SLURMminer: A Tool for SLURM System Analysis with Process Mining

Zahra Sadeghibogar[1,*], Alessandro Berti[1], Marco Pegoraro[1] and Wil M.P. van der Aalst[1]

[1]*Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany*

Abstract

SLURMminer is a tool designed to analyze SLURM systems in High-Performance Computing (HPC) clusters. It utilizes process mining techniques to generate event logs, extract process models, and visualize critical business intelligence metrics. The tool's unique log extraction approach for SLURM clusters allows for a detailed analysis of jobs and workflows. By providing insights into workflow execution, workload patterns, system utilization, workload balancing, and anomaly detection, SLURMminer equips researchers and system administrators with essential data for performance assessment and user behavior analysis. This paper details installation procedures and describes the tool's functionalities, facilitating efficient management and optimization of HPC clusters with SLURMminer.

Keywords

HPC, SLURM, Process Mining

1. Introduction

High-Performance Computing (HPC) clusters are instrumental in a range of disciplines, such as scientific research, data analytics, and complex computational simulations. A key aspect of maximizing cluster performance, effective resource allocation, and efficient workload management is rigorous monitoring and analysis [1]. SLURM (Simple Linux Utility for Resource Management), a prevalent job scheduler in HPC environments, is central to this monitoring process.

In the context of process mining, SLURMminer stands out from existing tools and approaches in the field. While other tools [2, 3, 4, 5] focus mainly on job-level performance in HPC clusters, SLURMminer uniquely emphasizes the use of process mining for a comprehensive analysis of running jobs. It allows a comprehensive understanding of active processes, workflows, and overall efficiency in SLURM-based clusters.

The tool's emphasis on process mining revolutionizes HPC cluster management. It empowers administrators and users to identify bottlenecks in scientific workflows and optimize resource

BPM 2023 Demos Resources Forum, September 11–15 2023, Utrecht (NL)

*Corresponding author.

sadeghi@pads.rwth-aachen.de (Z. Sadeghibogar); a.berti@pads.rwth-aachen.de (A. Berti); pegoraro@pads.rwth-aachen.de (M. Pegoraro); wvdaalst@pads.rwth-aachen.de (W. M.P. v. d. Aalst)

http://mpegoraro.net/ (M. Pegoraro); https://vdaalst.com/ (W. M.P. v. d. Aalst)

0000-0002-6340-9669 (Z. Sadeghibogar); 0000-0001-8515-3089 (A. Berti); 0000-0002-8997-7517 (M. Pegoraro); 0000-0002-0955-6940 (W. M.P. v. d. Aalst)

CEUR Workshop Proceedings (CEUR-WS.org)

allocation and workload management. The use of process mining techniques uncovers patterns, bottlenecks, and anomalies that were previously overlooked by conventional monitoring tools, leading to improved system comprehension and resource utilization. On the other side, SLURMminer can demonstrate how its capabilities can optimize processes, increase efficiency, and uncover hidden opportunities for improvement within organizations that rely on SLURM-based HPC clusters for their computational needs.

SLURMminer was developed on the foundations laid by the author's previous research [6], where two innovative methods were proposed to generate an event log from the extracted SLURM logs. With this event log serving as a data source, process mining techniques are then applied, giving rise to a richer, more detailed level of analysis for HPC cluster monitoring.

2. Implementation and Features

SLURMminer, a Python-based tool built using the PM4Py library, facilitates advanced process mining operations on SLURM event data.

Constructed on the Flask framework[1], it allows HTTP/HTTPS deployment and is compatible with enterprise-grade servers. With Flask's Cross-Origin Resource Sharing (CORS[2]) support, SLURMminer offers a smooth user experience by facilitating multi-origin requests.

The user interface of SLURMminer, crafted using the Bootstrap framework[3], ensures responsive design, accelerated development time, and browser compatibility. Bootstrap's customizability and integration with JavaScript libraries permit dynamic and interactive UI component development.

SLURMminer stands out for its ability to observe live event streams from the SLURM system, a feature enabled by Flask and PM4Py. It processes and visualizes real-time SLURM events, providing users with instantaneous insights into system behavior and performance.

3. Use Cases and Examples

SLURMminer, an evolutionary tool built upon preceding research, enables thorough SLURM system analysis by integrating system logging, process mining, and business intelligence visualization. The following list proposes some use cases of the tool:

- *System Logging*: Users first engage with the system logging component, logging into the HPC system via a dedicated form on the user interface (UI) (Fig.1, right). The mechanism monitors the SLURM system's ongoing jobs and records them using the "squeue" command. Logging initiation and reset are controlled by "Start Logging" and "Restart Logging" buttons, respectively, with the latter resetting the log located in the "slurm-log" directory (Fig.1, left).
- *Process mining*: The SLURM log generated can be leveraged by process mining technologies. An event log, structured for process mining, is generated by the "Generate

[1] http://flask.pocoo.org/
[2] https://flask.palletsprojects.com/
[3] https://getbootstrap.com/docs/

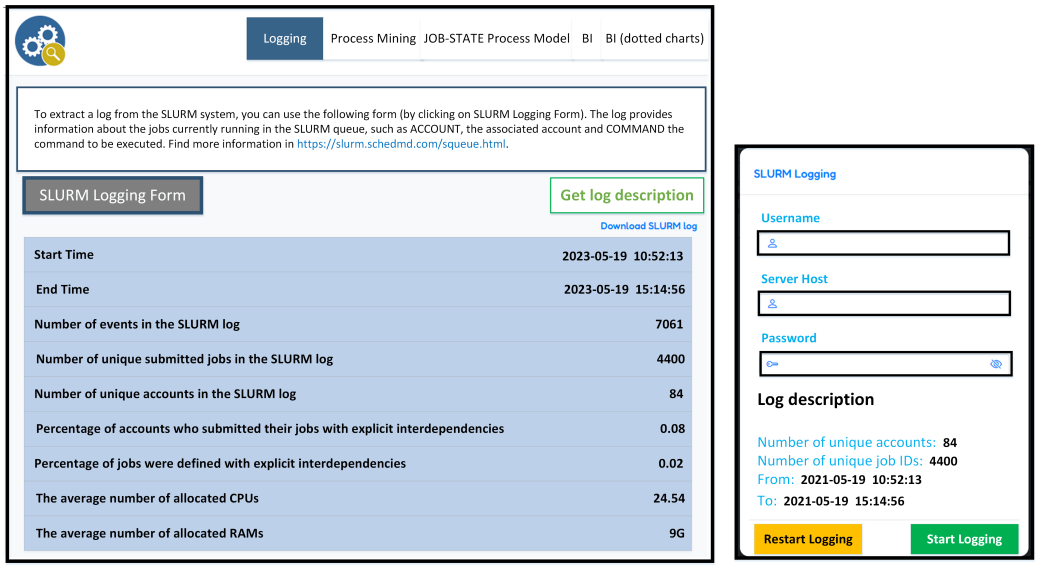

Figure 1: SLURMminer Interface: Home Page with Access to Logging Forms (right) and Log Descriptions.

and Download Normal Log" button. This log underpins the creation of process models visualizing SLURM system operations, extending knowledge about job executions and possible inefficiencies (Fig.2). The normal event log also elucidates the job execution lifecycle, with stages like PENDING, CONFIGURING, RUNNING, and COMPLETING.

– *Business Intelligence visualization*: SLURMminer encapsulates performance analysis and visualization, providing crucial metrics and graphical illustrations. These visuals cover details such as job counts, job distribution, and the status of active or pending jobs (Fig.3, right). An additional feature identifies batch activities, where numerous similar computations are launched simultaneously. The tool flags potential batching instances by monitoring job initiations within a defined time window and applies a threshold to filter out high-frequency accounts, resulting in a plot representing potential batch activity periods (Fig.3,left).

SLURMminer amalgamates system logging, process mining, and business intelligence visualization, providing a robust analytical tool for effective HPC cluster management.

4. Tool Repository and Demonstration

SLURMminer's codebase is available on GitHub at https://github.com/zasab/SLURMminer. The README.md file outlines the setup procedure.

Prerequisites for installation include Node.js[4] and NPM for frontend, and Python with `pip`[5] for backend dependencies.

[4]https://nodejs.org/
[5]https://www.python.org/

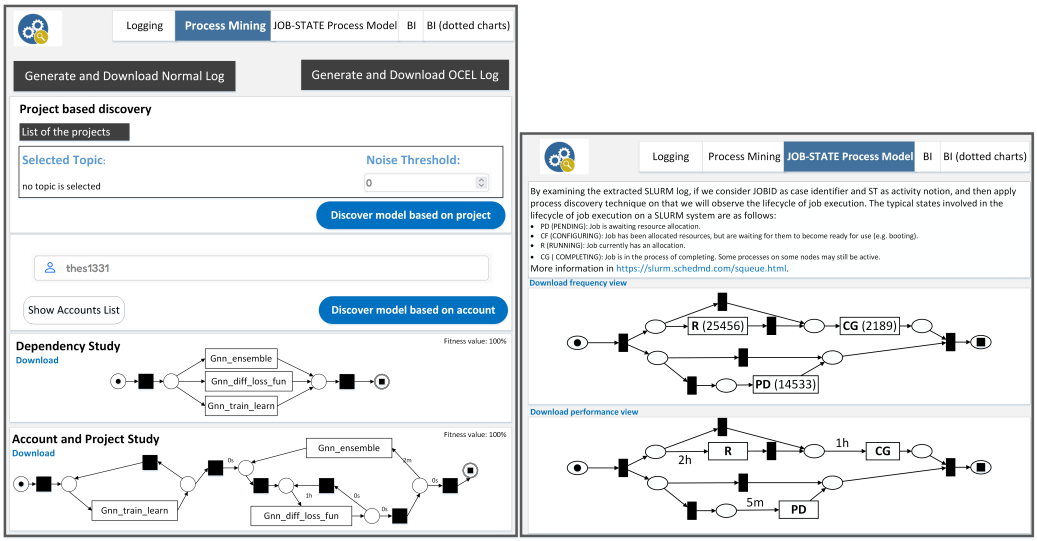

Figure 2: Process mining analysis in SLURMminer: Discovering Process Models (left) and Examining Job Execution States (right).

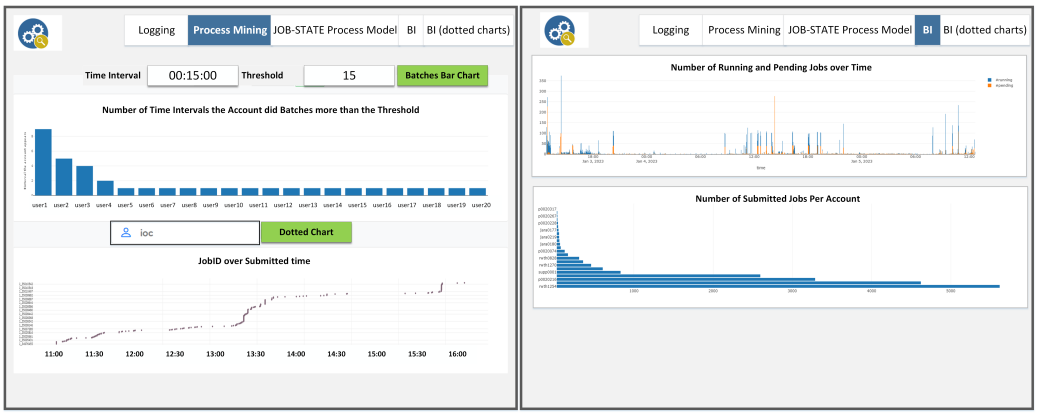

Figure 3: Business Intelligence visualization: Job-Level Analysis (right) and Batching Analysis (left) with Plots

To start, clone the repository via `git clone`, then install the required Python packages using `pip install`. Launch the Flask server by executing the "app.py" script, making it available in your web browser. For frontend setup, navigate to the "DEV-front" directory and run `npm install` followed by `npm run dev`. Once the Node.js server is up, access the application via the provided "Local" URL (e.g., http://localhost:5173/).

Please check out my demo video of the tool paper at https://youtu.be/y0ujrvHY2kE for a detailed demonstration.

5. Conclusion

SLURMminer, bridging process mining with SLURM-based HPC cluster management, provides a novel look at job behaviors and anomaly detection, forging a path toward improved performance and refined optimization strategies. Its intuitive interface provides sophisticated analytical abilities to administrators and users, improving resource management and job scheduling in HPC contexts.

Future iterations of SLURMminer envision a more feature-rich platform, with an expanded array of metrics and a broader spectrum of process mining techniques. These enhancements promise to augment SLURMminer's impact, continuously steering the evolution of SLURM-based HPC cluster management toward greater efficiency and productivity.

Acknowledgements

The authors gratefully acknowledge the German Federal Ministry of Education and Research (BMBF) and the Ministry of Education and Research of North-Rhine Westphalia for supporting this work/project as part of the NHR funding. Also, we thank Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

[1] A. B. Yoo, M. A. Jette, M. Grondona, SLURM: simple linux utility for resource management, in: Job Scheduling Strategies for Parallel Processing, 9th International Workshop, JSSPP 2003, Seattle, WA, USA, June 24, 2003, Revised Papers, volume 2862 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 44–60. URL: https://doi.org/10.1007/10968987_3. doi:10.1007/10968987_3.

[2] R. Dietrich, F. Winkler, A. Knüpfer, W. E. Nagel, PIKA: center-wide and job-aware cluster monitoring, in: IEEE International Conference on Cluster Computing, CLUSTER 2020, Kobe, Japan, September 14-17, 2020, IEEE, 2020, pp. 424–432. doi:10.1109/CLUSTER49012.2020.00061.

[3] P. Kunz, HPC Job-Monitoring with SLURM, Prometheus, and Grafana (2022).

[4] A. Pal, P. Malakar, MAP: A visual analytics system for job monitoring and analysis, in: IEEE International Conference on Cluster Computing, CLUSTER 2020, Kobe, Japan, September 14-17, 2020, IEEE, 2020, pp. 442–448. doi:10.1109/CLUSTER49012.2020.00063.

[5] T. Röhl, J. Eitzinger, G. Hager, G. Wellein, LIKWID monitoring stack: A flexible framework enabling job specific performance monitoring for the masses, in: 2017 IEEE International Conference on Cluster Computing, CLUSTER 2017, Honolulu, HI, USA, September 5-8, 2017, IEEE Computer Society, 2017, pp. 781–784. doi:10.1109/CLUSTER.2017.115.

[6] Z. Sadeghibogar, A. Berti, M. Pegoraro, W. M. P. van der Aalst, Applying process mining on scientific workflows: a case study, CoRR abs/2307.02833 (2023). URL: https://doi.org/10.48550/arXiv.2307.02833. doi:10.48550/arXiv.2307.02833. arXiv:2307.02833.