Bringing Carbon Awareness to Multi-cloud Application Delivery

Diptyaroop Maji[1], Ben Pfaff [2+], Vipin P R[3], Rajagopal Sreenivasan[3]
Victor Firoiu[3], Sreeram Iyer[3], Colleen Josephson[4+], Zhelong Pan[3], Ramesh K. Sitaraman[1]

[1]University of Massachusetts Amherst, [2]Feldera, [3]VMware, [4]University of California Santa Cruz

ABSTRACT

Data centers consume roughly 1–2% of the world's electricity, with the majority of it attributed to compute, making the computing industry a substantial source of greenhouse gas emissions. Resources in data centers typically focus on providing high performance and availability, but the question of sustainability in managing these distributed resources often goes unnoticed over these other metrics. This problem will only exacerbate as the data center computing demand continues to increase.

In this paper, we address the sustainability aspect of load balancing in VMware's Avi Global Server Load Balancer (GSLB). Our GSLB deployment spans data centers across geographies and clouds and relies on geographical proximity to shift client application requests to the closest data center. In this work, we enhance the GSLB service to additionally consider the real-time carbon intensity at each data center as a factor in making a load-balancing choice. Our carbon-aware prototype shows an average of 21% and a maximum of 51% reduction in carbon emissions while operating with an acceptable latency.

CCS CONCEPTS

• **Social and professional topics** → **Sustainability**; • **Computer systems organization** → **Cloud computing**.

KEYWORDS

spatial load balancing, marginal carbon intensity, stateless workloads, data center computing

ACM Reference Format:
Diptyaroop Maji[1], Ben Pfaff [2+], Vipin P R[3], Rajagopal Sreenivasan[3] and Victor Firoiu[3], Sreeram Iyer[3], Colleen Josephson[4+], Zhelong Pan[3], Ramesh K. Sitaraman[1]. 2023. Bringing Carbon Awareness to Multi-cloud Application Delivery. In *2nd Workshop on Sustainable Computer Systems (HotCarbon '23), July 9, 2023, Boston, MA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3604930.3605711

[+]Research performed while at VMware.

HotCarbon '23, July 9, 2023, Boston, MA, USA

ACM ISBN 979-8-4007-0242-6/23/07…$15.00
https://doi.org/10.1145/3604930.3605711

1 INTRODUCTION

Data centers consume an estimated 1–2% of global electricity production [1], resulting in significant carbon emissions in the atmosphere and contributing to global warming. Data centers in the United States alone directly use about 40,000 GWh/year [3]. The data center electricity demand is only expected to rise in the near future [10] due to an accelerated computation demand leading to data center expansions, thus exacerbating the carbon emission problem. As data centers typically source electricity from the local grid, the carbon emission associated with data center computing depends on the region where a particular data center resides. A region generates electricity using a mixture of renewable (e.g., solar) and non-renewable (e.g., coal) sources, which vary over time as the demand changes. For example, zero-carbon solar generation may dominate during the day, while high-carbon natural gas taking over at night. The sources generating electricity also vary across regions. For example, electricity in Quebec primarily comes from zero-carbon hydropower, while in Poland, it is mainly generated from high-carbon coal. Thus, data centers residing in regions with a high non-renewable source percentage may emit more carbon into the atmosphere than a data center in a green region for the same amount of computing.

To tackle the carbon emissions generated due to data center computing, organizations are moving towards carbon-neutral [19], carbon-negative [15] or even carbon-free [9] operations. For example, VMware is working towards achieving net-zero carbon emissions for its operations and supply chain by 2030 [21]. Consequently, there has been an influx of carbon-aware techniques to reduce carbon emissions due to data center computing. Many computing loads are delay tolerant (e.g., batch workloads) and can be delayed to greener hours. For example, Google now shifts execution of load to greener times of the day [18]. However, a significant portion of data center workloads are latency sensitive (e.g., web requests) and need to be executed as soon as possible. Although these workloads cannot be shifted in time, we can leverage the fact that these workloads are usually deployed in multi-cloud environments with geographical fault tolerance. The availability of geographically distributed data centres that can serve these workloads enables us to spatially redirect such workloads to greener regions. When done right, we expect the solution to honor latency constraints and not degrade the user experience while at the same time reducing the carbon impact.

Research contribution. In this paper, we study how VMware's Avi Global Server Load Balancer (GSLB) [20] can help reduce its customers' environmental impacts when serving latency-sensitive workloads. Avi GSLB serves state-of-the-art applications that require high availability. These applications are deployed in multiple geographically distributed data centers which are either private,

public, or a hybrid of both. It balances loads at two different levels: between data centers using DNS and within a data center using a distributed data plane. We attempt to show how refining our GSLB data center selection algorithm can reduce the carbon impact of a distributed service without degrading user experience. Specifically, we make the following contributions:

- We describe the carbon-aware optimization that now considers the real-time carbon intensity of different regions as another factor in selecting a data center to serve customer requests in addition to the incumbent client-to-server distance.
- We describe the design of our GSLB and develop a prototype that makes carbon-aware decisions and redirects stateless workloads to greener regions. Our approach shows an average 21% carbon reduction over incumbent algorithms when subjected to typical customer workloads.

Roadmap. The rest of the paper is as follows: Section 2 provides background on carbon metrics and motivates the need for carbon-aware spatial load balancing. Section 3 describes related work. Section 4 describes the design of our modifications and justifies our choice of carbon intensity metric. Section 5 evaluates the prototype that we built. Section 6 discusses the limitations of our prototype and future challenges of carbon-aware spatial load balancing. Section 7 gives our next steps. Finally, Section 8 concludes the paper.

2 BACKGROUND AND MOTIVATION

In this section, we provide a brief background on the carbon emissions when electricity is generated from various renewable and non-renewable sources and the carbon intensity associated with electricity generation and consumption. We also motivate the need to consider the carbon intensities of different regions during spatial load balancing to reduce the carbon emissions from computing.

Carbon emission factors. The *carbon emission factor* (CEF) of a source is the amount of operational carbon emissions attributable to a source of energy, in grams of carbon dioxide (or its equivalent) per kilowatt-hour, written gCO_2eq/kWh. *Direct CEF* measures operational emissions when a source is converted to a unit of generated electricity. Renewable sources like solar, wind, etc., emit no carbon when generating electricity, so their direct CEFs are zero. Non-renewable sources have nonzero CEFs that vary based on the type of plant and other factors. In this paper, we use the median emission factor obtained from [4, 16] for each such source. *Lifecycle CEF* of a source measures operational emissions and infrastructural emissions upstream/downstream. Renewable sources have nonzero lifecycle CEFs due to manufacturing, distribution, maintenance, end-of-life disposal, etc.

Carbon intensity. The *carbon intensity (CI)* for a region is defined as the amount of carbon emitted per unit of electricity generated/consumed in that region, in gCO_2eq/kWh. There are two types of CIs:

- *Average carbon intensity (ACI)* is the average of the CEFs of each electricity-generating source in the region, weighted by the fraction of electricity produced by each source. Whether to use lifecycle or direct CEF for calculating ACI is up to a carbon optimization system to choose and justify.

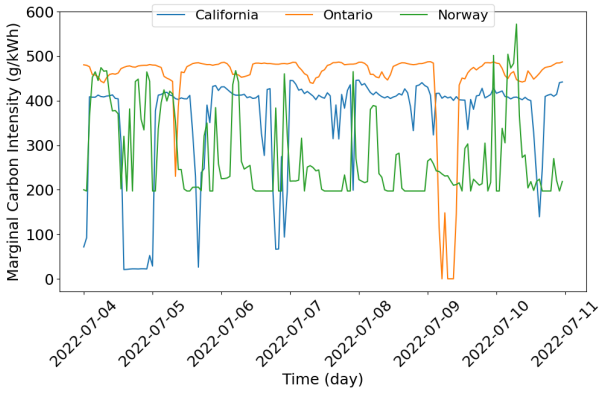

Figure 1: Spatial variations in MCI across different regions. Some regions show a high variability whereas MCI in other regions is fairly constant.

- *Marginal carbon intensity (MCI)* is the weighted average of the CEFs of sources that are on the margin, i.e., the subset of sources responsible for generating an additional unit of electricity to meet a new load to the grid at any given time. Direct CEFs are used, since MCI only considers incremental electricity generation.

Consider a simplified grid with equal solar (zero CI) and coal (say, CI = 1000 gCO_2eq/kWh). Then, the ACI is 500 gCO_2eq/kWh. However, if imposing an additional kWh load would be generated from coal, the MCI is 1000 gCO_2eq/kWh; if it would be generated from solar, the MCI is zero. Real power grids usually have multiple sources on the margin.

Global Server Load Balancer (GSLB). The VMware Avi load balancer influences the choice of data center via DNS. A client's HTTP or other request looks up a DNS name specific to a particular service. The DNS server that responds to it is provided by Avi's GSLB, which chooses one of the available data center IPs for the reply. Most of the application load balancers deployed with GSLB are designed so that content can be served optimally from any data center. This stateless paradigm simplifies the design and paves the way for scalable, highly available applications.

Motivation. We observe two things while building a carbon-aware load balancer. First, the sources and the fraction of electricity a specific source generates vary across regions. For example, Texas is heavily powered by coal, whereas Norway has a high use of renewables. Consequently, both the ACI and the MCI vary spatially and enable the opportunity to shift loads to greener regions (with lower ACI or MCI) to reduce carbon emissions. Figure 1 shows how MCI varies across three regions in North America and Europe over a week in July (data provided by Watttime [22]). Whether to use ACI or MCI is an open research problem; section 4.1 explains our design choice.

Second, most requests that are load balanced in enterprise networks are short-lived. The lifetime of a typical HTTP request ranges from a few milliseconds to, at most, tens of minutes.

We build on the above observations and evaluate the carbon gains realized by load-balancing stateless, short-lived requests across data

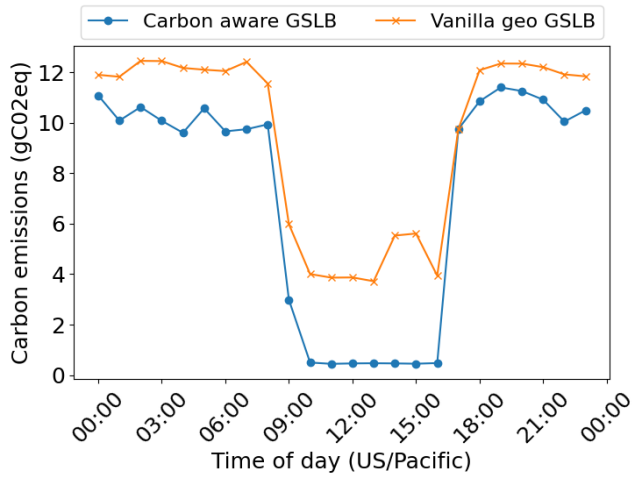

Figure 2: Carbon emissions in simulation with and without carbon-aware load balancing.

centers. We built a simple simulation to further motivate our work by estimating the potential for reducing carbon impact through load balancing. Our simulation achieves a compromise between latency (approximated using the distance between data centers and clients) against the carbon intensity at each data center. In the simulation, clients distributed across the United States send HTTP traffic and choose a server in California or Massachusetts during each hour of a simulated 24-hour period, using real historical data from the WattTime [22]. California had very low carbon intensity via solar power during the sunniest hours of the day. In this period, the carbon-aware GSLB directed most client requests to California, resulting in near-zero carbon emissions. Figure 2 shows the simulation results, which show that the carbon-aware GSLB always emits less carbon for the same amount of compute, resulting in an average of 36% reduction in carbon emissions over the day.

3 RELATED WORK

Gao et al. [7], Doyle et al. [5], and Zhou et al. [26] optimize carbon emissions along with other factors like electricity cost, service level agreements (SLAs), access latency etc., by routing client requests across data centers. Our algorithm is inspired by these works and optimizes carbon emissions and load-shifting distance. [7, 26] run the optimization periodically while we do it online on a per-request basis. We also consider MCI for carbon awareness because MCI is potentially a better metric for shifting short-lived workloads spatially (see Section 4.1), whereas the above works consider ACI. Our algorithm is also implemented from a DNS and GSLB perspective — we send back the IP of the server residing in the optimal data center as a DNS response to the client, who then sends the actual request to the optimal data center. We have limited knowledge about the actual workload. In contrast, other works redirect actual client loads in their optimization.

Recently, Zheng et al. [25] proposed migrating workloads to regions with curtailed renewable energy. In contrast, for a client request, we only select the data centers which can serve that request and choose the optimal among them. MCI inherently captures information about curtailment, and we do not need to take that into consideration explicitly. Lindberg et al. [11] shift data center loads geographically based on MCI. However, unlike our work, they do not account for the load-shifting distance, which may incur a high round-trip time latency if the workloads are shifted to a distant data center.

Google has also recently started shifting their media processing workloads to greener regions [8]. However, very little knowledge about their design is available publicly. Furthermore, media processing workloads are delay-tolerant. Thus, to the best of our knowledge, our work is the first industrial-grade system that can shift latency-sensitive stateless applications over geographically distributed hybrid clouds in a carbon-aware fashion.

Temporal load balancing. Delay-tolerant jobs can also be load-balanced by running them at a time when carbon intensity is low. Google now proactively shape the loads within their data centers and schedule the execution of delay-tolerant workloads to greener hours [18]. Weisner et al. [23] measure the trade-offs between carbon reduction and the time taken to complete a job by delaying jobs to different times in the future. These approaches reduce carbon emissions by shifting delay-tolerant loads to greener times. While temporal load balancing yields carbon savings, our work considers spatial load balancing and is complementary to such approaches.

4 SYSTEM DESIGN

Our current GSLB considers latency as the most important single metric for application performance. Different applications can tolerate different amounts of latency, ranging from 10–20 ms in some production workloads to sub-seconds in others. The GSLB uses an estimated geographical distance d_i between the client and data center i, in miles, as a proxy for latency and obtains it using the MaxMind geolocation database [14] to translate the IP address of the client to a geographical location and calculating the distance to each data center. Then, it selects the geographically closest data center to the client. Using the precise location of a client instead of a broader geographical location to estimate the client to data center distance would not necessarily be more effective since DNS requests can sometimes be cached at a higher level than an individual client.

Our modified GSLB considers the MCI of the region in which the data center resides, in addition to the client's estimated distance from each data center. The GSLB handles thousands of requests per second, and any carbon-optimization algorithm implemented on it must be lightweight to work on a real system. Hence, we choose a simple linear scoring function to select the optimal data center in terms of MCI and the distance between the client and the data center. Our load balancer uses the following scoring function to choose a data center:

For each data center i in $1 \ldots n$, it computes its score S_i as:

$$S_i = \lambda \, \mathrm{MCI}_i + (1 - \lambda) d_i, \tag{1}$$

with the condition

$$d_i \leq d_{\max} \tag{2}$$

where MCI_i is the current marginal carbon intensity (MCI) at data center i, in gCO_2eq/kWh, d_i is the estimated geographical distance

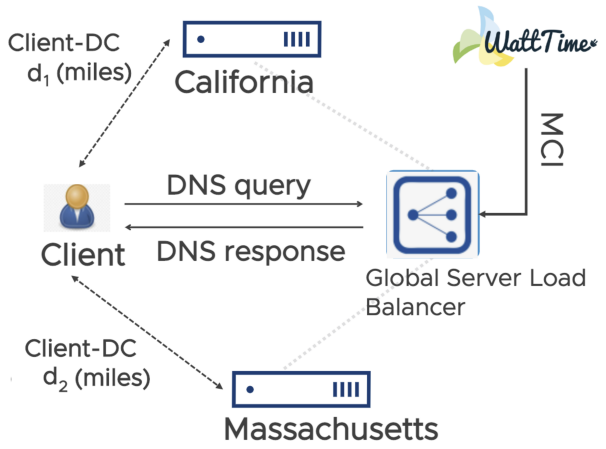

Figure 3: Carbon-aware load balancing architecture.

(in miles) between the client and the data center, $d_{\max}$ is the maximum allowed client-to-data center distance, and $0 \leq \lambda \leq 1$ is a weighting factor. Both MCI_i and d_i are scaled down to have values between 0 and 1 to counter their difference in magnitude. The algorithm chooses the data center satisfying the condition with the lowest score.

Figure 3 shows the design of our carbon-aware load-balancing implementation. In the figure, a client sends a DNS query for a service load balanced by the GSLB (most services will have more than the two choices of data centers shown in the figure). The GSLB computes a score per data center using Equation 1 per DNS request and chooses the data center with the lowest score. Then, it sends a DNS response with its choice to the client, thus directing application traffic to that data center for service until the DNS TTL expires.

The λ variable weights the importance of carbon intensity in the choice of data center. With $\lambda = 0$, only distance is considered; with $\lambda = 1$, only carbon intensity; and intermediate values produce intermediate weighting. Applications that can tolerate increased latency can use a higher λ. We used $\lambda = 0.67$ in our simulation and prototype, weighing carbon intensity twice as important as distance.

The GSLB uses a minimal time-to-live (TTL) in the responses it sends to clients. MCI can change quickly; so this helps to ensure that clients do not keep using one particular data center when a different one becomes preferable.

4.1 MCI as a design choice

The WattTime [22] service we used for this study provides both average carbon intensity (ACI) and marginal carbon intensity (MCI) data. There is currently no broad consensus on which is more appropriate for load-balancing. We chose MCI, as shown in Equation 1, for the reasons described below.

First, several regions generate surplus electricity from renewable sources at various times of day and year, which are usually curtailed to maintain the supply and demand balance. For example, curtailed energy from solar and wind was over 600 GWh in California in March 2023 [2]. This curtailed energy is essentially a clean source of electricity that is wasted. Thus, a carbon-aware load balancer operating on every request would inherently benefit by redirecting the majority of requests to a region with ongoing curtailment. Since MCI is based on the sources on the margin, MCI can capture curtailment situations, as renewable sources will be on the margin during those times. A zero MCI indicates energy curtailment and excess renewable energy in that region. Using MCI would thus take advantage of green energy which otherwise would have gone unused. Using ACI, which considers all energy generation rather than just marginal generation, would not take advantage of these opportunities to nearly the same extent.

Second, the enterprise workloads we consider in this paper are short-lived and also incur small increase in the data center load. We assume that incremental changes in energy load due to the load balancer are insignificant compared to the overall amount of energy demand in the grid. We believe this assumption is valid for initial adoption. This provides another motivation to choose MCI as it can correctly capture the carbon emissions due to this small incremental demand in a region.

Having said that, if our load-balancing strategy is widely adopted and the increase in demand is no longer insignificant compared to the grid demand increase, further study would be warranted. Also, if we want to spatially redirect jobs that run for more extended time periods so that sources on the margin may change while the job is executing, an initial decision based only on MCI may not be the best metric and other carbon metrics may be needed. Such design choices are beyond the scope of this paper and is left as future work. However, our approach will work with any such carbon metric. The carbon intensity signal is an input to our system, and our approach is not dependent on a specific ground truth source. Based on factors like the type of workloads, customer needs etc., we can seamlessly integrate our system with any available open-source [12, 13, 24], or commercial carbon-intensity providing services [6, 17, 22].

5 EVALUATION

Section 2 motivated our work through a simple simulation. Here, we expand on this experiment using a prototype based on real traffic. Our prototype deployed our system in Azure Cloud on servers in Azure data centers in US-West (Washington), US-East (Virginia) and UK-South. The availability of carbon intensity data limited our choice of data center locations for the prototype. On the other hand, clients do not have the same constraint, so we spread them more widely across the US and Europe. We limited our deployments only to the Azure cloud in this paper. However, our GSLB can support any public, private, or hybrid clouds.

The prototype ran for 2 hours at 12× real-time; so that the runtime represents 24 hours. During that time, it generated traffic from each client according to the traffic profile for our customers such as Adobe and Paypal, which consists of 70% GET requests of 1 kB size, 20% of GET/POST requests of 5 to 32 kB, and 10% large GETs of size 1 MB to 100 MB. We limited concurrent connections to 1,000 per second and 4 requests per connection.

We deployed our GSLB DNS server in the Azure US-West region. During the run, every hour, the GSLB adopted an update to marginal carbon intensity information from WattTime historical data for one

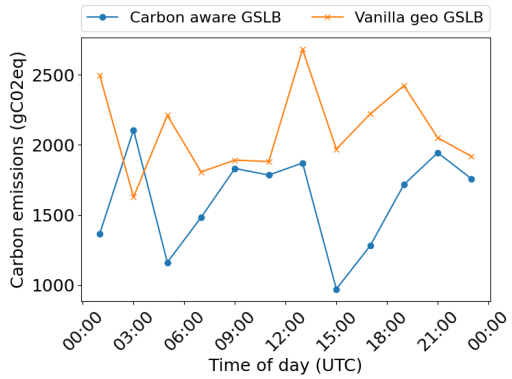

Figure 4: Carbon emissions for each load-balancing algorithm.

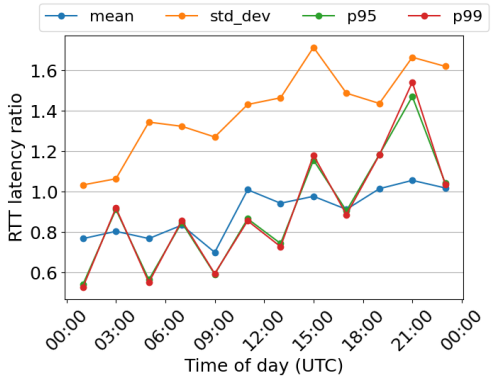

Figure 5: Ratio of round-trip latency for carbon-aware algorithm versus using geographical distance only.

particular day in January 2022 that we selected. We used a DNS TTL of 30 s to encourage clients to frequently re-check the choice of server.

We set $d_{\max} = 2000$ km to ensure that clients contact servers within a reasonable latency bound when differences in carbon intensity are extreme. This ensures that, for example, a client in Germany will not use a server in the United States, even if that would yield lower carbon intensity.

To evaluate the carbon savings possible with our carbon-aware algorithm, we compare it against the regular algorithm that considers only geographical distance. Figure 4 shows the carbon emissions for each algorithm, calculated using the carbon intensity data and carbon emission factors reported by WattTime. The space between the two plotted lines on the graph represents the carbon emission savings of the carbon-aware algorithm, which is the benefit of our work. Our results show an average carbon emissions savings of 21% and peaking at 51% savings.

Carbon awareness will only be adopted if its performance cost is not too high. To test this, Figure 5 compares latency for the two algorithms across the 24-hour period. For most of the day, the mean latency of the greener algorithm is between about 80% and 110% of the algorithm that only considers distance. The slightly reduced latency for much of the day surprised us. Perhaps it can be explained by the difference between network topology and geographical distance. The 95th and 99th percentiles show both lower and higher values and a spike late in the day. The standard deviation plot shows that the carbon-aware algorithm has considerably more latency variability than the standard algorithm. However, most importantly, the mean latency of our carbon-aware algorithm is comparable to our baseline algorithm, which means that our solution can easily be used in a real system.

6 LIMITATIONS OF OUR PROTOTYPE

In this paper, we describe a working prototype of the green load balancing extension to VMware's NSX Avi Global Server Load Balancer (GSLB). While our prototype shows significant potential for reducing carbon emissions via spatial load balancing, more needs to be done to build and deploy a carbon-aware GSLB that can work in a production setting for multiple type of workloads at scale. We discuss limitations of our work and future challenges below:

First, our GSLB assumes that servers can be scaled automatically and instantaneously in response to variations in demand. For instance, we assume that servers can be brought into service quickly to absorb traffic spikes. However, in reality, there is a time overhead for ramping up or down the servers. Carbon-aware GSLBs in production need to consider auto-scaling protocols and their associated overheads. Second, in case a green data center is saturated, the GSLB needs to redirect the requests to another data center while still maintaining other constraints. Thus, a carbon-aware GSLB should incorporate accurate real-time load information from all data centers into its decision making. Third, we only considered stateless workloads in this work. As MCI changes with time, consecutive requests from a given client could be redirected to different data centers by our carbon-aware GSLB. While this does not raise issues for a stateless service, to host stateful services, mechanisms to migrate state information across data centers is required, adding extra overhead to our carbon-aware GSLB. Finally, unavailability of carbon intensity data in some developing countries may make it infeasible to implement a carbon-aware GSLB in those regions.

7 FUTURE WORK

In this prototype, we limited our evaluation to 24 hours. However, we acknowledge that the carbon footprint of power can vary widely depending on seasonality. In the future, we plan to evaluate our algorithm's performance more robustly over longer time periods and different seasons. We aim to test our prototype with real customer workloads in production environments and, ultimately, to roll it out to customers who could use it to achieve part of their carbon reduction goals.

Additionally, we plan to extend our research in other directions. As mentioned in Section 6, our current GSLB prototype has limitations that need to be addressed. We plan to focus on adding load and price information in our optimization as a next step. One way to add load awareness would be introducing a load threshold and not redirecting clients to data centers with load above that threshold. This would prevent a green data center from becoming

overloaded. Additionally, considering the energy cost could be especially valuable for high-cost regions, such as areas in Europe with power shortages. If additional factors were to be considered, then the simple approach of having the administrator specify a single λ as a configuration weight may no longer work; so either the administrator would need to specify multiple weights or the load balancer could intelligently generate preconfigured profiles. Even with the current simple setup, a friendlier way to configure would be to specify an SLA for latency, with the load balancer choosing the greenest data center that meets the SLA, with a feedback loop to mitigate SLA violation. Also, our load balancer currently uses the geographical distance from clients to servers to approximate latency between them. In the future, measuring latency directly could yield better results.

8 CONCLUSION

In this paper, we have shown how VMware's Avi GSLB can play a significant role in reducing the carbon footprint of applications considerably by steering traffic to greener data centers. We built a prototype that showed an average of 21% and a maximum 51% carbon emissions saving with spatial load balancing. There is ample scope for further innovations and improvements in this area. Enterprises struggle to showcase carbon savings, and such advancements provide absolute metrics for organizations to quantify savings and garner more carbon credits by reducing their carbon emissions. We hope our approach paves the way for further enterprise-level carbon optimization solutions.

ACKNOWLEDGMENTS

The work was supported in part by NSF grant 2105494 and a grant from VMware.

REFERENCES

[1] AKCP Datacenter Usage. 2022. The Real Amount of Energy A Data Center Uses. (2022). https://www.akcp.com/blog/the-real-amount-of-energy-a-data-center-use/

[2] California ISO. 2023. Managing oversupply. http://www.caiso.com/informed/Pages/ManagingOversupply.aspx.

[3] David Mytton. 2022. How much energy do data centers use? (2022). https://davidmytton.blog/how-much-energy-do-data-centers-use/

[4] Department of Business, Energy and Industrial Strategy. 2021. Greenhouse gas reporting: conversion factors 2021. Retrieved October 4, 2022 from https://www.gov.uk/government/publications/greenhouse-gas-reporting-conversion-factors-2021

[5] Joseph Doyle, Robert Shorten, and Donal O'Mahony. 2013. Stratus: Load balancing the cloud for carbon emissions control. *IEEE Transactions on Cloud Computing* 1, 1 (2013), 1–1.

[6] Electricity Maps. 2022. Electricity Maps: Forecasted carbon intensity. https://static.electricitymaps.com/api/docs/index.html#forecasted-carbon-intensity.

[7] Peter Xiang Gao, Andrew R Curtis, Bernard Wong, and Srinivasan Keshav. 2012. It's not easy being green. *ACM SIGCOMM Computer Communication Review* 42, 4 (2012), 211–222.

[8] Google. 2021. We now do more computing where there's cleaner energy. https://blog.google/outreach-initiatives/sustainability/carbon-aware-computing-location/.

[9] Google. 2022. A policy roadmap for 24/7 carbon-free energy. https://cloud.google.com/blog/topics/sustainability/a-policy-roadmap-for-achieving-247-carbon-free-energy.

[10] IEA. 2023. Data Centres and Data Transmission Networks. https://www.iea.org/reports/data-centres-and-data-transmission-networks.

[11] Julia Lindberg, Bernard C Lesieutre, and Line A Roald. 2022. Using Geographic Load Shifting to Reduce Carbon Emissions. *arXiv preprint arXiv:2203.00826* (2022).

[12] Diptyaroop Maji, Prashant Shenoy, and Ramesh K Sitaraman. 2022. CarbonCast: multi-day forecasting of grid carbon intensity. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 198–207.

[13] Diptyaroop Maji, Ramesh K Sitaraman, and Prashant Shenoy. 2022. DACF: day-ahead carbon intensity forecasting of power grids using machine learning. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*. 188–192.

[14] MaxMind 2022. MaxMind GeoIP Databases & Services: Industry Leading IP Intelligence. https://www.maxmind.com/en/geoip2-services-and-databases.

[15] Microsoft. 2020. Microsoft will be carbon negative by 2030. https://blogs.microsoft.com/blog/2020/01/16/microsoft-will-be-carbon-negative-by-2030/.

[16] R. K. Pachauri and L. A. Meyer. 2014. Synthesis Report: Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Retrieved October 4, 2022 from https://archive.ipcc.ch/pdf/assessment-report/ar5/wg3/ipcc_wg3_ar5_annex-iii.pdf#page=7

[17] Zhelong Pan. 2022. Cielo: Carbon Intensity Service. https://confluence.eng.vmware.com/display/OCTO/Cielo+-+Carbon+Intensity+Service.

[18] Ana Radovanovic, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, et al. 2021. Carbon-aware computing for datacenters. *arXiv preprint arXiv:2106.11750* (2021).

[19] Reuters. 2019. Amazon vows to be carbon neutral by 2040, buying 100,000 electric vans. https://www.reuters.com/article/us-amazon-environment/amazon-vows-to-be-carbon-neutral-by-2040-buying-100000-electric-vans-idUSKBN1W41ZV.

[20] VMware. 2022. NSX Advanced Load Balancer. https://www.vmware.com/products/nsx-advanced-load-balancer.html.

[21] VMware. 2023. Journey to Net Zero. https://www.vmware.com/company/net-zero.html.

[22] Watttime. 2022. Watttime: The Power to Choose Clean Energy. https://www.watttime.org/.

[23] Philipp Wiesner, Ilja Behnke, Dominik Scheinert, Kordian Gontarska, and Lauritz Thamsen. 2021. Let's Wait Awhile: How Temporal Workload Shifting Can Reduce Carbon Emissions in the Cloud. In *Proceedings of the 22nd International Middleware Conference* (Québec city, Canada) *(Middleware '21)*. Association for Computing Machinery, New York, NY, USA, 260–272. https://doi.org/10.1145/3464298.3493399

[24] Xiaoyang Zhang and Dan Wang. 2023. A GNN-based Day Ahead Carbon Intensity Forecasting Model for Cross-Border Power Grids. In *Proceedings of the Fourteenth ACM International Conference on Future Energy Systems*. 361–373.

[25] Jiajia Zheng, Andrew A Chien, and Sangwon Suh. 2020. Mitigating curtailment and carbon emissions through load migration between data centers. *Joule* 4, 10 (2020), 2208–2222.

[26] Zhi Zhou, Fangming Liu, Yong Xu, Ruolan Zou, Hong Xu, John CS Lui, and Hai Jin. 2013. Carbon-aware load balancing for geo-distributed cloud services. In *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE, 232–241.