EVALUATING THE USAGE OF PYTHON LIBRARIES ON A PRODUCTION SUPERCOMPUTER

Thomas M. Papka
Data Science, Loyola University / Data Sciences and Learning, Argonne National Laboratory
tpapka@loyola.edu

Venkatram Vishwanath and Filippo Simini
Argonne Leadership Computing Facility, Argonne National Laboratory
venkat@anl.gov and fsimini@anl.gov

ABSTRACT

Advances in AI and ML are transforming scientific computing and shaping Python library usage in HPC environments. On **Polaris**, Argonne's GPU-accelerated supercomputer (44 PF peak, 560 NVIDIA A100 GPUs, 2,240 AMD EPYC CPUs), we analyzed Python usage trends using two datasets:

- **XALT** – detailed runtime monitoring of libraries, executables, and job metadata
- **PySnooper** – higher-level snapshots of library usage

By correlating these logs with scheduler data, we revealed **domain-specific adoption of AI/ML libraries**. Results are delivered via visualizations and a real-time dashboard, helping scientists:

- Track Python usage trends
- Identify performance impacts from non-optimized libraries
- Guide improvements to Conda environments and HPC infrastructure planning

This analysis provides **actionable insights** for software provisioning, user support, and enabling AI-driven science.

MOTIVATION

- Argonne's default Conda environment supports common workflows but usage analysis shows frequently used packages not included by default
- Understanding which domains customize the environment versus use it as-is guides software provisioning improvements
- Identifying the most commonly used libraries provides insight into the HPC software landscape

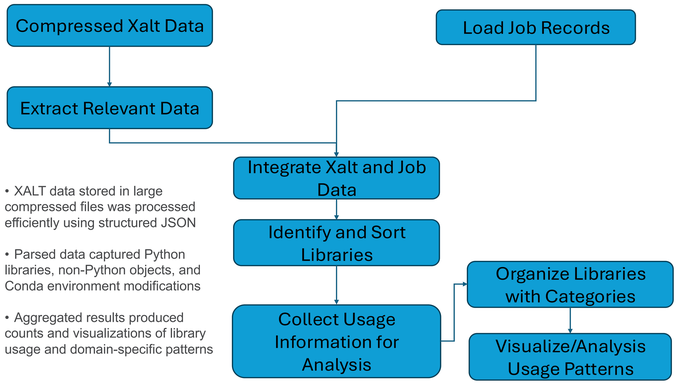
PYSNOOPER VERSUS XALT

PySnooper	XALT*
Lightweight, quick snapshots Captures basic library usage, execution paths, run location	**Comprehensive, detailed tracking** Includes all PySnooper data plus runtime details and non-Python processes
Lower detail, not all jobs captured	Captures nearly every Python job → robust usage trends
Minimal overhead → best for fast checks	Slightly higher overhead but still minimal relative to runtime

*With XALT enabled, library imports show minor overhead: +0.24s for a single library (16% increase) and +6.1s for 95 libraries (47% increase). The added cost is small and limited to the import phase, with negligible impact on overall runtime.

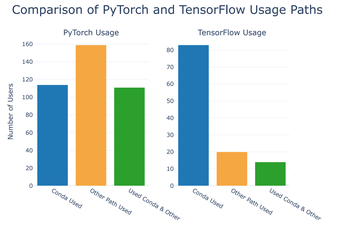
Polaris is a GPU-accelerated supercomputer located at Argonne National Laboratory, delivering over 44 petaflops of peak performance with 560 NVIDIA A100 GPUs and 2,240 AMD EPYC Milan CPUs. It supports large-scale modeling, simulation, data analysis, and AI/ML workloads for cutting-edge scientific research.

OVERVIEW OF METHOD

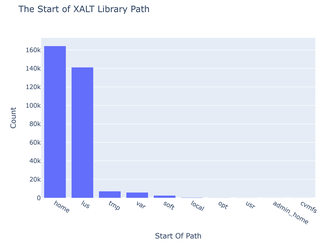


- XALT data stored in large compressed files was processed efficiently using structured JSON
- Parsed data captured Python libraries, non-Python objects, and Conda environment modifications
- Aggregated results produced counts and visualizations of library usage and domain-specific patterns

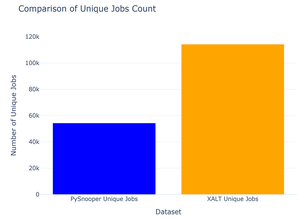
LIBRARY ENVIRONMENT USAGE



Bar charts **of PyTorch and TensorFlow users on Polaris by environment.** PyTorch users primarily run in custom environments (160), with 110 using the base Conda environment or both. TensorFlow users mostly run in the base environment (80), with 20 in other environments and 15 in both, showing a stronger reliance on the base environment.

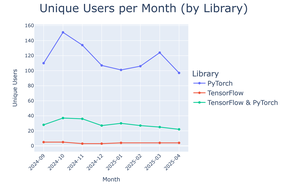


The graph shows the origins of library paths used by Polaris users**. Most libraries are loaded from home directories, which can negatively impact performance**.

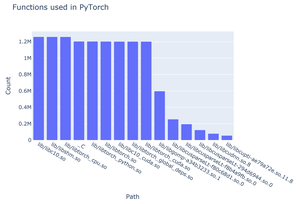


XALT identifies nearly twice as many unique jobs as PySnooper, showing its broader system-level coverage and highlighting the value of combining tools for comprehensive workflow analysis.

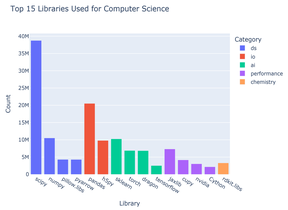
RESULTS



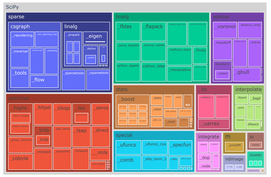
Most users rely on PyTorch, with few using TensorFlow, highlighting PyTorch's dominance and motivating deeper analysis of its usage.



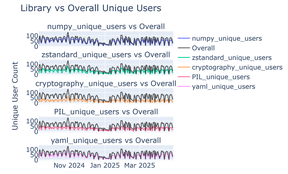
PyTorch usage centers on core tensor libraries, CUDA acceleration, and full runtime components supporting multiprocessing and dependencies.



Computer Science workloads are dominated by NumPy, SciPy, and Pandas, reflecting a focus on data analysis and AI.



SciPy usage is dominated by core linear algebra, sparse matrix operations, and Cython-based extensions for high-performance scientific computing.



Charts show NumPy, Zstandard, Cryptography, PIL, and YAML closely track overall Python usage, indicating broad adoption across Polaris.

FUTURE WORK

Future work will analyze runtime and job sizes for Python libraries and track emerging usage trends over time. We will also examine libraries outside the base Conda environment and investigate the factors driving library popularity to identify the most effective tools for scientific research. In addition, we plan to build an interactive dashboard for daily monitoring of library usage on Polaris. These efforts will provide deeper insight into scientific workflows and guide improvements to HPC software provisioning.

CONCLUSION

- 🧩 Clear patterns in Python library usage on Polaris
- 🚀 Core scientific, AI, and performance libraries dominate
- 📊 Trends over time guide Conda environment optimization
- ⚙️ Insights support more efficient scientific workflows
- 🗓️ Ongoing analysis improves HPC resource management
- 🔭 Deeper investigation enables broader AI adoption

Acknowledgment
This research used resources of the Argonne Leadership Computing Facility, a U.S. Department of Energy (DOE) Office of Science user facility at Argonne National Laboratory and is based on research supported by the U.S. DOE Office of Science-Advanced Scientific Computing Research Program, under Contract No. DE-AC02-06CH11357. Travel support provided by Loyola University and Argonne National Laboratory.

References
[1] Kapil Agrawal, Mark R. Fahey, Robert McLay, and Doug James. 2014. User Environment Tracking and Problem Detection with XALT. In 2014 First International Workshop on HPC User Support Tools. 32–40. https://doi.org/10.1109/HUST.2014.6
[2] Misha Salim and Kyle Felker. 2025. PyModuleSnooper. https://github.com/Argonne-lcf/PyModuleSnooper.