**Sciences du Nord Nature Science and Technology**
**Vol. 1, No. 02, July 2024: pp. 86-99**
**E-ISSN: 3046-6504**
**P-ISSN: 3047-4922**
**DOI: 10.71238/snnst.v2i02.182**

# Carbon-Aware Workload Shaping for Event-Driven Microservices Across Multi-Region Clouds

**Srinivas Nune [1], Karthik Juluri [2]**
[1] Independent Researcher and Srinivas.nune9792@gmail.com
[2] Independent Researcher and securityoperations40@gmail.com
Coresponding Author: Srinivas.nune9792@gmail.com

**ARTICLE HISTORY**

**Received July, 2024** **Revised July, 2024** **Accepted July, 2024**

**ABSTRACT**

Cloud infrastructure that power event-driven microservices is increasingly deployed across geographic regions and uses electricity on a scale that is a significant source of greenhouse gas emissions in the world. This paper integrates the results of 20 peer-reviewed and industry sources on carbon-aware computing, sustainable data-center design, and event-driven microservice architecture to create a unified framework for shaping workloads in a carbon-aware fashion for multi-region clouds. The synthesis brings together recent work on carbon-aware scheduling and scaling systems with temporal workload-shifting and existing microservice communication patterns to shed light on how the event-driven pipelines can be reorganized in terms of time, space, and elasticity while maintaining the latency service-level agreements. The quantitative results presented are based on the reported carbon reduction from elastic carbon scaling up to fifty-one per cent, carbon reduction from latency-aware geographic provisioning up to seventy per cent and carbon savings from temporal shifting up to more than twenty per cent during weekends in several national grids. The paper also places these gains in perspective against the overall information and communication technology (ICT) emissions curve, which saw global data-center electricity consumption grow to around two hundred five terawatt-hours in 2018, or nearly one per cent of the world's total electricity consumption – despite a five-hundred-fifty-percent increase in computing workloads. A suggested reference architecture combines carbon-intensity forecasting, event-broker-level routing and elastic autoscaling to form a closed loop that works for Kafka-based and serverless event-driven applications.

***Keywords:*** *Autoscaling; Carbon Intensity; Carbon-Aware Computing; Data-Center Energy; Event-Driven Architecture; Geographic Load Balancing; Green Software; Microservices; Multi-Region Cloud; Sustainable Computing; Temporal Shifting; Workload Shaping*

## INTRODUCTION

Cloud computing infrastructure is critical to global communication, and electricity consumption in networks, data centers, and connected devices is increasingly contributing to anthropogenic carbon emissions. Despite a 550 percent increase in computational workload and a tenfold increase in Internet Protocol traffic largely driven by virtualization and improved processor efficiency global data center electricity consumption is estimated to have reached approximately 205 terawatt-hours in 2018, or about 1 percent of global electricity consumption. The contribution of information and communication technology to global electricity demand is projected to rise from about one percent of global electricity consumption in 2007 to more than fifty-one percent by 2030. Furthermore, this sector’s relative contribution to global electricity emissions is projected to increase from about one percent in 2007.

In this context, the development of scalable, loosely coupled distributed applications is a typical practice to adopt an event-driven microservice architecture. Message brokers, event streams, and asynchronous communication are often used to create independently deployable services that respond to events, rather than synchronous calls, and decompose monolithic applications. Latency reduction and fault tolerance are the main drivers of multi-region deployment of such systems, and this is also an opportunity: The majority of event-driven workloads are also likely to be latency tolerant, elastic, and generally uncoupled.

This paper builds on the evidence presented in the literature of carbon-aware datacenter design, cloud workload scheduling, and event-driven microservice architecture, and forms a holistic understanding of how workload-shaping techniques can be applied specifically to multi-region, event-driven systems, and the quantitative emissions-reduction capabilities, trade-offs, and architectural considerations involved. The remainder of this article is organized as follows: the Literature Review synthesizes prior work on the carbon footprint of digital infrastructure and on event-driven microservice design; the Methods section describes the review and synthesis procedure; Results and Discussion presents the four carbon-aware workload-shaping techniques, the proposed reference architecture, and the associated trade-offs; and the Conclusion summarizes the findings and directions for future work.

## LITERATURE REVIEW

### *The Carbon Footprint of Cloud and Communication Infrastructure*

Digital infrastructure's environmental impact has been measured from multiple complementary perspectives. Global data-center energy analysis using a bottom-up approach showed that while workloads are growing rapidly, energy consumption of data centers was relatively flat from 2010 to 2018 due to a fourfold drop in electricity use per computation on typical volume servers and a twenty-fivefold increase in energy use for storage capacity with comparatively small incremental power consumption. This efficiency-focused decoupling is in contrast to previous projections on communication technology, which had modeled three scenarios – best, expected, worst – of consumer devices, communication networks and data centers until 2030, establishing a scenario where communication technology could account account for more than half of global electricity consumption by 2030 in the worst-case scenario. If left unaddressed, by 2040 the contribution of computing devices, data centers, and communication networks to global carbon dioxide emissions could reach more than 14 percentor more than half of the global transportation sector’s current contribution. Table 1 consolidates all these projections and helps you understand the breadth of market opportunities [1].

**Table 1: Global ICT and Data-Center Energy and Carbon Footprint Projections Synthesized From References**

| Source | Metric | Reported Value | Basis / Horizon |
| --- | --- | --- | --- |
| Masanet et al. | Global data-center electricity use | ~205 TWh (~1% of global electricity) | 2018 |

| | | | |
|---|---|---|---|
| Masanet et al. | Growth in DC electricity use despite +550% compute-instance growth | +6% | 2010–2018 |
| Andrae & Edler | Communication-technology electricity use, worst-case scenario | up to 51% of global electricity | 2030 (projected) |
| Andrae & Edler | Communication-technology share of global GHG emissions, worst case | up to 23% | 2030 (projected) |
| Belkhir & Elmeligi | ICT relative contribution to global GHG emissions | grows from 1.0–1.6% (2007) to >14% of 2016-level emissions | 2040 (projected) |
| Radovanović et al. | Global data-center industry share of electricity demand | 3%–13% (projected range) | 2030 |

*Note. Values are drawn directly from the cited studies' reported findings; no figures have been extrapolated beyond what each source states.*

Table 1 reveals that the divergence of historical efficiency-driven stabilization and projected worst-case growth trajectories make workload-level carbon awareness, instead of relying on hardware efficiency only, a recent interest of system research. Since all the scheduling decisions are made in software, carbon-aware workload shaping is a complementary control knob to efficiency in physical infrastructure, and can be used without changing the underlying infrastructure [2].

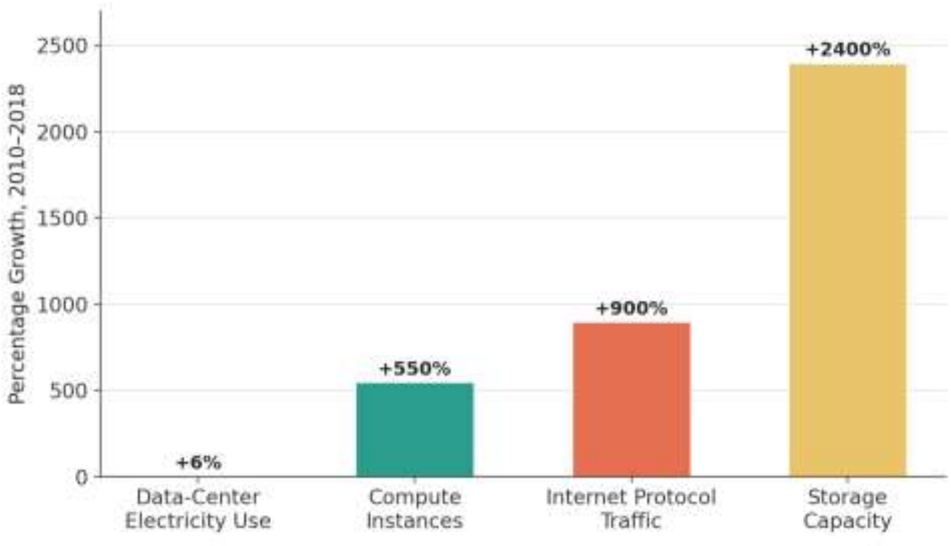

**Figure 1: Global Data-Center Electricity Use Compared with Underlying Workload Growth Drivers, 2010–2018**

*Note. Percentage growth figures are derived from reported values in Masanet et al. (electricity use, compute instances, internet-protocol traffic, storage capacity).*

### *Event-Driven Microservice Architectures in Multi-Region Clouds*

Event-driven microservice design is an application architecture in which the logic of the application is divided into services that are only concerned with the production, detection, and consumption of discrete events, but not direct service-to-service requests. In the literature two main integration patterns have been identified: choreography, where the services publish and subscribe on events without being centrally coordinated; and orchestration, where a central controller coordinates the sequence of service invocations. While choreographed systems allow for high decoupling and independent scalability and are typically atop distributed log platforms, they are more complex to implement for end-to-end visibility and data consistency; orchestrated systems provide clearer control flow and ease debugging, but have a coordination bottleneck and are more tightly coupled between orchestrator and downstream services. A systematic review of deployment and communication patterns in microservice architectures identified significant variation across organizations in the relative use of these trade-offs, with hybrid patterns of using choreography for event-driven flows of high throughput and orchestration for business-critical transactional flows gaining in popularity [3], [4].

This is complemented by research that has documented the widespread use of per-service data ownership and eventual consistency in data-management practices for microservices, which serve to decrease the need for cross-region data synchronization and therefore make it easier to move individual instances of a service from one region to another, while keeping the consistency guarantees. The deployment of microservices across multiple regions has been shown to enhance performance and scalability, especially by choosing replicas across each region to optimize network latency to the end user [5]. When deployed in multiple regions, however, the additional regions come with additional energy mixes and additional region-specific carbon intensities, which adds additional degrees of freedom for carbon-aware placement decisions, a property that architectures that are optimized for latency and availability do not heavily leverage.

**Table 2: Comparative Communication Patterns in Event-Driven Microservice Architectures**

| Dimension | Choreography | Orchestration |
|---|---|---|
| Coupling | Low; services react independently to events | Higher; services depend on central controller |
| Visibility / Debugging | More difficult end-to-end tracing | Clearer control flow and audit trail |
| Scalability | High; independent horizontal scaling per service | Bounded by orchestrator throughput |
| Failure Handling | Localized; failures isolated to consumers | Centralized; orchestrator is a single point of coordination |
| Carbon-Shaping Suitability | Favorable for temporal and elastic shaping of batch consumers | Favorable for latency-constrained spatial routing of transactional flows |

*Note. Comparison synthesized from documented microservice deployment and communication pattern research.*

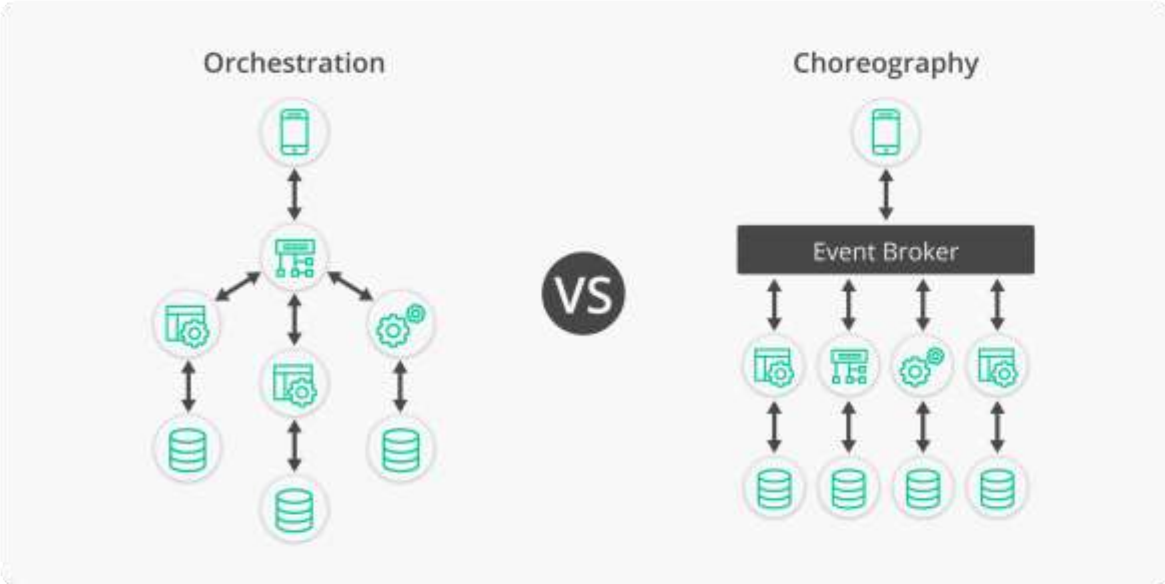


**Figure 2: Choreography and Orchestration Integration Patterns in Event-Driven Microservices (Solace)**

## METHODS

This article follows a narrative, integrative literature-synthesis design rather than an original empirical experiment. The unit of analysis is the peer-reviewed and industry-reported evidence on carbon-aware computing, sustainable data-center design, and event-driven microservice architecture, and the objective is to synthesize this evidence into a single, internally consistent framework for multi-region, event-driven workload shaping.

Sources were identified through a targeted search of major computing and sustainability venues (including ACM, IEEE, Springer, and Science/Sustainability-indexed outlets) together with publicly documented industry systems, using search terms such as “carbon-aware scheduling,” “carbon-aware datacenter,” “geographical load balancing,” “temporal workload shifting,” and “event-driven microservices.” Twenty sources published between 2015 and 2023 were retained after screening for direct relevance to at least one of three themes: (i) the measured or projected carbon and energy footprint of digital infrastructure, (ii) architectural and communication patterns in event-driven microservices, and (iii) quantitative carbon-aware scheduling, scaling, or routing techniques.

For each retained source, the reported quantitative findings (percentage carbon or energy savings, latency impact, and cost overhead) were extracted and tabulated, then grouped into four recurring technique families: temporal workload shifting, spatial (geographic) load balancing, elastic carbon scaling, and latency-constrained carbon-aware scheduling. These groupings were then used to derive a comparative matrix (Table 7) and to compose the reference architecture presented in the Results and Discussion section, following the analytic-synthesis approach commonly used in integrative technology reviews.

## RESULTS AND DISCUSSION

### *Carbon-Aware Techniques for Workload Shaping*

#### *Temporal Workload Shifting*

Carbon intensity analysis across the regions in Germany, Great Britain, France and California for an entire year showed that most regions could still achieve savings of over twenty per cent if they moved delay-tolerant work to the weekend, with a slight delay (one working day) to the same effect yielding savings of over five per cent in all four regions studied and a carbon-aware approach to scheduling even within a smaller window still resulting in savings over ten per cent in several regions. These conclusions were based on the fact that the short-term shifting potential can frequently be greatest before the sunrise on a solar intensive grid or during the evening hours of the day when fossil generation is reduced to a minimum overnight. Google's production-scale Carbon-Intelligent Compute System takes a similar approach—it operates at the level of the data center cluster, using carbon-intensity forecasts from the day ahead, and demand-prediction models to create virtual capacity curves, which delay workloads that are carbon flexible until times of the day when the grid is less carbon-intensive—at the time of reporting, the system is shaping load in over twenty data centers on four continents, impacting electricity use of up to fifteen-point-five terawatt-hours [6], [7]. Temporal shifting is most relevant for event-driven microservice pipelines, where event consumers are asynchronous, batch, and have a tolerance for queued backlogs without impacting their consumer service-level objectives, such as aggregating analytics data, digesting notifications, or generating reports at a later time.

**Table 3: Reported Regional Savings from Temporal Workload Shifting**

| Shifting Strategy | Reported Savings | Regions Studied |
|---|---|---|
| Weekend-oriented shifting of delay-tolerant workloads | > 20% in most regions | Germany, Great Britain, France, California |
| Next-working-day deferral | > 5% across all regions | Germany, Great Britain, France, California |
| Carbon-aware scheduling within shorter windows | > 10% in several regions | Germany, Great Britain, France, California |
| Per-capita annual emissions (reference context) | 8.4 t $CO_2e$ (2019) | Germany and Great Britain |

*Note. Figures reported by Wiesner et al. based on year-2020 regional grid carbon-intensity analysis.*

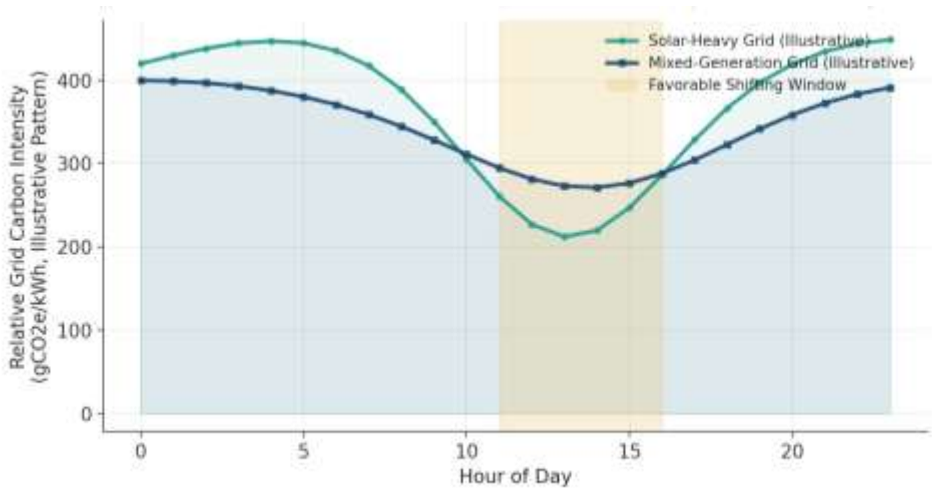

**Figure 3: Conceptual Diurnal Carbon-Intensity Pattern Motivating Temporal Shifting**

*Note. This figure is an illustrative schematic of the diurnal shape described in the temporal-shifting literature; it does not represent measured values from a specific grid.*

***Spatial Load Balancing Across Regions***

The carbon intensity of an electric grid can vary significantly from one location to another, reflecting differences in the composition of power generation facilities, which are utilized by geographic load balancing. Early versions of geographic load balancing used distribution algorithms to direct demand to regions with faster-responding renewable energy sources. This demonstrates that the proportion of renewable energy consumed by a geographic region can be significantly increased through dynamic pricing signals proportional to carbon emissions. Further extending this idea, datacenter-level renewable investment modelling showed that deployment of battery storage systems that cover 80% depth of discharge (DOF) reduced total carbon by about 5% on average, with a range of low-carbon efficiency from 3% to 9% across the regions – and a selected efficient carbon-responsive operational policy reduced total carbon by about 50% compared with some of the most common approaches to carbon scheduling so far established. That same modeling also strongly suggested that there was an increasing return to fewer and fewer hours of carbon-free power generation: The last five percent of coverage (from ninety-five to ninety-nine-point-nine percent of hours) required more than five times the investment in renewable generation required to achieve the first 95 percent [8], [9].

**Table 4: Renewable Energy Investment and Storage Effects on Operational Carbon**

| Intervention | Reported Effect |
|---|---|
| 80% depth-of-discharge battery storage | Lowers total carbon ~5% on average (range 3%–9% across data-center regions) |
| Renewable coverage 0% → 95% vs. 95% → 99.9% | Latter increment requires > 5x more renewable-generation investment |
| Carbon Responder (carbon-aware scheduling policy) | ~2x greater carbon reduction versus baseline adapted methods |

*Note. Figures reported by Acun et al. (Carbon Explorer framework).*

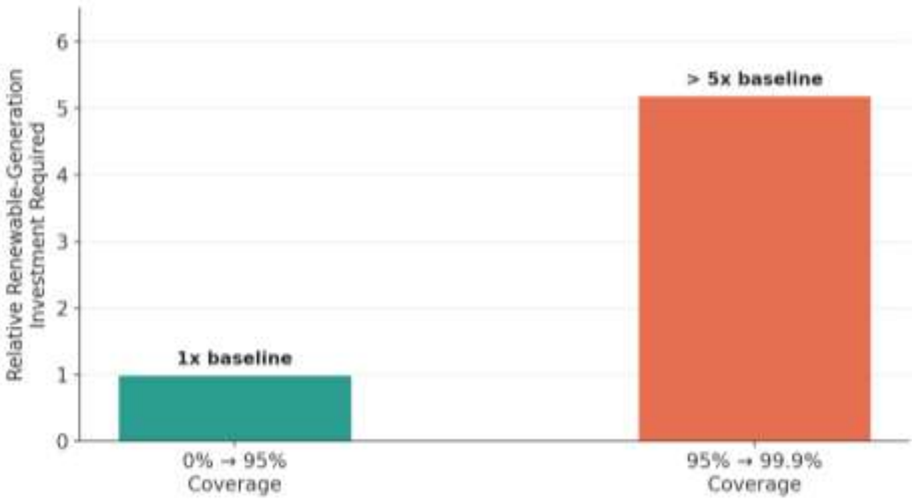

**Figure 4: Diminishing Returns of Renewable-Energy Investment Toward 24/7 Carbon-Free Coverage**

***Elastic Carbon Scaling***

Elastic carbon scaling, based on the same concept as automatic scaling in the cloud, dynamically allocates resources to running jobs according to relative changes in the network's carbon intensity. Using real-world workloads such as machine learning training and messaging interfaces, as well as commercial cloud platforms, they found that the additional cost never exceeded 18 percent under the most unfavorable conditions they tested [10]. Event-based consumer group scalability on platforms such as Kafka which allows for increasing or decreasing the number of consumers by changing the number of partitions allocated to each consumer group constitutes elastic carbon scalability. This is a characteristic of carbon scalability that can be naturally applied to event-based microservice deployments: the number of consumer group replicas can be increased when carbon intensity is low, then reduced or scaled down as carbon intensity rises.

***Latency-Constrained Carbon-Aware Scheduling for Distributed Web Services***

Many event-driven request-response paths are interactive, and the latency sensitive, so that cannot afford the multi-hour delay imposed by temporal shifting; they seek scheduling approaches that simultaneously minimize carbon intensity and network latency. A recently introduced distributed web services provisioner and scheduler for carbon-aware placements formulated the placement problem as a multi-objective optimization problem that takes into account the variable carbon intensity of different regions, re-provisioning capacity between regions while at the same time avoiding regions that do not meet round-trip latency requirements due to carbon-aware routing [11], [12]. When considering carbon awareness along with the impact on latency budgets, the approach achieved carbon-emission reductions of up to seventy percent compared with baseline provisioning across various commercial cloud regions, without measurable degradation to latency performance, suggesting that carbon awareness does not need to come with a cost to latency performance when both are considered together and both are modeled, with the approach applied across diverse cloud regions. This class of technique is the most directly translatable to event-driven microservices running synchronous request-reply interactions, while being layered on top of the otherwise asynchronous event backbones, like application programming interface.

**Table 5: Reported Carbon Savings from Elastic and Latency-Constrained Scheduling Systems**

| System | Comparator Baseline | Reported Carbon Savings | Latency / Cost Impact |
|---|---|---|---|
| CarbonScaler | Carbon-agnostic execution | up to 51% | cost overhead ≤ 18% |
| CarbonScaler | Suspend-resume policy | up to 37% | — |
| CarbonScaler | Best static scaling policy | up to 8% | — |
| CASPER | Baseline provisioning methods | up to 70% | no latency degradation |
| Carbon Explorer (Responder) | Baseline adapted approaches | ~2x reduction | — |

*Note. Figures reported by Hanafy et al. (CarbonScaler), Souza et al. (CASPER), and Acun et al. (Carbon Explorer).*

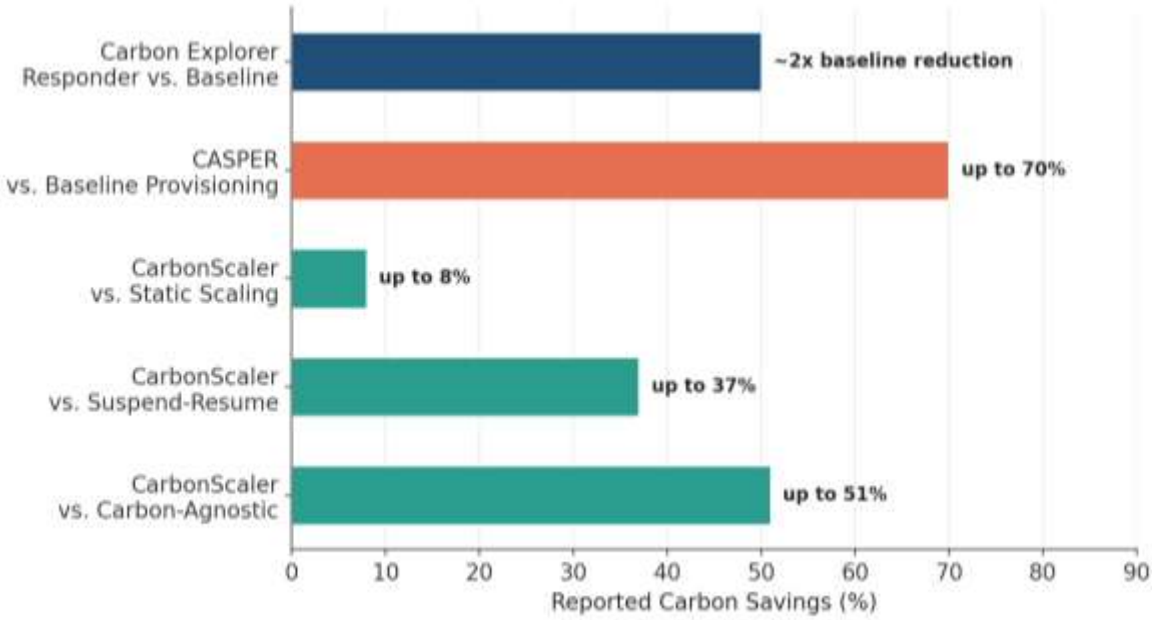


**Figure 5: Reported Carbon Savings of Carbon-Aware Scheduling and Scaling Systems**

*A Synthesized Framework for Carbon-Aware Workload Shaping in Event-Driven Microservices Architectural Components*

By combining the mechanisms described in the previous sections, we can envision a reference architecture for carbon-aware workload formation in multi-region event-driven microservices consisting of four interconnected components. Taking these mechanisms into account as a whole, the reference architecture for carbon-aware workload formation in multi-region event-driven microservices. The event-aware routing layer sits on top of or within the message routing layer and incorporates delay and latency sensitivity metadata into events. As a result, the system can distinguish between similar batch flows that are delay-tolerant and can wait for temporal delays and request-response-based event flows that require low-latency processing. The placement and scaling controller uses estimates and event metadata to make three decisions: whether to defer a set of events to a time window with lower carbon intensity within the cluster; which region to place event streams in based on latency and capacity; and how many consumer replicas of a partition set should be active at a given carbon intensity level. Lastly, a service-level-objective guard rail sets upper limits on latency and growth of the backlog, ensuring carbon-optimization decisions do not infringe on the responsiveness and latency guarantees of downstream applications, similar to the weighted trade-off parameter utilized in carbon-aware scheduling research to trade off carbon savings with responsiveness [13].

**Table 6: Architectural Components of the Synthesized Carbon-Aware Workload Shaping Framework**

| Component | Function | Analogous Production Mechanism |
|---|---|---|
| Carbon-Intensity Forecasting Module | Produces short-horizon regional carbon-intensity forecasts | Day-ahead pipelines in Google's Carbon-Intelligent Compute System |

| | | |
|---|---|---|
| Event-Broker-Aware Routing Layer | Tags events with deferability and latency metadata | Kafka-based event-streaming platforms |
| Placement and Scaling Controller | Decides deferral, regional routing, and replica count | CASPER multi-objective optimization; CarbonScaler marginal allocation |
| SLO Guard Rail | Enforces latency and backlog ceilings | Latency-weighting trade-off parameter used in CASPER |

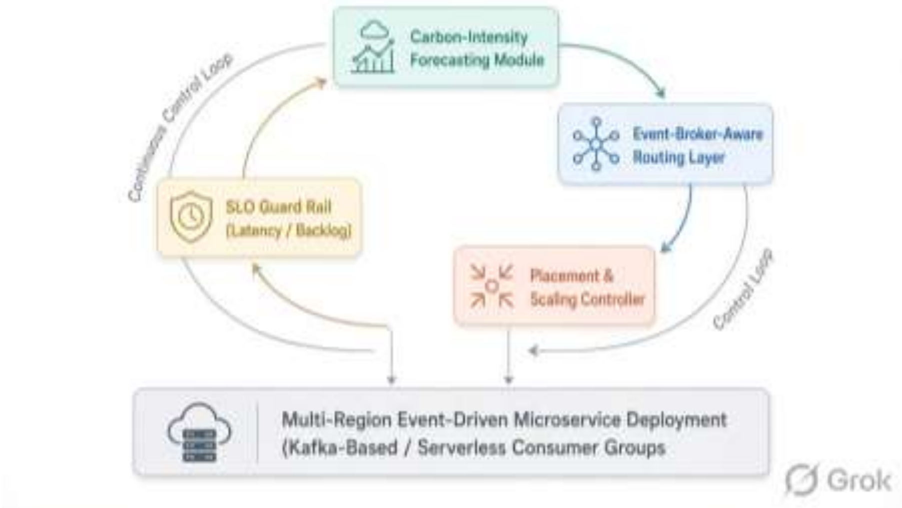


**Figure 6: Synthesized Carbon-Aware Workload Shaping Framework: Component Architecture (Grok)**

***Formal Problem Formulation***

The operational carbon emissions of a workload executing over a discretized time horizon can be expressed as follows.

$$C_{op} = \Sigma_{t=1}^{T} P(t) \cdot CI(t) \cdot \Delta t \quad (1)$$

Where P(t) denotes instantaneous power draw and CI(t) denotes the grid carbon intensity, typically expressed in grams of carbon-dioxide equivalent per kilowatt-hour, prevailing at time t. A corresponding carbon-efficiency metric, analogous to conventional energy efficiency, can be defined as follows.

$$CE = W / C\_op \quad (2)$$

Where W represents completed useful work. For multi-region placement, the routing and scaling controller can be formulated as a constrained optimization problem.

$$minimize \ \Sigma\_r \, \Sigma\_t \, x\_(r,t) \cdot P\_r(t) \cdot CI\_r(t) \quad (3)$$

$$subject\ to \ \ \Sigma\_r \, x\_(r,t) = 1 \ for\ all\ t$$

$$L\_r(t) \leq L\_max \ for\ all\ assigned\ (r,t)$$

$$\Sigma\_t \, x\_(r,t) \cdot P\_r(t) \leq Cap\_r$$

Where x_(r,t) denotes the fraction of a given event stream assigned to region r at time t, L_r(t) is the resulting end-to-end latency, L_max is the applicable service-level latency ceiling, and Cap_r is the provisioned capacity of region r. This formulation generalizes the latency-constrained objective used in carbon-aware web-service scheduling research to the finer-grained, per-event-stream context of microservice deployments [14].

***Comparative Analysis of Carbon-Aware Strategies***

Table 7 shows the four synthesized approaches juxtaposed by the numbers of reported carbon savings, latency effect, cost overhead, and type of workload in an event-driven style. While each of these techniques can provide meaningful and measurable savings, from mid-single-digit percentages on average for datacenter-level renewable & storage investment, up to seventy percent for spatial load balancing, only datacenter-level investment is a direct impact on operational carbon; and datacenter-level investment has a steeply declining marginal return as it nears full decarbonization, so it's not a panacea for workload-level shaping.

These four techniques are not in isolation, but additively: an event-driven pipeline may use latency-constrained spatial scheduling to route interactive request-response services, an elastic scaling of horizontally scalable stream processors, and a temporal shifting of batch analytics consumers can all increase the emissions reduction that can be achieved by any one of these techniques working alone. The comparative evidence also shows that the scaling cost overhangs are not too high even under poor conditions, meaning that the top driver is architectural readiness, which in this case is a set of deferability metadata, regional redundancy, and elastic consumer-group scaling—rather than cost [15].

**Table 7: Comparative Matrix of Carbon-Aware Workload Shaping Techniques for Event-Driven Microservices**

| Technique | Peak Reported Savings | Latency Sensitivity | Best-Fit Event Type | Primary Constraint |
|---|---|---|---|---|
| Temporal Shifting | > 20% (weekend) | Low tolerance for delay-sensitive flows | Batch / async consumers | Backlog growth limits |
| Elastic Carbon Scaling | up to 51% | Moderate; no completion-time extension | Horizontally scalable stream consumers | Cost overhead ≤ 18% |
| Spatial Load Balancing | up to 70% | Requires latency-budget modeling | Request-response services with regional redundancy | Cross-region data consistency |
| Renewable & Storage Investment | ~5% average (3–9% range) | None; infrastructure-level | All workload types | Diminishing marginal returns |

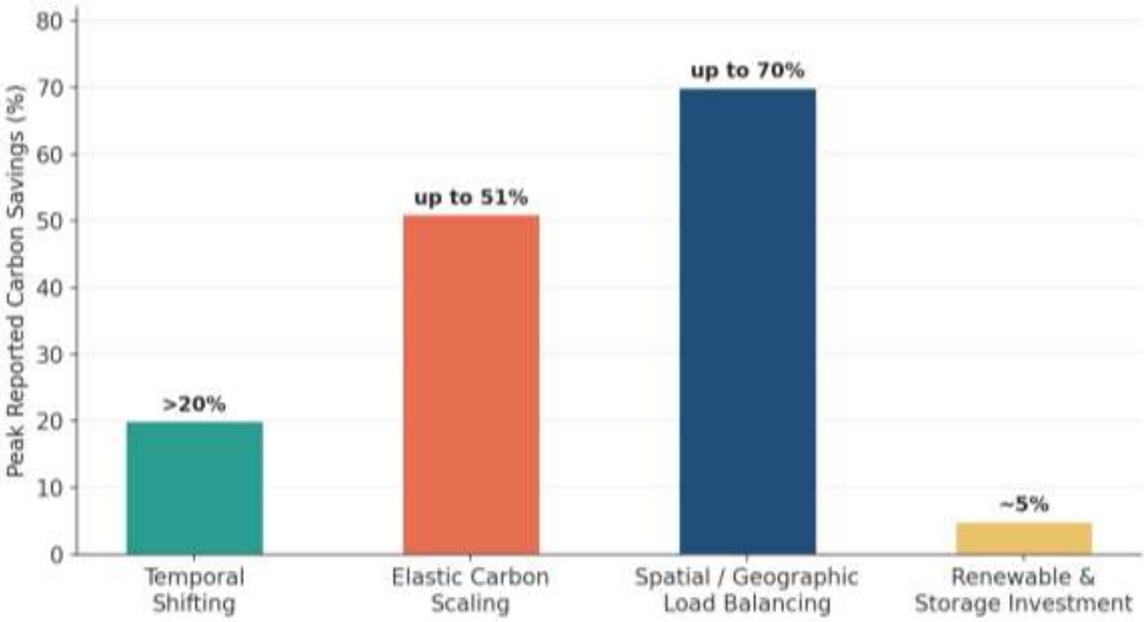


**Figure 7: Comparative Peak Carbon Savings Across Workload Shaping Techniques**

***Discussion: Trade-Offs, Barriers, and Regulatory Considerations***

There are several challenges to implementing carbon-aware workload shaping in production event-driven systems. The research involved with data-management microservices has established that per-service databases and eventual consistency make the safe pause and move of stateful event consumers more difficult because consumers with local materialized views can't always be paused or moved without the risk of staleness or duplicate processing. Sustainable serverless computing research has also found that abstracting service infrastructure placement from the developer can also hide the very signals (instance location, scaling behavior, etc.) that the carbon-aware controllers need to know about, indicating that more needs to be built into the platform layer than left to the individual service developers [16], [17].

From a regulatory and organizational perspective, the increasing disparity between historical emissions stabilization based on efficiency gains and worst-case scheduling and orchestration growth based on communications technology suggests a need to directly incorporate carbon awareness into scheduling and orchestration layers, instead of just relying on hardware refresh cycles or renewable-energy procurements. Geographically, the route towards regions with cleaner grids brings in issues of distribution, as the region with the highest carbon intensity variation may not be the region with the highest latency distance to end-users, potentially impacting the distribution of the carbon-aware benefits of routing and the remainder of the latency cost of the routing decision. These challenges can be tackled by making coordinated changes to event-broker design, service-level-objective specification, and cross-region data-consistency protocols, not by making any one particular change in any one layer of the stack [18].

## CONCLUSION

The synthesized evidence points to significant and largely additive carbon reduction potential for event-driven microservices deployed across multiple region cloud deployments, with individual microservices achieving carbon reductions from 5 percent when using carbon-aware workload-shaping techniques to over 70 percent, using latency-aware geographic provisioning with renewable and storage investments. There is a need for architectural readiness – deferability-aware event metadata, regionally redundant service deployment and elastic consumer-group scaling – in addition to availability of underlying algorithms, which this literature has synthesized here, and which are well developed. Empirical testing of combined temporal-spatial-elastic controllers on production Kafka-based or serverless event-driven pipelines would be useful to build on top of the systems synthesized in this paper, as would be a tighter coupling of carbon-intensity forecasting into event-broker/service-mesh infrastructure.

## REFERENCES


[1] B. Acun *et al.*, "Carbon Explorer: A Holistic Framework for Designing Carbon Aware Datacenters," in *ASPLOS 2023 - Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, Association for Computing Machinery, 2023, pp. 118–132. doi: 10.1145/3575693.3575754.

[2] A. S. G. Andrae and T. Edler, "On Global Electricity Usage of Communication Technology: Trends to 2030," *Challenges*, vol. 6, no. 1, pp. 117–157, 2015, doi: 10.3390/challe6010117.

[3] L. Belkhir and A. Elmeligi, "Assessing ICT global emissions footprint: Trends to 2040 & recommendations," *J. Clean. Prod.*, vol. 177, pp. 448–463, 2018, doi: 10.1016/j.jclepro.2017.12.239.

[4] A. Bellemare, *Building Event-Driven Microservices*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2020. [Online]. Available: https://www.oreilly.com/library/view/building-event-driven-microservices/9781492057888/

[5] M. Dayarathna, Y. Wen, and R. Fan, "Data Center Energy Consumption Modeling: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 732–794, 2016, doi: 10.1109/COMST.2015.2481183.

[6] W. A. Hanafy, Q. Liang, N. Bashir, D. Irwin, and P. Shenoy, "CarbonScaler: Leveraging Cloud Workload Elasticity for Optimizing Carbon-Efficiency," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 7, no. 3, pp. 1–28, 2023, doi: 10.1145/3626788.

[7] B. Stopford, *Designing Event-Driven Systems: Concepts and Patterns for Streaming Services with Apache Kafka*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2018. [Online]. Available: https://www.oreilly.com/library/view/designing-event-driven-systems/9781492038252/

[8] I. Karabey Aksakalli, T. Çelik, A. B. Can, and B. Tekinerdogan, "Deployment and communication patterns in microservice architectures: A systematic literature review," *J. Syst. Softw.*, vol. 180, 2021, doi: 10.1016/j.jss.2021.111014.

[9] R. N. Laigner, Y. Zhou, M. A. V. Salles, Y. Liu, and M. Kalinowski, "Data Management in Microservices: State of the Practice, Challenges, and Research Directions," *Proc. VLDB Endow.*, vol. 14, no. 13, pp. 3348–3361, 2021, doi: 10.14778/3484224.3484232.

[10] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew, "Greening Geographical Load Balancing," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 657–671, 2015, doi: 10.1109/TNET.2014.2308295.

[11] J. Malmodin and D. Lundén, "The Energy and Carbon Footprint of the Global ICT and E&M Sectors 2010–2015," *Sustainability*, vol. 10, no. 9, 2018, doi: 10.3390/su10093027.

[12] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science (80-. ).*, vol. 367, no. 6481, pp. 984–986, 2020, doi: 10.1126/science.aba3758.

[13] A. Rahmatulloh, F. Nugraha, R. Gunawan, and I. Darmawan, "Event-Driven Architecture to Improve Performance and Scalability in Microservices-Based Systems," in *2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, IEEE, 2022, pp. 1–6. doi: 10.1109/ICADEIS56544.2022.10037390.

[14] H. F. O. Rocha, *Practical Event-Driven Microservices Architecture: Building Sustainable and Highly Scalable Event-Driven Microservices*. Berkeley, CA: Apress, 2022. doi: 10.1007/978-1-4842-7468-2.

[15] A. Souza *et al.*, "CASPER: Carbon-Aware Scheduling and Provisioning for Distributed Web Services," in *Proceedings of the 14th International Green and Sustainable Computing Conference (IGSC 2023)*, Association for Computing Machinery, 2023, pp. 67–73. doi: 10.1145/3634769.3634812.

[16] V. Velepucha and P. Flores, "A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges," *IEEE Access*, vol. 11, pp. 88339–88358, 2023, doi: 10.1109/ACCESS.2023.3305687.

[17] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen, "Let's Wait Awhile: How Temporal Workload Shifting Can Reduce Carbon Emissions in the Cloud," in *Proceedings of the 22nd International Middleware Conference*, Association for Computing Machinery, 2021, pp. 260–272. doi: 10.1145/3464298.3493399.

[18] P. Wiesner, D. Scheinert, T. Wittkopp, L. Thamsen, and O. Kao, “Cucumber: Renewable-Aware Admission Control for Delay-Tolerant Cloud and Edge Workloads,” in *Euro-Par 2022: Parallel Processing*, vol. 13440, Springer International Publishing, 2022, pp. 218–232. doi: 10.1007/978-3-031-12597-3_14.