Magellan: A Carbon-Aware Task Scheduler for Geo-Distributed Systems

William Kaminski
Tufts University
Medford, Massachusetts, USA
wkamin01@tufts.edu

Hari Sundar (Advisor)
Tufts University
Medford, Massachusetts, USA
hari.sundar@tufts.edu

ABSTRACT

Magellan is a decentralized carbon-aware scheduler that live-migrates long-running, stateful HPC and ML workloads across geo-distributed sites to exploit variation in grid carbon intensity. Its decision model accounts for makespan, compute and transfer carbon, migration overhead, and monetary cost under an explicit budget cap. Across CloudLab and Google Cloud evaluations, Magellan reduces lifecycle carbon by up to **92.8% (13×)** relative to static execution, while reducing makespan by **11.6%** at approximately neutral monetary cost, all without requiring application source-code modifications.

CCS CONCEPTS

• **Computer systems organization** → **Distributed architectures**; *Cloud computing*.

KEYWORDS

carbon-aware scheduling, sustainable computing, process migration, checkpoint/restore, geo-distributed systems

1 PROBLEM AND MOTIVATION

Data-center electricity is a fast-growing contributor to global emissions [8]. The carbon intensity of the electricity that powers a job (gCO_2e/kWh) is not constant: it varies hour to hour as solar and wind generation ramp, and across regions where generation mixes are largely uncorrelated. In one 72-hour trace, carbon intensity in Boston, Spain, and South Australia diverges by factors of 4.4× and 12.7×. The same computation therefore emits considerably different amounts depending on *when* and *where* it runs.

Temporal schedulers only delay work until the local grid improves and are bounded by one region's carbon profile. When local conditions worsen, they must idle. Spatial shifting promises more headroom, but realizing it means relocating a job that is already running. The workloads that dominate cluster hours are the ones prior work excludes because moving them requires transporting live process state across a wide-area network. We ask whether stateful jobs can be migrated for carbon *profitably* once the subsequent overheads are accounted for.

2 BACKGROUND AND RELATED WORK

Temporal systems such as LinTS [13], GREEN [16], and GAIA [6] shift work in time. Spatial approaches remain largely simulated or stateless: CASPER [15] balances web requests, ECMR [11] simulates ML placement, and carbon-aware networking [10] routes traffic rather than execution state. DMTCP [2] and CRIU [1] provide checkpoint/restore, while decentralized schedulers CAGE [9], EdgeDecAp [14], and DARO [7] coordinate placement but not long-running stateful migration. To our knowledge, no prior system combines live stateful migration, priced migration overheads, and decentralized, capacity-aware arbitration on real infrastructure.

3 APPROACH AND UNIQUENESS

Magellan adds a *spatial* control knob to carbon-aware scheduling and treats migration as a first-class, priced operation rather than a free primitive.

A unified decision model. Each peer periodically re-scores every task it owns, choosing $x^* = \arg\min_x(\alpha\widetilde{T}_x + \beta\widetilde{C}_x + \gamma\widetilde{\$}_x)$ over $\{\text{CONTINUE}, \text{PAUSE}, \text{MIGRATE}\}$. Destination runtime is adjusted for heterogeneous execution rates as $T_c' = T_c(r_{\text{src}}/r_{\text{dst}})$, so a cleaner but slower node is penalized accordingly. Carbon is split into compute carbon and *transfer* carbon, with transfer energy modeled as $E_{\text{transfer}} = S(\epsilon_0 + \epsilon_d d/1000)$ for state volume S and great-circle distance d. Destination carbon is a window-averaged linear forecast, so the score reflects the grid a job *arrives* into. Cost includes destination compute price plus per-GB egress, and a hard budget cap $K_{\max}$ prunes MIGRATE from the action set once $K + \$_{\text{mig}} \geq K_{\max}$. Weights (α, β, γ) are calibrated offline by grid search over the simplex on a year of traces, then adapted online within ±25% as slack changes.

A break-even condition. We derive a break-even condition that compares the source-destination carbon gap against checkpoint, transfer, restore, and network-energy overheads. For our measured parameters, profitable inter-continental migrations, the minimum profitable remaining runtime is workload- and path-dependent.

Transparent stateful migration. A common checkpoint-restart interface uses CRIU for unmodified process workloads and native checkpoints for LLM training and Dendro-GR.. Asynchronous pre-staging moves immutable datasets and model weights *while the job keeps running*; only volatile state crosses the stop-the-world window. Accepted migrations transfer both execution state and task ownership, while inter-site links are modeled by bandwidth, latency, distance, and cost per gigabyte.

4 RESULTS AND CONTRIBUTIONS

We evaluate on CloudLab bare metal (`tc`/`netem`-shaped links; twelve emulated world regions replaying ElectricityMaps traces at 15-minute resolution) [3] and on Google Cloud (seven `e2-medium` VMs in seven regions, with real routing and egress fees) [5], running PyTorch LLM-style training, Dendro-GR numerical-relativity solver [4], and pyperformance [12] workloads. We report medians of five trials over 24–72 h traces.

Magellan reduces lifecycle carbon by up to **92.8% (13×)** relative to static execution while reducing mean makespan by **11.6%** at approximately neutral monetary cost. On Google Cloud, it approaches the carbon footprint of the consistently clean France baseline while avoiding its roughly 3× waiting-time penalty. Under measured capacity constraints, savings remain **57.9%** at low utilization and degrade gracefully as the cluster saturates. Scaled to 100 concurrent tasks, destination-side arbitration prevents oversubscription while priced bandwidth naturally staggers migrations. Microbenchmarks show transfer dominating migration latency at 1 Gbps and checkpoint/restore dominating at 10 Gbps.

Our contributions are: (1) a multi-objective cost model that unifies carbon, money, makespan, and migration overhead under an explicit budget cap; (2) a fully decentralized scheduling architecture with source-side bidding and destination-side arbitration; (3) a checkpoint-restart runtime supporting CRIU and native workload checkpoints, requiring no source changes; (4) validation on both bare-metal and commercial-cloud infrastructure rather than in simulation; and (5) resource-aware coordination of competing migrations without destination oversubscription. GPU and RDMA state, cross-architecture restore, and predictive rather than reactive control are our next steps.

REFERENCES

[1] F. Andrijauskas, I. Filigoi, D. Davila, et al. 2024. CRIU: Checkpoint Restore in Userspace for computational simulations and scientific applications. *EPJ Web of Conferences* 295 (2024), 07046. https://doi.org/10.1051/epjconf/202429507046 26th International Conference on Computing in High Energy & Nuclear Physics (CHEP 2023).

[2] Jason Ansel, Kapil Arya, and Gene Cooperman. 2009. DMTCP: Transparent Checkpointing for Cluster Computations and the Desktop. In *2009 IEEE International Symposium on Parallel & Distributed Processing (IPDPS)*. IEEE, 1–12. https://doi.org/10.1109/IPDPS.2009.5161091

[3] Dmitry Duplyakin, Robert Ricci, Aleksander Maricq, Gary Wong, Jonathon Duerig, Eric Eide, Leigh Stoller, Mike Hibler, David Johnson, Kirk Webb, et al. 2019. The design and operation of {CloudLab}. In *2019 USENIX annual technical conference (USENIX ATC 19)*. 1–14. https://dl.acm.org/doi/10.5555/3358807.3358809

[4] Milinda Fernando, David Neilsen, Eric Hirschmann, and Hari Sundar. 2019. A Scalable Framework for Adaptive Computational General Relativity on Heterogeneous Clusters. In *Proceedings of the 33rd ACM International Conference on Supercomputing (ICS '19)*. ACM, Phoenix, AZ, USA, 1–12. https://doi.org/10.1145/3330345.3330382

[5] Google LLC. 2025. Google Cloud Platform. https://cloud.google.com. Accessed: 2025-12-09.

[6] Walid A. Hanafy, Qianlin Liang, Noman Bashir, and Prashant Shenoy. 2024. Going Green for Less Green: Optimizing the Cost of Reducing Cloud Carbon Emissions. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '24)*. ACM, San Diego, CA, USA, 544–559. https://doi.org/10.1145/3620665.3640397

[7] Syed Mafooq Ul Hassan, Marios Touloupou, Jacopo Castellini, Pablo Strasser, Alexandros Kalousis, and Herodotos Herodotou. 2026. DARO: An Auction-Based Multi-Agent Reinforcement Learning Framework for Task Scheduling in the Cloud Continuum. In *Proceedings of the 16th International Conference on Cloud Computing and Services Science*. 293–300. https://doi.org/10.5220/0014920000004039

[8] Nicola Jones. 2018. How to stop data centres from gobbling up the world's electricity. *Nature* 561 (2018), 163–166. https://doi.org/10.1038/d41586-018-06610-y

[9] A. Hasan Mahmud and S. S. Iyengar. 2016. A Distributed Framework for Carbon and Cost Aware Geographical Job Scheduling in a Hybrid Data Center Infrastructure. In *2016 IEEE International Conference on Autonomic Computing (ICAC)*. 75–84. https://doi.org/10.1109/ICAC.2016.21

[10] Diptyaroop Maji, Ben Pfaff, Rajagopal Sreenivasan, Colleen Josephson, et al. 2023. Bringing Carbon Awareness to Multi-cloud Application Delivery. In *2nd Workshop on Sustainable Computer Systems (HotCarbon '23)*. ACM, Boston, MA, USA. https://groups.cs.umass.edu/ramesh/wp-content/uploads/sites/3/2023/08/a6-maji.pdf

[11] Zicong Miao, Lei Liu, Haijing Nan, Weize Li, et al. 2024. Energy and carbon-aware distributed machine learning tasks scheduling scheme for the multi-renewable energy-based edge-cloud continuum. *Science and Technology for Energy Transition* 79, 1–18. https://doi.org/10.2516/stet/2024019

[12] Python Software Foundation. 2026. pyperformance: Python Performance Benchmark Suite. https://github.com/python/pyperformance. Accessed: 2026-09-17.

[13] Elvis Rodrigues, Jacob Goldverg, and Tevfik Kosar. 2025. Carbon-Aware Temporal Data Transfer Scheduling Across Cloud Datacenters. arXiv:2506.04117 [cs.DC] https://arxiv.org/abs/2506.04117

[14] Sven Smolka, Leon Wißenberg, and Zoltán Ádám Mann. 2023. EdgeDecAp: An Auction-Based Decentralized Algorithm for Optimizing Application Placement in Edge Computing. *J. Parallel and Distrib. Comput.* 175 (2023), 22–36. https://doi.org/10.1016/j.jpdc.2023.01.002

[15] Abel Souza, Nikhil Nambiar, Erik Elmroth, et al. 2023. CASPER: Carbon-Aware Scheduling and Provisioning for Distributed Web Services. In *2023 International Green and Sustainable Computing Conference (IGSC)*. IEEE, 1–8. https://doi.org/10.1109/IGSC61655.2023.00015

[16] Kaiqiang Xu, Decang Sun, Han Tian, Junxue Zhang, and Kai Chen. 2025. GREEN: Carbon-efficient Resource Scheduling for Machine Learning Clusters. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. USENIX Association, Philadelphia, PA. https://www.usenix.org/conference/nsdi25/presentation/xu-kaiqiang