SAND2025-01494C

Prototyping and Characterization of Advanced Environments for AI+HPC on Commodity Systems

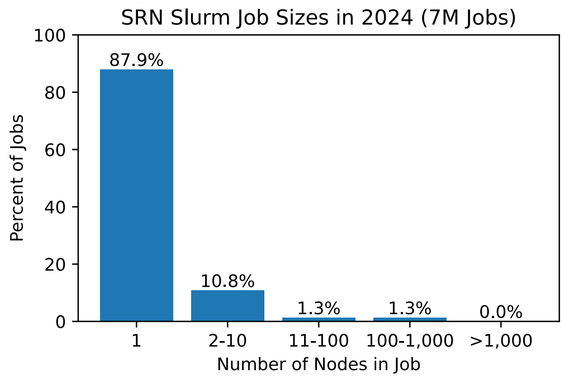
Craig Ulmer, Ben Schwaller, Cory Lueninghoener, Patrick Carlson, Ryan Prescott, and Jim Brandt

With a Multi-division team from 5500, 8700, 9300, and 9700

Problem: AI Workloads and economics are driving platform architects towards *multi-tenancy*

Abundance of Small Jobs

Analysis of 2024 SRN Capviz cluster jobs revealed 77% ran less than five minutes and 88% used only one node.



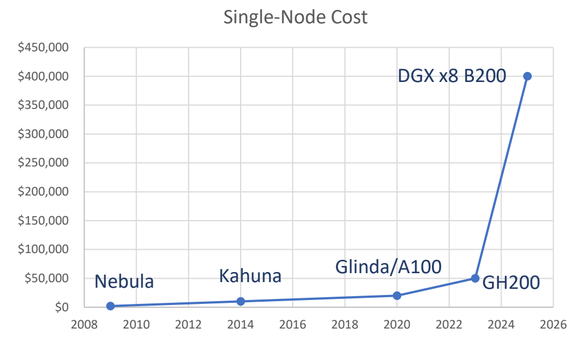
Emerging GPU Demands

AI workloads use GPUs in different ways:

- **Small-scale R&D:** Kahuna users employ 1-16 GPU nodes for inference and small model development.
- **Large-scale Training**: Multiple mission spaces have identified a need to use large DGX GPU platforms for training.
- **Production Inference**: Atlas/Shirty use microservices to allow developers to embed AI in their applications without requiring the user to have a local GPU.

Large Node Economics

DGX nodes with 8x GPUs offer the best option for training and can be networked together for massive jobs. However, they are expensive: $400k/node. Thus, maximizing utilization is essential.



Multi-use AI/HPC Platform

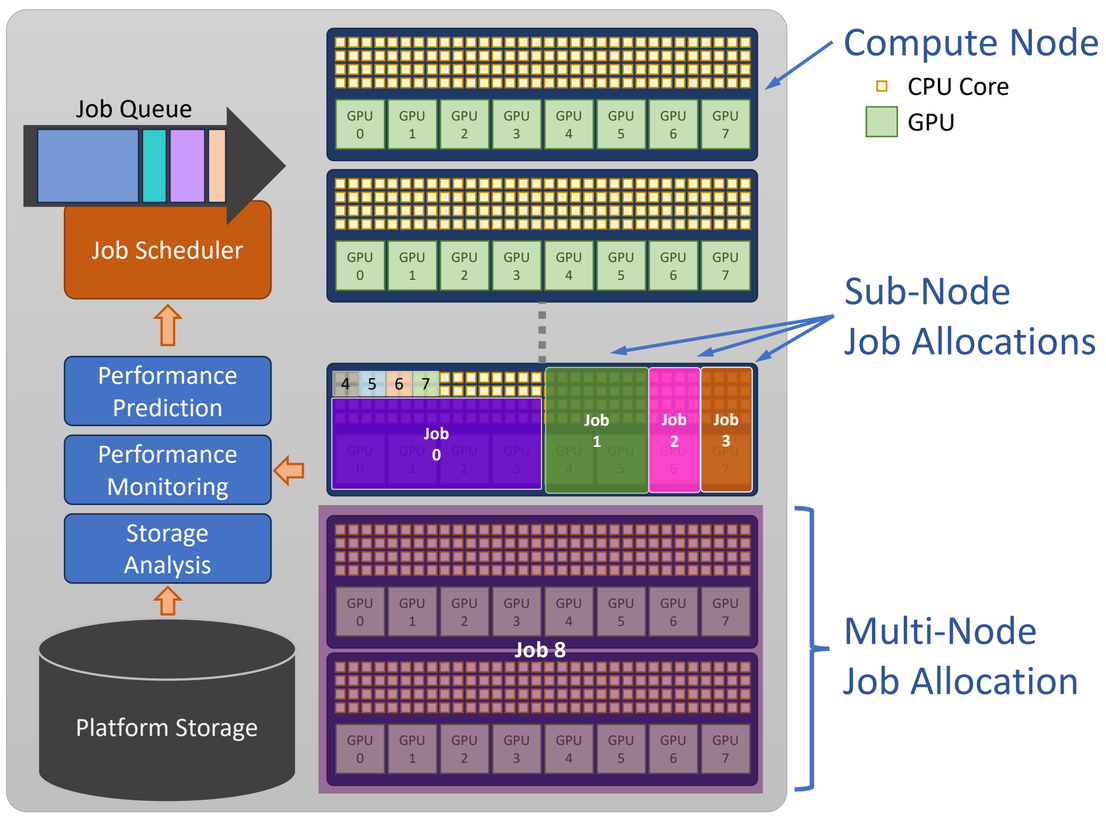
Ideally, a single platform with large resources can be architected to serve the AI needs of multiple communities. It is unclear whether Sandia's current approaches to managing HPC/HPDA Slurm clusters will be efficient enough.

Systems research must investigate:

- Mechanisms to share compute node resources among multiple users.
- Aggressive job analytics to ensure high resource utilization.

Multi-tenant Vision *How can we share node resources?*

Ann Gentile, Dena Vigil, Brian Adams, Jim Brandt, Cara Corey, and Chris Cuellar



Potential Mechanisms

Slurm Node Sharing

While Sandia HPC platforms typically allocate at the node granularity, Slurm partitions can be configured to allow multiple users to share a node. Resources are constrained via cgroups.

Virtualization

Open source virtualization tools provide a well-defined path for administrators to split a node's resources with hard constraints.

Kubernetes

Kubernetes provides a robust platform for scheduling containers on resources. Recent support for Flux enables tasks to be mapped to nodes.

Slurm VM Prototypes

Cory Lueninghoener and Patrick Carlson

A preliminary investigation of commercial and open source tools revealed different paths for on-prem cloud experiments:

- Commercial hardware/software products such as Oxide offer a complete solution with fine-grain controls.
- Commercial software products such as OpenShift are in use at Sandia and could be extended for GPU use, but may be *cost prohibitive*.
- Open Source software such as Incus, Terraform, and Ansible provide a DIY means of creating a low-cost prototype.

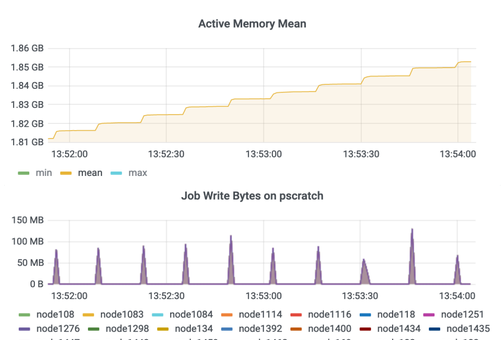
Experiments with an Incus cloud demonstrated that VMs could serve as Slurm compute nodes. However, additional work with optimizing InfiniBand and GPU use is required.

Job Understanding *How do we infer what resources a job really needed?*

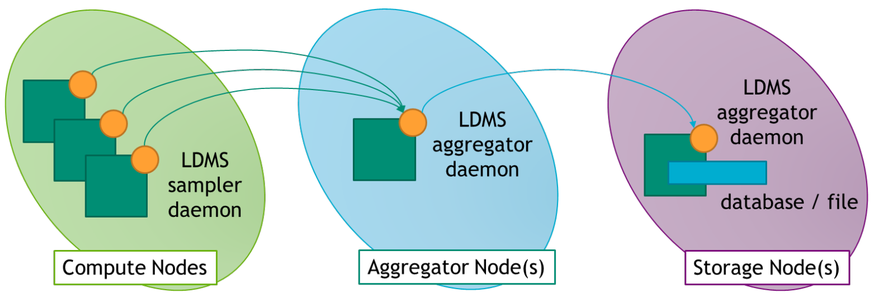
LDMS and AppSysFusion

Ben Schwaller and Jim Brandt

The Lightweight Distributed Metric Service collects fine-grained details about the resources used by a job. AppSysFusion combines system and application data collected by LDMS into a single analysis plane.

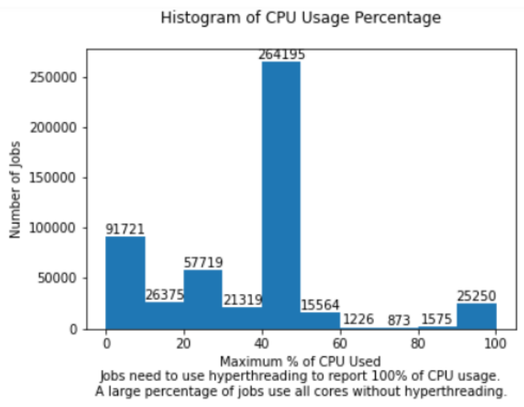


EMPIRE Run Memory Leak and Filesystem Write Behavior



Discovering Underutilization

LDMS summarizations can reveal jobs with underutilized resources such as memory and CPU.

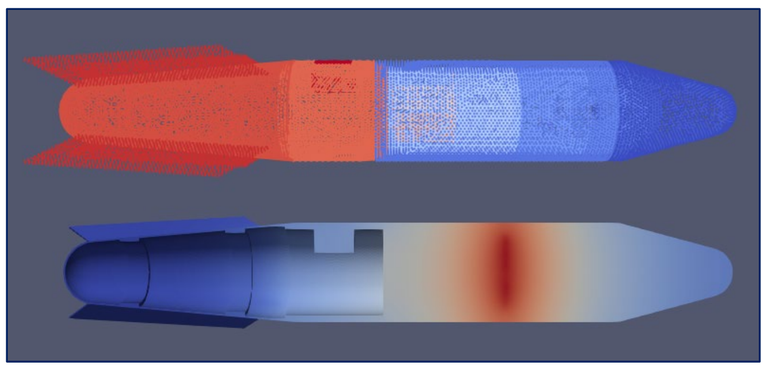


Leveraging AI to Summarize Workflow Artifacts

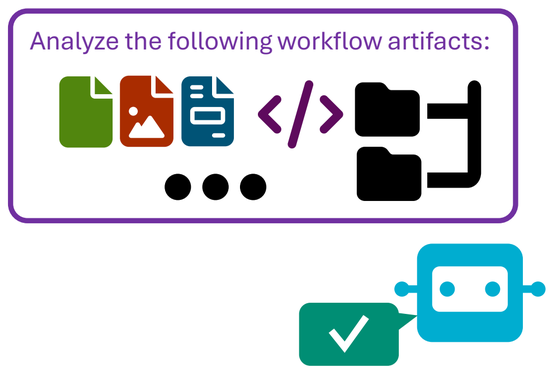
Luke McCormick, Kevin Olson, and Craig Ulmer

Workflows generate a large number of file artifacts that document what took place during an analysis. While these files have valuable information about user intent and workflow performance, the sheer number and variety of files makes it difficult for humans to consume.

We are leveraging Chain-of-Thought AI techniques to interrogate a collection of workflow files and summarize what happened. An agent walks a SAW project directory, calls tools to digest known formats, and collects notes that are later summarized.



SAW's Unclassified Demo Project



Which file artifacts are most important?
Which modsim tools were used?
How many CPU/GPUs did each task use?
What mesh simplifications were made?
Summarize what this job did.
....

The `rocket.asm` file is an assembly file, which suggests it contains information about how different parts or components are assembled or connected together. Given its presence in the `cad-files` folder, it is likely related to the CAD...

SAND NUMBER: SAND2025-01494C