M3SA: Exploring Datacenter Performance and Climate-Impact with Multi- and Meta-Model Simulation and Analysis

Radu Nicolae
Vrije Universiteit Amsterdam, The Netherlands
Amsterdam, Netherlands
r.nicolae@student.vu.nl

Dante Niewenhuis
Vrije Universiteit Amsterdam, The Netherlands
Amsterdam, Netherlands
d.niewenhuis@student.vu.nl

Sacheendra Talluri
Vrije Universiteit Amsterdam, The Netherlands
Amsterdam, Netherlands
s.talluri@vu.nl

Alexandru Iosup
Vrije Universiteit Amsterdam, The Netherlands
Amsterdam, Netherlands
alexandru.iosup@gmail.com

Abstract

Datacenters are vital to our digital society, but consume a considerable fraction of global electricity and demand is projected to increase. To improve their sustainability and performance, we envision that simulators will become primary decision-making tools. However, and unlike other fields focusing on key societal infrastructure such as waterworks and mass transit, datacenter simulators do not yet combine multiple independent models into their operation and thus suffer from issues associated with singular models, such as specialization, and lack of adaptability to operational phenomena. To address this challenge, we propose M3SA, a datacenter simulation and analysis framework that uses discrete-event simulation to predict, for each model, the impact on climate and performance under various realistic datacenter conditions, and then combines these predictions. We design an architecture for simulating multiple concurrent models (*Multi-Model*), a technique for integrating the results of multiple models into a *Meta-Model*, and a procedure for quantifying Meta-Model accuracy. Through experiments with an M3SA prototype, we reproduce and enhance a peer-reviewed experiment with multi-model analysis and identify that M3SA can halve error rates of singular models (from 7.59% to 3.81%), with under 20% computational overhead.

CCS Concepts

• **Computer systems organization** → **Cloud computing**.

Keywords

datacenters, multi-model simulation, meta-model simulation, performance, energy utilization, climate impact, sustainability, OpenDC

ACM Reference Format:
Radu Nicolae, Dante Niewenhuis, Sacheendra Talluri, and Alexandru Iosup. 2026. M3SA: Exploring Datacenter Performance and Climate-Impact with Multi- and Meta-Model Simulation and Analysis. In *Proceedings of the 23rd ACM International Conference on Computing Frontiers (CF '26), May 19–21, 2026, Catania, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3801487.3806065

ACM ISBN 979-8-4007-2568-5/26/05
https://doi.org/10.1145/3801487.3806065

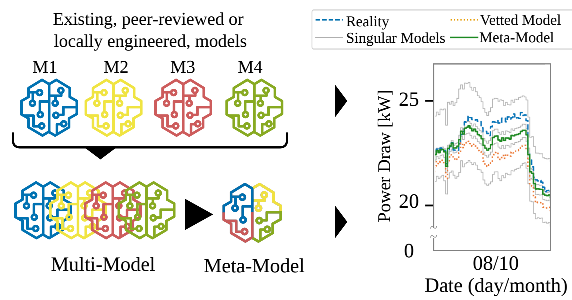

Figure 1: M3SA Multi-Model simulation uses multiple models simultaneously (M1-M4 in this figure). M3SA Meta-Model integrates all results. Figure 4 depicts a specific scenario.

1 Introduction

Our society and especially our economy are increasingly dependent on digital services. Correspondingly, new datacenters are being built and existing datacenters scaled up [1, 4, 19]. Before building or expanding, operators use simulators to project infrastructure performance and climate impact [3, 32, 35]. For example, in the EU research project Graph Massiziver, simulators predict speedup, failure cost, energy consumption, and CO_2 emissions for massive-scale infrastructure [31]. *Datacenter modeling* remains generally challenging, due to the unpredictable behavior of datacenter software and hardware ecosystems [7, 20] and the continuous emergence of new devices and applications [1, 19]. In contrast, in this work, we focus on *how the model is used in simulation*. Current simulators, such as the open-source OpenDC [21], CloudSim [8, 18], and SimGrid [9, 23], already support *single-model simulation*–the simulation is based on a single integral model, where each component or phenomenon is modelled with a partial model, e.g., one for CPUs and one for storage, and the simulator integrates them. However, as §3.2 showcases by quantifying single-model error, *single* integral models can lead to errors when encountering edge-cases or new scenarios. Addressing this gap, in this work, we design, prototype, and evaluate the M3SA (read as [mesa]) framework for *multi-model* (and *meta-model*) simulation and analysis of datacenters.

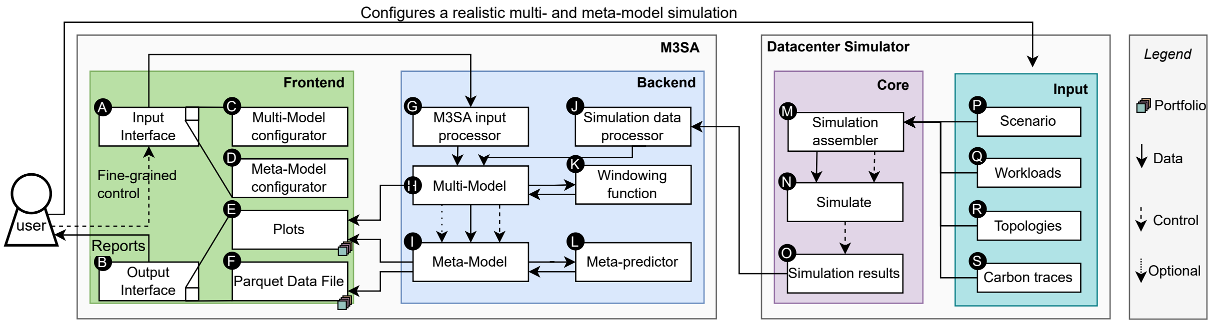

Figure 2: Architecture of M3SA, a Multi- and Meta- Model Simulation Analyzer, integrated with a black-boxed simulator.

Albeit essential, we argue that single integral (*singular*) models are sometimes insufficient for robust predictions – a singular model makes accurate predictions for the context it has been trained for, and often fails to capture some real-life edge-case scenarios [12, 21, 25]. Moreover, singular models are often calibrated to ideal or specific scenarios, and may lack adaptability to different *operational phenomena*, such as unusual spikes in demand and equipment failures. Symptomatically, no current model exists that is accepted in the community as a (de facto) standard; different popular simulators propose default models with various degrees of overlap and specialization [8, 10, 14, 21]. In this work, we posit that combining multiple models and contrasting their predictions for *the same* simulated scenario can retain the strengths of each model and alleviate their individual biases.

Figure 1 illustrates our approach, as a high-level abstraction (left) and through a real-world experiment (right, sample from §3.2). Here, M3SA consists of four singular models, the peer-reviewed Models I-IV. Firstly, M3SA assembles a *Multi-Model* from predictions of singular models, e.g., as a plot or dataframe. Then, M3SA provides a *Meta-Model*, which predicts by aggregating predictions of all singular models. Figure 1 (right) showcases the Meta-Model alleviating individual biases and better aligning with the measured reality. In this experiment, the Meta-Model delivers better accuracy than a state-of-the-art, peer-reviewed model [29], where the Meta-Model (green) curve better matches the reality (blue) curve, than the vetted model (orange, dotted curve).

Simulating datacenters using multiple models is a novelty in computer systems. However, other computer-science-related fields already adopt variations of multi- and meta-models (e.g., ensemble learning and multi-model combination techniques in machine learning (ML) and statistical modelling). Conceptually closest to the M3SA *Meta-Model*, we identify bootstrap aggregating (bagging), a state-of-the-art in ML for improving accuracy through prediction aggregation [6]. Beyond computer science, large-scale, coarse-grained simulations in weather and climate simulations [25, 33], and small-scale, detailed simulations in ecology [16] already use instruments that rely on multiple models, calibrated and adjusted for the specific needs of their scientific field.

In this work, we address the open challenge of *leveraging multiple models and integrating their results into a Meta-Model* with a three-fold contribution:

C1. (**Design**) We design M3SA, the first multi- and meta-model simulator for datacenters (§2); we design M3SA to work with generic discrete-event simulators, as an on-top layer.

C2. (**Experimentation**) We prototype M3SA and embed it into OpenDC [21], a peer-reviewed, state-of-the-art datacenter simulator. We further equip M3SA with 8 commonly used, peer-reviewed models, and conduct three large-scale, trace-driven datacenter experiments to: reproduce a peer-reviewed, open-science datacenter experiment [29] (§3.2 presents the first experiment).

C3. (**Open science**) We contribute to open science all the software and data [28] created in this project.

2 Design of M3SA: A tool for Multi- and Meta-Model Simulation and Analysis

In this section, we design M3SA, a system for datacenter simulation using *multiple* models. We expand the full design process in [27].

We design M3SA to be capable of operating coupled or decoupled from a datacenter simulator. Figure 2 depicts an overview of the system's architecture, in which M3SA extends a black-boxed simulator. In our experience, the main effort to couple M3SA to a specific simulator is put in matching the simulator output with the M3SA input. We test this by first designing and prototyping M3SA, and then coupling it with the state-of-the-art simulator OpenDC [4, 21] that has been used to explore capacity planning[5], risk management[22], and sustainability techniques[30].

In the M3SA architecture, the user interacts with the system through the *Input Interface* (component A) and *Output Interface* (B) interfaces. The M3SA process begins from the input, where the users configure the *Multi-Model* (C) and the *Meta-Model* (D). The simulation process, triggered and controlled by the M3SA backend, happens between M-S: the system sets up a simulation based on user-input (G), simulates, and centralizes predictions. Further, the M3SA backend retrieves the predictions made by the simulation infrastructure, one per singular model, and assembles the Multi-Model. To avoid redundant computation, the *Meta-Model* predicts using already leveraged and processed data from the Multi-Model. Lastly, the system plots the predictions of the Multi- and Meta-Model (E) and outputs the Meta-Model predictions to a lossless-compressed data file (F). For details, see the technical report [27].

The Multi-Model component (H) leverages multiple models into a unified tool. Within this component, predictions of singular models, which M3SA has already produced through its Datacenter Simulator backend, are centralized; the user can select the form of data

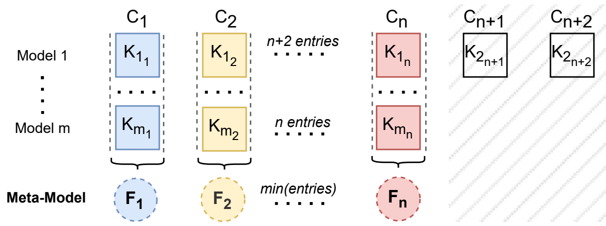

Figure 3: The Meta-Model predictor, aggregating predictions of *m* models vertically, for each time-step C_1 to C_n. Steps C_{n+1} and C_{n+2} are discarded.

Table 1: Systems Under Observation (SUO).

ID	Source	Scale [#hosts]	Resources per host
S1	SURF	277	128 GB RAM, 16 cores, 2.1 Ghz

Table 2: Workload traces used in experiments. S = scientific, SUO = system under observation, CH = CPU hours (millions), SR = sampling rate.

ID	Name	Type	SUO	Duration	Jobs	CH	SR
WT1	SURF-22	S	S1	7 days	7,850	0.31	30 s

representation. To increase performance and alleviate redundancy, the system reads from the input data file only the metrics selected by the user (typically, at configuration time) and discards irrelevant data. The Multi-Model process then filters prediction data from singular models with a user-selected windowing function, which minimizes the dataset for better performance and removes noise for better visual comprehension. After completion, the Multi-Model is ready for further operations, such as Meta-Model computation

The Meta-Model component (❶) uses an automated process to combine predictions made with multiple models into a highly accurate result. The Meta-Model strategy is similar to bootstrap aggregating (bagging), a state-of-the-art technique adopted in machine learning for improving accuracy and robustness, through prediction aggregation [6]. The Meta-Predictor (🅛), detailed in Figure 3, which receives a set of predictions over time, one per input model, with time divided into equal chunks, then aligns predictions across all models, removes entries when too few models provide predictions, and aggregates the remainder. Through this design, the Meta-Model alleviates error rates and individual biases (§3.2).

To quantify accuracy, we use the Mean Absolute Percentage Error (MAPE) [2, 24, 29], equally penalizes positive and negative errors: MAPE [%] $= \frac{1}{n}\sum_{i=0}^{n}\left|\frac{R_i - S_i}{R_i}\right| \times 100$ where, where n=number of samples, R=real-world data, S=simulation data, and i=sample index. Assisting further accuracy evaluation, M3SA is extensible to other metrics, e.g., NAD [29], RMSE [11], MAE [15, 17].

3 Complex, Trace-based Experiments

In this section, we reproduce and enhance a peer-reviewed experiment with M3SA capabilities, providing new insights into datacenter simulation and operation. In the technical report [27], through M3SA, we explore in multiple experiments the impact of failures and workload migration on energy usage and CO_2 emissions.

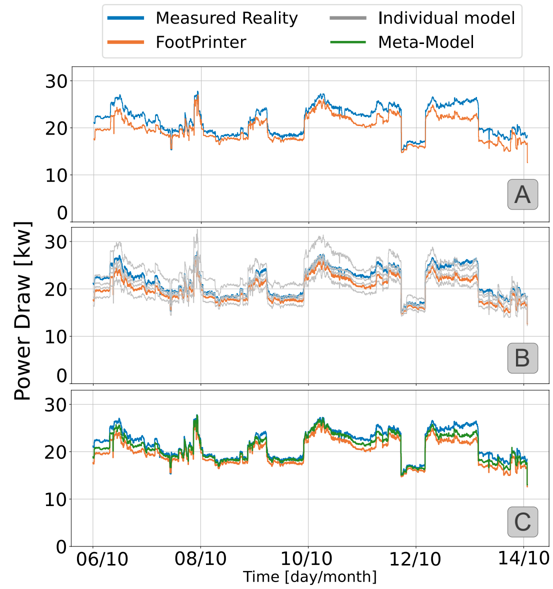

Figure 4: Simulation results vs. measured reality: (A) FootPrinter simulator, peer-reviewed experiment [29]; (B) Multi-Model with four singular models vs. FootPrinter; (C) Meta-Model derived from the Multi-Model at point B.

3.1 Experiment setup

To set up our experiments, we first implement a prototype of M3SA, interface it with OpenDC, and provide the system as open-source. Tables 1 and 2 detail the simulated infrastructure and its workload, respectively. The data is collected and open-sourced by SURF [29], the largest HPC facility for science in the Netherlands.

S1 running WT1: SURF-22 is a scientific workload trace used in peer-reviewed experiments [29]. SURF-22 traces scientific jobs run in production, batch, with an average duration of 39.52 CPU-hours, and also related energy use.

Power (draw) models: We only model power draw due to CPU usage in this work. For the Multi-Model, we equip the simulator with a library of eight commonly used, mathematical-based power-draw models [8, 13, 14, 21]. We detail each power (draw) model in the technical report [27, Appendix A].

3.2 Enhancing a peer-reviewed experiment

Niewenhuis et al. propose FootPrinter [29], a tool for predicting the CO_2 footprint of datacenters. Figure 4A shows the experiment we reproduce from [29], and Figure 4B and C show M3SA's capabilities for enriching experiments to achieve higher accuracy and more detailed explanations. Illustrative for the differences between approaches, whereas Footprinter employs a single, hand-tuned energy model, M3SA uses multiple, generic energy models.

Better explainability: Figure 4B additionally shows four singular models, whose results were automatically produced through the Multi-Model component. It seems appealing to select the most accurate model, from this sub-plot or peer-reviewed publications, by comparing against the ground truth provided by the measured reality. However, in real-world situations, and increasingly when opaque clouds provide resources, the analyst is often unaware

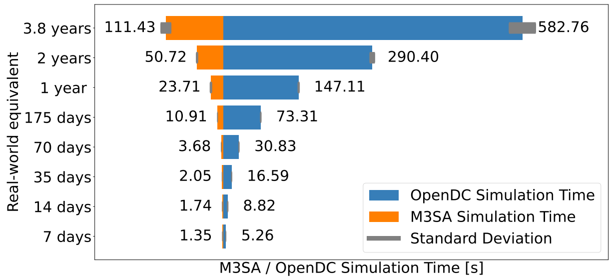

Figure 5: M3SA performance overhead compared to OpenDC simulation runtime. The horizontal axis depicts system response: (left) M3SA's overhead, (right) the OpenDC simulation time. On the vertical axis, discretely, is the duration of the workload; higher values cause higher runtimes.

of the ground-truth. The results in Figure 4B show that three of the singular models match the measured reality, while a fourth model constantly overestimates. Using the Multi-Model approach, the analyst can identify singular model biases by contrasting the predictions at each moment in time, thus increasing the overall prediction explainability[1] over singular-model approaches, where inaccuracies may not even be observable.

"All models are wrong, but some are useful" (George Box, 1976). Simulating with a robustly accurate model is a critical yet non-trivial challenge in ICT. Figure 4C shows the predictions of a Meta-Model, automatically derived from the Multi-Model presented in Figure 4B, and compared against the predictions of the FootPrinter model and the measured reality. We quantify accuracy against measured reality and obtain a MAPE of 7.59% for the average singular model, and a MAPE of 3.81% (≈50% better) for the Meta-Model. The improvement in error rate is due to the *median* aggregation function, which effectively filters extremes, reduces biases, and improves the robustness of the prediction, while remaining model-free.

Compared to state-of-the-art FootPrinter's MAPE of 3.15%, the Meta-Model has a higher error rate. However, FootPrinter underlies a manual simulation model, explicitly trained for a single trace (here, SURF-22). In contrast, M3SA enables the versatile, automatic re-use of generic models.

M3SA runtime overhead under realistic conditions: Throughout this experiment, M3SA operated on 20,170 samples, or about 7 days of operation with SURF monitoring rate of 30 seconds [21]. To assess the runtime overhead of M3SA, we further create datasets of various sizes, equivalent to different datacenter operational times, including 2 years and up to nearly 4 years, by modifying the monitoring granularity of the simulator.

Figure 5 summarizes the results of this performance-analysis experiment. The M3SA overhead includes synthesizing a Multi-Model, computing the Meta-Model, and outputting the results. We run each experiment 10 times on a standard MacBook M2 Pro and represent the standard deviation in gray (full analysis in [27], Appendix B). On average, M3SA adds less than 20% time overhead to the overall simulation process, even for massive-scale datasets, and can simulate years of datacenter operation within minutes.

[1]Resolving the discrepancies between models, and identifying a single best model, requires capabilities beyond the scope of this work.

4 Related Work

Simulating using multiple models is a novelty in computer systems; however, as introduced in Section 1, other scientific fields already employ multi-model simulations, such as machine learning (ensemble learning, bagging [6]), statistical modelling, virology [12], weather and climate simulation [33], or ecology [16]. Furthermore, existing ICT simulators (e.g., OpenDC [21], CloudSim [8]) or general simulation frameworks (e.g., Mosaik [34]) enable co-simulation by combining and coordinating individual models or tools, yet each for a different purpose (e.g., a model for CPU utilization, a model for CO_2 emissions, and a simulation tool for datacenter thermal system). In ICT, existing simulators currently rely on singular model predictions [8, 9, 18, 21, 23]. M3SA complements this related work.

Closest to our work, we identify COVID-19 Forecast Hub [12], used to predict the COVID-19 hospitalizations, and provides 93 models, out of which users can select and leverage in a unified visualization. This is equivalent to the *Multi-Model* component of M3SA but for a different field. In weather and climate simulation, Myhre et al. [25] investigate harmful emissions using seven models, aggregated (averaged) into a "Mean Model"; this is equivalent to the *Meta-Model* component of M3SA.

Datacenter simulators are useful tools for the community and are widely used in predicting ICT infrastructure. However, prior to this work, no simulator provides multi- and meta-simulation capabilities but relies on singular-model predictions [8, 9, 18, 21, 23].

5 Conclusion and Future Work

Understanding the performance and climate impact of existing and upcoming datacenters is essential to our society and economy. In this work, we designed, implemented, and evaluated M3SA, a framework for multi- and meta-model simulation and analysis of datacenters. We evaluated M3SA deployed in OpenDC using real-world workloads. Results indicate M3SA's ability to predict in complex scenarios, with enhanced explainability and robustness over approaches that only use singular models. Such capabilities can aid analysts and C-level decision-makers to better reason about real-world scenarios and make informed decisions. In our experiments, M3SA can simulate years of datacenter operation in minutes, leading to lower error rate than singular models, and enabling detailed explanation without significant overhead; our technical report [27] presents further results.

We have released M3SA as an open-sourced system that can be applied as a top layer to datacenter simulators and tested its operation with the commonly used open-source simulator OpenDC. We envision future research in exploring MCDA and various methods of leveraging a Meta-Model, such as dynamically allocating weights to predictions of singular models, and are currently embedding M3SA into a simulation-based *digital twin* [26].

Acknowledgments

This work was partially supported by EU MSCA CloudStars (g.a. 101086248) and EU Horizon Graph Massivizer (g.a. 101093202). This research is partly supported by a National Growth Fund through the Dutch 6G flagship project "Future Network Services."

References

[1] 2024. https://www.gartner.com/en/documents/5178763.
[2] 2024. https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusu/insights_metrics_MAPE.html.
[3] 2025. https://graph-massivizer.eu/.
[4] G. Andreadis et al. 2018. A reference architecture for datacenter scheduling. In *SC*.
[5] G. Andreadis et al. 2022. Capelin. *TPDS* (2022).
[6] L. Breiman. 1996. Bagging Predictors. *ML* (1996).
[7] N. Bronson et al. 2021. Metastable failures in distributed systems. In *HotOS*.
[8] R. N. Calheiros et al. 2011. CloudSim. *Softw. Pract. Exp.* (2011).
[9] H. Casanova. 2001. Simgrid. In *CCGrid*.
[10] H. Casanova et al. 2025. Lowering entry barriers to developing custom simulators of distributed applications and platforms with SimGrid. *PC* (2025).
[11] T. Chai et al. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? *GMD* (2014).
[12] E. Cramer et al. 2022. The united states covid-19 forecast hub dataset. *Nature* (2022).
[13] R. Ferreira da Silva et al. 2019. Accurately Simulating Energy Consumption of I/O-Intensive Scientific Workflows. In *ICCS*.
[14] X. Fan et al. 2007. Power provisioning for a warehouse-sized computer. In *ISCA*.
[15] A. Gunawardana et al. 2009. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *JMLR* (2009).
[16] X. A. Harrison et al. 2018. A brief introduction to mixed effects modelling and multi-model inference in ecology. *PeerJ* (2018).
[17] J. L. Herlocker et al. 2004. Evaluating collaborative filtering recommender systems. *TOIS* (2004).
[18] T. B. Hewage et al. 2024. CloudSim express. *SPE* (2024).
[19] IDC. 2024. https://www.idc.com/getdoc.jsp?containerId=US52131624.
[20] A. Iosup et al. 2018. Massivizing Computer Systems. In *ICDCS*.
[21] F. Mastenbroek et al. 2021. OpenDC 2.0. In *CCGrid*.
[22] F. Mastenbroek et al. 2025. RADiCe. *FGCS* (2025).
[23] J. McDonald et al. 2024. An exploration of online-simulation-driven portfolio scheduling in Workflow Management Systems. *FGCS* (2024).
[24] J. J. M. Moreno et al. 2013. Using the R-MAPE index as a resistant measure of forecast accuracy. *Psicothema* (2013).
[25] G. Myhre et al. 2017. Multi-model simulations of aerosol and ozone radiative. *ACP* (2017).
[26] National Academy of Engineering and National Academies of Sciences, Engineering, and Medicine. 2024. *Foundational Research Gaps and Future Directions for Digital Twins*.
[27] R. Nicolae et al. 2026. M3SA technical report. http://arxiv.org/abs/2603.29778.
[28] R. Nicolae et al. 2026. M3SA trace archive. https://github.com/atlarge-research/m3sa-traces.
[29] D. Niewenhuis et al. 2024. FootPrinter. In *ICPE*.
[30] D. Niewenhuis et al. 2026. OpenDC-STEAM. *CCGRID* (2026).
[31] R. Prodan et al. 2022. Towards extreme and sustainable graph processing for urgent societal challenges in Europe. In *Cloud Summit*.
[32] C. Ren et al. 2012. Carbon-Aware Energy Capacity Planning for Datacenters. In *MASCOTS*.
[33] R. W. Schunk et al. 2016. Space weather forecasting with a multimodel ensemble prediction system (MEPS). *RS* (2016).
[34] C. Steinbrink et al. 2019. CPES Testing with mosaik. *Applied Sciences* (2019).
[35] S. Tuli et al. 2022. COSCO. *TPDS* (2022).

A Artifact Appendix

A.1 Abstract

The source code of M3SA, both coupled with and decoupled from OpenDC, is available at github.com/atlarge-research/opendc-m3sa. M3SA decoupled can be found in `m3sa` directory and M3SA coupled with OpenDC in `m3sa-opendc`. We facilitate experiment and visualization reproducibility through a reproducibility capsule, available at github.com/atlarge-research/opendc-m3sa-reproducibility-capsule. The reproducibility capsule is Docker-based and supports two main commands, one for reproducing the isolated experiment presented in the article, and a command for reproducing all the experiments and visualizations presented in the technical report [27].

We now present the checklist for reproducing the experiment.

(1) **Program:** OpenDC-M3SA
(2) **Binary:** Default OpenDC JAR, also provided in the capsule
(3) **Data set:** workload traces available in `experiments` folder and detailed in https://github.com/atlarge-research/opendc-traces
(4) **Run-time environment:** Docker container (tested on `docker-27.4` and above, `python:3.12-slim` base image, bundling `JDK-21`)
(5) **Metrics:** power draw in the simulated environment (in the technical report, also CO2 emissions)
(6) **Output:** the figures, in pdf format. Running the capsule generates (reproduces) *(i) the figures from the article*, and (ii) other figures, from other experiments included in the technical report. For (i), the output figures `figure-9A.pdf`, `figure-9B.pdf`, and `figure-9C.pdf` correspond to the article's Figure 4 A, B, and C, respectively.
(7) **Approximate disk space required:** 2.5 GiB, including the article and technical report results.
(8) **Time needed to prepare workflow:** Under 5 minutes
(9) **Time needed to complete experiments:** ≈ 2 min for the article experiment. ≈ 15 min for all the capsule, including the experiments in the technical report.
(10) **Publicly available?:** Yes
(11) **Code and Data licenses:** AGPL-3.0
(12) **Archived:** 10.5281/zenodo.19072789

A.2 Description

A.2.1 How to access. We provide the reproducibility capsule via GitHub and Zenodo. We recommend viewing the capsule via GitHub and following the steps from the README.md.

(1) Reproducibility capsule:
Github: opendc-m3sa-reproducibility-capsule
Zenodo: 10.5281/zenodo.19072789.
(2) M3SA-OpenDC: atlarge-research/opendc-m3sa
Coupled: `m3sa-opendc` directory
Decoupled: `m3sa` directory.

A.2.2 Hardware dependencies (no specific hardware needed). We tested the reproducibility capsule on X86-64 and ARM architecture, on MacOS and generic Linux distributions.

A.2.3 Software dependencies. Docker v27.4 or higher and a Unix shell for running the commands presented in Appendix A.4.

A.3 Installation

The reproducibility capsule can run via Docker or locally. We recommend the Docker-based capsule, using the pre-built Docker image. We detail the steps below.

```
# Step 1: Use the pre-built Docker image
# pull the pre-built Docker image
docker pull --platform linux/amd64 radu33/m3sa:m3sa-experiment
docker tag radu33/m3sa:m3sa-experiment m3sa-experiment

# Step 2: Run only the experiment in the article
docker run --platform linux/amd64 --rm -v $(pwd):/app/reproduced \
    m3sa-experiment experiment1

# (Optional) Step 3: Run all the experiments
docker run --platform linux/amd64 --rm -v $(pwd):/app/reproduced \
    m3sa-experiment
```

A.4 Evaluation and expected results

The results are the plots (i) from the paper and (ii) technical report, in pdf format, included in the current working directory. For (i), the output `figure-9A,-9B,-9C.pdf`, correspond to article's `figure-4A,-4B,-4C`, respectively.