Integrating Ecovisor into Mosaik Co-Simulation

Marvin Steinke and Henrik Nickel
Technische Universität Berlin

***Abstract*—To reduce emissions, cloud platforms must increasingly rely on renewable energy sources such as solar and wind power. Nevertheless, the issue of volatility associated with these sources presents a significant challenge, since current energy systems conceal such unreliability in hardware. Souza et al. have devised a solution to this issue by creating an "ecovisor". This system virtualizes the energy infrastructure and allows for software-defined control to be accessible by applications.**

I. Introduction

THE surge of cloud platforms has had a significant impact on many businesses and individuals, offering access to innovative and valuable applications that frequently require significant computational resources. A notable example for this is represented by OpenAI's chat generative pre-trained transformer (ChatGPT) cloud platform, which is based on the GPT-3 model developed by Brown et al. [1] and achieved more than a million users within the first five days of its launch [2]. The increasing demand for more computational power has lead cloud platforms to become an essential part of the digital landscape [3]. These platforms allow for the storage, processing and management of large amounts of data and computational resources, which can be used to run applications and services that are beyond the capabilities of traditional hardware systems.

However, while the growth of cloud platforms has brought many benefits, it has also raised concerns about their environmental impact. As the demand for computational resources increases, so does the carbon emissions generated by the energy consumption of these platforms [4]. Despite their ecological impact, the growth of cloud platforms shows no signs of slowing down. According to Gartner Inc. worldwide end-user spending on public cloud services is forecasted to grow 20.7% to total $591.8 billion in 2023 [5]. To mitigate their impact on the environment, cloud platforms are now looking for ways to reduce their carbon footprint [6, 7]. It is imperative to adopt cleaner energy sources for powering data centers, both in the cloud and at the edge.

Although clean energy offers numerous advantages, it is perceived as being unreliable due to two key factors. One, the generation of renewable energy sources such as solar and wind is affected by environmental changes, and two, the carbon-intensity of grid power fluctuates as the grid uses different types of generators with varying carbon emissions to meet demand [8]. The field of computing possesses a distinct advantage in terms of reducing its carbon impact through the utilization of cleaner energy sources [9]. However, current cloud applications are unable to apply these benefits to optimize their carbon efficiency. This is because the energy system obscures the instability of clean energy with a reliability abstraction, which gives no control or visibility into the energy supply. As a result, applications cannot adjust their power usage in response to changes in the availability and carbon-intensity of renewable energy [10].

Souza et al. [10] proposed a solution to this issue by creating an "ecovisor", which virtualises the energy system and provides software-based control over it. This ecovisor enables each application to manage the instability of clean energy through software, customized to meet its unique requirements. However, even their small-scale prototype is intricately designed and incorporates several expensive components, such as a DC power supply equipped with a solar array simulation capability that costs almost $10,000.

Though the creation of the ecovisor is a promising solution to the issue of reducing the carbon footprint of cloud platforms, a large scale prototype for research and development purposes would not only be significantly cost intensive, but also time consuming. An alternative approach to consider is to simulate *only* the ecovisor infrastructure with real applications as a Software-In-The-Loop (SIL) implementation. This would allow for the optimization of energy usage in response to changes in the availability and carbon-intensity of renewable energy, without the need for a physical implementation. By simulating the ecovisor, applications can still manage the instability of clean energy through software, but at a lower cost and with less time investment.

To this end, we propose the utilization of a co-simulation tool, Mosaik, to integrate this ecovisor simulation into, and evaluate its impact on the carbon footprint of cloud platforms. Mosaik is a simulation framework for power systems, communication networks, and building automation, which can be used to model and analyze the performance of the ecovisor in real-world applications [11].

In the following sections of this article, we will commence by presenting the necessary contextual information in Section II to ensure comprehension of our methodology. This includes co-simulation environments with Software- (SIL) and Hardware-In-The-Loop (HIL) strategies, the Mosaik co-simulation tool and a closer examination of the ecovisor infrastructure and its interface to applications. We will then review some related literature with similar approaches in Section III. Subsequently, we present our approach of integrating the ecovisor into Mosaik in Section IV. The approach is divided into the simulation of the ecovisor itself and the interface to external applications, beyond the scope of the co-simulation. We evaluate our approach in Section V with exemplary data, representing a realistic scenario that covers all edge cases. After discussing future research directions related to this article in Section VI, we conclude this article in Section VII

by summarizing the main points and contributions.

II. Background

A. Co-Simulations and SIL / HIL

Co-simulations are used to model and analyze complex systems with multiple interacting components, each of which may have different properties and behaviors. They involve combining simulation models from different domains, such as control, power, and thermal management, to create a unified model that accurately represents the behavior of the overall system.

One of the main advantages of co-simulations over regular simulations is their ability to capture the interactions between different components of the system. Regular simulations often make simplifying assumptions that can lead to inaccurate results. Co-simulations, on the other hand, can account for the interactions between components and provide a more accurate representation of the system's behavior. This makes co-simulations particularly useful for designing and optimizing complex systems like datacenters. The virtual environment can save time and resources, reduce the risk of failure, and lead to more efficient and sustainable datacenters [12].

Co-simulations can be further enhanced by incorporating the SIL and HIL methodology. This approach involves integrating real-world components, such as software and hardware, into the simulation environment to better reflect the actual system behavior. The integration of real components allows for a more accurate representation of the system's behavior and can also identify potential issues that may arise in real-world scenarios [13].

B. Mosaik

Mosaik is an open-source co-simulation tool that allows for the integration of different simulation models from various domains into a unified simulation environment. Mosaik provides a Python-based framework for developing and executing co-simulations, enabling the creation of complex simulations with interacting components. It supports the development of co-simulation scenarios by providing an API for defining simulation models, connecting them to form a simulation network, and specifying simulation scenarios. The tool also provides various visualization tools and data analysis capabilities to analyze the results of the simulation. Furthermore, Mosaik provides a library of pre-existing simulation models that can be used to build custom simulations.

C. Ecovisor [1]

Figure 1 shows an overview of the ecovisor's design which manages resources and energy using containers or virtual machines as the basic unit. An instance-level API is chosen to align with existing cloud APIs and to support higher-level cluster or cloud-level APIs. The ecovisor extends an existing orchestration platform that provides basic container

[1]The information presented in this section is a summary of Section 3 and 3.1 from Souza et al.'s work [10], with some modifications made for clarity and conciseness.

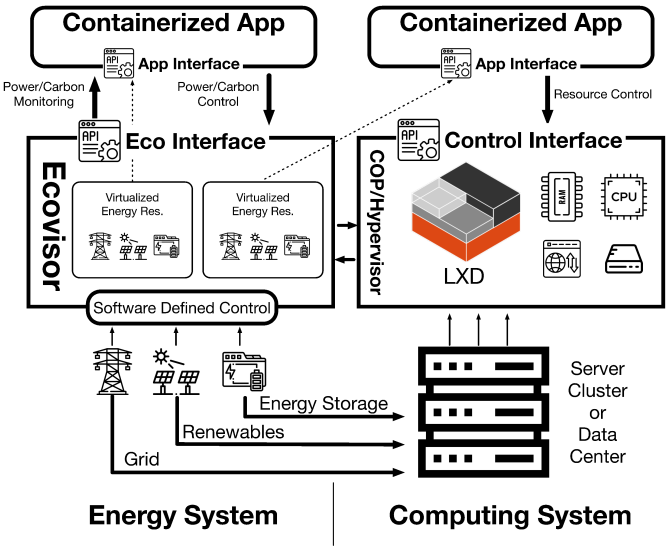

Figure 1. Ecovisor Design (Souza et al.) [10]

or VM management and monitoring functions. Container Orchestration Platforms (COPs) are used to manage resources and applications. COPs provide virtual clusters composed of multiple containers with specified resource allocations that can grow or shrink over time. COPs include a scheduling policy that determines resource allocation under constraints, and COPs are resilient to resource revocations. This resiliency is useful for designing carbon-efficient applications as low-carbon energy may cause power shortages that also manifest as resource revocations.

The virtual energy system includes a virtual grid connection, a virtual battery, and a virtual solar array. The system provides getters and setters methods for monitoring and controlling the virtual power supply and demand, including per-container power caps, battery charging, and discharging rates, as shown in Table I. The system uses virtual solar power first to meet demand and charges the virtual battery with any excess solar power. When there is not enough solar power, the virtual energy system uses grid power to make up the difference, while accounting for carbon emissions and power usage over discrete time intervals. The ecovisor system provides a uniform centralized interface for accessing energy-related information and historical data.

III. Related Work

In research and development processes, the utilization of physical testbeds is a widespread practice [14, 15]. They serve as a medium for testing and evaluating new technologies, systems, and products prior to their market launch or further development. However, some scenarios and conditions may not be feasible or safe to recreate in a physical testbed. Simulations can provide a controlled and cost-effective environment for testing such scenarios and conditions [16]. When specific physical or software components require a certain degree of realism though, simulations or co-simulations with SIL or HIL methodologies can effectively address these limitations.

Table I
ECOVISOR'S API (SOUZA ET AL.) [10]

Function Name	Type	Input	Return Value	Description
`set_container_powercap()`	Setter	ContainerID, kW	N/A	Set a container's power cap
`set_battery_charge_rate()`	Setter	kW	N/A	Set battery charge rate until full
`set_battery_max_discharge()`	Setter	kW	N/A	Set max battery discharge rate
`get_solar_power()`	Getter	N/A	kW	Get virtual solar power output
`get_grid_power()`	Getter	N/A	kW	Get virtual grid power usage
`get_grid_carbon()`	Getter	N/A	$g \cdot CO_2/kW$	Get virtual grid power usage
`get_battery_discharge_rate()`	Getter	N/A	kW	Get current rate of battery discharge
`get_battery_charge_level()`	Getter	N/A	kWh	Get energy stored in virtual battery
`get_container_powercap()`	Getter	ContainerID	kW	Get a container's power cap
`get_container_power()`	Getter	ContainerID	kW	Get a container's power usage
`tick()`	Notification	N/A	N/A	Invoked by ecovisor every Δt

For instance, Beilharz et al. [17], introduce Marvis, a framework that provides a comprehensive staging environment for testing distributed IoT applications. Marvis orchestrates hybrid testbeds, co-simulated domain environments, and a central network simulation to create a representative operating environment for the system. However, Marvis does not provide a virtualized energy system, which is crucial to meet the requirements of our problem statement.

Hagenmeyer et al. [18] investigate the interplay of different forms of energy on various value chains in Energy Lab 2.0. The focus is on finding novel concepts to stabilize the volatile energy supply of renewables through the use of storage systems and the application of information and communication technology tools and algorithms. The smart energy system simulation and control center is a key element of Energy Lab 2.0 and consists of three parts: a power-hardware-in-the-loop experimental field, an energy grid simulation and analysis laboratory, and a control, monitoring, and visualization center. While this smart energy system simulation is similar to the simulated ecovisor in our approach, the control center is the only entity with software-based control over this energy system. The ecovisor infrastructure, however, provides multiple applications with the ability to manage their energy supply themselves which is an essential factor for our approach.

In conclusion, to the best of our knowledge, there is no current approach that can simulate an energy system virtualizer with software-based control, comparable to the ecovisor, with SIL capabilities.

IV. APPROACH

A. Simulation of the Ecovisor

B. Interface to External Applications

ToDo: Second itteration on writing

For connecting the Ecovisor-Model to a real workload, we had to expose the API described in [10]. We have tried to implement it as it would probably be implemented in a real cloud environment. To achive that, we implemented the API of the Ecoviser with a RedisDB as value store between the ecovisor and the api-server. This also ensures atomiticity on the writes of the values.

The API integration itself cosists of three parts, the API-Server, the RedisDB and the Ecovisor-Redis-Interface

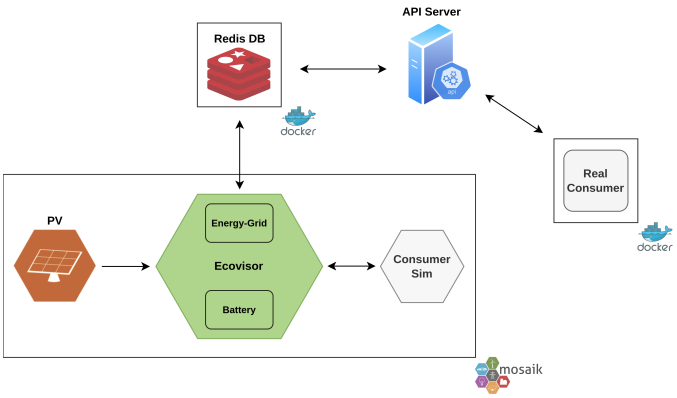

Figure 2. System Design

1) API-Server: The API Server exposes the Ecoviser-APi to the "Workloads". It is implemented with the FastAPI Framework and Utilizes Uviorn to handel multiple clients accessing the API. The Module is started as aan independent thread. In earlyer versions we tried to implement it inside of the ecovisor model, but due to the way uvicorn starts the API-Server, it would stops the execution of the simulation until the API-Server is stopped and so we decided to design it as a standalone module. This als enables the API-Server to be scaled indipendetn from the Ecovisor-Model and the RedisDB. This may be helpful in bigger simulations with distributed infrastructur.

2) RedisDB: The RedisDB is used to interchange the values provided by eiterh ecovisor or the workload application. The database itself is a fast in memory key value store, wich is started as a docker container via the docker python libary to keep manual managment of the simulation to a minimum.

In general the Database can be exchanged with any other database by adapting the Ecovisor-Redis-Interface in the Ecovisor model. This can be useful when simulation should be integrated in a production grade cloud environment like kubernetes or open stack.

3) Ecovisor-Redis-Interface:

4) Dataflow:

V. Evaluation

VI. Future Work

While our approach focused on a single simulation of the ecovisor, our use of Mosaik demonstrates its effectiveness for large-scale smart grid simulations. In future research, we suggest interconnecting multiple ecovisor systems to share resources and further reduce carbon emissions. This network could be distributed across different geographic regions, as carbon intensity varies depending on location. By incorporating carbon information services like Electricity Maps [19], this could enable carbon-efficiency optimizations such as Let's Wait Awhile [20] or Cucumber [21] from Wiesner et al. This would enhance the potential for carbon reduction at a larger scale.

VII. Conclusion

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[2] G. Brockman, "Chatgpt just crossed 1 million users," Twitter, dec 2022, accessed: 16.02.2023. [Online]. Available: https://twitter.com/gdb/status/1599683104142430208?s=20&t=-7V8elwpZ93qiIhwxnzVBA

[3] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, pp. 7–18, 2010.

[4] M. A. B. Siddik, A. Shehabi, and L. Marston, "The environmental footprint of data centers in the united states," *Environmental Research Letters*, vol. 16, no. 6, p. 064017, 2021.

[5] Gartner Inc., "Gartner forecasts worldwide public cloud end-user spending to reach nearly $600 billion in 2023," oct 2022, accessed: 16.02.2023. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2022-10-31-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023

[6] Google LLC, "Building a carbon-free future for all," accessed: 16.02.2023. [Online]. Available: https://sustainability.google/commitments/carbon/

[7] Amazon.com, Inc., "Sustainability in the cloud," accessed: 16.02.2023. [Online]. Available: https://sustainability.aboutamazon.com/environment/the-cloud

[8] N. Scarlat, M. Prussi, and M. Padella, "Quantification of the carbon intensity of electricity produced and used in europe," *Applied Energy*, vol. 305, p. 117901, 2022.

[9] S. Murugesan, "Harnessing green it: Principles and practices," *IT Professional*, vol. 10, no. 1, pp. 24–33, 2008.

[10] A. Souza, N. Bashir, J. Murillo, W. Hanafy, Q. Liang, D. Irwin, and P. Shenoy, "Ecovisor: A virtual energy system for carbon-efficient applications," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 2023, pp. 252–265.

[11] C. Steinbrink, M. Blank-Babazadeh, A. El-Ama, S. Holly, B. Lüers, M. Nebel-Wenner, R. P. Ramírez Acosta, T. Raub, J. S. Schwarz, S. Stark, A. Nieße, and S. Lehnhoff, "Cpes testing with mosaik: Co-simulation planning, execution and analysis," *Applied Sciences*, vol. 9, no. 5, 2019.

[12] M. Vogt, F. Marten, and M. Braun, "A survey and statistical analysis of smart grid co-simulations," *Applied Energy*, vol. 222, pp. 67–78, 2018.

[13] T. Kelemenová, M. Kelemen, L. Miková, V. Maxim, E. Prada, T. Lipták, and F. Menda, "Model based design and hil simulations," *American Journal of Mechanical Engineering*, vol. 1, no. 7, pp. 276–281, 2013.

[14] M. H. Cintuglu, O. A. Mohammed, K. Akkaya, and A. S. Uluagac, "A survey on smart grid cyber-physical system testbeds," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 446–464, 2017.

[15] J. Mambretti, J. Chen, and F. Yeh, "Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn)," in *2015 International Conference on Cloud Computing Research and Innovation (ICCCRI)*, 2015, pp. 73–79.

[16] N. Mansouri, R. Ghafari, and B. M. H. Zade, "Cloud computing simulators: A comprehensive review," *Simulation Modelling Practice and Theory*, vol. 104, p. 102144, 2020.

[17] J. Beilharz, P. Wiesner, A. Boockmeyer, F. Brokhausen, I. Behnke, R. Schmid, L. Pirl, and L. Thamsen, "Towards a staging environment for the internet of things," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2021, pp. 312–315.

[18] V. Hagenmeyer, H. Kemal Çakmak, C. Düpmeier, T. Faulwasser, J. Isele, H. B. Keller, P. Kohlhepp, U. Kühnapfel, U. Stucky, S. Waczowicz, and R. Mikut, "Information and communication technology in energy lab 2.0: Smart energies system simulation and control center with an open-street-map-based power flow simulation example," *Energy Technology*, vol. 4, no. 1, pp. 145–162, 2016.

[19] Electricity Maps A/S , "Reduce carbon emissions with actionable electricity data," accessed: 16.02.2023. [Online]. Available: https://www.electricitymaps.com/

[20] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen, "Let's wait awhile," in *Proceedings of the 22nd International Middleware Conference*. ACM, dec 2021.

[21] P. Wiesner, D. Scheinert, T. Wittkopp, L. Thamsen, and O. Kao, "Cucumber: Renewable-aware admission control for delay-tolerant cloud and edge workloads," in *Euro-Par 2022: Parallel Processing*. Springer International Publishing, 2022, pp. 218–232.