K2S-Gate: A Secure Kubernetes-to-Slurm Gateway with Unified Service Catalogs for Web-based AI and HPC Applications

Gukhua Lee
Supercomputing Infrastructure Center
Korea Institute of Science and Technology Information *(KISTI)*
Deajeon, Republic of Korea
ghlee@kisti.re.kr

Joon Woo
Supercomputing Infrastructure Center
Korea Institute of Science and Technology Information *(KISTI)*
Deajeon, Republic of Korea
wjnadia@kisti.re.kr

Taeyoung Hong
Supercomputing Infrastructure Center
Korea Institute of Science and Technology Information *(KISTI)*
Deajeon, Republic of Korea
tyhong@kisti.re.kr

***Abstract*—The rapid growth of AI and high-performance computing (HPC) applications demands seamless integration of cloud-native and HPC environments, yet heterogeneous software stacks keep these systems isolated, limiting unified deployment of and access to web-based applications. This work presents K2S-Gate, a secure Kubernetes-to-Slurm gateway that bridges cloud-native and HPC infrastructures through a unified web platform. K2S-Gate introduces a job-to-service transformation paradigm that converts Slurm batch jobs into dynamically accessible web services, enabling one-click execution on HPC compute nodes without SSH tunneling or port forwarding. It also provides a template-based Unified Service Catalog for deployment across heterogeneous environments. For secure, scalable access, K2S-Gate integrates a Secure Access Layer combining single sign-on (SSO), an OAuth2-based identity provider (IdP), and an Istio ingress gateway. Validated on the KISTI-5 supercomputer and integrated into the MyKSC portal, K2S-Gate demonstrates its practicality in real-world HPC environments.**

Keywords—Kubernetes-to-Slurm, job-to-service, web-based application, secure gateway, HPC system, cloud-native

I. SUMMARY

The increasing use of AI and HPC workloads has created a growing demand for the efficient utilization of both cloud-native and HPC infrastructures [1]. Cloud platforms based on Kubernetes provide flexible and interactive environments for running applications, whereas HPC systems typically use batch schedulers such as Slurm to manage large-scale computational workloads. However, integrating these two environments remains challenging because they use different software stacks, execution models, and access mechanisms. As a result, cloud and HPC resources are often managed and used as separate computing environments.

Several efforts have attempted to integrate Kubernetes and HPC environments, but most focus on specific aspects of the integration. At the infrastructure level, SchedMD's Slinky runs Slurm components within Kubernetes and manages the Slurm controller and compute daemons using Kubernetes operators and custom resources [2]. CoreWeave's SUNK takes a different approach by integrating Slurm with Kubernetes scheduling, allowing Slurm and Kubernetes workloads to share computing resources [3]. These approaches provide infrastructure-level integration between Slurm and Kubernetes, but they mainly focus on resource deployment and scheduling rather than providing HPC applications as web-based services. At the user-access level, Open OnDemand provides browser-based access to interactive HPC applications, including Jupyter and remote desktop environments [4]. However, its architecture primarily focuses on providing a web portal for accessing HPC resources rather than providing a common service model across Kubernetes and HPC environments. Other studies have explored running cloud-native workloads on HPC systems [1] and scaling AI inference workloads across Kubernetes and Slurm environments [5]. Although these approaches improve interoperability between cloud-native and HPC systems, there is still a need for a unified mechanism that allows web-based applications to be deployed and accessed consistently across both environments. K2S-Gate addresses this gap by connecting Kubernetes-based service management with Slurm-based HPC execution and providing a common service abstraction for web-based AI and HPC applications.

To address this limitation, we propose K2S-Gate, a secure Kubernetes-to-Slurm gateway that connects cloud and HPC environments through a unified application service model. The main contribution of K2S-Gate is a job-to-service transformation mechanism that converts a Slurm batch job into a web-accessible service at runtime. This mechanism allows users to access and interact with applications running on HPC compute nodes through a web interface, similar to cloud-native applications. As a result, users can utilize HPC resources without directly dealing with the underlying batch scheduling and access mechanisms.

The architecture of K2S-Gate consists of three main components: a gateway layer, a unified service catalog, and a secure access layer. The gateway layer connects Kubernetes and Slurm by translating application requests into Slurm jobs. When a user selects an application and its resource configuration through the web portal, K2S-Gate generates a Slurm job script and submits the job through the Slurm REST API. The application is then scheduled on an HPC compute node and executed in a containerized environment using Singularity. To provide web-based access to the running application, K2S-Gate uses a callback and dynamic routing mechanism. Once the application starts and becomes ready, the compute node sends runtime information, including the host, IP address, port, and access token, back to K2S-Gate. Based on this information, K2S-Gate dynamically creates Istio ServiceEntry and VirtualService resources to connect the application running on the HPC compute node to the Kubernetes networking layer. User requests are then routed through the Istio ingress gateway to the corresponding application. This approach provides secure access to HPC applications without directly exposing the compute nodes to users.

The Unified Service Catalog provides predefined YAML-based templates that define application deployment, resource requirements, and routing configurations. These templates hide the underlying infrastructure details and allow users to launch applications such as JupyterLab and Visual Studio Code through a simple web interface. By using standardized application definitions, K2S-Gate provides a consistent way to deploy and manage applications across different execution environments.

For secure access, K2S-Gate provides a Secure Access Layer that integrates SSO, an OAuth2-based IdP, and the Istio ingress gateway. All external requests are authenticated and routed through the gateway, so HPC compute nodes do not need to be directly exposed to users. In addition, token-based access control is used to restrict each application session to the authorized user. This design provides isolated access to individual application instances and supports multiple users in a shared HPC environment.

The proposed system was implemented as part of MyKSC, the web-based service portal for the KISTI supercomputing infrastructure, and validated on the Neuron GPU system. The implementation demonstrates that K2S-Gate simplifies access to HPC resources by allowing users to launch web-based applications with a single click. Users can access these applications through a web browser without manually configuring SSH tunnels or port forwarding.

K2S-Gate also extends the cloud-native application model to HPC systems by providing interactive, service-oriented access to applications executed as Slurm jobs. This approach makes HPC applications easier to access while preserving the existing Slurm-based resource management environment. K2S-Gate is currently being prepared for deployment on KISTI-6 Hangang, with future extensions planned for workload orchestration, user-specific environment management, and intelligent support services.

Overall, K2S-Gate provides a practical approach to connecting cloud-native and HPC environments. By combining Slurm-based job execution with Kubernetes-based service management and secure web access, the system enables AI and HPC applications to be deployed and accessed through a consistent web-based interface.

ACKNOWLEDGMENT

This research was supported by Korea Institute of Science and Technology Information(KISTI) (No. (KISTI) K26L1M2C2).

REFERENCES

[1] A. Chazapis, E. Maliaroudakis, M. Marazakis, F. Nikolaidis, and A. Bilas, “Running Cloud-Native Workloads on HPC with High-Performance Kubernetes,” *arXiv preprint arXiv:2409.16919*, 2024. Available: https://arxiv.org/abs/2409.16919

[2] S. Malinowski, A. Mutschelknaus, M. Warnicke, and T. Wickberg, "Slinky: The Missing Link Between Slurm and Kubernetes," presented at the Cray User Group Conference (CUG 2025), May 2025. [Online]. Available: https://slurm.schedmd.com/MISC25/Slinky-CUG2025.pdf

[3] CoreWeave, "SUNK: Slurm on Kubernetes," CoreWeave Documentation. [Online]. Available: https://docs.coreweave.com/docs/products/sunk

[4] D. Hudak, D. Johnson, A. Chalker, J. Nicklas, E. Franz, T. Dockendorf, and B. L. McMichael, "Open OnDemand: A web-based client portal for HPC centers," J. Open Source Softw., vol. 3, no. 25, p. 622, 2018, doi: 10.21105/joss.00622.

[5] T. Trappen, R. Keßler, R. Pabel, V. Achter, and S. Wesner, "Automated Dynamic AI Inference Scaling on HPC-Infrastructure: Integrating Kubernetes, Slurm and vLLM," in *Proc. 1st Int. Workshop Next-Gen Middleware MLOps Distrib. Syst. (MIND '25)*, Dec. 2025, pp. 13–18, doi: 10.1145/3774902.3776632.