A Digital Twin Framework for Liquid-cooled Supercomputers as Demonstrated at Exascale

Wesley Brewer
Oak Ridge National Laboratory
Oak Ridge, TN, USA
brewerwh@ornl.gov

Matthias Maiterth
Oak Ridge National Laboratory
Oak Ridge, TN, USA
maitherthm@ornl.gov

Vineet Kumar
Oak Ridge National Laboratory
Oak Ridge, TN, USA
kumarv@ornl.gov

Rafal Wojda
Oak Ridge National Laboratory
Oak Ridge, TN, USA
wojdarp@ornl.gov

Sedrick Bouknight
Oak Ridge National Laboratory
Oak Ridge, TN, USA
bouknightsl@ornl.gov

Jesse Hines
Oak Ridge National Laboratory
Oak Ridge, TN, USA
hinesjr@ornl.gov

Woong Shin
Oak Ridge National Laboratory
Oak Ridge, TN, USA
shinw@ornl.gov

Scott Greenwood
Oak Ridge National Laboratory
Oak Ridge, TN, USA
greenwoodms@ornl.gov

David Grant
Oak Ridge National Laboratory
Oak Ridge, TN, USA
grantdr@ornl.gov

Wesley Williams
Oak Ridge National Laboratory
Oak Ridge, TN, USA
williamswc@ornl.gov

Feiyi Wang
Oak Ridge National Laboratory
Oak Ridge, TN, USA
fwang2@ornl.gov

***Abstract*—We present ExaDigiT, an open-source framework for developing comprehensive digital twins of liquid-cooled supercomputers. It integrates three main modules: (1) a resource allocator and power simulator, (2) a transient thermo-fluidic cooling model, and (3) an augmented reality model of the supercomputer and central energy plant. The framework enables the study of "what-if" scenarios, system optimizations, and virtual prototyping of future systems. Using Frontier as a case study, we demonstrate the framework's capabilities by replaying six months of system telemetry for systematic verification and validation. Such a comprehensive analysis of a liquid-cooled exascale supercomputer is the first of its kind. ExaDigiT elucidates complex transient cooling system dynamics, runs synthetic or real workloads, and predicts energy losses due to rectification and voltage conversion. Throughout our paper, we present lessons learned to benefit HPC practitioners developing similar digital twins. We envision the digital twin will be a key enabler for sustainable, energy-efficient supercomputing.**

***Index Terms*—digital twins, exascale computing, energy efficiency, augmented reality, data center power, electronics cooling**

I. Introduction

A drastic reduction in power consumption was the key to realizing exascale supercomputing. In 2008, DARPA published a study where they made projections about what it would take to reach a target design for exascale computing of 20 megawatts (MW) per exaflop (EF) [1]. They projected that an exascale system could be achieved in 2015, requiring between 68 and 155 MW/EF [2]. Frontier was deployed in 2021 and achieved 1.102 EF in June 2022 using an average power of 21.1 MW [3]; its performance improved in November 2023 to 1.194 EF performance at 22.7 MW [4], and again in June 2024 to 1.206 EF at 22.8 MW [5]. To connect the dots on how this has played out over the past 15 years, consider that in 2009 technology the projected energy cost of scaling the Jaguar supercomputer to exascale performance would have required about three gigawatts. This was markedly reduced down to 330 MW/EF with the advent of GPUs in the Titan supercomputer in 2012, and further down to 65 MW/EF for Summit in 2018, and finally down to 19 MW/EF for Frontier. At the same time each generation of supercomputer has achieved a tenfold increase in performance: from 2.5 petaflops (PF) for Jaguar, to 27 PF for Titan, to 200 PF for Summit, to 2 EF for Frontier.

While significant efforts in hardware optimization have driven these drastic advancements in efficiency, we have reached a point where the system is so highly optimized that little room remains for further improvements. On the operational side, two main areas of research have been instrumental for improving datacenter efficiency: simulations [6], and analysis of system telemetry [7]. Additional improvements necessitate innovative tools that focus on end-to-end

SC24, November 17-22, 2024, Atlanta, Georgia, USA
979-8-3503-5291-7/24/$31.00 ©2024 IEEE

improvement, such as mixed-precision iterative refinement [8], AI-based surrogate models [9], and digital twins (DTs). *We hypothesize that developing a comprehensive digital twin of both the facility as well as the supercomputer will provide a robust tool for end-to-end optimization across multiple facets.*

DTs have emerged as a means of merging both telemetry and simulations to develop a holistic virtual representation of the system, bridging both the physical and virtual worlds. The AIAA digital engineering integration committee defines a DT as [10]:

> "...a set of virtual information constructs that mimics the structure, context, and behavior of an individual / unique physical asset, or a group of physical assets, is dynamically updated with data from its physical twin throughout its life cycle, and informs decisions that realize value."

Here, the mimicking *structure* refers to the 3D modeling of the physical assets (racks, servers, pumps, etc.), while mimicking *behavior* refers to developing either simulations or AI/ML models, and *dynamically updated* speaks of telemetry data generated from the physical twin.

DTs provide value proposition at multiple stages of the data center lifecycle: (1) planning/design, (2) construction, and (3) operations [11]. During the planning/design phase, the DT can enable more informed decisions for future acquisitions, and provide predictive capabilities of energy efficiency for future systems. This can be accomplished via virtual prototyping capabilities, such as the ability to virtually design and test the cooling system, or design virtual networks to study multi-application interactions on congestion performance. During the construction phase, the DT streamlines construction costs, reduces construction waste, and reduces outages when retrofitting systems. During the operations phase, the DT enables predictive maintenance, extends asset life through reliability and availability modeling, and enhances cooling efficiency by optimizing system behavior. For these reasons and more, digital twins are widely becoming "management best practice" in various industries [12].

To build such a tool, we needed to be able to integrate both system telemetry and simulation, integrate models from across multiple domains, and demonstrate the approach through real-world examples. To that end, in this paper we provide the following specific contributions:

1) An open-source reusable digital twin framework for data centers, ExaDigiT, with modular sub-components supported by telemetry and simulation,
2) Extensive verification and validation studies of the framework,
3) Demonstration of the framework at exascale for Frontier.

Our paper is structured as follows: In Section II, we first give a background of other research related to digital twin modeling of the various facets of the data center, then in Section III we give an overview of the ExaDigiT architecture, discussing how each component was developed, then in Section IV we discuss verification and validation of the framework, and finally close with Section VI on conclusions and future work. Throughout the various sections, we have highlighted valuable lessons learned, denoted as "findings".

II. Background

Research towards development of digital twins for data centers has mainly been separate, siloed efforts focused on either data center cooling, network performance, power consumption, or visualization efforts. We discuss the history of research for energy consumption, cooling models, and visual analytics, identifying the state-of-the-art (SOTA) techniques in each area, highlighting for each area how our work differs.

1) Energy Consumption Models: There have been a couple of rather extensive surveys of previous work on power consumption modeling of *data centers*, which are generally air-cooled CPU-only systems. Dayarathna and Wen [13] conducted an in-depth analysis of over 200 power consumption models. From their detailed analysis, they deduced that as of 2015: (1) a significant portion of the research primarily focused on the lower levels of the data center hierarchy, often overlooking the broader perspective of the entire facility, (2) many studies considered only a limited set of server metrics, (3) the accuracy of these models remained a major concern. More recently, Jin et al. [14] surveyed literature on power consumption modeling efforts of CPU-only air-cooled servers. They analyzed 47 different models, which they classify into the following model categories: additive, BA (baseline power + active power), simple regression, multiple regression, power function, non-linear, polynomial, and other. Regarding GPU-enabled HPC systems, Sîrbu and Babaoglu's work [15] represents the SOTA in power systems modeling of HPC systems. They developed a predictive model for total system power of a hybrid CPU-GPU-MIC supercomputer. Their model is based on three components: (1) using support vector regression to predict power per job before the jobs are started, (2) predicting job duration via a simple heuristic, and (3) predicting total system power based on the measured power of computing units. They mention that few models address system-level prediction of power, a point that was originally made by Dayarathna and Wen [13]. Our model focuses on predicting the total system power at fixed time intervals as well as modeling the losses due to energy conversion.

2) Thermo-Fluid Cooling Models: Most of the work on data center cooling has been focused on air-cooled systems, e.g. [16]–[18], especially focused on *cooling efficiency*. Zohdi [19] presents the development of a digital twin and machine learning framework to model an idealized air-cooled system with thermal effects using direct numerical simulation (DNS). The machine learning framework uses genetic algorithms to learn the optimal parameters, such as flow rates in/out of multiple room vents. Moreover, Zhang et al. [20] built a digital twin for data centers called "Smart DC". Their framework uses a combination of Computational Fluid Dynamics (CFD), 6SigmaDC [21], demonstrated for an air-cooled data center, along with AI in the form of XGBoost to optimize the control parameters of the air conditioning system in order to optimize

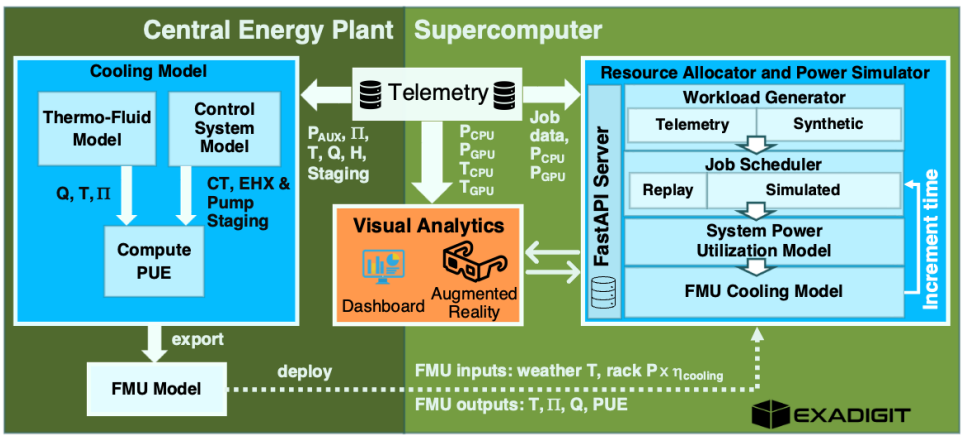


Fig. 1. ExaDigiT architectural overview.

the Power Usage Effectiveness (PUE) of a datacenter. They show that their method can effectively reduce the PUE from 1.15 to 1.08. In terms of liquid-cooled systems, Heydari et al. [22] represents the SOTA in this field, having performed extensive analysis of liquid-cooled systems by running CFD simulations using both Macroflow [23] for blade-level flow loops, 6SigmaET [24] for thermal simulations of cold plates, along with 6SigmaRoom [21] for simulating the air flow in the room. Whereas their cooling model is based on expensive proprietary software, our cooling model is built on an open-source Modelica framework [25], [26] and supports modeling the transient dynamics of the entire liquid cooling system.

3) Visual Analytics: There has also been work primarily focused on the *visualization* aspects of supercomputers, with efforts mainly targeting HPC monitoring [27]–[29], visualizing file system activity [30], and visualizing network traffic [31]–[33]. Bergeron et al. [29] and Riha et al. [28] represent the SOTA in HPC monitoring. Our visual analytics approach is unique in that it combines telemetry and simulations in a single environment, supporting both augmented reality (AR) and dashboard, enabling the replay of live system telemetry or launching "what-if" simulation experiments.

To summarize, there has been a significant amount of research investigating either power or cooling or visualizations, but a conspicuous lack of research attempting to build a comprehensive open-source tool for holistically modeling the system. The one exception that we are aware of is the work by NVIDIA [34], which uses Cadence 6SigmaDCX and NVIDIA Modulus [35] to model an air-cooled data center, NVIDIA Air for modeling the network via Cumulus virtual machines, and NVIDIA Omniverse for managing and visualizing 3D assets. While we initially experimented with these tools, the toolkit did not provide support for modeling liquid-cooled systems, resource allocation, and energy consumption; therefore we decided to forge our own open-source framework.

In this work, we present our development of an open-source digital twin framework, which is currently designed for modeling liquid-cooled systems. The open stack of software consists of a Python-based Resource Allocator and Power Simulator (RAPS) module, a Modelica-based thermo-fluids cooling model, and C++-based 3D interactive augmented reality model utilizing Unreal Engine 5 (UE5), as shown in Fig. 1.

III. ExaDigiT Architecture

In this section we first give an overview of the high-level ExaDigiT architecture, and then step through the process of development from requirements analysis, technical specifications, and then discuss the details of the various components. Fig. 1 shows the architectural overview of the various components of ExaDigiT. There are three main modules that we develop: (1) RAPS, (2) a cooling model, and (3) visual analytics capabilities. The RAPS module can replay workloads from telemetry, reschedule them, or simulate synthetic workloads on the supercomputer to analyze the resulting energy consumption; further details are provided in Section III-B. The cooling model simulates thermo-fluid dynamics and control of the Central Energy Plant (CEP), which itself includes three components: (1) a thermo-fluid model for predicting temperatures (T), pressures (Π), and flow rates (Q); (2) a control system model for predicting the staging of cooling towers, hot/cold water pumps, and heat exchangers; and (3) a sub-module for predicting the system PUE. This will be discussed in further detail in Section III-C. Both RAPS and the cooling model can be interfaced either via a terminal console, the web-based dashboard, or the augmented reality environment for visual insights, which will be discussed in Section III-D and shown in Fig. 6.

Each module of the digital twin generally falls into one of the following five categories [36]:

(L1) The *descriptive* twin models the physical assets using both CAD models such as 3D modeling files (e.g., Autocad/Revit), as well as game engines (e.g., Unreal Engine, Unity, or NVIDIA Omniverse).

(L2) The *informative* twin incorporates the telemetry data to provide real-time data insights into the physical twin.

(L3) The *predictive* twin utilizes telemetry data to develop data-driven predictive models using AI/ML.

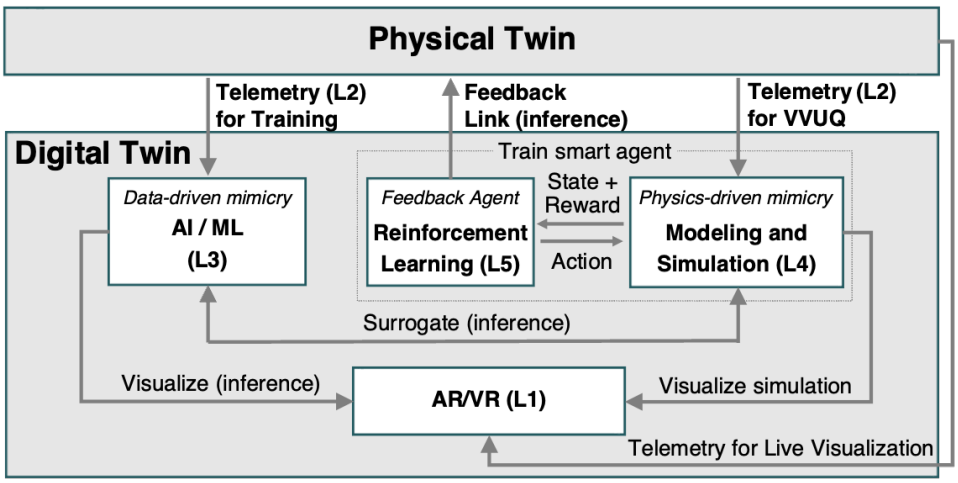


Fig. 2. Relationships between DT levels.

(L4) The *comprehensive* twin leverages modeling and simulation techniques to provide virtual prototyping capabilities that are able to address "what-if" scenarios – generally not in real-time.

(L5) The *autonomous* twin uses techniques such as reinforcement learning to learn to make autonomous decisions for system optimization.

These levels relate to each other as depicted in Fig. 2. This paper covers using L1 for visualization, L2 for validation, and L4 for modeling and simulation. AI/ML L3 models are built on telemetry data, and therefore are fundamentally interpolative and thus often do not extrapolate well [37]; however, they can generally be inferenced in real-time. L4 simulations, based on first principles, are the primary engine of digital twins [38]; unlike AI/ML models, they generally require a longer development time, and utilize telemetry data for validation. They are also more computationally expensive, generally making real-time operation unfeasible. Despite these challenges, L4 simulations are extrapolative and can be effectively used for virtual prototyping. An alternative approach is to use the simulations to generate data to train a machine-learned *surrogate model*, which has the advantage of being able to run in real-time, but can also be used to model virtual prototypes [39]. Reinforcement learning (L5) is used primarily for training autonomous agents that can be used to make control decisions in order to optimize processes. An example of L5 would be training an agent to perform automated setpoint control for improved cooling efficiency by minimizing setpoint overshoot of proportional-integral-derivative (PID) controllers [40].

Finding 1. *Simulations are the primary building blocks of digital twins. However, machine-learned models should also have a significant role for modeling system workloads, i.e., application fingerprinting. Furthermore, system telemetry plays critical roles in our development of machine-learned models and for validation.*

A. Requirements Analysis and Gathering

At the project's outset, we interviewed several HPC engineers to conduct a requirements analysis—our goal was to identify potential use cases for the digital twin and pinpoint opportunities for demonstrations. We list some of the potential use cases here that resulted from those discussions:

- Understanding dynamic energy consumption and losses in the system, as well as visualizing energy consumption on a per-job basis.
- Virtually extending the cooling system to support a secondary HPC system in the future, and evaluating its impact on cooling performance of the current system (an idea further explored in [41]).
- Understanding temperature problems in the past and problems with cooling loops by visualizing heat maps in the system.
- Early detection of thermal throttling–when the system has to reduce CPU or GPU clock frequency to reduce the heat generated by the system.
- Understanding how weather correlates to GPU temperatures on the system.
- Understanding water quality issues and how they affect system performance. Water-based coolants can suffer from biological growth in the blade-level cooling system causing blockage to specific nodes. Can these types of blockages be detected?
- Studying effects of SCADA cybersecurity attacks, e.g., [42], [43].
- Using visualization to debug Slingshot network congestion, i.e., identify hot spots in the system.

All of these use cases directly impact performance, power consumption, and operational efficiency. From our analysis, we identified several categories for potential use cases: augmented reality, forensic analysis and diagnostics, predictive modeling, failure detection, operational optimization, "what-if" scenarios, and virtual prototyping.

Finding 2. *From discussions with various stakeholders including HPC engineers and project managers from both data centers and industrial suppliers, we found that there was a significant need to develop a robust comprehensive digital twin framework for data centers. Such a tool would be valuable for forensic diagnostics and augmentations of operational systems as well as virtual prototyping for future systems.*

The second step in developing the DT involved gathering all the required technical specifications covering the various facets of the architecture, such as converter efficiency curves for rectifiers and voltage converters, pump curves, thermal resistance curves of cold plates, etc. Gathering such information involved many challenges, such as contacting several different organizations, working with different file formats, etc. Understanding and documenting the various technical requirements will be helpful for future developments of digital twins, especially if standard exchange formats can be developed for specifying all the necessary information required to build the DT.

Finding 3. *DT development is a holistic effort requiring everyone to be on board, making it challenging to get started; it touches every aspect of the organization, crossing boundaries of the system and organizations. To effectively integrate digital twins with the procurement lifecycle, it is important to identify the stakeholders for each component subgroup to ensure the model is ready as the system comes online.*

B. Resource Allocator and Power Simulator

Our requirements analysis and gathering revealed the need for a module that could accurately simulate resource allocation and power prediction. Therefore, we have developed the RAPS module, which is a tight integration of both the job scheduler in concert with dynamic power consumption calculations, which we describe in detail in this section. Algorithm 1 shows the pseudocode for the RAPS module. An initial array of jobs is created either synthetically or from telemetry data, where each job is characterized by: (1) the number of nodes required, (2) the wall time, and (3) CPU/GPU utilization traces[1] for a given trace quanta.[2] Jobs from telemetry may be replayed using the physical twin's scheduling policy, or may use a built-in scheduler, as described in III-B4, which is also used for synthetic jobs. Time is advanced every second and power is computed at every second for the system; however, the cooling model is only called every 15s during the simulation.

1) Estimation of Power Conversion Losses: As shown in Fig. 3, each rack of Frontier consists of four shelves, each shelf has two chassis, and each chassis contains four active rectifiers and eight compute blades, yielding a total of 64 blades and 32 rectifiers per rack. Each rack is directly supplied with three-phase power from the distribution transformer switchboard. The three-phase power is distributed over the rack supplying 32 power rectifiers connected in parallel. The group of four rectifiers shares a common output DC bus, providing power to eight blades; these blades, in turn, feed power to 16 step-down DC-DC converters that are connected in parallel.

[1]Since our system telemetry lacks CPU/GPU utilization, we linearly interpolate power to utilization. In the future, we plan to use profiling to obtain CPU/GPU/network utilization traces for a set of representative applications, as discussed in [44].

[2]Set to 15s in this work to correspond with system telemetry data and will be further discussed in Section IV.

TABLE I
COMPONENT OVERVIEW OF THE FRONTIER SUPERCOMPUTER

Component	Quantity
Number of CDUs	25
Racks per CDU	3
Chassis per Rack	8
Rectifiers per Rack	32
Blades per Rack	64
Nodes per Rack	128
SIVOCs per Rack	128
Switches per Rack	32
Nodes Total	9472

Component	Power
GPU (Idle)	88 W
GPU (Max)	560 W
CPU (Idle)	90 W
CPU (Max)	280 W
RAM (Avg)	74 W
NVMe (Avg)	30 W
NIC (Avg)	80 W
Switch (Avg)	250 W
CDU (Avg)	8700 W

The role of the rectifier is to convert the three-phase AC energy supplied from the distribution transformer into constant energy that is distributed among the blades. In every partition, constant energy is distributed through a common DC bus, so that in case of rectifier failure, blades are continuously powered and should perform their job without any interruption. Each blade consists of two isolated DC-DC step-down converters, also known as Super Intermediate Voltage Converters (SIVOCs), as shown in Fig. 3 [45], which further steps down the 380V DC voltage from the rectifier into 48V DC voltage that supplies power to the node. The total efficiency of the energy conversion η_{system} of the active rectifier and the SIVOC converter is given by:

$$\eta_{system} = \eta_R \eta_S = \frac{P_{R_{DC}}}{P_{R_{AC}}} \frac{P_{S_{48V}}}{P_{R_{DC}}} = \frac{P_{S_{48V}}}{P_{R_{AC}}} \tag{1}$$

and the total power conversion loss P_L is given by:

$$P_L = P_{L_R} + P_{L_S} = P_{R_{DC}}(P_{R_{AC}} - P_{S_{48V}}), \tag{2}$$

where $P_{R_{AC}}$ is the rectifier input power, $P_{R_{DC}}$ is the rectifier output power, and $P_{S_{48V}}$ is the SIVOC output power. η_R and η_S is the rectifier and converter efficiency, which were respectively determined to be 0.96 and 0.98. Therefore the total system efficiency according to (1) is roughly[3] 0.94.

2) Dynamic Power Estimation: Frontier is composed of 9472 "Bard Peak" nodes, with each blade housing two such nodes. Each node contains an AMD EPYC™ 7A53 "Trento" 64-core 2-GHz CPU and four AMD Instinct MI250X GPUs [2]. Each node's power can be computed by summing its individual components summarized in Table I:

$$P_{node} = P_{CPU} + 4P_{GPU} + 4P_{NIC} + P_{RAM} + 2P_{NVMe} \tag{3}$$

[3]Note, that these efficiencies are simplifications for this discourse (yet still within one percent of the actual value); in reality, the efficiency slightly varies based on input power, which is discussed in more detail in [46].

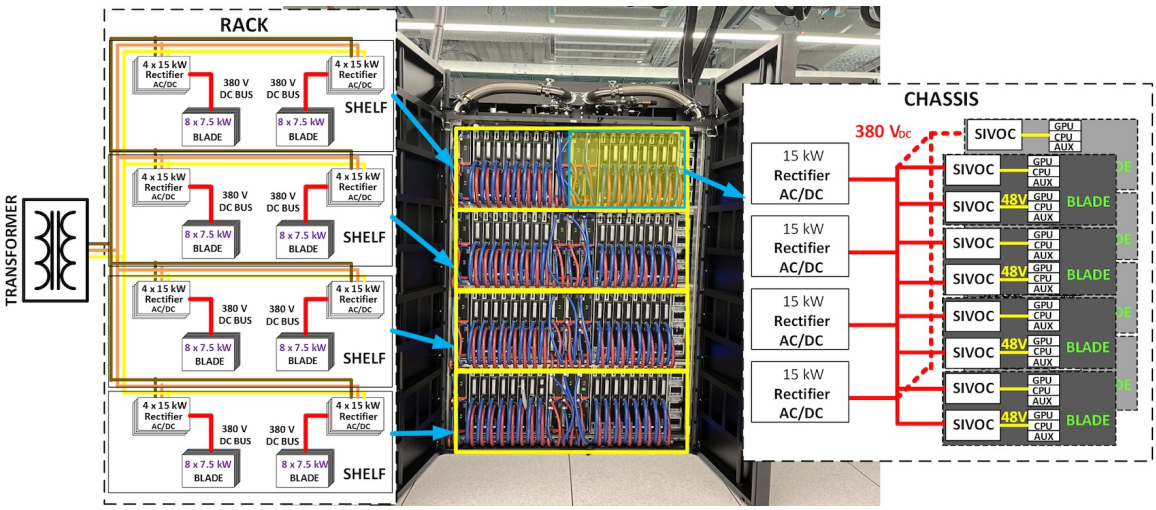


Fig. 3. Frontier rack-level power distribution and voltage conversion.

The [idle, peak] values for P_{CPU} and P_{GPU} respectively are [90, 280] and [88, 560] watts. We set $P_{RAM} = 74$W based on the mean RAM power, $P_{NIC} = 20$W (four per node), and we use $P_{NVMe} = 15$W (two per node). Every second in the simulation, P_{node} is computed for every node by linearly interpolating between [idle, peak] power values for the time-indexed value in the CPU/GPU utilization traces to get the P_{CPU} and P_{GPU} values. After computing P_{node}, rectification and conversion losses P_{L_R} and P_{L_S} are applied. Then, power is summed at the rack level, which includes the 250W of power [47] for each of the 32 network switches per rack:

$$P_{rack} = \sum_{i}^{N=128} P^{i}_{node} + 32P_{switch} \tag{4}$$

Next, the three racks associated with each cooling distribution unit (CDU) is summed, which gives 25 dynamic power outputs, corresponding to the 25 CDUs. The power output is multiplied by a cooling efficiency[4] of 0.945 before feeding it into the cooling model, which is described in Section III-C. To get the total system power, P_{system}, we sum these 25 power values and add the power cost to operate the pumps in each of the 25 CDUs, which is a constant 8.7 kW per CDU, or 217.5 kW in total. We calculate the system at peak utilization (9472 nodes, with CPUs and GPUs operating at full capacity) to consume 28.2 MW and show a breakdown of the individual power contributors in Fig. 4.

3) Synthetic Workloads: In order to model system workloads, we simply analyze system telemetry data to obtain average and standard deviations for quantities such as average job arrival time t_{avg}, number of nodes required, and wall time. Then it simply generates randomly distributed values for average CPU/GPU utilizations. We still have much work to do on the topic of "application fingerprinting" to develop more accurate models of jobs. This is an area where AI/ML can be useful for developing a job generator. One promising tool that can be used in this capacity is Kronos [48].

4) Modeling Job Arrival and Scheduling: RUNSIMULATION submits jobs to the queue according to a Poisson process [49], where an exponential distribution is used to model the time between job arrivals given by:

$$\tau = -\frac{\ln(1-U)}{\lambda} \tag{5}$$

Here, τ represents the time interval for the next job submission, determined by a uniformly distributed random variable U in the interval (0, 1); λ, defined as the inverse of the average arrival time (t_{avg}), is a configuration parameter that can be determined from telemetry data as $1/t_{avg}$, where t_{avg} denotes the average interval between job arrivals. Jobs are scheduled according to a given policy, such as Shortest Job First (SJF) or First Come First Served (FCFS), with plans to soon implement more sophisticated algorithms and evaluate their impact on the overall system.

5) Output Statistics: At the end of the run, a report is provided that outputs statistics on: (1) the number of jobs completed, (2) the throughput (jobs/hour), (3) average power consumed in MW, (4) total energy consumed in MW-hr, (5) rectification and conversion losses in MW (6) CO2 emissions in metric tons, and (7) total energy costs in USD. CO2 emissions are calculated by multiplying the average system power $\bar{P}_{system}$ by the following emissions factor from [50]:

$$E_f = E_I \times 1\,\text{metric ton}/2204.6\,\text{lbs} \times 1/\eta_{system} \tag{6}$$

where E_I represents the emission intensity in CO_2/MWh, currently set at 852.3; however, this value can vary regionally and even hourly [51].

6) Deployment: In order to make RAPS more accessible as a useful tool for the technical staff and HPC engineers, we deploy it on an internally hosted Kubernetes (K8s) cluster.

[4]Computed from telemetry data as heat removed divided by power consumed, discussed in Section IV-1 and plotted in Fig. 9.

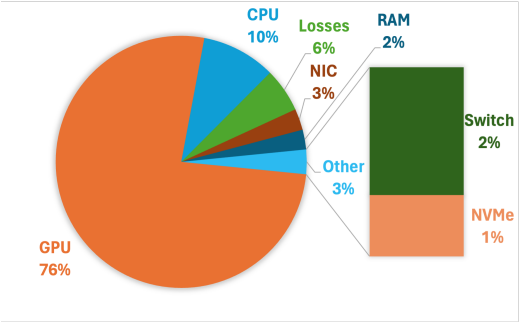


Fig. 4. Frontier power utilization breakdown based on peak CPU/GPU utilization of its 9472 nodes.

This allows us to host a dashboard interface, developed in TypeScript with ReactJS, shown in Fig. 6, for launching simulations to perform "what-if" scenarios and plot/analyze the results. Each case runs in a separate K8s pod, and the results of each experiment can be saved in the Apache Druid database and recalled later. This type of deployment enables users to easily perform a wide range of experiments and quickly plot the results. The dashboard also relies on a backend HTTP REST API for accessing telemetry data from the physical twin.

Finding 4. *There are multiple approaches to accurately modeling power, based on time granularity or level of detail. We found that using a coarse-grained, job-centric simulation – which traces CPU/GPU utilization and includes resource allocation and power loss modeling – provides good estimates of dynamic power for an exascale digital twin.*

C. Cooling Model

Having presented the RAPS module's ability to predict and manage power consumption, we now transition to modeling Frontier's liquid cooling system, designed to rapidly dissipate the vast amounts of energy lost as heat due to resistive loads caused by the many voltage converters, GPUs, CPUs, and DIMMs in the system.

1) Frontier's Cooling System: Fig. 5 shows a simplified layout of the overall cooling system for Frontier, which contains three cooling loops joined by heat exchangers. The *cooling tower loop* circulates through five cooling towers (CT), each with four cells, totaling 20 independent cells. The flow continues through four cooling tower water pumps (CTWP1-4) at approximately 9000-10000 gpm; it then passes through the five intermediate heat exchangers (EHX1-5). The *primary pump loop* flows from EHX1-5 through the four high temperature water pumps (HTWP1-4) at approximately 5000-6000 gpm. The flow then reaches the HEX-1600 heat exchangers, one per each of the 25 CDUs, which serve all 74 racks. In each of these 25 *CDU-rack loops*, flow passes through the HEX-1600 heat exchanger to the CDU pump and then the flow is split to serve three racks. In each rack, the flow passes through 64 compute blades (each with two nodes), through two CPU cold plates, and eight GPU cold plates. Therefore, each CDU cools a total of 192 blades, or 384 nodes.

Algorithm 1 RAPS Pseudocode

```
procedure MAIN
    Initialize scheduler
    Generate list of jobs to be executed
    Call RUNSIMULATION
end procedure
procedure RUNSIMULATION(jobs, timesteps)
    for each timestep do
        Add newly arriving jobs to pending queue
        Call SCHEDULEJOBS with pending jobs
        Call TICK
    end for
end procedure
procedure TICK                          ▷ Called every second
    Increment current time
    for each running job do
        if job is completed then
            Release nodes
            Update power state to idle
        end if
    end for
    Recalculate power consumption
    Apply rectification and conversion losses
    if timestep mod 15 = 0 then         ▷ Update every 15s
        Call FMU cooling model
        Update UI/Status
    end if
end procedure
procedure SCHEDULEJOBS(jobs)
    for each job in jobs do
        if enough nodes available then
            Assign nodes to job
            Update power state to active
        else
            Add job to pending queue
        end if
    end for
end procedure
```

2) Modelica: Rather than using a high-fidelity computational fluid dynamics (CFD) approach, which is typically used in air-cooled systems, we have opted to use Modelica – a system-level modeling language that offers a good balance between advanced predictive capability and simulation time. Modelica is an open-source, acausal, object-oriented programming language used for modeling complex cyber-physical systems [25]. The Modelica standard library can be easily extended using a variety of commercial and open-source packages. Both commercial and open-source integrated development environments (IDEs) are available to develop and run the Modelica model. One of the principal advantages of Modelica lies in its acausal, declarative style which affords a clean separation between the ordinary differential equations (ODEs) that are being coded by the user, and the solution procedure for the complex system of equations.

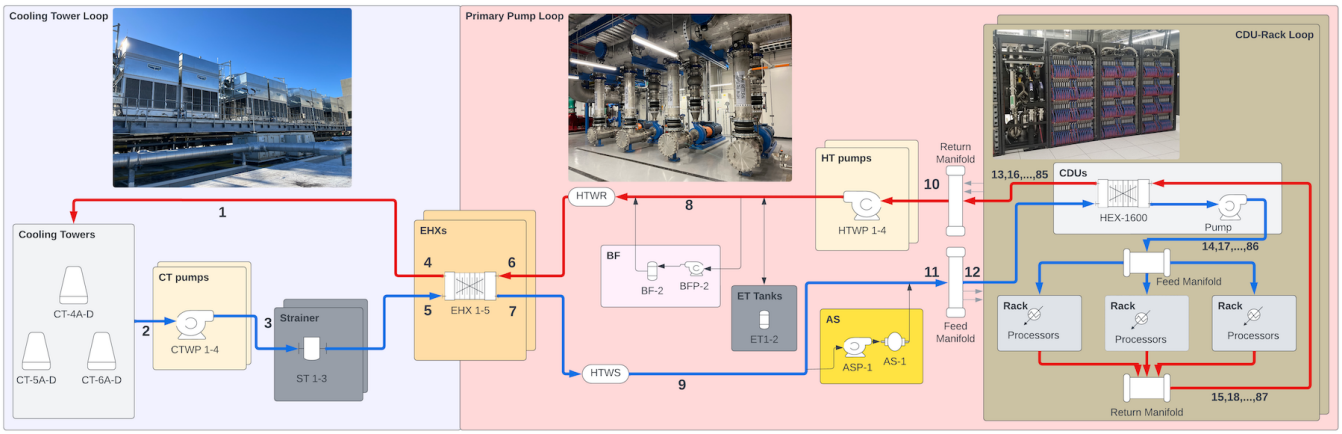


Fig. 5. Simplified schematic of Frontier cooling system with enumerated locations where the cooling model predicts pressures, temperatures, and flow rates.

Finding 5. *For modeling the data center cooling, while there are several commercially available tools, they offer limited extensibility while also being cost prohibitive (on the order of tens of thousands of dollars per year). Despite its steep learning curve, Modelica as an open-source framework provides the best value and path for meeting our current needs, as well as for building a vibrant digital twin community.*

3) Model Dependencies: The 'Modelica.Fluid' library, part of the Modelica Standard Library, solves zero-dimensional and one-dimensional thermo-fluid flow in a variety of fluidic components such as pipes, fluid machines such as pumps, vessels, valves, fittings, etc [52], [53]. The governing equations are formulated based on a finite volume method with a staggered grid scheme for momentum. The component equations and the media models are decoupled from each other in the 'Modelica.Fluid' library allowing the user to specify incompressible or compressible media, single-component or multi-component mixtures (including two-phases), as well as separate correlations for pressure drop and heat transfer coefficients [52], [53]. Different closure relations are available based on flow regime, i.e., laminar, turbulent and transition flow.

For the current study, the Transient Simulation Framework of Reconfigurable Models (TRANSFORM) library was utilized in conjunction with the Modelica buildings library (MBL). TRANSFORM is a Modelica-based open-source library to enable rapid development of dynamic, advanced energy systems with an extensible system modeling tool [26], [54]. On the other hand, MBL provides components for modeling building performance, including HVAC systems, and controls [55]. Specifically, the variable fan speed-based cooling tower model was used from MBL. All other components are either derived from or directly accessible within the TRANSFORM library.

4) Thermo-Fluids Model: All the sub-models shown in Fig. 5 are constructed with volumes (reservoirs) for mass sources, resistances for pressure drops, pumps, heat exchangers, and sensors, according to the templated layout described in [56]. The model takes as inputs wet-bulb (outdoor) temperature and heat extracted in watts for each of the 25 CDUs. The model produces a total of 317 outputs for each timestep of simulation (currently 15s), which is broken down as follows. For each of the 25 CDUs, there are 11 model outputs: work done by the CDU pump (station 14); primary and secondary flow rates (stations 12, 14 respectively); supply and return temperatures and pressures (stations 12-15). For the primary pump loop, the model outputs information related to the number of pumps and heat exchangers staged, and power consumption and pump speed for each of the four hot temperature water pumps (HTWPs). For the cooling tower loop, the model outputs the number of cooling towers (CTs) staged and power consumed by the four cold temperature water pumps (CTWPs), and the power consumed by the 16 CT fans. The PUE is computed by summing the total facility energy and dividing by P_{system}.

5) Control system model: In order to accurately model the cooling system and predict PUE, it was necessary to model Frontier's control system, which enables the staging of cooling towers, heat exchangers, and pumps. The control system logic is divided between the CEP and the supercomputer. A detailed overview of the control system logic is beyond the scope of this paper; therefore, we provide a concise explanation here.[5] The Modelica model captures the essentials of the control logic, which activates once the physical cooling system begins auto-operation, after the start-up sequence is complete. The model is described for each cooling loop here:

- *CDU-rack loop*: A PID controller is used to regulate the CDU relative percent pump speeds based on the loop differential pressure, and a control value is used to regulate the primary coolant flow based on a set secondary supply temperature. Most of the PID parameters have been taken from the physical controller where available, and tuned using telemetry data where parameters were not available. Both the CDU pumps are assumed to be in operation at all

[5]See Kumar et al. [57] for a detailed description of the cooling model.

times with the same speeds – a reasonable approximation as evidenced by telemetry data.

- *Primary pump loop*: A PID controller is used to regulate the four HTWPs. The HTWPs are staged up/down depending on the relative percent pump speeds of the running pumps. The intermediate heat exchangers (EHXs) are staged based on the number of CTs in operation.
- *Cooling tower loop*: The CTWP speed is regulated based on the CT supply header pressure, which is maintained within a given pressure range; this informs the staging up/down of the four CTWPs in concert with the relative percent speeds of the running pumps. The CTs are staged up/down based on header pressure and the gradient of the hot temperature water supply (HTWS) temperature.

Therefore, the criteria to achieve HTWS temperature stability informs both the staging of the CTs directly and the EHXs indirectly. This non-linearity is handled in the model via a delay transfer function between the primary pump loop, which requires the number of CTs, and the cooling tower loop, which requires the HTWS temperature. Therefore, any disturbance in a CDU would affect its control valve, which in-turn would affect the overall system to respond as just described. Future efforts will focus on optimizing the control parameters to achieve better system stability such as responding quickly to a surge in power to the CDU.

6) Exporting the Cooling Model: Using the Functional Mock-up Interface (FMI) standard [58], the cooling model is integrated into the digital twin framework as a Functional Mock-up Unit (FMU). An FMU is a model which has been wrapped in the standard FMI interface and can therefore be used in any software or deployment scenario which has implemented the FMI. In this effort, the Dymola IDE was used to both develop and export the system model as an FMU. The cooling model FMU is then imported into the RAPS module via the FMPy Python module, which is queried from the visual analytics module via FastAPI [59].

Finding 6. *Using a simplified system-level Modelica model yields good transient solutions. Any additional improvement in fidelity would need to be weighed against model complexity which affects model convergence and simulation time.*

D. Visual Analytics

After discussing the backend modeling for resource allocation, power consumption, and thermo-fluidic cooling, we discuss how to interface with the models to visualize insights. For users, it is important to have effective ways to interact with the DT. This enables the realization of value offered by the combined system, either by providing a good overview of what is happening within the system, or by exploring details that are otherwise only accessible with major efforts. Visual analytics in the context of digital twins helps users to interact effectively with the DT to discover new insights [60]. We primarily explored *augmented reality (AR)* as a visual analytics tool, but also experimented with the RAPS dashboard interface, as discussed in Section III-B6, for launching simulations and for plotting statistics.

Representing the physical asset in a 3D virtual space simplifies spatial information correlation more intuitively than using more complex data analysis techniques. With this in mind, we implemented a virtual representation of Frontier and the CEP in UE5[6] as shown in Figure 6, as viewed through a Microsoft HoloLens 2 AR headset. The visualization allows for system interaction, 3D navigation, and the querying of simulation and telemetry data. Users can run this from their desktops or laptops, or as an AR application. Also, in AR it is possible to overlay the information onto the physical system, gaining information and insights that are otherwise not realizable. An interactive or programmable level of detail was the key to make our UE5 model performant and responsive with the large number of components and associated telemetry present in the system.

Finding 7. *For a complex system such as Frontier, with its volume, variety, and velocity (3Vs) of data, we found that augmented reality coupled with dashboards is one of the most effective ways to provide timely visual insights into the system. This allows us to seamlessly navigate the system complexity and its simulations, both in space and time, from system overview to blade and component-level views.*

IV. Verification, Validation, & Functional Tests

The National Academy of Sciences, Engineering, and Medicine (NASEM) recently published a report of findings and recommendations for digital twins [62]. One of their key recommendations was to deeply embed verification, validation, and uncertainty quantification (VVUQ) into the development of digital twins. Following such advice, we prioritized extensive V&V of our power and cooling models after the initial development phase, and also have implemented UQ into our RAPS module. We see *verification* as running some basic tests to see if the model is performing as expected, whereas *validation* involves comparing the models with more extensive system telemetry data. In this section, we discuss V&V related to the power and cooling of our DT, while also demonstrating telemetry replay for multiple historical scenarios.

Table II shows the telemetry data used to validate the digital twin. This set of telemetry data closely follows what Shin et al. [7] collected and analyzed from the Summit supercomputer, due to the proximity of the operational use cases our work aims to address.

Finding 8. *We observe that one of the most effective ways to perform verification and validation studies of the power and cooling models is by replaying system telemetry at multiple levels through the digital twin.*

1) Cooling model V&V: A validation study of the entire cooling model, as shown in Fig. 7, was conducted using $\sim$24 hours of 2024-04-07 telemetry data from the CEP and the datacenter down to the level of the CDU. The only inputs to the model is the power supplied to the 25 CDUs (and associated three racks per CDU) and the wet-bulb (outside) temperature.

[6]See Maiterth et al. [61] for a detailed description of the augmented reality model.

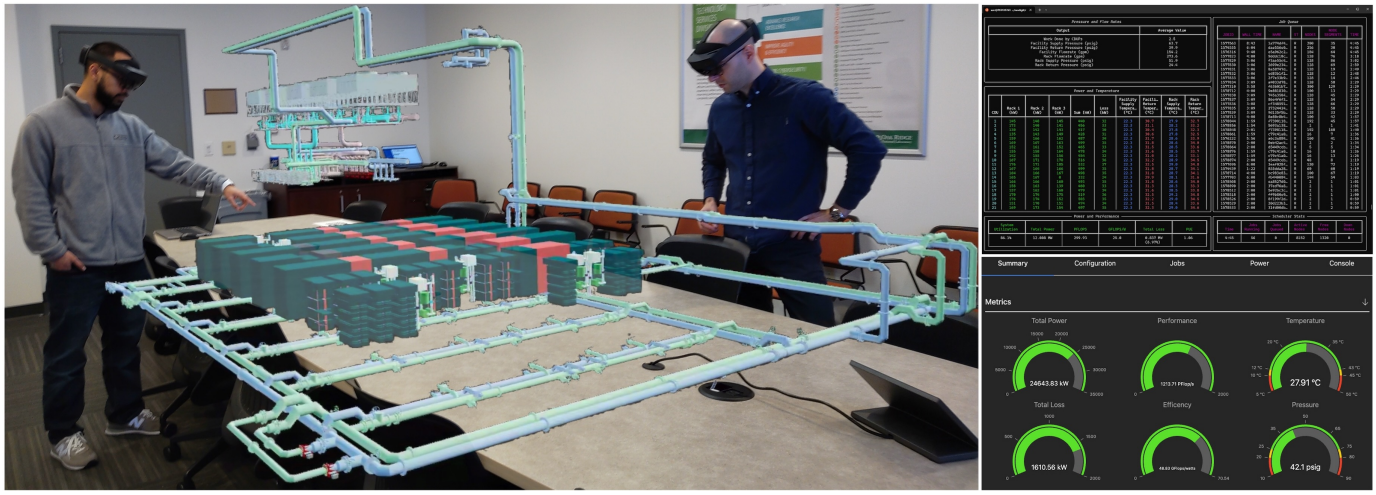


Fig. 6. ExaDigiT interfaces: AR model as projected in meeting room (left), terminal interface (top-right), and web-based dashboard (bottom-right).

TABLE II
SPECIFICATION OF TELEMETRY DATA USED FOR VALIDATION

RAPS	**Model Schema (Resolution, Length)**
Inputs	jobs: List[Dict]: job_name: str job_id: int node_count: int start_time: float cpu_power: List[float] (15s, variable length) gpu_power: List[float] (15s, variable length)
Output	measured_power: List[float] (1s)

Cooling Model	**Model Schema (Resolution, Length)**
Inputs	rack_power: List[float] (15s, 25) wetbulb_temperature: float (60s)
Outputs (CDU)	{htw,ctw}_flow_rates: List[float] (15s, 25) cdu_temps: List[float] (15s, 25) cdu_pump_speeds: List[float] (15s, 25) cdu_pump_power: List[float] (15s, 25)
Outputs (CEP)	facility_flow_rates: List[float] (2m, 2) {supply,return}_temps: List[float] (1m∼10m, 2) {supply,return}_pressures: List[float] (30s∼10m, 2) {htwp,ctwp}_pump_power: List[float] (10m, 4) {htwp,ctwp}_pump_speed: float (2m) num_{ctwp,htwp,ehx,ct}_staging: int (variable) pue: float (15s interpolated)

The power supplied to the CDUs for the validation exercise was calculated in terms of heat removed by the cooling water:

$$H = \rho \cdot Q \cdot \Delta T \cdot c \tag{7}$$

where H is the extracted heat measured in watts, ρ is the density of water in kg/m^3, Q is the flow rate in the CDU-rack loop in m^3/s, ΔT is the temperature differential caused by the heat exchanger in °C and c is the specific heat capacity of water in $\mathrm{J}/(\mathrm{kg} \cdot {}^\circ\mathrm{C})$.

Comparing model predictions with telemetry data reveals several insights. For most parameters, such as those in Fig. 7, the model performs well and is able to predict the response of the physical cooling system to the change in the compute load. Overall, both the root mean square error (RMSE) and the mean absolute error (MAE) of the parameters shown in Fig. 7 are within reasonable bounds. The model predictions for the CDU secondary supply temperatures, although not displayed, exhibit greater fluctuation than the physical system, which does a good job maintaining the temperature at the setpoint. This discrepancy will require further investigation.

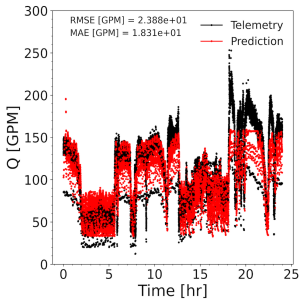
The comparison between the PUE predicted by the model and that calculated from telemetry data are shown in Fig. 7(d). The model-predicted PUE is within 1.4 percent of the telemetry-based PUE for the range of data tested. Calculations for both the model-predicted and telemetry-based PUEs are based on power consumption P_{AUX} from the following auxiliary systems: CDU pumps, HTWPs, CTWPs and CT fans. Low-power auxiliary systems, such as air-handling systems, were not modeled or included in the PUE calculations.

2) RAPS V&V: We performed initial verification of RAPS by predicting idle and peak power, and also included a High Performance Linpack (HPL) [63] benchmark. To set the system at idle, we simply set CPU and GPU utilizations to zero on all nodes; to test the HPL core phase, we set all four GPUs on each node to 79% utilization and the CPU to 33% utilization (inferred from telemetry data); to test peak power, we set all CPU and GPU utilizations to 100%. Table III shows the results of this study. Fig. 8 shows synthetic benchmark tests including HPL and OpenMxP [64] in conjunction with the cooling system predictions of primary return temperature.

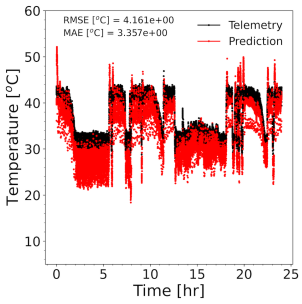
3) Functional Tests: After verifying the model could predict idle, HPL core phase, and peak power well, we used RAPS to replay 183 days of telemetry data from Frontier, from 2023-09-06 to 2024-03-18. Results are summarized in Table

TABLE III
RAPS POWER VERIFICATION TESTS

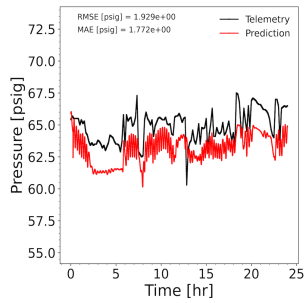
Tests	**Nodes**	**Telemetry (MW)**	**RAPS (MW)**	**% Error**
Idle power	9472	7.4	7.24	2.1%
HPL (core)	9216	21.3	22.3	4.7%
Peak power	9472	27.4	28.2	3.1%



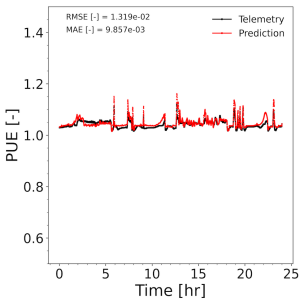
(a) Primary CDUs flow rate predictions (Station 12 in Fig. 5)



(b) Primary CDUs return temp. predictions (Station 12 in Fig. 5)



(c) HTW supply pressure predictions (Station 10 in Fig. 5)



(d) PUE Modelica model predictions

Fig. 7. Cooling model validation tests. Modelica model predictions (exported as an FMU) vs. telemetry data for the CDU and the CEP.

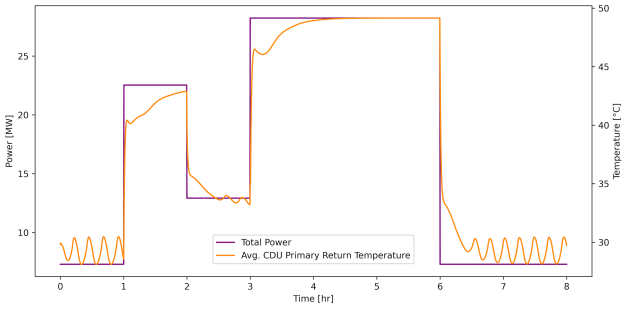


Fig. 8. Synthetic benchmark verification test. Total system power predicted by RAPS and the transient temperature response predicted by the cooling model.

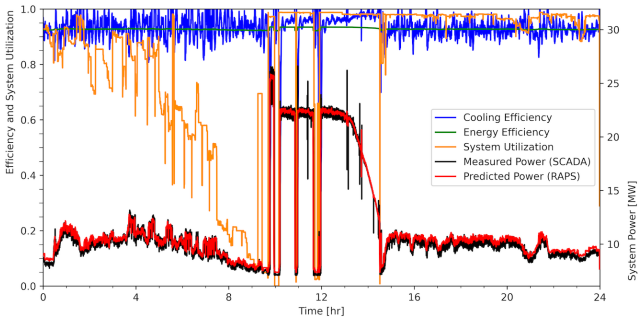


Fig. 9. Telemetry replay validation test of 24-hour period on 2024-01-18 for Frontier containing an HPL run.

IV. Each 24-hour replay takes about nine minutes to run with cooling, or just three minutes without; the entire analysis takes about an hour when running the different days in parallel on a single Frontier node. Average conversion losses amount to 1.14 MW (6.74%) which works out to about $900k per year.

Fig. 9 shows a 24-hour period replay, which includes 1238 jobs in total, 400 of which are single-node jobs, and four back-to-back HPL 9216-node jobs, among others. This plot illustrates the instantaneous system power, P_{system}, with both predicted values (in red) and measured values (in black); the energy efficiency, η_{system} (green), as defined in (1); the cooling efficiency (blue), defined as $\eta_{cooling} = H/P_{system}$; and the system utilization (orange), calculated as the ratio of active nodes to the total available nodes.

Now we can begin to envision ways to improve overall efficiency through virtual modifications to Frontier's DT. The first idea we tested was modifying Frontier to use "smart load-sharing rectifiers". The rectifiers reach an optimal efficiency of 96.3% at 7.5 kW, but near idle the efficiency drops 1-2%. Instead of sharing the chassis load across all four rectifiers, rectifiers are dynamically staged on as needed, so that rectifiers are always operating at their peak efficiency regions. While this modification yielded only a modest efficiency gain of 0.1%, it translates into a not so insignificant yearly cost savings of approximately $120k, based on the same 183 days of data in Table IV. A second test, inspired by [65] and [66], focused on switching the Frontier DT to direct 380V DC power, instead of AC power. This modification substantially increased the system efficiency from 93.3% to 97.3%, a potential savings of $542k per year, while also reducing the carbon footprint by 8.2%.

Finding 9. *Considerable losses are incurred during both AC-DC rectification and DC-DC voltage conversion. Frontier exhibits an average conversion loss of 1.1 MW and maximum of 1.8 MW. Such losses must be accounted for in digital twins, especially when using such a tool to study energy efficiency.*

V. Generalizing ExaDigiT

While we initially developed ExaDigiT to model Frontier, we later worked on generalizing the framework to be able to support development of a variety of architectures. To this end, we determined to use a number of JSON files for input specification, to minimize the level of code changes that must be made to model a particular system. For example, the generalized version of RAPS inputs configuration files describing the system architecture, the cooling system, the scheduler, and the power system. A pluggable architecture was developed for reading different types of bespoke telemetry datasets. We are still actively working to extend RAPS to handle multi-partition systems, such as Setonix, which have separate partitions for CPU-only nodes and CPU+GPU nodes. A secondary challenge for generalization is addressing shared node configurations, where each node may be shared by multiple users. This generalized approach has recently been used by others [67] to model Italy's Marconi100 supercomputer, along with an associated PM100 open telemetry dataset [68].

TABLE IV
DAILY STATISTICS OF DT FROM TELEMETRY REPLAY OF 183 DAYS.

Parameter	Min	Avg	Max	Std
Avg Arrival Rate, $t_{avg}(s)$	17	138	2988	331
Avg Nodes per Job	39	268	5441	626
Avg Runtime (m)	17	39	101	14
Jobs Completed	32	1575	5157	1171
Throughput (jobs/hr)	1.3	66	215	49
Avg Power (MW)	10.2	16.9	23.0	2.4
Loss (MW)	0.52	1.14	1.84	0.15
Loss (%)	6.26	6.74	8.36	0.11
Total Energy Consumed (MW-hr)	129	405	553	64
Carbon Emissions (tons CO_2)	53	168	229	26

For the thermo-fluids model, an automated cooling system model (AutoCSM) method was developed that automates much of the process of developing cooling systems for digital twins [41]. AutoCSM, based on Python, inputs a JSON input specification of the architecture of the system, and outputs an initial model of the system, which can then be compiled into an FMU. Currently, AutoCSM outputs Modelica code, but in the future can be extended to support other system modeling languages such as JuliaSim [69]. We have plans to use the AutoCSM approach to model Marconi100's cooling system [70].

Our augmented reality model has been used by others to develop AR models for both Finland's LUMI and Australia's Setonix supercomputers [71]. To simplify the development process, we also plan to extend the code to support dynamic asset generation based on JSON configuration files. Furthermore, we plan on using an OpenXR based implementation to be able to support a variety of head-mounted displays, such as MetaQuest and Apple Vision Pro.

VI. CONCLUSIONS & FUTURE WORK

We presented the development of a comprehensive digital twin framework for modeling liquid-cooled supercomputers called ExaDigiT,[7] comprising three main modules: a Resource Allocator and Power Simulator (RAPS), a thermo-fluidic cooling model, and an AR-based visual analytics module. An extensive amount of work went into modeling the various aspects of the supercomputer and central energy plant. ExaDigiT's development revealed several important insights: (1) simulations provide the foundational building blocks DTs, with machine learning being important for workload characterization; (2) in modeling power and cooling, it was important to strike a balance between fidelity and complexity, achieved through job-centric power simulations and system-level thermo-fluid simulations; (3) augmented reality is an effective means of interactively visualizing the complexity of the digital twin; (4) system telemetry replay is vital for model verification and validation to evaluate the trustworthiness of DTs; (5) modeling the up to 1.8 MW of power losses due to rectification and voltage conversion will be key to using the DT for energy efficiency studies.

[7]Available at https://exadigit.github.io.

By considering a comprehensive model of the entire supercomputer, we can study complex cross-disciplinary transient behaviors to provide insights into operational strategies, "what-if" scenarios, system diagnostics, as well as serving as a design tool for virtual prototyping. The development of an open-source framework for comprehensive modeling of digital twins has significant implications in multiple areas, particularly for the design of future energy-efficient supercomputers. There has been considerable interest among supercomputing centers around the world in both using and contributing to our ExaDigiT framework. To accommodate the growing interest in such a tool, we have organized an ExaDigiT community, which currently has 93 global participants. This group meets monthly with workgroups meeting regularly to advance the framework's development, use case studies, and its application across various supercomputer architectures.

While the digital twin framework presented in this paper has focused on asset visualization (L1), telemetry (L2), and modeling and simulation (L4), we aim to focus future efforts on developing data-driven models (L3), such as machine-learned models of workloads, and reinforcement learning agents (L5) that provide continuous feedback to the physical twin. Such agents will enable the creation of a "living" DT [72], [73] that operates alongside the physical twin. We hope our efforts will stimulate and enable more energy-efficient supercomputers in the future, by providing a powerful, comprehensive framework along with fostering a global community of researchers to work together toward building a strong future for sustainable energy-efficient supercomputing. In the long term, investigating multi-scale approaches to bridge discrete event simulators, such as Gem5 [74], SST/Macro [75], and SimGrid [76], with comprehensive digital twins for end-to-end optimization will be key for advancing the state of the art [77].

ACKNOWLEDGMENTS

This research was sponsored by and used resources of the Oak Ridge Leadership Computing Facility (OLCF), which is a DOE Office of Science User Facility at the Oak Ridge National Laboratory (ORNL) supported by the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. We thank the technical staff at ORNL, without whom this work would not have been possible, including: Scott Atchley, Matt Sieger, Chris Zimmer, Paul Abston, Jim Rogers, Matt Ezell, Robert Gillen, Dane de Wet, Kazi Asifuzzaman, Nathan Parkison, Cory Spradlin, Brian Reagan, John Holmen, Amir Shehata, Nick Hagerty, Seung-Hwan Lim, Ahmad Maroof Karimi. From HPE, we would like to thank Cullen Bash, Tim Dykes, Matt Slaby, and Justin Queen who provided us with invaluable help along the way. Thanks to Jake Webb from Cadre5, LLC for his significant contributions to the dashboard development. Moreover, we want to acknowledge our growing ExaDigiT open source community for their enthusiastic support and engagement, which has been a significant source of inspiration and motivation for this work. Finally, ChatGPT was utilized for converting Python code into pseudocode in Algorithm 1, grammar enhancements, and assisting with table formatting.

REFERENCES

[1] K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, K. Hill, J. Hiller *et al.*, "Exascale computing study: Technology challenges in achieving exascale systems," *Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep*, vol. 15, p. 181, 2008.

[2] S. Atchley, C. Zimmer, J. Lange, D. Bernholdt, V. Melesse Vergara, T. Beck, M. Brim, R. Budiardja, S. Chandrasekaran, M. Eisenbach *et al.*, "Frontier: Exploring exascale," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2023, pp. 1–16.

[3] E. Strohmaier, J. Dongarra, H. Simon, and M. Meuer, "Top500 List - June 2022," https://top500.org/lists/top500/2022/06/, 2022, [Online; accessed 20-March-2024].

[4] ——, "Top500 List - June 2023," https://top500.org/lists/top500/2023/11/, 2023, [Online; accessed 5-March-2024].

[5] ——, "Top500 List - June 2024," https://top500.org/lists/top500/2024/06/, 2024, [Online; accessed 13-August-2024].

[6] W. Zuo, M. Wetter, J. VanGilder, X. Han, Y. Fu, C. Faulkner, J. Hu, W. Tian, and M. Condor, "Improving data center energy efficiency through end-to-end cooling modeling and optimization. final report," Univ. of Colorado, Boulder, CO (United States), Tech. Rep., 2021.

[7] W. Shin, V. Oles, A. M. Karimi, J. A. Ellis, and F. Wang, "Revealing power, energy and thermal dynamics of a 200PF pre-exascale supercomputer," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–14.

[8] A. Haidar, H. Bayraktar, S. Tomov, J. Dongarra, and N. J. Higham, "Mixed-precision iterative refinement using tensor cores on GPUs to accelerate solution of linear systems," *Proceedings of the Royal Society A*, vol. 476, no. 2243, p. 20200110, 2020.

[9] L. Su and S. Naffziger, "Innovation for the next decade of compute efficiency," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2023, pp. 8–12.

[10] AIAA Digital Engineering Integration Committee, "Digital twin: definition & value," *AIAA and AIA Position Paper*, 2020.

[11] A. Glaser, "19 ways digital twins improve data center sustainability," 2023. [Online]. Available: https://venturebeat.com/ai/19-ways-digital-twins-improve-data-center-sustainability/

[12] B. Violino, "Digital twins are set for rapid adoption in 2023," *CNBC Technology Executive Council*, January 2023, accessed: 2024-08-21. [Online]. Available: https://www.cnbc.com/2023/01/21/digital-twins-are-set-for-rapid-adoption-in-2023.html

[13] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 732–794, 2015.

[14] C. Jin, X. Bai, C. Yang, W. Mao, and X. Xu, "A review of power consumption models of servers in data centers," *applied energy*, vol. 265, p. 114806, 2020.

[15] A. Sîrbu and O. Babaoglu, "A data-driven approach to modeling power consumption for a hybrid supercomputer," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 9, p. e4410, 2018.

[16] K.-P. Lee and H.-L. Chen, "Analysis of energy saving potential of air-side free cooling for data centers in worldwide climate zones," *Energy and Buildings*, vol. 64, pp. 103–112, 2013.

[17] S.-W. Ham and J.-W. Jeong, "Impact of aisle containment on energy performance of a data center when using an integrated water-side economizer," *Applied Thermal Engineering*, vol. 105, pp. 372–384, 2016.

[18] Y. Fu, M. Wetter, and W. Zuo, "Modelica models for data center cooling systems," Univ. of Colorado, Boulder, CO (United States), Tech. Rep., 2018.

[19] T. Zohdi, "A digital-twin and machine-learning framework for precise heat and energy management of data-centers," *Computational Mechanics*, vol. 69, no. 6, pp. 1501–1516, 2022.

[20] Z. Zhang, Y. Zeng, H. Liu, C. Zhao, F. Wang, and Y. Chen, "Smart DC: an AI and digital twin-based energy-saving solution for data centers," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–6.

[21] Cadence, "Datacenter design software," Available at https://www.cadence.com/en_US/home/resources/datasheets/datacenter-design-software-ds.html, February 2024, accessed: 2024-03-12.

[22] A. Heydari, P. Shahi, V. Radmard, B. Eslami, U. Chowdhury, S. Saini, P. Bansode, H. Miyamura, D. Agonafer, and J. Rodriguez, "Liquid to liquid cooling for high heat density liquid cooled data centers," in *International Electronic Packaging Technical Conference and Exhibition*, vol. 86557. American Society of Mechanical Engineers, 2022, p. V001T01A007.

[23] Innovative Research LLC, "MacroFlow: A software tool for rapid flow and thermal design of electronics cooling, semiconductor processing, and general flow systems," Available at https://inresllc.com/macroflow-overview.html, February 2024, accessed: 2024-03-12.

[24] Cadence Design Systems, "Cadence Celsius EC Solver," https://www.cadence.com/en_US/home/tools/system-analysis/thermal-solutions/celsius-ec-solver.html, accessed: 2023-03-12.

[25] M. Association. (2020, 06) Modelica standard library v4.0.0 (2020-06-04). [Online]. Available: https://github.com/modelica/ModelicaStandardLibrary/releases/tag/v4.0.0

[26] M. S. Greenwood, "TRANSFORM - TRANsient Simulation Framework of Reconfigurable Models," Computer Software, Sep. 2017, accessed on August 30, 2023. [Online]. Available: https://github.com/ORNL-Modelica/TRANSFORM-Library

[27] E. Rosenthal. (2018) Augmented reality technology shows promise for summit surveillance. Accessed: 2023-10-23. [Online]. Available: https://www.olcf.ornl.gov/2018/11/05/augmented-reality-technology-shows-promise-for-summit-surveillance/

[28] L. Riha, K. Sethia, K. Kadlubiak, M. Hrabankova, M. Spetko, M. Jaros, O. Vysocky, P. Svobodova, P. Strakos, and T. Panoc, "HPCMonitor4B: Tool for monitoring Blender jobs on HPC clusters," 2024, gNU General Public License v3.0. [Online]. Available: https://code.it4i.cz/blender/hpcmonitor4b

[29] B. Bergeron, M. Hubbell, D. Sequeira, W. Williams, W. Arcand, D. Bestor, C. Byun, V. Gadepally, M. Houle, M. Jones *et al.*, "3D real-time supercomputer monitoring," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2021, pp. 1–7.

[30] V. Averbukh, A. Bersenev, M. Forghani, A. Igumnov, D. Manakov, A. Popel, S. Sharf, and P. Vasev, "Visualizing a supercomputer: A case of objects regrouping," in *CEUR Workshop Proceedings*, vol. 2485. CEUR-WS, 2019, pp. 74–76.

[31] C. Zhou, K. L. Summers, and T. P. Caudell, "Graph visualization for the analysis of the structure and dynamics of extreme-scale supercomputers," in *Proceedings of the 2003 ACM symposium on Software visualization*, 2003, pp. 143–149.

[32] T. Fujiwara, J. K. Li, M. Mubarak, C. Ross, C. D. Carothers, R. B. Ross, and K.-L. Ma, "A visual analytics system for optimizing the performance of large-scale networks in supercomputing systems," *Visual Informatics*, vol. 2, no. 1, pp. 98–110, 2018.

[33] A. Bhatele, N. Jain, Y. Livnat, V. Pascucci, and P.-T. Bremer, "Analyzing network health and congestion in dragonfly-based supercomputers," in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2016, pp. 93–102.

[34] NVIDIA, "Data center digital twins," Blog post, 2023, available online: https://docs.omniverse.nvidia.com/digital-twins/latest/data-center.html.

[35] O. Hennigh, S. Narasimhan, M. A. Nabian, A. Subramaniam, K. Tangsali, Z. Fang, M. Rietmann, W. Byeon, and S. Choudhry, "NVIDIA SimNet™: An AI-accelerated multi-physics simulation framework," in *International conference on computational science*. Springer, 2021, pp. 447–461.

[36] Autodesk, "What is digital twin technology? and what are the benefits?" 2023. [Online]. Available: https://www.autodesk.com/solutions/digital-twin

[37] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annual review of fluid mechanics*, vol. 52, no. 1, pp. 477–508, 2020.

[38] X. Zhao, B. Maldonado Puente, S. Liu, S.-H. Lim, W. Gurecky, D. Lu, M. Howell, F. Liu, W. Williams, and P. Ramuhalli, "Knowledge-informed uncertainty-aware machine learning for time series forecasting of dynamical engineered systems," Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), Tech. Rep., 2023.

[39] R. Wang, D. Xia, Z. Cao, Y. Wen, R. Tan, and X. Zhou, "Toward data center digital twins via knowledge-based model calibration and reduction," *ACM Transactions on Modeling and Computer Simulation*, vol. 33, no. 4, pp. 1–24, 2023.

[40] A. Todd, A. Purkayastha, H. Egan, D. Sickinger, M. Eash, S. Serebryakov, J. Hanson, M. Slaby, N. Wunder, N. Guba *et al.*, "Artificial

intelligence for data center operations (AIOps)," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2021.

[41] S. Greenwood, V. Kumar, and W. Brewer, "Thermo-fluid modeling framework for supercomputing digital twins: Part 2, automated cooling models," in *Proceedings of the American Modelica Conference*. University of Connecticut: Modelica Association, October 14-16 2024.

[42] R. E. Gillen, L. A. Anderson, C. Craig, J. Johnson, R. Anderson, A. Craig, and S. L. Scott, "Design and implementation of full-scale industrial control system test bed for assessing cyber-security defenses," in *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2020, pp. 341–346.

[43] M. M. Salim, D. Camacho, and J. H. Park, "Digital twin and federated learning enabled cyberthreat detection system for IoT networks," *Future Generation Computer Systems*, 2024.

[44] J. K. Holmen, M. N. Newaz, S. Yoginath, M. Maiterth, A. Shehata, N. Hagerty, C. Zimmer, and W. Brewer, "Towards the development of an exascale network digital twin," in *Proceedings of the Cray User Group 2024 Conference*, Perth, Australia, May 2024.

[45] HPE, "HPE Cray - AMD EPYC NVIDIA SXM4 - Replace a SIVOC," April 2021. [Online]. Available: https://support.hpe.com/hpesc/public/videoDisplay?videoId=vpsg00004472en_us

[46] R. Wojda, M. Maiterth, and W. Brewer, "Dynamic modeling of power conversion stages for an exascale supercomputer," in *Proceedings of the IEEE Energy Conversion Congress & Exposition (ECCE)*, October 2024.

[47] D. De Sensi, S. Di Girolamo, K. H. McMahon, D. Roweth, and T. Hoefler, "An in-depth analysis of the Slingshot interconnect," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020, pp. 1–14.

[48] ECMWF, "Kronos," https://github.com/ecmwf/kronos, 2023, [Online; accessed 5-March-2024].

[49] Y. Fan, Z. Lan, T. Childers, P. Rich, W. Allcock, and M. E. Papka, "Deep reinforcement agent for scheduling in HPC," in *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2021, pp. 807–816.

[50] Environmental Protection Agency, "Greenhouse gases equivalencies calculator - calculations and references," https://www.epa.gov/energy/greenhouse-gases-equivalencies-calculator-calculations-and-references, 2023, accessed: 2024-02-26.

[51] B. Li, R. Basu Roy, D. Wang, S. Samsi, V. Gadepally, and D. Tiwari, "Toward sustainable HPC: Carbon footprint estimation and environmental implications of HPC systems," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2023, pp. 1–15.

[52] F. Casella, M. Otter, K. Proelss, C. Richter, and H. Tummescheit, "The Modelica fluid and media library for modeling of incompressible and compressible thermo-fluid pipe networks," in *Proceedings of the 5th international modelica conference*, 2006, pp. 631–640.

[53] R. Franke, F. Casella, M. Sielemann, K. Proelss, M. Otter, and M. Wetter, "Standardization of thermo-fluid modeling in Modelica.Fluid," 9 2009. [Online]. Available: https://www.osti.gov/biblio/988180

[54] M. S. Greenwood, B. R. Betzler, A. L. Qualls, J. Yoo, and C. Rabiti, "Demonstration of the advanced dynamic system modeling tool transform in a molten salt reactor application via a model of the molten salt demonstration reactor," *Nuclear Technology*, vol. 206, no. 3, pp. 478–504, 2020.

[55] M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang, "Modelica buildings library," *Journal of Building Performance Simulation*, vol. 7, no. 4, pp. 253–270, 2014.

[56] M. S. Greenwood, S. M. Cetiner, T. J. Harrison, and D. Fugate, "A templated approach for multi-physics modeling of hybrid energy systems in Modelica," 8 2017. [Online]. Available: https://www.osti.gov/biblio/1427611

[57] V. Kumar, S. Greenwood, W. Brewer, D. Grant, N. Parkison, and W. Williams, "Thermo-fluid modeling framework for supercomputing digital twins: Part 1, demonstration at exascale," in *Proceedings of the American Modelica Conference*. University of Connecticut: Modelica Association, October 14-16 2024.

[58] M. Association, "Functional mock-up interface," 2023. [Online]. Available: https://fmi-standard.org/

[59] S. Ramírez, "FastAPI framework," https://github.com/tiangolo/fastapi, 2024, accessed: 2024-03-29.

[60] H. Zheng, T. Liu, J. Liu, and J. Bao, "Visual analytics for digital twins: a conceptual framework and case study," *Journal of Intelligent Manufacturing*, pp. 1–16, 2023.

[61] M. Maiterth, W. Brewer, D. D. Wet, S. Greenwood, V. Kumar, J. Hines, S. Bouknight, Z. Wang, T. Dykes, and F. Wang, "Visualizing an exascale data center digital twin: Considerations, challenges and opportunities," in *Proceedings of the IEEE Visualization & Visualization Conference (VIS2024)*. IEEE, October 13-18 2024.

[62] National Academies of Sciences, Engineering, and Medicine, *Foundational Research Gaps and Future Directions for Digital Twins*. Washington, DC: The National Academies Press, 2023. [Online]. Available: https://doi.org/10.17226/26894

[63] J. J. Dongarra, P. Luszczek, and A. Petitet, "The LINPACK benchmark: past, present and future," *Concurrency and Computation: practice and experience*, vol. 15, no. 9, pp. 803–820, 2003.

[64] H. Lu, M. Matheson, F. Wang, W. Joubert, A. Ellis, and V. Oles, "OpenMxP – opensource mixed precision computing," Oak Ridge National Lab (ORNL), Oak Ridge, TN (United States), Tech. Rep., 2023.

[65] M. Ton, B. Fortenbery, and W. Tschudi, "DC power for improved data center efficiency," *Lawrence Berkeley National Laboratory*, vol. 10, 2008.

[66] J. Salazar, "New Hikari supercomputer starts solar HVDC," *HPCwire*, 2016, accessed: 01-April-2024. [Online]. Available: https://www.hpcwire.com/off-the-wire/new-hikari-supercomputer-starts-solar-hvdc/

[67] R. Kabir, "Personal communication," August 2024.

[68] F. Antici, M. Seyedkazemi Ardebili, A. Bartolini, and Z. Kiziltan, "PM100: A job power consumption dataset of a large-scale production HPC system," in *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, 2023, pp. 1812–1819.

[69] C. Rackauckas, M. Gwozdz, A. Jain, Y. Ma, F. Martinuzzi, U. Rajput, E. Saba, V. B. Shah, R. Anantharaman, A. Edelman *et al.*, "Composing modeling and simulation with machine learning in Julia," in *2022 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE, 2022, pp. 1–17.

[70] M. S. Ardebili, A. Bartolini, A. Acquaviva, and L. Benini, "Rule-based thermal anomaly detection for tier-0 HPC systems," in *International Conference on High Performance Computing*. Springer, 2022, pp. 262–276.

[71] T. Dykes, "Personal communication," August 2024, email correspondence, May 19, 2024.

[72] S. Sarkar, A. Naug, R. Luna, Gutierrez, A. Guillen-Perez, V. Gundecha, S. Ghorbanpour, A. R. Babu, S. Mousavi, L. D. Kashyap, and D. Markovikj, "DCRL-Green: Sustainable data center environment and benchmark for multi-agent reinforcement learning," in *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks*, 2023.

[73] B. D. Allen, "Digital twins and living models at NASA," in *Digital Twin Summit*, 2021.

[74] J. Lowe-Power, A. M. Ahmad, A. Akram, M. Alian, R. Amslinger, M. Andreozzi, A. Armejach, N. Asmussen, B. Beckmann, S. Bharadwaj *et al.*, "The gem5 simulator: Version 20.0+," *arXiv preprint arXiv:2007.03152*, 2020.

[75] C. L. Janssen, H. Adalsteinsson, S. Cranford, J. P. Kenny, A. Pinar, D. A. Evensky, and J. Mayo, "A simulator for large-scale parallel computer architectures," *International Journal of Distributed Systems and Technologies (IJDST)*, vol. 1, no. 2, pp. 57–73, 2010.

[76] B. Camus, A.-C. Orgerie, and M. Quinson, "Co-simulation of FMUs and distributed applications with SimGrid," in *Proceedings of the 2018 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 2018, pp. 145–156.

[77] F. Suter, W. Brewer, M. Maiterth, R. F. da Silva, and H. Casanova, "Comprehensive digital twins of leadership computing facilities to gain full insight on energy-efficiency optimization," in *Proceedings of the 2024 Energy-Efficient Computing for Science Workshop*. Bethesda, MD: DOE/ASCR, September 2024.

Appendix: Artifact Description/Artifact Evaluation

I. Overview of Contributions and Artifacts

A. Paper's Main Contributions

We built a comprehensive digital twin framework, and exemplify it at exascale using the Frontier supercomputer. For this, we developed three main modules: (1) a resource allocator & power simulator (A_1), (2) a thermo-fluidic cooling model (A_2), and (3) an augmented-reality model (A_3). Using these our main contributions are:

C_1 A open-source digital twin framework for data centers, ExaDigiT, with modular sub-components supported by telemetry and simulation.
C_2 Extensive verification and validation of the framework.
C_3 Demonstration of the framework at exascale for Frontier.

B. Computational Artifacts

The computational artifacts are listed here along with their respective URLs and DOIs:

A_1 https://code.ornl.gov/exadigit/raps (10.11578/dc.20240627.4)
A_2 https://code.ornl.gov/exadigit/datacenterCoolingModel (10.11578/dc.20240628.5)
A_3 https://code.ornl.gov/exadigit/exadigitUE5 (10.11578/dc.20240214.1)

The following table relates the computational artifacts to their contributions and related paper elements:

Artifact ID	Contributions Supported	Related Paper Elements
A_1	C_2	Table III
A_1	C_3	Table IV
A_1	C_3	Figure 8b
A_2	C_2	Figure 7
A_2	C_2	Figure 8a
A_3	C_3	Figure 6

II. Artifact Identification

For each artifact, A_1, A_2, and A_3, we provide its relation to contributions, expected results, expected reproduction time, artifact setup, artifact execution, and artifact analysis.

A. Computational Artifact A_1

The Resource Allocator and Power Simulator (RAPS) module is used to schedule workloads on the digital twin and compute dynamic power consumption and related statistics. RAPS is able to either replay historical data that has been run on Frontier, or dynamically schedule synthetic workloads. It also interfaces with the cooling model A_2 to predict cooling dynamics. In the paper, RAPS was used to generate Figs. 4, 8, and 9, as well as Tables III and IV.

Relation To Contributions

Table III, related to contribution C_2, verifies system power under three specific conditions: idle, peak power, and during the core phase of an High Performance Linpack (HPL) 9216-node run. Fig. 9, associated with contribution C_3, was created by replaying 24 hours of telemetry data collected on 2024-01-18, using RAPS, which includes an HPL benchmark run and is representative of large workloads on an exascale supercomputer. Table IV, also related to C_3, was generated by replaying 183 days of Frontier telemetry data, providing daily job scheduling and power consumption statistics.

Expected Results

Table III is expected to show both telemetry and RAPS predictions of power for three test cases: idle power, peak power, and HPL core phase, with RAPS predictions anticipated to be within approximately 5% of the telemetry data. Fig. 9 is expected to display energy efficiency, cooling efficiency, measured and predicted power, and system utilization. Table IV provides a statistical analysis of 183 days of telemetry data, from 2023-09-06 to 2024-03-18, replayed through RAPS. It summarizes the minimum, average, maximum, and standard deviations for values such as: average arrival rate, average nodes per job, average runtime, throughput (jobs per hour), average power (MW), loss (MW), loss percentage, total energy consumption (MW-hr), and carbon emissions (CO_2).

Expected Reproduction Time (in Minutes)

The artifact setup times varied between 30 minutes to one hour, with execution times ranging from one to two minutes for smaller datasets and up to one hour for larger datasets.

Artifact Setup (incl. Inputs)

Hardware: Any computer that can run Python.

Software: The RAPS module runs on any operating system that supports Python 3.9. If RAPS is being used in conjunction with the cooling model, the cooling model requires either Linux (glibc $>=$ 2.34) or Windows 10 (or later).

Datasets/Input: The inputs specific to the paper elements are as follows. For Table III, the inputs are defined as the `test` workload in `workload.py`, with specific settings for each test case, including the idle test (CPU=0, GPU=0), HPL core phase test (CPU=0.33, GPU=4$\times$0.79), and peak power test (CPU=1, GPU=4). For Fig. 9, the inputs include job details such as `job_name`, `job_id`, `node_count`, `time_start`, `cpu_power`, and `gpu_power` from the "slurm/joblive" and "jobprofile" telemetry parquet files dated 2024-01-18. For Table IV, the required inputs are similar job details from the telemetry parquet files spanning from 2023-09-06 to 2024-03-18.

Installation & Deployment: To set up the environment, first install Python 3.9. Then, download the source code by running the command `git clone https://code.ornl.gov/exadigit/raps`. After downloading, set up the Python environment using `pip install -e .` to complete the installation. The deployment can also be done via a Docker container, which is defined by a Dockerfile. Use the command `docker build -t raps .` to build the container, and the command `docker run -it raps` to run it.

Artifact Execution

Table III involves two tasks: (T1) running the test workload using the command `python main.py -w test`, and (T2) obtaining telemetry data for idle, peak, and HPL core cases. To determine idle and peak power, historical data from Frontier was searched, and for HPL power, the `sum_node_power` field was used from the telemetry data. The Fig. 8 workflow also consists of two tasks: (T1) running RAPS using the command `python main.py -o --plot power loss -f /data/slurm/joblive/date=2024-01-18 /data/jobprofile/date=2024-01-18` to replay job sequences from 2024-01-18 and compute power and energy consumption losses, saved as parquet files; and (T2) calculating the heat extracted from telemetry data as described in the "Artifact Analysis" section. Similarly, Table IV for RAPS involves two tasks: (T1) running RAPS for each input day using a similar command as above and can be automated with the `meta_run.sh` script found in the `scripts` directory. The `-o` switch directs RAPS to output job statistics to `simulation_results/.../stats.out`. (T2) calculating output statistics, such as average arrival rate, average nodes per job, and average runtime, using the `telemetry.py` module for each date, for example, by running `python -m raps.telemetry -f /data/slurm/joblive/date=2024-01-18 /data/jobprofile/date=2024-01-18`. Note: for users without access to Frontier telemetry data, the open PM100 dataset may be used instead, which involves downloading `job_table.parquet` from https://zenodo.org/records/10127767 and then running the command `python main.py --system marconi100 -f /data/job_table.parquet`.

Artifact Analysis (incl. Outputs)

Table III requires only the collation of both RAPS predictions and telemetry results to compute the percentage error. For Fig. 9, the energy efficiency of the system is calculated using the formula $1 - P_L$, where P_L represents the energy conversion losses. Loss values are obtained from the `loss_history.parquet` file, and the "Power with Losses" data is retrieved from the `power_history.parquet` file. System utilization is output as `util.parquet`. The "Cooling Efficiency" is computed as $\eta_{cooling} = H/P$, where the heat extracted H is calculated from Equation 7. Table IV involves collating all data from running both `main.py` and `raps.telemetry` for 183 days and computing the minimum, average, maximum, and standard deviation values.

B. Computational Artifact A_2

A_2 is a Modelica-based model of Frontier's cooling system, which models the transient thermo-fluid dynamics within the Central Energy Plant (CEP). The artifact can be run either as a standalone module (using heat inputs H from Equation 7) or in tandem with RAPS A_1 (using RAPS power outputs). Fig. 7 was created using the cooling model as a standalone module with heat inputs from system telemetry, whereas Fig. 8 was created using the cooling model with RAPS power outputs.

Relation To Contributions

The high-level relationship to contribution C_2 is demonstrated through two figures. Fig. 7 uses telemetry data from 2024-04-07, which was input into the cooling model to generate validation plots for flow rate, temperature, pressure, and Power Usage Effectiveness (PUE). Fig. 8 employs data from a synthetic test workload fed into the cooling model to produce a plot that verifies how the average cooling distribution unit (CDU) primary return temperature changed over time in relation to the total power of the system.

Expected Results

The expected results for the specific paper elements are as follows. For Fig. 7, there should be a good agreement between the predictions produced by the cooling model and the real telemetry data. For Fig. 8, when the cooling model is run with the synthetic benchmark test, the total power is expected to align with the power observed from real system observations using those benchmarks. Additionally, the average CDU primary return temperature is expected to exhibit a trend similar to the total power over time.

Expected Reproduction Time (in Minutes)

For Figs. 7 and 8, the artifact setup took approximately 5-30 minutes, execution took 12-25 minutes (involving obtaining power and weather inputs and processing them through the cooling model), and analysis (including setup and plot generation) took about 5 minutes each.

Artifact Setup (incl. Inputs)

Setup of the thermo-fluids model:

Hardware:

- Computer (min. 1 GB RAM and 400 MB storage).

Software:

- Dymola (version 2022x 64-bit) https://www.3ds.com/products/catia/dymola
  - Windows 10 (or later) with Microsoft C/C++ compiler, Linux: SUSE/11, RHEL/6.6, Ubuntu/22.04 with gcc.
- Modelica (version 4.0.0) https://github.com/modelica/ModelicaStandardLibrary/releases
- Modelica Libraries
  - TRANSFORM Library (version 0.5) https://github.com/ORNL-Modelica/TRANSFORM-Library

  - Modelica Buildings Library (version 11.0.0) https://github.com/lbl-srg/modelica-buildings
- FMPy (version 0.3.16) https://github.com/CATIA-Systems/FMPy

Datasets/Input: For Fig. 7, the inputs include (1) facility *telemetry* data with 25 CDU power values and (2) wet-bulb temperatures collected every 15 seconds over 24 hours from real facility telemetry; predicted power was derived by applying Equation 7 to this data to calculate the heat removed on 2024-04-07. For Fig. 8, the data is generated from a *synthetic* test workload that included an HPL run on 9216 nodes for one hour, an OpenMxP run on 2800 nodes for one hour, and a Max Power run on 9472 nodes for three hours.

Installation & Deployment: To set up the cooling model, first install Dymola version 2022x 64-bit from the Dassault Systèmes website. Then, install the TRANSFORM library by cloning it from the ORNL-Modelica GitHub repository, and the Modelica Buildings Library (MBL) by cloning it from the LBNL GitHub repository. Next, clone the cooling model repository from the A_2 URL given in Section I.B. Open the `setup.mos` file in a text editor and update the library locations if necessary. Launch Dymola, navigate to the Simulation tab, and run the `setup.mos` script or manually load the libraries. The cooling model is then deployed as a Functional Mock-up Unit (FMU).

Instructions for deploying FMU to RAPS: In Dymola, navigate to the File menu and select the Export option. Choose FMU (Functional Mock-up Unit) as the export type and select FMI 2.0 with "co-simulation" as the FMU type. Click "Export" to generate the FMU file. Then, copy the `filename.fmu` to the RAPS/models folder and update the FMU_PATH in the `config/frontier/cooling.json` file to point to the new FMU file location.

Artifact Execution

The general workflow for using the cooling model consists of first passing power and weather inputs to the model. All inputs were collected at 15-second intervals within a user-defined time frame, and originated from synthetic or telemetry replay workloads. After inputs were received, the cooling model generates outputs specifying various temperatures, pressures, flow rates, and other measurements of both Frontier's cooling system and the CEP. The values of those outputs were then processed and displayed in the RAPS A_1 console interface, and could also be saved for later data analysis.

For Fig. 7, 24 hours of *telemetry* data from 2024-04-07 were analyzed including: total system power, an array of 25 CDUs power values, and weather conditions recorded every 15 seconds. These data inputs were fed into the cooling model using the command `python runFMU.py`, which produced FMU results into a .csv file. The resulting data were then plotted using `python plot_sys_data.py` to visualize the cooling model's predicted flow rate, temperature, pressure, and PUE compared to telemetry data.

For Fig. 8, RAPS was used to run a *synthetic* workload consisting of four benchmark tests over a simulated period of eight hours. This duration was chosen to capture a comprehensive snapshot of each benchmark's performance. Due to the synthetic nature of the workload, a constant weather temperature of 290K was applied at each timestep. The cooling model generates 317 outputs, including the primary return temperatures for each CDU. Fig. 8 was produced using the command: `python main.py -c -t 8h -w test -p Tr_pri`.

Artifact Analysis (incl. Outputs)

The artifact is a Modelica-based thermo-fluid cooling model developed in the Dymola IDE and deployed as a Functional Mock-up Unit (FMU). It accepts 25 CDU powers and a wet-bulb temperature as inputs, simulates transient dynamics, and returns 317 outputs, including temperatures, pressures, flow rates, and other specific variables related to the Frontier supercomputer and its CEP. For the specific outputs related to the paper elements, Fig. 7 displays the cooling model predictions (in red) compared with telemetry data (in black) for the CDU and CEP. Fig. 8 shows the average CDU primary return temperature alongside the total system power for synthetic benchmark tests.

C. Computational Artifact A_3

A_3 is the augmented reality (AR) implementation of the digital twin, called exadigitUE5. It is implemented using UnrealEngine5.1 and can run on a desktop computer with Microsoft HoloLens 2 as AR Head Mounted Display (HMD). A_3 allows simulations to be instantiated from A_1 and A_2 and displays system telemetry onto the 3D scene. The AR environment is used to interact with the different parts of the system and provides an interface to the other modules of the digital twin.

Relation To Contributions

In general, A_3 relates to both C_1 and C_3 as it serves as a foundational building block of our framework, which can be reused to visualize similar systems, as demonstrated with Frontier. Specifically, in relation to the paper elements and C_3, Fig. 6 illustrates exadigitUE5 in action, presenting telemetry data and simulations of the exascale system Frontier. The digital twin is projected onto a desk in a meeting room, viewed through the camera of a Microsoft HoloLens 2, capturing a first-person perspective with an AR overlay feature of the HMD. A laptop visible in the background runs exadigitUE5, streaming the visualization to the HMD, allowing the user to select cabinets to inspect and telemetry data to display. The CDUs show the results of a simulation step of Artifact A_1, while the rectifiers and GPUs, although barely visible, display telemetry data from a data replay on 2023-09-11, covering a single 15-second snapshot around 0:00-0:30. The system is interactive and operates in real-time, with data display and simulation results requiring user-initiated queries.

Expected Results

The expected result for the specific paper element, Fig. 6, is the reproduction of a visualization that illustrates interaction with the digital twin. This figure highlights user interaction with the system, demonstrating the usability of contribution C_3. The figure also includes a scr eenshot of the dashboard and terminal console. The terminal console is simply a snapshot A_1 RAPS of the screen output for telemetry replay of 2024-01-18 as described in Section II.A. The dashboard interface shown in Fig. 6 is a snapshot of an HPL run replay from 2024-04-22 Frontier telemetry. Note, running the dashboard requires building and deploying the RAPS Docker container with Server as described in the RAPS README.md file; the details of this process are not included in this artifact description.

Expected Reproduction Time (in Minutes)

Artifact Setup (with dependencies): $\sim$ 4h
Artifact Execution: $\sim$ 10min

Artifact Setup (incl. Inputs)

Setup of exadigitUE5:

Hardware:

- Windows Computer (HoloLens2 requirement)
- Microsoft HoloLens2
- Frontier (optional)
- Frontier Telemetry Service (not public)

Software:

- exadigitUE5: https://code.ornl.gov/exadigit/exadigitue5/-/tags/0.2.0-SC24-submission
- Unreal Engine 5.1: https://github.com/EpicGames/UnrealEngine/tree/5.1 Version: commit 9bd9c84
- Microsoft Windows (dependency for HoloLens2)
- Visual Studio 2022 (Version 17.9.2)
- Microsoft-OpenXR-Unreal: https://github.com/microsoft/Microsoft-OpenXR-Unreal/ Version: commit 5c2d81b
- MixedReality-UXTools-Unreal: https://github.com/microsoft/MixedReality-UXTools-Unreal/ Version: commit 89f04fe
- WorldLockingTools: https://github.com/microsoft/WorldLockingTools-Unreal Version: commit f929156

Datasets/Input: At the time of writing, exadigitUE5 ingests data in three ways: synthetic data, as outlined in the README.md under ExternalResources/process_sample.py; stored local data from the Frontier telemetry service, which is not publicly accessible; and telemetry data obtained via API calls to the Frontier Telemetry Service, also not available to the public. For Figure 6 specifically, the input used was a stored local dataset—a 15-second snapshot captured around 0:00-0:30 on 2023-09-11.

Installation & Deployment: Installation instruction in README.md https://code.ornl.gov/exadigit/exadigitue5/-/blob/0.2.0-SC24-submission/README.md?ref_type=tags

Setup: `git clone --recurse-submodules git@code.ornl.gov:exadigit/exadigitue5.git` or `git clone --recurse-submodules https://code.ornl.gov/exadigit/exadigitue5.git`
Install Unreal Engine 5.1 from the EpicGames Installer or https://github.com/EpicGames/UnrealEngine/tree/5.1 (Note: repository access required, see instructions https://www.unrealengine.com/en-US/ue-on-github.)
For UEFMI, follow the installation guide at https://github.com/ORNL-Modelica/UnrealEngine-FMIPlugin (see README.md#Installation).

Running: For sample data run `ExternalResources/process_sample.py` (prior `pip install pandas pyarrow`).

Artifact Execution

To generate the paper elements using exadigitUE5, first load the project in Unreal Engine 5.1 and select the preferred mode (Viewport, Standalone Game, or VR-Preview). For Fig. 6, open the pre-compiled project and load the VRDemo level, where a table-sized Frontier model appears at table height, anchored by scanning a printed QR code with the headset. Using the Hand-Menu, load the selected components and the Central Energy Plant, then trigger the Telemetry Step to load input data for a specified timestep. Follow this by triggering the Simulation Step to execute artifact A1, which visualizes results on various gauges (temperature, pressure, and flow) on the CDU representations. Finally, capture a screenshot within the headset. Detailed usage instructions are available in the exadigitUE5 documentation.

Artifact Analysis (incl. Outputs)

The artifact described is an Augmented Reality visualization of the digital twin, which is able to both ingest and display telemetry data, display the central energy plant, and run the simulation as described in A_1. This can also be executed directly from a desktop, without the HMD. Specific Artifact Output for exadigitUE5:

- AR interaction and experience with the digital twin of Frontier.
- Alternatively desktop interaction and experience with the digital twin of Frontier.

Specific Output for the paper elements (Description):

- Fig. 7: A fancy screenshot as seen through the HMD by the user, as a raw image capture.