CaRE: Towards Carbon and Resource Efficient Orchestration at the Cloud-Edge Continuum

Georgia Christofidi Francisco Álvarez Terribas Jesus Alberto Omaña Iglesias

Nicolas Kourtellis Thaleia Dimitra Doudali

1. Problem Space

Challenge: Increased Carbon Emissions due to **exponential growth** of Computing.

a BigScience initiative · Meta · BLOOM · 176B params · 59 languages · Open-access

Key drivers:
- ML applications
- Generative AI
- Video streaming

AI Model	Carbon Impact of Training*	Real-word equivalent example
GPT-3	500 metric tons of CO2eq.[1]	500 round-trip flights from Madrid to New York for one passenger.
GPT-4	12,456 - 14,994 metric tons CO2eq (estimated).[2]	50-60 fully loaded Boeing 747 flights.

*Training only accounts for 43% of lifecycle carbon emissions. [1]

Problem: **Resource**, **Performance,** and **Cost** are compromised when reducing CO_2.

Resource Awareness
- Resource Waste
- Energy Inefficiency
- Increased Cost

Idle!

Temporal Shifting

Cost Awareness

Spatial Shifting

Small national companies need **additional budget** to rent remote resources in greener regions.

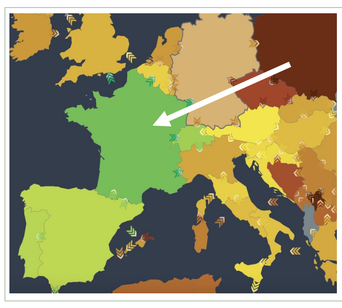
Performance Awareness

Only **specific types** of jobs can be shifted in time.

Not all workloads can wait!

Takeaway: Optimizing Carbon + Resource + Cost + Performance = Harder than it looks.

Solution: **Spatial** and **Temporal** Workload Shifting.



[3] Spatial Shifting ☑

Fossil-fuel-heavy regions → Workload Migration → Greener areas

Temporal Shifting ☑

Pause with no strong latency requirements (e.g., batch jobs)

Resume when green energy available.

Sources [1]: Beyond Efficiency: Scaling AI Sustainably
[2]: https://towardsdatascience.com/the-carbon-footprint-of-gpt-4-d6c676eb21ae
[3] https://app.electricitymaps.com/map/72h

2. Experimental Methodology

1. Experimental Methodology

Usecase: Company with entire cloud-edge infrastructure deployed in Spain.

Location	Carbon Intesity
Spain ES	206 gCO2eq/kWh
Sweden SE	20 gCO2eq/kWh

The lower the better

Goal: Quantify the additional **cost ($)** to rent resources in Sweden to reduce the **carbon footprint**.

2. Experimental details

Applications (using the Microservices benchmark **DeathStarBench**)

Social Network: Users send requests to compose posts.

Media streaming: Movie platform where users can log in and upload movie reviews.

Workload 10 minutes

- 1,000 requests to each application
- Time steps follow a Poisson distribution, emulating multiple concurrent users

3. Preliminary Results

1. Composing and uploading a movie review is **more computationally demanding** than creating a social media post.

Application	AVG Latency
Social Network	9.49 ms
Media Streaming	26.08 ms

2.89x

	App (Location)	Carbon (mgCO2eq)	Local ($/hr)
ES	Social Network (Spain)	72.72	0.0912
S	Social Network (Sweden)	7.06	0.0864
ES	Media Streaming (Spain)	166.17	0.0456
S	Media Streaming (Sweden)	16.13	0.0432

~10x ~2x

*Source: **Amazon EC2 On-Demand Pricing**. Hourly rate in the eu-south-2 region for Spain, eu-north-1 region for Sweden.

2. Running the applications in Sweden, is a much more **sustainable** solution.

*3. Hosting the media streaming in Sweden will lead to a **higher impact** in sustainability.*

4. **Double the budget** is needed for similar infrastructure in a different country. Users from Spain will connect first to the closest DC → the application runs on both locations.

Takeaway: Become **greener** → More **money**. Choose wisely what to offload!

We need an **application-specific solution** for the **carbon – cost trade-off.**

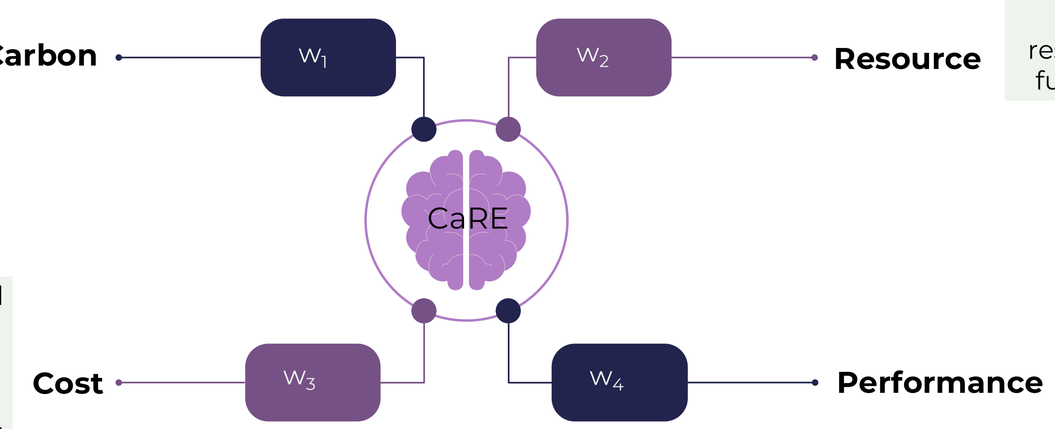
4. Proposed Approach

1. Minimizes emissions with spatial shifting (Data from Electricity Maps).

2. Minimizes idle resources by predicting future resource usage.

3. Minimizes the overall cost of the infrastructure (Data from On-Demand Plans for Amazon EC2).

4. Minimizes network and request execution latency, enforcing the SLAs.



CaRE **prioritizes** the optimization metrics according to the **specific application requirements** and the user preferences.

Current Application: **Microservices**

Future Work: Extend to **serverless** applications.

Takeaway: CaRE jointly optimizes the **carbon**, **resource** and **cost** efficiency of the workloads, complying with **SLAs**.

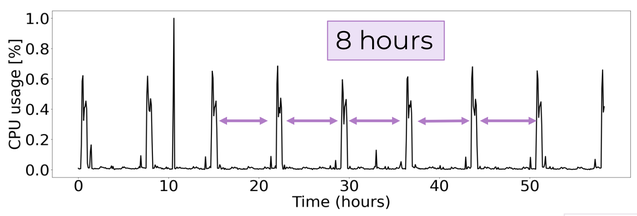
5. Challenge

1. Accurately Predicting Resource Usage. Proposed Approach: **Persistent Forecast**.

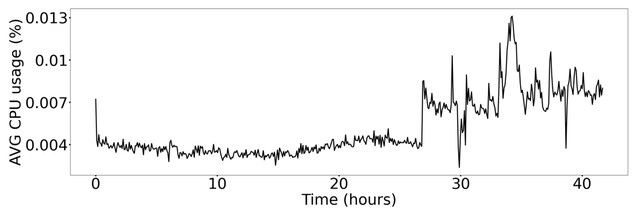
Assume **resource usage repeats itself periodically**.

Highly **accurate** on cloud data with average prediction error 7%.

User behaviours follow predictable cycles.



2. **Limitations** of the Persistent Forecast – **hard to predict patterns**.

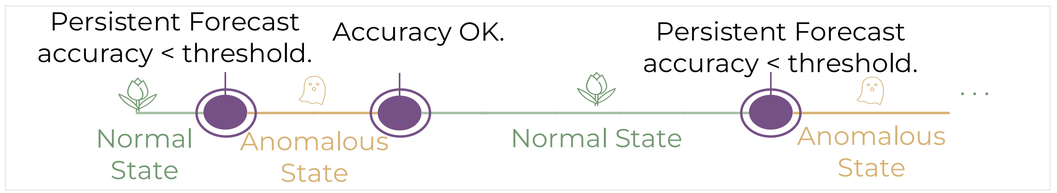


We deploy **anomaly detection techniques**, to predict highly dynamic resource usage.

3. Handling **Anomalies** with a **Two-Model Approach**.

When the persistent forecast accuracy drops below a threshold, we enter an **anomalous state**.

Fallback Mechanism that predicts:
- **Duration** of the anomaly.
- **Resource usage** during this time.



For the **anomaly detection model** we will explore a variety of ML and non-ML methods commonly used for anomaly detection.