Check for updates

ACM DIGITAL LIBRARY | Association for Computing Machinery | acm open

Latest updates: https://dl.acm.org/doi/10.1145/3708035.3736060

SHORT-PAPER

The National Research Platform: Stretched, Multi-Tenant, Scientific Kubernetes Cluster

DEREK WEITZEL, University of Nebraska–Lincoln, Lincoln, NE, United States

ASHTON GRAVES, University of Nebraska–Lincoln, Lincoln, NE, United States

SAM ALBIN, University of Nebraska–Lincoln, Lincoln, NE, United States

HUIJUN ZHU, University of Nebraska–Lincoln, Lincoln, NE, United States

FRANK K WUERTHWEIN, San Diego Supercomputer Center, San Diego, CA, United States

MAHIDHAR TATINENI, San Diego Supercomputer Center, San Diego, CA, United States

View all

Open Access Support provided by:

University of Nebraska–Lincoln

San Diego Supercomputer Center

University of California, San Diego

PDF Download
3708035.3736060.pdf
23 January 2026
Total Citations: 2
Total Downloads: 879

Published: 20 July 2025

Citation in BibTeX format

PEARC '25: Practice and Experience in Advanced Research Computing
July 20 - 24, 2025
Ohio, Columbus, USA

Conference Sponsors:
SIGHPC

https://doi.org/10.1145/3708035.3736060
ISBN: 9798400713989

The National Research Platform: Stretched, Multi-Tenant, Scientific Kubernetes Cluster

Derek Weitzel
University of Nebraska - Lincoln
Lincoln, NE, USA
dweitzel@unl.edu

Ashton Graves
University of Nebraska - Lincoln
Lincoln, NE, USA
agraves10@unl.edu

Sam Albin
University of Nebraska - Lincoln
Lincoln, NE, USA
sam.albin@unl.edu

Huijun Zhu
University of Nebraska - Lincoln
Lincoln, NE, USA
huijun.zhu@unl.edu

Frank Wuerthwein
San Diego Supercomputer Center
La Jolla, CA, USA
fkw@ucsd.edu

Mahidhar Tatineni
San Diego Supercomputer Center
La Jolla, CA, USA
mahidhar@sdsc.edu

Dmitry Mishin
San Diego Supercomputer Center
La Jolla, CA, USA
dmishin@ucsd.edu

Elham Khoda
San Diego Supercomputer Center
La Jolla, CA, USA
ekhoda@ucsd.edu

Mohammad Sada
San Diego Supercomputer Center
La Jolla, CA, USA
mfsada@ucsd.edu

Larry Smarr
University of California, San Diego
La Jolla, CA, USA
lsmarr@ucsd.edu

Thomas DeFanti
University of California, San Diego
La Jolla, CA, USA
tdefanti@ucsd.edu

John Graham
San Diego Supercomputing Center
La Jolla, CA, USA
jjgraham@ucsd.edu



Figure 1: Map of NRP sites throughout the US. Not pictured are the several international locations.

Abstract

The National Research Platform (NRP) represents a distributed, multi-tenant Kubernetes-based cyberinfrastructure designed to facilitate collaborative scientific computing. Spanning over 75 locations in the U.S. and internationally, the NRP uniquely integrates varied computational resources, ranging from single nodes to extensive GPU and CPU clusters, to support diverse research workloads including advanced AI and machine learning tasks. It emphasizes flexibility through user-friendly interfaces such as JupyterHub and low level control of resources through direct Kubernetes interaction. Critical operational insights are discussed, including security

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

PEARC '25, Columbus, OH, USA

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1398-9/25/07
https://doi.org/10.1145/3708035.3736060

enhancements using Kubernetes-integrated threat detection, extensive monitoring, and comprehensive accounting systems. This paper highlights the NRP's growing importance and scalability in addressing the increasing demands for distributed scientific computational resources.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; **Machine learning**; • **Computer systems organization** → **Dependable and fault-tolerant systems and networks**.

Keywords

Distributed Computing, Kubernetes, High Throughput Computing, Artificial Intelligence

ACM Reference Format:

Derek Weitzel, Ashton Graves, Sam Albin, Huijun Zhu, Frank Wuerthwein, Mahidhar Tatineni, Dmitry Mishin, Elham Khoda, Mohammad Sada, Larry Smarr, Thomas DeFanti, and John Graham. 2025. The National Research Platform: Stretched, Multi-Tenant, Scientific Kubernetes Cluster. In *Practice and Experience in Advanced Research Computing (PEARC '25), July 20–24, 2025, Columbus, OH, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3708035.3736060

1 Introduction

The NSF supported research computing community has established infrastructures tailored to different levels of contributions. ACCESS [2] provides an umbrella organization primarily for large resources, each of which are located in a single data center and run their own batch schedulers. The OSG [14] federates clusters in different locations worldwide that each have their own batch scheduler. The OSG then overlays a fabric of services on top to federate the individual clusters to enable distributed High Throughput Computing[15]. The National Research Platform (NRP) federates individual compute and storage nodes into a global scale resource cluster across an a priori unlimited number of locations. The NRP thus operates at levels in the vertical software stack below either OSG or ACCESS. The entirety of the NRP, encompassing all globally distributed hardware locations, currently presents as a single 'batch cluster' to users.

The NRP has attracted a growing number of users. In calendar year 2024, there were 670 NRP Nautilus Namespaces that used 5M GPU-hours and 50M CPU Core-Hours on the NRP. These Namespaces can be divided into 4 main categories: Community Software (2), Observatories (13), NRP Operations (130), Individual campus researchers (525). These latter 525 Namespaces consumed >3/4 of the GPU-Hrs and 2/3 of the CPU-hrs used by all 670 NRP Nautilus Namespaces active in 2024. The individual PIs of these 525 researcher Namespaces came from 50 campuses, including 23 Minority Serving Institutions. These campuses include ones located in a dozen NSF designated Established Program to Stimulate Competitive Research (EPSCoR) [12] States. The campuses span from Carnegie R1/R2 campuses to community colleges. This broad national breadth of campuses are critical to the future development of a US trained AI/ML workforce.

The NRP can federate resources ranging from a single node to large collections of resources located in the same data center, enabling groups to contribute hardware according to their capacity. NRP offers integration at either the IPMI or the Kubernetes[8] layer, thus releasing the resource owners of the burden of system and cluster administration, in addition to offering higher level services. The National Research Platform (NRP) originated from the Pacific Research Platform [16], which began as a testbed for evaluating network connectivity across multiple sites. It has since evolved into a specialized platform focused on GPU hosting for Artificial Intelligence (AI) workloads. Additionally, the NRP provides educational support through fully-managed Jupyter notebook environments, LLM as a service, training resources, and curated containers. It also fosters community formation through a chat platform implemented via Matrix support channels.

Each institution contributes hardware to the NRP for its own reasons. The reduced burden of system administration attracts both institutions with limited resources as well as institutions for which research computing is not a primary focus (e.g. regional networking organizations and Internet2). Other institutions are attracted by the higher level services and the community building aspects of NRP. For example, institutions like the University of Missouri participate because their researchers already utilize the NRP and wish to expand their usage, including experimenting with new and/or unique hardware configurations [7]. For these researchers, the capability to run workloads on their dedicated nodes while seamlessly extending to additional research computing resources significantly amplifies their scientific impact. In addition, a number of projects use NRP as a distributed service deployment platform to build higher level services on top of NRP.

Some sites contribute hardware to meet funding requirements, such as the 20% resource-sharing mandate in recent Campus Cyberinfrastructure (CC*) [11] solicitations. In these cases, the contributed hardware is shared among NRP users, and in return, the NRP user-support team actively assists these sites by training their local users. Additionally, the NRP systems team supports the site's local management of the hardware and ensures compliance with any required institutional policies, monitoring security, and keeping software at all systems levels up-to-date. Increasingly, we see the community building aspects of NRP to lead to one institution hosting equipment bought by another institution, thereby contributing to the NRP community owned cluster by providing rackspace, power, and networking.

The continued growth in the NRP user community lead to the implementation of a richer set of policies. This includes a resource allocation mechanism for the most heavily in demand GPUs, an Acceptable Use Policy, and enhancements to security, monitoring, and their automation, as detailed in Section 4. Overall, the NRP continues to grow in both users and resources, while providing innovative offerings such as hosted LLMs[1] and other shared services.

The remainder of this paper describes the operations of the NRP, a distributed, multi-tenant Kubernetes cluster designed for scientific computing, and shares our experiences in supporting users to effectively leverage the platform.

2 Distributed Deployment

The Network Research Platform consists of over 420 nodes distributed across more than 70 sites as shown in Figure 1. These sites

[1] Including those that provided assistance in editing this paper!

include R1 institutions such as the University of California, San Diego, R2 institutions like the University of South Dakota, and network points-of-presence (PoPs) that provide high-quality network connectivity.

Managing and deploying the NRP management software across multiple administrative domains poses significant challenges. To overcome these hurdles, we utilize Ansible [6], a powerful automation platform, to streamline the deployment and maintenance of our services. Specifically, Ansible simplifies the installation process of new nodes by automatically configuring and installing necessary software, including:

- **Containerization**: The NRP installs the container runtime which will run the user pods.
- **Kubernetes**: The orchestration layer that the NRP uses to schedule workflows.
- **System settings**: The NRP sets numerous settings on the system such as increasing the default MTU on the network interface to support jumbo frames and turning on automatic security package updates.

Integration of resources into the NRP can happen at many different layers.

(1) **Hardware-Level Integration** - NRP administrators directly manage the node hardware using the Intelligent Platform Management Interface (IPMI), including installing and updating firmware and operating systems. For security, sites must restrict IPMI access exclusively to the designated NRP gateway through firewall policies.
(2) **Operating System-Level Integration** - Sites provide nodes pre-installed with NRP's preferred operating system (Ubuntu). NRP administrators then install the necessary software with Ansible, integrate the nodes into the cluster, and handle ongoing OS maintenance and updates.
(3) **Peering Kubernetes** - In special cases, the NRP integrates resources by establishing peer relationships with external Kubernetes clusters using Admiralty[1]. This allows secure, seamless sharing and processing of workloads across clusters.

In order to keep track of all of the nodes in the NRP, we use NetBox[9] as the source of truth for tracking hardware and network devices. It maintains an up-to-date inventory of nodes, interfaces, and other critical infrastructure. To automate the process of adding hardware, we deploy a customized version of the netbox-agent [17] on each NRP node. Additionally, NetBox's journal feature is integrated with our Ansible workflows to log hardware changes and maintenance events, providing a transparent and auditable history of all actions performed on the infrastructure.

3 User Interface

Users primarily interact with the NRP either directly via Kubernetes or through gateways such as JupyterHub [5] and Coder [3]. To gain Kubernetes access, users log into the NRP portal to download a personalized Kubernetes configuration file containing Kubernetes RBAC-compliant OIDC tokens issued via CILogon federated authentication. Users are restricted to Kubernetes namespaces specific to their research groups, limiting their actions accordingly to within their namespace. Direct Kubernetes API interaction through command-line tools (e.g., `kubectl`) enables fine-grained resource management.

For interactive usage, the NRP hosts centralized instances of JupyterHub and Coder. The public JupyterHub instance simplifies container management through predefined hardware profiles, allowing users to easily select CPU, memory, and accelerator configurations while automatically provisioning persistent storage. It supports various software environments, including MATLAB, R, Python, X11 desktops, PyTorch, and TensorFlow. Similarly, Coder provides users with browser-based GUI Desktop/VS Code environments, complete with pre-configured persistent storage and templates for FPGA/GPU tools and libraries. Additionally, NRP documentation guides research groups in deploying customized JupyterHub and Coder instances within dedicated namespaces, while a hosted GitLab instance facilitates CI/CD workflows and Docker image management.

3.1 Computational Resources

As of Spring 2025, the NRP comprises in excess of 1,400 GPUs, 28,000 CPUs, and 161 TB of RAM distributed across more than 420 nodes. The GPU models available vary considerably, ranging from the older NVIDIA RTX 2080 Ti to the state-of-the-art Grace Hopper H200, with numerous intermediate models. Users can specify preferred GPU models or provide a list of acceptable GPUs. The Kubernetes scheduler then allocates resources accordingly.

With the rising demand for high-end GPUs, such as the growing demand shown in Figure 2, the NRP implemented an allocation system aimed at optimizing GPU utilization. Following the introduction of the reservation system for A100 GPUs, average GPU utilization across the cluster improved significantly from 21.49% to 31.37%. Although there is still room for further enhancement, this marked improvement underscores the effectiveness of the reservation system.

In addition to traditional NRP workloads submitted via Kubernetes or managed through Jupyter notebooks, the NRP cluster is also utilized by low-priority, preemptible jobs sourced from the OSG. This strategy maintains consistent cluster activity by running tasks that can be easily interrupted if higher-priority workloads arise, simultaneously offering substantial resources to the OSG community. In 2024 alone, the NRP contributed 6.5 million CPU hours to OSG projects.

3.2 Storage

The National Research Platform employs diverse storage technologies, primarily relying on Ceph[18], a distributed filesystem orchestrated by the Rook operator, to deliver substantial storage capacity. Users access Ceph storage via Kubernetes Persistent Volumes. They can choose CephFS, a shared filesystem with managed namespaces, or Ceph Block Devices, ideal for handling smaller files. The distributed nature of NRP poses both challenges and opportunities. Among the challenges is the inherent data access latencies in a globally distributed system. To ensure minimal latency and optimal performance for users across the globe, Ceph storage is strategically distributed across five U.S. regions accessible to general users: West Coast, Central (Great Plains), East Coast, Southeast (Florida), and Pacific (including Guam and Hawaii). Users select storage regions,

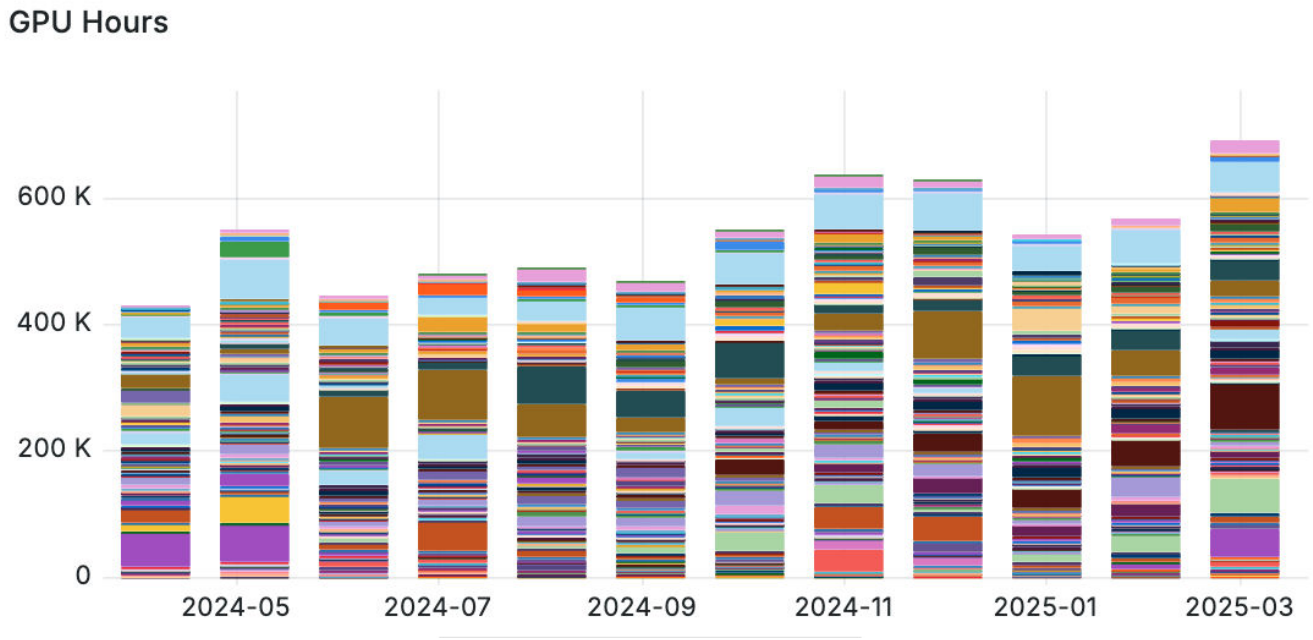


Figure 2: GPU hours by research group on the NRP over time showing growth of GPU use. For the 12 month period, the total is 6,605,042 GPU hours from 412 research groups.

aligning computational workloads to the same region, minimizing latency and optimizing performance.

The NRP minimizes the impact of networking, power, or other physical failures in one location by distributing storage across multiple locations, and configuring each regional Ceph cluster to distribute data replicas across multiple physical locations. Any one location may experience an outage without leading to any filesystem outage. This strategy increases resilience while distributing the network and IO load across a region.

4 Security

One significant challenge for our project was effectively detecting and responding to malicious activities, particularly incidents involving cryptominers executed through unsecured user-operated services like JupyterHub instances. Our initial detection strategy relied on Prometheus alerts triggered by node overloads, which often represented legitimate usage, underscoring the need for explicit alerts pinpointing malicious activity. After evaluating several options, we selected Falco[4], an open-source threat detection tool sponsored by the Cloud Native Computing Federation (CNCF), widely adopted by the Kubernetes community.

Falco integrates with Kubernetes and container runtimes, capturing detailed process information (e.g., commands, arguments, hierarchies) along with metadata such as container images, namespaces, and pod identifiers. Its extensible architecture enables the implementation of custom and community-developed rules, significantly enhancing our ability to rapidly identify and respond to cryptomining and other malicious threats within the NRP environment.

5 Accounting and Measurements

Initially, we used standard Kubernetes tools (Prometheus, Thanos, Grafana) for accounting and monitoring. As data demands increased, Grafana became insufficient, prompting us to switch to ObservableHQ [13] for interactive, JavaScript-based dashboards. To overcome Thanos scaling limitations with accounting queries, we implemented custom caching, query segmentation, and a Redis caching layer – significantly improving performance and stability.

Beyond accounting, we maintain continuous network and system monitoring. We use sFlow for detailed network flow data, analyzed via ElastiFlow and InMon, and deploy distributed PerfSONAR instances across all nodes to consistently track network bandwidth and connectivity.

6 Future Directions

The default Kubernetes scheduler's FIFO behavior limits the NRP's capability to handle diverse scientific workloads optimally. To overcome this, we have begun experimenting with advanced scheduler plugins such as YuniKorn[19], which introduces sophisticated scheduling capabilities including quotas, reservations, and fair-share scheduling. Our ongoing collaboration with the YuniKorn development community aims to resolve identified issues and enhance scheduler performance.

The NRP is increasingly expanding to include smaller educational institutions, such as smaller R2 universities and community colleges like the San Diego Community College District. These institutions are not only contributing resources but also utilizing the NRP to support instructional programs in AI and machine learning, a trend we anticipate will continue to grow, broadening access and advancing workforce training across educational contexts.

Acknowledgments

This work was funded by the NSF awards 2112167, 2120019. This paper was edited using LLMs hosted on the NRP [10] as well as commercial LLMs.

References

[1] Admirality. 2025. *Multi-Cluster Kubernetes. Simplified. | Admiralty.* Admirality. https://admiralty.io/

[2] Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. 2023. ACCESS: Advancing Innovation: NSF's Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support. In *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good* (New York, NY, USA, 2023-09-10) *(PEARC '23)*. Association for Computing Machinery, 173–176. doi:10.1145/3569951.3597559

[3] Coder. 2025. *Coder - Cloud Development Environment: Remote & Self Hosted.* https://coder.com/

[4] Falco. [n. d.]. *Falco.* https://falco.org/

[5] Brian E. Granger and Fernando Pérez. [n. d.]. Jupyter: Thinking and Storytelling With Code and Data. 23, 2 ([n. d.]), 7–14. doi:10.1109/MCSE.2021.3059263 Conference Name: Computing in Science & Engineering.

[6] Red HAt. [n. d.]. *Ansible Collaborative.* https://www.redhat.com/en/ansible-collaborative

[7] J. Alex Hurt, Grant J. Scott, Derek Weitzel, and Huijun Zhu. [n. d.]. Adventures with Grace Hopper AI Super Chip and the National Research Platform. doi:10.48550/arXiv.2410.16487 arXiv:2410.16487 [cs]

[8] Kubernetes. [n. d.]. *Kubernetes.* https://kubernetes.io/

[9] NetBox Labs. [n. d.]. *Enterprise-Grade, SaaS Network Source of Truth.* https://netboxlabs.com/

[10] NRP. [n. d.]. *NRP-Managed LLMs.* https://nrp.ai/documentation/userdocs/ai/llm-managed/

[11] NSF. 2025. *Campus Cyberinfrastructure (CC*) | NSF - National Science Foundation.* https://www.nsf.gov/funding/opportunities/cc-campus-cyberinfrastructure

[12] NSF. 2025. *Established Program to Stimulate Competitive Research (EPSCoR) | NSF - National Science Foundation.* https://www.nsf.gov/funding/initiatives/epscor

[13] ObservableHQ. [n. d.]. *Observable: Build expressive charts and dashboards with code.* https://observablehq.com/

[14] The Open Science Grid Executive Board on behalf of the Osg Consortium:Ruth Pordes, Don Petravick, Bill Kramer, Doug Olson, Miron Livny, Alain Roy, Paul Avery, Kent Blackburn, Torre Wenaus, Frank Würthwein, Ian Foster, Rob Gardner, Mike Wilde, Alan Blatecky, John McGee, and Rob Quick. [n. d.]. The open science grid. 78, 1 ([n. d.]), 012057. doi:10.1088/1742-6596/78/1/012057

[15] Rajesh Raman, Miron Livny, and Marvin Solomon. 2000. Resource management through multilateral matchmaking. In *Proceedings the Ninth International Symposium on High-Performance Distributed Computing*. IEEE, 290–291.

[16] Larry Smarr, Camille Crittenden, Thomas DeFanti, John Graham, Dmitry Mishin, Richard Moore, Philip Papadopoulos, and Frank Würthwein. [n. d.]. The Pacific Research Platform: Making High-Speed Networking a Reality for the Scientist. In *Proceedings of the Practice and Experience on Advanced Research Computing: Seamless Creativity* (New York, NY, USA, 2018-07-22) *(PEARC '18)*. Association for Computing Machinery, 1–8. doi:10.1145/3219104.3219108

[17] Solvik. [n. d.]. *Solvik/netbox-agent.* https://github.com/Solvik/netbox-agent

[18] 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.

[19] YuniKorn. [n. d.]. *Welcome to Apache YuniKorn Apache YuniKorn.* https://yunikorn.apache.org/