Vessim: A Testbed for Carbon-Aware Applications and Systems

Philipp Wiesner, Ilja Behnke, Paul Kilian, Marvin Steinke, Odej Kao

Technische Universität Berlin

Berlin, Germany

Abstract

To reduce the carbon footprint of computing and stabilize electricity grids, there is an increasing focus on approaches that align the power usage of IT infrastructure with the availability of clean energy. Unfortunately, research on energy-aware and carbon-aware applications, as well as the interfaces between computing and energy systems, remains complex due to the scarcity of available testing environments. To this day, almost all new approaches are evaluated on custom simulation testbeds, which leads to repeated development efforts and limited comparability of results.

In this paper, we present Vessim, a co-simulation environment for testing applications and computing systems that interact with their energy systems. Our testbed connects domain-specific simulators for renewable power generation and energy storage, and enables users to implement interfaces to integrate real systems through software and hardware-in-the-loop simulation. Vessim offers an easy-to-use interface, is extendable to new simulators, and provides direct access to historical datasets. We aim to not only accelerate research in carbon-aware computing but also facilitate development and operation, as in continuous testing or digital twins. Vessim is publicly available: https://github.com/dos-group/vessim.

CCS Concepts

• **Computing methodologies → Simulation environments**; • **Social and professional topics → *Sustainability***.

Keywords

carbon-aware computing, co-simulation, software-in-the-loop, microgrid, smart grid, continuous testing, digital twin

1 Introduction

The rapid adoption of compute-intensive technologies like machine learning and big data processing has accelerated the growing demand for computing power in recent years [16, 81]. While the capacity of hyperscale datacenters doubled within only four years [26], also their global energy usage has increased by 10 – 60 % between 2015 and 2021 [32]. This shows that improvements in hardware and software efficiency are still able to mitigate significant portions of
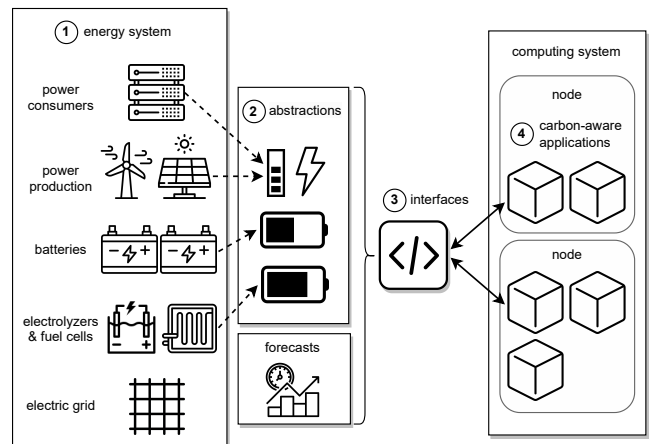
Figure 1: Vessim enables research on various aspects regarding interactions between computing and energy systems.

growth in computing demand [47]. However, further efficiency improvements are becoming increasingly challenging, suggesting that future growth will likely translate into increasing energy demand more directly [7, 32, 47].

To decrease the carbon footprint of computing despite increasing energy demand, a new paradigm called *carbon-aware computing* has received much attention in academia [4, 28, 61, 77] and industry [13, 31, 38, 51]. Carbon-aware computing aims at aligning the power usage of compute infrastructure with the availability of variable renewable energy sources like solar and wind, for instance, by deferring or migrating flexible workloads. To perform carbon-aware decisions, we need up-to-date information on the datacenter's energy system – which increasingly are microgrids comprising renewable power generation and energy storage. However, today's energy systems are designed to hide underlying complexities [7, 68], so relevant metrics are often not available to applications. Examples include the real-time volume of on-site power generation, battery state of charge, or the current carbon intensity (gCO$_2$/kWh) of the public grid. To enable researchers and developers to make their approaches carbon-aware, we need to improve the interplay between future energy and computing systems and create novel interfaces for information exchange. For example, Souza et al. proposed *ecovisors* to virtualize the energy system and provide software-defined access to applications [68].

Nonetheless, research and development of applications that interact with energy systems, as well as the design of abstraction layers and interfaces, remains challenging due to the scarcity of available test environments. Carbon-aware approaches often involve globally distributed scenarios, where evaluations must be carried out over multiple days or months. Only large industrial players can afford

to test their approaches on real energy and computing infrastructures [61] and due to the complexity and cost of constructing hardware testbeds, as well as their limited applicability for distributed settings, only a few approaches are evaluated on such [43, 68]. As a result, the majority of existing works simulate at least parts of the energy system [2, 10, 18, 22–25, 28, 29, 37, 42, 45, 46, 73, 77, 78, 83], for example, by replaying carbon intensity or solar production traces. This leads to repeated software development efforts by researchers and little to no comparability between approaches. Although first approaches use model-based analysis for planning carbon-aware datacenters [1], there exist no solutions for actually testing their runtime behavior.

To effectively develop, operate, evaluate, and test carbon-aware approaches, we see the need for a modular and extensible testing environment. In this paper, we propose Vessim, a co-simulation testbed that combines domain-specific simulators for the different aspects of energy systems and enables the integration of real applications and hardware. Our contributions are:

- we propose a co-simulation design for connecting different simulators for power generation and energy storage to realistically represent microgrids for computing systems
- we explain how real computing systems can be connected to Vessim through software/hardware-in-the-loop simulation
- we demonstrate the usability of our testbed in an example scenario

Our goal is to facilitate and accelerate the development of carbon-aware applications and systems by enabling rapid prototyping and reproducible testing.

2 Case for a Co-Simulation Testbed

The testing of computer systems in simulated or emulated environments is commonly encountered where software depends on the behavior of physical systems, such as in the Internet of Things (IoT), edge computing, or cyber-physical systems [8, 9, 14, 30, 64, 72, 75]. The modular nature of co-simulation allows the usage of established domain-specific simulators and has led to open standards like the Functional Mock-up Interface (FMI) [5] and various co-simulation frameworks [57, 64, 70].

Research opportunities. Using a co-simulation testbed enables research on various aspects related to the interplay of energy and computing systems, as shown in Figure 1:

① **Energy system composition.** A co-simulation testbed facilitates rapid and cost-effective experimentation to determine optimal configurations for renewable power, batteries, or hydrogen storage [58] in datacenters. It enables parallel assessment of various compositions and exploration of otherwise expensive or inaccessible technologies.

② **Energy system abstractions.** Microgrids involve diverse producers, consumers, and storage types. While many complexities are irrelevant to carbon-aware applications, some are safety-critical. Configurable testbeds are valuable for experimenting with new abstractions, such as virtualizing the energy system [7].

③ **Energy system interfaces.** Our testbed enables research on new interfaces covering (i) the integration of new components and abstractions, (ii) the handling of external data like weather forecasts, (iii) the coordination of geo-distributed computing and energy systems, and (iv) security concerns when exposing the energy system.

④ **Carbon-aware applications.** Reproducible testbeds allow developers to prototype carbon-aware approaches quickly. In particular, we aim at providing access to current standards and best practices [27] and encourage the use of common datasets and baselines for evaluations.

Opportunities in development and operation. Besides facilitating research, Vessim can support the development and quality assurance of carbon-aware applications and systems. For example, it can be applied in continuous integration testing, or used to validate software roll-outs in a controlled environment.

Vessim can furthermore enhance the operation of carbon-aware datacenters by serving as a digital twin to predict future system states and help with decision-making. This allows the integration of autonomous control strategies based on game theory [73] or reinforcement learning [44, 82], which have been proposed in recent years. Moreover, digital twins can be used for risk assessment in case of extreme events such as power outages.

3 Vessim Overview

Vessim is based on the co-simulation framework Mosaik [70], which was originally developed for smart grid scenarios but has since its initial release in 2011 evolved into a comprehensive general-purpose co-simulator. Mosaik offers APIs for connecting continuous and discrete-event simulators from any programming language as well as real systems through a TCP interface. It furthermore enables users to specify which of these subsystems can interact during simulation runs. At runtime, Mosaik coordinates the synchronization of subsystems and message passing.

Vessim provides a high-level Python API (§6) on top of Mosaik, which is designed to facilitate common use cases in the testing of carbon-aware systems. In particular, users can define an arbitrary number of *Microgrids* that each represents the energy system of a (distributed) computing system. Vessim specifies a set of possible co-simulation components per microgrid (*actors*, the *grid*, *controllers*, and *energy storage*) which are all modeled by Mosaik subsystems and interact with each other as illustrated in Figure 2. The adaptable architecture of Mosaik facilitates easy extension of our framework.

Besides supporting the design of test scenarios through opinionated interfaces, Vessim offers a wide range of integrated simulators and ready-to-use datasets which are outlined in §4. Furthermore, Vessim enables software/hardware-in-the-loop simulation to integrate real applications and computing systems. This capability is summarized in §5 and described in more detail in [79].

The following sections explain the different types of co-simulation components that exist in Vessim. The underlying Mosaik subsystems are executed in the following order: 1. *actors*, 2. *grid*, 3. *controllers*, and 4. *energy storage*.
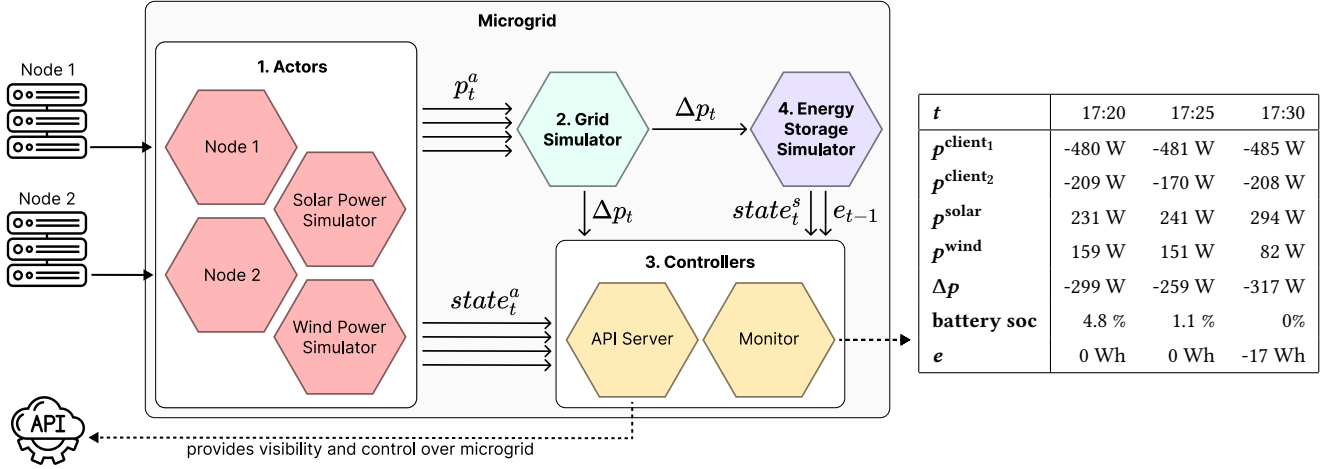
Figure 2: Vessim allows the simulation of multiple Microgrids in parallel where each consists of the depicted components. Hexagons represent a co-simulation subsystem managed by Mosaik [70]. *Actors* represent consumers/producers; the *grid* simulator determines the current excess/deficit in power; *controllers* can be used for monitoring, decision-making, or providing APIs to computing systems; *energy storage* simulators can charge/discharge batteries depending on user-defined policies.

3.1 Actors

Actors represent power producers and consumers within a microgrid. At each simulation step t, each actor $a \in A$ relays its current power production ($p_t^a > 0$) or consumption ($p_t^a < 0$) to the grid simulator and, optionally, additional $state_t^a$ information to the controllers. Examples of actors range from real servers equipped with power meters to entirely simulated systems, e.g. replaying previously recorded traces. We provide an overview of already integrated subsystems in §4 and explain software and hardware-in-the-loop capabilities to integrate physical or virtual nodes in §5.

3.2 Grid

Vessim's default grid simulator aggregates the power production or consumption of all actors to determine the grid's current excess or deficit, i.e. $\Delta p_t = \sum_{a \in A} p_t^a$.

While this representation of power flows is accurate enough for most use cases, some scenarios require more precise modeling. There are various tools for power system analysis like pandapower [74] or PyPSA [11] that are already integrated in Mosaik and can be used to periodically solve optimal power flows between actors. Such modeling is useful when, for example, considering transmission loss over long distances or when testing approaches that optimize for curtailed energy [78, 84], which is usually caused by congested grids. Considering the expected impacts of growing datacenter power demands on the reliability of grids [41], power system analysis can also be valuable to investigate approaches that incorporate grid planning and management.

3.3 Controllers

Users can define a number of controllers to monitor the state of a microgrid and perform actions based on this information. At every step t, each controller $c \in C$ receives all currently available information within the microgrid: $state_t^a \ \forall a \in A$, the grid's Δp_t,

the energy storage's $state_t^s$, and the energy surplus/deficit since the last step e_{t-1} (see §3.4). Additionally, users can define custom input signals to be used for decision-making. For instance, the carbon intensity of energy from the public grid is a popular metric for scheduling carbon-aware applications.

Based on this information, controllers can persist the state of the microgrids over time, visualize this information in graphical user interfaces, or perform actions like throttling compute nodes or configuring the battery energy storage system. A prominent use case of controllers is to provide API access to real computing systems to enable visibility and control over the microgrid §5.2. An example scenario involving a custom controller can be found in §6.

3.4 Energy Storage

While the above-mentioned components operate sequentially in the co-simulation, they all execute at the same step t, effectively stepping "simultaneously." In contrast, the energy storage simulator manages the charging and discharging of energy storage according to a user-defined policy, thereby modeling the progression of time between simulation steps.

The energy storage simulator receives Δp_t along with the duration d that passes between $t-1$ and t. A charge policy defines how Δp_t is to be settled with the energy storage. Any energy deficit/surplus that has not been settled between $t-1$ and t will be returned as e_{t-1}, together with the new battery state $state_t^s$. In other words, e_{t-1} represents the energy either drawn from ($e_{t-1} < 0$) or fed into ($e_{t-1} > 0$) the public grid. If no storage is present, this simulator simply returns $e_{t-1} = \Delta p_{t-1} \cdot d$.

For instance, if $\Delta p_{t-1} = -300W$ and the previous step's duration is $d = 300s$, the default storage policy will try to discharge $300W \cdot 300s = 25Wh$. In case only $10Wh$ are left in the storage, the simulator would return $e_{t-1} = -15Wh$. Users have the flexibility to configure the internal step size of the energy storage simulator and the level of realism in storage modeling, as detailed in §4.2.

4 Simulators and Datasets

This section describes which simulators and datasets are already integrated in Vessim, as well as possibilities for future extensions.

4.1 Renewable Power Production

Vessim provides utilities for replaying historical weather and energy production data, as well as historical forecasts. For example, we include two ready-to-use solar production datasets including forecasts provided by Solcast [67]—one dataset covering ten cities in Germany with high geographic proximity and another dataset covering ten globally distributed cities. If users require data from specific times and locations, they can utilize weather data services [56, 66, 67] that provide estimates of historical solar irradiance data (surface power density in W/m^2) worldwide. This data can be multiplied by the panel area and efficiency factor (usually 15-20 %) and optionally factors like incidence angle, temperature, shading, soiling, or snow coverage [52].

For wind energy, we must consider factors like turbine type and efficiency, rotor diameter, blade pitch, and hub height. For this, Vessim integrates the System Advisor Model (SAM) by the National Renewable Energy Laboratory of the US Department of Energy [21] which can simulate single turbines as well as wind farms.

Next steps. Besides solar and wind, the SAM provides simulators for various alternative renewable energy sources like concentrating solar power [50], marine energy [39], biomass combustion [35], or geothermal power [53]. Furthermore, the hardware-in-the-loop interfaces deployed for feeding the power usage of actual servers to a simulation could also be used to inform the simulation about the power supply of real producers.

4.2 Energy Storage

Vessim supports modeling various types of energy storage with vastly different levels of realism. Alongside a simple battery model that does not consider any conversion losses, Vessim includes the C/L/C model for lithium-ion batteries [36]. This model accounts for efficiency and imposes linear charging/discharging rate limits and has previously been used in related work [1]. For even more realistic battery modeling, we also provide an interface to the physics-based battery simulator PyBaMM [71], which features an extensive range of battery models and simulation solvers.

Next steps. We are currently assessing how different battery simulators affect experiment quality, runtime, and resource usage. Additionally, Vessim could be extended by more diverse types of batteries. For instance, the SAM [17] also includes simulators for lead-acid, lithium-ion, vanadium redox flow, and all iron flow batteries [54] as well as models for battery degradation over time [65, 69]. Matlab/Simulink [48] supports modeling of short-term energy storage options such as flywheels or supercapacitors [49], commonly used in uninterruptible power supplies.

Lastly, the idea to further reduce the reliance of datacenters on grid electricity through hydrogen storage, i.e. by providing them with fuel cells or electrolyzers, is prominent across industry [62] and research [40, 55, 58, 63]. Simulators for fuel cells can be found in the SAM and Matlab/Simulink.

4.3 Historical Datasets and Live Data

Next to the two solar datasets explained in §4.1, Vessim includes an exemplary dataset containing historical data and historical forecasts of the Marginal Operating Emissions Rate (MOER) [12]—a metric for describing the carbon intensity of grid energy—provided by WattTime [76]. Data providers like Solcast, WattTime, or Electricity Maps [19] offer historical datasets for many regions around the globe which can directly be loaded within Vessim. Furthermore, Vessim provides utilities for (i) returning perfect forecasts (i.e. returning the actual future data points), (ii) replaying historical forecasts, and (iii) including predictive models which generate forecasts during simulation.

Next steps. Besides traces for weather, renewable power generation, or carbon intensity, there also exist various public datasets with cluster workload traces, such as from Alibaba [3], Azure [6], or Google [80]. When provided with appropriate power models that map infrastructure utilization to expected power demand, such traces can be replayed during co-simulations to model the power usage of large-scale computing systems.

Furthermore, we want to implement live data connectors to popular data providers or weather stations such as [59].

5 Software & Hardware-in-the-Loop

This section outlines how Vessim enables the integration of real computing systems into an energy system co-simulation, which is explained in more detail in [79]. The integration involves two steps: First, reflecting the computing system's power demand in the simulation. Second, enabling the computing system to exert visibility and control over the energy system.

It is important to note that *as soon as real software or hardware is part of the simulation, we have to process the co-simulation in real-time* which is already supported by Mosaik [70]. However, for real-time simulations, we must ensure that all computations within a simulated time step take less time than the elapsed real time. Complex subsystems that are prone to exceed this time limit should therefore be equipped with sufficiently powerful—or even multiple—machines. On the other hand, if *all* components of a Vessim experiment are simulated, the speed of the co-simulation is mainly determined by the complexity of its subsystems and the performance of the host system. Users can assess months or years of simulated time within minutes of real-time, which can be useful to, for example, investigate the impact of battery aging.

5.1 Computing System Power Demand

Vessim enables the integration of physical or emulated hardware through special *actors* which periodically query and relay the real or emulated power demand.

Physical nodes. For physical nodes, there exist different ways of estimating the power usage of hardware, like Intel's *Running Average Power Limit (RAPL)* technology. However, if precise power demand measurement is required, it is recommended to resort to external energy meters as most devices cannot accurately measure

their power consumption using only software. Our current prototype queries power demand directly from a metered power distribution unit, but we are already working on actors that retrieve relevant information from monitoring systems like Prometheus [20].

Virtual nodes. For virtual nodes, like VMs or containers, users need to define one or more metrics that are considered significant indicators of the node's power consumption (like CPU usage). Using power models, these metrics can then be translated into the virtual node's power usage. Power models can be based on mathematical models [33] or benchmarks such as SPECpower [15]. As with physical nodes, this data can usually be queried from existing monitoring systems.

Overhead. Power demand for cooling, uninterruptible power supply, or lighting is traditionally reflected in a datacenters power usage effectiveness (PUE) [34] which can be incorporated into the corresponding actor. For more precise modeling of overhead, users can resort to advanced power models [60] or, even better, measure the power demand of their real computing system beforehand under different conditions.

5.2 Enabling Visibility and Control

As illustrated in Figure 2, Vessim comes with a *controller* that serves a REST API, similar to the one proposed in ecovisor [68] which provides applications with visibility (e.g. the current carbon intensity and solar production) and control (e.g. setting battery charge rates or throttling the power usage of applications) over the simulation. The exposed information and API design are fully configurable by users, see points ② and ③ in Figure 1.

When applications interface with a real-time simulation, there is a risk that the simulation will become overloaded in certain situations and fall behind schedule (see the introductory comment in §5). For example, a battery simulation may exceed its maximum processing time per simulation step if a large number of applications send requests to update their virtual battery configuration. To avoid this, Vessim's API server comprises a buffer that collects and aggregates SET requests to forward them to the simulation at predefined intervals. Likewise, the server includes a cache containing the relevant state of the energy system, ensuring that GET requests do not disrupt the simulation. See [79] for more details.

6 Example Scenario

We demonstrate the current capabilities and usability of Vessim through an example scenario set in an edge computing environment that involves software-in-the-loop simulation. Listing 1 shows how this scenario is implemented using Vessim's Python API.

Actors. After creating an vs.Environment, we add a single microgrid to the simulation. This microgrid contains two *actors*: The first actor contains a vs.HttpPowerMeter that periodically queries the power usage of a single physical node (NVIDIA Jetson TX2) which is connected to a metered power distribution unit. The Jetson is running a stress test at full load during the entire experiment.

The second actor represents a small solar panel with 50 W maximum output. Production is modeled using a vs.HistoricalSignal that represents the "San Francisco" location in the "solcast2022_global" dataset that is included in Vessim.

```python
import vessim as vs


class CustomController(vs.Controller):
    ...


environment = vs.Environment()
environment.add_microgrid(
  actors=[
    vs.Actor("jetson", signal=vs.HttpPowerMeter(
      "https://127.0.0.2:8080/power"),
    vs.Actor("solar", signal=vs.HistoricalSignal.load(
      "solcast2022_global", column="San Francisco",
      params={"scale": 50}))
  ],
  storage=vs.SimpleBattery(capacity=250, initial_soc=0.6,
    min_soc=0.5),
  controllers=[
    vs.Monitor(outfile="result.csv"),
    vs.CustomController(moar=vs.HistoricalSignal.load(
      "watttime2022"), column="CAISO_NORTH")
  ],
  step_size=60, # 1 min
)
environment.run(until=24 * 3600)  # 24 h
```

Listing 1: Scenario definition with Vessim's Python API. The scenario connects a solar and battery simulation with a physical, power-metered device (NVIDIA Jetson TX2).

Grid and energy storage. Our example uses Vessim's default grid simulator that simply aggregates the current production/consumption from all consumers. The microgrid is equipped with a vs.SimpleBattery with 250 Wh capacity that has an initial state of charge (SoC) of 60 %. The configured charge policy allows a minimum SoC of 50 %.

Controllers. We instantiate two controllers: First, we configure a vs.Monitor to record all simulation state in a CSV file. The results are visualized in Figure 3.

Second, we implement a custom controller which has access to another vs.HistoricalSignal, representing the MOER in the CAISO_NORTH region where San Francisco is located. Due to space limitations, we do not provide the controller's code in this paper, but describe its actions and their effect on the simulation:

(1) Once the battery reaches its minimum SoC, the controller queries renewable power production forecasts (historical forecasts are part of the "solcast2022_global" dataset that the actor replays). Due to promising predictions, it temporarily lowers the minimum SoC to 30 % to avoid consuming carbon-intensive grid energy (MOER above 400 gCO2/kWh).

(2) Once 30 % SoC is reached, the microgrid draws energy from the now comparably clean grid (MOER below 60 gCO2/kWh). As evening approaches, the controller adjusts the charge policy to maintain a constant charging rate of 40 W to refill the battery. This requires drawing more energy from the grid, as displayed in the 4th and 5th line of Figure 3.

(3) At 6pm, the microgrid transitions back to island mode, meaning that is does not draw energy from the public grid anymore. To ensure overnight operation, at 8pm, the controller activates power-saving mode on the Jetson, reducing its power demand from ~15 W to ~7.5 W.
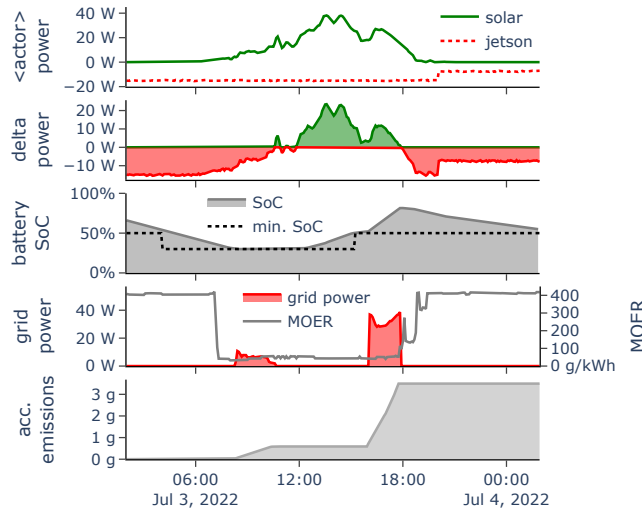

Figure 3: Results of the example scenario. 1. p_t^a of both actors, 2. Δp_t computed by the grid simulator, 3. the battery's SoC (part of $state_t^s$), 4. energy deficit ($-e_t$) & MOER, and 5. the accumulated carbon emissions ($-e_t \cdot$ MOER).

This scenario demonstrates Vessim's user-friendly programming interface and many of its currently implemented features. Our proposed testbed enables interactions between real computing systems and a fully simulated energy system.

Further examples. Our paper on software-in-the-loop simulation for carbon-aware applications [79] presents a similar, yet more detailed scenario based on Vessim, which includes physical and virtual nodes. FedZero [78] does not incorporate software-in-the-loop simulation but leverages Vessim to simulate multiple distributed energy systems in parallel, faster than real-time.

7 Conclusion

While research in sustainable computing is gaining momentum, many studies overlook real-world complexities due to the lack of testing environments. To fill this gap, we presented Vessim, a versatile co-simulation testbed for energy-aware and carbon-aware applications and systems, which connects energy domain-specific simulators with real software and hardware.

We invite researchers across the computing and energy disciplines to participate in the development of a common co-simulation testbed. Our goal is to accelerate progress in carbon-aware computing and research on datacenters' impact on power grids by continuously integrating the latest ideas from both fields.

In addition to the many opportunities for future work outlined in this paper, our primary focus will be on validating and calibrating Vessim simulations against real systems. Another valuable direction would be to extend Vessim to cover further aspects of energy systems beyond electricity, such as water and thermal management.

References

[1] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. 2023. Carbon Explorer: A Holistic Framework for Designing Carbon Aware Datacenters. In *ASPLOS*. https://doi.org/10.1145/3575693.3575754

[2] Anup Agarwal, Jinghan Sun, Shadi Noghabi, Srinivasan Iyengar, Anirudh Badam, Ranveer Chandra, Srinivasan Seshan, and Shivkumar Kalyanaraman. 2021. Redesigning Data Centers for Renewable Energy. In *ACM Workshop on Hot Topics in Networks (HotNet)*. https://doi.org/10.1145/3484266.3487394

[3] Alibaba. 2023. Alibaba Cluster Trace Program. Retrieved May 2023 from https://github.com/alibaba/clusterdata

[4] Thomas Anderson, Adam Belay, Mosharaf Chowdhury, Asaf Cidon, and Irene Zhang. 2022. Treehouse: A Case For Carbon-Aware Datacenter Software. In *Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon)*. https://doi.org/10.1145/3630614.3630626

[5] Modelica Association. 2022. Functional Mock-up Interface Specification 3.0. Retrieved May 2023 from https://fmi-standard.org/docs/3.0/

[6] Microsoft Azure. 2022. Microsoft Azure Traces. Retrieved May 2023 from https://github.com/Azure/AzurePublicDataset

[7] Noman Bashir, Tian Guo, Mohammad Hajiesmaili, David Irwin, Prashant Shenoy, Ramesh Sitaraman, Abel Souza, and Adam Wierman. 2021. Enabling Sustainable Clouds: The Case for Virtualizing the Energy System. In *ACM Symposium on Cloud Computing (SoCC)*. https://doi.org/10.1145/3472883.3487009

[8] Ilja Behnke, Lauritz Thamsen, and Odej Kao. 2019. HéCtor: A Framework for Testing IoT Applications Across Heterogeneous Edge and Cloud Testbeds. In *12th IEEE/ACM International Conference on Utility and Cloud Computing Companion*. https://doi.org/10.1145/3368235.3368832

[9] Jossekin Beilharz, Philipp Wiesner, Arne Boockmeyer, Florian Brokhausen, Ilja Behnke, Robert Schmid, Lukas Pirl, and Lauritz Thamsen. 2021. Towards a Staging Environment for the Internet of Things. In *PerCom Workshops*. https://doi.org/10.1109/PerComWorkshops51409.2021.9431087

[10] Nicolas Beldiceanu, Bárbara Dumas Feris, P. Gravey, Sabbir Hasan, Claude Jard, Thomas Ledoux, Yunbo Li, Didier Lime, Gilles Madi Wamba, Jean-Marc Menaud, Pascal Morel, Michel Morvan, Marie-Laure Moulinard, Anne-Cécile Orgerie, Jean-Louis Pazat, Olivier Roux, and Ammar Sharaiha. 2017. Towards energy-proportional Clouds partially powered by renewable energy. *Computing* 99 (2017). https://doi.org/10.1007/s00607-016-0503-z

[11] T. Brown, J. Hörsch, and D. Schlachtberger. 2018. PyPSA: Python for Power System Analysis. *Journal of Open Research Software* 6, 4 (2018). Issue 1. https://doi.org/10.5334/jors.188

[12] Duncan S. Callaway, Meredith Fowlie, and Gavin McCormick. 2018. Location, Location, Location: The Variable Value of Renewable Energy and Demand-Side Efficiency Resources. *Journal of the Association of Environmental and Resource Economists* 5, 1 (2018), 39–75. https://EconPapers.repec.org/RePEc:ucp:jaerec:doi:10.1086/694179

[13] Sylvain Cazard. 2022. Counting the Cost of Carbon: Why IT Efficiency Matters. VMWare. Retrieved May 2023 from https://news.vmware.com/esg/counting-carbon-it-efficiency-matters

[14] Mehmet Hazar Cintuglu, Osama A. Mohammed, Kemal Akkaya, and A. Selcuk Uluagac. 2017. A Survey on Smart Grid Cyber-Physical System Testbeds. *IEEE Communications Surveys & Tutorials* 19, 1 (2017).

[15] Standard Performance Evaluation Corporation. 2023. SPECpower benchmark. Retrieved May 2023 from https://www.spec.org

[16] Payal Dhar. 2020. The carbon impact of artificial intelligence. *Nature Machine Intelligence* 2 (2020).

[17] Nicholas DiOrio and Steven Janzou. 2019. *Fuel Cells in SAM: User Guide (DRAFT)*. Technical Report. National Renewable Energy Laboratory.

[18] Jesse Dodge, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A. Smith, Nicole DeCario, and Will Buchanan. 2022. Measuring the Carbon Intensity of AI in Cloud Instances. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. https://doi.org/10.1145/3531146.3533234

[19] Electricity Maps ApS. 2023. Reduce carbon emissions with actionable electricity data. Retrieved May 2023 from https://www.electricitymaps.com

[20] The Linux Foundation. 2024. Prometheus - Monitoring system & time series database. https://prometheus.io

[21] Janine M. Freeman, Jennie Jorgenson, Paul Gilman, and T. Ferguson. 2014. *Reference Manual for the System Advisor Model's Wind Power Performance Model*. Technical Report. National Renewable Energy Laboratory.

[22] Íñigo Goiri, William Katsak, Kien Le, Thu D. Nguyen, and Ricardo Bianchini. 2013. Parasol and GreenSwitch: Managing Datacenters Powered by Renewable Energy. *SIGPLAN Not.* 48, 4 (2013). https://doi.org/10.1145/2499368.2451123

[23] Íñigo Goiri, Kien Le, Thu D. Nguyen, Jordi Guitart, Jordi Torres, and Ricardo Bianchini. 2012. GreenHadoop: Leveraging Green Energy in Data-Processing Frameworks. In *EuroSys*. https://doi.org/10.1145/2168836.2168843

[24] Íñigo Goiri, Ryan Beauchea, Kien Le, Thu D. Nguyen, Md. E. Haque, Jordi Guitart, Jordi Torres, and Ricardo Bianchini. 2011. GreenSlot: Scheduling energy consumption in green datacenters. In *International Conference for High Performance Computing, Networking, Storage, and Analysis (SC)*. https://doi.org/10.1145/2063384.2063411

[25] Léo Grange, Georges Da Costa, and Patricia Stolf. 2018. Green IT scheduling for data center powered with renewable energy. *Future Generation Computer Systems* 86 (2018). https://doi.org/10.1016/j.future.2018.03.049

[26] Synergy Research Group. 2021. *Hyperscale Data Center Capacity Doubles in Under Four Years; the US Still Accounts for Half.* Technical Report. https://www.srgresearch.com/articles/as-hyperscale-data-center-capacity-doubles-in-under-four-years-the-us-still-accounts-for-half-of-the-total

[27] Green Software Foundation (GSF). 2022. Software Carbon Intensity (SCI) Specification. Retrieved May 2023 from https://grnsft.org/sci

[28] Walid A. Hanafy, Qianlin Liang, Noman Bashir, David Irwin, and Prashant Shenoy. 2023. CarbonScaler: Leveraging Cloud Workload Elasticity for Optimizing Carbon-Efficiency. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, Article 57 (2023). https://doi.org/10.1145/3626788

[29] Walid A. Hanafy, Qianlin Liang, Noman Bashir, Abel Souza, David Irwin, and Prashant Shenoy. 2024. Going Green for Less Green: Optimizing the Cost of Reducing Cloud Carbon Emissions. In *ASPLOS*. ACM. https://doi.org/10.1145/3620666.3651374

[30] Jonathan Hasenburg, Martin Grambow, and David Bermbach. 2021. MockFog 2.0: Automated Execution of Fog Application Experiments in the Cloud. *IEEE Transactions on Cloud Computing* (2021). https://doi.org/10.1109/TCC.2021.3074988

[31] Blaine Hauglie. 2023. Xbox Is Now the First Carbon Aware Console, Update Rolling Out to Everyone Soon. Microsoft. Retrieved May 2023 from https://news.xbox.com/en-us/2023/01/11/xbox-carbon-aware-console-sustainability/

[32] International Energy Agency (IEA). 2022. Data Centres and Data Transmission Networks. Retrieved May 2023 from https://www.iea.org/reports/data-centres-and-data-transmission-networks

[33] Leila Ismail and Huned Materwala. 2020. Computing Server Power Modeling in a Data Center: Survey, Taxonomy, and Performance Evaluation. *Comput. Surveys* 53, 3, Article 58 (2020). https://doi.org/10.1145/3390605

[34] ISO/IEC 30134-2:2016 2016. *Power usage effectiveness (PUE)*. Standard.

[35] Jennie Jorgenson, Paul Gilman, and Aron Dobos. 2011. *Technical Manual for the SAM Biomass Power Generation Model*. Technical Report. National Renewable Energy Laboratory.

[36] Fiodar Kazhamiaka, Catherine Rosenberg, and Srinivasan Keshav. 2019. Tractable lithium-ion storage models for optimizing energy systems. *Energy Informatics* 2 (2019). https://doi.org/10.1186/s42162-019-0070-6

[37] Kibaek Kim, Fan Yang, Victor M. Zavala, and Andrew A. Chien. 2017. Data Centers as Dispatchable Loads to Harness Stranded Power. *IEEE Transactions on Sustainable Energy* 8, 1 (2017). https://doi.org/10.1109/TSTE.2016.2593607

[38] Ross Koningstein. 2021. We now do more computing where there's cleaner energy. Google. Retrieved May 2023 from https://blog.google/outreach-initiatives/sustainability/carbon-aware-computing-location/

[39] Alison LaBonte, O'Connor Patrick, Cash Fitzpatrick, Kathleen Hallett, and Yet Li. 2013. Standardized cost and performance reporting for marine and hydrokinetic technologies. In *1st Marine Energy Technology Symposium*.

[40] Nouhaila Lazaar, Mahmoud Barakat, Morad Hafiane, Jalal Sabor, and Hamid Gualous. 2021. Modeling and control of a hydrogen-based green data center. *Electric Power Systems Research* 199 (2021). https://doi.org/10.1016/j.epsr.2021.107374

[41] Liuzixuan Lin, Rajini Wijayawardana, Varsha Rao, Hai Nguyen, Emmanuel Wedan GNIBGA, and Andrew A. Chien. 2024. Exploding AI Power Use: an Opportunity to Rethink Grid Planning and Management. In *ACM e-Energy*. https://doi.org/10.1145/3632775.3661959

[42] Liuzixuan Lin, Victor M. Zavala, and Andrew Chien. 2021. Evaluating Coupling Models for Cloud Datacenters and Power Grids. In *ACM e-Energy*. https://doi.org/10.1145/3447555.3464868

[43] Longjun Liu, Hongbin Sun, Chao Li, Tao Li, Jingmin Xin, and Nanning Zheng. 2017. Managing Battery Aging for High Energy Availability in Green Datacenters. *IEEE Transactions on Parallel and Distributed Systems* 28, 12 (2017). https://doi.org/10.1109/TPDS.2017.2712778

[44] Ning Liu, Zhe Li, Jielong Xu, Zhiyuan Xu, Sheng Lin, Qinru Qiu, Jian Tang, and Yanzhi Wang. 2017. A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning. In *IEEE ICDCS*. https://doi.org/10.1109/ICDCS.2017.123

[45] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. 2012. Renewable and Cooling Aware Workload Management for Sustainable Data Centers. In *ACM SIGMETRICS*. https://doi.org/10.1145/2318857.2254779

[46] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H. Low, and Lachlan L.H. Andrew. 2011. Greening Geographical Load Balancing. In *ACM SIGMETRICS*. https://doi.org/10.1145/1993744.1993767

[47] E. Masanet, A. Shehabi, Nuoa Lei, S. Smith, and J. Koomey. 2020. Recalibrating global data center energy-use estimates. *Science* 367 (2020).

[48] MathWorks, Inc. 2023. Matlab Fuel Cell Stack. Retrieved May 2023 from https://de.mathworks.com/help/sps/powersys/ref/fuelcellstack.html

[49] MathWorks, Inc. 2023. Matlab Supercapacitor. Retrieved May 2023 from https://de.mathworks.com/help/sps/powersys/ref/supercapacitor.html

[50] M. Mehos, Hank Price, Robert Cable, David Kearney, Bruce Kelly, Gregory Kolb, and Frederick Morse. 2020. Concentrating Solar Power Best Practices Study. *National Renewable Energy Laboratory. NREL/TP-5500-75763* (2020).

[51] Microsoft. 2023. Windows Update is now carbon aware. Retrieved May 2023 from https://support.microsoft.com/en-us/windows/windows-update-is-now-carbon-aware-a53f39bc-5531-4bb1-9e78-db38d7a6df20

[52] Dragana D. Milosavljević, Tijana S. Kevkić, and Slavica J. Jovanović. 2022. Review and validation of photovoltaic solar simulation tools/software based on case study. *Open Physics* 20, 1 (2022). https://doi.org/doi:10.1515/phys-2022-0042

[53] Gregory L. Mines. 2016. *GETEM User Manual*. Technical Report. Idaho National Laboratory.

[54] Austin Nelson, Blake Lundstrom, Nicholas DiOrio, Aron Dobos, and Steven Janzou. 2015. *Technoeconomic modeling of battery energy storage in SAM*. Technical Report. National Renewable Energy Laboratory.

[55] Haider Niaz, Mohammad H. Shams, Amin Zarei, and Jay Liu. 2022. Leveraging renewable oversupply using a chance-constrained optimization approach for a sustainable datacenter and hydrogen refueling station: Case study of California. *Journal of Power Sources* 540 (2022). https://doi.org/10.1016/j.jpowsour.2022.231558

[56] OpenWeatherMap. 2023. Weather forecasts, nowcasts and history in a fast and elegant way. Retrieved May 2023 from https://openweathermap.org

[57] Bryan Palmintier, Dheepak Krishnamurthy, Philip Top, Steve Smith, Jeff Daily, and Jason Fuller. 2017. Design of the HELICS high-performance transmission-distribution-communication-market co-simulation framework. In *Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*. https://doi.org/10.1109/MSCPES.2017.8064542

[58] Jean-Marc Pierson, Gwilherm Baudic, Stéphane Caux, Berk Celik, Georges Da Costa, Léo Grange, Marwa Haddad, Jérôme Lecuivre, Jean-Marc Nicod, Laurent Philippe, Veronika Rehn-Sonigo, Robin Roche, Gustavo Rostirolla, Amal Sayah, Patricia Stolf, Minh-Thuyen Thi, and Christophe Varnier. 2019. DATAZERO: Datacenter With Zero Emission and Robust Management Using Renewable Energy. *IEEE Access* 7 (2019). https://doi.org/10.1109/ACCESS.2019.2930368

[59] Volker Quaschning. 2024. Meteorologische Station der HTW Berlin. https://wetter.htw-berlin.de/en

[60] Ana Radovanovic, Bokan Chen, Saurav Talukdar, Binz Roy, Alexandre Duarte, and Mahya Shahbazi. 2021. Power Modeling for Effective Datacenter Planning and Compute Management. *IEEE Transactions on Smart Grid* (2021). https://doi.org/10.1109/TSG.2021.3125275

[61] Ana Radovanovic, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, Saurav Talukdar, Eric Mullen, Kendal Smith, Mariellen Cottman, and Walfredo Cirne. 2022. Carbon-Aware Computing for Datacenters. *IEEE Transactions on Power Systems* 38, 2 (2022). https://doi.org/10.1109/TPWRS.2022.3173250

[62] John Roach. 2022. Hydrogen fuel cells could provide emission free backup power at datacenters, Microsoft says. Microsoft. https://news.microsoft.com/source/features/sustainability/hydrogen-fuel-cells-could-provide-emission-free-backup-power-at-datacenters-microsoft-says/

[63] Genevieve Saur, Vanessa Arjona, Amberlie Clutterbuck, and Eric Parker. 2019. Hydrogen and Fuel Cells for Data Center Applications Project Meeting: Workshop Report. *National Renewable Energy Laboratory. NREL/TP-5400-75355* (2019).

[64] Björn Schünemann. 2011. V2X Simulation Runtime Infrastructure VSimRTI: An Assessment Tool to Design Smart Traffic Management Systems. *Computer Networks* 55, 14 (2011), 10 pages. https://doi.org/10.1016/j.comnet.2011.05.005

[65] Kandler Smith, Aron Saxon, Matthew Keyser, Blake Lundstrom, Ziwei Cao, and Albert Roc. 2017. Life prediction model for grid-connected Li-ion battery energy storage system. In *American Control Conference (ACC)*. https://doi.org/10.23919/ACC.2017.7963578

[66] SolarAnywhere. 2023. Trusted Global Solar Data & Intelligence Services. Retrieved May 2023 from https://www.solaranywhere.com

[67] Solcast. 2023. Solar Api and Solar Weather Forecasting Tool. Retrieved May 2023 from https://solcast.com

[68] Abel Souza, Noman Bashir, Jorge Murillo, Walid Hanafy, Qianlin Liang, David Irwin, and Prashant Shenoy. 2023. Ecovisor: A Virtual Energy System for Carbon-Efficient Applications. In *ASPLOS*. https://doi.org/10.1145/3575693.3575709

[69] R. Spotnitz. 2003. Simulation of capacity fade in lithium-ion batteries. *Journal of Power Sources* 113, 1 (2003). https://doi.org/10.1016/S0378-7753(02)00490-1

[70] Cornelius Steinbrink, Marita Blank-Babazadeh, André El-Ama, Stefanie Holly, Bengt Lüers, Marvin Nebel-Wenner, Rebeca P. Ramírez Acosta, Thomas Raub, Jan Sören Schwarz, Sanja Stark, Astrid Nieße, and Sebastian Lehnhoff. 2019. CPES Testing with mosaik: Co-Simulation Planning, Execution and Analysis. *Applied Sciences* 9, 5 (2019). https://doi.org/10.3390/app9050923

[71] Valentin Sulzer, Scott G. Marquis, Robert Timms, Martin Robinson, and S. Jon Chapman. 2021. Python Battery Mathematical Modelling (PyBaMM). *Journal of Open Research Software* (2021). https://doi.org/10.5334/jors.309

[72] Moysis Symeonides, Zacharias Georgiou, Demetris Trihinas, George Pallis, and Marios D. Dikaiakos. 2020. Fogify: A Fog Computing Emulation Framework. In *IEEE/ACM Symposium on Edge Computing (SEC)*. https://doi.org/10.1109/SEC50012.2020.00011

[73] Minh-Thuyen Thi, Jean-Marc Pierson, Georges Da Costa, Patricia Stolf, Jean-Marc Nicod, Gustavo Rostirolla, and Marwa Haddad. 2020. Negotiation game for joint IT and energy management in green datacenters. *Future Generation Computer Systems* 110 (2020). https://doi.org/10.1016/j.future.2019.11.018

[74] Leon Thurner, Alexander Scheidler, Florian Schäfer, Jan-Hendrik Menke, Julian Dollichon, Friederike Meier, Steffen Meinecke, and Martin Braun. 2018. Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems. *IEEE Transactions on Power Systems* 33, 6 (2018). https://doi.org/10.1109/TPWRS.2018.2829021

[75] Jorge Val Ledesma, Rafał Wisniewski, and Carsten Skovmose Kallesøe. 2021. Smart Water Infrastructures Laboratory: Reconfigurable Test-Beds for Research in Water Infrastructures Management. *Water* 13, 13 (2021). https://doi.org/10.3390/w13131875

[76] WattTime. 2023. The Power to Choose Clean Energy. Retrieved May 2023 from https://www.watttime.org

[77] 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 *ACM/IFIP International Middleware Conference*. https://doi.org/10.1145/3464298.3493399

[78] Philipp Wiesner, Ramin Khalili, Dennis Grinwald, Pratik Agrawal, Lauritz Thamsen, and Odej Kao. 2024. FedZero: Leveraging Renewable Excess Energy in Federated Learning. In *ACM e-Energy*. https://doi.org/10.1145/3632775.3639589

[79] Philipp Wiesner, Marvin Steinke, Henrik Nickel, Yazan Kitana, and Odej Kao. 2023. Software-in-the-loop simulation for developing and testing carbon-aware applications. *Software: Practice and Experience* 53, 12 (2023), 2362–2376. https://doi.org/10.1002/spe.3275

[80] John Wilkes. 2020. *Google cluster-usage traces v3*. Technical Report. Google Inc. https://github.com/google/cluster-data/blob/master/ClusterData2019.md.

[81] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga Behram, Jinshi Huang, Charles Bai, Michael Gschwind, Anurag Gupta, Myle Ott, Anastasia Melnikov, Salvatore Candido, David Brooks, Geeta Chauhan, Benjamin Lee, Hsien-Hsin S. Lee, Bugra Akyildiz, Maximilian Balandat, Joe Spisak, Ravi Jain, Mike Rabbat, and Kim M. Hazelwood. 2022. Sustainable AI: Environmental Implications, Challenges and Opportunities. In *MLSys*. https://doi.org/10.48550/arXiv.2111.00364

[82] Chenhan Xu, Kun Wang, Peng Li, Rui Xia, Song Guo, and Minyi Guo. 2020. Renewable Energy-Aware Big Data Analytics in Geo-Distributed Data Centers with Reinforcement Learning. *IEEE Transactions on Network Science and Engineering* 7, 1 (2020). https://doi.org/10.1109/TNSE.2018.2813333

[83] Jiajia Zheng, A. Chien, and S. Suh. 2020. Mitigating Curtailment and Carbon Emissions through Load Migration between Data Centers. *Joule* 4 (2020). https://doi.org/10.1016/j.joule.2020.08.001

[84] Jiajia Zheng, Andrew A. Chien, and Sangwon Suh. 2020. Mitigating Curtailment and Carbon Emissions through Load Migration between Data Centers. *Joule* 4, 10 (2020), 2208–2222. https://doi.org/10.1016/j.joule.2020.08.001