Modernizing ATLAS PanDA for a sustainable multi-experiment future

T. Korchuganova, A. Alekseev, F. H. Barreiro Megino, K. De, W. Guan, E. Karavakis, A. Klimentov, F. Lin, T. Maeno, P. Nilsson, T. Wenaus, Z. Yang, X. Zhao on behalf of the ATLAS Computing Activity

October 19-25, 2024 | 27th International Conference on Computing in High Energy & Nuclear Physics

Overview of PanDA system in the ATLAS experiment

PanDA - Production ANd Distributed Analysis workflow management system

In production for the ATLAS experiment since 2006, continuously enhancing its capabilities:

- Adapted to run seamlessly on a variety of platforms, including GRID, commercial cloud services*, and HPC facilities.
- Embracing job and hardware heterogeneity:
  - Successfully accommodated diverse job requirements such as multiprocessing, multithreading, and varying memory needs.
  - Expanded support for a wide range of processors, including ARM architectures, and integrated the use of accelerators like GPUs†.
- Demonstrated significant scalability, as ATLAS tripled its resource utilization over the past 10 years.

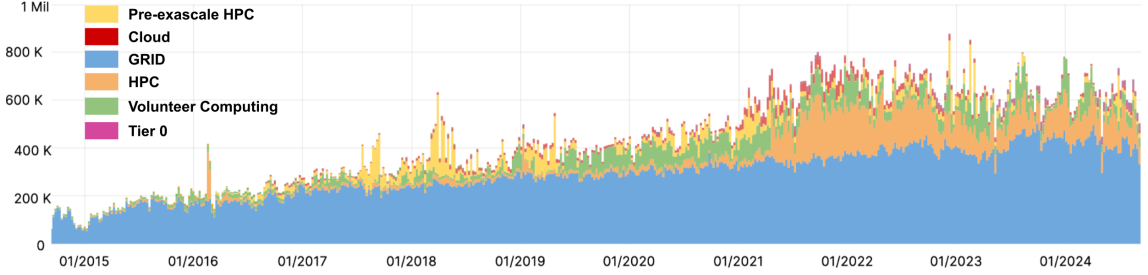

Figure 1 – # cores running ATLAS workflows weekly average

The success of the PanDA WMS has drawn interest from other experiments in High-Energy Physics (HEP) and beyond, fostering cross-experimental collaboration. The Vera Rubin Observatory currently uses PanDA WMS in production, and several other experiments are exploring its potential for future integration.

* "Total Cost of Ownership and Evaluation of Google Cloud Resources for the ATLAS Experiment at the LHC" plenary talk by David South

† "Using the ATLAS experiment software on heterogeneous resources" talk in Track 7 by Johannes Elmsheuser

Motivation

ATLAS Workflow Management System review

In early 2024, ATLAS conducted an architectural review to assess the functionality of its existing components within the workflow and workload management ecosystem.
The outcome is that the current system shows no apparent signs of scalability limitations or critical defects, several issues still require attention.

Recommendations:

- Streamline and modernize legacy code accumulated over nearly two decades of ATLAS operations.
- Maximize the use of continuous integration and testing frameworks to improve system reliability and development efficiency.
- Enhance decision-making by integrating greater workflow awareness into the system core.
- Develop a more user-friendly and accessible GUI to broaden its usability across different user groups.
- Implement dynamic, real-time monitoring to showcase system dynamism and responsiveness.
- Adopt a more structured and strategic approach to project management.
- Pursue R&D projects with clear long-term integration and operational goals to ensure sustainable development.

Introduced project management organization

Streams

1. Codebase optimization & modernization

- Code cleanup and modernization
- Enhancement of modularity
- DB simplification
- Explore CI/CD frameworks
- Standardization and auto-documentation of API

2. System optimization and enhancement

- General development
- Service metrics
- Speed up job generation
- Heterogeneity of resources
- PanDA queue and site objects

3. Outreach

- Documentation
- Community meetings
- Communication channels for announcements and issue tracking
- Video demos and tutorials
- Journal papers
- Revamp website

4. Interactive and dynamic workflow-oriented platform

- Technology choice and their scalability
- Integration of workflow execution engine
- Introducing dynamism in monitoring
- Unified and sophisticated interface
- Common authN/Z scheme

- Each stream has a manager who itemizes milestones, estimates time to complete, assigns tasks, and tracks their status.
- Streams cut across all components to enhance the mobility of developers and ensure a more flexible and integrated development process.
- Everybody in the team is involved in all the streams; for example, the manager of Stream 4 can be assigned tasks in Stream 2.

Stream 1: Code-base optimization & modernization

- **Upgrades and Migration:** Updating to the latest versions of operating systems, Python, and package dependencies to enhance performance and security. Also, supporting Kubernetes-based deployment for all PanDA WMS components.
- **Code Refactoring:** Focus on refactoring the code to improve readability and maintainability, and remove outdated or obsolete code to simplify the codebase.
- **Reinforcement of Coding Standards:** Implement and enforce coding standards with the help of code validation tools to ensure quality and consistency throughout the development process.
- **Kubernetes-Based CI/CD:** Develop a Kubernetes-based testbed to facilitate and optimize Continuous Integration (CI) and Continuous Deployment (CD) processes, supporting automated testing and deployment.

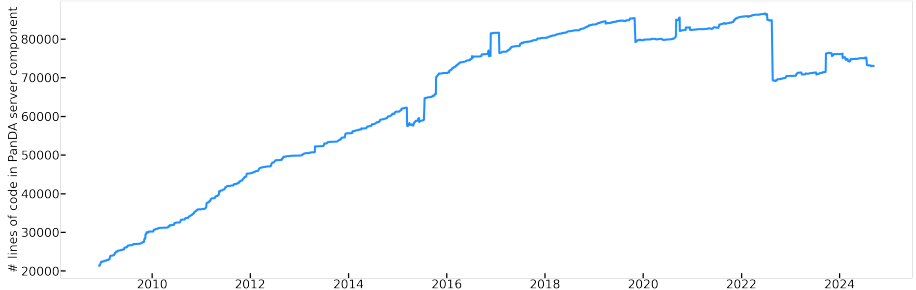

Figure 2 — # lines of code in the PanDA server component over time

Stream 2: System optimization and enhancement

Examples of the latest projects

Coordinating development issues in a more organized and structured manner, navigating long-term milestones while accepting short interruptions to address daily issues.

High memory (HIMEM):

- While the official memory recommendation for sites is to provide 2GB/vCPU, the reality is that the memory usage of our jobs has diversified up and down.
- This development aims to provide more memory granularity in order for very high memory jobs (up to 4-6 GB/vCPU) to balance out with low memory jobs (under 1 GB/vCPU)

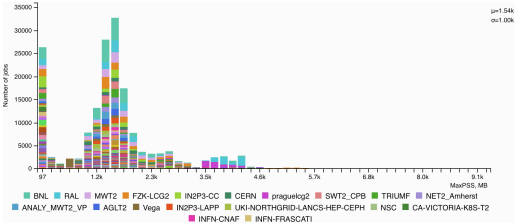

Figure 3 — Usual distribution of max PSS/core of ATLAS workloads (random 1k tasks finished on 19 September 2024)

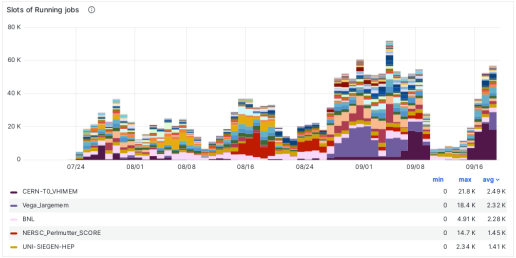

Figure 4 — # cores running very high memory jobs

Stream 2: System optimization and enhancement

Examples of the latest projects

Job generation performance:
With current setup each job generation thread can generate 22 jobs/second leading to several issues:

- overhead before scheduling jobs to sites leading to slow task completion
- inability to fill a site rapidly, e.g. post-downtime or opportunistic sites
- unbalanced job distribution caused by multiple nodes running in parallel to achieve sufficient job generation throughput

Prediction of jobs' CPU and Memory requirements:
Currently most tasks run scout jobs to measure actual resource requirements such as execution time, memory footprint, and CPU efficiency, before proceeding at full scale. The goal is to skip the scouting phase, obvious overhead in task execution, through reliable prediction of jobs' resource requirements, improving entire task completion time.

- This work is part of the REDWOOD project[‡] to bolster system resilience through optimized workload distribution and data placement
- The ML model to predict resource requirements will serve as a crucial building block for running the reinforcement learning to optimize the system
- Analysis of historical data, establishing AI/ML development framework, making an AI/ML model, implementing a new component using the model, and actual operation with the new component to improve the system

[‡] "REDWOOD: Resilient Federated Workflows in a Heterogeneous Computing Environment", DOE ASCR funded project

Stream 3: Outreach

- **Keeping documentation up-to-date:** Ensure that documentation is current and accurate `https://panda-wms.readthedocs.io/en/latest/`
- **Revamped the PanDA website:** Update and improve the PanDA website (`https://pandawms.org`) for better usability and information accessibility.
- **Published a general paper:** A comprehensive paper describing the PanDA system has been published.§
- **Monthly PanDA community forums:** Introduced monthly forums with dedicated slots for each experiment using or planning to use the PanDA system; 6 meetings have been conducted.
- **Plans for "how to" tutorials:** Recording and making "how to" tutorials to guide users is in progress.
- **Better integration with the distributed analysis community:** As a first step, implementing an interface for users to provide feedback and satisfaction levels on task execution.

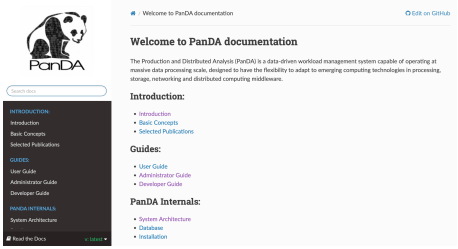

Figure 5 — PanDA documentation front page

Figure 6 — PanDA paper published in "Computing and Software for Big Science" journal

§Maeno, T., Alekseev, A., Barreiro Megino, F.H. et al. PanDA: Production and Distributed Analysis System. Comput Softw Big Sci 8, 4 (2024)

Stream 4: Interactive and dynamic workflow-oriented platform

- Develop an interactive and dynamic workflow oriented platform in front of the PanDA system
- Expand support of complex workflows for both production and analysis ATLAS and other experiments
- Optimizing algorithms in the system with awareness of entire workflows, starting from the brokerage
- Rework the BigPanDA monitoring system[¶] into an interactive web-platform to have all steps of running workflows in one place, including submission, monitoring, debugging and reconfiguration of parameters
- Currently exploring available technologies and frameworks to be used in the new system

[¶] T. Korchuganova, A. Alekseev et al BigPanDA monitoring system evolution in the ATLAS Experiment, EPJ Web of Conf. 295 04010 (2024)

Results & Outcome

- Improved and clarified project management approach by clearly defining 4 streams and focusing on long-term milestones and increasing visibility on progress
- Started modernizing and streamlining the codebase, enhancing maintainability and supporting k8s-based deployment for improved scalability and efficiency.
- Working on improvements in job generation performance and resource utilization, paving the way for efficient handling of diverse workloads.
- Launched monthly PanDA community forums, providing a platform for discussion and feedback among current and potential users.
- Progressing towards creating an interactive, workflow-aware platform with advanced monitoring and user interface capabilities to support complex workflows and broaden PanDA's usability