Choosing the Right Battery Model for Data Center Simulations

Paul Kilian
KTH Royal Institute of Technology
Stockholm, Sweden

Philipp Wiesner
Technische Universität Berlin
Berlin, Germany

Odej Kao
Technische Universität Berlin
Berlin, Germany

Abstract

As demand for computing resources continues to rise, the increasing cost of electricity and anticipated regulations on carbon emissions are prompting changes in data center power systems. Many providers are now operating compute nodes in microgrids, close to renewable power generators and energy storage, to maintain full control over the cost and origin of consumed electricity. Recently, new co-simulation testbeds have emerged that integrate domain-specific simulators to support research, development, and testing of such systems in a controlled environment. Yet, choosing an appropriate battery model for data center simulations remains challenging, as it requires balancing simulation speed, realism, and ease of configuration.

In this paper, we integrate and analyze four different battery models for data center scenarios within the co-simulation framework Vessim. The results show that linear models, which consider inefficiencies and power limits, closely match the behavior of complex physics-based models in short-term experiments while offering faster execution not requiring knowledge on electrochemical reactions and circuit-level dynamics. In contrast, simple, lossless models fail to accurately represent complex performance and provide no runtime advantage.

ACM Reference Format:
Paul Kilian, Philipp Wiesner, and Odej Kao. 2024. Choosing the Right Battery Model for Data Center Simulations. In *Proceedings of 1st International Workshop on Low Carbon Computing (LOCO'24).* ACM, New York, NY, USA, 3 pages.

1 Introduction

As the adoption of modern data-intensive technologies such as artificial intelligence continues to accelerate across industries, the need for computational power is rising steadily [9, 26], leading to increased power usage of today's hyperscale data centers [13, 16]. Recent work shows that improvements in hardware and software efficiency is unlikely to alleviate future power demands [4, 13, 16], which puts pressure on operators to find other ways to decrease their carbon footprint with potential carbon pricing mechanisms on the horizon [5, 6]. These developments have shifted the focus towards a new paradigm called *carbon-aware computing* in both academia [2, 11, 18] and industry [7, 12, 15, 17], aiming to align the power usage of computing resources with local power production of renewable energy sources. This is made possible due to the fact that many computing workloads are inherently flexible in both location and time of execution, allowing them to be distributed across datacenters and scheduled for execution at times.

For the deployment of such approaches, there is a necessity to design interfaces, allowing interactions between computing and power systems. Because only large industrial players have the capacity and budget to carry out their design experiments on real hardware [18], recent work introduced Vessim [23] as a testbed for carbon-aware systems, enabling applications to gain visibility and control over a co-simulated microgrid.

Because of the rather volatile nature of renewable energy sources like solar and wind, the energy output of localized power systems will fluctuate over time, which makes the use of energy storage systems such as batteries necessary to deal with upcoming imbalances between demand and supply side [21]. Thus far, however, the opportunities of modeling battery systems in such testbeds are severely limited. This is in part due to the very little research that has been done on determining suitable battery models for various simulation use cases, even though there is a big trade-off between simple and efficient models that lack any physical properties, and physics-based models that are hard to parameterize and slow in their execution.

In this work, we therefore aim to specify the interfaces that are necessary to integrate battery models of different complexity into a microgrid co-simulation, and analyze, how these models compare. Our contributions are:

- we extend the co-simulation framework Vessim to enable step-wise execution of battery simulations
- we implement three new battery models within Vessim on top of the existing simple battery model
- we analyze the behavior and runtime of all four models in simulated scenarios

2 Battery Simulator Integration

Vessim [23] allows users to define individual microgrids to represent the power systems of computing infrastructure like data centers, utilizing the Mosaik co-simulation framework [19]. Each microgrid

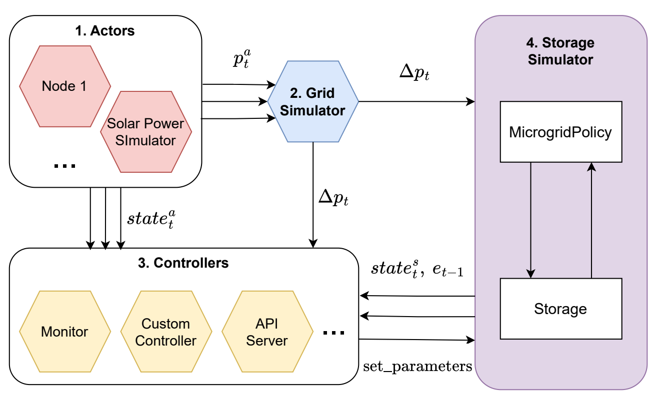

Figure 1: Co-simulation architecture for step-wise execution of battery models, based on Vessim [23].

LOCO'24, December 3, 2024, Glasgow, United Kingdom
2024.

can consist of multiple simulator responsible for modeling generators, consumers, and energy storage as well as user-defined controllers, which can provide applications visibility and control over the power system through software-in-the-loop simulation [25].

This work refines the existing co-simulation architecture to include an energy storage interface, energy management policies, and a connection between the controllers and the storage simulator, enabling control of the simulated battery system during runtime. The resulting architecture of a Vessim microgrid is shown in Figure 1.

The **Storage** interface standardizes the used battery model, allowing for charging and discharging operations through an *update* function, taking the power delta and the duration of the (dis-)charge, and returning the energy that was stored/discharged. A **Storage** has a Battery Management System (BMS) as part of the model to not allow batteries to be used in ways that would be prevented by an actual BMS like over- or undercharging, and to capture the battery's key metrics like State-of-Charge (SoC).

The **MicrogridPolicy** object manages the balance of the received power-delta, and lets users define rules for storage, exchange of energy with the large utility grid, and curtailment. At every simulation time-step, the policy instructs the **Storage** to (dis-)charge at a certain rate for a specific time according to the given rules, and computes the net energy exchanged with the grid, thus separating the management of grid and storage from the battery model itself.

By passing the battery's state and the exchanged grid delta to the controllers, and allowing the tweaking of BMS and policy parameters during runtime via the Mosaik scheduler, this architecture enables visibility and control over the battery system.

3 Battery Models

We compared four different mathematical models of rechargeable lithium-ion battery packs in the extended Vessim framework by parameterizing them to represent a group of INR21700 M50 cells.

(1) The **SimpleBattery** model was already implemented in Vessim, and simulates a basic, lossless battery that has a fixed capacity, and does not consider any inefficiencies or physical properties of a battery pack.
(2) The **CLCBattery** implements the C-L-C model as described by Kazhamiaka et al. [14]. It is a simple linear model for single lithium-ion battery cells, but considers properties such as (dis-)charging inefficiencies and power limits. This model is also used by the Carbon Explorer framework [1]. Both this and the *SimpleBattery* model have been parameterized using the cells' product specification and PyBaMM simulations [20].
(3) The **PybammBattery** uses the PyBaMM framework [20] to simulate a single lithium-ion cell that behaves according to the Single Particle Model, which takes the electrochemical processes inside a battery into account. The computation is scaled by the number of cells, and the parameterization for the modeled cell stems from Chen et al. [8].
(4) Finally, the **LiionBatteryPack** models the behavior of a full lithium-ion battery pack, where all individual cells are again emulated using PyBaMM's Single Particle Model, and the battery pack's circuit is solved using methods of the Liionpack framework [22].

4 Experimental Results

Besides different scenarios for comparing e.g. the models' speed of charge (to be presented at the workshop), we conducted a set of experiments in a more realistic setting, representing the power system of a small data center over two days. The scenario simulates servers with constant power consumption powered by a solar panel, using solar radiation data in Berlin from June 2021. Batteries are discharged to a minimum SoC of 30% to ensure operation during grid outage. Figure 2 depicts the resulting SoC over time of the different models during this experiment.

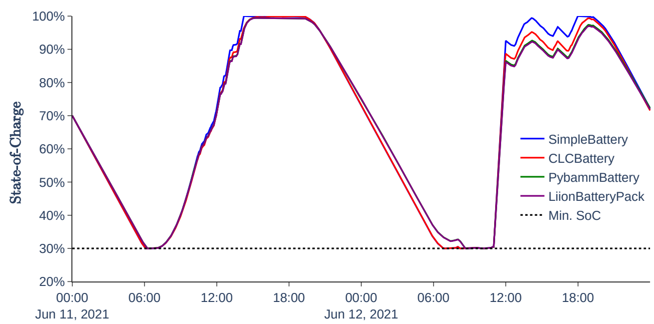

Figure 2: SoC Progression of the models during experiment.

We observe, that the *SimpleBattery* fails to accurately represent the battery's SoC, especially for charging at an almost full state, while also providing inaccurate estimations of exchanged grid energy. The *PybammBattery* and the *LiionBatteryPack* experiments exhibit that the circuit's power losses have little impact on the results in this experiment. Additionally, these models are hard to configure and analyze, requiring knowledge electrochemical reactions and circuit-level dynamics. Compared to this, the C-L-C model—given correct parameterization—accurately represents a lithium-ion cell's behavior in short-term experiments.

We analyze the models' execution time on a GCP C3 node. The *SimpleBattery* and *CLCBattery* show similar execution times due to their linearity, with a median time-step of the *PybammBattery* taking around 40 times longer. Compared to the other models, the execution time of the *LiionBatteryPack* scales linearly with the number of cells, whereas the execution time is nearly 500 times higher than for the simplest model for a battery pack of 256 cells.

5 Outlook

As a next step, we aim to extend our analysis by incorporating recent battery degradation models [10], which estimate battery lifetime based on factors like temperature, state of charge, depth-of-discharge, and (dis)charge rates. Integrating these models into data center co-simulations can play an important role in control strategies and the planning of future investments. Furthermore, we plan to apply our battery simulator integration in systems research, for example, in the development and testing of energy-aware federated learning systems with battery-powered clients [3, 24].

Future work should aim to establish reliable methods for benchmarking high-level battery models against real battery systems used in data centers. This is a challenging task, due to the inherent uncertainties in accurately estimating the SoC in physical batteries.

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] 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

[3] Amna Arouj and Ahmed M. Abdelmoniem. 2022. Towards energy-aware federated learning on battery-powered clients. In *1st ACM Workshop on Data Privacy and Federated Learning Technologies for Mobile Edge Network (FedEdge '22)*. https://doi.org/10.1145/3556557.3557952

[4] 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

[5] Rohan Best, Paul J Burke, and Frank Jotzo. 2020. Carbon pricing efficacy: Cross-country evidence. *Environmental and Resource Economics* 77, 1 (2020), 69–94.

[6] James K Boyce. 2018. Carbon pricing: effectiveness and equity. *Ecological Economics* 150 (2018), 52–61.

[7] 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

[8] Chang-Hui Chen, Ferran Brosa Planella, Kieran O'regan, Dominika Gastol, W Dhammika Widanage, and Emma Kendrick. 2020. Development of experimental techniques for parameterization of multi-scale lithium-ion battery models. *Journal of The Electrochemical Society* 167, 8 (2020), 080534. https://doi.org/10.1149/1945-7111/ab9050

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

[10] Paul Gasper, Nina Prakash, and Kandler Smith. 2024. BLAST-Lite. https://github.com/NREL/BLAST-Lite.

[11] 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

[12] 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/

[13] 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

[14] Fiodar Kazhamiaka, Catherine Rosenberg, and Srinivasan Keshav. 2019. Tractable lithium-ion storage models for optimizing energy systems. *Energy Informatics* 2 (2019), 1–22. https://doi.org/10.1186/s42162-019-0070-6

[15] 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/

[16] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. 2020. Recalibrating global data center energy-use estimates. *Science* 367, 6481 (2020), 984–986.

[17] 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

[18] 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

[19] 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

[20] 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

[21] Xingguo Tan, Qingmin Li, and Hui Wang. 2013. Advances and trends of energy storage technology in microgrid. *International Journal of Electrical Power & Energy Systems* 44, 1 (2013), 179–191. https://doi.org/10.1016/j.ijepes.2012.07.015

[22] Thomas Tranter, Robert Timms, Valentin Sulzer, Ferran Planella, Gavin Wiggins, Suryanarayana Karra, Priyanshu Agarwal, Saransh Chopra, Srikanth Allu, Paul Shearing, et al. 2022. liionpack: A Python package for simulating packs of batteries with PyBaMM. *Journal of Open Source Software* 7, 70 (2022). https://doi.org/10.21105/joss.04051

[23] Philipp Wiesner, Ilja Behnke, Paul Kilian, Marvin Steinke, and Odej Kao. 2024. Vessim: A Testbed for Carbon-Aware Applications and Systems. In *3rd Workshop on Sustainable Computer Systems (HotCarbon)*.

[24] Philipp Wiesner, Ramin Khalili, Dennis Grinwald, Pratik Agrawal, Lauritz Thamsen, and Odej Kao. 2024. FedZero: Leveraging Renewable Excess Energy in Federated Learning. In *15th ACM International Conference on Future and Sustainable Energy Systems (e-Energy)*. https://doi.org/10.1145/3632775.3639589

[25] 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

[26] 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