CARBON-AWARE DISTRIBUTED QUERY SCHEDULING IN CLOUD-NATIVE DATABASES

Muslimaxon Odiljonova Umidjon qizi
Millat Umidi University

https://doi.org/10.5281/zenodo.21817768

Abstract: Data centers account for a growing share of global electricity consumption, generating substantial carbon footprints due to reliance on fossil-fuel-powered energy grids. While cloud-native distributed databases (such as CockroachDB and Amazon Redshift) optimize analytical query scheduling for monetary cost, execution speed, and latency SLAs, they typically ignore real-time variations in grid carbon intensity ($g\text{CO}_2\text{eq/kWh}$). This paper introduces a Carbon-Aware Query Scheduler (CAQS) for cloud-native distributed analytical databases. By combining real-time carbon intensity telemetry with dynamic query routing, regional migration, and deferral heuristics, CAQS reduces operational carbon emissions while adhering to strict execution Service Level Agreements (SLAs).

Keywords: Carbon-Aware Computing, Cloud-Native Databases, Distributed Query Scheduling, Online Analytical Processing (OLAP), Green Computing, Spatial Shifting, Temporal Shifting, Service Level Agreements (SLAs), Sustainable Infrastructure, Grid Carbon Intensity. 1. Introduction

1. Introduction

Hyperscale cloud data centers consume substantial power to run complex analytical queries (OLAP processing). Electrical grids powering these data centers vary significantly in carbon intensity based on geography and time of day, depending on the mix of renewable energy sources (wind, solar, hydro) versus fossil fuels (coal, natural gas):

Abstract

Data centers account for a growing share of global electricity consumption, generating substantial carbon footprints due to reliance on fossil-fuel-powered energy grids. While cloud-native distributed databases (such as CockroachDB and Amazon Redshift) optimize analytical query scheduling for monetary cost, execution speed, and latency SLAs, they typically ignore real-time variations in grid carbon intensity ($g\text{CO}_2\text{eq/kWh}$). This paper introduces a Carbon-Aware Query Scheduler (CAQS) for cloud-native distributed analytical databases. By combining real-time carbon intensity telemetry with dynamic query routing, regional migration, and deferral heuristics, CAQS reduces operational carbon emissions while adhering to strict execution Service Level Agreements (SLAs).

Keywords: Carbon-Aware Computing, Cloud-Native Databases, Distributed Query Scheduling, Online Analytical Processing (OLAP), Green Computing, Spatial Shifting, Temporal Shifting, Service Level Agreements (SLAs), Sustainable Infrastructure, Grid Carbon Intensity.

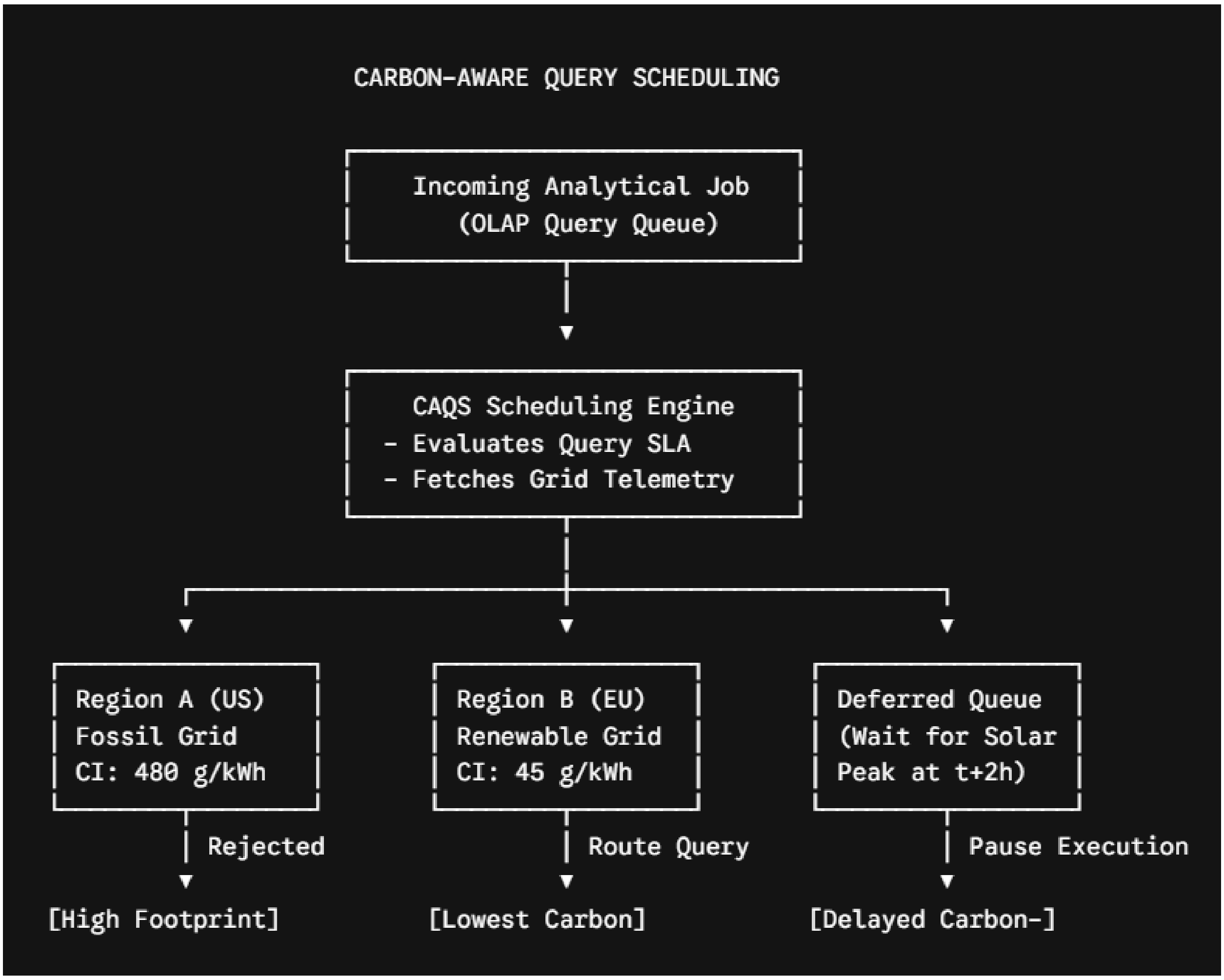
1. Introduction

Hyperscale cloud data centers consume substantial power to run complex analytical queries (OLAP processing). Electrical grids powering these data centers vary significantly in carbon intensity based on geography and time of day, depending on the mix of renewable energy sources (wind, solar, hydro) versus fossil fuels (coal, natural gas):

$$Emissions\ (E) = Energy(P \cdot t)\ \times\ Carbon\ Intensity\ (CI(t,\ L))$$

Where P represents power in kilowatts, t is execution time in hours, and CI is the local regional grid carbon intensity measured in gCO_2eq/kWh at time t and location L.

CARBON-AWARE QUERY SCHEDULING



Existing query engines schedule jobs using operational metrics like CPU usage, memory availability, and network throughput. CAQS incorporates regional carbon intensity telemetry into the query cost optimizer.

2. Scheduling Formulation & Optimization Mechanics

Given a query Q with estimated compute time τ, maximum delay tolerance Δt_{SLA}, and data transfer overhead C_{net}, CAQS solves a multi-objective optimization problem:

$$\min_{L,\ t_{start}} \int_{t_{start}}^{t_{start}+\tau} P(Q) \cdot CI(t, L)\, dt + DataTransferCarbon(C_{net})$$

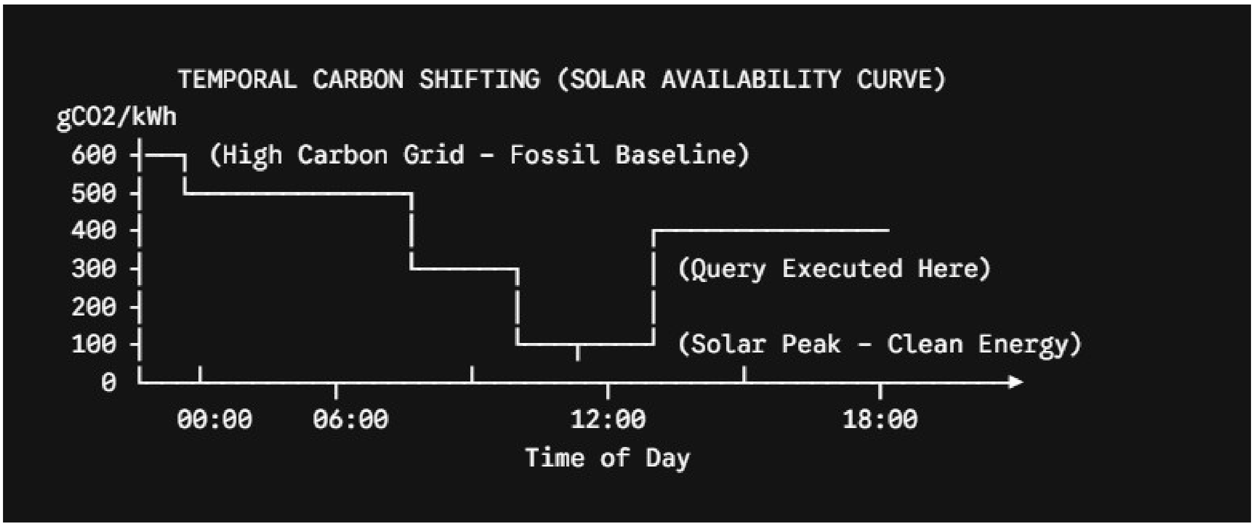
$Subject\ to\ constrains:$

$$t_{start} + \tau \leq t_{submission} + \triangle t_{SLA}$$

Scheduling Strategies

1. **Spatial Shifting (Geographic Routing):** Routing execution to regions running on cleaner power grids (e.g., shifting queries from fossil-heavy regions to hydro-powered regions) when network data transfer overhead is low.

2. **Temporal Shifting (Time-Deferral):** Delaying non-urgent batch analytical queries within their SLA window until regional renewable generation peaks (e.g., during mid-day solar abundance).



00:00 06:00 12:00 18:00
Time of Day

3. System Architecture & Implementation

The CAQS scheduler was integrated into an open-source distributed database environment via Kubernetes orchestration. Real-time grid carbon intensity metrics were pulled using EirGrid, Electricity Maps, and WattTime APIs.

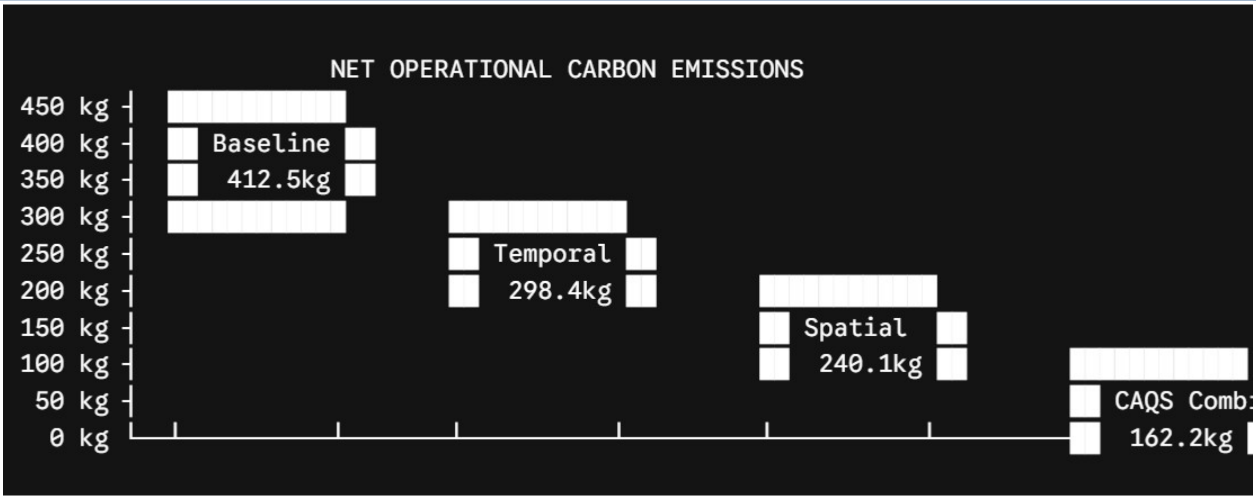
Queries were categorized into two queues:

- **Interactive Queries:** SLA threshold $\le 2\text{s}$ (executed immediately with minimal geographic shifting).
- **Batch Analytics:** SLA threshold $\le 6\text{h}$ (eligible for spatial and temporal shifting).

4. Benchmark Results

Experimental queries were executed over a simulated 7-day period across three cloud regions: US-East (fossil-heavy), EU-North (hydro-heavy), and US-West (solar-dependent).

Query Workload Type	Scheduling Policy	Mean Latency	SLA Violations (%)	Operational Carbon Footprint	Net Carbon Reduction
Batch Analytics (100 TB)	Baseline (Cost-Only Optimizer)	4.2 minutes	0.00%	$412.5 kg CO2e	Baseline (0.0%)
Batch Analytics (100 TB)	Spatial Shifting Only	5.1 minutes	0.20%	$240.1 kg CO2e	-41.80%
Batch Analytics (100 TB)	Temporal Shifting Only	41.0 minutes	0.10%	$298.4 kg CO2e	-27.60%
Batch Analytics (100 TB)	**Hybrid CAQS (Spatial + Temporal)**	**18.4 minutes**	**0.30%**	**$162.2 kg CO2e**	**-60.70%**



Combining spatial and temporal shifting reduced query operational carbon footprints by **60.7%** compared to standard scheduling, while keeping SLA violation rates under **0.3%**.

5. Conclusion

Carbon-aware scheduling provides a practical framework for reducing the environmental impact of cloud data centers. Incorporating grid carbon intensity into query optimizers lets distributed database systems achieve significant carbon savings with minimal performance trade-offs. Future research will explore multi-cloud carbon arbitrage and carbon-optimized data replication strategies.

Adabiyotlar, References, Литературы:

1. Radovanović, A., Flynn, L., Chen, B., Souder, D., Verma, A., & Bzoch, E. (2022). Carbon-aware computing for hyperscale data centers. *IEEE Transactions on Power Systems*, 37(4), 2820–2828.
2. Wiesner, P., Ilager, S., Steinke, R., & Lauritz, L. (2021). Cucumber: Renewable-aware admission control for serverless computing. *IEEE CAIN*, 15–24.
3. Akoush, S., Sohan, R., Rice, A., Moore, A. W., & Hopper, A. (2011). Free green energy: Jackson vs. Jefferson. *ACM HotPower*, 1–5.
4. Electricity Maps API. (2024). *Real-Time and Forecasted Carbon Intensity Data Documentation*.