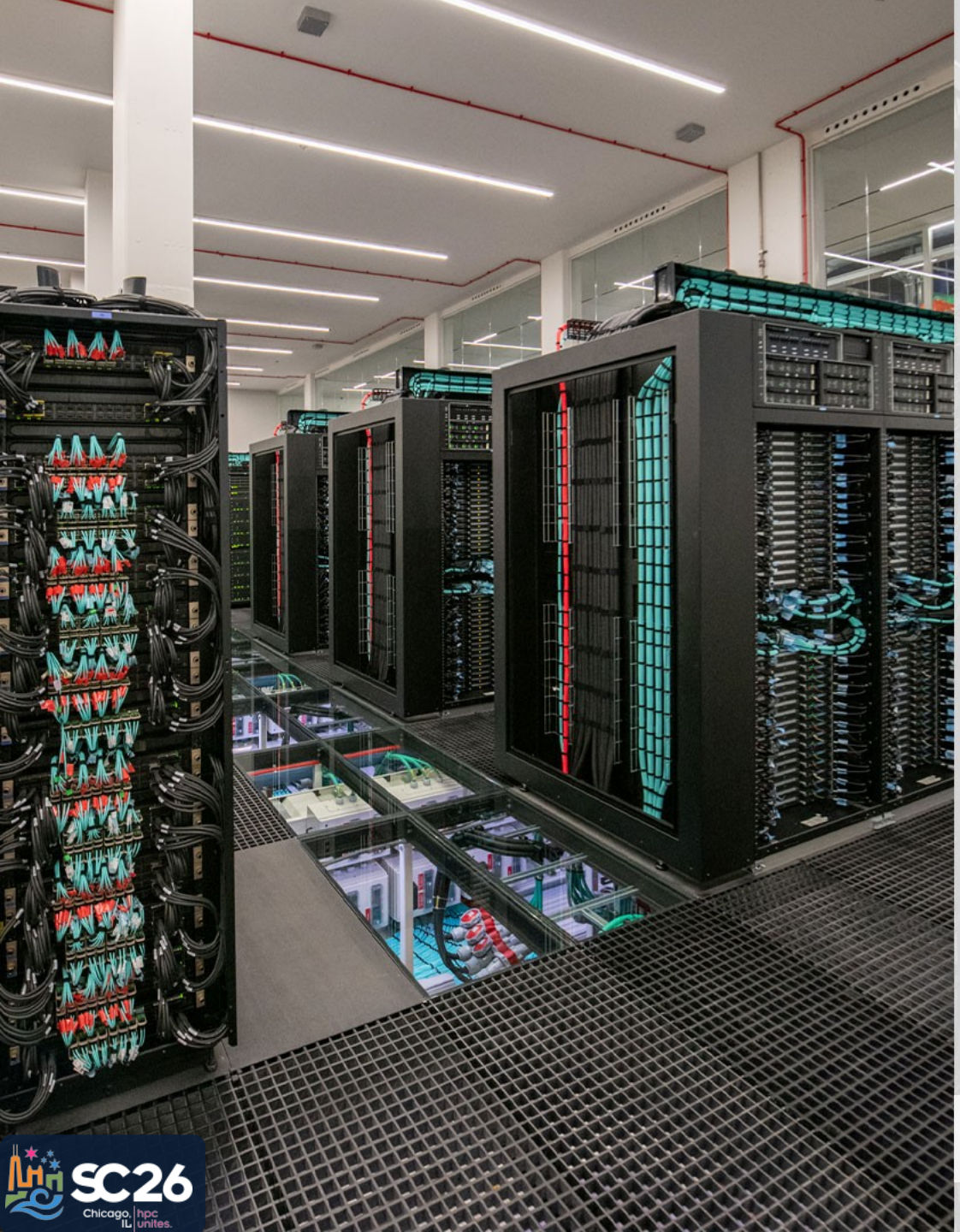


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Interoperable Performance Prediction through Standardized Metadata Capture and Analysis

Nicolò Giacomini, **Débora Pina**, Rosa M. Badia, Raül Sirvent

21st Workshop on Workflows in Support of Large-Scale Science

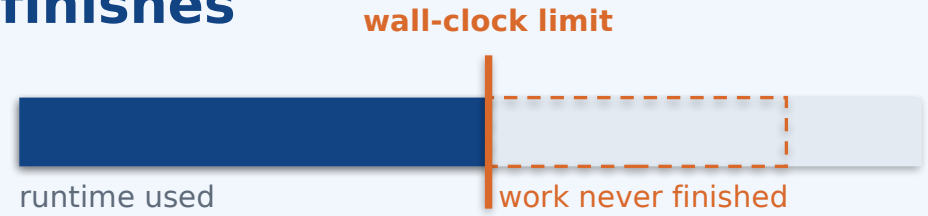
SC26 Chicago, IL hpc unites.

Every HPC job starts with a guess

```
#SBATCH --time=??:??:??
```

GUESS TOO LOW

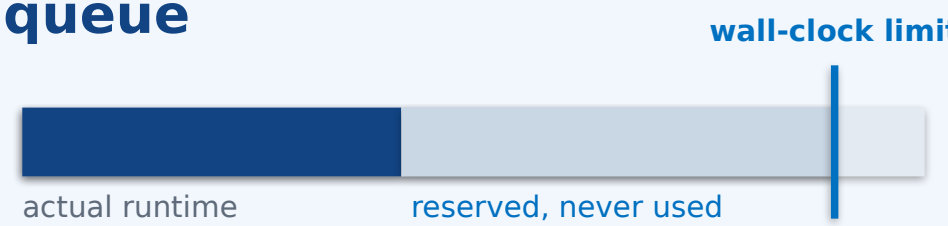
The job is killed before it finishes



Every resource it consumed is wasted.

GUESS TOO HIGH

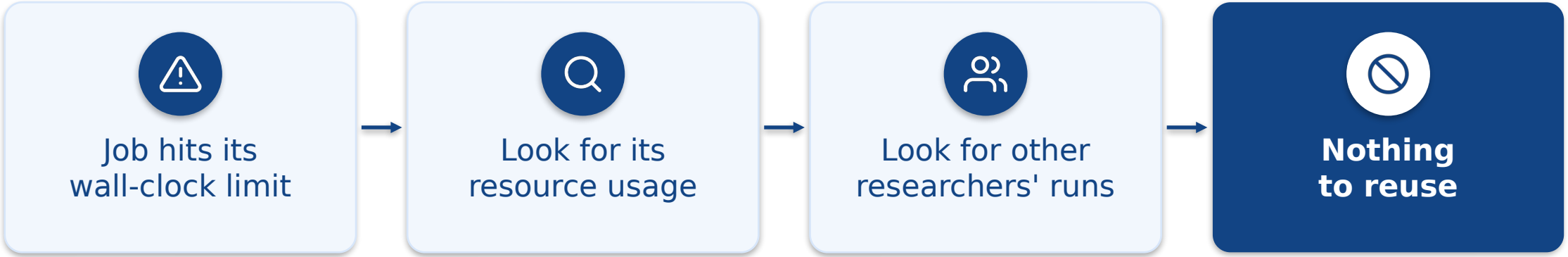
The job waits longer in the queue



The scheduler backfills around a reservation that is never used.

SLURM asks for a wall-clock limit before execution, yet parallel applications rarely scale linearly and users estimate from little prior knowledge.

Failed runs leave nothing behind



When the data exists, it is scattered

- Profiler
- Scheduler
- Workflow engine
- Logs

OUR POSITION

Execution data should be a standardized, reusable artifact of every workflow run.

Our approach: a lifecycle for execution knowledge

Capture

A SLURM-aware profiler records CPU, memory, disk and network use during the run.

Standardize

Usage is linked to the workflow run and packaged in a community standard (WRROC).

Store

Crates persist in ProvStor: a queryable RDF graph plus an object store.

Reuse

Any consumer can query it; here, a hybrid runtime predictor.

COMPSs runtime

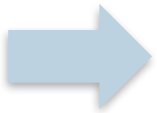
Any system that speaks WRROC

Every run feeds the shared knowledge base, and every future run can draw on it.

Knowledge that outlives the run

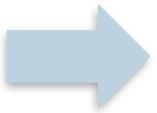
Any system can contribute

- Galaxy
- Snakemake
- StreamFlow
- Nextflow
- COMPSs





Shared WRROC knowledge base



Any tool can consume

- **Runtime predictor (ours)**
- Scheduler plugins
- Resource selection
- Performance analysis
- LLM assistant



Knowledge accumulates: every execution enriches the base and makes the next prediction better.

Capture: profiling without interfering

A separate SLURM job step

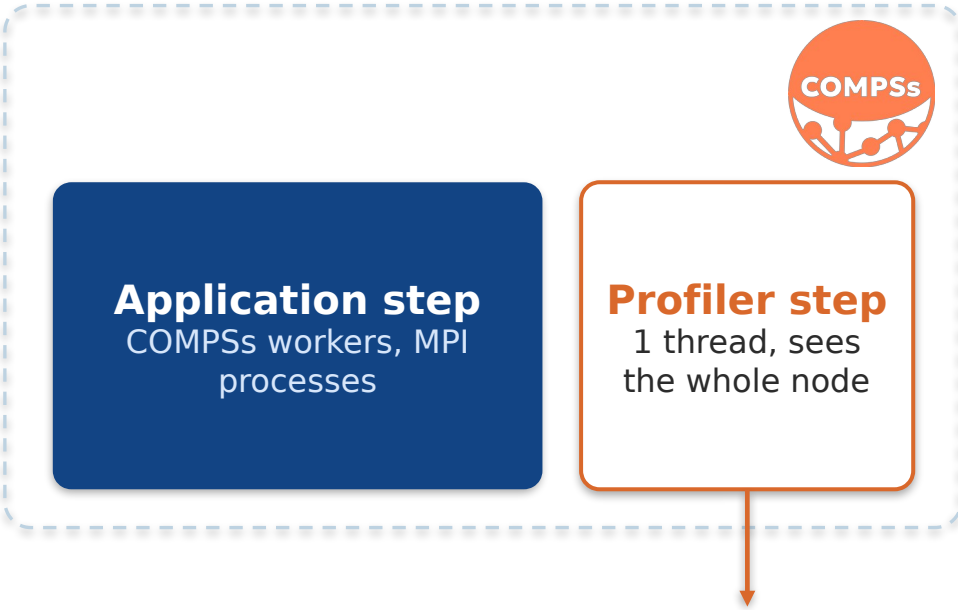
`--overlap` `--cpu-bind=none`

A single thread that reads system interfaces

Integrated into the COMPSs runtime; runs in the background

One CSV per node, sampled every 5 s

Interval set with COMPSS_PROFILING_INTERVAL



resource_profiling_{hostname}.csv

```
CPU,MEM,BYTE_SENT,BYTE_RECV,BYTE_READ_DISK,BYTE_WRITE_DISK,TIME_READ_DISK,TIME_WRITE_DISK,TIME
0.2,15.0,38690,43424,0,4096,0,0,2026-01-09 11:01:18
5.0,15.8,2869504,4889920,0,45056,0,1,2026-01-09 11:01:23
94.0,17.5,3593340,980947,2826240,1003520,12,0,2026-01-09 11:01:28
100,17.5,23196,28816,151552,127741952,29,899,2026-01-09 11:01:33
100,17.5,279244,82911,0,9789440,0,14,2026-01-09 11:01:38
```

Standardize: resource usage in a shared standard

RO-Crate

Workflow Run RO-Crate (WRROC): the community standard for describing a workflow run. Usage data goes in two forms:

Summary values → resourceUsage

```
{
  "@id": "#gs06r1b32.cpuAvg",
  "@type": "PropertyValue",
  "name": "cpuAvg",
  "propertyID": "…/ro/terms/compss#cpuAvg",
  "unitCode": "…/qudt.org/vocab/unit/PERCENT",
  "value": "82.42"
}
```

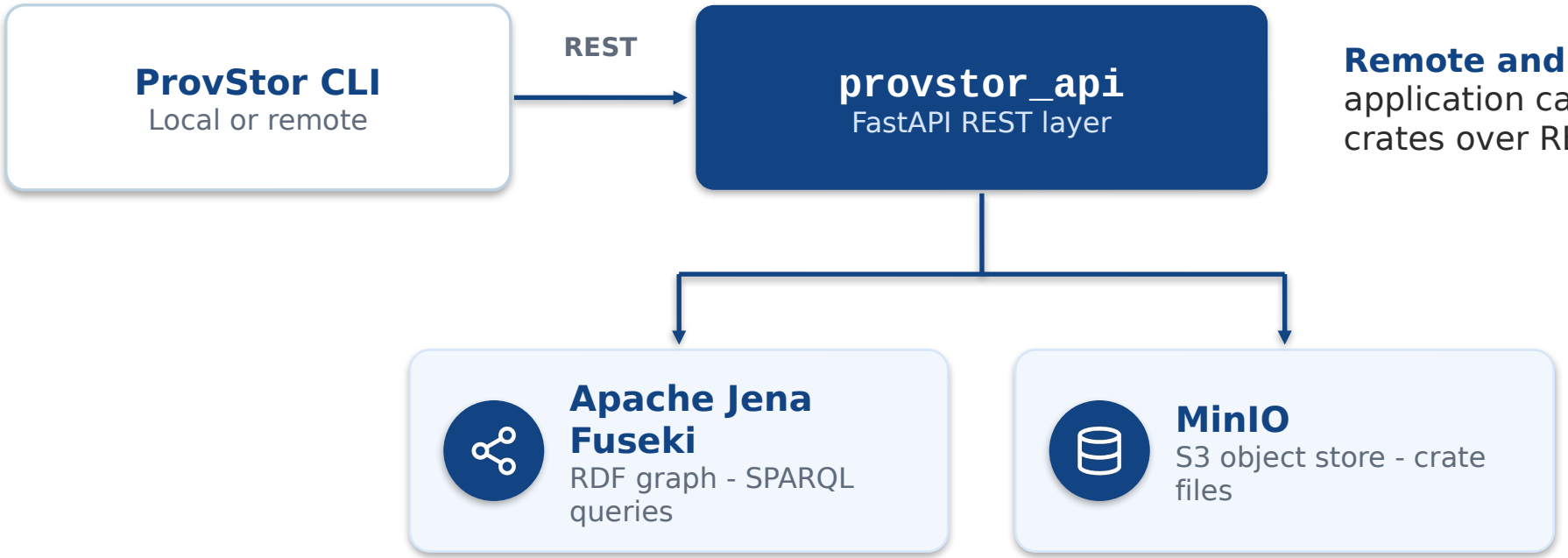
Property (node gs06r1b32)	Value	Unit
cpuAvg / cpuMax	82.42 / 99.98	%
memAvg / memMax	16.62 / 17.11	%
executionTime (run)	208,862	ms

Readings over time → CSV files

```
{
  "@id": "runtime_logs/stats/static_
          resource_profiling_gs06r1b32.csv",
  "@type": "File",
  "about": {"@id": "#COMPSs_WRROC_..._43881133"},
  "encodingFormat": ["text/csv", …],
  "description": "Resource Usage CSV file…"
}
```

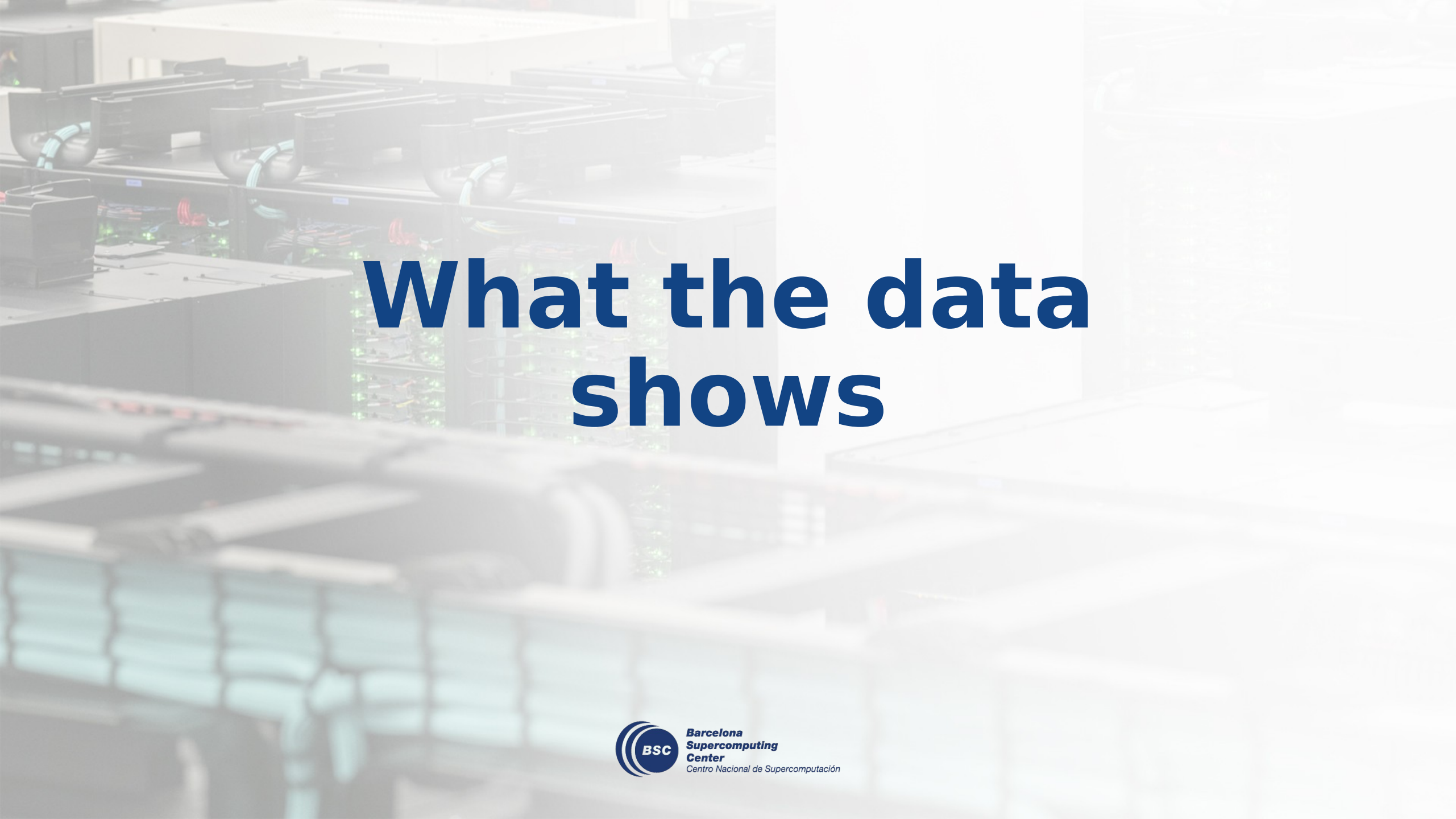
TIME	CPU (%)	MEM (%)	…
15:05:32	0.02	14.97	…
15:05:37	77.90	16.09	…
15:05:42	99.95	16.75	…

Store: a queryable, shared knowledge base



Remote and interoperable: any application can upload and query crates over REST.

Metadata becomes a graph you can query by relationship; **files** stay in object storage.



What the data shows

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Real executions with matrix multiplication

538

execution records captured on **MareNostrum 5**

511

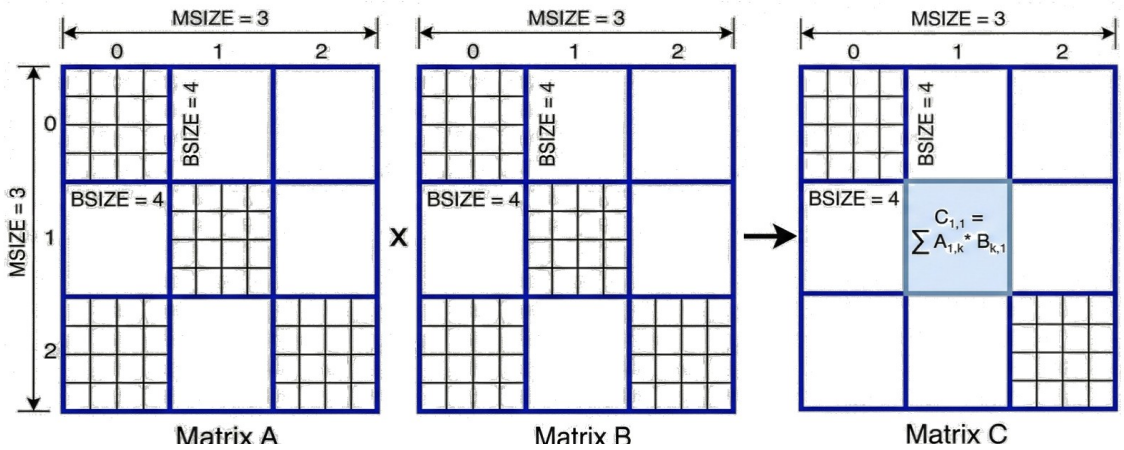
after trimming the long tail

7

features per run: workload, hardware, CPU & memory use, runtime

112

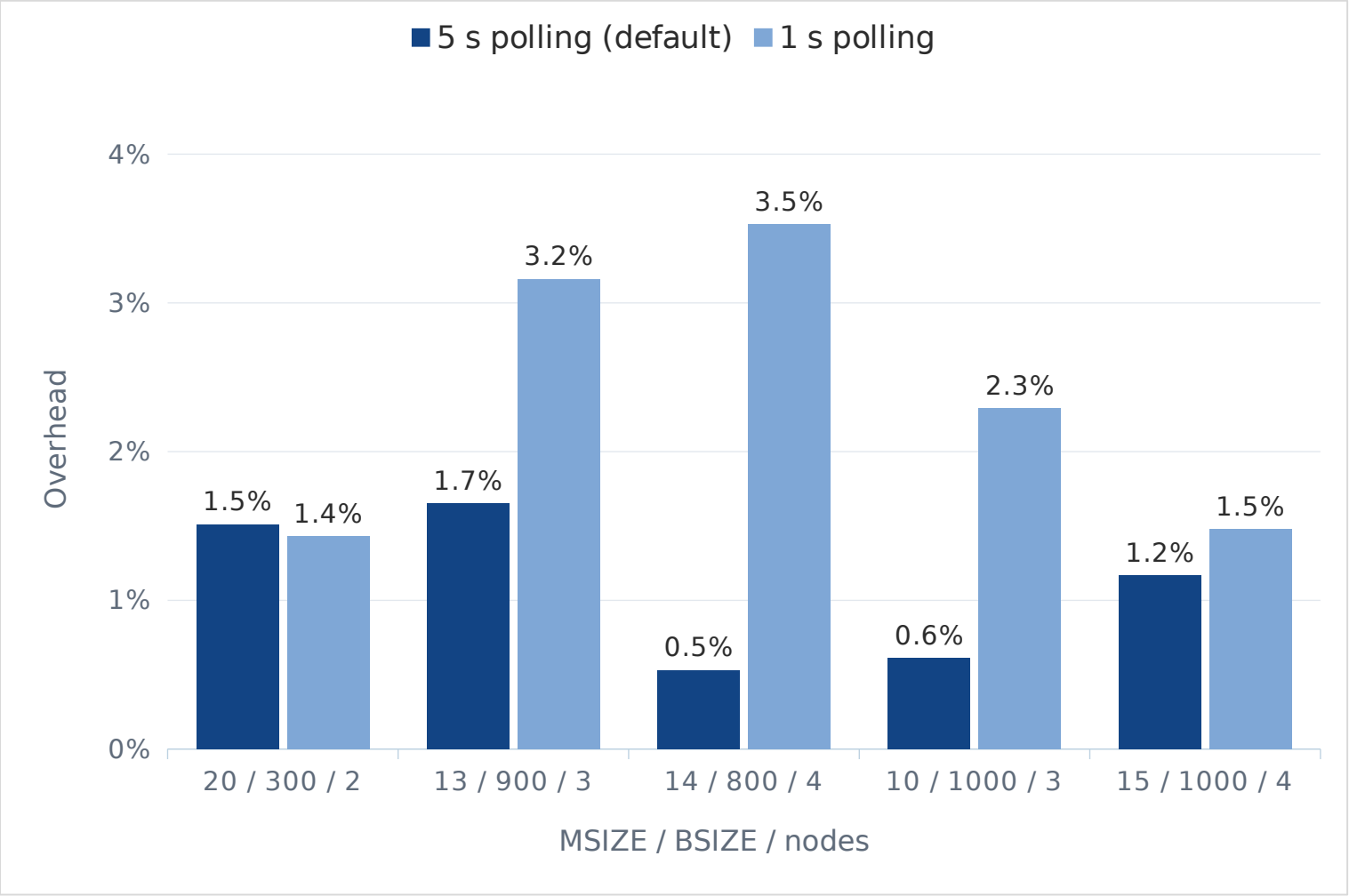
CPU-only tasks per node



MSIZE blocks per matrix side: how many tasks

BSIZE elements per block side: how much work per task

Profiling overhead is negligible



< 2%

overhead at 5 s polling

< 4%

overhead at 1 s polling

What you run drives runtime

0.877

Block size

Spearman ρ (Pearson r = 0.680): the strongest driver, monotonic but not linear

0.227

Matrix size

Pearson r: much weaker, since data dependencies limit how many tasks run at once

PCA separates three independent factors: hardware scale, number of tasks, and block size on its own.

Scheduler-level metrics alone miss the main driver: keep the workload context that workflow metadata carries.

The reused data predicts runtime safely

Physics + data

USL baseline fitted on measured speedup; XGBoost learns the residuals

Prudent by design

Asymmetric loss: underestimates, the errors that kill jobs, cost more

Robust

MAE ≈ 107 s; stable under K-fold CV, where a neural network exceeded 40%

~16%

error, K-fold CV

MSIZE	BSIZE	Nodes	Actual (min)	Predicted (min)	Error
9	500	2	3.55	3.15	**−11.23%**
22	257	2	5.17	5.32	**+2.77%**
22	600	2	76.65	82.28	**+7.34%**
43	345	5	26.66	24.40	**−8.48%**
37	200	6	3.12	3.45	**+10.46%**
70	200	8	12.92	15.38	**+19.10%**
35	200	**13**	1.91	3.83	**+100.35%**

+ overestimate (safe) − underestimate

Weak spot: very short jobs (~2 min on 13 nodes)

N. J. Gunther, A General Theory of Computational Scalability Based on Rational Functions, 2008

Conclusions and future work

What we showed

01 Resource usage can be captured with under 4% overhead and packaged in a community standard

02 Workload parameters are the main drivers of runtime

03 Standardized metadata is directly reusable, enough for safe runtime prediction

Next

- Validate on WRROC from other workflow systems
- More applications and HPC environments
- Hardware descriptors for cross-site transfer
- A dedicated technique for short jobs

Thank you!

Débora Pina
debora.pina@bsc.es

This paper has been partially supported by the projects CEX2021-001148-S, PID2023-147979NB-C21 and PCI2024-155070-2 from the MCIN/AEI and MICIU/AEI /10.13039/501100011033 and by FEDER, UE, by the Departament de Recerca i Universitats de la Generalitat de Catalunya, research group MPiEDist (2021 SGR 00412), and the Horizon Europe project ODISSEE (grant number 101188332). The views and opinions expressed are solely those of the author(s) and do not necessarily reflect those of the European Union. Neither the European Union nor the European Commission can be held responsible for them.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación