Carbon- and Precedence-Aware Scheduling for Data Processing Clusters

Adam Lechowicz
University of Massachusetts Amherst
USA
alechowicz@cs.umass.edu

Rohan Shenoy
University of California Berkeley
USA
rohan.shenoy@berkeley.edu

Noman Bashir
Massachusetts Institute of Technology
USA
nbashir@mit.edu

Mohammad Hajiesmaili
University of Massachusetts Amherst
USA
hajiesmaili@cs.umass.edu

Adam Wierman
California Institute of Technology
USA
adamw@caltech.edu

Christina Delimitrou
Massachusetts Institute of Technology
USA
delimitrou@csail.mit.edu

Abstract

As large-scale data processing workloads continue to grow, their carbon footprint raises concerns. Prior research on carbon-aware schedulers has focused on shifting computation to align with the availability of low-carbon energy, but these approaches assume that each task can be executed independently. In contrast, data processing jobs have precedence constraints that complicate decisions, since delaying an upstream "bottleneck" task to a low-carbon period also blocks downstream tasks, impacting makespan. In this paper, we show that carbon-aware scheduling for data processing benefits from knowledge of both time-varying carbon and precedence constraints. Our main contribution is PCAPS, a carbon-aware scheduler that builds on state-of-the-art scoring or probability-based techniques – in doing so, it explicitly relates the structural importance of each task against the time-varying characteristics of carbon intensity. To illustrate gains due to fine-grained task-level scheduling, we also study CAP, a wrapper for any carbon-agnostic scheduler that generalizes the provisioning ideas of PCAPS. Both techniques allow a user-configurable priority between carbon and makespan, and we give basic analytic results to relate the trade-off between these objectives. Our prototype on a 100-node Kubernetes cluster shows that a moderate configuration of PCAPS reduces carbon footprint by up to 32.9% without significantly impacting total efficiency.

CCS Concepts: • **Software and its engineering** → **Scheduling**; • **Social and professional topics** → *Sustainability*.

Keywords: precedence constraints, data processing, carbon-aware scheduling

SIGCOMM '25, Coimbra, Portugal

ACM ISBN 979-8-4007-1524-2/2025/09
https://doi.org/10.1145/3718958.3750478

ACM Reference Format:
Adam Lechowicz, Rohan Shenoy, Noman Bashir, Mohammad Hajiesmaili, Adam Wierman, and Christina Delimitrou. 2025. Carbon- and Precedence-Aware Scheduling for Data Processing Clusters. In *ACM SIGCOMM 2025 Conference (SIGCOMM '25), September 8–11, 2025, Coimbra, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3718958.3750478

1 Introduction

Concerns about the climate impact of machine learning (ML) and artificial intelligence (AI) models have primarily considered the footprint of training [11, 22] or, in some cases, inference [15]. However, as model sizes have ballooned, the *data processing* tasks that must be completed before training account for almost one-third of the cumulative computation for an AI model during its life cycle [23].

Therefore, efforts towards sustainable AI must consider and optimize the carbon footprint of data processing. Even beyond sustainability, companies such as Microsoft have implemented *internal carbon pricing* for short- and long-term decisions [7, 19] that assign financial responsibility for operational CO_2 emissions. In data centers, current schedulers do not consider the time-varying aspect of carbon intensity and the resulting compute-carbon impact – this must change to accommodate these additional concerns.

Data processing (e.g., Spark) workloads are composed of *precedence-constrained tasks* where e.g., the outputs of one operation are the inputs to another [24], forming a directed acyclic dependency graph (DAG). Optimal scheduling of precedence-constrained jobs is known to be NP-hard [14], so existing work is split between simple settings that are studied theoretically to obtain approximation guarantees [3, 4, 13, 16] and experimental settings where data-driven heuristics and evolutionary approaches have been developed [5, 10, 12, 18, 25]. A select few works have considered multi-objective variants of the problem that balance e.g., energy-efficiency or cost against performance [8, 17, 20]. *Carbon-efficiency*, however, often conflicts with energy-efficiency: the time-varying nature of the grid means that minimizing carbon may require energy-inefficient processing "bursts" during

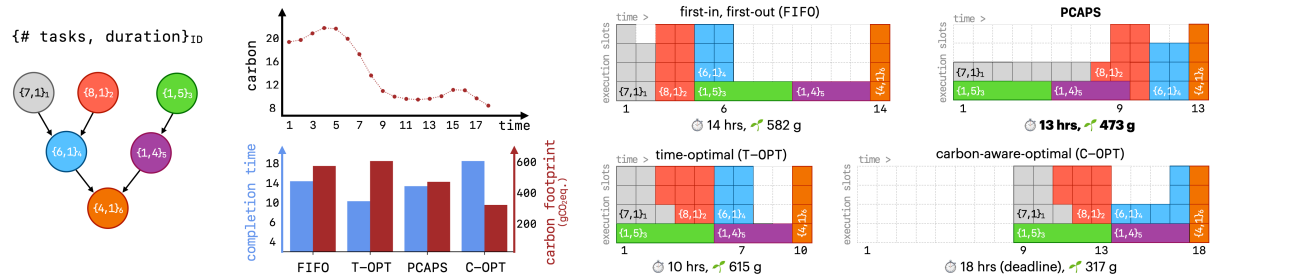

Figure 1. Four schedules for a motivating DAG and 18-hour carbon intensity trace (on the left hand side). Compared to a carbon-agnostic first-in first-out (FIFO) scheduler, the time-optimal schedule (T-OPT) prioritizes green and purple stages early to reduce makespan. A carbon-aware-optimal schedule (C-OPT) with a *deadline* to finish the DAG within 18 hours reduces carbon emissions by 51.2%, at the expense of increasing makespan by 28.5% compared to FIFO. By prioritizing green and purple stages during high-carbon periods, PCAPS reduces carbon by 23.1% and still reduces makespan by 7% compared to FIFO.

low-carbon periods. To address the multi-objective setting of carbon-aware scheduling for precedence-constrained tasks, we propose a middle-ground approach: namely, we seek an interpretable framework that comes with provable trade-off guarantees between carbon emissions and performance while catering to realistic scenarios. Our main contributions are as follows:

1. PCAPS (**P**recedence- and **C**arbon-**A**ware **P**rovisioning and **S**cheduling), a carbon-aware scheduler that defines a notion of relative importance for each task to make fine-grained scheduling decisions and achieve a favorable trade-off between carbon savings and performance.
2. CAP (Carbon-Aware Provisioning), a simplification of PCAPS that reconfigures the resources available to data processing jobs without replacing an existing scheduler, making it easier to implement (*see full paper*).
3. We evaluate PCAPS and CAP in experiments using a Spark simulator and prototypes for Spark on Kubernetes. [1]

2 Carbon-Aware DAG Scheduling Problem

Each job is represented as a directed acyclic graph (DAG) $\mathcal{J} = \{\mathcal{V}, \mathcal{E}\}$, where each node in $\mathcal{V}$ is a task, and each edge in $\mathcal{E}$ encodes precedence constraints between tasks – e.g., for tasks $j, j' \in \mathcal{V}$, an edge $j \to j'$ indicates that j' cannot start until after j has completed. We index continuous time by $t \geq 0$. The goal of a typical scheduler is *performance* in terms of makespan or average job completion time. We additionally consider the goal of *carbon emissions* – given a time-varying carbon signal described by a function $c(t) : t \geq 0$, the objective is to minimize a combination of typical metrics (i.e., makespan) and the overall carbon footprint (on a global, cluster basis). In an online setting, future carbon values are unknown to the scheduler. We follow prior work [2] and assume there are constants L and U such that $L \leq c(t) \leq U : t \geq 0$. In practice, these values capture e.g., short-term forecasts of grid carbon conditions. Due to the additional objective of carbon, an scheduler must consider the time-varying carbon intensity while scheduling the nodes of a job DAG(s) to balance the goal of reducing carbon footprint against traditional metrics of performance – see Fig. 1 for an illustration of this trade-off for four different schedules.

3 Design & Evaluation

We briefly detail the design of our PCAPS scheduler and report our main experiment results – see our full paper for details.

The key idea of PCAPS is a notion of *relative importance* inferred from a set of probabilities or scores assigned to tasks that are ready to execute. Many state-of-the-art DAG schedulers (e.g,. Decima [18], Graphene [9]) operate by scoring (resp. assigning probabilities to) all tasks. In doing so, these techniques encode important information about DAG structure – a high score or probability is likely to indicate e.g., a bottleneck task. PCAPS builds on top of this existing intuition – for a given task v with score/probability p_v, we define the relative importance $r_v \coloneqq p_v / \max_{u \in \mathcal{A}} p_u \in [0, 1]$, where $\mathcal{A}$ denotes the set of tasks that are ready to execute.

For carbon-awareness, PCAPS is designed to ramp up during low-carbon periods and ramp down during high-carbon periods, while ensuring that certain bottleneck tasks are still scheduled during high-carbon periods to avoid adversely increasing makespan. To do so, PCAPS takes cues from the related theoretical literature on carbon-aware scheduling [2]

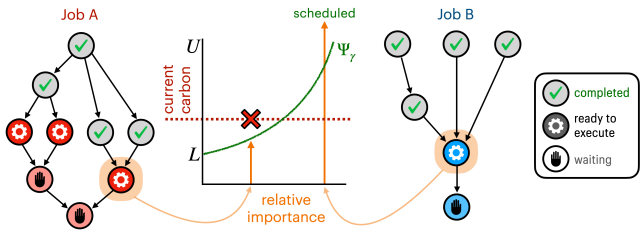

Figure 2. Illustrating PCAPS's carbon-awareness filter. Jobs A and B are DAGs found in TPC-H queries and Alibaba traces [1, 21]. Highlighted nodes detail two possible outcomes. In job A, the highlighted node has low relative importance, so it is deferred. In contrast, job B's highlighted node is a *bottleneck* task with high relative importance: even when the current carbon intensity is high, such tasks are scheduled to avoid increasing makespan.

[1] Full paper and code: github.com/umass-solar/carbon-aware-dag.

and defines a threshold function Ψ_γ that considers the current carbon and the relative importance of a task. $\gamma \in [0, 1]$ is a user-specified parameter that controls the "strictness" of the function: $\gamma = 0$ recovers carbon-agnostic actions, while $\gamma = 1$ is maximally carbon-aware. Given a task's relative importance r, we define $\Psi_\gamma(r) \coloneqq (\gamma L + (1-\gamma)U) + [U - (\gamma L + (1-\gamma)U)] \frac{\exp(\gamma r)-1}{\exp(\gamma)-1}$. PCAPS uses the Ψ_γ function in a carbon-awareness filter that decides which tasks can be scheduled – see Fig. 2 for an intuition of this. In the full paper, we give analytic bounds to characterize the trade-off between carbon savings and makespan for PCAPS.

We implement PCAPS and CAP as a prototype for Spark on Kubernetes, and conduct additional large-scale experiments in a realistic Spark simulator. We use workloads from TPC-H benchmarks [21] and Alibaba production DAG traces [1], alongside historical carbon traces for six grid regions from Electricity Maps [6]. In addition to the default scheduling behavior of Spark on Kubernetes, we implement the Decima scheduler [18] as a carbon-agnostic baseline. Decima uses reinforcement learning to assign a probability to each task – in our PCAPS implementation, we use Decima's probabilities for the computation of relative importance. Table 1 reports carbon reduction, makespan, and job completion time metrics for all schedulers in our prototype experiments – see the full paper for individual experiments and results.

Table 1. Summary of prototype results averaged over all tested carbon traces. Each metric is normalized with respect to the Spark / Kubernetes default behavior. PCAPS and CAP are configured to be moderately carbon aware (i.e., $\gamma = 0.5$).

Metric normalized w.r.t. Default	**Default**	**Decima** [18]	**CAP**	**PCAPS**
Carbon Reduction (%)	0%	1.2%	24.7%	**32.9%**
Avg. Makespan	1.0	0.857	1.126	**1.013**
Avg. JCT	1.0	0.852	1.996	**1.381**

References

[1] Alibaba. 2018. Cluster data collected from production clusters in Alibaba for cluster management research. https://github.com/alibaba/clusterdata/tree/master/cluster-trace-v2018

[2] Roozbeh Bostandoost, Adam Lechowicz, Walid A. Hanafy, Noman Bashir, Prashant Shenoy, and Mohammad Hajiesmaili. 2024. LACS: Learning-Augmented Algorithms for Carbon-Aware Resource Scaling with Uncertain Demand. In *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems* (Singapore, Singapore) *(e-Energy '24)*. Association for Computing Machinery, New York, NY, USA, 27–45. https://doi.org/10.1145/3632775.3661942

[3] Sami Davies, Janardhan Kulkarni, Thomas Rothvoss, Jakub Tarnawski, and Yihao Zhang. 2020. Scheduling with Communication Delays via LP Hierarchies and Clustering. arXiv:2004.09682 [cs.DS] https://arxiv.org/abs/2004.09682

[4] Sami Davies, Janardhan Kulkarni, Thomas Rothvoss, Jakub Tarnawski, and Yihao Zhang. 2021. *Scheduling with Communication Delays via LP Hierarchies and Clustering II: Weighted Completion Times on Related Machines*. Society for Industrial and Applied Mathematics, 2958–2977. https://doi.org/10.1137/1.9781611976465.176

[5] Lawrence Davis. 2014. Job shop scheduling with genetic algorithms. In *Proceedings of the first International Conference on Genetic Algorithms and their Applications*. Psychology Press, 136–140.

[6] Electricity Maps. 2023. Electricity Map. https://www.electricitymap.org/map.

[7] Jessica Fan, Werner Rehm, Giulia Siccardo, and McKinsey & Company. 2021. The state of internal carbon pricing. https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/the-state-of-internal-carbon-pricing.

[8] Íñigo Goiri, Kien Le, Thu D. Nguyen, Jordi Guitart, Jordi Torres, and Ricardo Bianchini. 2012. GreenHadoop: Leveraging Green Energy in Data-Processing Frameworks. In *Proceedings of the 7th ACM European Conference on Computer Systems* (Bern, Switzerland) *(EuroSys '12)*. Association for Computing Machinery, New York, NY, USA, 57–70. https://doi.org/10.1145/2168836.2168843

[9] Robert Grandl, Srikanth Kandula, Sriram Rao, Aditya Akella, and Janardhan Kulkarni. 2016. GRAPHENE: Packing and Dependency-Aware Scheduling for Data-Parallel Clusters. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. USENIX Association, Savannah, GA, 81–97. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/grandl_graphene

[10] Nathan Grinsztajn, Olivier Beaumont, Emmanuel Jeannot, and Philippe Preux. 2020. Geometric deep reinforcement learning for dynamic DAG scheduling. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 258–265. https://doi.org/10.1109/ssci47803.2020.9308278

[11] Walid A. Hanafy, Qianlin Liang, Noman Bashir, David Irwin, and Prashant Shenoy. 2023. CarbonScaler: Leveraging Cloud Workload Elasticity for Optimizing Carbon-Efficiency. *Proc. of the ACM on Measurement and Analysis of Computing Systems* 7, 3 (Dec 2023). arXiv:2302.08681 [cs.DC]

[12] Muhammed Tawfiqul Islam, Shanika Karunasekera, and Rajkumar Buyya. 2021. Performance and cost-efficient spark job scheduling based on deep reinforcement learning in cloud computing environments. *IEEE Transactions on Parallel and Distributed Systems* 33, 7 (2021).

[13] Alexandra Anna Lassota, Alexander Lindermayr, Nicole Megow, and Jens Schlöter. 2023. Minimalistic Predictions to Schedule Jobs with Online Precedence Constraints. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 18563–18583. https://proceedings.mlr.press/v202/lassota23a.html

[14] J. K. Lenstra and A. H. G. Rinnooy Kan. 1978. Complexity of Scheduling under Precedence Constraints. *Operations Research* 26, 1 (1978), 22–35. http://www.jstor.org/stable/169889

[15] Baolin Li, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. 2023. Clover: Toward Sustainable AI with Carbon-Aware Machine Learning Inference Service. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Denver, CO, USA) *(SC '23)*. Association for Computing Machinery, New York, NY, USA, Article 20, 15 pages.

[16] Shi Li. 2017. Scheduling to Minimize Total Weighted Completion Time via Time-Indexed Linear Programming Relaxations. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 283–294. https://doi.org/10.1109/focs.2017.34

[17] Wenyu Liu, Yuejun Yan, Yimeng Sun, Hongju Mao, Ming Cheng, Peng Wang, and Zhaohao Ding. 2023. Online job scheduling scheme for low-carbon data center operation: An information and energy nexus perspective. *Applied Energy* 338 (2023), 120918.

[18] Hongzi Mao, Malte Schwarzkopf, Shaileshh Bojja Venkatakrishnan, Zili Meng, and Mohammad Alizadeh. 2019. Learning Scheduling Algorithms for Data Processing Clusters. In *Proceedings of the ACM Special Interest Group on Data Communication* (Beijing, China) *(SIGCOMM '19)*. Association for Computing Machinery, New York, NY, USA, 270–288. https://doi.org/10.1145/3341302.3342080

[19] Brad Smith and Microsoft Corporation. 2019. We're increasing our carbon fee as we double down on sustainability. https://blogs.microsoft.com/on-the-issues/2019/04/15/were-increasing-our-carbon-fee-as-we-double-down-on-sustainability/.

[20] Yu Su, Vivek Anand, Jannie Yu, Jian Tan, and Adam Wierman. 2024. Learning-Augmented Energy-Aware List Scheduling for Precedence-Constrained Tasks. *ACM Trans. Model. Perform. Eval. Comput. Syst.* (2024). https://doi.org/10.1145/3680278

[21] TPC-H. 2018. The TPC-H Benchmarks. https://www.tpc.org/tpch/

[22] 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

[23] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable AI: Environmental Implications, Challenges and Opportunities. *Proceedings of Machine Learning and Systems (MLSys)* 4 (2022), 795–813.

[24] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2012. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation* (San Jose, CA) *(NSDI'12)*. USENIX Association, USA, 2.

[25] Yunfan Zhou, Xijun Li, Jinhong Luo, Mingxuan Yuan, Jia Zeng, and Jianguo Yao. 2022. Learning to Optimize DAG Scheduling in Heterogeneous Environment. In *2022 23rd IEEE International Conference on Mobile Data Management (MDM)*. 137–146. https://doi.org/10.1109/MDM55031.2022.00040