Towards the Development of an Exascale Network Digital Twin

John K. Holmen*, Md Nahid Newaz†, Srikanth Yoginath*, Matthias Maiterth*,
Amir Shehata*, Nick Hagerty*, Chris Zimmer*, Wesley Brewer*
*National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, USA
†Department of Computer Science and Informatics, Oakland University, Oakland, USA

***Abstract*—Exascale high performance computing (HPC) systems introduce new challenges related to fault tolerance due to the large component counts needed to operate at such scales. For example, the exascale Frontier system consists of approximately 60 million components. These counts warrant the investigation of new approaches for helping to ensure the functionality, performance, and usability of such systems. An approach explored by the ExaDigiT project is use of digital twins to help inform decisions related to the physical Frontier system. This paper discusses a subset of ExaDigiT's Facility Digital Twin (FDT), the Network Digital Twin (NDT), which focuses on Frontier's network as a target use case. We present the various strategies tested and early challenges faced towards the development of an exascale NDT, with the hope that such knowledge would benefit other practitioners who are interested in developing a similar digital twin.**

***Index Terms*—Digital Twin, Exascale, Network**

I. Introduction

In Fall of 2022, the DOE Frontier system debuted at No. 1 on the TOP500 with an HPL score of 1.102 Exaflop/s[1], a notable feat given that it was the first exascale system to appear on the Top500 list. Frontier consists of approximately 60 millions components. The size and complexity of this and other exascale systems warrant the investigation of new approaches for helping to ensure the functionality, performance, and usability of such systems.

One such approach currently being explored is use of digital twins. The National Academy of Sciences, Engineering, and Medicine (NASEM) define a digital twin as [17]:

> "...a set of virtual information constructs that mimics the structure, context, and behavior of a natural, engineered, or social system (or system-of-systems), is dynamically updated with data from its physical twin, has a predictive capability, and informs decisions that realize value. The bidirectional interaction between the virtual and the physical is central to the digital twin."

Of interest to this work and a related effort are the predictive capabilities of digital twins. Specifically, their ability to be used as a tool for continuous monitoring, early fault detection, and identification of root causes.

Frontier entered production in April of 2023. As of March 2024, approximately 1.8 million jobs and counting have been run across Frontier. Jobs range from production user workloads to routine system testing and span from single-node counts to the full 9,408 nodes. These jobs offer a great variety of data that can be used to inform a digital twin of the physical systems. Such data can help gain insights from visualization, assess what-if scenarios, optimize system operation, or aid virtual prototyping. One such effort aiming to make use of this data to inform a digital twin is the ExaDigiT project.

ExaDigiT is a multidisciplinary effort spanning a variety of institutions worldwide. A key goal of this community-driven effort involves the design and development an open-source framework for developing digital twins of liquid-cooled supercomputers. The effort is guided by eight working groups focused on different aspects of the FDT. As a part of the efforts, various models for the FDT have been designed for aspects such as power utilization.

However, development of an NDT is still in its early stages due to difficulties in both modeling the network at exascale, as well as challenges in acquiring data from network communications. NDTs have a wide variety of potential use cases, such as:

1) Evaluating Future Network Topologies: By virtually prototyping new network architectures, using network simulators, organizations can foresee potential challenges and optimize the network before deploying it physically.
2) Optimizing Job Schedulers: NDTs can be used in combination with power-related data and reinforcement learning techniques to optimize job scheduling for power efficiency.
3) Network Behavior Anomaly Detection: Unsupervised machine learning can be applied on the NDT data for detecting network anomalies or failures.
4) Situational Awareness: Visualization techniques can provide insights into network congestion and other conditions, helping decision-makers understand the network's state better.

[1]https://top500.org/lists/top500/2022/06/

5) Network Model of Workloads: NDTs can be used to simulate different workload scenarios, optimizing network design and management to handle these workloads effectively.

This paper describes our initial efforts investigating approaches and tools that may be helpful when preparing to develop an NDT of Frontier. Specifically, we explored NDT use cases and multiple tools including SST Macro, CrayPat, Darshan+autoperf, and MPI tracing tools. The target use case aims to collect network data and power data to help understand the relationship between network congestion and energy consumption of network-related hardware. Challenges and issues encountered throughout this investigation are also discussed.

The remainder of this paper is structured as follows: Section II provides background on related work, Section III describes our current methodology for simulating, visualizing, and validating the NDT. Section IV discusses existing challenges and opportunities, and Section V concludes this paper.

II. Background

In this section, we provide a background on the previous work towards developing a digital twin of Frontier called "ExaDigiT", a background of Frontier's hardware architecture and Slingshot network, and a background of previous work on modeling networks.

A. ExaDigit

ExaDigiT is an open-source framework for developing comprehensive digital twins of liquid-cooled supercomputers, as shown in Fig. 1 and presented in [19]. ExaDigiT contains three main modules: (1) a thermo-fluidic cooling model, (2) a Resource Allocator and Power Simulator (RAPS), and (3) a visual analytics module, which uses both dashboards and augmented reality. Telemetry data may be used for verification and validation purposes by replaying the data through the digital twin.

The integration of the NDT into the ExaDigiT FDT framework involves either simulating or replaying workloads. While telemetry provides CPU and GPU power traces at 15-second time quanta, telemetry is not provided for network utilization, which motivates part of the study in this paper, i.e., how can we measure network communication statistics on applications.

The initial goal for ExaDigiT was to develop a framework for end-to-end optimization of supercomputers. Such a tool can be used for optimizing energy consumption and reducing carbon emissions by making use of data center optimizers such as the DCRL-Green tool for evaluating and designing carbon-efficient data centers [21], as well as mitigating network congestion [2]. To date, about six months of Frontier telemetry have been replayed through the digital twin to perform verification and validation, as well as to study power and cooling efficiency. Such a framework allows researchers to perform virtual prototyping and optimizations of existing and future systems.

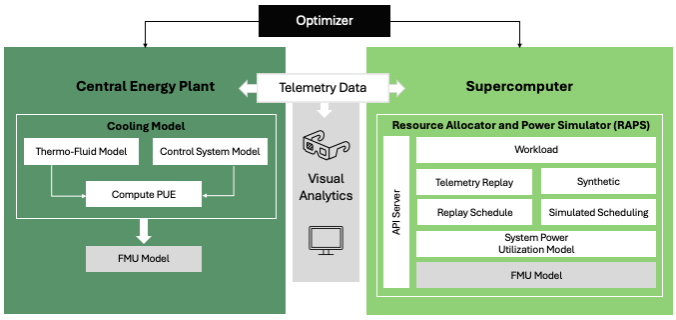

Fig. 1. ExaDigiT Facility Digital Twin (FDT) architecture [19].

The ExaDigiT effort has generated interest by supercomputing centers from around the world, including CSC (Finland), CINES (France), EPCC (UK), Pawsey Supercomputing Centre (Australia), Jülich Supercomputing Centre (Germany), as well as industrial partners such as Hewlett Packard Enterprise (HPE) and NVIDIA. To accommodate the growing interest in developing such a comprehensive digital twin, the ExaDigiT community organizes monthly large-group meetings, in addition to multiple workgroups with their own meetings. Current workgroups include: Use Cases and Digital Twin Architectures, Documentation, Application Fingerprinting, AI/ML/RL, Visual Analytics, Power and Cooling, Network, and Verification, Validation, and Uncertainty Quantification (VVUQ).

In addition to the predictive capabilities, ExaDigiT contains a visual analytics module, which has two types of interfaces: a web-based dashboard for launching simulations and an augmented reality (AR) model based on Unreal Engine 5. The AR model can be used for visually interacting with the system as well as replaying and visualizing power and thermal telemetry. One of the interesting use cases of the augmented reality model related to NDTs, is to be able to visualize network congestion on the Slingshot network. The purpose of such a visualization would be to provide a tool for HPC engineers to debug challenging network congestion problems, which are otherwise difficult to debug from a command-line console. Fig. 2 uses the ExaDigiT's AR model to reveal the interconnectedness of Frontier's nodes via its Slingshot interconnect.

B. Frontier's Network

Frontier consists of 9,408 HPE Cray EX235a nodes. Each node features one 64-core AMD EPYC™ 7A53 CPU and four AMD MI250Xs GPUs, each with two Graphics Compute Dies (GCDs). On a given node, four Slingshot network interface controller (NIC) are used to connect each GPU to the Slingshot fabric. Direct connection of the NIC to each GPU is a strategic innovation of the node design, since data mostly resides on the GPUs. Frontier's nodes are connected by HPE's Slingshot 11 interconnect. The network has a three-hop dragonfly topology consisting of 80 groups: one management, five I/O, and 74 compute groups. This topology is a direct network which uses non-minimal routing to take advantage of additional fabric paths to achieve higher bandwidth than using minimal paths

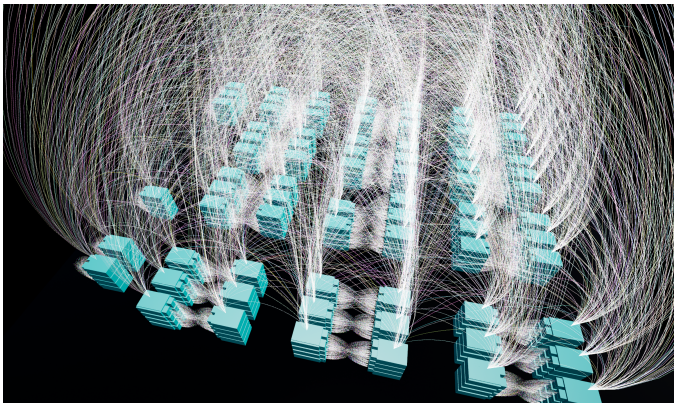

Fig. 2. Visualizations of Frontier network in Unreal Engine 5.

only. More details on Frontier's network can be found in a recent system architecture paper [3].

C. Networking Models

There have been a number of different approaches used to tackle the problem of network modeling, such as: queuing theory [8], network flow theory [12], parallel discrete-event simulations [13], virtual-machine-based solutions [22], and machine-learning techniques [2]. Moreover, analytical models have been used to model network congestion [5]. The type of digital twin may vary depending on the intended purpose. Some digital twins are focused more on designing networks, e.g., NVIDIA Air [1], while others focus more on managing networks e.g., NetGraph [15], and even others for debugging network congestion [5]. We currently use SST Macro [20] with the Dragonfly motif for modeling Frontier's network, as we are primarily concerned with modeling network traffic within fixed time quanta.

III. Methodology

After giving context on the FDT, Frontier's Slingshot network, and prior work, we now discuss methodologies explored when investigating how to develop an NDT of Frontier. In its current state, the ExaDigiT FDT is able to schedule either synthetic workloads or replay historical workloads by representing jobs as CPU and GPU utilization traces at 15-second intervals. While the workloads run, a system power utilization model dynamically estimates energy consumption and energy conversion losses; simultaneously, a thermo-fluidic cooling model dynamically predicts flow rates, temperatures, pressure, pump speeds, etc. throughout the cooling system. The immediate use case explored as a part of this work entails the incorporation of network-related data alongside more fine-grained power data (e.g., NIC power consumption). Specifically, capturing the amount of packets sent and received across the network at 15-second intervals to align with the operating interval of the FDT. We anticipate such information may be helpful for better understanding how network congestion impacts network-related hardware power consumption. Such an understanding could be used in conjunction with, for example, application fingerprinting to model representative job workloads or schedule jobs anticipated to make notable use of Frontier's network infrastructure around one another.

For a functional NDT, first we must ensure that the network model closely replicates the operational functionality of Frontier's network and then ensure that the model can meet the real-time constraints posed by the operational Frontier network. Since the NDT works in tandem with Frontier's network, the predictions that the NDT provides using the telemetry data communicated by Frontier's network in real-time should provide valuable insights, at least in near real-time, for the NDT to be beneficial. To address the former requirement, cycle-accurate simulation tools like SST Macro can be used to develop a validated simulation model of Frontier's communication network. However, the latter requirement is hard to meet since such tools are characterized by extremely slow runtime performance and cannot meet the real-time constraints of an operational Frontier network. To address both of these concerns, we propose to (a) start with the development of a network simulator for the Frontier's network using SST Macro, (b) validate the simulation results from SST Macro using the telemetry data, (c) use this validated model to run many scenarios to generate large volumes of operational network data, (d) use the collected data to train an accurate and efficient machine learning model, and (e) use the faster inference machine learning models as NDT that holds a better promise to meet the real-time constraints of the operational Frontier network. This approach is summarized in Fig. 5. The sections following discuss progress implementing this approach and observations on the steps that were actively worked upon.

A. Network Simulations

While there are a number of possible techniques that may be used to model high-performance networks, given the size of Frontier's network and the level of realism that we desired, we believe that using a parallel discrete event simulator was the optimal choice for our goals. To that end, we used the Structural Simulation Toolkit Macro Element Library, also known as SST Macro [16].

We first created trace profiles of applications using the SST DUMPI trace library[2] for a number of representative HPC benchmarks and proxy applications (e.g., LULESH, miniVite, NAS Parallel Benchmarks, OSU Micro-Benchmarks). SST DUMPI generates a trace of the application's execution, which includes timing information, size of messages sent or received, the ranks of processes involved in communication, and other relevant data. The traces generated by DUMPI can be used within the broader SST framework. SST is capable of simulating complex computer systems, i.e., functioning as an NDT, and the DUMPI traces can be fed into these simulations to study how an MPI application would perform under different conditions or hardware configurations.

In Fig. 3 we show three different types of visualization plots from replaying SST DUMPI traces: fixed-time quanta

[2]https://github.com/sstsimulator/sst-dumpi

(FTQ), spyplot (either using NIC or MPI ranks), and a network congestion plot. FTQ plots are a histogram providing a time-dependent profile of what the application is doing. Here, the FTQ plot at the top shows the percentage of time spent in compute relative to MPI at a fixed interval of 1 ms. Spyplots visualize traffic patterns via communication matrices, showing either the number of messages or number of bytes sent between two network endpoints. Here, the spyplot in the middle shows the number of bytes transferred between NICs of different nodes. Network congestion plots use color to visualize network congestion across a given topology. Here, the network congestion plot at the bottom shows network congestion for six dragonfly groups, having 18 switches in each group.

B. Simulation Validation

After simulation, our next goal was to identify a way to verify and validate simulated network statistics. Initial efforts to collect network-related data used CrayPat [10] and Darshan+autoperf [7]. While these tools provided valuable information, e.g., on time spent in specific MPI calls, it was difficult to use these tools to compute statistics at fixed time intervals over the course of the run; instead, statistics were summarized across runs as a whole. For this reason, we explored use of MPI tracing utilities to capture more fine-grained statistics during job run time. Specifically, we explored use of Libfabric's hook fabric provider utility, fi_hook[3] and MPI's profiling interface, PMPI[4]. Using these utilities, we have been able to capture MPI call details allowing us to identify which ranks are interacting with one another, messages sizes, and when the interactions occur during the run. Paired with timestamps, this data allows us to bin MPI-related activity over time to aggregate details such as message counts across the target time interval. Though we have not yet done so, we anticipate this approach to be helpful for validating SST's MPI-based spyplots capturing communication details between ranks. Note, further investigation remains to find a way to validate NIC-based spyplots.

Our long-term goal is to use the approaches used to validate our network simulation scenarios (e.g., the planned workflow in Fig. 4) to validate the digital twin. We use utilities such as fi_hook to extract statistics and use SST DUMPI and SST Macro respectively to generate traces and simulate the traces. We can compare the outputs of both methods to perform validation studies.

C. Training Machine-learned NDTs

Once we are able to accurately simulate the network using SST Macro, we can run simulations on a number of applications to curate a dataset for training a task-specific NDT as shown in Fig. 5. In addition to data from the simulations, we can also utilize data from the scheduler database, as well as system telemetry data, which are described in more detail in Table I. Because many different simulations will be required to

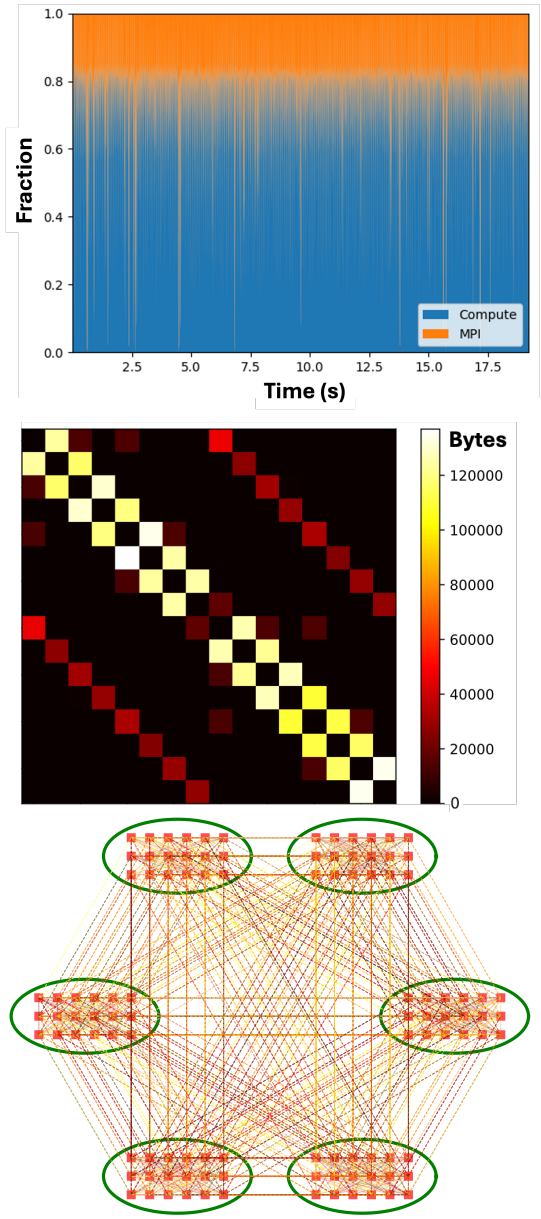

Fig. 3. Sample SST Macro simulation results. From top to bottom: (1) Fixed-time quanta visualization (1ms time epoch) from simulation of Nasa Parallel Benchmark (NPB) class D block-tridiagonal solver at 1764 ranks, (2) traffic pattern NIC spyplot running 1024 ranks on 16-node MiniVite random geometric graph (RGG) benchmark, and (3) Network congestion plot.

curate an effective training dataset, integrating the validation and simulation techniques into an existing test harness or framework could help ease the process.

Depending on the end goal, different types of strategies may be employed. Graph neural networks (GNN) are a natural fit for modeling link-level network traffic, e.g., [11]. Recurrent neural networks (RNN), such as Long Short-term Memory (LSTM), along with temporal convolutional networks (TCN) are able to capture the network's temporal behavior for applications such as forecasting network traffic [6]. Reinforcement Learning (RL) approaches are appropriate for solving opti-

[3]https://ofiwg.github.io/libfabric/v1.13.1/man/fi_hook.7.html
[4]https://github.com/hagertnl/mpi-trace

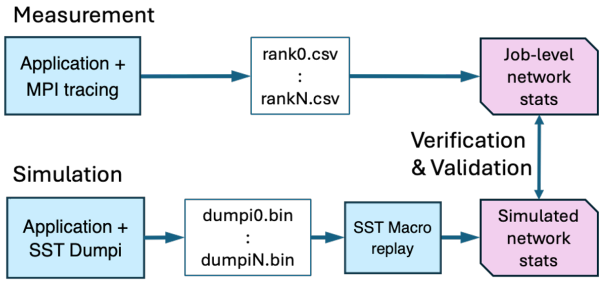

Fig. 4. Workflow for validation of network simulations.

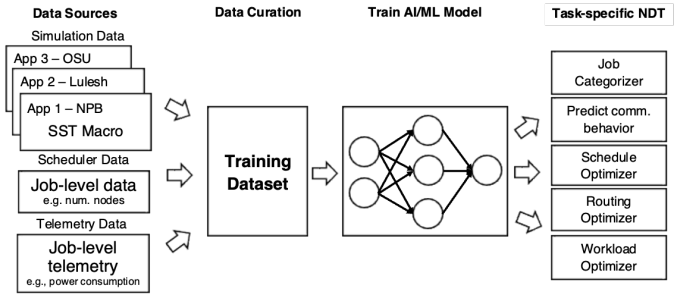

Fig. 5. Workflow for developing task-specific NDTs.

mization problems, such as routing, schedule, and workload optimizations [4]. Almasan et al. [2] gives a good survey of different approaches for developing machine-learned NDTs.

IV. Challenges & Opportunities

After presenting the methodology of developing of an NDT, we now discuss the challenges and opportunities involved in developing an NDT for Frontier.

A. Early Challenges

1) Data Availability: While telemetry data is readily available for power and cooling data, network-related telemetry data must be obtained by systematically profiling applications. Such a systematic analysis of applications can take a considerable amount of time and research to develop a way to both profile job workloads efficiently and extract the information needed for validation with the NDT.

2) Data Visualization: Dealing with vast amounts of networking data becomes a challenge as it complicates the scene complexity when visualizing the digital twin. In the context of the NDT, the current visualization captures all network connections between Frontier nodes (e.g., Fig. 2). The amount of interconnectedness, however, can make it challenging to gain meaningful insights related to network traffic. In the future, finding meaningful subsets of connections to visualize could be beneficial for improving the usefulness of the visualization component. Such a visualization tool could help HPC engineers debug network congestion problems, that are otherwise difficult to diagnose.

TABLE I
Types of data used to train NDT.

Job-level Data	
MPI Comm. Matrix	Between parallel tasks (MPI Ranks)
NIC Comm. Matrix	Between network endpoints (Compute Nodes)
Network Congestion Values	On each NIC (Intra-job interference)
Scheduling Info	Compute time, Comm. time, Est. runtime, Num. processes, Num. nodes
Power	Power traces of GPU & CPU
Topology-level Data	
Network Congestion Values	On each NIC (Inter-job interference)
Scheduling Data	Task-to-core mapping, Job-to-node mapping
System Throughput	All jobs completion time
Other System Metrics	Average system utilization, Bandwidth utilization

3) Differing Timescales: Incorporating an NDT into the FDT (Fig. 1) presents a challenge as the timescale of interest is several orders of magnitude faster than the timescales of interest for cooling and power simulators. For example, the switching technology of the dragonfly network is between 100 to 350ns [9], [14], whereas Frontier system telemetry is typically collected at 15-second time quanta. To work around this challenge, a separate module with its own built-in scheduler for studying issues such as network congestion could be used. Such a module could be used to compute the amount of time spent in communication, which could be used to augment CPU/GPU utilization traces informing the FDT.

4) Simulation Time: Another difficulty we encountered had to do with the length of time required to replay the traces through SST Macro. Queue policies on Frontier limit jobs to two hours for less than 92 nodes. We estimated that replaying a single MPI rank on each node of Frontier would take more than 30 hours. Tools such as SST-Core seek to speed up this process by $\sim$7x [18], which still does not provide enough speedup to fit within a two-hour allocation.

5) Tools: Another hurdle to overcome involved evaluating the different types of tools that could be used to generate the type of data necessary to validate the NDT. With a variety of profiling and tracing tools available, it was unclear which tools would best align with our NDT goals. In particular, the 15-second operating interval of the FDT was problematic for reasons noted in IV-A3. Initially, we explored using profiling tools such as CrayPat [10] and Darshan+autoperf [7]. However, such tools were not able to profile the applications in a temporal way. MPI tracing tools, such as fi_hook, were found to align better with our goals of extracting intercommunication statistics at 15-second intervals.

B. Opportunities

While it has taken a considerable amount of time and effort to understand and evaluate different types of tools and benchmarks for validating and simulating an NDT, we believe we now have an established approach that can be used towards building an exascale NDT. Scaling the NDT to exascale will require overcoming of the hurdles mentioned in Section IV,

and performing validations and simulations for a variety of benchmarks at increasing scales. Once we have sufficiently verified and validated the method, we plan to demonstrate dynamic visualizations of network traffic within ExaDigiT's AR model, as well as integration of the network statistics into the FDT for investigating energy efficiency.

V. Conclusions

The size and complexity of exascale HPC systems warrant the investigation of new approaches for helping to ensure the functionality, performance, and usability of such systems. An approach explored for easing maintenance of the exascale Frontier system is the use of digital twins to help inform decisions related to the physical system. This exploration has been pursued as a part of the international ExaDigiT project. The incorporation of an NDT into the FDT has been a key area of interest for the project.

This paper discussed early efforts investigating ways to incorporate an NDT into ExaDigiT's FDT. As a part of this, the paper discussed NDT use cases, early challenges, methodologies, and long-term goals for the effort. Next steps include validating simulated results across a number of applications, developing effective dynamic visualizations of the NDT, and working to identify meaningful time intervals to operate an NDT in the context of the FDT. Moreover, we would like to use these techniques discussed to develop models for power predictions, study network congestion, and routing optimization.

VI. Acknowledgments

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

References

[1] "NVIDIA Air: Transform your data center operations using digital twins." accessed: 2023-10-23. [Online]. Available: https://www.nvidia.com/en-us/networking/ethernet-switching/air/

[2] P. Almasan, M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. López, A. A. P. Perales, P. Harvey, L. Ciavaglia, L. Wong *et al.*, "Network digital twin: Context, enabling technologies, and opportunities," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 22–27, 2022.

[3] S. Atchley, C. Zimmer, J. Lange, D. Bernholdt, V. Melesse Vergara, T. Beck, M. Brim, R. Budiardja, S. Chandrasekaran, M. Eisenbach, T. Evans, M. Ezell, N. Frontiere, A. Georgiadou, J. Glenski, P. Grete, S. Hamilton, J. Holmen, A. Huebl, D. Jacobson, W. Joubert, K. Mcmahon, E. Merzari, S. Moore, A. Myers, S. Nichols, S. Oral, T. Papatheodore, D. Perez, D. M. Rogers, E. Schneider, J.-L. Vay, and P. K. Yeung, "Frontier: Exploring exascale," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '23. New York, NY, USA: Association for Computing Machinery, 2023.

[4] G. Bernárdez, J. Suárez-Varela, A. López, B. Wu, S. Xiao, X. Cheng, P. Barlet-Ros, and A. Cabellos-Aparicio, "Is machine learning ready for traffic engineering optimization?" in *2021 IEEE 29th International Conference on Network Protocols (ICNP)*. IEEE, 2021, pp. 1–11.

[5] 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.

[6] J. Bi, X. Zhang, H. Yuan, J. Zhang, and M. Zhou, "A hybrid prediction method for realistic network traffic with temporal convolutional network and lstm," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1869–1879, 2021.

[7] S. Chunduri, S. Parker, P. Balaji, K. Harms, and K. Kumaran, "Characterization of MPI usage on a production supercomputer," in *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2018, pp. 386–400.

[8] W. J. Dally, "Performance analysis of k-ary n-cube interconnection networks," *IEEE transactions on Computers*, vol. 39, no. 6, pp. 775–785, 1990.

[9] 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.

[10] L. DeRose, B. Homer, D. Johnson, S. Kaufmann, and H. Poxon, "Cray performance analysis tools," in *Tools for High Performance Computing: Proceedings of the 2nd International Workshop on Parallel Tools for High Performance Computing, July 2008, HLRS, Stuttgart*. Springer, 2008, pp. 191–199.

[11] M. Ferriol-Galmés, K. Rusek, J. Suárez-Varela, S. Xiao, X. Shi, X. Cheng, B. Wu, P. Barlet-Ros, and A. Cabellos-Aparicio, "Routenet-erlang: A graph neural network for network performance evaluation," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 2018–2027.

[12] L. R. Ford, "Network flow theory," 1956.

[13] R. M. Fujimoto, "Parallel discrete event simulation," *Communications of the ACM*, vol. 33, no. 10, pp. 30–53, 1990.

[14] S. Hirasawa, H. Yamaki, and M. Koibuchi, "Packet forwarding cache of commodity switches for parallel computers," in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2021, pp. 366–376.

[15] H. Hong, Q. Wu, F. Dong, W. Song, R. Sun, T. Han, C. Zhou, and H. Yang, "Netgraph: An intelligent operated digital twin platform for data center networks," in *Proceedings of the ACM SIGCOMM 2021 Workshop on Network-Application Integration*, 2021, pp. 26–32.

[16] 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.

[17] 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

[18] A. Ramaswamy, N. Kumar, A. Neelakantan, H. Lam, and G. Stitt, "Scalable behavioral emulation of extreme-scale systems using structural simulation toolkit," in *Proceedings of the 47th International Conference on Parallel Processing*, 2018, pp. 1–11.

[19] A. Redacted, "A digital twin framework for liquid-cooled supercomputers as demonstrated at exascale," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, November 2024, to be published.

[20] A. F. Rodrigues, K. S. Hemmert, B. W. Barrett, C. Kersey, R. Oldfield, M. Weston, R. Risen, J. Cook, P. Rosenfeld, E. Cooper-Balis *et al.*, "The structural simulation toolkit," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 4, pp. 37–42, 2011.

[21] S. Sarkar, A. Naug, A. Guillen, R. Luna, V. Gundecha, A. R. Babu, and S. Mousavi, "Sustainability of data center digital twins with reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 832–23 834.

[22] S. B. Yoginath, K. S. Perumalla, and B. J. Henz, "Virtual machine-based simulation platform for mobile ad-hoc network-based cyber infrastructure," *The Journal of Defense Modeling and Simulation*, vol. 12, no. 4, pp. 439–456, 2015.