Presented at the 1st Workshop on Hot Topics in Ethical Computer Systems (HotEthics 2024), held in conjunction with ASPLOS 2024

Intergenerational Embodied Carbon

Aster Plotnik
University of Toronto
Canada
amanda.plotnik@mail.utoronto.ca

Karthik Ganesan
University of Toronto
Canada
karthik.ganesan@mail.utoronto.ca

Natalie Enright Jerger
University of Toronto
Canada
enright@ece.utoronto.ca

Mark C. Jeffrey
University of Toronto
Canada
mcj@ece.utoronto.ca

ABSTRACT

The dawning of the age of accelerators has been a boon for computer architects and more broadly design and innovation in the computer hardware industry. However, as we spin and respin new accelerator designs, we must be cognizant of the broader sustainability implications of frequent redesign and accelerator churn. Specifically, we introduce the notion of lifetime carbon amortization and impact of legacy carbon due to frequent design refreshes. We argue for a more thoughtful and sustainable approach to accelerator design and deployment.

1 INTRODUCTION

Hardware specialization has become commonplace to enable modern applications with high computational intensity in the face of a Moore's Law slowdown. Relative to general-purpose CPUs, domain-specific accelerators have advantages such as decreased latency and increased energy efficiency [8]. However, domains that are sufficiently popular to warrant the cost and labor of specialization have applications that tend to evolve quickly. One such celebrity domain is deep learning, where new models quickly rise in popularity and create massive demand (e.g., transformers [14]). New models may not map well to accelerators designed for older models (e.g., CNNs). For example, in the Google TPU family of chips, custom hardware is added in each generation to support the latest model innovations [9]. For hyper-specialized hardware to keep up with this rapid application-driven change, new chips must be designed often enough to stay relevant.

Although acceleration offers well-known performance and energy benefits at run time, the environmental impact of deploying so many generations of accelerators remains unclear. This is concerning, as designers of specialized hardware make an implicit judgment that the application they target is worth the associated carbon footprint. Profitable models with high computational demands such as those for chatbots and recommendation systems are often deployed by large companies headquartered in the developed world. This leads to disconnect between those who benefit most from DNNs and those who are most directly harmed by the impacts of climate change in the developing world [7]. In this work, we discuss the carbon impact of frequently updating accelerators and provide recommendations for sustainable accelerator deployment.

2 BACKGROUND

The carbon emissions attributed to the lifetime of one computing device consist of two components: embodied and operational emissions [5]. Embodied emissions include those generated during the manufacturing of the device. Operational emissions stem from the usage of device, due to the CO_2 emitted by its power source. However, with datacenters shifting to more renewable energy sources, operational emissions are decreasing [6]. Major datacenter operators such as Google and Meta intend to reduce the carbon footprints of their operations. Meta already powers 100% of their datacenters with renewable energy [1], while Google aims to do so by 2030 [2]. Assuming these pledges are upheld, a device's embodied carbon footprint will dominate its operational carbon footprint.

The total carbon footprint of a device *per unit time* is a combination of its operational emissions and a fraction of the embodied emissions. The longer a device is in use, the lower the embodied emissions per unit time, as this component is amortized over time. This observation enables holistically analyzing carbon footprint over multiple generations.

3 LIFETIME CARBON AMORTIZATION

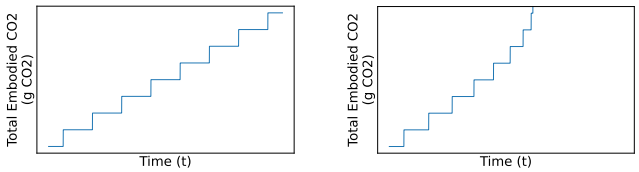
Embodied carbon has been viewed as a one-time cost incurred at manufacturing time. Consequently, its impact is difficult to analyze over the lifetime of the chip. Amortization can characterize the utility attained from the investment of embodied carbon.

3.1 Embodied carbon across chip generations

We present two ways of viewing embodied carbon emissions: *intragenerational* and *intergenerational* embodied carbon. The former is a one-time carbon emissions investment for a chip design that will be used for a given period of time. However, the domain using this chip may outlive the usefulness of that single chip generation; new chip generations that support increasing application demands lead to a *chip family*. Therefore, intragenerational embodied carbon is accumulated across multiple generations, resulting in the intergenerational embodied carbon of the chip family.

Fig. 1 illustrates the total (intergenerational) embodied carbon footprint over time for a chip family.[1] It increases with a step each time a new chip generation is manufactured. When new generations are manufactured at a regular cadence (Fig. 1a), the convex hull increases linearly. But increasingly rapid evolution of application

[1]If operational carbon emissions reach zero, then intergenerational embodied carbon emissions equal intergenerational total carbon emissions.



(a) Uniform generation releases (b) Increasingly frequent releases

Figure 1: Total embodied carbon over time as eight generations are manufactured on (a) a regular schedule and (b) an increasingly frequent schedule.

demands could provoke an increasingly rapid rate of new chip designs (Fig. 1b), causing intergenerational embodied carbon to increase superlinearly. The trend of increasing embodied emissions can be slowed by either reducing embodied carbon emissions in the manufacturing process or undergoing fewer manufacturing cycles by extending the lifetime of each chip. Ultimately, intergenerational embodied carbon increases monotonically.

3.2 Characterizing accelerators under replacement

Solely considering the embodied carbon of one device generation obscures the impact of frequent design refreshes. For example, several generations of Google TPUs were deployed only 1-3 years apart. Thus, characterizing the carbon of a single generation does not give the full picture, as frequently replaced devices have less time to amortize per-chip embodied carbon emissions.

Fig. 2 illustrates how embodied carbon is amortized over time when an accelerator is replaced. At the moment of deployment, the amortized carbon goes to infinity as the accelerator has not yet been operational for any time. As the accelerator operates, the embodied carbon per unit time is amortized as

$$\frac{1}{p} \cdot \frac{EC}{t}$$

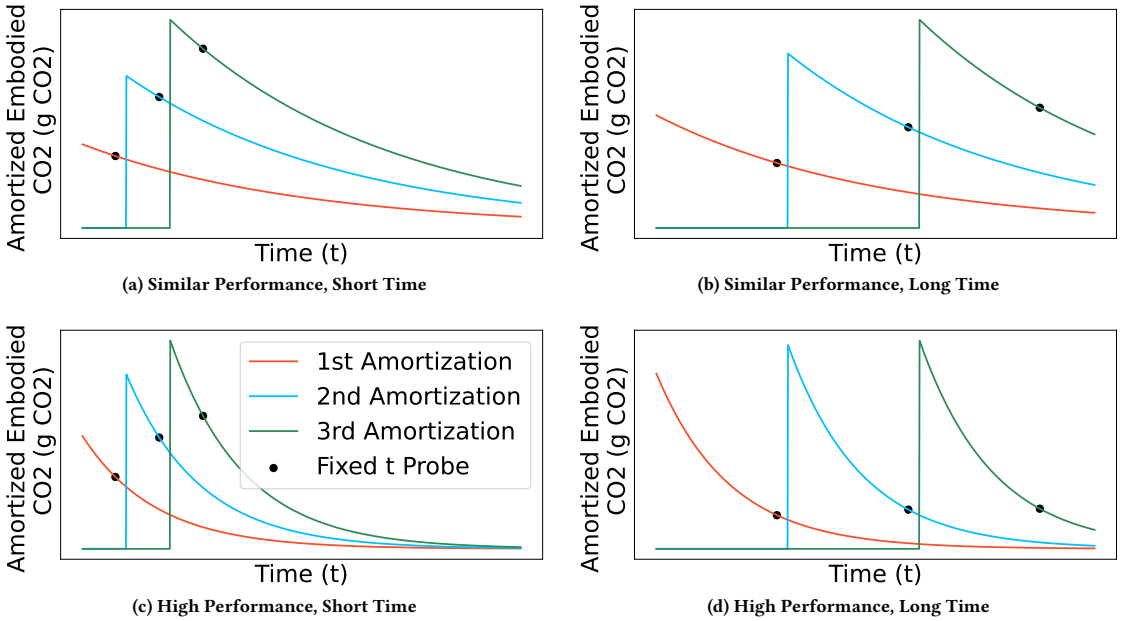
where EC is embodied carbon, t is time from deployment, and p is performance normalized to the previous generation. Performance affects the amortization decay rate. This metric has units of CO_2/operation. The four plots of Fig. 2 show the Cartesian product of *(i)* the time between manufacture of each generation, and *(ii)* performance of a new chip generation relative to its predecessor.[2]

Each amortization curve represents the embodied carbon when a new design is introduced. When a new accelerator replaces an existing one, this is an additional embodied carbon investment into the chip family's application domain (e.g., tensor processing). Therefore, future work performed by this new accelerator amortizes both its own and the previous generations' embodied carbon. This can be expressed as

$$\frac{EC_1}{p_1 \times t_1} + \frac{EC_2}{p_2 \times t_2} + \cdots + \frac{EC_n}{p_n \times t_n}$$

for each of n generations of a given accelerator family. Note that t_1 is time from deployment *of the first generation*, even if it is no longer in use, and similar for other terms. When the time between generations is short (Fig. 2(a) and (c)), the embodied carbon is not well amortized. In other words, subsequent generations of the

[2]We assume a simplification that the accelerator is always running at full utilization, recognizing this is unlikely in practice [15].



(a) Similar Performance, Short Time (b) Similar Performance, Long Time

(c) High Performance, Short Time (d) High Performance, Long Time

Figure 2: Amortized embodied carbon over time across 3 generations of designs with varying predecessor-relative performance and time to replacement.

accelerator that replace the capabilities of the previous generation still bear the legacy carbon cost of their predecessors, thus every generation will have an additional term when calculating its carbon footprint.

The more frequently accelerators are replaced, the more pronounced this problem becomes. The less time a first generation accelerator is given to amortize its embodied carbon investment, the higher the legacy carbon that will be passed to the second generation, and so forth. In Fig. 2, this phenomenon can be observed in the higher peaks of amortized embodied carbon when moving from one amortization to the next. The presence of a legacy carbon cost will slow the speed at which the following generation will amortize its embodied carbon per work. Since embodied carbon per use is amortized per unit time, it is impossible to completely eliminate legacy carbon no matter how long a previous generation is in use.

3.3 Fixed t Probe

Meaningfully weighing the impact of embodied carbon and performance to describe the effects of amortization is challenging because the maximum amortized embodied carbon value will always be infinity at time of deployment. In addition, generations beyond the first must consider relative impact of the previous generation, so the time cannot simply be factored out. In order to fairly compare the impact of amortization between generations, we use a fixed time value relative to each deployment (infinite impulse), and calculate each preceding generation's t value relative to the probe as necessary. This captures the effects of changes to embodied carbon cost, time, and legacy carbon. It also allows for modeling of accelerators which may not uniformly perform useful work over their entire lifespans. An accelerator with a lower fixed t probe value is more efficiently using its embodied carbon investment. Comparing Fig. 2(b) and (d), the higher-performance accelerator better amortizes the embodied carbon than the lower-performance accelerator at the fixed time t. The highly performant systems, probed at a later time t, have little legacy carbon to pass along to the next generation. In cases where embodied carbon cost per generation and performance are fixed, the fixed t probe value can only increase due to the cumulative nature of amortized embodied carbon. However, in some cases the fixed t probe increases more quickly between generations, indicating faster loss of the benefits of amortization. This simple check gives an estimate of relative amortization effects after each new deployment.

4 DISCUSSION

We provide examples of frequent accelerator replacement and discuss the sustainability of application acceleration. We also place our proposed lifetime amortization model in the context of prior work in carbon emissions modeling.

Frequent design refresh: A popular example of a specialized accelerator is the Google TPU for DNNs. Table 1 summarizes the relevant specifications for the different TPU generations. We see that the gap between generations is shrinking, highlighting the push for rapid design iterations. We also show additional capabilities added in each generation, to match the changing requirements of ML workloads. While the TPUv1 focused on general purpose DNN inference, subsequent generations add hardware support to speedup newer models. The TPUv2 added 'sparse cores' to speed up embeddings, which are common for natural language processing (NLP) tasks, while the TPUv4 specifically added support for accelerating LLMs. Thus, anyone wishing to obtain the fastest possible performance for new models would have to switch to the latest TPU generation. With very short development cycles of 1-3 years, these accelerators are meant to be replaced quickly. Thus, it is unclear whether each generation will be used long enough to offset their embodied carbon cost, even in the best case of 100% utilization of these designs. Factoring the lifetime amortization into design decisions may alter accelerator designs, leading to less churn and more sustainable solutions.

Table 1: Operations supported in hardware for Google TPU accelerators. For usage, I=Inference and T=Training.

	TPUv1 [10]	TPUv2/v3 [12]	TPUv4 Lite [9]	TPUv4 [9]
Year	2016	2019	2020	2022
Usage	I	T+I	I	T
Datatype	INT8	BF16	BF16	BF16
SparseCores		✓	✓	✓
Transformer		✓	-	-
↳ BERT			✓	✓
↳ LLM				✓

What applications are worth accelerating? In 2023, Meta released MTIA, their first hardware accelerator designed specifically for their in-house deep learning recommendation system (DLRS) [4]. MTIA is specialized for a single task, making it unlikely it will provide high performance on other tasks. Recommendation systems are extremely profitable (e.g., ~35% of Amazon's sales [11]). Thus, it is likely Meta will continue to use the best possible DLRS, updating both the software and hardware frequently. This is also driven by Meta evaluating their design using the total cost of ownership (TCO), which favours solutions that yield immediate returns verses those that are advantageous over the long term. The effects of this can be seen in practice, as Meta recently announced their second generation MTIA accelerator, less than a year after the announcement of the first generation [13]. Such an approach does not factor in any consideration to the environmental impact of designs which are solely meant to achieve the highest possible performance. Although new accelerators are likely to decrease operational carbon, considering increased renewables in datacenters lessens the impact of operational carbon as discussed earlier. Putting the embodied carbon into perspective encourages us to ask, do these types of accelerators provide enough (or any) societal good to be worth the environmental cost?

Existing models: Models that have been proposed to evaluate carbon footprint of devices include ACT [5] and FOCAL [3]. Both models focus on evaluating the carbon footprint of individual devices. The ACT model emphasizes a design philosophy that encourages designing leaner accelerators that are not over-provisioned for their QoS requirements. Reducing the complexity of an accelerator reduces area requirement, which is heavily correlated with embodied carbon footprint. The FOCAL model as well uses area as a proxy for embodied carbon footprint. However, given rapidly increasing demand for compute resources, it is unclear whether the advantages of over-provisioning for future-proofing a system are

adequately considered. An accelerator that is designed to minimally meet QoS requirements will run into the problem of more frequent replacement if is unable to keep up with changing trends. Thus, considering the context of an accelerator is a crucial component of evaluating carbon footprint. Models such as ACT and FOCAL should be used in conjunction with this work when evaluating sustainability of accelerators.

5 CONCLUSION

In this paper, we argue that increasingly specific accelerators that are quickly developed and target rapidly changing applications are not sustainable. Although we have used deep learning and recommendation systems as motivating examples for our argument, our observations and conclusions are relevant to other types of accelerators including those in robotics, cryptography, cryptocurrency, and more. As architects we have an opportunity to shape what applications are accelerated–in this process, we must ask ourselves who benefits from that acceleration and who bears the environmental costs of manufacturing accelerators. Ideally, a strong societal benefit should guide and shape our designs. Furthermore, we encourage architects to consider incorporation of some general purpose elements or reconfigurability or other hardware incentives for users to keep a specific generation of hardware in use for longer. Finally, while we have focused our discussion on environmental impacts of manufacturing, frequent accelerator replacement has another key environmental downside: generation of large quantities of e-waste. As future work, we plan to further elaborate on our model and incorporate other key considerations such as waste into our evaluation.

ACKNOWLEDGMENTS

We sincerely thank Victor Kariofillis, Jiaxiang Li, Jingyang Liu, Gilead Posluns, Iris Uwizeyimana, Guozheng Zhang, and Jiechen Zhao for their helpful feedback. The code for Fig. 1 was adapted from ChatGPT. We gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant RGPIN-2020-04179. This research was undertaken, in part, thanks to funding from the Canada Research Chairs program and through the support of the University of Toronto McLean Award.

REFERENCES

[1] 2023. 2023 Meta sustainability report. Meta Sustainability. (2023). https://sustainability.fb.com/wp-content/uploads/2023/07/Meta-2023-Sustainability-Report-1.pdf.

[2] 2022. Carbon-free energy performance at google data centers (2021). Google Sustainability Resources. (June 2022). https://www.gstatic.com/gumdrop/sustainability/2021-carbon-free-energy-data-centers.pdf.

[3] Lieven Eeckhout. 2024. FOCAL: a first-order carbon model to assess processor sustainability. In *Proc. 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. DOI: 10.1145/3620665.3640415.

[4] Amin Firoozshahian, Joel Coburn, Roman Levenstein, Rakesh Nattoji, Ashwin Kamath, Olivia Wu, Gurdeepak Grewal, Harish Aepala, Bhasker Jakka, Bob Dreyer, Adam Hutchin, Utku Diril, Krishnakumar Nair, Ehsan K. Aredestani, Martin Schatz, Yuchen Hao, Rakesh Komuravelli, Kunming Ho, Sameer Abu Asal, Joe Shajrawi, Kevin Quinn, Nagesh Sreedhara, Pankaj Kansal, Willie Wei, Dheepak Jayaraman, Linda Cheng, Pritam Chopda, Eric Wang, Ajay Bikumandla, Arun Karthik Sengottuvel, Krishna Thottempudi, Ashwin Narasimha, Brian Dodds, Cao Gao, Jiyuan Zhang, Mohammed Al-Sanabani, Ana Zehtabioskuie, Jordan Fix, Hangchen Yu, Richard Li, Kaustubh Gondkar, Jack Montgomery, Mike Tsai, Saritha Dwarakapuram, Sanjay Desai, Nili Avidan, Poorvaja Ramani, Karthik Narayanan, Ajit Mathews, Sethu Gopal, Maxim Naumov, Vijay Rao, Krishna Noru, Harikrishna Reddy, Prahlad Venkatapuram, and Alexis Bjorlin. 2023. MTIA: first generation silicon targeting meta's recommendation systems. In *Proc. 50th ACM/IEEE International Symposium on Computer Architecture (ISCA)*. DOI: 10.1145/3579371.3589348.

[5] 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 *Proc. 49th ACM/IEEE International Symposium on Computer Architecture (ISCA)*. DOI: 10.1145/3470496.3527408.

[6] 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 *Proc. IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. DOI: 10.1109/HPCA51647.2021.00076.

[7] Alexa Hagerty and Igor Rubinov. 2019. Global AI ethics: a review of the social impacts and ethical implications of artificial intelligence. *arXiv preprint arXiv:1907.07892*. DOI: 10.48550/arXiv.1907.07892.

[8] John L. Hennessy and David A. Patterson. 2019. A new golden age for computer architecture. *Communications of the ACM*, 62, 2, 48–60. DOI: 10.1145/3282307.

[9] Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Clifford Young, Xiang Zhou, Zongwei Zhou, and David A Patterson. 2023. TPU v4: an optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *Proc. 50th ACM/IEEE International Symposium on Computer Architecture (ISCA)*. DOI: 10.1145/3579371.3589350.

[10] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. 2017. In-datacenter performance analysis of a tensor processing unit. In *Proc. 44th ACM/IEEE Annual International Symposium on Computer Architecture (ISCA)*. DOI: 10.1145/3079856.3080246.

[11] Ian MacKenzie, Chris Meyer, and Steve Noble. 2013. How retailers can keep up with consumers. McKinsey Insights. (Oct. 2013). https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers.

[12] Thomas Norrie, Nishant Patil, Doe Hyun Yoon, George Kurian, Sheng Li, James Laudon, Cliff Young, Norman Jouppi, and David Patterson. 2021. The design process for Google's training chips: TPUv2 and TPUv3. *IEEE Micro*, 41, 2, 56–63. DOI: 10.1109/MM.2021.3058217.

[13] Eran Tal, Nicolaas Viljoen, Joel Coburn, and Roman Levenstein. 2024. Our next-generation meta training and inference accelerator. AI at Meta Blog. (Apr. 2024). https://ai.meta.com/blog/next-generation-meta-training-inference-accelerator-AI-MTIA/.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. 31st Conference on Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[15] Ganesh Venkatesh, Jack Sampson, Nathan Goulding, Saturnino Garcia, Vladyslav Bryksin, Jose Lugo-Martinez, Steven Swanson, and Michael Bedford Taylor. 2010. Conservation cores: reducing the energy of mature computations. In *Proc. 15th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 205–218. DOI: 10.1145/1736020.1736044.