TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

CIDS ZIH Information Services and High Performance Computing

Job Monitoring using PIKA

Frank Winkler

44th VI-HPS Tuning Workshop
Dresden, 26 February 2024

HPC Performance Analysis

- Performance is crucial for HPC
- Hence, measurement, analysis, and validation are also important
- Numerous established tools are available
- Problem solved?

- ... for those who are aware of the problem
- Active preparation is necessary
- Not continuous for all jobs of all users

HPCToolkit

VAMPIR

READEX
Runtime Exploitation of Application Dynamism
for Energy-efficient eXascale computing

MAQAO

LIKWID TOOLS

scalasca

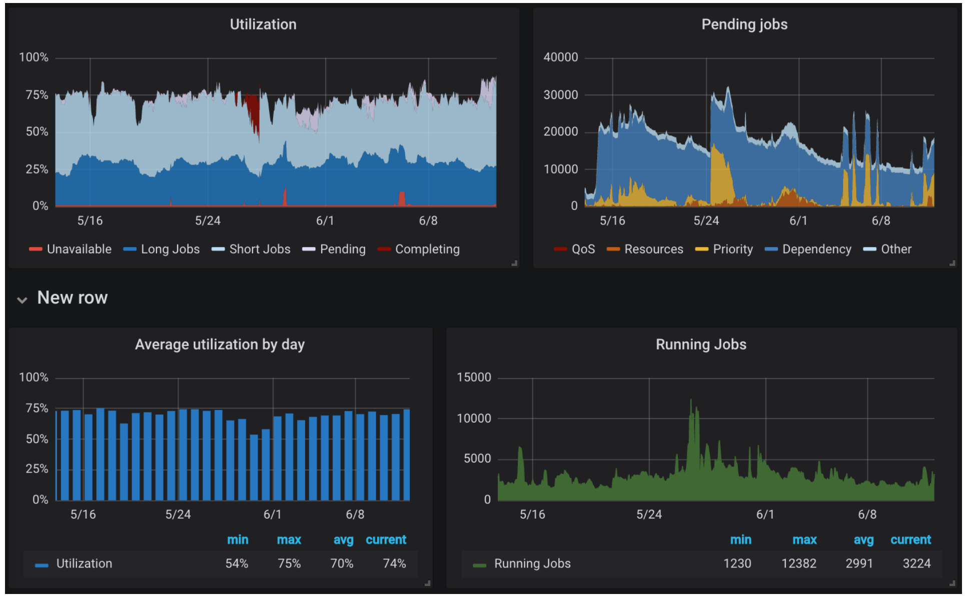
Score-P
Scalable performance measurement
infrastructure for parallel codes

ZIH

Continuous HPC Monitoring: Goals

Cluster-wide administrative view

- Existing generic solutions
- Pure hardware perspective



	min	max	avg	current
Utilization	54%	75%	70%	74%

	min	max	avg	current
Running Jobs	1230	12382	2991	3224

Grafana: Generic visualization of cluster utilization

Job-specific view for users and admins

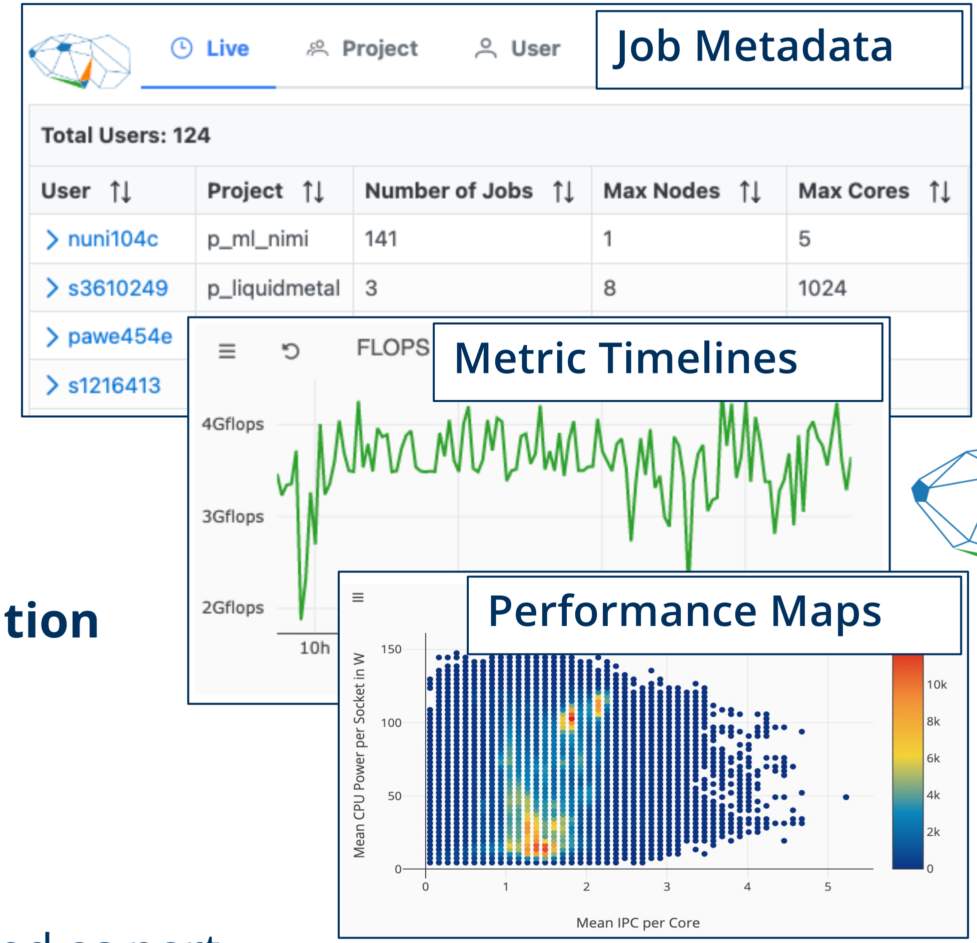
- Representation for projects, users, jobs

Continuous

- For all jobs, without explicit activation/preparation
- Minimal overhead, coarse granularity
- Performance history available on demand (retrospectively)

PIKA: Continuous HPC Job Monitoring

- Non-intrusive **data acquisition** on all cluster nodes
- Continuous **data collection**
- Web frontend for live and post-mortem **visualization**
- Detection of pathological jobs
- Automatic **job analysis and classification**
- Long-term **data storage**



Total Users: 124

User	Project	Number of Jobs	Max Nodes	Max Cores
nuni104c	p_ml_nimi	141	1	5
s3610249	p_liquidmetal	3	8	1024
pawe454e				
s1216413				

Funded by the DFG project ProPE, continued as part of NHR@TUD at ZIH.

PIKA: Continuous HPC Job Monitoring

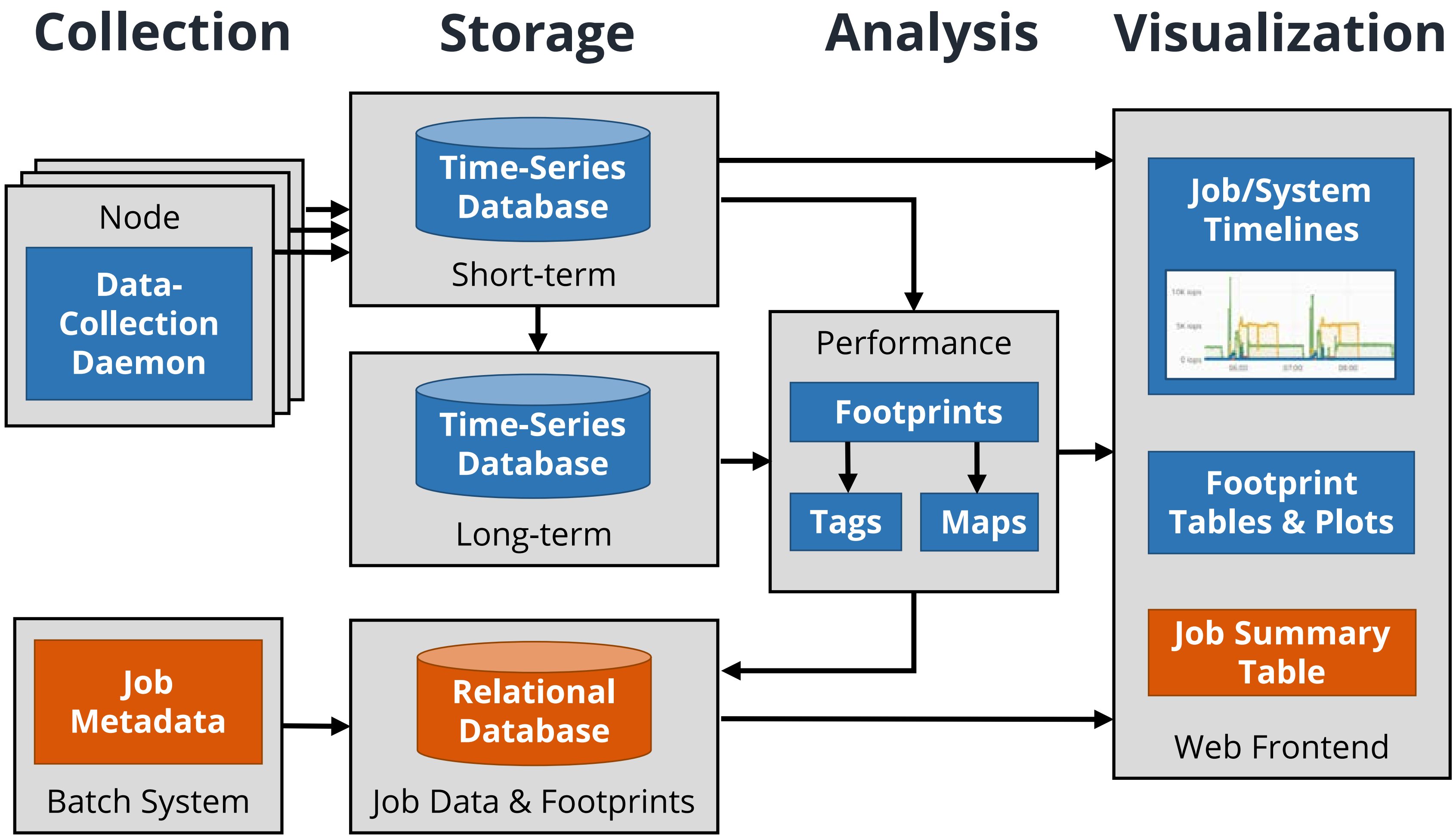
“Pikas prefer rocky slopes and graze on a range of plants, mostly grasses, flowers and young stems. In the autumn, they pull hay, soft twigs and other stores of food into their burrows to eat during the long cold winter.”

Source: Walters, Martin (2005).
Encyclopedia of animals.
Parragon. p. 203. ISBN 978-1-40545-669-2.

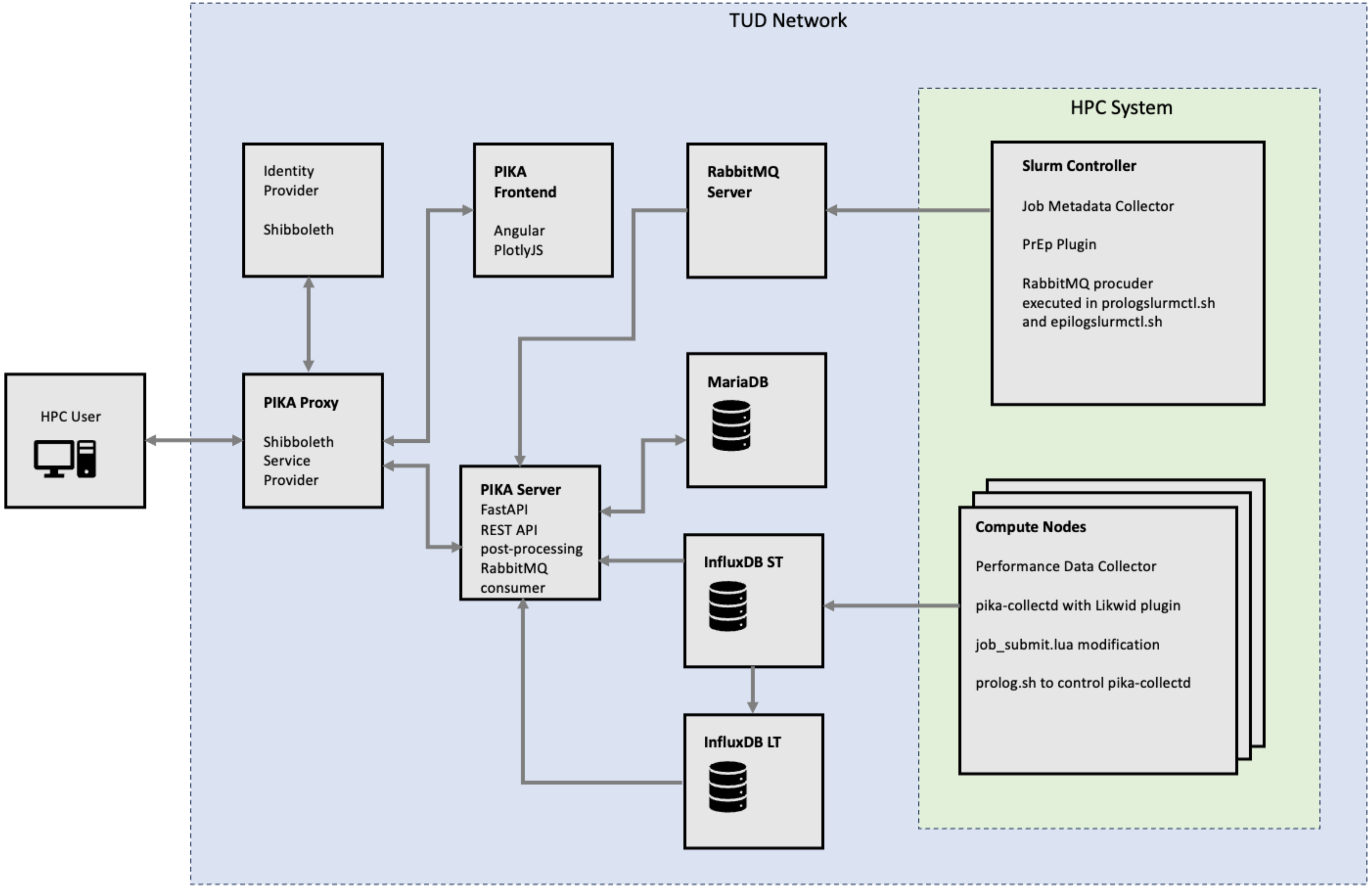
Outline

- PIKA infrastructure
- PIKA web frontend
- Hands-on exercise
- Discussions

PIKA Architecture Overview



PIKA Architecture at TUD



PIKA Metadata Collection

Slurm PrEp Plugin to capture job metadata:

- Unique job identifier, ArrayID
- Project, user, job name
- Start and end time, walltime
- Status (running, completed, timeout, failed, OOM, cancelled)
- Requested resources
  - Partition
  - Allocated compute nodes
  - Allocated CPUs on each node
  - Exclusive nodes
  - Main memory
  - GPUs per node

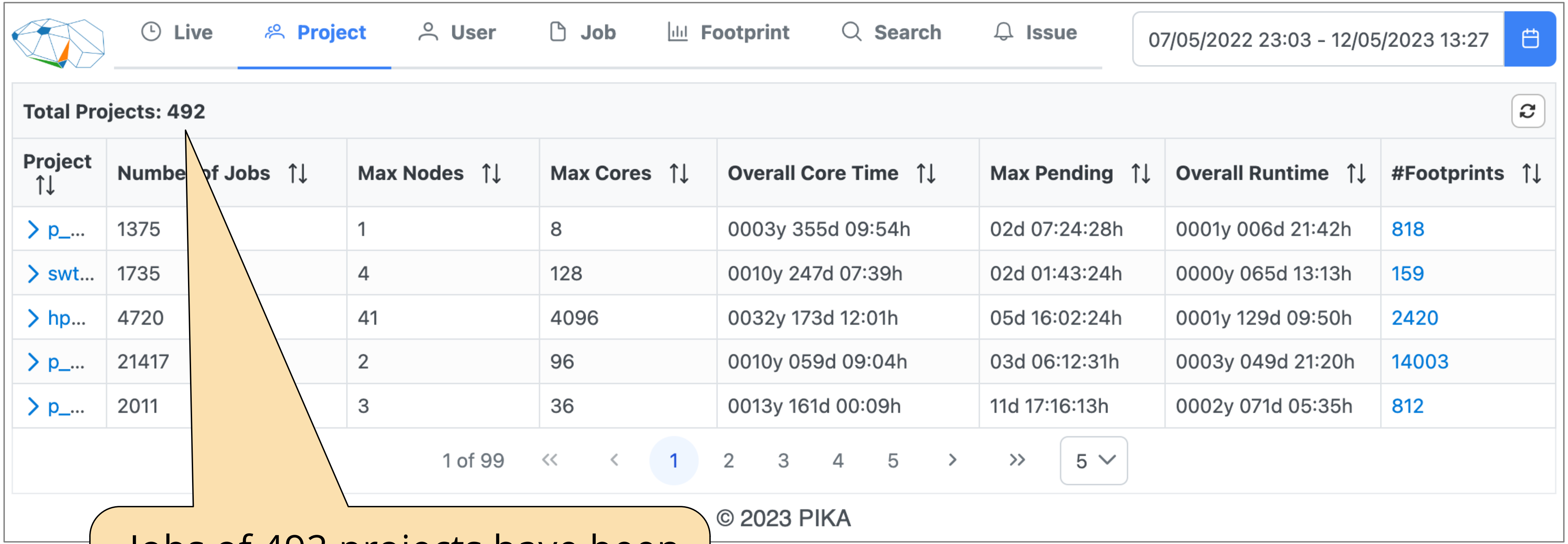
PIKA Runtime Data Collection

Monitored Metrics	Data Source	Hardware Unit
Instructions per Cycle (IPC)	LIKWID	Hardware Thread
FLOPS (SP Normalized)		Hardware Thread
Main Memory Bandwidth		CPU/Socket
Power Consumption		CPU/Socket
CPU Usage	proc & sysfs	Hardware Thread
Main Memory Utilization		Node
Network Bandwidth		Node
File I/O Bandwidth & Metadata	Local disk, Filesystems (Lustre)	Disk, Lustre Instance
GPU Usage	NVML	GPU
GPU Memory Utilization		
GPU Power Consumption		
GPU Temperature		

Collection daemon **collectd**

- One collector/plugin for each metric source
- CPU counters are collected with LIKWID
- Hardware thread metrics are summarized to the physical CPU core

PIKA Job Visualization – Tables



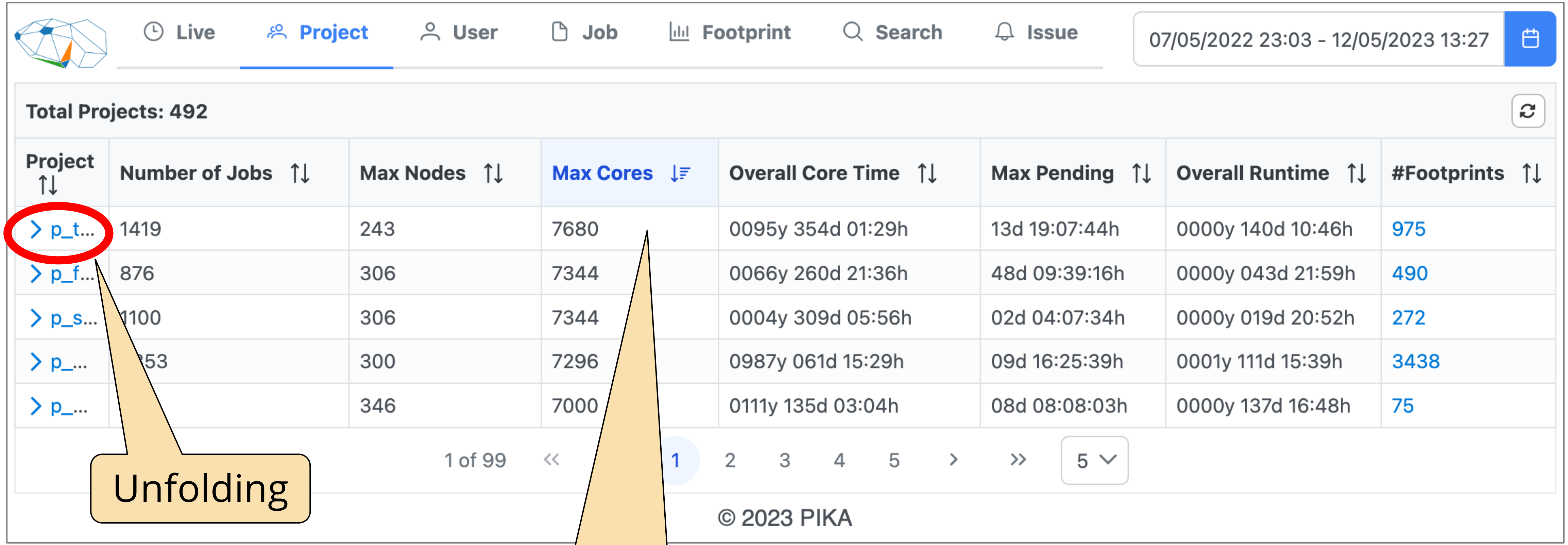
Total Projects: 492

Project ↑↓	Numbe of Jobs ↑↓	Max Nodes ↑↓	Max Cores ↑↓	Overall Core Time ↑↓	Max Pending ↑↓	Overall Runtime ↑↓	#Footprints ↑↓
> p_...	1375	1	8	0003y 355d 09:54h	02d 07:24:28h	0001y 006d 21:42h	818
> swt...	1735	4	128	0010y 247d 07:39h	02d 01:43:24h	0000y 065d 13:13h	159
> hp...	4720	41	4096	0032y 173d 12:01h	05d 16:02:24h	0001y 129d 09:50h	2420
> p_...	21417	2	96	0010y 059d 09:04h	03d 06:12:31h	0003y 049d 21:20h	14003
> p_...	2011	3	36	0013y 161d 00:09h	11d 17:16:13h	0002y 071d 05:35h	812

© 2023 PIKA

Jobs of 492 projects have been recorded for the selected time interval (top right).

PIKA Job Visualization – Tables



Total Projects: 492

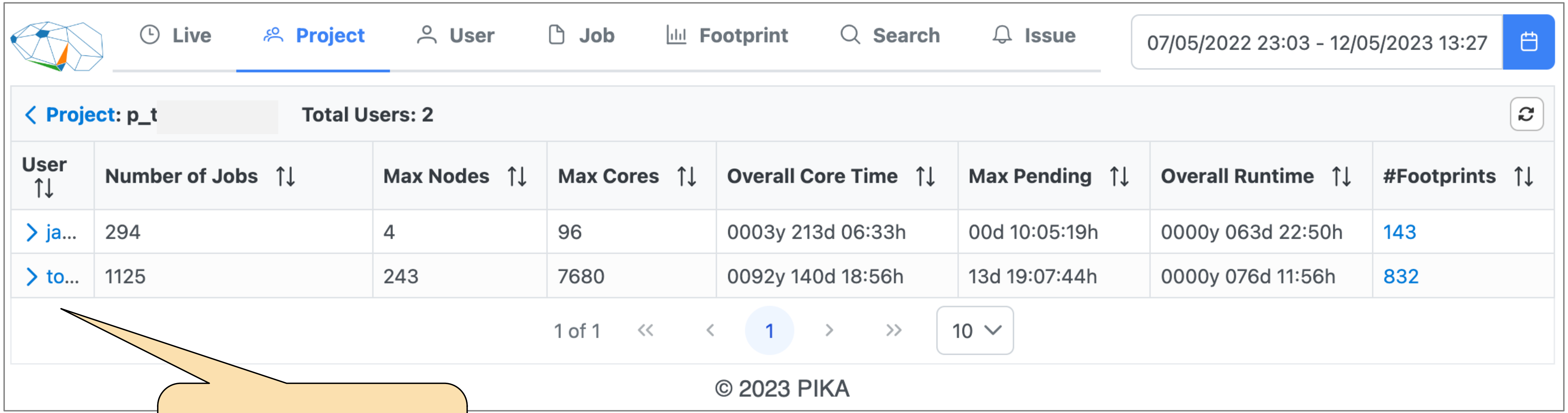
Project ↑↓	Number of Jobs ↑↓	Max Nodes ↑↓	Max Cores ↓	Overall Core Time ↑↓	Max Pending ↑↓	Overall Runtime ↑↓	#Footprints ↑↓
› p_t...	1419	243	7680	0095y 354d 01:29h	13d 19:07:44h	0000y 140d 10:46h	975
› p_f...	876	306	7344	0066y 260d 21:36h	48d 09:39:16h	0000y 043d 21:59h	490
› p_s...	1100	306	7344	0004y 309d 05:56h	02d 04:07:34h	0000y 019d 20:52h	272
› p_...	[illegible]353	300	7296	0987y 061d 15:29h	09d 16:25:39h	0001y 111d 15:39h	3438
› p_...	[illegible]	346	7000	0111y 135d 03:04h	08d 08:08:03h	0000y 137d 16:48h	75

© 2023 PIKA

Unfolding

Get project with the highest number of cores.

PIKA Job Visualization – Tables



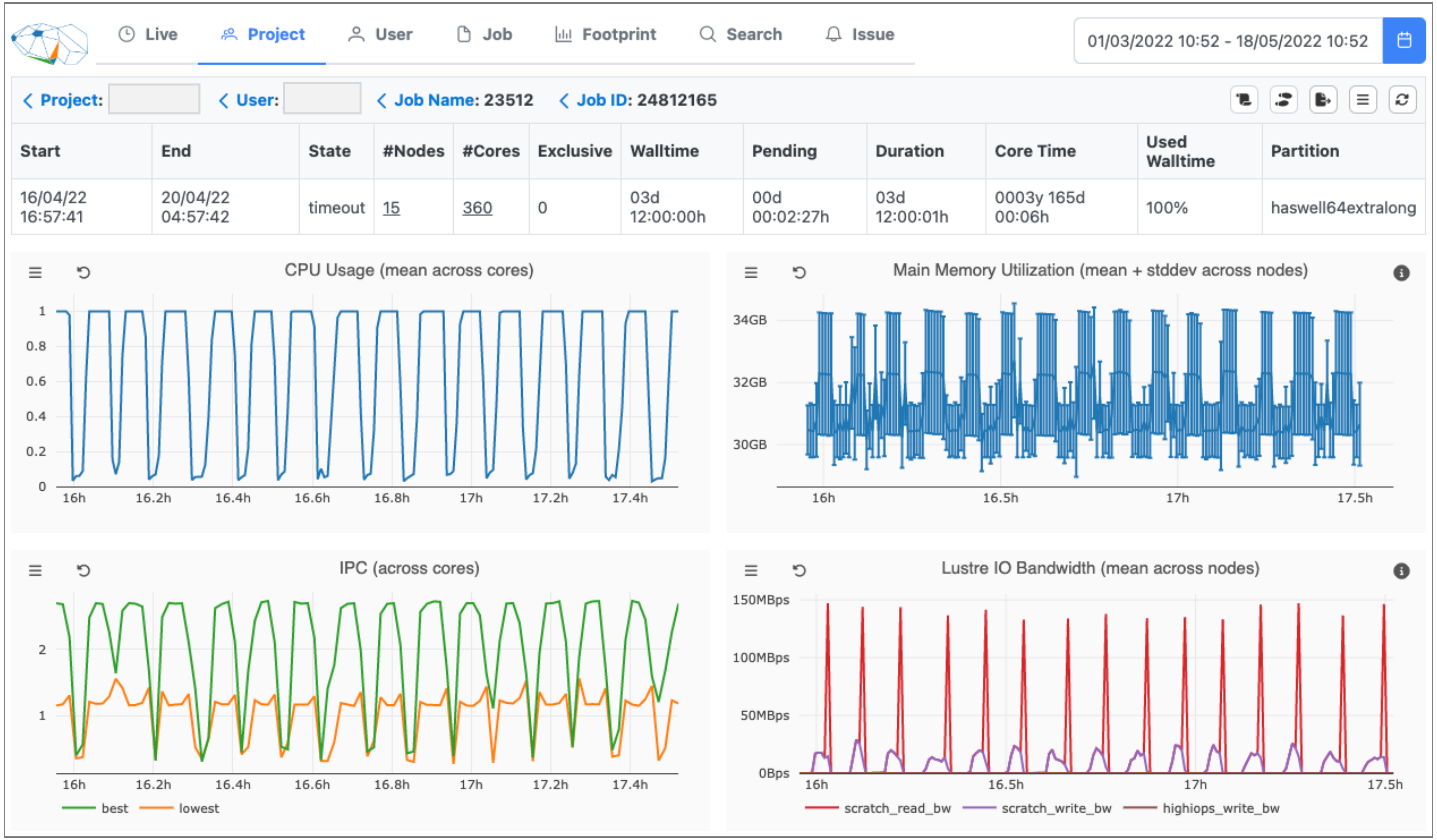
Project: p_t[redacted] **Total Users: 2**

User	Number of Jobs	Max Nodes	Max Cores	Overall Core Time	Max Pending	Overall Runtime	#Footprints
ja...	294	4	96	0003y 213d 06:33h	00d 10:05:19h	0000y 063d 22:50h	143
to...	1125	243	7680	0092y 140d 18:56h	13d 19:07:44h	0000y 076d 11:56h	832

© 2023 PIKA

Project “p_t” has two users.

PIKA Job Visualization – Metadata & Timelines



< Project: < User: < Job Name: 23512 < Job ID: 24812165

Start	End	State	#Nodes	#Cores	Exclusive	Walltime	Pending	Duration	Core Time	Used Walltime	Partition
16/04/22 16:57:41	20/04/22 04:57:42	timeout	15	360	0	03d 12:00:00h	00d 00:02:27h	03d 12:00:01h	0003y 165d 00:06h	100%	haswell64extralong

CPU Usage (mean across cores)

Main Memory Utilization (mean + stddev across nodes)

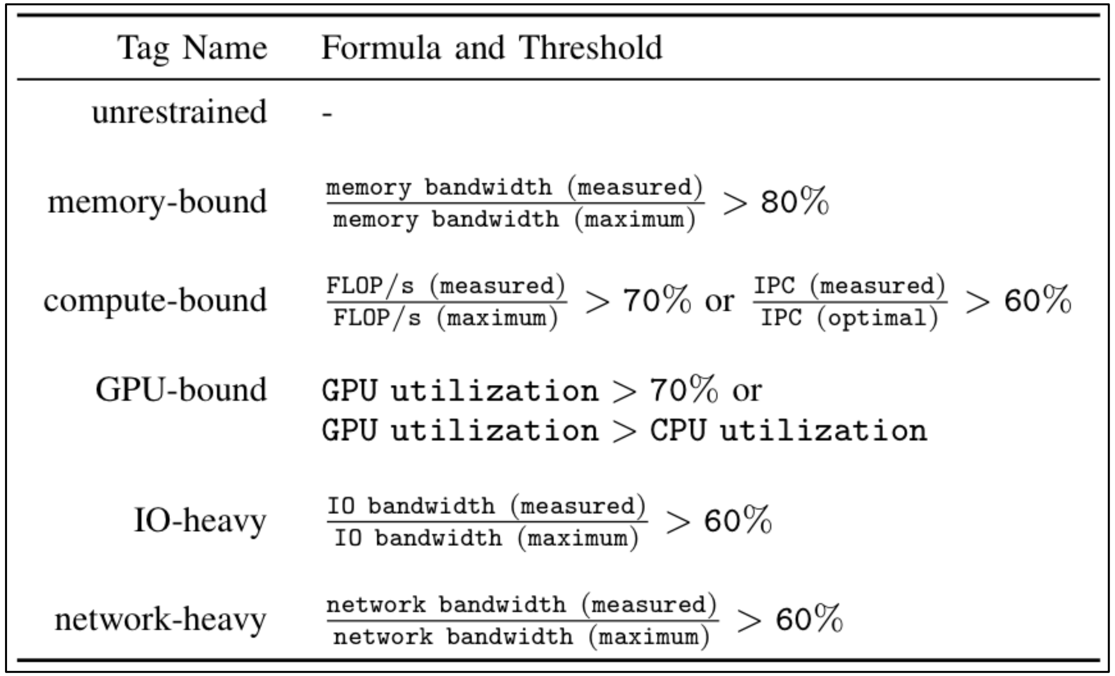
IPC (across cores)

Lustre IO Bandwidth (mean across nodes)

PIKA Post Processing

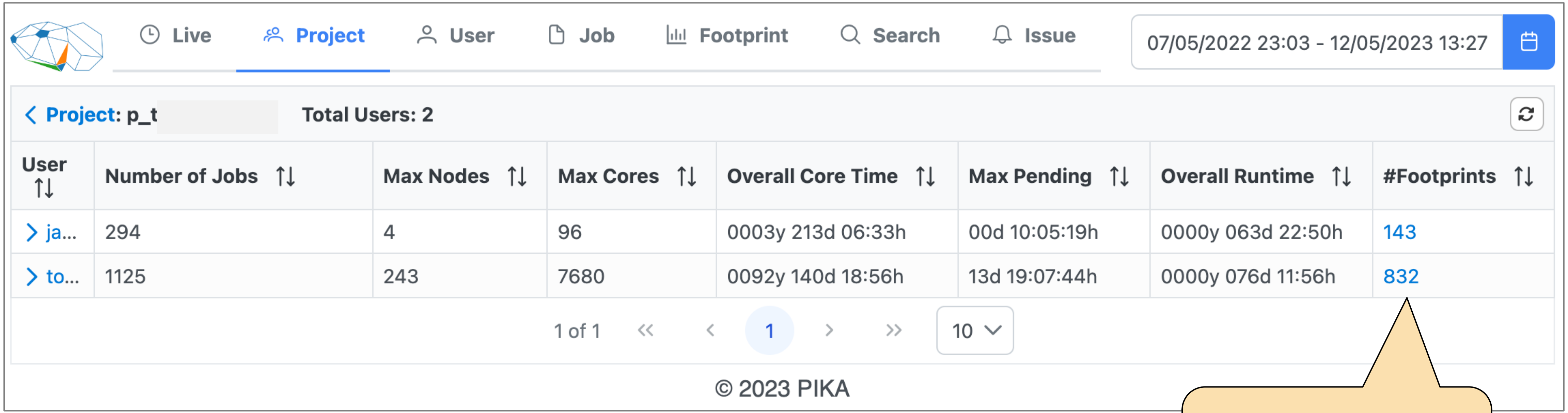
Job characterization via tagging

- **Footprints** based on summarized runtime data
  - **Average** (CPU and GPU usage, IPC, FLOPS, main memory bandwidth, CPU and GPU power, InfiniBand traffic)
  - **Total** (file IO read/write)
  - **Maximum** (host and GPU memory usage)
- Job tags based on formulas and thresholds



Tag Name	Formula and Threshold
unrestrained	-
memory-bound	$\frac{\texttt{memory bandwidth (measured)}}{\texttt{memory bandwidth (maximum)}} > 80\%$
compute-bound	$\frac{\texttt{FLOP/s (measured)}}{\texttt{FLOP/s (maximum)}} > 70\%$ or $\frac{\texttt{IPC (measured)}}{\texttt{IPC (optimal)}} > 60\%$
GPU-bound	$\texttt{GPU utilization} > 70\%$ or $\texttt{GPU utilization} > \texttt{CPU utilization}$
IO-heavy	$\frac{\texttt{IO bandwidth (measured)}}{\texttt{IO bandwidth (maximum)}} > 60\%$
network-heavy	$\frac{\texttt{network bandwidth (measured)}}{\texttt{network bandwidth (maximum)}} > 60\%$

PIKA Post Processing



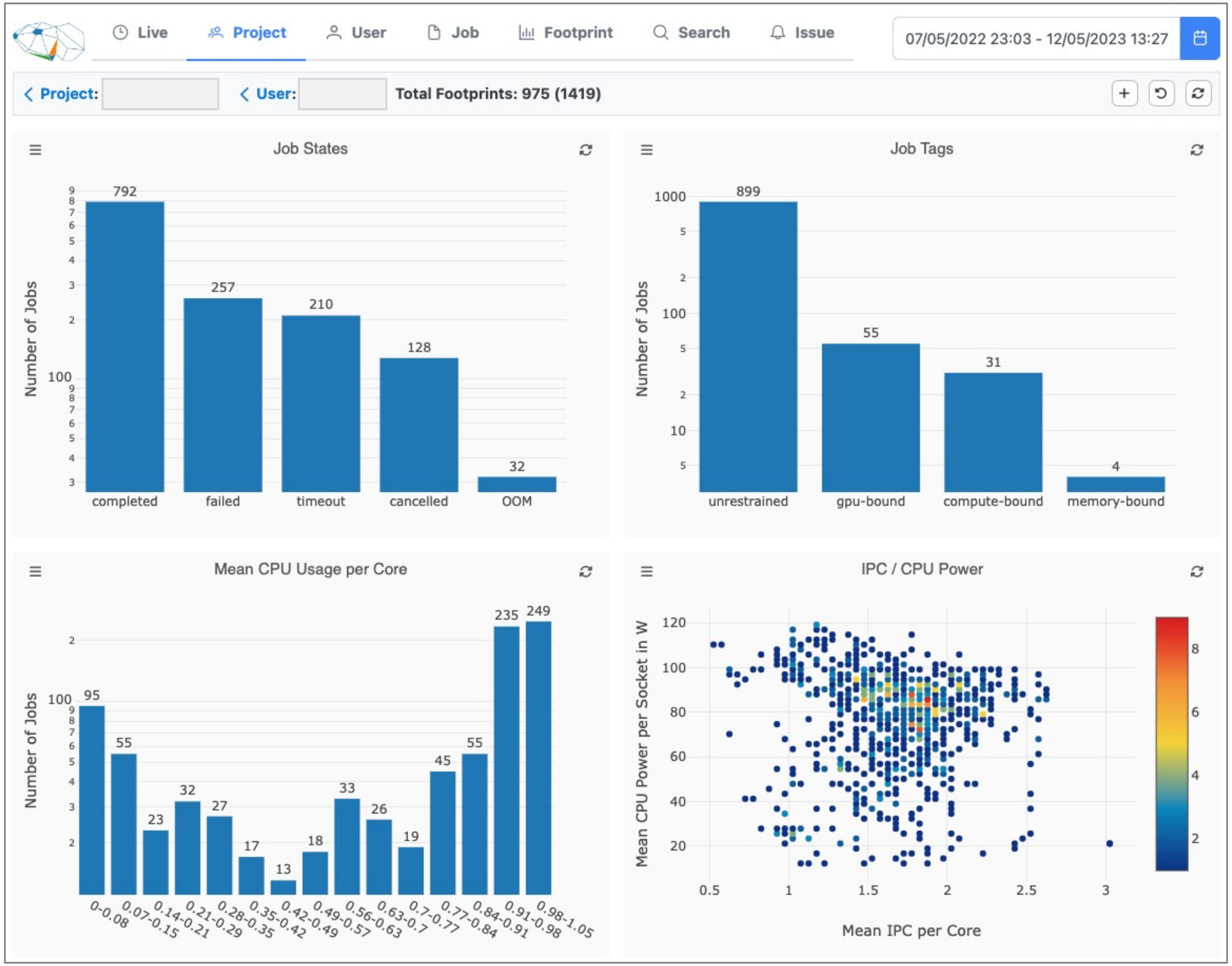
Project: p_t[redacted] Total Users: 2

User	Number of Jobs	Max Nodes	Max Cores	Overall Core Time	Max Pending	Overall Runtime	#Footprints
ja...	294	4	96	0003y 213d 06:33h	00d 10:05:19h	0000y 063d 22:50h	143
to...	1125	243	7680	0092y 140d 18:56h	13d 19:07:44h	0000y 076d 11:56h	832

© 2023 PIKA

832 user jobs are tagged.

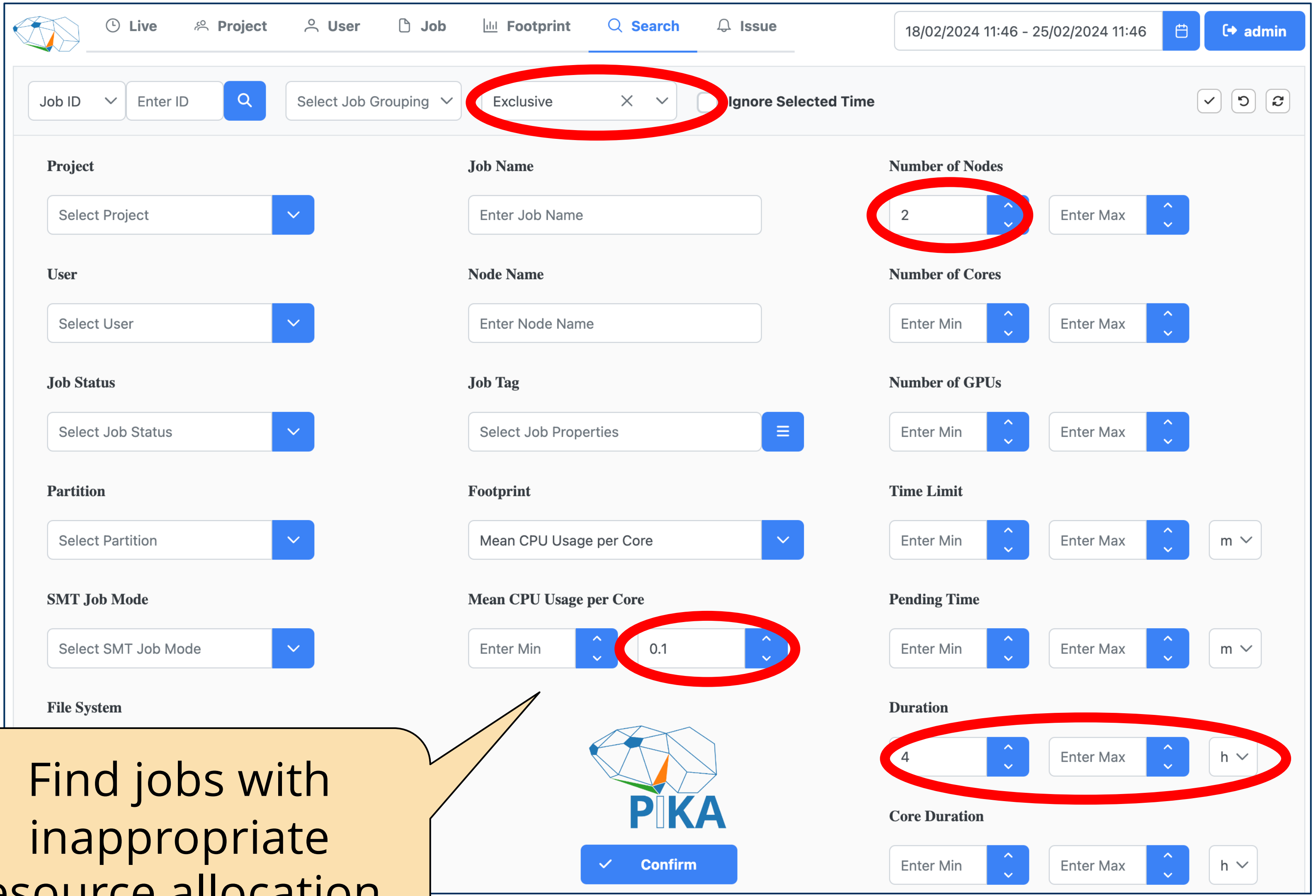
PIKA Job Visualization – Footprints



07/05/2022 23:03 - 12/05/2023 13:27

Project: User: Total Footprints: 975 (1419)

PIKA – Search for Suboptimal Jobs



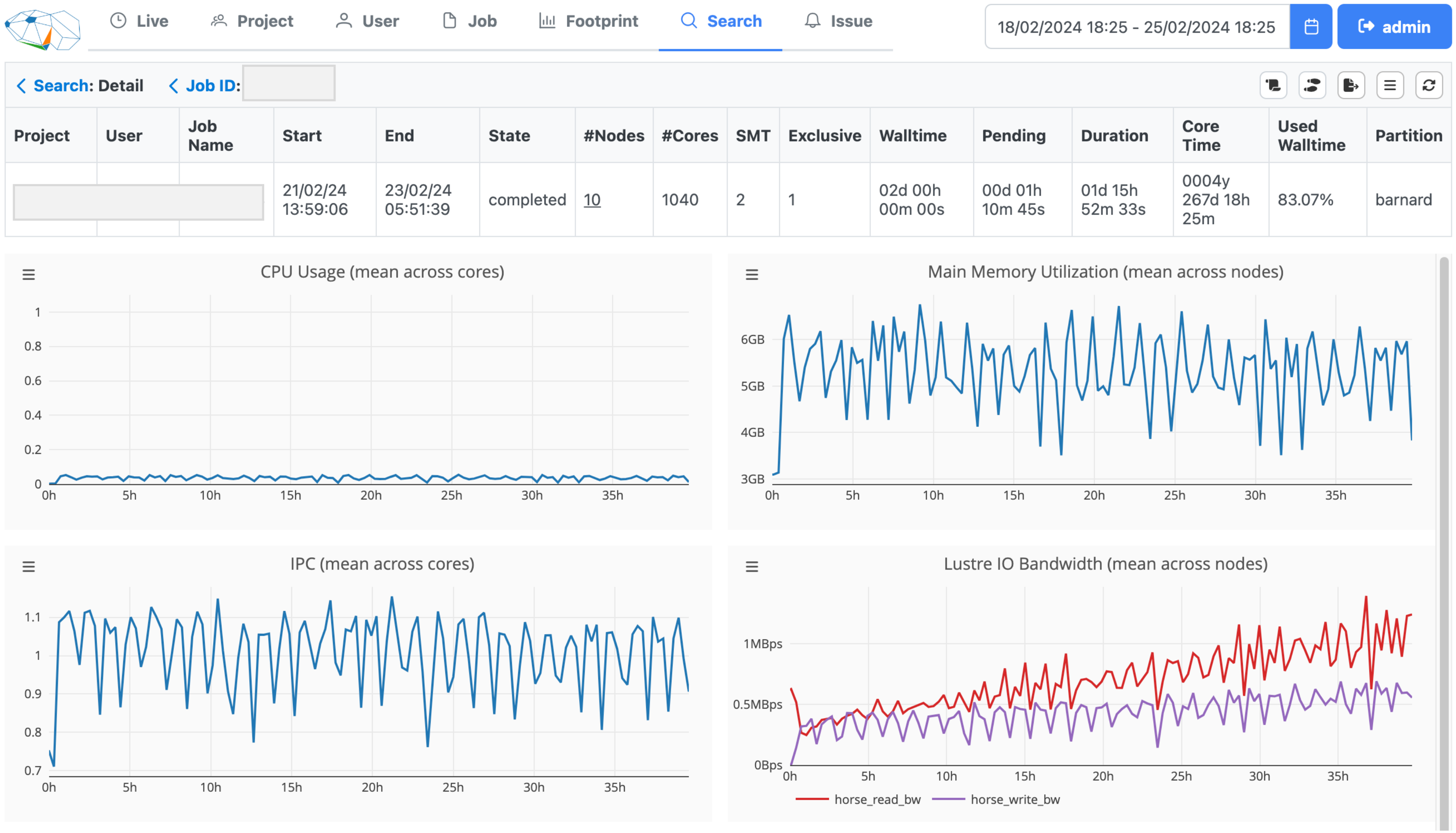
18/02/2024 11:46 - 25/02/2024 11:46

Job ID | Enter ID | Select Job Grouping | Exclusive | □ Ignore Selected Time

Project Select Project	**Job Name** Enter Job Name	**Number of Nodes** 2 – Enter Max
User Select User	**Node Name** Enter Node Name	**Number of Cores** Enter Min – Enter Max
Job Status Select Job Status	**Job Tag** Select Job Properties	**Number of GPUs** Enter Min – Enter Max
Partition Select Partition	**Footprint** Mean CPU Usage per Core	**Time Limit** Enter Min – Enter Max m
SMT Job Mode Select SMT Job Mode	**Mean CPU Usage per Core** Enter Min – 0.1	**Pending Time** Enter Min – Enter Max m
File System		**Duration** 4 – Enter Max h
		Core Duration Enter Min – Enter Max h

Find jobs with inappropriate resource allocation.

PIKA – Search for Suboptimal Jobs



18/02/2024 18:25 - 25/02/2024 18:25

Search: Detail   Job ID: [redacted]

Project	User	Job Name	Start	End	State	#Nodes	#Cores	SMT	Exclusive	Walltime	Pending	Duration	Core Time	Used Walltime	Partition
[redacted]	[redacted]	[redacted]	21/02/24 13:59:06	23/02/24 05:51:39	completed	10	1040	2	1	02d 00h 00m 00s	00d 01h 10m 45s	01d 15h 52m 33s	0004y 267d 18h 25m	83.07%	barnard

PIKA Issue Analysis

Automatic detection of job performance issues on eligible jobs

- **Prerequisite:**
  - Duration >= 1 hour
  - Number of physical cores > 1
  - Slurm Status: completed, out of memory, timeout
  - Metric timeline vectors*: CPU/GPU load, memory usage, I/O bandwidths and I/O metadata operations
- **Heuristics** to detect inefficient jobs
- **Criteria for efficient usage**
- Shortest possible runtimes (compared to similar jobs)
- High utilization of the hardware
- Even distribution of computational workloads across processing units

* Sampled every 30 seconds

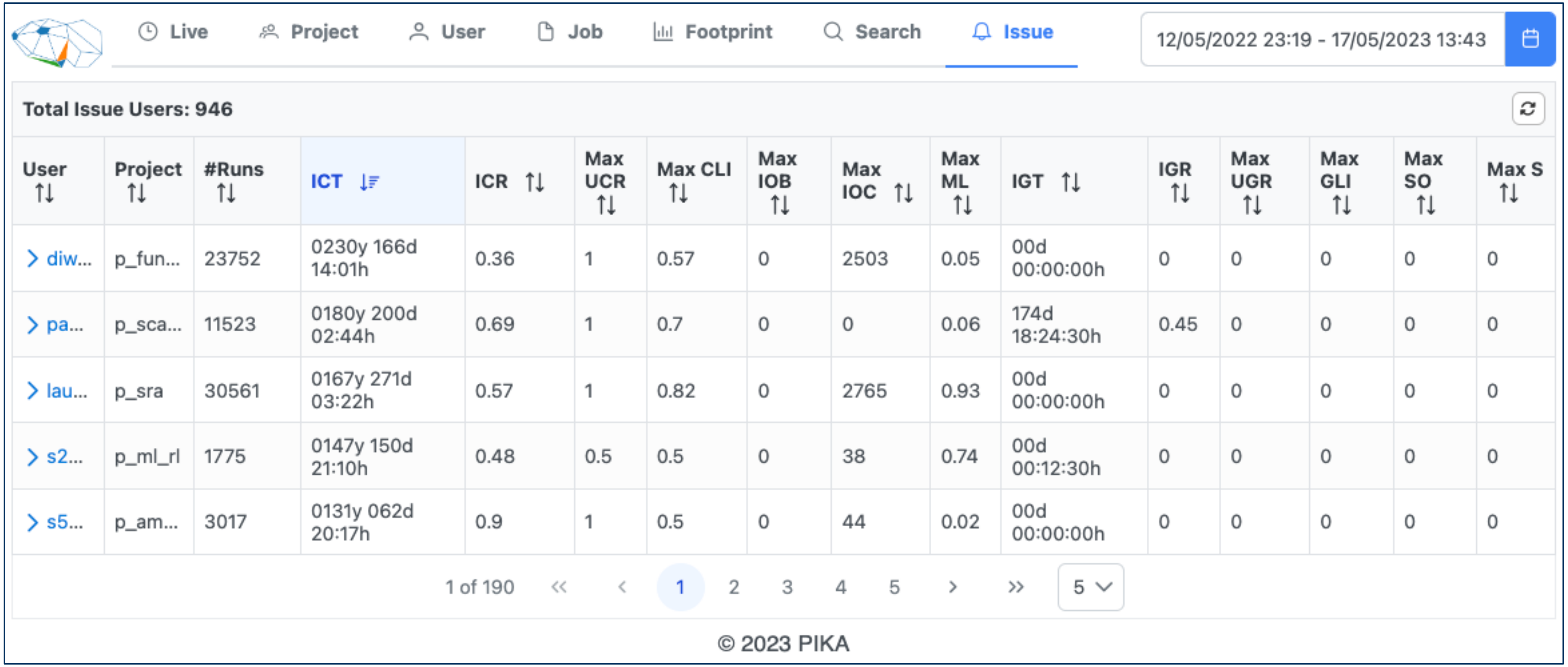
PIKA Issue Analysis – Summarized User View

Possible performance issues with the inefficient HPC jobs of a user

Performance Issue	Description
Idle CPU/GPU Time (**ICT/IGT**)	Summed time intervals of all CPUs/GPUs across all jobs in which the load was close to zero.
Idle CPU/GPU Ratio (**ICR/IGR**)	Quotient of “Idle CPU/GPU Time” and “Total CPU/GPU Time” across all jobs.
Maximum Unused CPU/GPU Ratio (**Max UCR/UGR**)	Maximum ratio of “unused” to “used” CPUs/GPUs across all jobs.
Maximum CPU/GPU Load Imbalance (**Max CLI/GLI**)	Maximum of the average standard deviation of CPU/GPU load across all jobs.
Maximum I/O Congestion (**Max IOC**)	Maximum rate of metadata operations at a measuring point across all jobs. The attribution per job starts with 40 operations.
Maximum I/O Blocking Phases (**Max IOB**)	Maximum periodic number of phases with an inverse correlation between CPU load and I/O metrics across all jobs. The attribution per job starts with 10 periodic phases.
Maximum Synchronous Offloading (**Max SO**)	Maximum periodic number of phases with an inverse correlation between CPU and GPU load across all jobs. The attribution per job starts with 10 periodic phases.
Maximum Memory Leak (**Max ML**)	Maximum of the linear increase of memory usage over time across all jobs.

PIKA Issue Analysis – Issue Table

User jobs are sorted by idle CPU time



12/05/2022 23:19 - 17/05/2023 13:43

Total Issue Users: 946

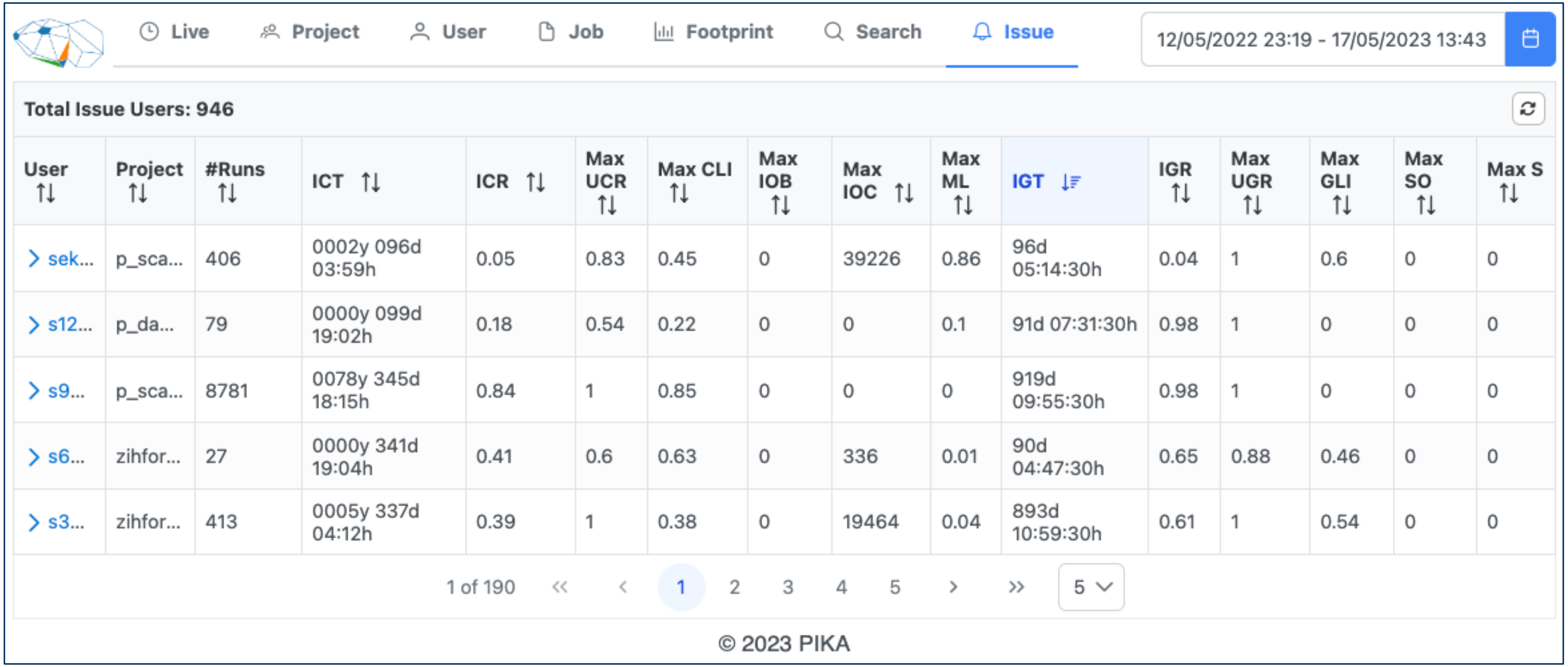
User	Project	#Runs	ICT	ICR	Max UCR	Max CLI	Max IOB	Max IOC	Max ML	IGT	IGR	Max UGR	Max GLI	Max SO	Max S
› diw...	p_fun...	23752	0230y 166d 14:01h	0.36	1	0.57	0	2503	0.05	00d 00:00:00h	0	0	0	0	0
› pa...	p_sca...	11523	0180y 200d 02:44h	0.69	1	0.7	0	0	0.06	174d 18:24:30h	0.45	0	0	0	0
› lau...	p_sra	30561	0167y 271d 03:22h	0.57	1	0.82	0	2765	0.93	00d 00:00:00h	0	0	0	0	0
› s2...	p_ml_rl	1775	0147y 150d 21:10h	0.48	0.5	0.5	0	38	0.74	00d 00:12:30h	0	0	0	0	0
› s5...	p_am...	3017	0131y 062d 20:17h	0.9	1	0.5	0	44	0.02	00d 00:00:00h	0	0	0	0	0

1 of 190 « ‹ 1 2 3 4 5 › » 5

© 2023 PIKA

PIKA Issue Analysis – Issue Table

User jobs are sorted by idle GPU time



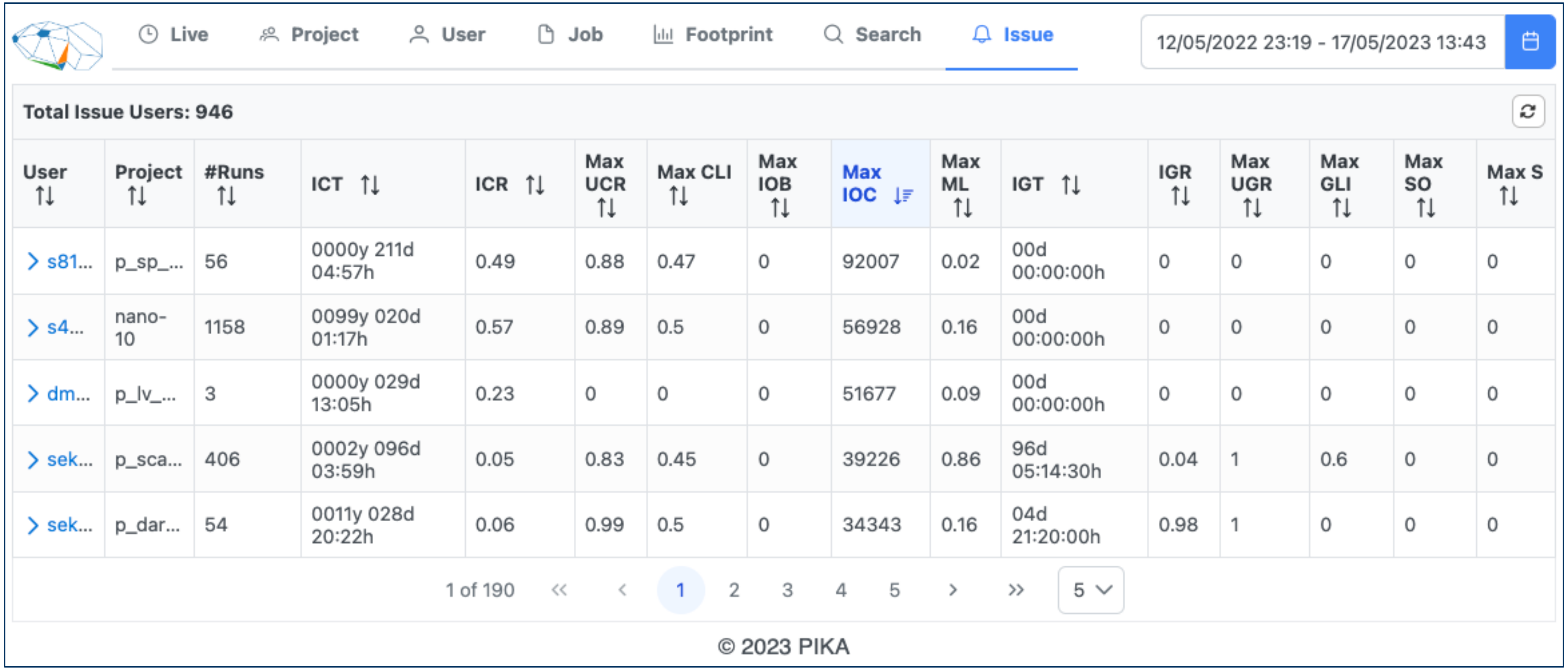
Total Issue Users: 946

User	Project	#Runs	ICT	ICR	Max UCR	Max CLI	Max IOB	Max IOC	Max ML	IGT	IGR	Max UGR	Max GLI	Max SO	Max S
› sek...	p_sca...	406	0002y 096d 03:59h	0.05	0.83	0.45	0	39226	0.86	96d 05:14:30h	0.04	1	0.6	0	0
› s12...	p_da...	79	0000y 099d 19:02h	0.18	0.54	0.22	0	0	0.1	91d 07:31:30h	0.98	1	0	0	0
› s9...	p_sca...	8781	0078y 345d 18:15h	0.84	1	0.85	0	0	0	919d 09:55:30h	0.98	1	0	0	0
› s6...	zihfor...	27	0000y 341d 19:04h	0.41	0.6	0.63	0	336	0.01	90d 04:47:30h	0.65	0.88	0.46	0	0
› s3...	zihfor...	413	0005y 337d 04:12h	0.39	1	0.38	0	19464	0.04	893d 10:59:30h	0.61	1	0.54	0	0

© 2023 PIKA

PIKA Issue Analysis – Issue Table

User jobs are sorted by maximum I/O congestion



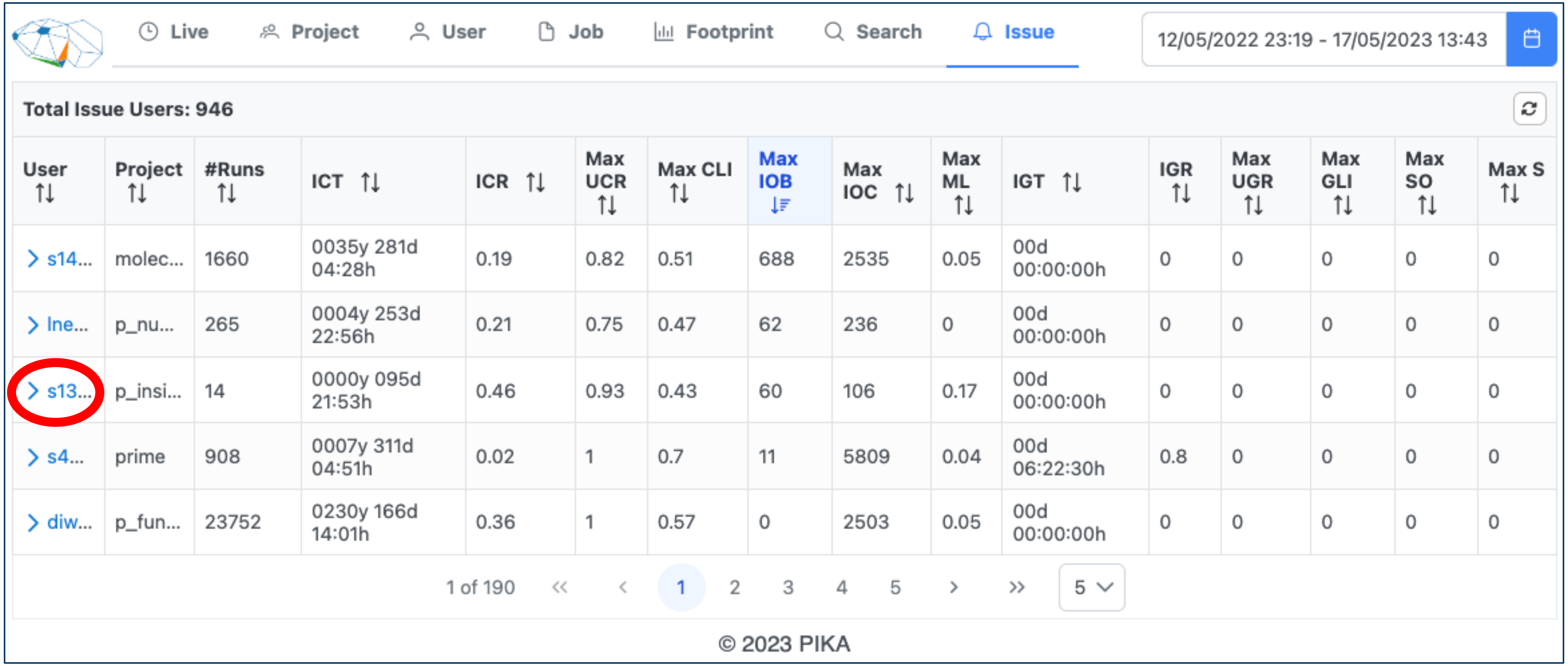
Total Issue Users: 946

User	Project	#Runs	ICT	ICR	Max UCR	Max CLI	Max IOB	Max IOC	Max ML	IGT	IGR	Max UGR	Max GLI	Max SO	Max S
s81...	p_sp_...	56	0000y 211d 04:57h	0.49	0.88	0.47	0	92007	0.02	00d 00:00:00h	0	0	0	0	0
s4...	nano-10	1158	0099y 020d 01:17h	0.57	0.89	0.5	0	56928	0.16	00d 00:00:00h	0	0	0	0	0
dm...	p_lv_...	3	0000y 029d 13:05h	0.23	0	0	0	51677	0.09	00d 00:00:00h	0	0	0	0	0
sek...	p_sca...	406	0002y 096d 03:59h	0.05	0.83	0.45	0	39226	0.86	96d 05:14:30h	0.04	1	0.6	0	0
sek...	p_dar...	54	0011y 028d 20:22h	0.06	0.99	0.5	0	34343	0.16	04d 21:20:00h	0.98	1	0	0	0

© 2023 PIKA

PIKA Issue Analysis – Issue Table

User jobs are sorted by maximum I/O blocking



12/05/2022 23:19 - 17/05/2023 13:43

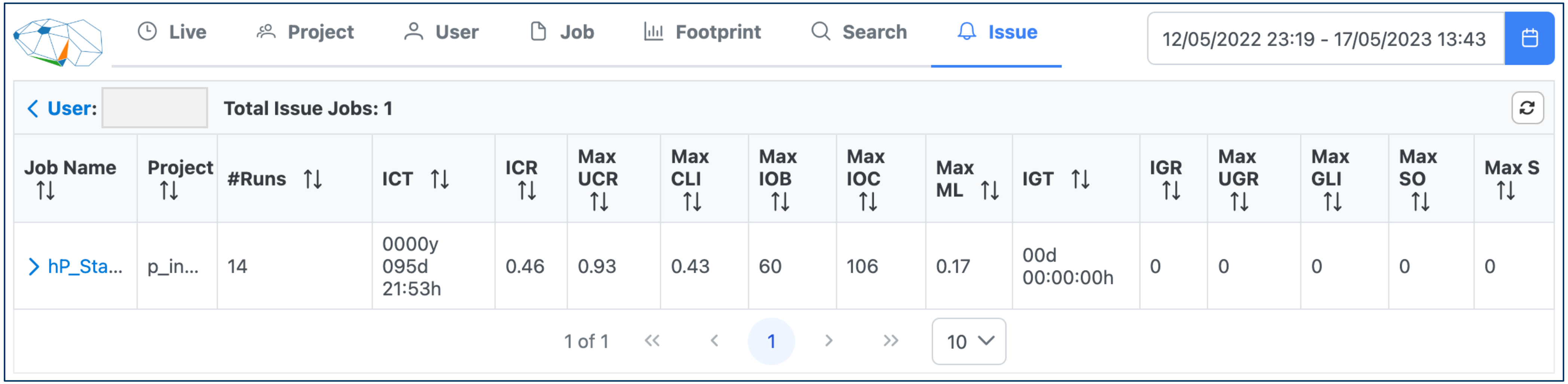
Total Issue Users: 946

User	Project	#Runs	ICT	ICR	Max UCR	Max CLI	Max IOB	Max IOC	Max ML	IGT	IGR	Max UGR	Max GLI	Max SO	Max S
s14...	molec...	1660	0035y 281d 04:28h	0.19	0.82	0.51	688	2535	0.05	00d 00:00:00h	0	0	0	0	0
lne...	p_nu...	265	0004y 253d 22:56h	0.21	0.75	0.47	62	236	0	00d 00:00:00h	0	0	0	0	0
s13...	p_insi...	14	0000y 095d 21:53h	0.46	0.93	0.43	60	106	0.17	00d 00:00:00h	0	0	0	0	0
s4...	prime	908	0007y 311d 04:51h	0.02	1	0.7	11	5809	0.04	00d 06:22:30h	0.8	0	0	0	0
diw...	p_fun...	23752	0230y 166d 14:01h	0.36	1	0.57	0	2503	0.05	00d 00:00:00h	0	0	0	0	0

© 2023 PIKA

PIKA Issue Analysis – Issue Table

User jobs with I/O blocking issues



12/05/2022 23:19 - 17/05/2023 13:43

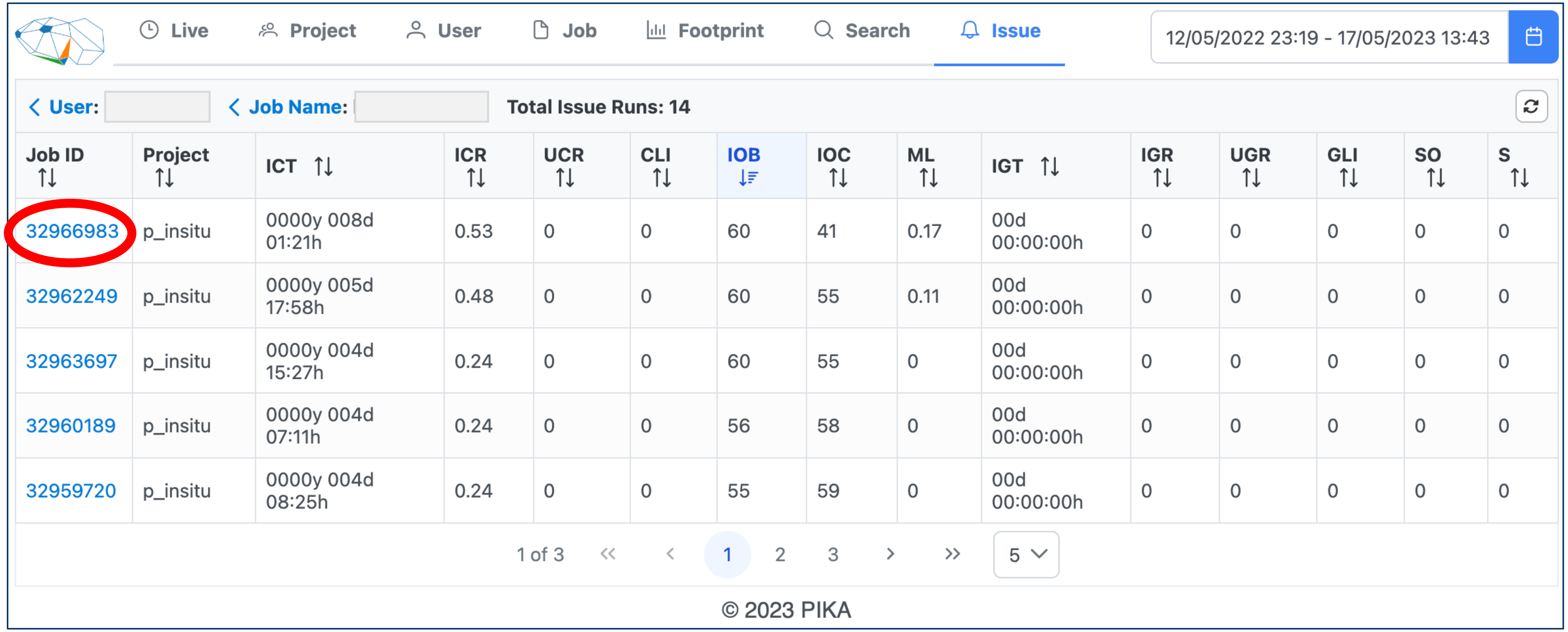
User: Total Issue Jobs: 1

Job Name	Project	#Runs	ICT	ICR	Max UCR	Max CLI	Max IOB	Max IOC	Max ML	IGT	IGR	Max UGR	Max GLI	Max SO	Max S
hP_Sta...	p_in...	14	0000y 095d 21:53h	0.46	0.93	0.43	60	106	0.17	00d 00:00:00h	0	0	0	0	0

1 of 1   1   10

PIKA Issue Analysis – Issue Table

User jobs with I/O blocking issues



12/05/2022 23:19 - 17/05/2023 13:43

User:

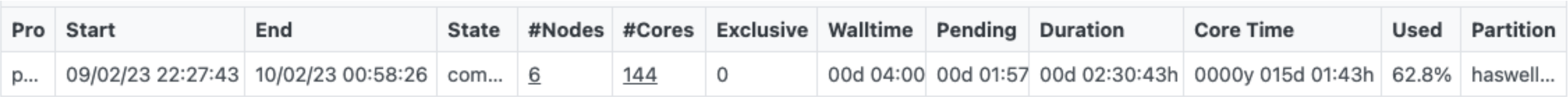
Job Name:

Total Issue Runs: 14

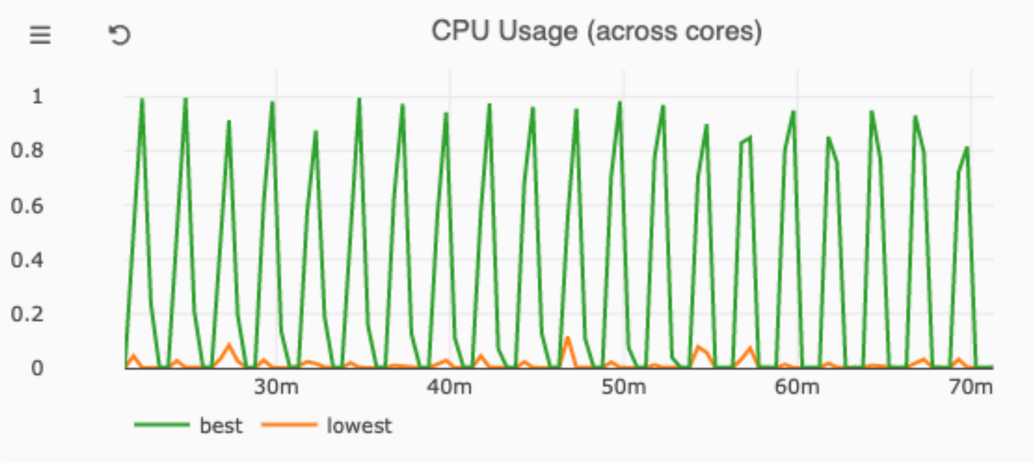
Job ID	Project	ICT	ICR	UCR	CLI	IOB	IOC	ML	IGT	IGR	UGR	GLI	SO	S
32966983	p_insitu	0000y 008d 01:21h	0.53	0	0	60	41	0.17	00d 00:00:00h	0	0	0	0	0
32962249	p_insitu	0000y 005d 17:58h	0.48	0	0	60	55	0.11	00d 00:00:00h	0	0	0	0	0
32963697	p_insitu	0000y 004d 15:27h	0.24	0	0	60	55	0	00d 00:00:00h	0	0	0	0	0
32960189	p_insitu	0000y 004d 07:11h	0.24	0	0	56	58	0	00d 00:00:00h	0	0	0	0	0
32959720	p_insitu	0000y 004d 08:25h	0.24	0	0	55	59	0	00d 00:00:00h	0	0	0	0	0

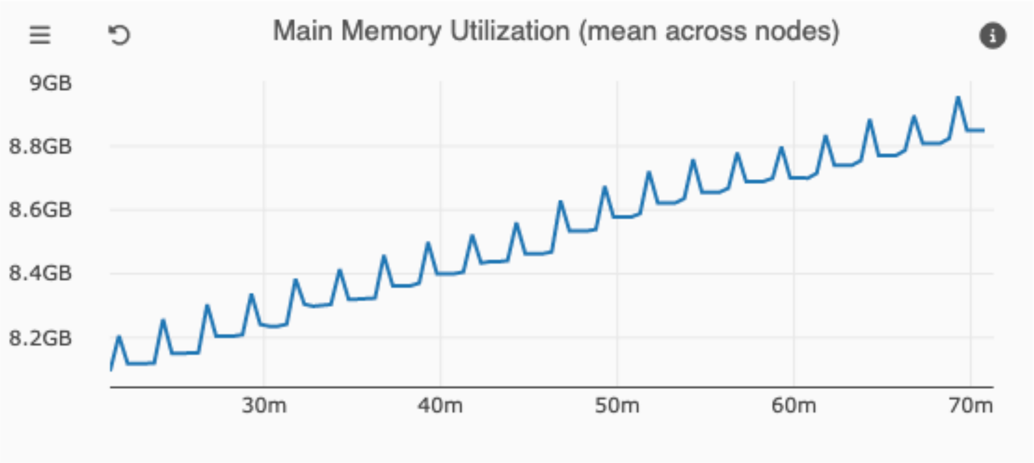
© 2023 PIKA

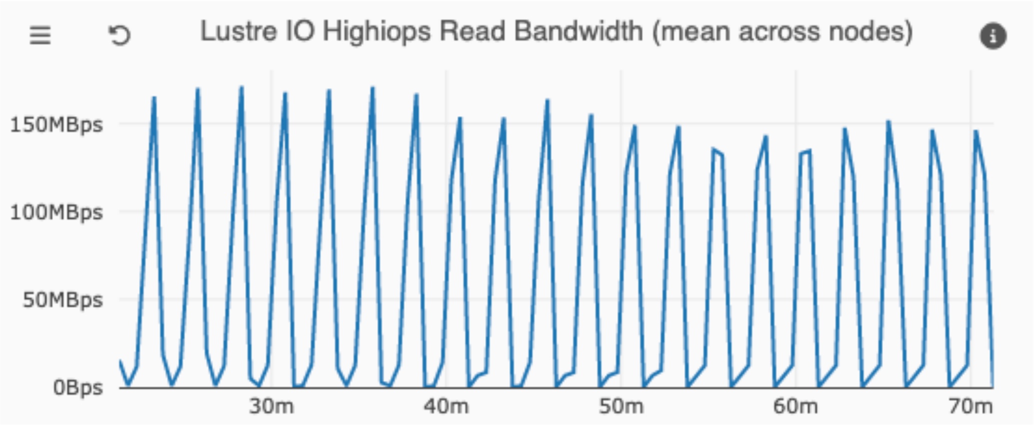
PIKA Issue Analysis – Metadata & Timelines

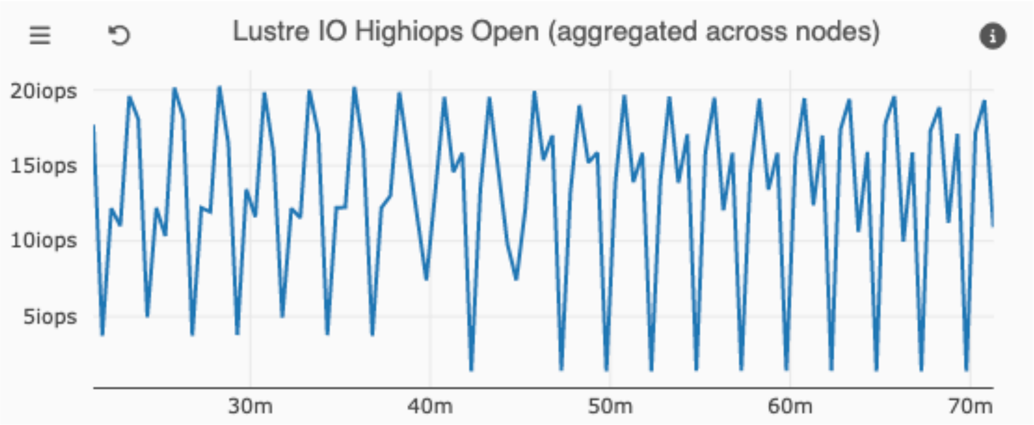


Pro	Start	End	State	#Nodes	#Cores	Exclusive	Walltime	Pending	Duration	Core Time	Used	Partition
p...	09/02/23 22:27:43	10/02/23 00:58:26	com...	6	144	0	00d 04:00	00d 01:57	00d 02:30:43h	0000y 015d 01:43h	62.8%	haswell...









Conclusion

PIKA is a hardware performance monitoring stack in order to identify potentially inefficient jobs.

- Easy access to processed performance data for individual jobs, projects, and users
- Performance overview facilitates quick identification of pathological or suboptimal jobs
- Awareness for performance analysis among new HPC users

R. Dietrich, F. Winkler, A. Knüpfer and W. Nagel, "PIKA: Center-Wide and Job-Aware Cluster Monitoring," 2020 IEEE International Conference on Cluster Computing (CLUSTER), Kobe, Japan, 2020, pp. 424-432.

https://gitlab.hrz.tu-chemnitz.de/pika

Hands-on Exercise

PIKA Hands-on (1)

Open the PIKA web interface and log in with the PIKA demo user:

https://pika.zih.tu-dresden.de

User / Password: TBA

PIKA Hands-on (2)

Connect to Barnard cluster via ssh:

```
% cp ${VIHPS_ROOT}/pika/run_nhr_pika_example.sh .

% sbatch run_nhr_pika_example.sh
  Submitted batch job 19349695
```

Open the PIKA web interface and log in with your own login credentials:

https://pika.zih.tu-dresden.de

- Click on the Live tab
- Update the timelines by clicking on the refresh button (data is updated about every minute)
- Examine the benchmark output of triad_avx and check whether FLOPS and Memory Bandwidth are displayed correctly in PIKA
- Do the IO bandwidths match the instructions in the job script?