories, as shown in Figure 6: **VC1** has 13.1% of threads with 250–500 score, **VC2** has 68.9% of threads with 550–700 score (2×**VC1**), and **VC3** has 18.2% of threads with (3×**VC1**). Since the biggest server in our case study datacenter has 32 threads, we filter all the jobs that require more than 32 cores.

2 – Case Study Findings. Table 4 shows SCI values for the servers sorted in ascending order of SCI, which is the energy-efficient ordering. For example, based on SCI metric, Xeon-E5-2620-v4 will be chosen before Xeon-Gold-6326 despite the former having 1.37× higher carbon footprint. Xeon-Gold-6326 is the second most efficient server, but it is 7th on the SCI ranking. The order is also suboptimal in other cases, such as Xeon-E5-2620-v4 and Xeon-E5-2650-v2, where an even less efficient server gets picked due to its lower embodied carbon. If the embodied carbon of servers over five years old is set to 0, the order is impacted even further. The three most efficient servers, Silver-4216, Gold-6326, and Silver-4116, will be ranked 2nd, 7th, and 4th, respectively.

While these ranking changes may seem minor, they can result in significant added carbon at the datacenter level when using SCI. To assess the datacenter-level impact, we compute the added

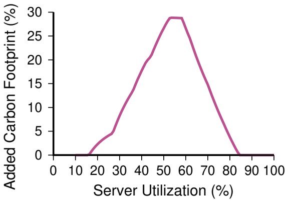

Figure 7: Embodied amortized across the lifespan.

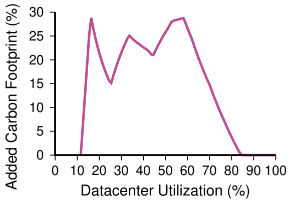

Figure 8: Embodied amortized during the first 5 years.

carbon under SCI and oSCI. To do that, we place the jobs on the servers based on their submission time. We do not replay the job trace and perform one-time job placement, akin to placing long-running jobs that never finish. Each job requires a certain number of virtual cores, and multiple jobs can be allocated to one server, which avoids stranded resources. A more realistic replay of job trace and placement is outside the scope of this work.

Figure 7 and Figure 8 show the added carbon due to SCI for the two approaches to amortize the embodied carbon for older servers. In both cases, using SCI results in nearly a 30% increase in carbon footprint for the datacenter due to the use of energy-inefficient servers. The added carbon for the first amortization approach is more than 5% when the datacenter utilization is between 27% and 78%, a range in which almost all datacenters operate. The second approach has an even higher added carbon cost (typically above 10%) across a broader utilization range of 13% to 80%. This demonstrates that even a slight change in the order of servers can significantly impact datacenter-level carbon. Furthermore, this result also shows how SCI is susceptible to an arbitrary setting of the expected lifespan. Finally, the cluster utilization in our job trace ranges from 40-80%; thus, using SCI will incur at >15% higher carbon footprint. Note that the first approach of amortizing over increasing periods beyond 5 years leads to double-counting of embodied carbon, as all embodied carbon would have been accounted for in 5 years.

5 IMPLICATIONS AND CONCLUSION

Next, we discuss the implications of using the three carbon-based metrics to schedule jobs on on-premise and cloud datacenters.

SCI quantifies the total carbon footprint of a functional unit, incorporating both operational and embodied emissions. SCI is an intuitive and comprehensive metric, but it is not well-suited for some decisions in sustainable computing. The metric implicitly requires that, for a server to be preferred over a reference, any increase in the server's embodied carbon footprint must reduce operational emissions by the same or a higher margin. However, since embodied carbon and operational carbon occur at different timescales, an arbitrary setting of the server's expected lifespan and embodied carbon accounting approach can perturb the embodied-to-operational carbon ratio. For example, as shown in Figures 4–8, different accounting approaches for embodied carbon and varying expected lifespans led to different operational carbon.

One key aspect of SCI is that it incentivizes using older hardware, which may have much lower embodied carbon per score (due to older and less energy-intense technology) than the newer servers with typically high embodied carbon. While the added carbon from smaller technology nodes has increased performance per unit area, it may not increase energy efficiency by the required margins for many processors. Therefore, as shown in Table 4, an older server can become an attractive alternative to a new server, especially when its embodied carbon has been amortized in its initial expected lifespan. In the worst case, it will sort servers from the oldest and least efficient (serving base demand at all times) to the newest and most efficient (only used for infrequent peak demand).

While SCI successfully incentivizes using older servers, it essentially provides an incentive to buy new servers, not use them, and use them only when they get old. We agree that the hardware should be used for longer and older servers should *serve* a purpose, but serving base demand using them is not a sustainable strategy. Older hardware should be kept, but its high operational carbon should only be accepted during peak demand. Using SCI to increase the operational carbon footprint is unnecessary and counter-intuitive. However, our analysis of metrics and case study findings requires that job scheduling decisions be decoupled from procurement decisions. When replacing an existing server, procurement teams can use SCI to compare existing servers against available options and purchase only the servers with a lower SCI. Note that the procurement for replacement differs from the procurement for new capabilities; if an emerging workload is critical but cannot be run on existing hardware, the purchase of new hardware will be SCI-agnostic. However, once a set of servers has been procured, their embodied carbon has been emitted; the only goal should be to reduce the operational cost of running the servers.

tSCI includes the datacenter-level embodied carbon and the operational carbon of the server running the job. This unified approach simplifies the allocation of carbon costs to users by aligning accounting and scheduling practices. However, embodied carbon estimates are highly uncertain due to variability in manufacturing processes, supply chain differences, and data quality issues. Relying on uncertain estimates for scheduling purposes risks making suboptimal decisions. Adding a noisy signal to an otherwise accurate accounting of operational carbon introduces errors that can cause incorrect resource allocation or prioritization decisions. Furthermore, as discussed in Section 3.3, accurately computing and tracking tSCI over time for large infrastructure, such as the public cloud, will incur significant overhead, prohibiting its use.

oSCI is the most effective metric for carbon-aware scheduling as operational carbon is the primary contributor in this scenario, and replacing existing hardware remains outside the scope at the timescales of scheduling decisions. Scheduling with oSCI focuses exclusively on reducing operational emissions since this is the only component that can be directly optimized for the procured hardware. Hardware replacement decisions affecting embodied carbon are orthogonal to workload scheduling and should not influence this process. Finally, using oSCI reduces the operational costs for the infrastructure, on-premise or cloud, as it picks the most efficient hardware and does not succumb to the *sunk carbon fallacy*.

ACKNOWLEDGEMENTS

We thank the SoCC reviewers and our shepherd, Timothy Zhu, for their valuable comments, which improved the quality of this paper, and WattTime for access to carbon intensity data. This work was supported in part by the NSF grants CNS-2325956, CNS-2105494, and CNS-2213636, the NSF CAREER Award CCF-2326182, the NSERC grants RGPIN-2021-03714 and DGECR-202100462, a Sloan Research Fellowship, and an Intel Research Award.

REFERENCES

[1] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. 2023. Carbon Explorer: A Holistic Framework for Designing Carbon Aware Datacenters. 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, 118–132. https://doi.org/10.1145/3575693.3575754

[2] Bilge Acun, Matthew Murphy, Xiaodong Wang, Jade Nie, Carole-Jean Wu, and Kim Hazelwood. 2021. Understanding Training Efficiency of Deep Learning Recommendation Models at Scale. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 802–814. https://doi.org/10.1109/HPCA51647.2021.00072

[3] Maria L. Alcaraz, Arash Noshadravan, Melissa Zgola, Randolph E. Kirchain, and Elsa A. Olivetti. 2018. Streamlined Life Cycle Assessment: A Case Study on Tablets and Integrated Circuits. *Journal of Cleaner Production* 200 (2018), 819–826. https://doi.org/10.1016/j.jclepro.2018.07.273

[4] Pradeep Ambati, Noman Bashir, David Irwin, and Prashant Shenoy. 2020. Waiting game: optimally provisioning fixed resources for cloud-enabled schedulers. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Atlanta, Georgia) *(SC '20)*. IEEE Press, Article 67, 14 pages.

[5] Pradeep Ambati, Noman Bashir, David Irwin, and Prashant Shenoy. 2021. Good Things Come to Those Who Wait: Optimizing Job Waiting in the Cloud. In *Proceedings of the ACM Symposium on Cloud Computing* (Seattle, WA, USA) *(SoCC '21)*. Association for Computing Machinery, New York, NY, USA, 229–242. https://doi.org/10.1145/3472883.3487007

[6] Luiz André Barroso and Urs Hölzle. 2007. The Case for Energy-Proportional Computing. *Computer* 40, 12 (2007), 33–37.

[7] Noman Bashir, Nan Deng, Krzysztof Rzadca, David Irwin, Sree Kodak, and Rohit Jnagal. 2021. Take it to the Limit: Peak Prediction-driven Resource Overcommitment in Datacenters. In *Proceedings of the Sixteenth European Conference on Computer Systems* (Online Event, United Kingdom) *(EuroSys '21)*. Association for Computing Machinery, New York, NY, USA, 556–573. https://doi.org/10.1145/3447786.3456259

[8] Noman Bashir, Priya Donti, James Cuff, Sydney Sroka, Marija Ilic, Vivienne Sze, Christina Delimitrou, and Elsa Olivetti. 2024. The Climate and Sustainability Implications of Generative AI. *An MIT Exploration of Generative AI* (March 27 2024). https://mit-genai.pubpub.org/pub/8ulgrckc.

[9] Noman Bashir, Tian Guo, Mohammad Hajiesmaili, David Irwin, Prashant Shenoy, Ramesh Sitaraman, Abel Souza, and Adam Wierman. 2021. Enabling Sustainable Clouds: The Case for Virtualizing the Energy System. In *Proceedings of the ACM Symposium on Cloud Computing (SoCC '21)*. Association for Computing Machinery, New York, NY, USA, 350–358. https://doi.org/10.1145/3472883.3487009

[10] Noman Bashir, David Irwin, and Prashant Shenoy. 2023. On the Promise and Pitfalls of Optimizing Embodied Carbon. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems* (Boston, MA, USA) *(HotCarbon '23)*. Association for Computing Machinery, New York, NY, USA, Article 15, 6 pages. https://doi.org/10.1145/3604930.3605710

[11] Noman Bashir, David Irwin, Prashant Shenoy, and Abel Souza. 2023. Sustainable computing-without the hot air. *ACM SIGENERGY Energy Informatics Review* 3, 3 (2023), 47–52.

[12] Anvita Bhagavathula, Leo Han, and Udit Gupta. 2024. Understanding the Implications of Uncertainty in Embodied Carbon Models for Sustainable Computing. In *HotCarbon Workshop on Sustainable Computer Systems*.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[14] Andrew Chien. 2023. Embodied Carbon (EC) is a Poor Architectural Metric, Round 2. https://www.sigarch.org/why-embodied-carbon-is-a-poor-architecture-design-metric-and-operational-carbon-remains-an-important-problem/.

[15] Andrew Chien. 2023. Why Embodied Carbon is a Poor Architecture Design Metric, and Operational Carbon Remains An Important Problem. https://www.sigarch.org/why-embodied-carbon-is-a-poor-architecture-design-metric-and-operational-carbon-remains-an-important-problem/.

[16] CPU-World. 2024. CPU-World: Microprocessors / CPUs. https://www.cpu-world.com/CPUs/CPU.html. Accessed October 2024.

[17] Robin Cubitt, Maria Ruiz-Martos, and Chris Starmer. 2012. Are bygones bygones? *Theory and decision* 73 (2012), 185–202.

[18] Peter J. Denning and Ted G. Lewis. 2017. Exponential Laws of Computing Growth. *Commun. ACM* 60, 1 (January 2017), 54–65.

[19] Daniel S. Berger et al. 2023. Reducing Embodied Carbon is Important. https://www.sigarch.org/reducing-embodied-carbon-is-important/.

[20] Green Software Foundation. 2021. Software Carbon Intensity (SCI) Specification. https://sci.greensoftware.foundation/. Accessed May 2024.

[21] Anshul Gandhi, Dongyoon Lee, Zhenhua Liu, Shuai Mu, Erez Zadok, Kanad Ghose, Kartik Gopalan, Yu David Liu, Syed Rafiul Hussain, and Patrick Mcdaniel. 2023. Metrics for Sustainability in Data Centers. *ACM SIGENERGY Energy Informatics Review* 3, 3 (2023), 40–46.

[22] Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. 2014. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition.* Cambridge University Press. Section 4.6: Dimensionality Reduction and Phase Plane Analysis.

[23] Viktor Gsteiger, Pin Hong Daniel Long, Yiran Jerry Sun, Parshan Javanrood, and Mohammad Shahrad. 2024. Caribou: Fine-Grained Geospatial Shifting of Serverless Applications for Sustainability. In *The 30th ACM Symposium on Operating Systems Principles (SOSP'24)*. ACM.

[24] Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. 2021. Chasing Carbon: The Elusive Environmental Footprint of Computing. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 854–867.

[25] Walid A. Hanafy, Qianlin Liang, Noman Bashir, David Irwin, and Prashant Shenoy. 2023. CarbonScaler: Leveraging Cloud Workload Elasticity for Optimizing Carbon-Efficiency. *Proc. ACM Meas. Anal. Comput. Syst.* 7, 3, Article 57 (dec 2023), 28 pages. https://doi.org/10.1145/3626788

[26] HEPiX. 2024. HS23 Scores. https://w3.hepix.org/benchmarking/scores_HS23.html. Accessed October 2024.

[27] Jonathan Koomey and Samuel Naffziger. 2015. Moore's Law Might be Slowing Down, but not Energy Efficiency. *IEEE spectrum* 52, 4 (2015), 35.

[28] Baolin Li, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. 2023. Clover: Toward Sustainable AI with Carbon-Aware Machine Learning Inference Service. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (<conf-loc>, <city>Denver</city>, <state>CO</state>, <country>USA</country>, </conf-loc>) *(SC '23)*. Association for Computing Machinery, New York, NY, USA, Article 20, 15 pages. https://doi.org/10.1145/3581784.3607034

[29] David Lo, Liqun Cheng, Rama Govindaraju, Luiz André Barroso, and Christos Kozyrakis. 2014. Towards Energy Proportionality for Large-scale Latency-critical Workloads. *ACM SIGARCH Computer Architecture News* 42, 3 (2014), 301–312.

[30] Diptyaroop Maji, Noman Bashir, David Irwin, Prashant Shenoy, and Ramesh K Sitaraman. 2024. The Green Mirage: Impact of Location-and Market-based Carbon Intensity Estimation on Carbon Optimization Efficacy. *arXiv preprint arXiv:2402.03550* (2024).

[31] Electricity Maps. 2022. Electricity Map. https://app.electricitymaps.com/map.

[32] Peter Mattson, Vijay Janapa Reddi, Christine Cheng, Cody Coleman, Greg Diamos, David Kanter, Paulius Micikevicius, David Patterson, Guenther Schmuelling, Hanlin Tang, Gu-Yeon Wei, and Carole-Jean Wu. 2020. MLPerf: An Industry Standard Benchmark Suite for Machine Learning Performance. *IEEE Micro* 40, 2 (2020), 8–16. https://doi.org/10.1109/MM.2020.2974843

[33] MIT. 2024. Bates Research and Engineering Center. https://bateslab.mit.edu/about/. Accessed October 2024.

[34] MIT. 2024. The Massachusetts Green High Performance Computing Center (MGHPCC). https://www.mghpcc.org/. Accessed October 2024.

[35] Mixeur. 2024. X86 CPU's Guide. https://www.x86-guide.net/en/cpu.html. Accessed October 2024.

[36] Ministry of Environment Taiwan. 2023. Taiwan's Emissions Increase Slightly in 2021, Well Below 2007 Peak. https://www.moenv.gov.tw/en/8A98EC390B973CB0/c55972e9-52dc-45f1-ad97-0619c5ccd2c7. Accessed October 2024.

[37] Elsa Olivetti and Randolph Kirchain. 2012. A Product Attribute to Impact Algorithm to Streamline IT Carbon Footprinting. In *International Symposium on Environmentally Conscious Design and Inverse Manufacturing*. Springer, 747–749.

[38] OpenBenchmarking Organization. 2024. OpenBenchmarking.org - Cross-Platform, Open-Source Automated Benchmarking Platform. https://openbenchmarking.org/. Accessed May 2024.

[39] Georgios Paschos. 2019. Time-scale Separation Principle. https://paschos.net/articles/time-scale-separation-principle/. Accessed May 2024.

[40] PassMark. 2024. CPU Benchmarks. https://www.cpubenchmark.net/. Accessed October 2024.

[41] Ana Radovanovic, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, Saurav Talukdar, Eric Mullen, Kendal Smith, Mariellen Cottman, and Walfredo Cirne. 2022. Carbon-Aware Computing for Datacenters. *IEEE Transactions on Power Systems* (2022), 1–1. https://doi.org/10.1109/TPWRS.2022.3173250

[42] Prashant Shenoy. 2024. Optimizing Embodied Emissions. *ACM SIGENERGY Energy Informatics Review* 4, 1 (2024), 1–2.

[43] 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 *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. ACM, 252–265.

[44] SPEC. 2024. CPU2017 Floating Point Speed. https://spec.cs.miami.edu/cpu2017/results/cfp2017.html. Accessed October 2024.

[45] GitHub Staff. 2022. GitHub Copilot. https://github.com/features/copilot. Accessed January 2024.

[46] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 3645–3650. https://doi.org/10.18653/v1/P19-1355

[47] Thanathorn Sukprasert, Abel Souza, Noman Bashir, David Irwin, and Prashant Shenoy. 2023. Quantifying the Benefits of Carbon-Aware Temporal and Spatial Workload Shifting in the Cloud. arXiv:2306.06502 [cs.DC]

[48] Zemin Sun, Geng Sun, Long He, Fang Mei, Shuang Liang, and Yanheng Liu. 2024. A Two Time-Scale Joint Optimization Approach for UAV-assisted MEC. arXiv:2404.04597 [eess.SY]

[49] Jennifer Switzer, Gabriel Marcano, Ryan Kastner, and Pat Pannuto. 2023. Junkyard Computing: Repurposing Discarded Smartphones to Minimize Carbon. In *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS 2023)*. ACM, 400–412. https://doi.org/10.1145/3575693.3575710

[50] Sean J. Taylor. 2020. Designing and Evaluating Metrics. https://medium.com/@seanjtaylor/designing-and-evaluating-metrics-5902ad6873bf.

[51] TechPowerUp. 2024. CPU Specs Database. https://www.techpowerup.com/cpu-specs/. Accessed October 2024.

[52] John Thiede, Noman Bashir, David Irwin, and Prashant Shenoy. 2023. Carbon Containers: A System-level Facility for Managing Application-level Carbon Emissions. In *Proceedings of the 2023 ACM Symposium on Cloud Computing*. 17–31.

[53] Muhammad Tirmazi, Adam Barker, Nan Deng, Md E. Haque, Zhijing Gene Qin, Steven Hand, Mor Harchol-Balter, and John Wilkes. 2020. Borg: the Next Generation. In *Proceedings of the Fifteenth European Conference on Computer Systems* (Heraklion, Greece) *(EuroSys '20)*. Association for Computing Machinery, New York, NY, USA, Article 30, 14 pages. https://doi.org/10.1145/3342195.3387517

[54] Amin Vahdat. 2024. Societal infrastructure in the age of Artificial General Intelligence. https://www.asplos-conference.org/asplos2024/main-program/.

[55] Jaylen Wang, Daniel S Berger, Fiodar Kazhamiaka, Celine Irvene, Chaojie Zhang, Esha Choukse, Kali Frost, Rodrigo Fonseca, Brijesh Warrier, Chetan Bansal, Jonathan Stern, Ricardo Bianchini, and Akshitha Sriraman. 2024. Designing Cloud Servers for Lower Carbon. In *Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA)*.

[56] Xiaorong Wang, Clara Na, Emma Strubell, Sorelle Friedler, and Sasha Luccioni. 2023. Energy and Carbon Considerations of Fine-Tuning BERT. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[57] Xinliang Wei, A B M Mohaimenur Rahman, Dazhao Cheng, and Yu Wang. 2023. Joint Optimization Across Timescales: Resource Placement and Task Dispatching in Edge Clouds. *IEEE Transactions on Cloud Computing* 11, 1 (2023), 730–744. https://doi.org/10.1109/TCC.2021.3113605

[58] WikiChip. 2024. WikiChip: Chips & Semi. https://en.wikichip.org/wiki/WikiChip. Accessed October 2024.

[59] World Resource Institute 1998. Greenhouse Gas Protocol. https://ghgprotocol.org/. Accessed May 2024.

[60] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, Michael Gschwind, Anurag Gupta, Myle Ott, Anastasia Melnikov, Salvatore Candido, David Brooks, Geeta Chauhan, Benjamin Lee, Hsien-Hsin Lee, Bugra Akyildiz, Maximilian Balandat, Joe Spisak, Ravi Jain, Mike Rabbat, and Kim Hazelwood. 2022. Sustainable AI: Environmental Implications, Challenges and Opportunities. In *Proceedings of Machine Learning and Systems*, D. Marculescu, Y. Chi, and C. Wu (Eds.), Vol. 4. 795–813. https://proceedings.mlsys.org/paper_files/paper/2022/file/462211f67c7d858f663355eff93b745e-Paper.pdf