RAPTOR: Reconfigurable Advanced Platform for Transdisciplinary Open Research

Hamed Najafi, Pratik Poudel, Kiavash Bahreini, Julio Ibarra, Fahad Saeed,
Yuepeng Li, Jayantha Obeysekera, Jason Liu
Florida International University
Miami, Florida, USA

ABSTRACT

Scientific research is increasingly relying on complex workflows that span multiple computing paradigms, including high-performance computing (HPC), high-throughput computing (HTC), and machine learning/artificial intelligence (ML/AI). Traditional monolithic computing infrastructures often struggle to accommodate these diverse and evolving demands. The Reconfigurable Advanced Platform for Transdisciplinary Open Research (RAPTOR) addresses this challenge by providing a dynamically reconfigurable computing environment that integrates with federated resources. RAPTOR's architecture enables dynamic provisioning between an HPC cluster and the Chameleon Cloud platform based on workload requirements, supporting bare-metal customization for specialized applications. This paper focuses on RAPTOR's reconfigurability features and demonstrates their effectiveness through quantitative performance evaluations across four scientific domains: computational proteomics, climate modeling, weather research, and hurricane risk assessment. Our results demonstrate that RAPTOR's reconfigurable design significantly enhances research productivity by providing an appropriate computing environment for diverse computational needs.

CCS CONCEPTS

• Computing methodologies → Distributed computing methodologies; **• Computer systems organization → Distributed architectures**; **Reconfigurable computing**.

KEYWORDS

Reconfigurable computing, scientific workflows, high-performance computing, cloud computing, federated resources

ACM Reference Format:
Hamed Najafi, Pratik Poudel, Kiavash Bahreini, Julio Ibarra, Fahad Saeed,, Yuepeng Li, Jayantha Obeysekera, Jason Liu. 2025. RAPTOR: Reconfigurable Advanced Platform for Transdisciplinary Open Research. In *The 34th International Symposium on High-Performance Parallel and Distributed Computing (HPDC '25), July 20–23, 2025, Notre Dame, IN, USA*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3731545.3744665

HPDC '25, July 20–23, 2025, Notre Dame, IN, USA

ACM ISBN 979-8-4007-1869-4/25/07
https://doi.org/10.1145/3731545.3744665

1 INTRODUCTION

Modern scientific research faces increasingly complex computational demands that span multiple computing paradigms. A single research project may require high-performance computing (HPC) for parallel simulations, high-throughput computing (HTC) for processing large volumes of independent tasks, and specialized environments for machine learning and artificial intelligence (ML/AI) applications. Traditional computing infrastructures—designed primarily for a single computing paradigm—often struggle to accommodate this diversity of needs, creating bottlenecks in scientific workflows and hindering interdisciplinary collaboration.

We designed and developed the Reconfigurable Advanced Platform for Transdisciplinary Open Research (RAPTOR) to address some of these challenges through a novel architecture that dynamically adapts to diverse computational requirements. Unlike static and monolithic computing environments, RAPTOR implements a reconfigurable design that allows compute nodes to be provisioned on demand between an HPC cluster and the Chameleon Cloud platform [8]. This flexibility enables researchers to access the most suitable computing environment for their specific applications—whether that involves bare-metal provisioning for specialized software stacks, integration with the Open Science Grid (OSG) [12] for high-throughput computing, or HPC cluster resources for tightly coupled parallel applications.

This paper demonstrates RAPTOR's effectiveness in supporting diverse scientific workflows across multiple domains, with a specific focus on how its reconfigurability features address key computational challenges. We present performance benchmarks from four scientific applications—proteomics analysis, sea level prediction, weather research, and hurricane risk assessment—highlighting RAPTOR's specific advantages compared to traditional computing environments.

The remainder of this paper is organized as follows. Section 2 presents RAPTOR's design considerations in addressing the various computational demands for modern research. In Section 3, we discuss RAPTOR's architecture which provides reconfigurability and adaptability to a diverse range of scientific workflows based on data characteristics and research requirements. In Section 4, we examine several representative applications to demonstrate RAPTOR's capabilities in advancing scientific discovery. By outlining these aspects, this study highlights how RAPTOR enhances computational efficiency, fosters interdisciplinary collaboration, and supports a broad spectrum of research domains. We conclude the paper and outline future directions in Section 5.

2 DESIGN CONSIDERATIONS AND POLYMORPHIC CAPABILITIES

Traditional computing infrastructures are typically optimized for a single computing paradigm, making them inefficient for the increasingly diverse computational demands of modern scientific research. We identified key challenges that informed RAPTOR's reconfigurable design:

- **Resource allocation inflexibility:** Fixed hardware allocations in traditional HPC environments lead to resource under-utilization for workflows with varying computational needs over time.
- **Software environment constraints:** Standard HPC clusters impose strict software environments that limit the deployment of specialized applications, such as emerging AI frameworks or domain-specific tools.
- **Interoperability barriers:** Institutional boundaries restrict access to specialized computing resources, hindering collaborative research that could benefit from shared infrastructure.
- **Workflow transition overhead:** Moving between computing paradigms (e.g., from HPC simulation to ML analysis) typically requires manual data transfer and environment reconfiguration, creating workflow bottlenecks.
- **Heterogeneous resources:** Modern scientific applications increasingly require the seamless integration of diverse computing resources, including CPUs, GPUs, high-memory nodes, and storage systems, which would be challenging to achieve in any monolithic system environments.

To address these challenges, RAPTOR implements a reconfigurable architecture with five key polymorphic capabilities:

1. **High-performance computing:** RAPTOR connects high-end compute nodes to FIU's existing HPC cluster, providing a unified environment for parallel applications with dynamic scaling capabilities that overcome fixed allocation limitations of traditional clusters.
2. **High-throughput computing:** Through OSG integration [5], RAPTOR enables the opportunistic execution of independent jobs across distributed resources, extending computational reach beyond institutional boundaries.
3. **Data-intensive computing:** RAPTOR nodes feature multi-processor, multi-core servers with GPU acceleration, large memory (up to 1.5 TB/node), and scalable storage for big data processing with dynamic resource allocation.
4. **Real-time on-demand provisioning:** Using Chameleon's bare-metal provisioning, researchers gain full control over software stacks, enabling customized environments for applications with strict deadlines or specialized configurations. While container solutions like Kubernetes or Singularity provide portability on HPC and cloud platforms, RAPTOR's emphasis on bare-metal provisioning offers a complementary path that ensures unrestricted hardware access and allows full OS/kernel customization.
5. **Federated resource integration:** RAPTOR operates as a federated resource with Chameleon and OSG, enabling researchers to access and utilize resources based on their specific requirements while maintaining secure isolation.

These polymorphic capabilities are enabled through several key reconfigurability mechanisms: (1) network-level reconfiguration using VLANs and software-defined networking, (2) system-level reconfiguration with automated OS provisioning, (3) resource management integration across multiple scheduling systems, and (4) unified storage access across computing environments. Together, these mechanisms allow RAPTOR to dynamically adapt its architecture to meet the diverse computational demands of scientific applications while maintaining efficient resource utilization.

RAPTOR's design intentionally balances two key sets of trade-offs. First, it weighs the increased management complexity and switching overhead of a reconfigurable system against the flexibility gains offered by specialized configurations. Second, it considers the security challenges introduced by federated access alongside the collaborative research opportunities that federation makes possible [2, 3]. Our evaluation of diverse scientific workflows demonstrates that these trade-offs are well-justified by the significant performance improvements and research productivity gains achieved through RAPTOR's reconfigurable architecture.

3 RAPTOR ARCHITECTURE

3.1 System Architecture

RAPTOR's architecture is designed around the principle of reconfigurability, enabling dynamic resource allocation across different computing environments. As shown in Figure 1, the system consists of three interconnected zones: the CHI Zone (Chameleon Cloud), the HPC Zone, and the Storage Zone [1].

The CHI Zone features RAPTOR Nodes with AMD EPYC processors, DDR4 memory, and NVIDIA A100 GPUs, provisioned via Chameleon's CHI-in-a-Box [9]. It includes Management Nodes for lease management and a Storage Server providing 1 TB of scratch storage per user. RAPTOR's bare-metal access eliminates virtualization overhead for performance-sensitive applications.

The HPC Zone is connected to FIU's High-Performance Cluster [4], which features over 3,000 Intel cores, high-memory nodes, and 22 GPUs. The cluster comprises Master Nodes, compute nodes, a Login Node, and a Visualization Node, utilizing both Ethernet and InfiniBand interconnects. It federates with OSG, extending capabilities beyond institutional boundaries, and uses DDN Storage for high-throughput data access.

A key innovation is the Provisioning Server which dynamically reconfigures nodes between zones using IPMI. This process is managed through a CLI tool that automates network settings, boot environments, and resource allocation. By default, nodes reside in the CHI Zone until specifically requested for HPC use.

The Storage Zone provides flexible, high-performance storage through a combination of technologies. The Storage Zone provides flexible storage through S3-compatible object storage, CEPH distributed storage [15], and iSCSI block storage, enabling researchers to scale resources according to specific workflow requirements.

While the current system uses specific core zones, its architecture could theoretically support more. However, the scalability of managing numerous zones, particularly their control plane, requires future investigation.

The network infrastructure consists of two key switches that enable RAPTOR's reconfigurability. A Dell Switch is positioned

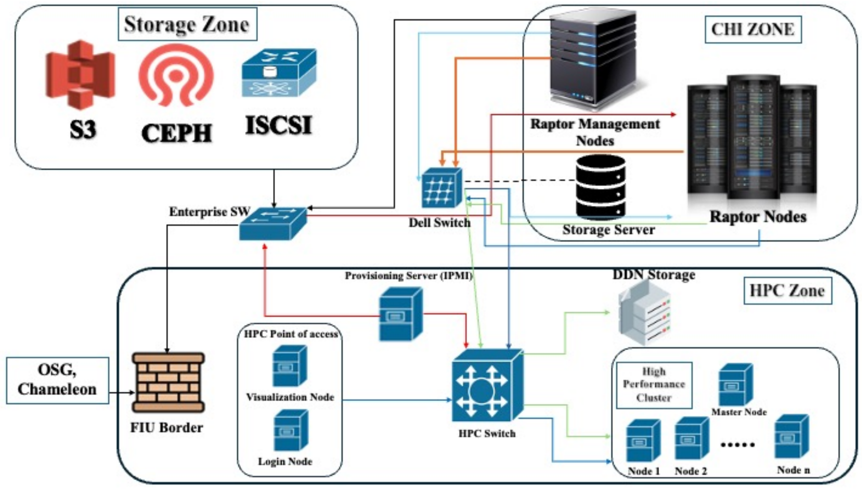

Figure 1: The Architecture of RAPTOR

between the CHI Zone and HPC Zone, facilitating the dynamic reassignment of computing resources between these environments. An Enterprise Switch connects the Storage Zone to both the HPC Zone and CHI Zone, ensuring that storage resources are accessible regardless of which computing environment is being used. These switches implement network-level reconfiguration, enabling RAPTOR's dynamic resource allocation capabilities.

3.2 Reconfigurability

RAPTOR achieves dynamic reconfigurability through several key mechanisms:

Network-level reconfiguration: The system uses Virtual Local Area Networks (VLANs) and software-defined networking [10] to dynamically reassign nodes between the CHI and HPC environments. This reconfiguration is managed by the Dell Switch and Enterprise Switch, which provide connectivity across zones.

System-level reconfiguration: The Provisioning Server handles operating system deployment and configuration, allowing nodes to boot into different environments based on workload requirements. This includes specialized OS images for bare-metal deployments through Chameleon and standard HPC environments for cluster-based workloads.

Resource management integration: RAPTOR integrates with both the Slurm workload manager [16] (for HPC workloads) and Chameleon's leasing system (for bare-metal provisioning). This allows researchers to request resources through familiar interfaces while leveraging RAPTOR's reconfigurability.

Automated workflow support: The system includes tools for transitioning data and workflows between environments, enabling researchers to move seamlessly between computing paradigms as their needs evolve. Importantly, RAPTOR's node reconfiguration between zones (e.g., from the HPC Zone to the CHI Zone or vice versa) occurs between application executions. Running applications, including any tightly-coupled modules, are allowed to complete their execution on their allocated resources before these resources become eligible for reprovisioning into a different environment for subsequent workloads. This operational model ensures that the reconfiguration process itself does not impact the performance of active computations. Table 1 summarizes RAPTOR's reconfigurability features and their benefits for scientific workflows.

Table 1: RAPTOR Reconfigurability Features and Benefits

Feature	Implementation	Benefit
Dynamic resource allocation	CLI tool for node reassignment between environments	Optimizes resource usage based on current research demands
Bare-metal provisioning	Chameleon CHI-in-a-Box integration	Enables customized environments for specialized applications
Federation with national resources	OSG and Chameleon Cloud integration	Extends computational reach beyond local resources
Seamless paradigm transitions	Unified storage access and workflow tools	Supports complex workflows spanning multiple computing paradigms

4 SCIENTIFIC APPLICATIONS AND PERFORMANCE EVALUATION

To demonstrate RAPTOR's effectiveness in supporting diverse scientific workflows, we evaluated its performance across four representative applications spanning different computational paradigms. We highlight the specific reconfigurability features for each application that addressed key computational challenges.

4.1 GPU-Accelerated Proteomics Analysis

Computational proteomics relies on comparing experimental mass spectrometry (MS) spectra against large databases of theoretical spectra to identify peptides. As database sizes grow to the terabyte scale for proteogenomic and metaproteomics analyses [6], traditional computing approaches become impractical, with processing times extending to weeks or months.

RAPTOR's reconfigurability features enabled significant performance improvements for the GiCOPS framework [7], which implements a distributed database search algorithm for peptide identification. By leveraging bare-metal provisioning, researchers configured specialized environments with optimized CUDA and MPI installations—the ability to dynamically allocate GPU-accelerated nodes on demand provided flexibility unavailable on traditional HPC resources.

Without making any modification to the code base, we reconfigured the RAPTOR environment (using bare-metal nodes), compiled MPI/CUDA GiCOPS code base, and then executed the experiments. Our experiments demonstrate that GiCOPS outperforms the existing GPU-based database search algorithms by more than 10 times in both closed- and open-search modes and 2 times as compared to the CPU-only HiCOPS. The code for GICOPS (MPI/CUDA) was compiled on RAPTOR without any modifications. The speedup attained when running GiCOPS on RAPTOR is comparable to the speedup obtained on traditional distributed memory architecture for same number of nodes.

Experiments used PRIDE datasets PXD055735, PXD055119, and PXD015384 [13], scaling MPI processes while maintaining constant parameters. Figure 2 shows execution time and speedup results. Our results show that we were able to successfully compile, troubleshoot, and execute MPI and CUDA based code bases with minimal modification on the RAPTOR machines.

By scaling workflows in the omics disciplines, RAPTOR empowers researchers to push the boundaries of computational biology, leading to faster biomarker discovery, improved disease modeling, and more effective drug development strategies. More specifically, our results show that RAPTOR's GPU-accelerated nodes provided a 6x-12x speedup (due to GPU node's availability) compared to CPU-only implementations (for 4 MPI processes with one GPU per process). This acceleration reduced processing time from days to hours, enabling more comprehensive analyses that would be impractical on traditional resources.

For this particular application, key benefits included customized CUDA/MPI configurations, on-demand GPU allocation, and seamless development-to-production transitions.

4.2 Machine Learning for Sea Level Prediction

Coastal flooding prediction requires processing large multivariate time series datasets to identify patterns and make forecasts. Traditional statistical approaches struggle with the complex, nonlinear relationships in these datasets, leading researchers to explore machine-learning techniques.

We used RAPTOR to evaluate PatchTST [11], a state-of-the-art multivariate time series transformer model, for predicting sea level variability at 21 National Oceanic and Atmospheric Administration (NOAA) tide gauges. The reconfigurable nature of RAPTOR allowed researchers to create custom environments with specialized deep-learning frameworks.

We compared training times for the PatchTST model on RAPTOR with those on local workstations for a multivariate dataset comprising 21 variables across 52,696 time steps. The results show a consistent speedup on RAPTOR compared to local workstations. This acceleration enabled researchers to explore more hyperparameter configurations and longer prediction horizons, significantly enhancing model accuracy for coastal flooding prediction.

This research significantly benefited from RAPTOR's reconfigurability, including: 1) a customized environment with specialized deep-learning frameworks; 2) having on-demand access to high-memory nodes for extensive dataset processing; and 3) the ability to scale computations based on model complexity.

Table 2: WRF Model Configuration

Resolution	Domain Size	Physics Options
1 km	500 x 500 km	Standard
1 km	500 x 500 km	Advanced microphysics
1 km	500 x 500 km	Advanced microphysics + land surface

4.3 Weather Research and Forecasting Model

The Weather Research and Forecasting (WRF) model [14] is a numerical weather prediction system designed for atmospheric research and operational forecasting. Running high-resolution WRF simulations requires substantial computational resources and specialized software configurations. RAPTOR's reconfigurability enabled researchers to deploy the WRF model for South Florida with customized environments optimized for meteorological simulations. The bare-metal provisioning capability enabled the installation of specialized libraries and dependencies that are challenging to configure in traditional HPC environments.

We evaluated WRF's performance on RAPTOR for high-resolution (1 km) simulations covering a 500 x 500 km domain centered on South Florida. Table 2 compares different simulation configurations.

The RAPTOR environment enabled high-resolution WRF simulations that were previously impractical due to software configuration challenges. These simulations offer crucial insights into local weather patterns that impact coastal communities, including severe weather events such as heavy rainfall and thunderstorms.

Key benefits for this application included specialized environment configuration for meteorological simulations, access to high-performance I/O for multi-source data integration, and ability to adjust resource allocation based on simulation complexity.

4.4 Stochastic Storm Simulation for Hurricane Risk Assessment

The Florida Public Hurricane Loss Model (FPHLM) [3] requires simulating over 140,000 synthetic storms to estimate property losses and insurance risks. This computationally intensive process involves independent simulations, which are ideal for high-throughput computing approaches.

RAPTOR's integration with the Open Science Grid (OSG) enabled the efficient execution of these simulations across distributed resources. By federating RAPTOR with OSG, researchers gained access to computational capacity far beyond local resources, significantly reducing the time required for comprehensive risk assessments.

For the 140,000 storm simulations, attempts to use only the local FIU High-Performance Cluster (RAPTOR's HPC Zone) faced significant throughput limitations. However, RAPTOR's orchestration of this workload across the federated Open Science Grid (OSG) dramatically reduced computation time from months to weeks. This highlights RAPTOR's cross-environment capability—leveraging OSG beyond the local HPC Zone—delivering quantitative performance gains compared to standard execution on institutional resources alone.

This research is benefited from RAPTOR's reconfigurability. More specifically, the federation with OSG enabled access to distributed computing resources beyond what has been offered at the university. Also, the high-throughput computing environment

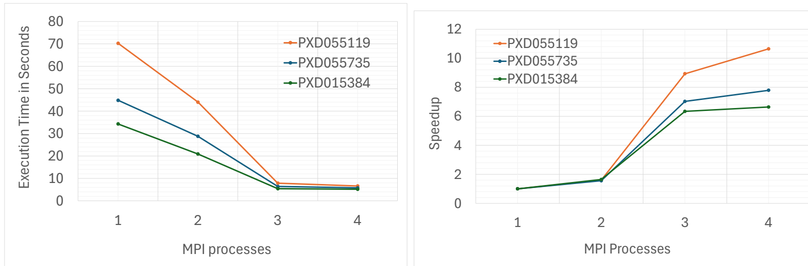

Figure 2: Results of HICOPS with multiple MPI processes.

can be optimized for many independent simulations. Efficient data management allows for large simulation outputs.

5 DISCUSSION AND CONCLUSION

RAPTOR demonstrates that a reconfigurable computing platform can effectively address the diverse computational needs of modern scientific research. RAPTOR overcomes key limitations of traditional computing infrastructures by providing dynamic allocation between HPC and cloud environments, bare-metal provisioning for specialized applications, and federation with national resources.

Our performance evaluations across four scientific domains show significant advantages compared to traditional approaches:

- **Accelerated computation:** For GPU-accelerated proteomics workflows, RAPTOR provided speedups of up to 12x.
- **Workflow flexibility:** The reconfigurable architecture enabled seamless transitions between computing paradigms, supporting complex workflows that span HPC, HTC, and ML/AI applications.
- **Resource efficiency:** Dynamic resource allocation optimizes utilization based on current research demands, avoiding the underutilization common in static computing environments.
- **Enhanced collaboration:** The federation with national resources extended computational reach beyond institutional boundaries, fostering multi-institutional research collaborations that transcend geographical boundaries.

These benefits directly address the challenges identified in the FlexScience workshop's focus on flexible computing infrastructures for scientific applications. RAPTOR's approach to reconfigurability offers valuable insights for designing the next generation of research computing platforms that can adapt to evolving computational demands.

Future work will integrate RAPTOR with our petabyte-scale storage system designed to support data-intensive applications. This integration will further enhance RAPTOR's capabilities for real-time simulations, high-throughput data processing, and machine learning-driven research.

ACKNOWLEDGMENTS

We would like to acknowledge the contribution of other team members who participated in the development of the project, including Mike Kirgan, Anu Chirinos, Anthony Bellantuono, Julian Garcia, Eric Johnson, Vasilka Chergarova, Peng Hou, Anupama John, Oscar Guzman, Jalen Atwell, and Syed Talha Khalid. This work was supported in part by the National Science Foundation under grants OAC-2126253, OAC-2312599, and IIS-2331908, and by the National Institutes of Health under grant R35GM153434.

REFERENCES

[1] 2025. Reconfigurable Computing — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Reconfigurable_computing. Accessed: 2025-05-26.

[2] Ryan Chard, Yadu Babuji, Zhuozhao Li, Tyler Skluzacek, Anna Woodard, Ben Blaiszik, Ian Foster, and Kyle Chard. 2020. Funcx: A federated function serving fabric for science. In *Proceedings of the 29th International symposium on high-performance parallel and distributed computing*. 65–76.

[3] J. M. Collins, D. Roache, N. Avissar, J. Jean-Michel, and S. Hamid. 2021. *Florida Public Hurricane Loss Model 8.1.* Technical Report. Florida International University. https://www.sbafla.com/method/Portals/Methodology/ModelSubmissions/2021/FIU_2021_Submission.pdf

[4] Florida International University. 2023. High Performance Computing at FIU: Panther Cluster. Website. Retrieved from https://research.fiu.edu/hpc/.

[5] James Frey, Todd Tannenbaum, Miron Livny, Ian Foster, and Steven Tuecke. 2002. Condor-G: A computation management agent for multi-institutional grids. *Cluster Computing* 5 (2002), 237–246.

[6] Muhammad Haseeb and Fahad Saeed. 2021. High performance computing framework for tera-scale database search of mass spectrometry data. *Nature computational science* 1, 8 (2021), 550–561.

[7] Muhammad Haseeb and Fahad Saeed. 2023. GPU-acceleration of the distributed-memory database peptide search of mass spectrometry data. *Scientific Reports* 13, 1 (2023), 18713.

[8] Kate Keahey, Pierre Riteau, Dan Stanzione, Tim Cockerill, Joe Mambretti, Paul Rad, and Paul Ruth. 2019. Chameleon: a scalable production testbed for computer science research. In *Contemporary High Performance Computing*. 123–148.

[9] Joe Mambretti, Jim Chen, and Fei Yeh. 2015. Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn). In *2015 international conference on cloud computing research and innovation (ICCCRI)*. IEEE, 73–79.

[10] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review* 38, 2 (2008), 69–74.

[11] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022).

[12] Ruth Pordes, Don Petravick, Bill Kramer, Doug Olson, Miron Livny, Alain Roy, Paul Avery, Kent Blackburn, Torre Wenaus, F Wurthwein, et al. 2007. The open science grid. In *Journal of Physics: Conference Series*, Vol. 78.

[13] Proteomics Team, EMBL-EBI. 2025. PRIDE – PRoteomics IDEntifications Database. https://www.ebi.ac.uk/pride/. Accessed: 2025-05-26.

[14] William C Skamarock, Joseph B Klemp, Jimy Dudhia, David O Gill, Zhiquan Liu, Judith Berner, Wei Wang, Jordan G Powers, Michael G Duda, Dale M Barker, et al. 2019. A description of the advanced research WRF model version 4. *National Center for Atmospheric Research: Boulder, CO, USA* 145, 145 (2019), 550.

[15] Sage Weil, Scott A Brandt, Ethan L Miller, Darrell DE Long, and Carlos Maltzahn. 2006. Ceph: A scalable, high-performance distributed file system. In *Proceedings of the 7th Conference on Operating Systems Design and Implementation (OSDI'06)*. 307–320.

[16] Andy B Yoo, Morris A Jette, and Mark Grondona. 2003. Slurm: Simple linux utility for resource management. In *Workshop on job scheduling strategies for parallel processing*. Springer, 44–60.