Interoperability and Standardization of Carbon-Aware Practices in Cross-Cloud Situations

Szabolcs Varga
School of Enterprise Computing and Digital Transformation, TU Dublin, Ireland
X00218709@mytudublin.ie

Introduction

In the past ten years, cloud computing has grown exponentially and Data centers use up over 415 TWh of electricity each year. This is about 1,5 percent of the world's electricity consumption. Carbon-aware computing aims at aligning digital workloads with low-carbon energy availability, but it has been hindered by fragmented implementations and inconsistent standards among cloud providers. The thesis examines interoperability challenges in cross-cloud carbon-aware computing and proposes a framework for portable and verifiable workload scheduling across Google Cloud, AWS, and Azure **using an API application created by the author**. Through a literature-review, abstraction-layer design and empirical prototype evaluation, the study delivers guidelines and an open-source toolkit that allows practical multi-cloud carbon-aware scheduling.

Research Question and Hypothesis

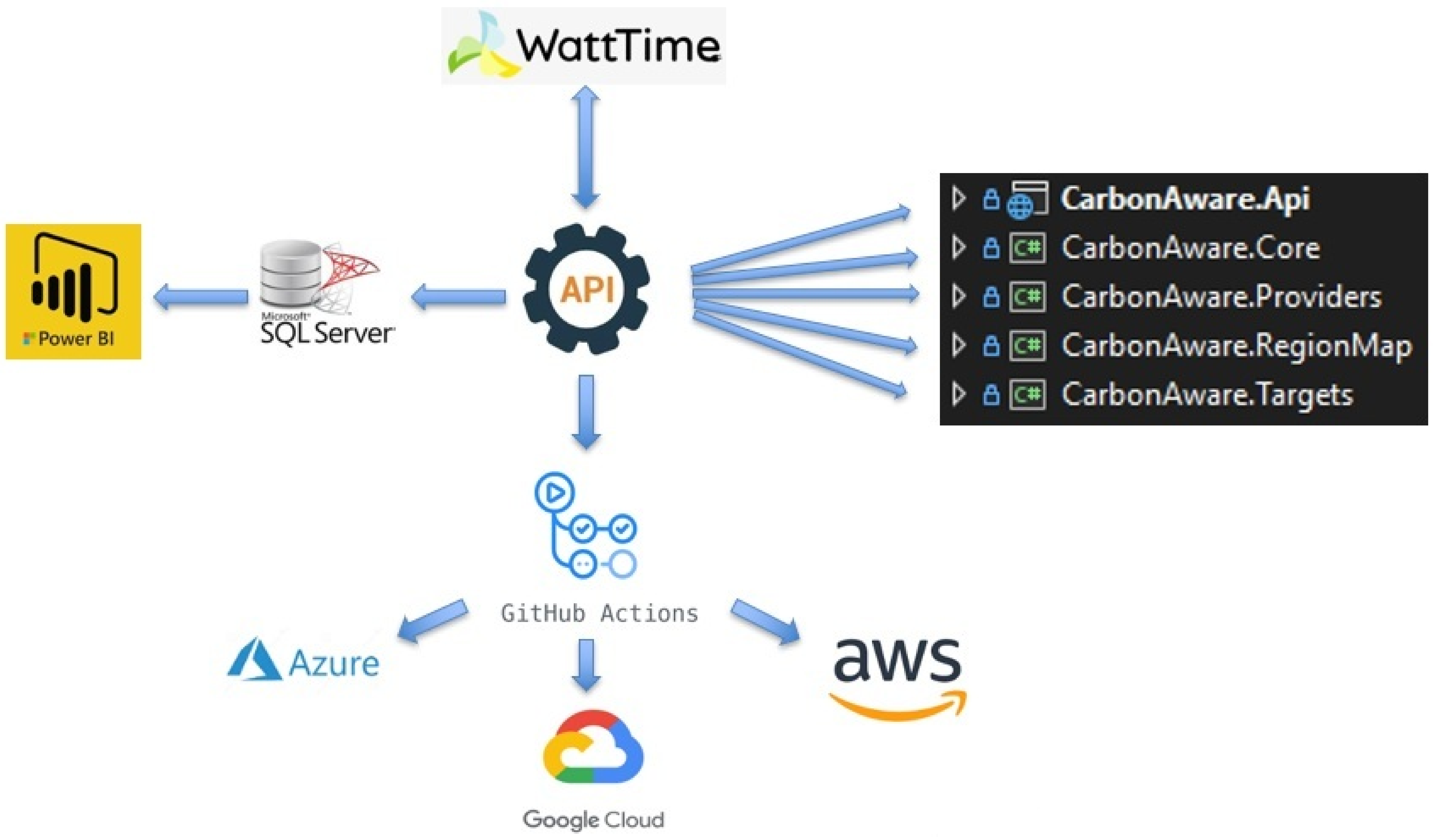
Research Question: How can carbon-aware workload scheduling be made interoperable and standardized across major cloud providers to promote scalable and sustainable computing? **Hypothesis:** By creating a vendor-agnostic abstraction layer that standardizes carbon intensity data formats, workload policies, and scheduling APIs, organizations can enable cross-cloud carbon-aware workload orchestration that is both technically feasible and environmentally beneficial.

Carbon Signals: Why MOER?

Carbon-aware scheduling is all about making smarter decisions that consider the real impact of our actions on the environment. This approach utilizes the **Marginal Operating Emissions Rate (MOER)** from **WattTime**, which measures the extra carbon emissions that result from increasing electricity demand at a specific time and place so the system can better pinpoint times when it's more eco-friendly to run workloads. This means we can schedule tasks across different cloud environments ultimately helping to reduce our carbon footprint.

Solution Architecture

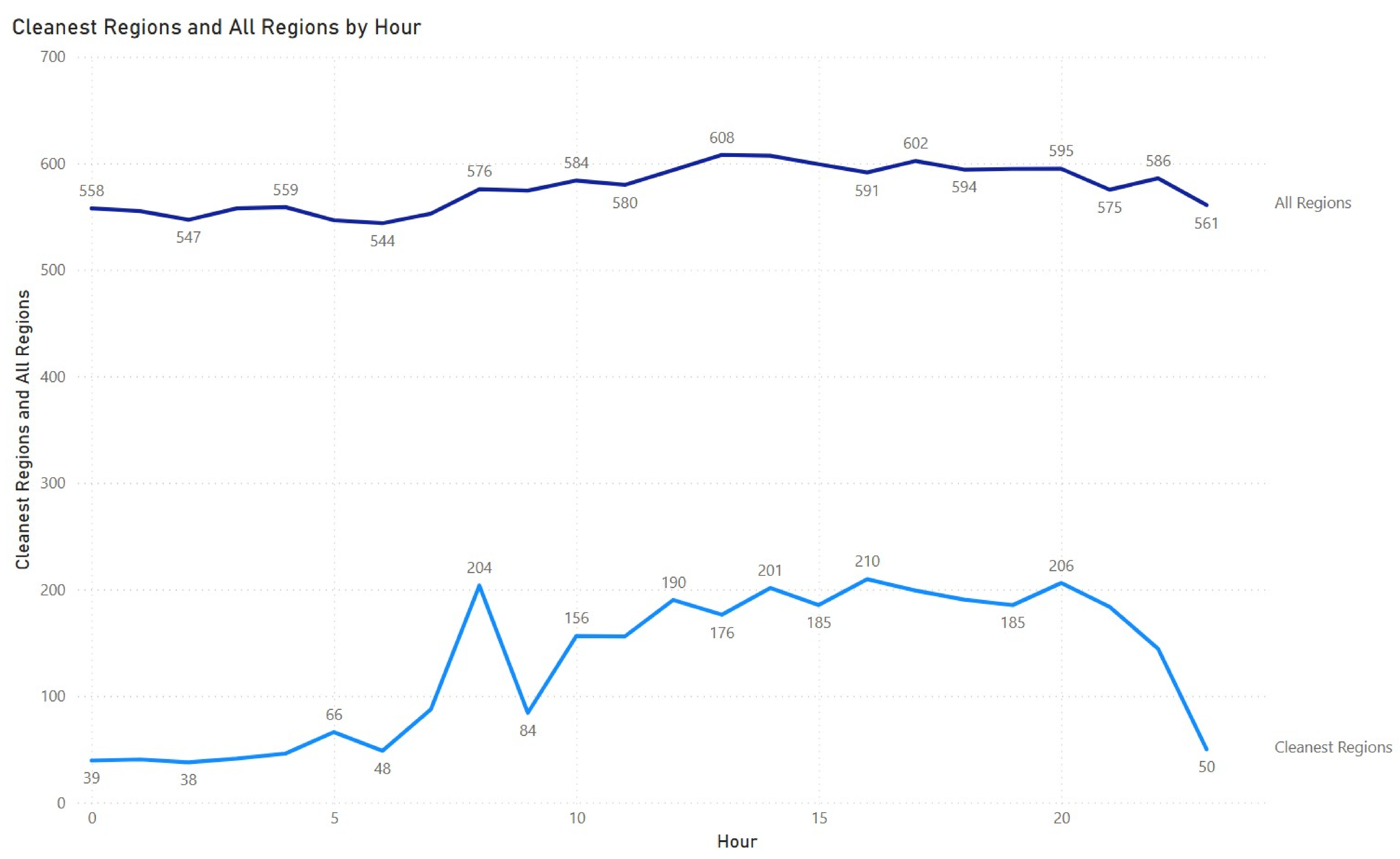
1. MOER signal acquisition from WattTime
2. API computation - advise
3. Github Actions call
4. IaC on the selected cloud provider
5. Logging/PowerBI - optional for troubleshooting/reports



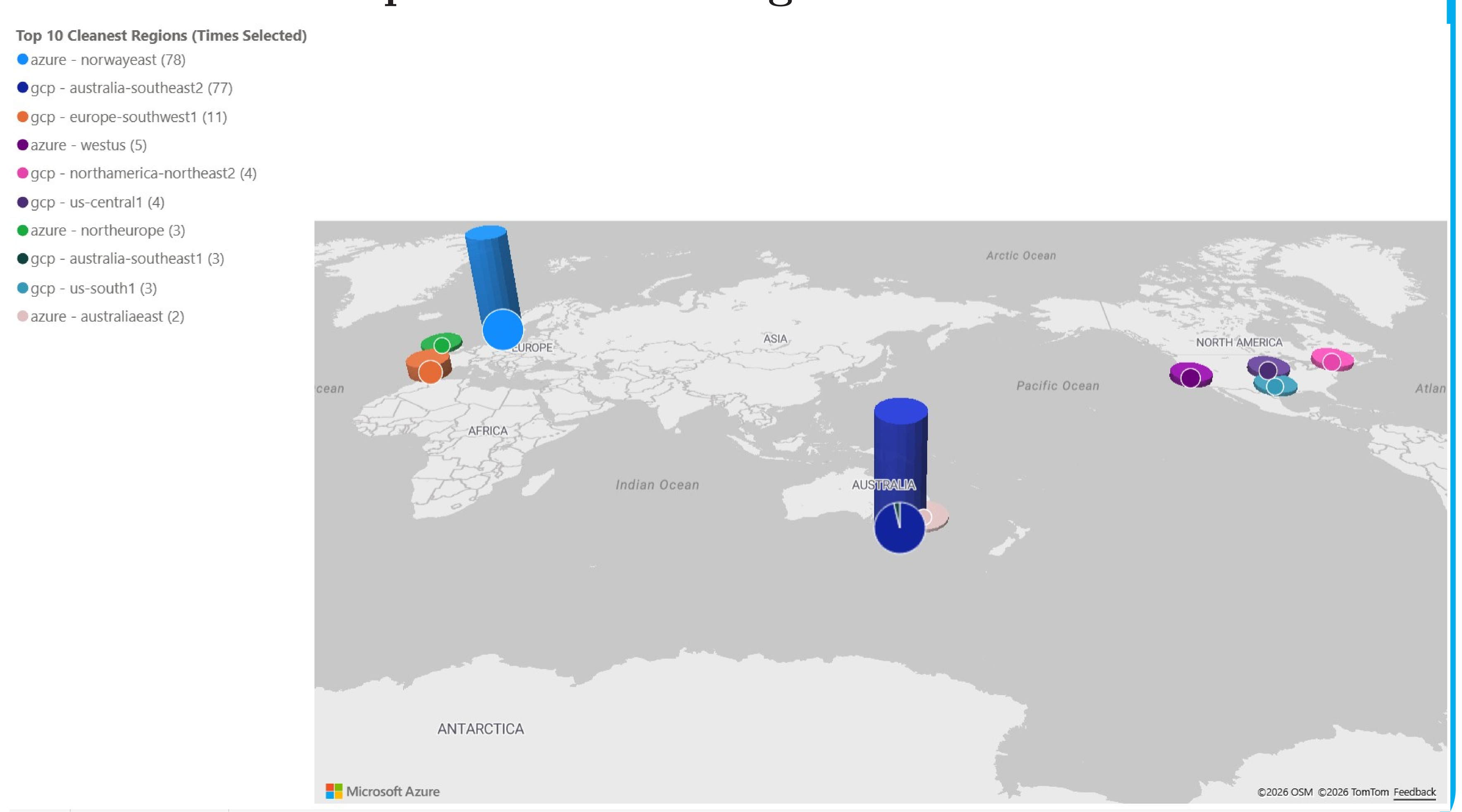
Testing Results

Testing Method Summary: the system was tested over three full 24-hour cycles, covering all available cloud regions. Each cycle ran the same workloads using different scheduling modes to allow fair comparison. Results were then analysed to understand emission savings, timing behaviour, and regional selection patterns. **First figure** shows the CO2 emissions of cleanest regions the application selected versus an average baseline over 24 hours timespan and the **second figure** shows the cleanest regions identified over the experimental runs.

CO2 emissions (g/Kwh) Cleanest vs All Regions



Top 10 Cleanest Regions Identified



Conclusions and Future Work

This thesis demonstrates that carbon-aware workload scheduling is both viable and effective. The Carbon-Aware API tool demonstrated how the carbon exposures could be dramatically reduced (up to 90%) automatically by exploiting a temporal and geographic differential in marginal CO2 emissions. Such results corroborate recent literature but add real world tools and practices for sustainable DevOps. Future work should look at running the system over longer periods to better capture seasonal trends, balance carbon savings with cost and latency, integrate with production orchestration systems, and directly measure emissions from live workloads.

QR Code for Recording