SHORT-PAPER

Fresco: A Public Multi-Institutional Dataset for Understanding HPC System Behavior and Dependability

JOSHUA MCKERRACHER, Purdue University, West Lafayette, IN, United States

PREETI MUKHERJEE, Purdue University, West Lafayette, IN, United States

RAJESH KALYANAM, Purdue University, West Lafayette, IN, United States

SAURABH BAGCHI, Purdue University, West Lafayette, IN, United States

Open Access Support provided by:

Purdue University

Published: 20 July 2025

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

Conference Sponsors:
SIGHPC

https://doi.org/10.1145/3708035.3736090
ISBN: 9798400713989

Fresco: A Public Multi-Institutional Dataset for Understanding HPC System Behavior and Dependability

Joshua McKerracher
Purdue University
West Lafayette, USA
jmckerra@purdue.edu

Preeti Mukherjee
Purdue University
West Lafayette, USA
mukher57@purdue.edu

Rajesh Kalyanam
Rosen Center for Advanced Computing
Purdue University
West Lafayette, USA
rkalyana@purdue.edu

Saurabh Bagchi
Purdue University
West Lafayette, USA
sbagchi@purdue.edu

Abstract

The scarcity of publicly available operational data from High Performance Computing (HPC) systems hinders research in critical areas like system dependability and resource optimization. While recent efforts, such as the Atlas project, have increased the availability of cluster traces, these datasets often lack fine-grained operational metrics linked to comprehensive job-level attributes across diverse environments. To address this gap, we introduce Fresco, a dataset containing data from 20.9 million jobs spanning 75 months collected from three major academic supercomputing clusters: Purdue's Anvil and Conte systems, and Texas Advanced Computing Center's Stampede. Fresco uniquely captures six key performance metrics alongside many job-level attributes such as resource allocations and execution outcomes. We detail our data integration process that transforms and standardizes the heterogeneous data sources into a consistent format. The resulting dataset enables researchers to investigate the relationships between job characteristics, resource consumption patterns, and system performance in academic HPC environments. We make this resource open source at **https://www.frescodata.xyz**. Our expectation is that this public release will facilitate research and operational improvements that had previously been impossible due to the unavailability of such data.

CCS Concepts

• **Computer systems organization → Dependable and fault-tolerant systems and networks**; **Dependable and fault-tolerant systems and networks**; • **Information systems → Digital libraries and archives**.

Keywords

Computer system usage, Data repository, Computer system dependability

PEARC '25, Columbus, OH, USA

ACM ISBN 979-8-4007-1398-9/25/07
https://doi.org/10.1145/3708035.3736090

ACM Reference Format:
Joshua McKerracher, Preeti Mukherjee, Rajesh Kalyanam, and Saurabh Bagchi. 2025. Fresco: A Public Multi-Institutional Dataset for Understanding HPC System Behavior and Dependability. In *Practice and Experience in Advanced Research Computing (PEARC '25), July 20–24, 2025, Columbus, OH, USA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3708035.3736090

1 Introduction

Large and complex HPC systems are vital for modern research and industry. Their effective management and reliable operation require detailed operational data for understanding system behavior. Despite the importance of data-driven approaches in HPC, readily available, public datasets with the necessary breadth and depth remain scarce. While valuable resources like the Atlas project [2] and the Computer Failure Data Repository (CFDR) [7] have begun to address the lack of public data, the initial releases often center around system-level and scheduler data and do not consistently provide fine-grained, job-level detail; specifically, they lack the combination of resource usage and user request information together in a time series structure. Consequently, these traces often present difficulties for conducting cross-institutional and holistic analyses, such as how failures are correlated with resource usage and user request patterns. Furthermore, single-institution or single-cluster failure datasets limit generalizability, especially regarding workload characteristics. To address these limitations, we introduce Fresco, a dataset designed to bridge this gap and funded through NSF projects 2016608 and 2016704. Fresco's data sources include Purdue's Conte and Anvil clusters along with Texas Advanced Computing Center's Stampede. Distinct from all current open source computer system usage repositories, Fresco differentiates itself through its data collection and integration approach. Fresco captures both detailed performance metrics (for compute nodes and networks) and extensive application job-level attributes. Fresco's unique contributions include:

- **Large temporal coverage:** Spanning 75 months and 20.9 million jobs, Fresco significantly exceeds the temporal scope of existing traces, addressing the demonstrated inadequacy of short-duration traces to represent workload characteristics accurately.

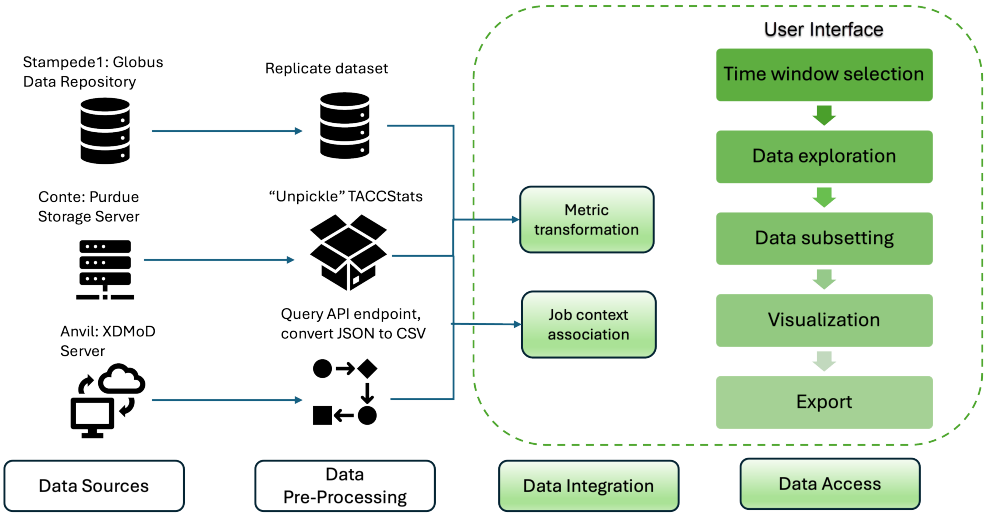


Figure 1: High-level overview of the process for constructing the Fresco dataset. The blocks highlighted in green are the key contributions of this paper.

- **Integrated performance metrics:** By capturing six key resource utilization metrics (CPU usage, GPU usage, memory usage, memory excluding disk cache, NFS activity, and block I/O) alongside job execution context, Fresco facilitates multi-dimensional analyses of the relationships between job characteristics and resource consumption patterns.
- **Cross-institutional and cross-cluster comparison:** The inclusion of data from three distinct academic HPC environments provides a heterogeneous mix of workloads, enabling researchers to develop and validate more generalizable resource management strategies and dependability techniques.
- **Standardized data integration:** The rigorous data transformation and integration process described in this paper allows for straightforward analysis across clusters, and provides a methodological framework that can be seamlessly used to incorporate data from other HPC environments in the future.

2 Related Work

The persistent scarcity of public HPC datasets has long hampered research and deployment into efficient and dependable operation of supercomputing clusters. The Atlas project [1] highlighted the over-reliance on the potentially unrepresentative 2011 Google trace. Their analysis revealed significant differences in job characteristics compared to workloads from LANL and Two Sigma. This over-dependence can optimize for specific workload types, potentially failing to generalize, and Atlas also found temporal variations within clusters. This lack of comprehensive datasets combining job-level attributes with detailed performance metrics across diverse institutional environments continues to impede research in crucial areas. Fresco aims to address this gap by providing a multi-institutional dataset capturing both job attributes and operational metrics, enabling more generalizable research across different HPC environments.

3 Methods

3.1 Dataset Description

Fresco is built from Purdue's Conte and Anvil, and the Texas Advanced Computing Center (TACC) Stampede supercomputers. It integrates two main data types: (i) **accounting data**: job scheduling information like submit/start/end times, requested resources, user, job name, exit status; and (ii) **performance metrics**: periodic measurements of CPU, GPU, memory, network, and I/O utilization. Details of the three clusters that serve as data sources are provided in Table 1. This combined data is organized into a unified schema containing temporal job metadata, resource allocation attributes, job identification, execution context, and fine-grained performance metrics.

3.2 Dataset Construction

Our high level workflow for constructing the Fresco dataset is illustrated in Figure 1. The integration process ensured consistency through standardization of datetime formats, normalization of job identifiers and accounting columns, and validation to match time-series records within job execution boundaries. Next, we briefly describe the data collection and pre-processing steps, before diving into the details of the data integration process.

As shown in Figure 2, each cluster required a different integration approach due to variations in data organization and collection methods.

3.2.1 Performance Metric Data Collection. Conte and Stampede utilize TACC Stats [3], a low-overhead infrastructure collecting system-wide performance data (CPU, I/O, network, filesystem) per node at regular intervals. Initially, this data was not organized at the job level; the initial Conte data is grouped into year-month folders, and the Stampede data is grouped into node folders. Each CSV file in these folders contains performance data for their corresponding device (one CSV file for each monitored metric) augmented with

Characteristic	Conte	Anvil	Stampede
Time Period	2015 - 2017	07/2022 – 05/2023	2013 - 2016
Jobs Captured	10.8M	1.4M	8.7M
Nodes	580	1,000	6,400
Cores	8,280	128,000	522,800
Architecture	2x Xeon Phi/node	2x AMD EPYC 7763 (128 cores/node), 256GB RAM + GPU partition (16 nodes w/4x A100)	Intel Xeon E5
Data Sources	TACC Stats, PBS/TORQUE	XDMoD, SLURM	TACC Stats, PBS/TORQUE
Performance Metrics in Dataset	CPU, memory, NFS, block I/O	CPU, GPU, memory, NFS, block I/O	CPU, memory, NFS, block I/O

Table 1: Comparison of Cluster Characteristics

Category	Variables
Temporal job metadata	`submit_time`, `start_time`, `end_time`
Resource allocation attributes	`timelimit`, `nhosts`, `ncores`, `account`, `queue`, `host`, `jid`, `unit`, `username`, and `jobname`,
Execution context	`exitcode` and `host_list`
Fine-grained performance metrics	`value_cpuuser`, `value_gpu`, `value_memused`, `value_memused_minus_diskcache`, `value_nfs`, and `value_block`

Table 2: Unified Fresco Schema

the job's unique ID, the identity of the node from which the data originated, and the data collection timestamp.

Anvil's performance metric data comes from XDMoD [6], an HPC monitoring framework. XDMoD collects time-series metrics (CPU, memory, GPU utilization, NFS activity, block I/O) through its Performance Co-Pilot. Job-level metrics were extracted by correlating system measurements with job execution periods from scheduler records, organized into CSV files with timestamps, node identifiers, metric types, and values.

3.2.2 Job Accounting Data Collection. For Fresco, job records include submission, start, end times, allocated resources, queue information, and completion status. These were extracted, anonymized, and formatted into standardized CSV files. Conte and Stampede employ the TORQUE implementation of PBS. TORQUE logs batch job events (queuing, start, end, etc.) with timestamps, user/group, requested/used resources in a fixed format within daily accounting logs. These anonymized records are formatted as comma-separated lists in monthly CSV files. Anvil's job accounting data is extracted from XDMoD, which maintains a relational database of job submissions, resource allocations, and execution metrics that have been derived from the SLURM *sacct* data.

4 Results - Data Integration

The construction of the Fresco dataset involved a comprehensive data integration process to combine accounting and performance metric data from the three different clusters. Due to variations in data formats and collection methods across the clusters, the integration process was tailored for each, aiming to achieve a consistent and unified dataset.

4.1 Stampede and Conte Data Integration

Both Stampede and Conte utilize TACC Stats to generate performance metrics, creating a natural alignment in their integration processes despite organizational differences in the raw data.

4.1.1 Step One: Metric Transformation. For both clusters, we focused on transforming four key metrics from the TACC Stats collection into standardized measurements:

(1) **Block I/O performance:** We calculated throughput by combining read/write sectors, normalizing by operation time, and converting to gigabytes per second using standard sector size.
(2) **CPU utilization:** We measured user-space activity as a percentage by calculating the proportion of CPU time spent in user and nice modes relative to total CPU time.
(3) **Memory usage:** We implemented dual metrics tracking both total physical memory consumption and application-specific memory, excluding disk cache, both expressed in gigabytes.
(4) **Network File System throughput:** We derived standardized transfer rates from filesystem read/write operations, expressed in megabytes per second.

While the transformation formulas remained consistent across both clusters, the implementation required different approaches due to data organization. Both transformations standardized outputs to the Fresco schema.

4.1.2 Step Two: Job Context Association. The second integration step involved linking transformed performance metrics with job accounting data from PBS/TORQUE. The integrated dataset preserved both job metadata and fine-grained performance metrics,

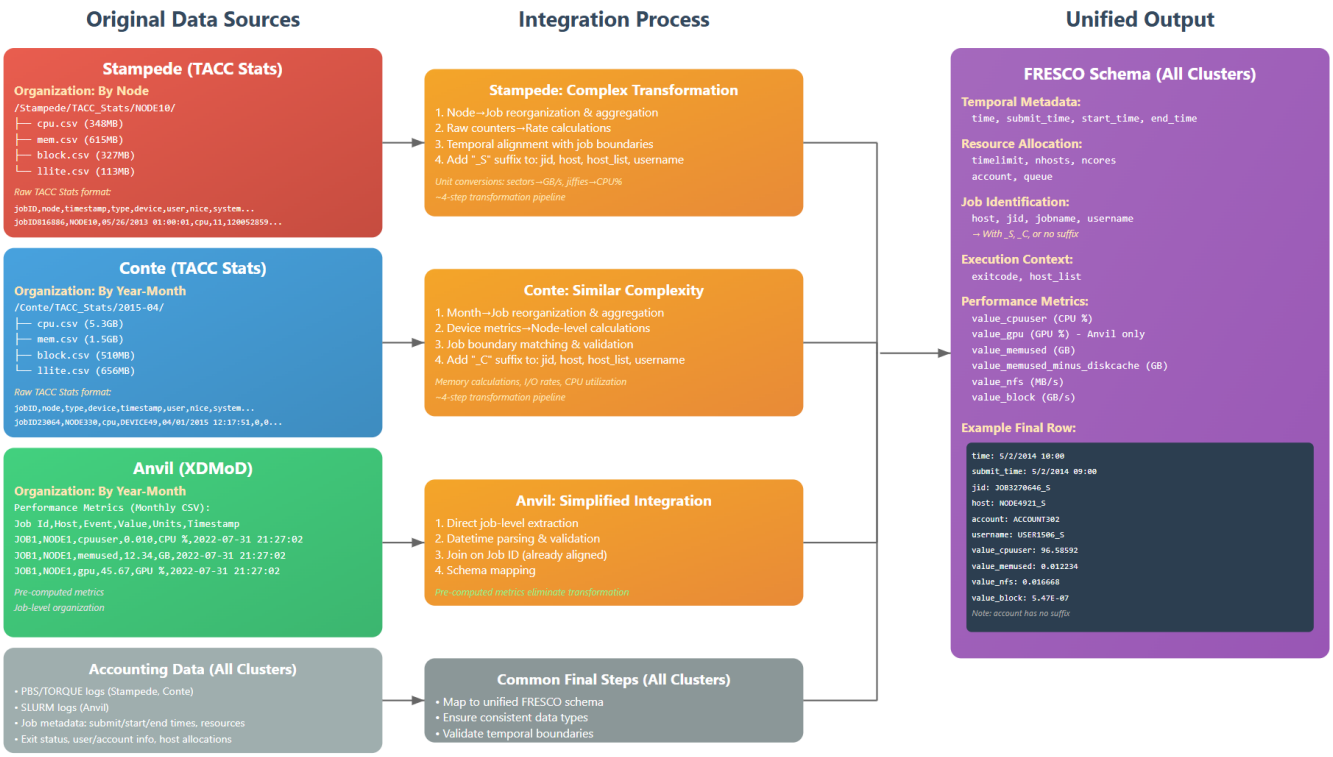


Figure 2: Fresco Data Integration Pipeline: This figure demonstrates the transformation process from original data sources (Stampede, Conte, and Anvil) through integration steps to the unified Fresco schema output.

creating a comprehensive view of resource utilization throughout each job's lifecycle. This process addressed several challenges:

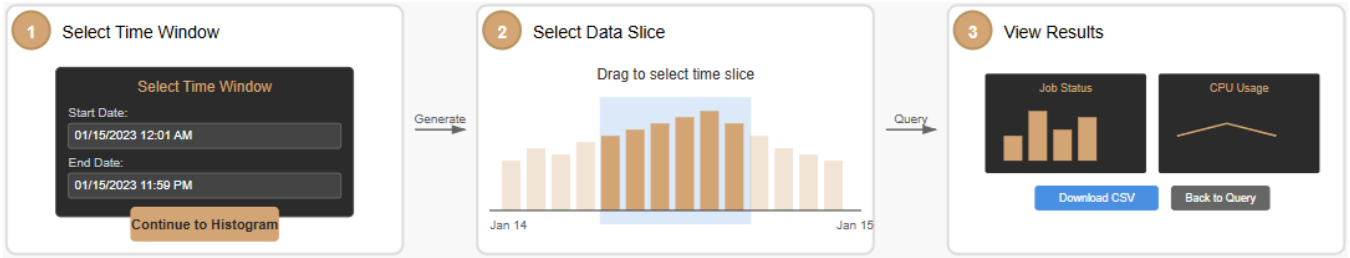
(1) **Temporal boundary matching:** Performance metrics were associated with jobs only when their timestamps fell within the corresponding job's execution window, ensuring accurate resource attribution. For instance, a CPU usage measurement timestamped 2016-12-05 09:30:00+00:00 for JOB123 would be included in the final dataset only if this timestamp occurs between the job's start and end times as recorded in the accounting data.
(2) **Node-job mapping:** Metrics were correlated with jobs through dual criteria: temporal overlap and node allocation consistency verified against scheduler records. This ensured that performance data was attributed only to jobs that were actually executing on the corresponding nodes during the measurement period.
(3) **Semantic alignment:** Job exit codes from PBS/TORQUE (used by Conte and Stampede) and SLURM (used by Anvil) were mapped to a unified taxonomy of standardized job states (COMPLETED, FAILED, ABORTED, etc.). This harmonization was necessary because the original scheduler-specific exit codes differed significantly across the three systems, requiring careful semantic translation to ensure consistent interpretation across the integrated dataset.

4.2 Anvil Data Integration

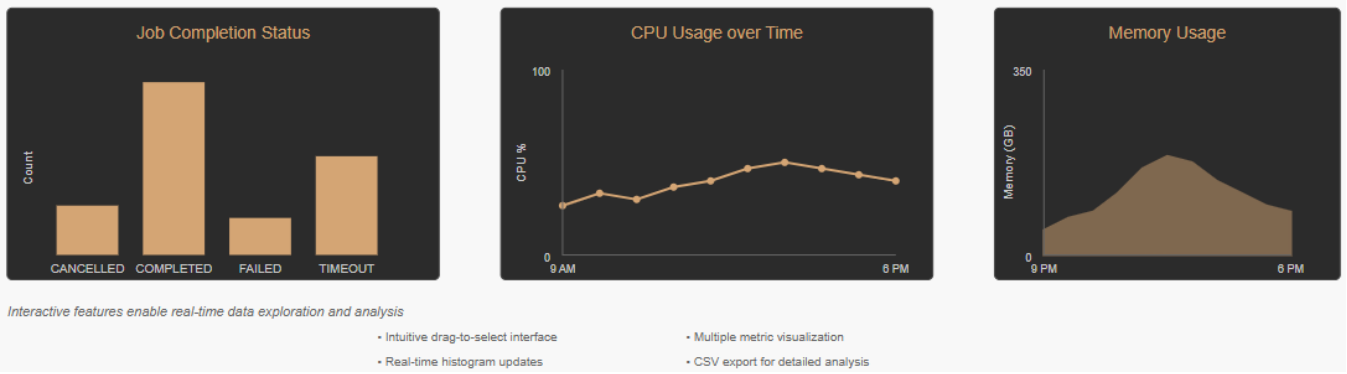
Anvil's data integration process differed as its job accounting and performance metric data was obtained from XDMoD [6] and was already organized into monthly files.

4.2.1 Step One: Performance Metric Data Extraction. Job-level metrics were extracted by correlating job performance metrics (from XDMoD) with job execution periods based on SLURM scheduler records. The resulting data is organized as CSV files with timestamps, node identifiers, metric types, and measurement values.

4.2.2 Step Two: Job Accounting Data Extraction and Association. Relevant fields like submission time, start time, end time, allocated resources, queue information, and completion status were extracted from the XDMoD accounting data. The raw accounting data was processed to anonymize user and project identifiers and then formatted into standardized CSV files. The integration of Anvil's performance metrics and job accounting data involved joining the two datasets based on job identifiers and ensuring that the performance metric timestamps fell within the start and end times of the corresponding jobs. The resulting data was then mapped to the unified Fresco schema, ensuring consistency with the data from Conte and Stampede.

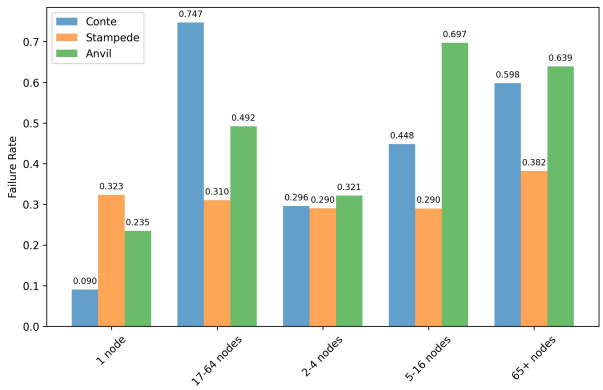


(a) The three-step query workflow for data selection

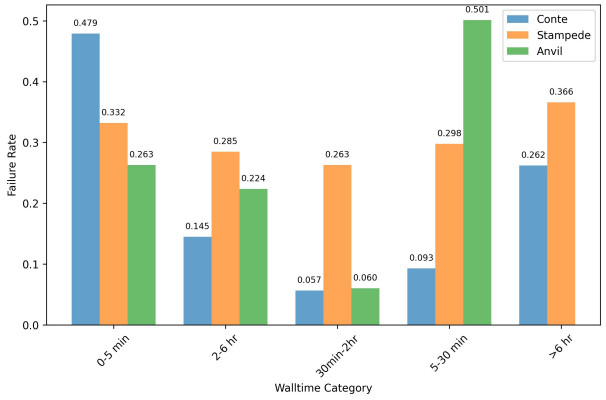


(b) Example visualizations generated from user queries with interactive features for real-time data exploration and analysis.

Figure 3: Fresco Query Interface and Sample Analysis Results: This figure depicts the design of the query interface and the presentation of the generated analytical results.



(a) Node count compared to failure rate, grouped by cluster. Note that the trend between node count and failure rate varies between clusters, with *Stampede* holding relatively stable, but failure count increasing sharply for *Anvil* and *Conte*.



(b) Walltime compared to failure rate, grouped by cluster. Across all clusters, extremely short or long jobs have an elevated failure rate, compared to medium duration jobs.

Figure 4: Combined Resource Allocation vs Job Failure Analysis across Fresco clusters.

4.3 Accessing the Fresco dataset

The Fresco dataset is designed to be easily accessible to the research community through a purpose-built web application. This section describes how researchers can access, explore, and download the data for their own analysis.

4.3.1 Dataset Organization and Storage. After the integration process described in previous sections, the unified dataset is broken into 1-hour chunks, converted to the efficient Parquet format, and stored in an Amazon Web Services (AWS) S3 bucket. This chunking strategy provides several advantages: it enables efficient querying over specific time periods, reduces memory requirements for analysis, and facilitates parallel processing of large datasets.

4.3.2 Web Interface. Researchers can access the Fresco dataset through a dedicated web application at https://www.frescodata.xyz/

query_builder. The interface guides users through several steps to query and visualize the data, beginning with time window selection (30 days ≈ 115k rows max). After selecting a date range, the interface displays a histogram showing record distribution over time. Users can then refine their search by dragging across a section of this histogram to select a specific time slice for focused analysis. As shown in Figure 4, the interface offers versatile visualization capabilities, allowing for simultaneous plotting of multiple metrics and job accounting data for comparative analysis. This shows over a part of a day, the breakdown of jobs by their completion status, and the CPU usage across all machines within a cluster.

Fresco's interactive visualization and in-browser filtering is enabled through Mosaic `vgplot` [4]. `vgplot` links interactive HTML elements to a client-side DuckDB [5] instance, dynamically constructing SQL queries to filter and process data efficiently. This architecture was chosen to allow a user to query and browse the expansive dataset interactively, reducing the friction between experiments or analyses on new sections of the data.

For more detailed work, researchers can download their selected dataset as a CSV file. The combination of intuitive visualization tools and raw data download capabilities supports both exploratory analysis and in-depth research.

4.4 Dataset Characterization and Analysis

To demonstrate the analytical capabilities of Fresco and provide insights into HPC job behavior patterns, we conducted an analysis of job failure rates across different resource allocation categories and execution durations. This analysis reveals important operational characteristics that can inform both system administrators and researchers studying HPC workload patterns.

4.4.1 Failure Rate Analysis Across Resource Allocation and Job Duration. Our analysis examined job failure patterns across both resource allocation scales and execution durations, categorizing jobs into five node allocation groups from single-node to very large parallel jobs (65+ nodes). The results reveal some consistent patterns across all three clusters.

Common Resource Allocation Patterns: Anvil and Conte demonstrate higher failure rates for jobs requiring larger node allocations, with single-node and small parallel jobs consistently showing better reliability than large-scale parallel jobs, while Stampede is relatively stable over node count. This variation may be explained by the unique nature of workloads on each cluster, present due to the diversity and scope of the dataset. This pattern holds across the different cluster architectures and time periods represented in the dataset.

Common Duration-Based Patterns: The walltime analysis reveals consistent temporal patterns across all systems. Very short jobs (0-5 minutes) show elevated failure rates across all clusters. Short to medium duration jobs (5 minutes to 6 hours) exhibit the lowest failure rates consistently. Long-running jobs (>6 hours) demonstrate increased failure rates across all three systems.

These patterns provide valuable insights for both system operators and users: jobs with moderate resource requirements and execution times tend to be most reliable across different HPC environments, while very large-scale, very long-running, or excessively short jobs consistently show higher failure rates regardless of the specific cluster architecture. One may infer that such patterns arrive due to user misconfiguration in the case of very short jobs and incorrect termination criteria, in the case of extensive and/or large-scale jobs.

5 Conclusion

Fresco offers a resource for the HPC research community for designing and operating dependable clusters by providing a unique, multi-institutional collection of fine-grained resource utilization data linked with job-level attributes.

Acknowledgments

This work was ably supported by Carol Song of Purdue's Information Technology Department, Stephen Harrell of the Texas Advanced Computing Center (TACC). This material is based in part upon work supported by the National Science Foundation under Grant Numbers CNS-2016704 and CCF-2140139. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

References

[1] George Amvrosiadis, Michael Kuchnik, Jun Woo Park, Chuck Cranor, Gregory R Ganger, Elisabeth Moore, and Nathan DeBardeleben. 2018. The Atlas cluster trace repository. *Usenix Mag* 43, 4 (2018), 29–35.

[2] George Amvrosiadis, Jun Woo Park, Gregory R Ganger, Garth A Gibson, Elisabeth Baseman, and Nathan DeBardeleben. 2018. On the diversity of cluster workloads and its impact on research results. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. 533–546.

[3] Todd Evans, William L Barth, James C Browne, Robert L DeLeon, Thomas R Furlani, Steven M Gallo, Matthew D Jones, and Abani K Patra. 2014. Comprehensive resource use monitoring for HPC systems with TACC stats. In *2014 First International Workshop on HPC User Support Tools*. IEEE, 13–21.

[4] Jeffrey Heer and Dominik Moritz. 2024. Mosaic: An Architecture for Scalable & Interoperable Data Views. *IEEE Transactions on Visualization and Computer Graphics* 30, 1 (2024), 436–446. doi:10.1109/TVCG.2023.3327189

[5] André Kohn, Dominik Moritz, Mark Raasveldt, Hannes Mühleisen, and Thomas Neumann. 2022. DuckDB-wasm: fast analytical processing for the web. *Proc. VLDB Endow.* 15, 12 (Aug. 2022), 3574–3577. doi:10.14778/3554821.3554847

[6] Jeffrey T Palmer, Steven M Gallo, Thomas R Furlani, Matthew D Jones, Robert L DeLeon, Joseph P White, Nikolay Simakov, Abani K Patra, Jeanette Sperhac, Thomas Yearke, et al. 2015. Open XDMoD: A tool for the comprehensive management of high-performance computing resources. *Computing in Science & Engineering* 17, 4 (2015), 52–62.

[7] Bianca Schroeder and Garth Gibson. 2015. The Computer Failure Data Repository (CFDR). https://www.usenix.org/cfdr