Characterizing Datacenter Server Generations for Lifetime Extension and Carbon Reduction

Jaylen Wang
Carnegie Mellon University
jaylenw@andrew.cmu.edu

Udit Gupta
Cornell University
ugupta@cornell.edu

Akshitha Sriraman
Carnegie Mellon University
akshitha@cmu.edu

I. INTRODUCTION

Scientists have made it clear that anthropogenic climate change is one of the greatest threats to both global health and the ecosystem of the planet [2], [9]. Tackling climate change will require multiple disciplines to come together to introduce novel and comprehensive solutions to reduce society's negative environmental impact. As such, researchers, designers, and maintainers of computer systems must also embrace this responsibility. Notably, sustainability efforts targeted towards datacenters (DCs) are important yet lacking, despite recent studies estimating that 2% of global emissions are a result of information and communication technology (ICT) [8]. Projections show that ICT could constitute 20% of anthropogenic emissions by 2030, with much of the increase attributable to an increased demand and capacity for DC computing [7].

To reduce emissions and waste, sustainable resource management has focused on ways to reduce, reuse, and recycle common resources such as oil and plastics; similar strategies are understudied for managing DC server hardware, where current practice is to dispose of server hardware every three to four years [3]. Minimizing hardware waste is especially crucial; previous work has shown that up to 50% of DC system emissions are "embodied" emissions, which result from the manufacturing and transport of server hardware [6].

Previous research has highlighted the issue of embodied emissions and the potential benefits of extending server lifetimes in DCs [5], [6], but few works have focused on practical methods for extending server lifetimes in a DC setting. Modern DC services are increasingly built using microservice architectures, wherein a complex web service is composed of numerous distributed microservices such as HTTP connection termination, key-value serving, query rewriting, access-control management, and protocol routing.

This work takes an initial step towards understanding how older hardware can be reused in a DC setting while preserving end-to-end service performance (i.e., tail latency). We analyze the impacts on performance when running all and parts of a microservice-based service on older server generations. From the results of our experiments, we identify specific operating regions where older hardware does not degrade, or can even improve, service performance in comparison to running the entire service on newer servers. For instance, we observe that older hardware can achieve better performance, as measured in latency, than newer hardware under low load conditions. We also find that certain microservice types as well as regions of a microservice call graph are more sensitive to being run on older hardware.

This study shows that there are "carbon inefficiencies" in current DC resource management strategies, as existing strategies can become more carbon efficient by scheduling on older hardware when appropriate. We motivate the need to explore scheduling applications and microservices while considering embodied and operational characteristics of hardware generations. Our work provides concrete insights into ways in which a system could leverage microservices' tolerance to older hardware generations to prevent environmentally costly server refreshes and hardware waste.

II. CHARACTERIZING SERVER GENERATIONS

To characterize how common DC services perform on different hardware generations, we run experiments comparing the performance of a microservice application on two generations of Intel and AMD servers. For each hardware vendor, the servers only differ by generation, so they are of the same SKU. The servers are all located in CloudLab DCs and accessed remotely [4]. The server characteristics are summarized in Table I. We study the end-to-end *Social Network* application in DeathStarBench. The benchmark allows users to create posts with text and images that are processed and filtered. This functionality is implemented as thirty core microservices that communicate with each other through Apache Thrift RPCs [1].

To account for variations in system setups among different server generations in CloudLab, we augment DeathStarBench with an experimental infrastructure that enables a more apples-to-apples and reproducible comparison for our experiments. The infrastructure allows us to evenly distribute the thirty microservices across the number of nodes under test, pin the microservice to the CPUs on a single socket of its assigned server, and keep the microservices that are colocated together constant between experiments. Each microservice is also constrained to the same amount of RAM, by using the RAM capacity of the most constrained server being compared and dividing the capacity evenly among the microservices colocated on the server. For all experiments, we use an open-loop load generator and sweep across low to high queries-

	Intel		AMD	
	Xeon E5-2660 v2	**Xeon E5-2660 v3**	**EPYC 7542**	**EPYC 7543**
Microarchitecture	Ivy Bridge (2012)	Haswell (2013)	Rome (2019)	Milan (2021)
Cores/Threads	10/20	10/20	32/64	32/64
Node	22 nm	22 nm	7 nm	7 nm
Base/Turbo (GHz)	2.2 / 3	2.6 / 3.3	2.9 / 3.4	2.8 / 3.7
LLC Cache Size	25 MB	25 MB	128 MB	256 MB
TDP (W)	95	105	225	225
RAM (DDR4)	256GB (1.6 GHz)	160GB (2.133 GHz)	256GB (3.2 GHz)	512GB (3.2 GHz)
Disk (SATA)	2 TB HDD	480 GB SSD	1.6 TB SSD	2 TB SSD
NIC	10Gb (PCIe v3)	10 Gb (PCIe v3)	25 Gb (PCIe v4.0)	25 Gb (PCIe v4.0)

TABLE I

CHARACTERISTICS OF TWO GENERATIONS (OLD ON THE LEFT, NEW ON THE RIGHT) OF INTEL AND AMD SERVERS USED IN EXPERIMENTS.

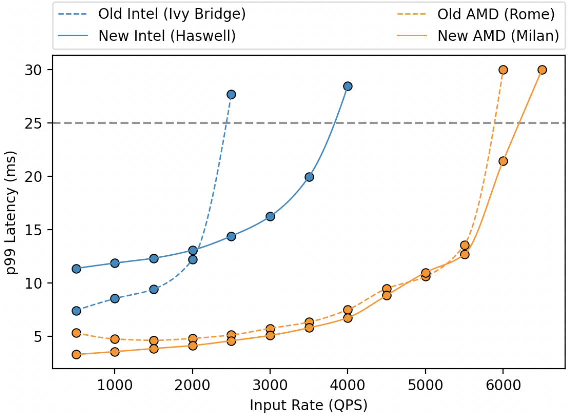

Fig. 1. Tail ($99^{th}\%$) latency across different load conditions (in QPS) for older and newer Intel and AMD server SKUs. Older Intel servers outperform newer ones at low load.

per-second (QPS) conditions until a saturation throughput is reached, while recording median and tail latencies.

A. End-to-End Service Characterization Across Server Generations

The first experiment aims to gain a coarse understanding of how an end-to-end DC service behaves on different server generations. The thirty microservices are distributed across eight nodes of the server type (i.e., same SKU and generation). The resulting plot of latency against increasing QPS is shown in Fig. 1. Latency above the dotted line cannot meaningfully be measured as the system is under saturation, where the offered load is unsustainable and queuing delays grow unbounded.

The results for Intel servers in Fig. 1 show that at lower load conditions, the service performs better when running on older server generations. This result indicates that upgrading and using newer server generations to serve lower load regions is an unnecessary carbon inefficiency, as older servers could perform the same or better. Hence, it is worthwhile to explore if scheduling services on older hardware under lower loads can save embodied carbon emissions.

B. Microservice-Based Characterization Across Server Generations

While Fig. 1 provides information on the tolerance of the end-to-end service when running on older hardware, it does not provide information on the tolerance of specific microservices or groups of microservices. To gain this insight, we conduct a set of experiments where all microservices were initially placed on a set of fifteen newer servers under the same experimental setup as in §II-A. Then, we place one microservice on an older server, while keeping all other microservices in the same configuration on the newer nodes. We perform a QPS sweep and repeat for each of the thirty microservices. We report our results on only the two generations of AMD nodes for brevity. Fig. 2 shows the results of this experiment for a representative sample of microservices. Microservices that are on the same call path are colored the same. “All New” shows resulting latencies when running all microservices on newer AMD nodes, while other markers show latencies when that named microservice is scheduled on an older AMD node (keeping all other microservices on the newer nodes). By comparing the performance of configurations with a specific microservice placed on an older server to the performance of the ”All New” configuration, the experiment indicates the effect on the end-to-end service latencies of placing a certain microservice on older hardware.

The data shows that certain microservices, such as the *user-timeline-service* and *user-timeline-mongodb*, are less tolerant to being scheduled on older hardware as it shows consistently higher latencies in comparison to “All New”. On the other hand, microservices, such as the *media-service* and *media-mongodb*, are more tolerant. One of the benefits of the microservice model is that individual components of the service can be optimized and handled independently. While the microservice model is often exploited for performance, the imbalance between microservices in tolerances to being placed on older hardware suggests there is also room for optimizing individual microservice scheduling for improved carbon efficiency.

Additionally, the data shows that groups of microservices that are on the same call path, even if they consist of different underlying functionalities, exhibit similar tolerances to older hardware placement. This result suggests that a defining factor in a microservice’s tolerance to older server placement is its location in the service call graph. Further research is needed to determine the specific features of call graph regions that make them more amenable to placement on older nodes.

III. POTENTIAL CARBON SAVINGS

To achieve real-world reductions in carbon emissions and hardware waste, the observations discussed in §II can be used to inform an end-to-end system that explores optimal scheduling and placement strategies that consider performance and carbon tradeoffs to enable extended hardware lifetimes.

To evaluate these lifetime extension policies, future work will need to develop ways to accurately account for operational and embodied carbon emissions reductions. Operational emissions can be accounted for by tracking the power usage of different system choices (i.e., power when scheduling on older vs. newer hardware). One way we have explored to account for embodied emissions for a system that promotes hardware reuse is to tackle it from a capacity planning perspective.

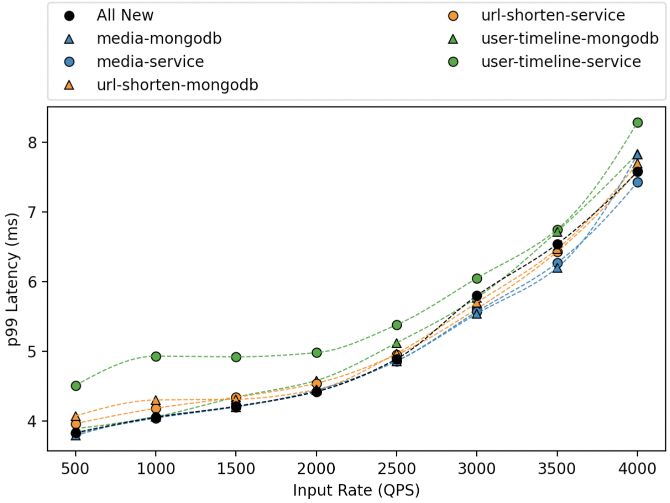

Fig. 2. Impact on end-to-end service tail ($99^{\text{th}}\%$) latency of placing certain single microservices on older hardware. Certain microservices are more (*media-service*) and less (*user-timeline-service*) tolerant, as measured by impact on end-to-end service latency, to placement on older hardware.

We can model the capacity planning of a DC as taking a point at which X servers are going to be upgraded, as part of a standard server refresh cycle. However, with a lifetime extension system that we propose and implement, we could keep a fraction F, where $0 < F < 1$, of X in production by running part or all of the services on the current (older) nodes. In this case, we could then say the embodied emissions that we saved would be $E_{new} * F * X$, where E_{new} is the embodied carbon of the new server. This accounting would be an underestimate of environmental improvements caused by the system, as it does not account for the hardware waste prevented by extending the server lifetime.

REFERENCES

[1] "Apache Thrift - Home." [Online]. Available: https://thrift.apache.org/

[2] L. Atwoli, A. H. Baqui, T. Benfield, R. Bosurgi, F. Godlee, S. Hancocks, R. Horton, L. Laybourn-Langton, C. A. Monteiro, I. Norman, K. Patrick, N. Praities, M. G. Olde Rikkert, E. J. Rubin, P. Sahni, R. Smith, N. Talley, S. Turale, and D. Vázquez, "Call for Emergency Action to Limit Global Temperature Increases, Restore Biodiversity, and Protect Health," *New England Journal of Medicine*, vol. 385, no. 12, pp. 1134–1137, Sep. 2021, publisher: Massachusetts Medical Society _eprint: https://doi.org/10.1056/NEJMe2113200. [Online]. Available: https://doi.org/10.1056/NEJMe2113200

[3] L. A. Barroso, J. Clidaras, and U. Hölzle, "The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second edition," *Synthesis Lectures on Computer Architecture*, vol. 8, no. 3, pp. 1–154, Jul. 2013, publisher: Morgan & Claypool Publishers. [Online]. Available: https://www.morganclaypool.com/doi/abs/10.2200/S00516ED2V01Y201306CAC024

[4] D. Duplyakin, R. Ricci, A. Maricq, G. Wong, J. Duerig, E. Eide, L. Stoller, M. Hibler, D. Johnson, K. Webb, A. Akella, K. Wang, G. Ricart, L. Landweber, C. Elliott, M. Zink, E. Cecchet, S. Kar, and P. Mishra, "The design and operation of cloudlab," in *Proceedings of the 2019 USENIX Conference on Usenix Annual Technical Conference*, ser. USENIX ATC '19. USA: USENIX Association, Jul. 2019, pp. 1–14.

[5] U. Gupta, M. Elgamal, G. Hills, G.-Y. Wei, H.-H. S. Lee, D. Brooks, and C.-J. Wu, "ACT: designing sustainable computer systems with an architectural carbon modeling tool," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ser. ISCA '22. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 784–799. [Online]. Available: https://doi.org/10.1145/3470496.3527408

[6] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu, "Chasing Carbon: The Elusive Environmental Footprint of Computing," Oct. 2020, arXiv:2011.02839 [cs]. [Online]. Available: http://arxiv.org/abs/2011.02839

[7] N. Jones, "How to stop data centres from gobbling up the world's electricity," *Nature*, vol. 561, no. 7722, pp. 163–166, Sep. 2018, bandiera_abtest: a Cg_type: News Feature Number: 7722 Publisher: Nature Publishing Group Subject_term: Energy, Engineering, Research data, Computer science. [Online]. Available: https://www.nature.com/articles/d41586-018-06610-y

[8] B. Knowles, *ACM TechBrief: Computing and Climate Change*. New York, NY, USA: Association for Computing Machinery, 2021.

[9] V. Masson-Delmotte, P. Zhai, A. Pirani, S. L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. B. R. Matthews, T. K. Maycock, T. Waterfield, Yelekçi, R. Yu, and B. Zhou, Eds., *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press, 2021.