Mastering HPC monitoring data: from zero to hero with LLview

Filipe Guimarães, *Thomas Breuer, Wolfgang Frings*

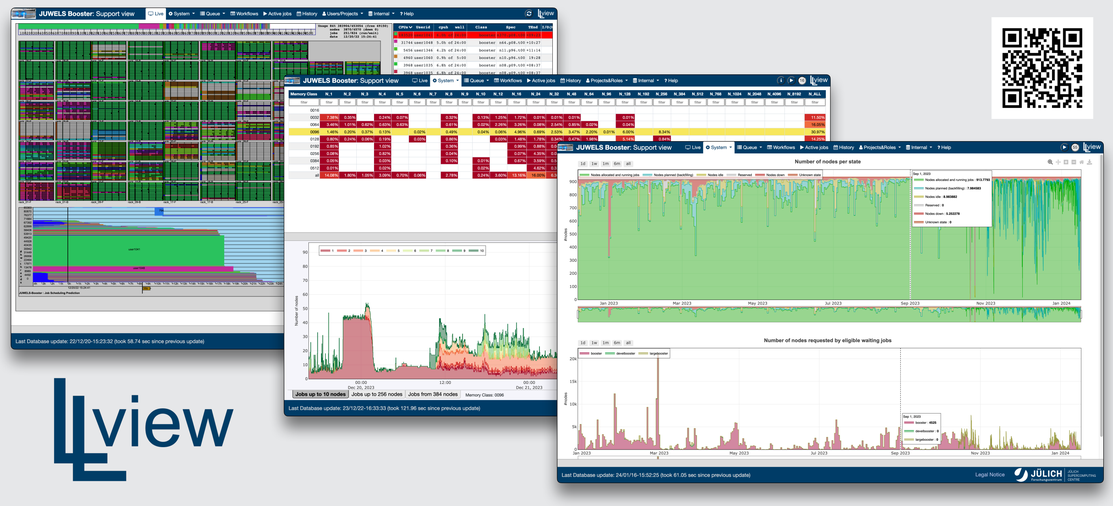
Jülich Supercomputing Centre, Forschungszentrum Jülich

Introduction

Monitoring of large high performance computer (HPC) systems facilitates real time error detection, optimization analysis and identification of jobs with incorrect setups or inefficient resource usage. To this end, we have created the open-source LLview infrastructure [1] to aid system administrators and job runners. With scalable performance and automatic minute-based updates, LLview offers:

- System-wide overview
- Near real-time data
- Automated reports for relevant jobs
- Role-based access control
- Statistical analysis
- Modular, extensible design

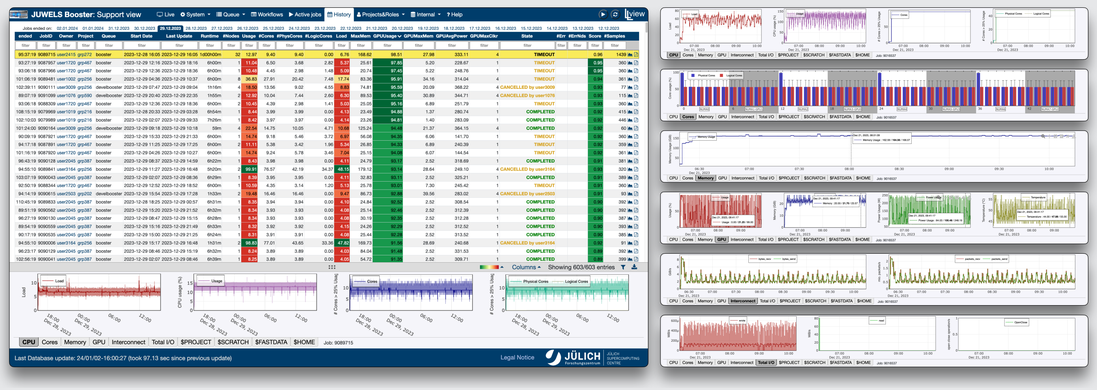
Different from other tools (such as Grafana), LLview is specifically designed with HPC systems in mind, providing a more job-focused view available for all users, with the respective permissions already taken into account.



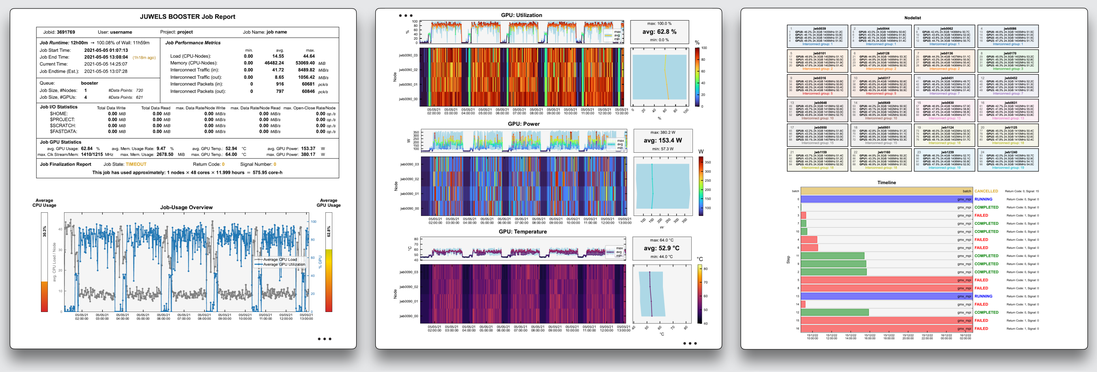
System-wide views on LLview web portal

Methods & Design

We developed the LLview infrastructure to collect, process, and aggregate existing monitoring metrics from the system without adding extra overhead to running jobs. These metrics are presented through a visually effective web portal, providing users with extensive job information. Additionally, detailed job reports are automatically generated and updated in self-contained HTML and PDF formats, facilitating long-term monitoring.



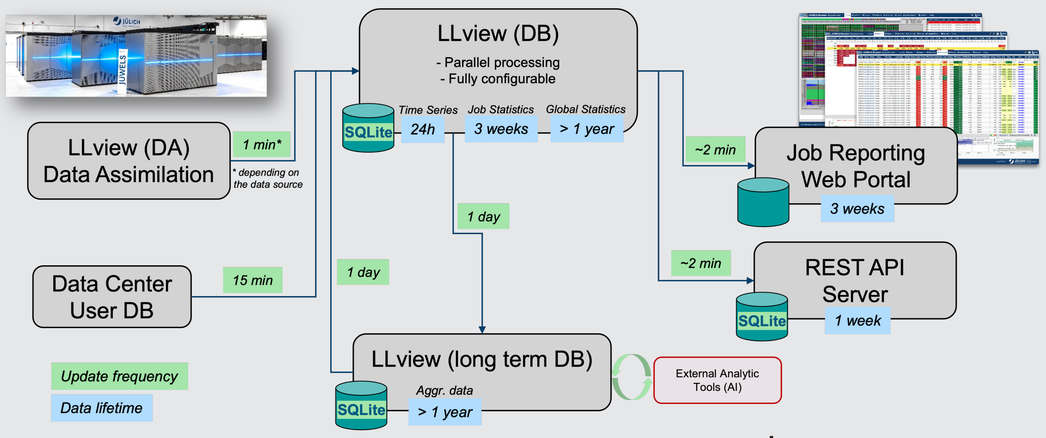
LLview Web Portal: The main tabs display the current running jobs (left) and the ones finished in the last three weeks. Each line represents a job, displaying a wealth of metrics in columns that are grouped by selectable categories. Information of the selected job is shown on dynamic footer graphs (right), including CPU, GPU and core usage, as well as interconnect and I/O.



Detailed Reports: The reports include general information (left), colorplots for all (CPU, GPU, I/O, etc.) metrics (center), interconnect data and job timeline (right), and associated error messages, if available.

Implementation

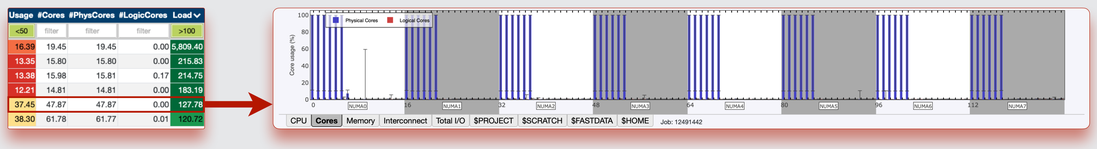
To generate and update the web portal and reports for all important jobs every few minutes, LLview uses an SQLite database. This database is filled and managed simultaneously on all cores of a single server to speed up processing. To improve performance further, the web portal is static and separate from the database. This setup guarantees predictable update rates and response times.



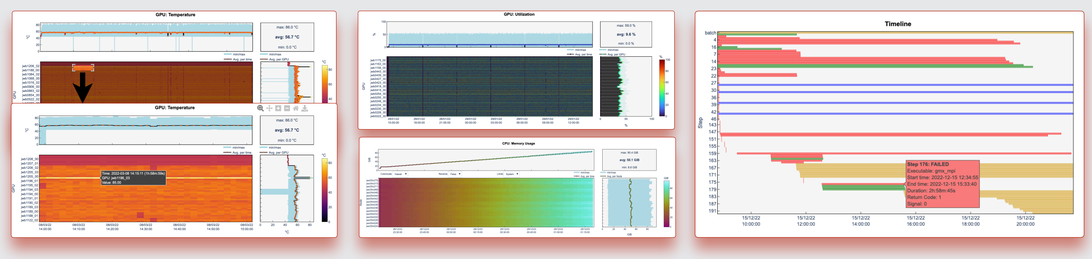
Data flow, lifetime and update cycles of LLview

Results

Support personnel often encounter issues using LLview—either in routine proactive checks or when contacted by users. The information used to identify them is also available to the users, enabling them to recognise the problems on their own (and probably many already do). To demonstrate the effectiveness of our approach, we present some observations below.



Core usage: High CPU load coexisting with low core usage (left) indicates underutilization and unused cores (right). This results in oversubscribed cores, with processes competing for resources. Addressing such issues has resulted in speedups of up to a factor of 32 for users.



(Left) GPU Temperature: The zoomed graph shows a single GPU with high temperature causing frequency throttling and job slowdown. **(Center) Resource Utilisation:** A job with a single GPU per node when four are available (top), and another exhibits increasing memory usage, indicating a possible memory leak (bottom). **(Right) Sub-Scheduling:** A job with multiple steps may have only some failing. Additional details for each step are available on LLview's HTML reports by hovering over the timeline.

Conclusion and Outlook

Our LLview monitoring solution enables quick issue identification and system optimization for users, support teams, and admins, offering a holistic view of HPC systems. It's widely used at the Jülich Supercomputing Centre, aiding in issue resolution and performance improvement. Designed for scalability, it's ready for integration with Europe's first exascale computer, JUPITER [2]. LLview also supports an independent database accessed via REST API, enabling regular AI analysis and automatic reporting to stakeholders. Its recent open-source release [3] fosters global collaboration with HPC centers.

References and Acknowledgements

[1] https://llview.fz-juelich.de.

[2] http://jupiter.fz-juelich.de.

[3] http://github.com/FZJ-JSC/LLview.

Contact: Filipe Guimarães

E-mail: f.guimaraes@fz-juelich.de

This work was supported by the State of North Rhine-Westphalia for application support of the JUWELS Cluster and Booster (SiVeGCS) and by the European Union's Horizon 2020 research and innovation program under grant agreements No. 955606 (DEEP-SEA), No. 955811 (IO-SEA) and No. 101033975 (EUPEX).