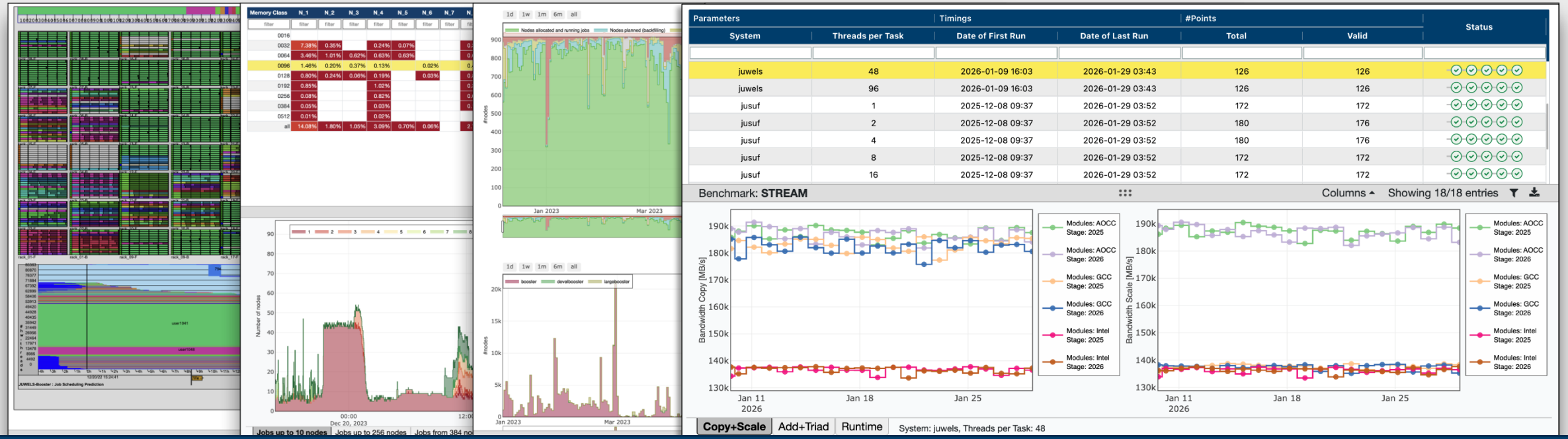


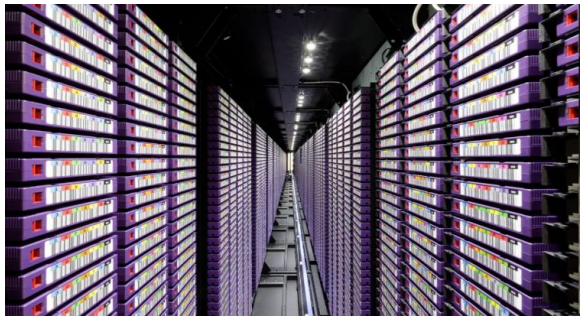
Centralised Dashboard for Continuous Benchmarking: From HPC Clusters to Quantum Processors

HPC Knowledge Portal annual meeting 2026

17.06.2026 I THOMAS BREUER | JÜLICH SUPERCOMPUTING CENTRE

LLview: Filipe S. M. Guimarães, Wolfgang Frings
QC: Ashwin K. Karnad, Carlos D. Gonzalez Calaza

JSC HPC System Overview

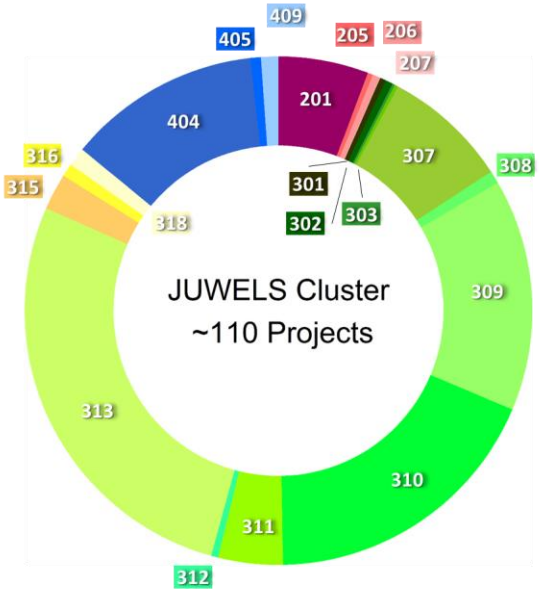


Bull ParTec MODULAR SUPERCOMPUTING SIPEARL Red Hat IBM MEGWARE

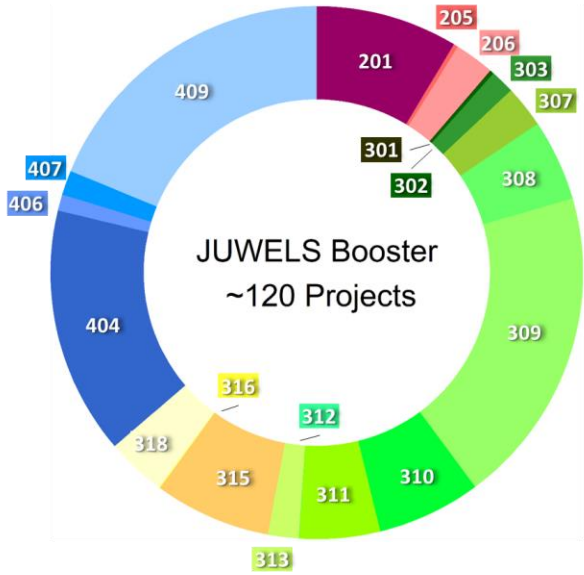
NVIDIA slurm workload manager AMD intel Rocky Linux pro-com DATENSYSTEME Lenovo

doitnow Mellanox TECHNOLOGIES JÜLICH Forschungszentrum

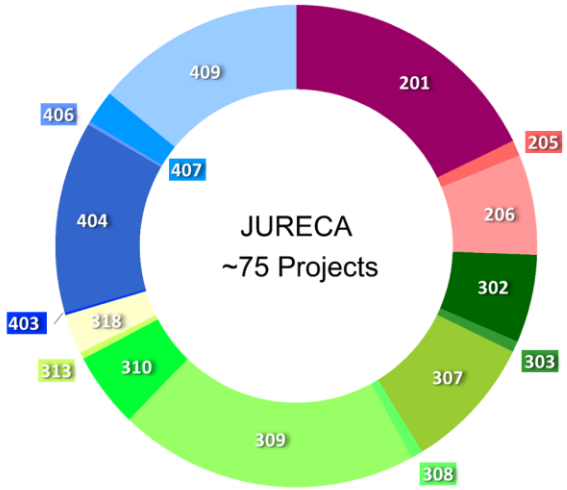
JSC HPC System Usage Overview



Reseach Fields
- 201 Basic Biological and Medical Research
- 205 Medicine
- 206 Neurosciences
- 207 Agriculture, Forestry and Veterinary Medicine
- 301 Molecular Chemistry
- 302 Chemical Solid State and Surface Research
- 303 Physical and Theoretical Chemistry
- 306 Polymer Research
- 307 Condensed Matter Physics
- 308 Optics, Quantum Optics and Physics of Atoms, Molecules and Plasmas
- 309 Particles, Nuclei and Fields
- 310 Statistical Physics, Soft Matter, Biological Physics, Nonlinear Dynamics
- 311 Astrophysics and Astronomy
- 312 Mathematics
- 313 Atmospheric Science, Oceanography and Climate Research
- 315 Geophysics and Geodesy
- 316 Geochemistry, Mineralogy and Crystallography
- 318 Water Research
- 404 Heat Energy Technology, Thermal Machines, Fluid Mechanics
- 405 Materials Engineering
- 409 Computer Science



Reseach Fields
- 201 Basic Biological and Medical Research
- 205 Medicine
- 206 Neurosciences
- 301 Molecular Chemistry
- 302 Chemical Solid State and Surface Research
- 303 Physical and Theoretical Chemistry
- 307 Condensed Matter Physics
- 308 Optics, Quantum Optics and Physics of Atoms, Molecules and Plasmas
- 309 Particles, Nuclei and Fields
- 310 Statistical Physics, Soft Matter, Biological Physics, Nonlinear Dynamics
- 311 Astrophysics and Astronomy
- 312 Mathematics
- 313 Atmospheric Science, Oceanography and Climate Research
- 315 Geophysics and Geodesy
- 316 Geochemistry, Mineralogy and Crystallography
- 318 Water Research
- 404 Heat Energy Technology, Thermal Machines, Fluid Mechanics
- 406 Materials Science
- 407 Systems Engineering
- 409 Computer Science



Reseach Fields
- 201 Basic Biological and Medical Research
- 205 Medicine
- 206 Neurosciences
- 302 Chemical Solid State and Surface Research
- 303 Physical and Theoretical Chemistry
- 307 Condensed Matter Physics
- 308 Optics, Quantum Optics and Physics of Atoms, Molecules and Plasmas
- 309 Particles, Nuclei and Fields
- 310 Statistical Physics, Soft Matter, Biological Physics, Nonlinear Dynamics
- 313 Atmospheric Science, Oceanography and Climate Research
- 318 Water Research
- 403 Process Engineering, Technical Chemistry
- 404 Heat Energy Technology, Thermal Machines, Fluid Mechanics
- 406 Materials Science
- 407 Systems Engineering
- 409 Computer Science

All active, peer-reviewed projects (no test or prepaccess) from Nov/25 until Apr/26

Analog quantum computers

D-Wave

Quantum annealer
(non-universal)
>4400 superconducting flux qubits
Hosting since 2025

Pasqal

Quantum simulator
(non-universal)
100 neutral atom qubits
Hosting since 2025

Universal quantum computers

CQC Project

5 superconducting qubits
Hosting since 2023

eleQtron

32 trapped ion qubits
gate-based
Coming 2026

IQM

5 superconducting qubits
gate-based
Hosting since 2023

ARQUE SYSTEMS

SOLID

JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

Motivation

Users	Developers	Operators
New configurations, changing libraries, …	New features, refactoring code, …	Update system, software stack, …

➡ **May cause silent regressions**

➡ **Solution: Periodic tests → Continuous Benchmarks**

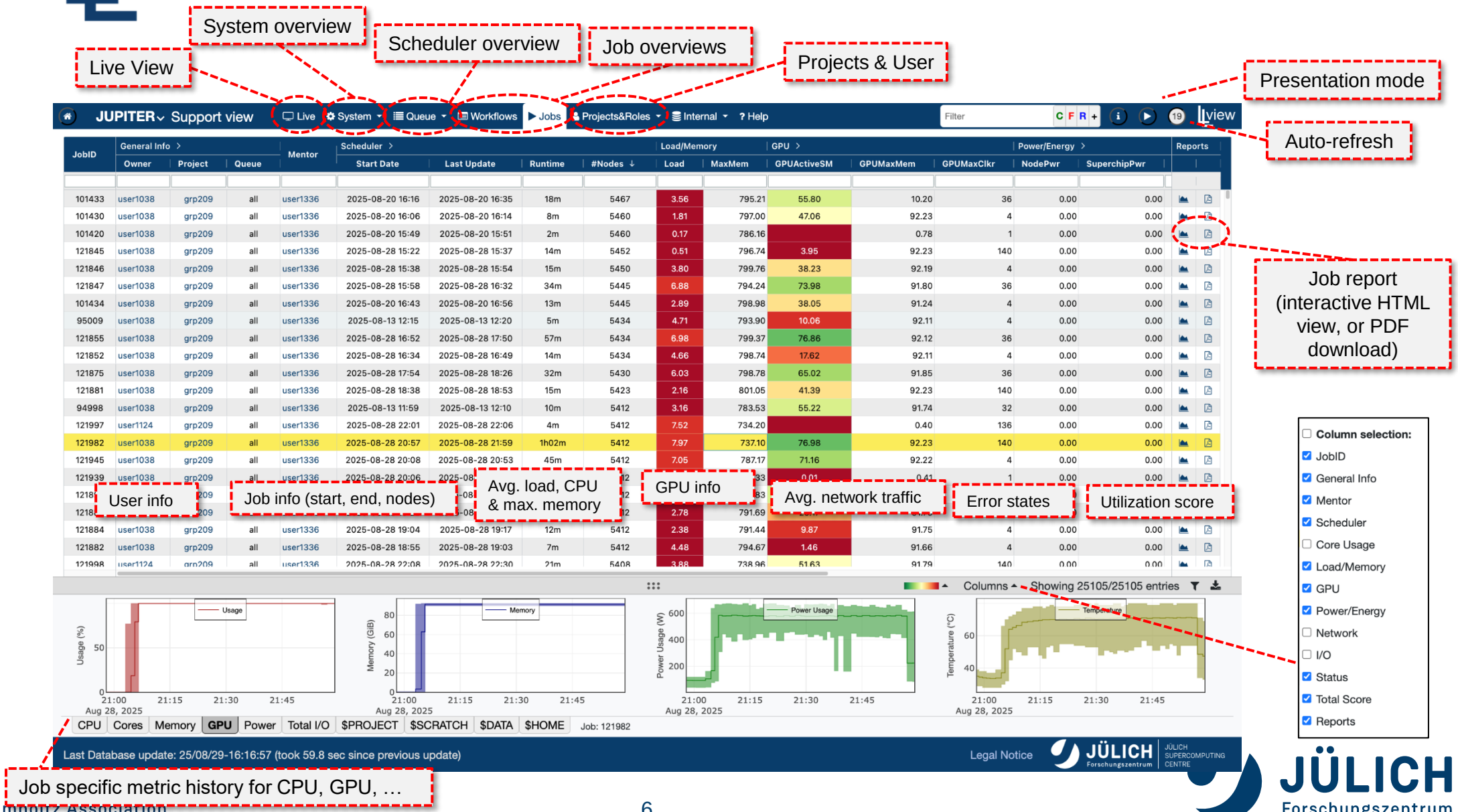
- Run the tests: store output files (e.g., into repositories or databases)
- Validate the results: check if data is correct or as expected
- Visualise data: compare historical jobs and see trends, especially with many components

➡ **Manually plotting or setting up a dashboard is not easy**

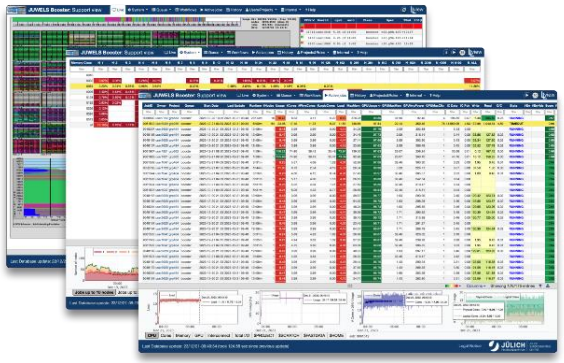
➡ **Performance drift goes unmonitored**

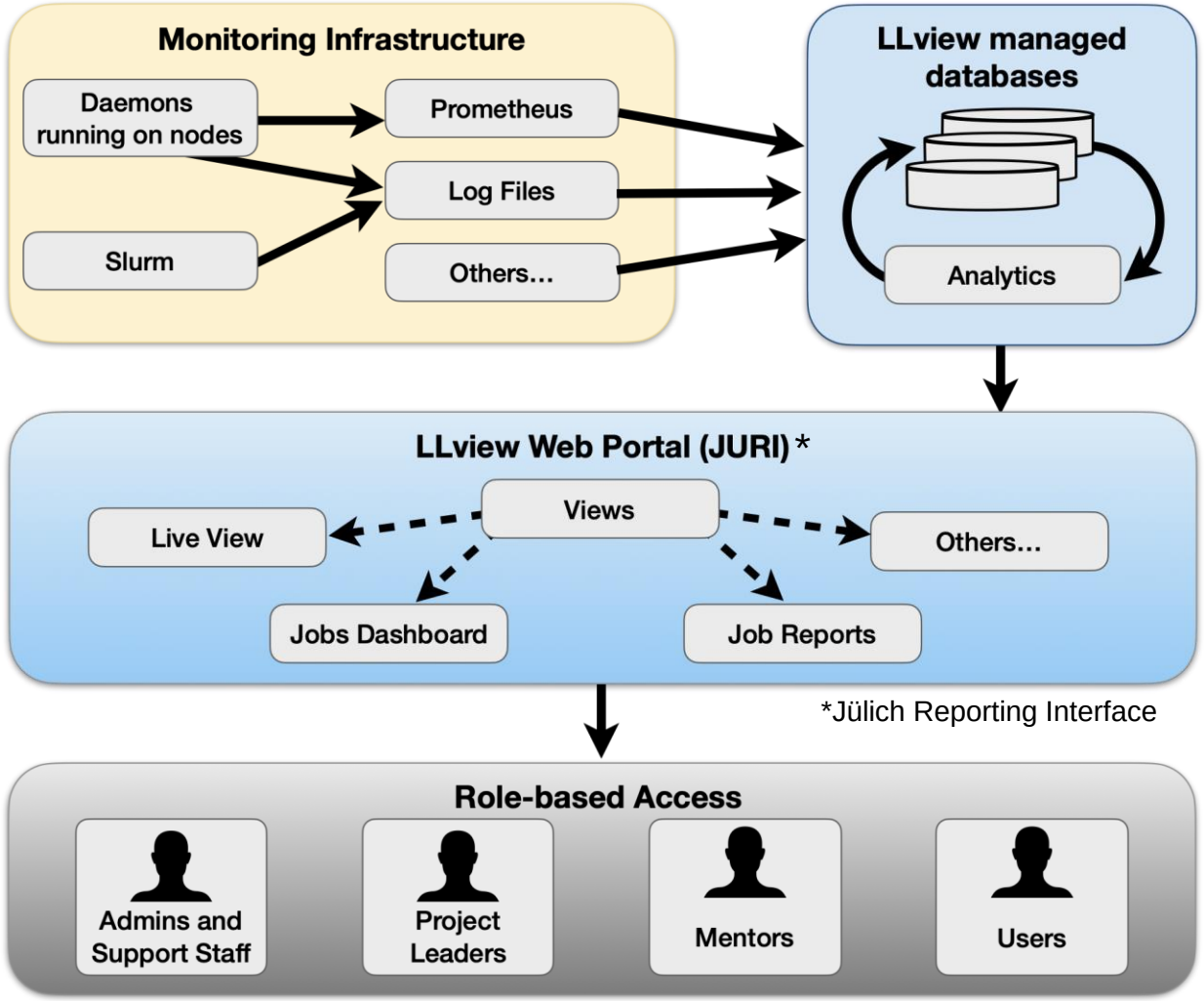
What we need: Turn those scattered, raw files into interactive, historical dashboards—without writing any frontend code.

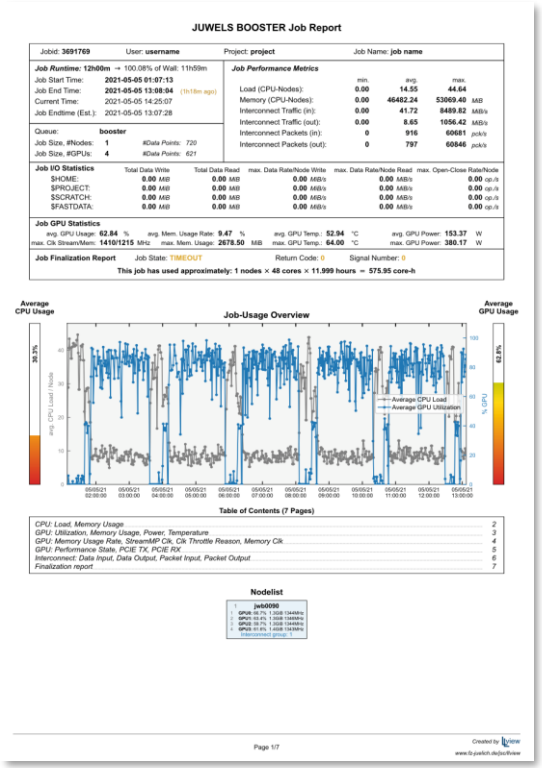
llview : Generic and configurable dashboard



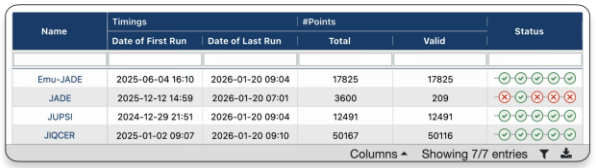
Simplified Scheme of LLview

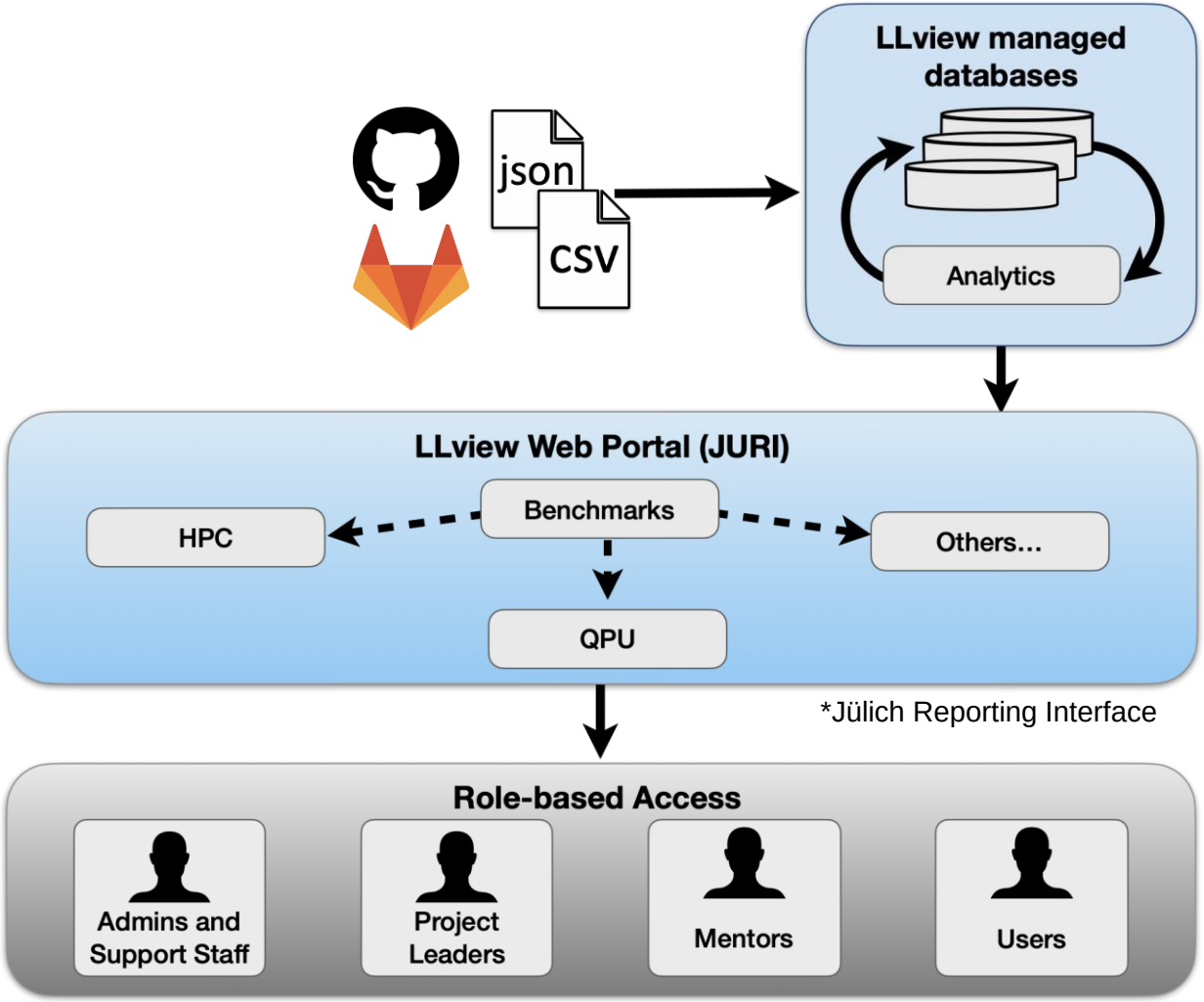


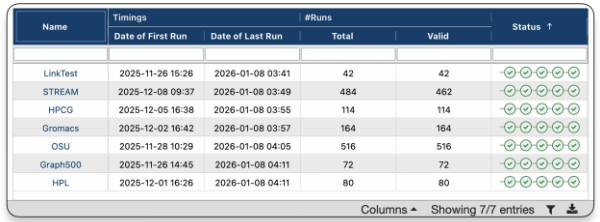


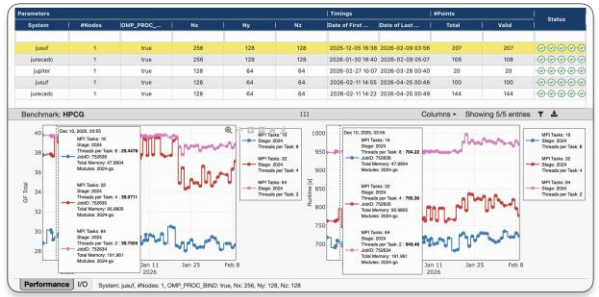


Simplified Scheme of LLview

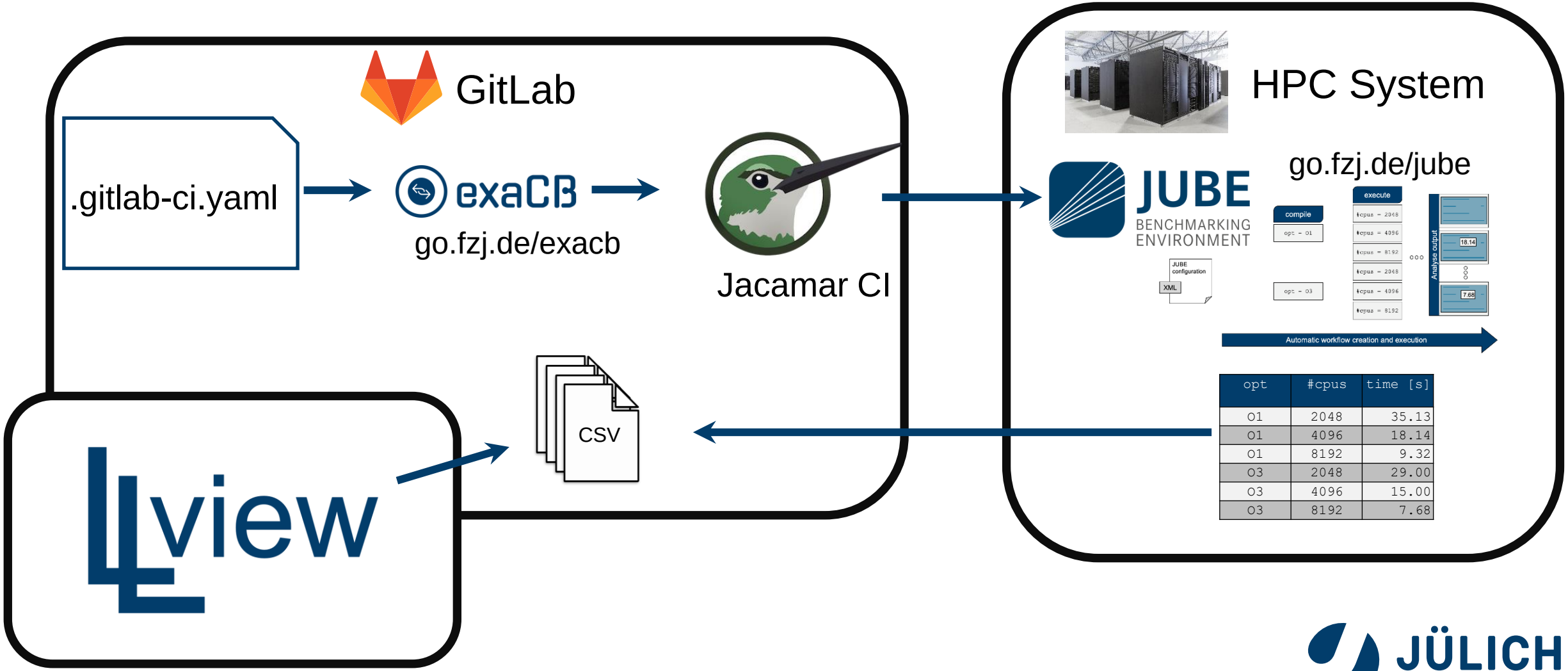








Central Visualisation for Continuous Benchmarking



opt	#cpus	time [s]
O1	2048	35.13
O1	4096	18.14
O1	8192	9.32
O3	2048	29.00
O3	4096	15.00
O3	8192	7.68

LLview provides a configuration-driven framework

1) Where is the data

```
HPCG:
  host: 'https://gitlab.jsc.fz-juelich.de/benchmarks/hpcg.git'
  branch: 'Results'
  token: "<your_token_here"
  sources: # Where files are located
    folders:
      - 'Results/jubench-hpcg'
    include: # Only include full paths with this pattern
      - '.*\.csv$'
```

LLview provides a configuration-driven framework

1) Where is the data

```yaml
HPCG:
  host: 'https://gitlab.jsc.fz-juelich.de/benchmarks/hpcg.git'
  branch: 'Results'
  token: "<your_token_here"
  sources: # Where files are located
    folders:
      - 'Results/jubench-hpcg'
    include: # Only include full paths with this pattern
      - '.*\.csv$'
```

2) What and how to parse

```yaml
metrics: # Definition of the metrics
  System:
    from: filename
    regex: 'jubench-hpcg_(.+?)_.+?\.csv'
    description: "System where the benchmark was run"
  OMP_PROC_BIND:
    header: 'OMP_PROC_BIND'
    description: 'Type of OpenMP Processor Binding'
    type: str
  Nx:
    header: 'nx'
    description: 'NX'
    type: int
  Ny:
    header: 'ny'
    description: 'NY'
    type: int
  Nz:
    header: 'nz'
    description: 'NZ'
    type: int
  Stage:
    header: 'stage'
    description: 'Stage'
    type: int
    default: 2024
  JobID:
    header: 'job_id'
    type: int
  '#Nodes':
    header: 'nodes'
    description: 'Number of
    type: int
  ts:
    from: filename
    regex: 'jubench-hpcg_.+?_(.+?)\.csv'
```

Results / jube-bench / Results / jubench-hpcg

jubench-hpcg

Name
..
jubench-hpcg_jupiter_2026-02-27T10:07+01:00.csv
jubench-hpcg_jupiter_2026-02-27T11:27+01:00.csv
jubench-hpcg_jupiter_2026-03-10T09:18+01:00.csv
jubench-hpcg_jupiter_2026-03-10T10:03+01:00.csv
jubench-hpcg_jupiter_2026-03-10T22:09+01:00.csv
jubench-hpcg_jupiter_2026-03-14T00:39+01:00.csv

Results / jube-bench / Results / jubench-hpcg / jubench-hpcg_jupiter_2026-02-27T11:27+01:00.csv

jubench-hpcg_jupiter_2026-02-27T11:27+01:00.csv — Find file | Blame | Edit

jubench-hpcg_jupiter_2026-02-27T11:27+01:00.csv 696 B — Code | Preview

jube_benchmark_id	system	modules	stage	nodes	taskspernode	threadspertask	OMP_PROC_BIND	nx	ny	nz	tot_mem	GB_read
0	JUPITER	GCC/13.3.0 OpenMPI UCX UCX-settings/RC	2025	1	144	2	true	128	64	64	54.0211	1204.86
0	JUPITER	GCC/14.3.0 OpenMPI UCX UCX-settings/RC	2026	1	144	2	true	128	64	64	54.0211	1245.65

LLview provides a configuration-driven framework

1) Where is the data

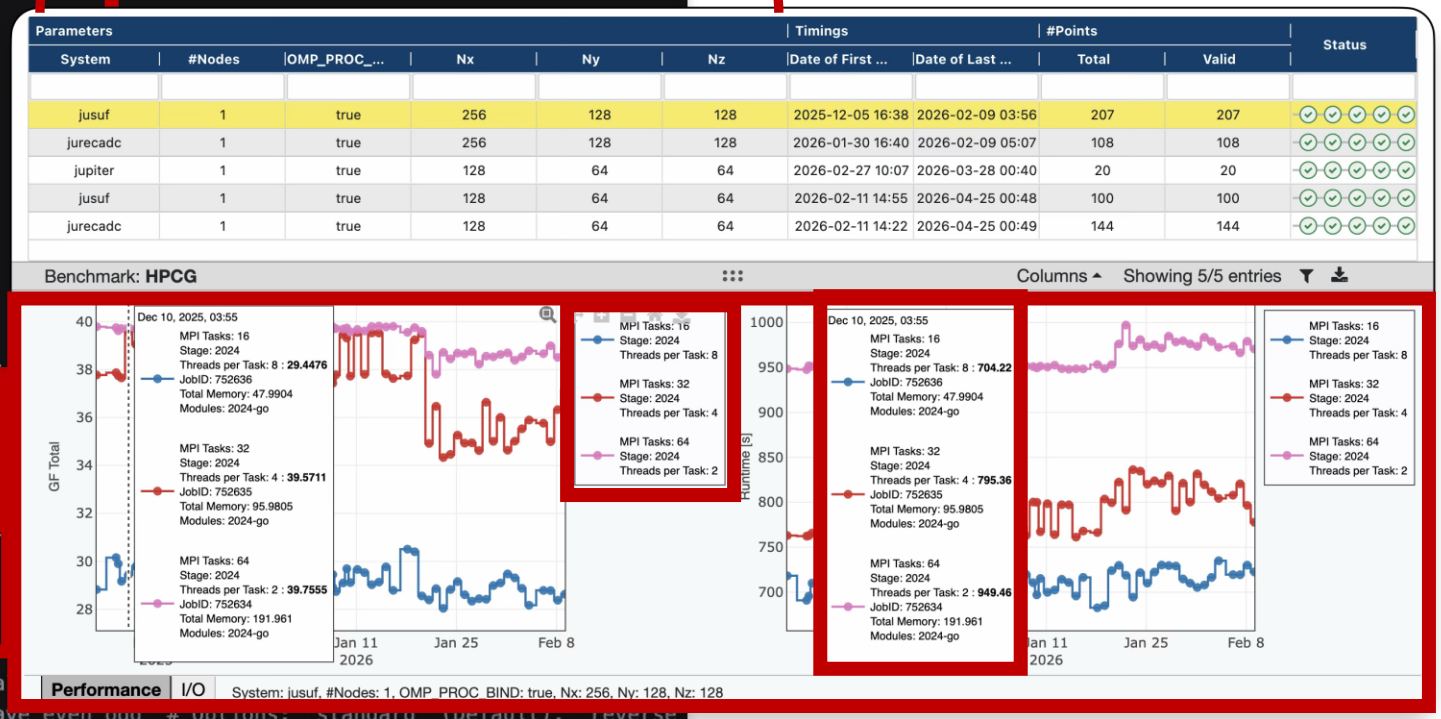
```yaml
HPCG:
  host: 'https://gitlab.jsc.fz-juelich.de/benchmarks/hpcg.git'
  branch: 'Results'
  token: "<your_token_here"
  sources: # Where files are located
    folders:
      - 'Results/jubench-hpcg'
    include: # Only include full paths with this pattern
      - '.*\.csv$'
```

2) What and how to parse

```yaml
metrics: # Definition of the metrics
  System:
    from: filename
    regex: 'jubench-hpcg_(.+?)_.+?\.csv'
    description: "System where the benchmark was run"
  OMP_PROC_BIND:
    header: 'OMP_PROC_BIND'
    description: 'Type of OpenMP Processor Binding'
    type: str
  Nx:
    header: 'nx'
    description: 'NX'
    type: int
  Ny:
    header: 'ny'
    description: 'NY'
    type: int
  Nz:
    header: 'nz'
    description: 'NZ'
    type: int
  Stage:
    header: 'stage'
    description: 'Stage'
    type: int
    default: 2024
  JobID:
    header: 'job_id'
    type: int
  '#Nodes':
    header: 'nodes'
    description: 'Number of nodes used in the job'
    type: int
  ts:
    from: filename
    regex: 'jubench-hpcg_.+?_(.+?)\.csv'
```

```yaml
table: # Metrics that will be placed as table columns
  - System
  - '#Nodes'
  - OMP_PROC_BIND
  - Nx
  - Ny
  - Nz
plots: # Configuration of footer plots
  tabs: # Organize plots into tabs
    Performance:
    - x: ts
      y: 'GF Total'
    - x: ts
      y: Runtime
    I/O:
    - x: ts
      y: 'Read Data'
    - x: ts
      y: 'Write Data'
    - x: ts
      y: 'Total Data'
  group_by: # Metrics that sh
    - 'MPI Tasks'
    - 'Threads per Task'
    - Stage
  annotations: # Metrics to b
    - JobID
    - 'Total Memory'
    - Modules
  # colormap: 'Accent' # Qua
  # sort_strategy: 'interleave_even_odd' # Options: 'standard' (default), 'reverse'
  skip: # Colours to skip (Default: [])
    - '#ffff99'
styles: # Styles for the trac
  mode: 'lines+markers'  # Ov
  marker:
```

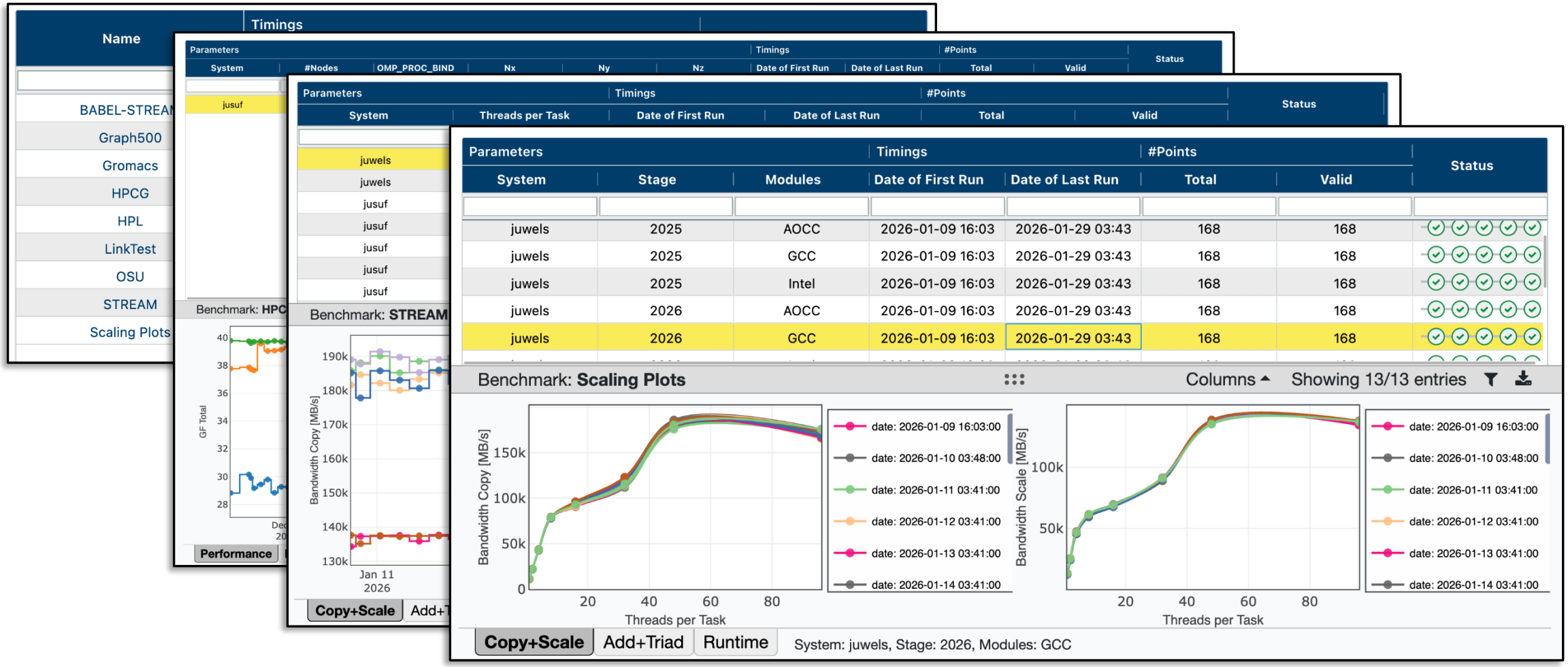
3) How to visualise



Documentation: llview.fz-juelich.de/docs

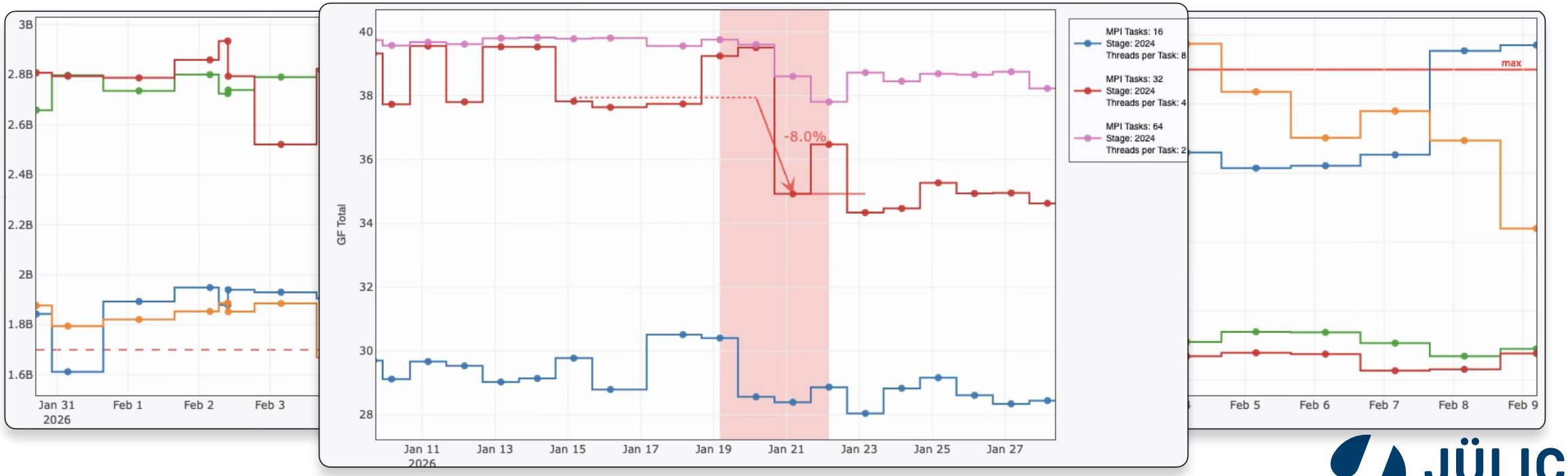
JÜLICH Forschungszentrum

LLview is a Central Visualization for HPC Benchmarks



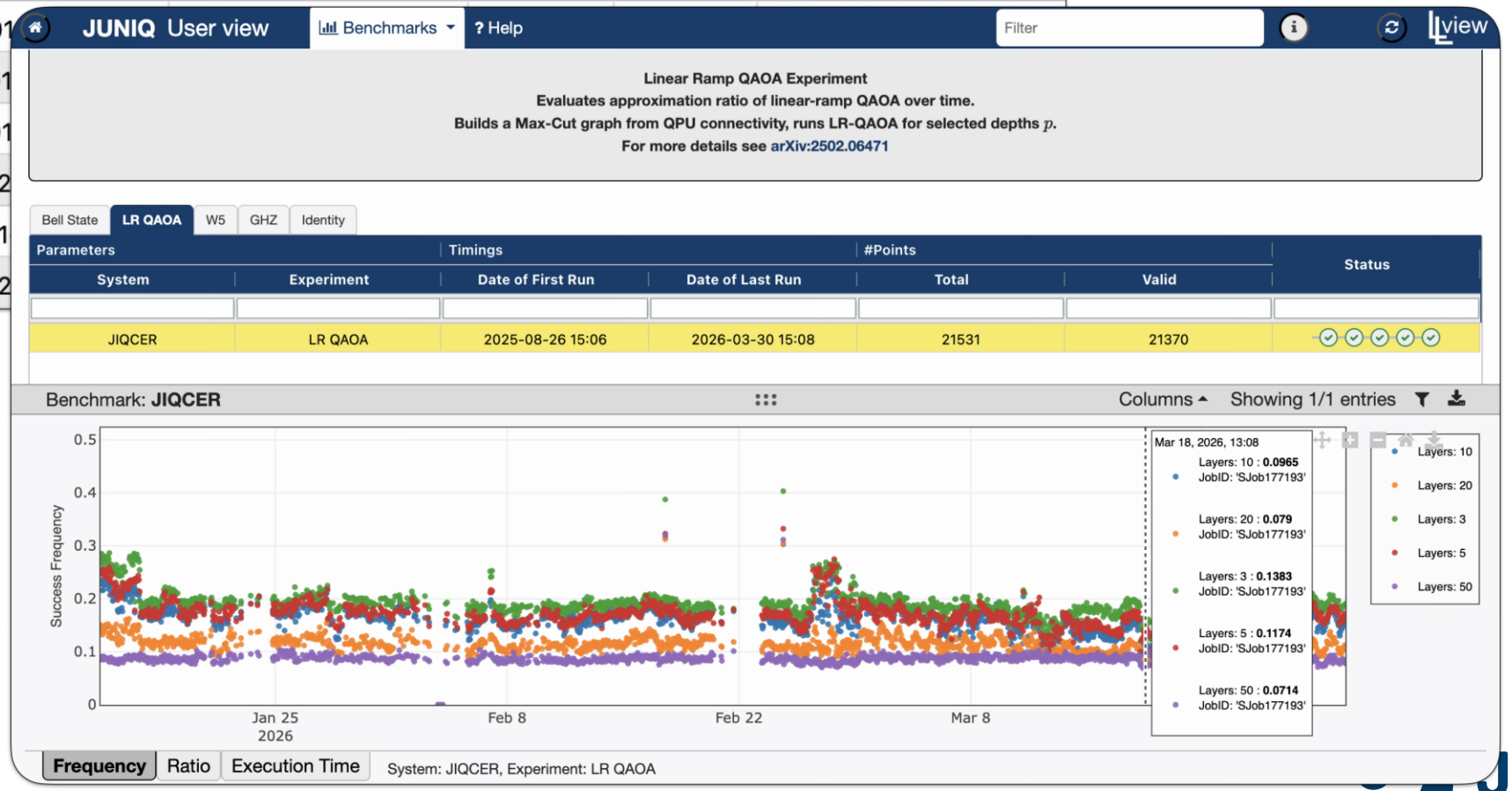
Benchmark data validation

- Intrinsic validation functions:
  - Range validator (data is above min and/or below max)
  - Outlier detector
  - Regression detector
-



LLview is also a Central Visualization for QC Benchmarks

Name	Timings		#Points		Status
	Date of First Run	Date of Last Run	Total	Valid	
CQC	2026-01				
Emu-JADE	2026-01				
Emu-JION	2026-01				
JADE	2025-12				
JIQCER	2025-01				
JUPSI	2024-12				



Bell State Benchmark

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

1. QPU or Qaptiva Maintenance
2. Uncontrolled temperature ramps
3. Single qubit mis-calibration events
4. Change default transpilation level / introducing transpilation control (like –O3 to –O0)

Identity Benchmark

Measure gate error accumulation:

- Initialize |1⋯1⟩ with X gates
- Apply 2k additional X gates (mathematically identity for any k)
- Measure frequency of |1⋯1⟩ vs circuit depth

1. Highest transpilation level (zero-operation)
2. Only applied 2 (of 5) qubits -> less errors
3. Applied to all 5 qubits
4. State preparation error (depth: 0)
5. In general: Gate error accumulation with increased depth (#rotations)

Outlook

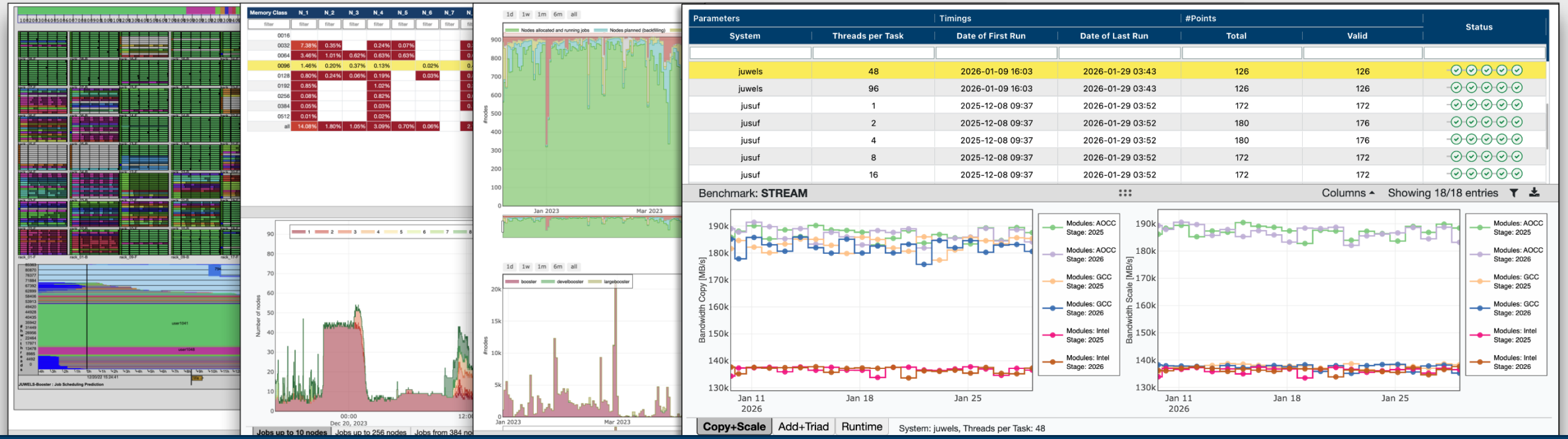
- Add other intrinsic methods of validation of the data (regression and outlier detection)
  - ➢ Implemented; not yet released
- Complete integration of JUPITER Benchmark Suite https://github.com/FZJ-JSC/jubench
- Make available for User and Project View on LLview

	Booster			Cluster
Name	GPU	GPU High-Scale	CPU	CPU
Amber	✓			
Arbor	✓	✓		
Chroma	✓	✓		
Gromacs	✓			
ICON	✓			
JUQCS	✓	✓		
nekRS	✓	✓		
ParFlow	✓			
PIConGPU	✓	✓		
Quantum ESPRESSO	✓			
SOMA	✓			
AI-MMoCLIP	✓			
AI-NLP	✓			
AI-ResNet	✓			
dynQCD				✓
NAStJA				✓
Graph500			✓	
HPCG	✓			✓
HPL	✓			✓
IOR			✓	✓
LinkTest			✓	✓
Multi-Flow IP			✓	
OSU	✓		✓	✓
STREAM	✓			✓

Conclusions

- We created a flexible framework that lowers the entry barrier for longitudinal analysis
- Decoupled benchmark execution from visualization
- Automated data validation and historical context metadata (e.g., Job ID, Commit Hash)
- Empower hosting entities and users to catch performance changes

- Open source (GPLv3): github.com/FZJ-JSC/llview
- Website: llview.fz-juelich.de
- Documentation: llview.fz-juelich.de/docs
- Contact: llview.jsc@fz-juelich.de



THANK YOU!

See you at ISC: posters, talk & booths

LLview: Filipe S. M. Guimarães, Wolfgang Frings

QC: Ashwin K. Karnad, Carlos D. Gonzalez Calaza