Frank Winkler (frank.winkler@tu-dresden.de)

PIKA: Continuous Job Performance Monitoring

NHR Monitoring Workshop 2022
07/18/2022 to 07/19/2022, Erlangen

Center for Information Services & High Performance Computing

Motivation

— Performance is crucial for HPC
— Therefore measurement, analysis and validation also important
— Numerous established tools available
— Problem solved?

— … for those who are aware of the problem
— Active preparation needed
— Not continuous for all jobs of all users

HPCToolkit

VAMPIR

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

MAQAO

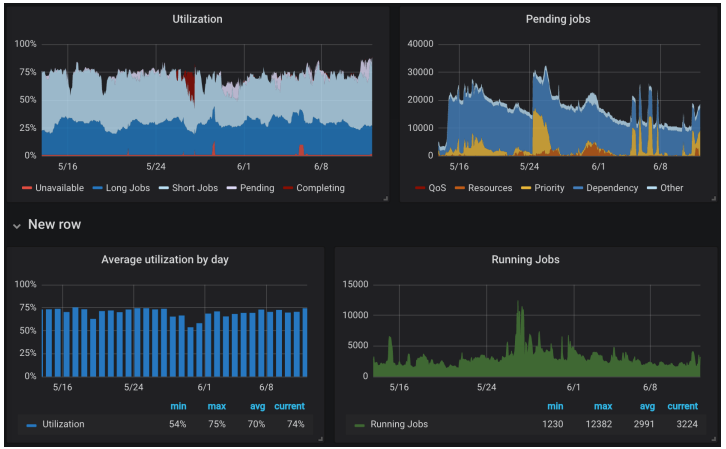
LIKWID TOOLS

scalasca

Score-P
Scalable performance measurement infrastructure for parallel codes

Motivation

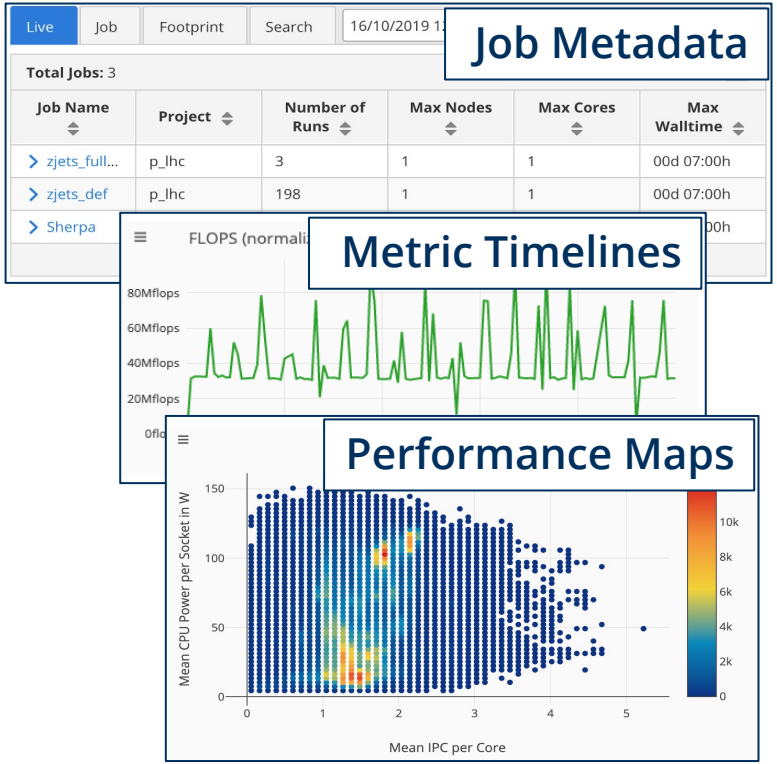
- Several commercial, community and site-specific monitoring solutions already exist
- Most systems provide a system view
- A few are job-aware and track performance data like Flop/s (only applies for site-specific solutions)
- Existing site-specific monitoring systems are complex and difficult to adapt to other sites



Visualization of cluster utilization in Grafana

PIKA: Continuous Job Performance Monitoring

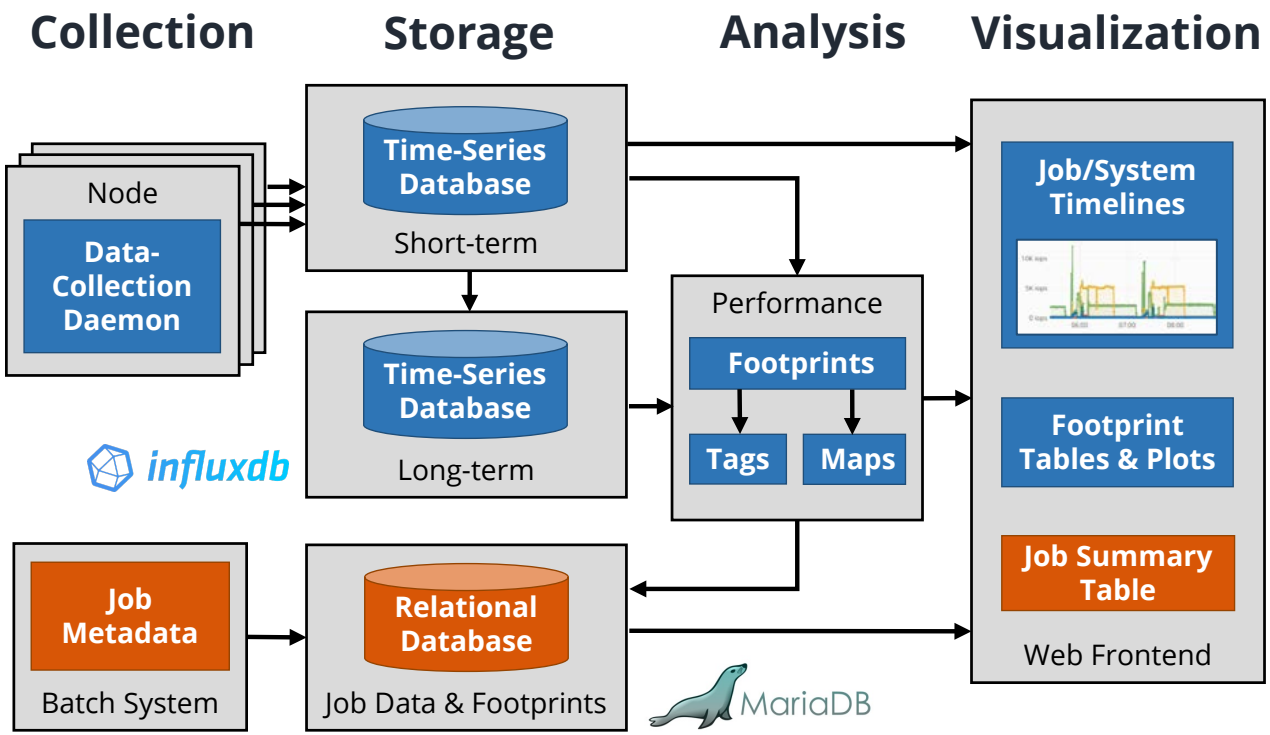
— Non-intrusive **data acquisition** on all cluster nodes
— Continuous **data collection**
— Web frontend for live and post-mortem **visualization**
— Detection of pathological jobs
— Automatic **job analysis and classification**
— Long-term **data storage**



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

PIKA

PIKA Architecture Overview



PIKA Metadata Collection

Slurm PrEp Plugin to capture the following 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
  - GPUs per node
  - Job script

PIKA Metadata Storage

Job metadata is stored in MariaDB (relational database) on a separate service node.

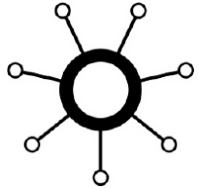
- Several indexes to improve query performance
- Statistics (September 2018 until today)
  - About 33 million jobs
  - About 12 GB of disk memory (six indices)
  - Job-specific query takes about 1ms

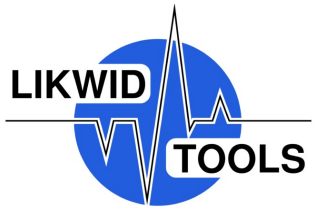
PIKA Runtime Data Collection

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

Collection daemon **collectd**

- One collector/plugin for each metric source
- CPU counters are collected with LIKWID



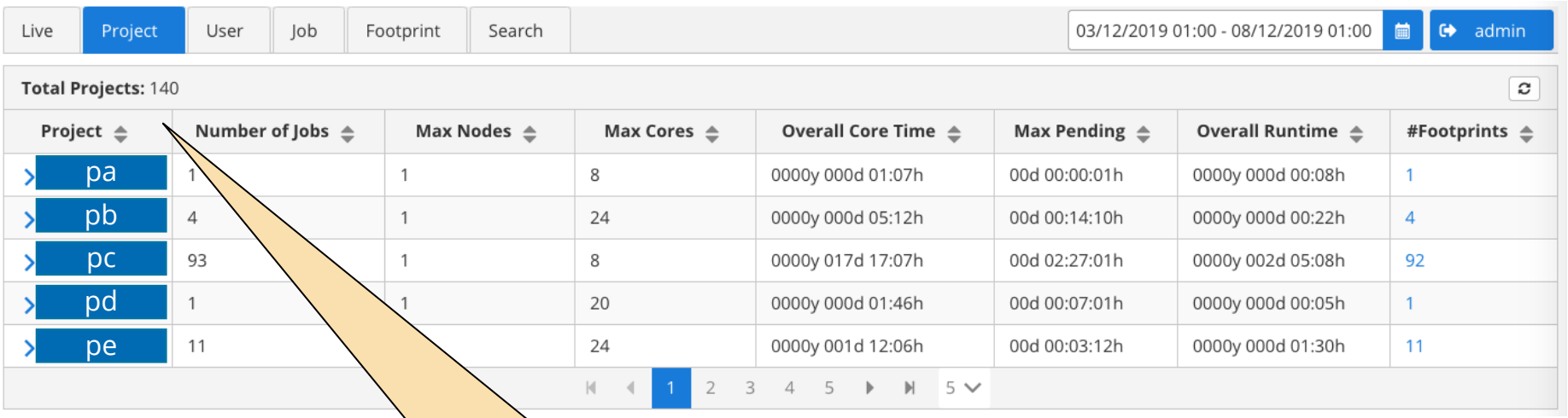


PIKA Runtime Data Storage with InfluxDB

Each metric value is tagged with timestamp, hostname as well as CPU or GPU

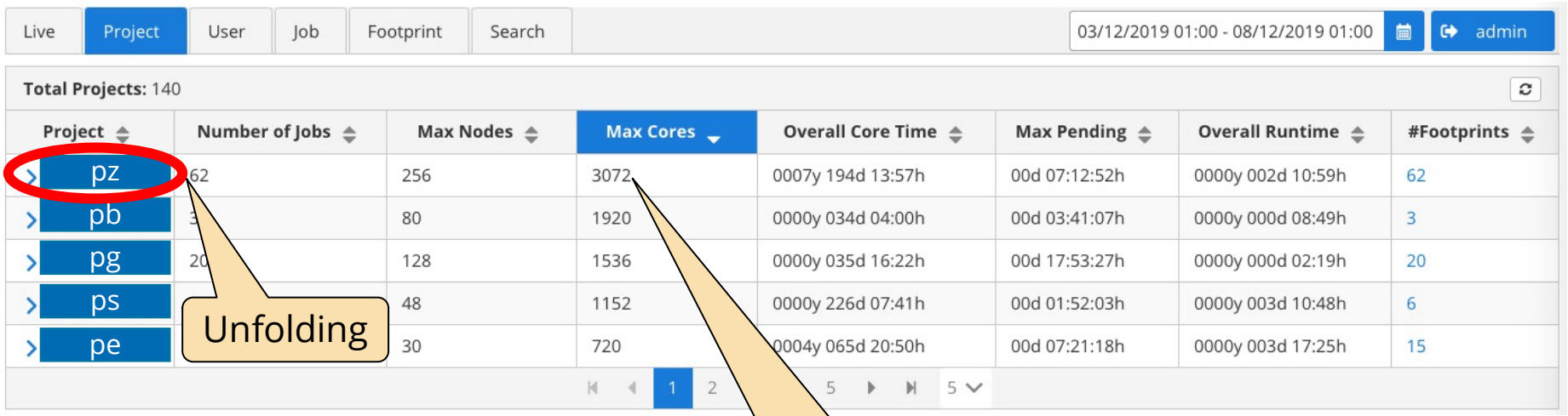
- Short-term storage
  - Retention policy: data older than 4 weeks is deleted
  - Shard length: 7 days
  - Complete shards are backed up and imported in long-term database
  - Requires SSD to handle more than 2K nodes
  - About 36GB of disk memory
- Long-term storage
  - No retention policy
  - Shard length: 7 days
  - Ceph filesystem
  - About 9GB per week → 0.5TB per year

PIKA Job Visualization – Tables

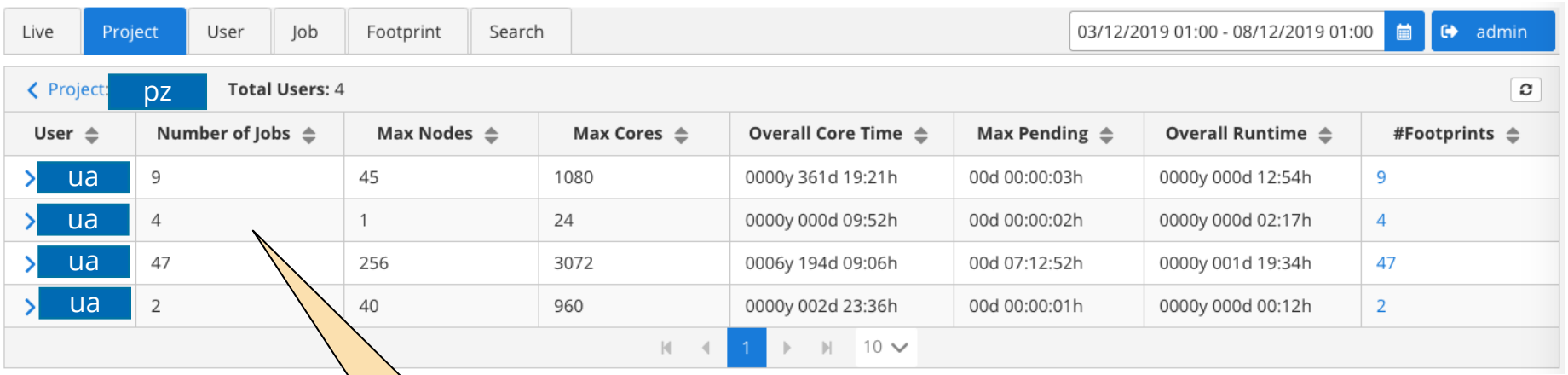


Jobs of 140 projects have been recorded for the selected time interval (top right).

PIKA Job Visualization – Tables

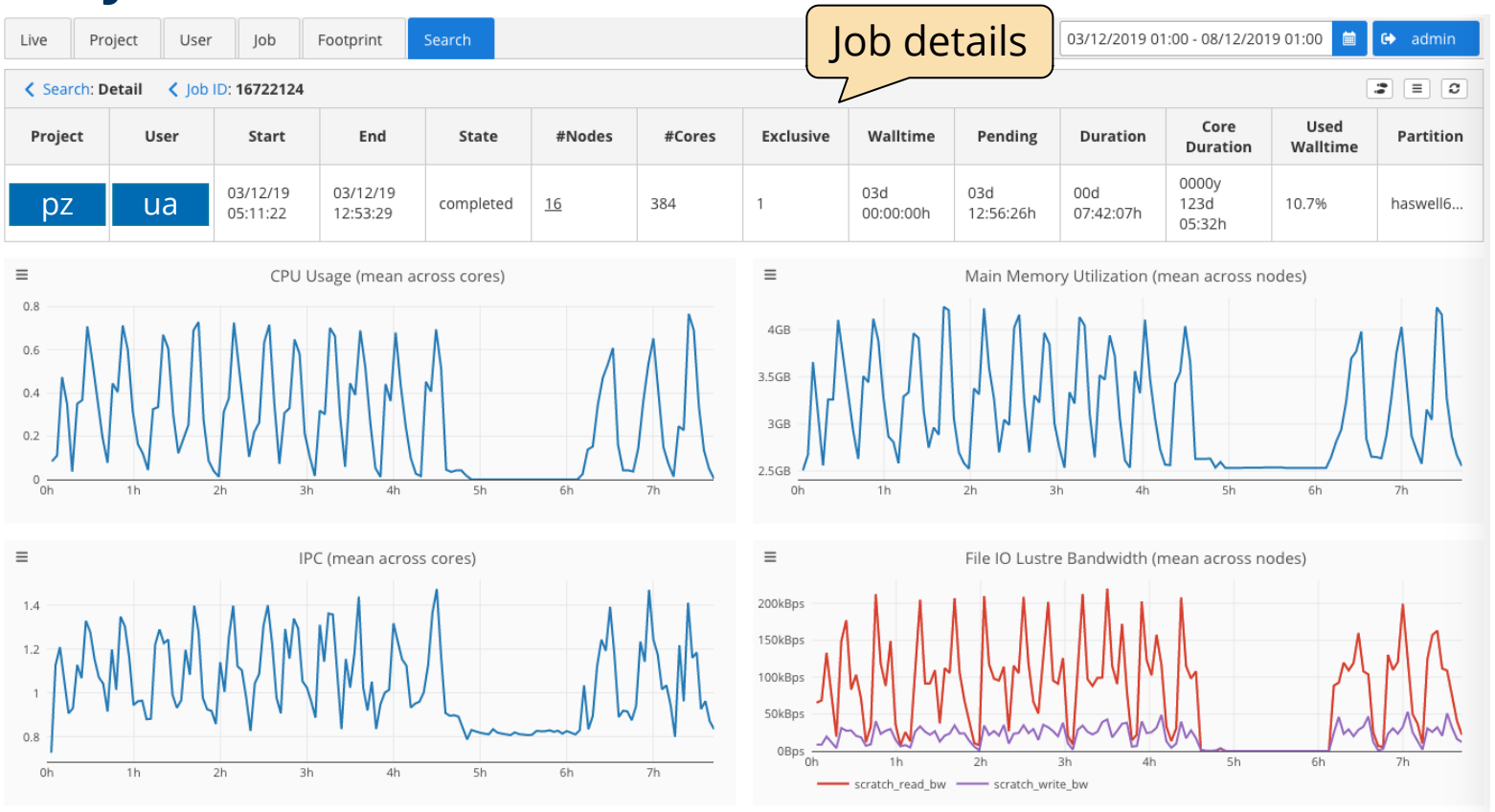


PIKA Job Visualization – Tables

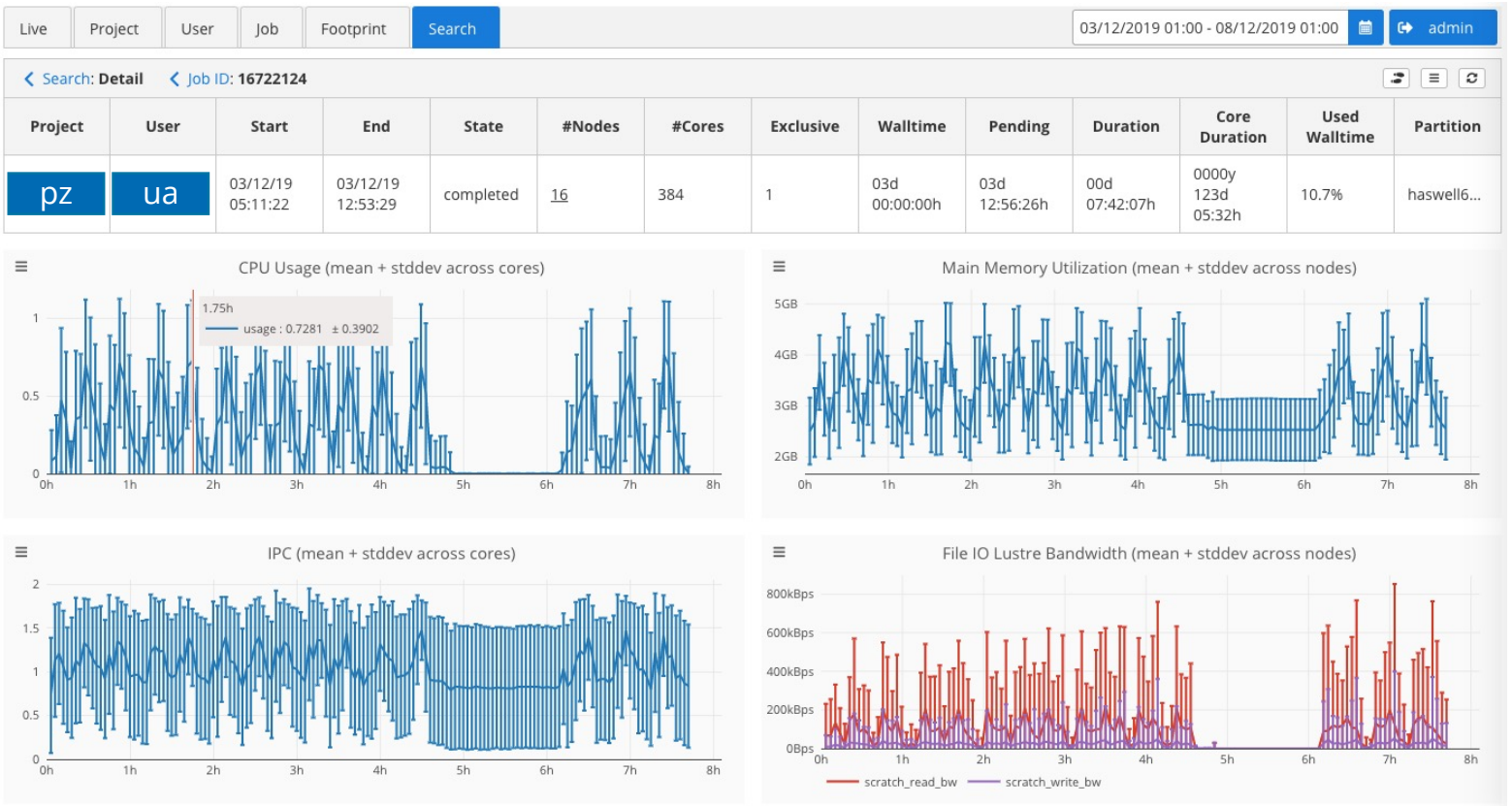


Project “pz” has 4 users and a total of 62 jobs.

PIKA Job Visualization – Timelines



PIKA Job Visualization – Timelines



PIKA Job Visualization – Timelines

PIKA Post Processing

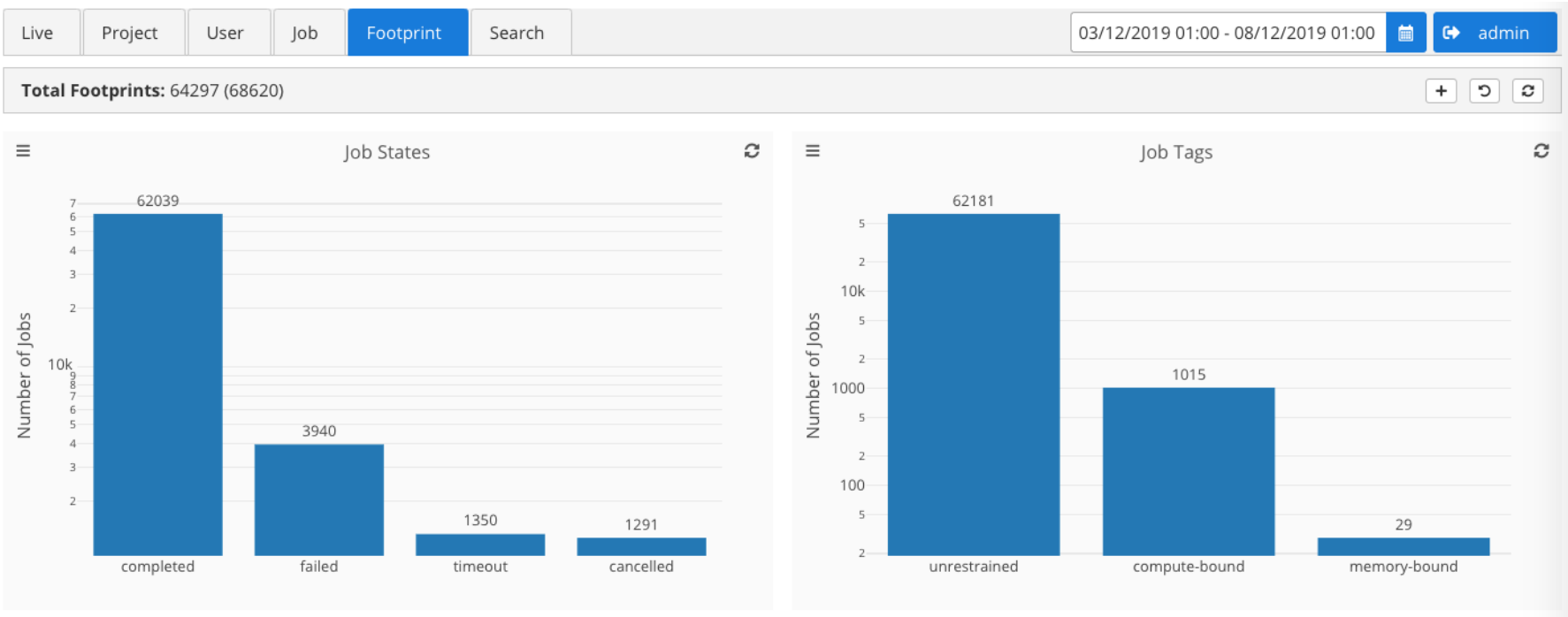
Job characterization via tagging

- **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

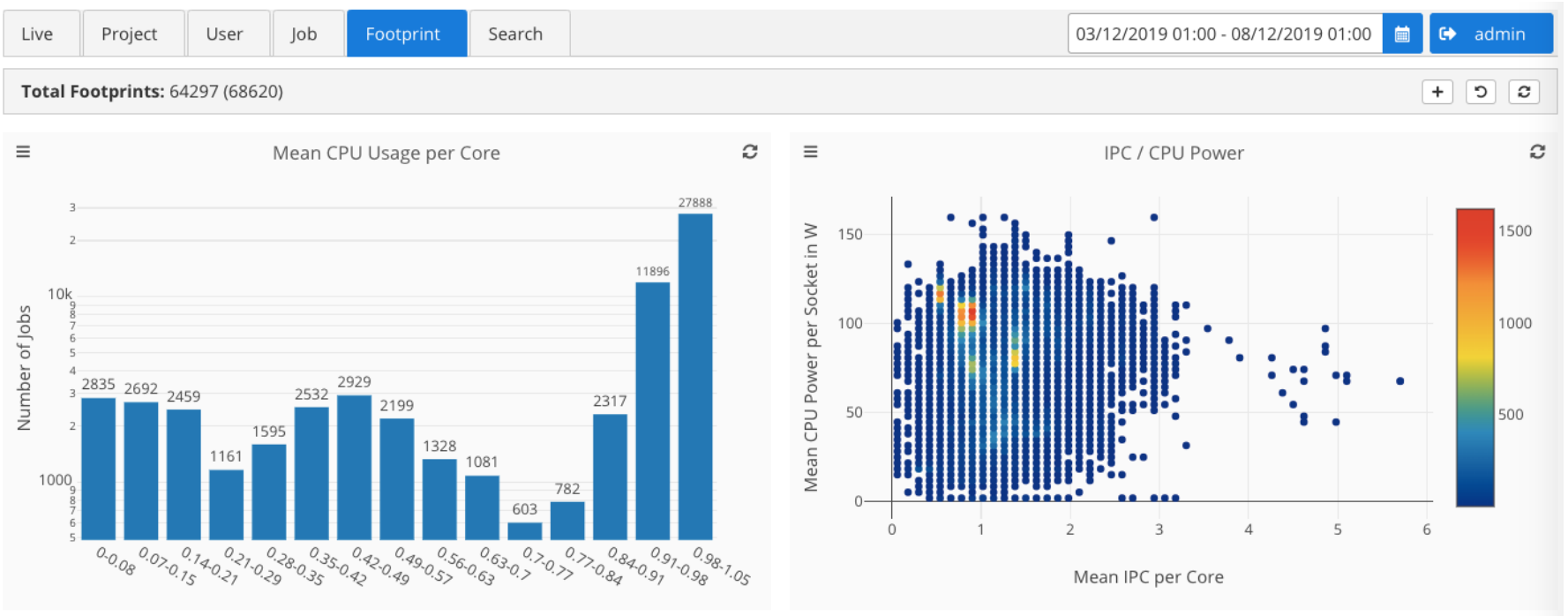
Tag Name	Formula and Threshold
unrestrained	-
memory-bound	$\frac{\texttt{memory bandwidth (measured)}}{\texttt{memory bandwidth (maximum)}} > 80\%$
compute-bound	$\frac{\texttt{FLOP/s (measured)}}{\texttt{FLOP/s (maximum)}} > 70\%$ or $\frac{\texttt{IPC (measured)}}{\texttt{IPC (optimal)}} > 60\%$
GPU-bound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

Automatic detection of job performance issues based on metadata (requested resources) and raw runtime data analysis (still in development)

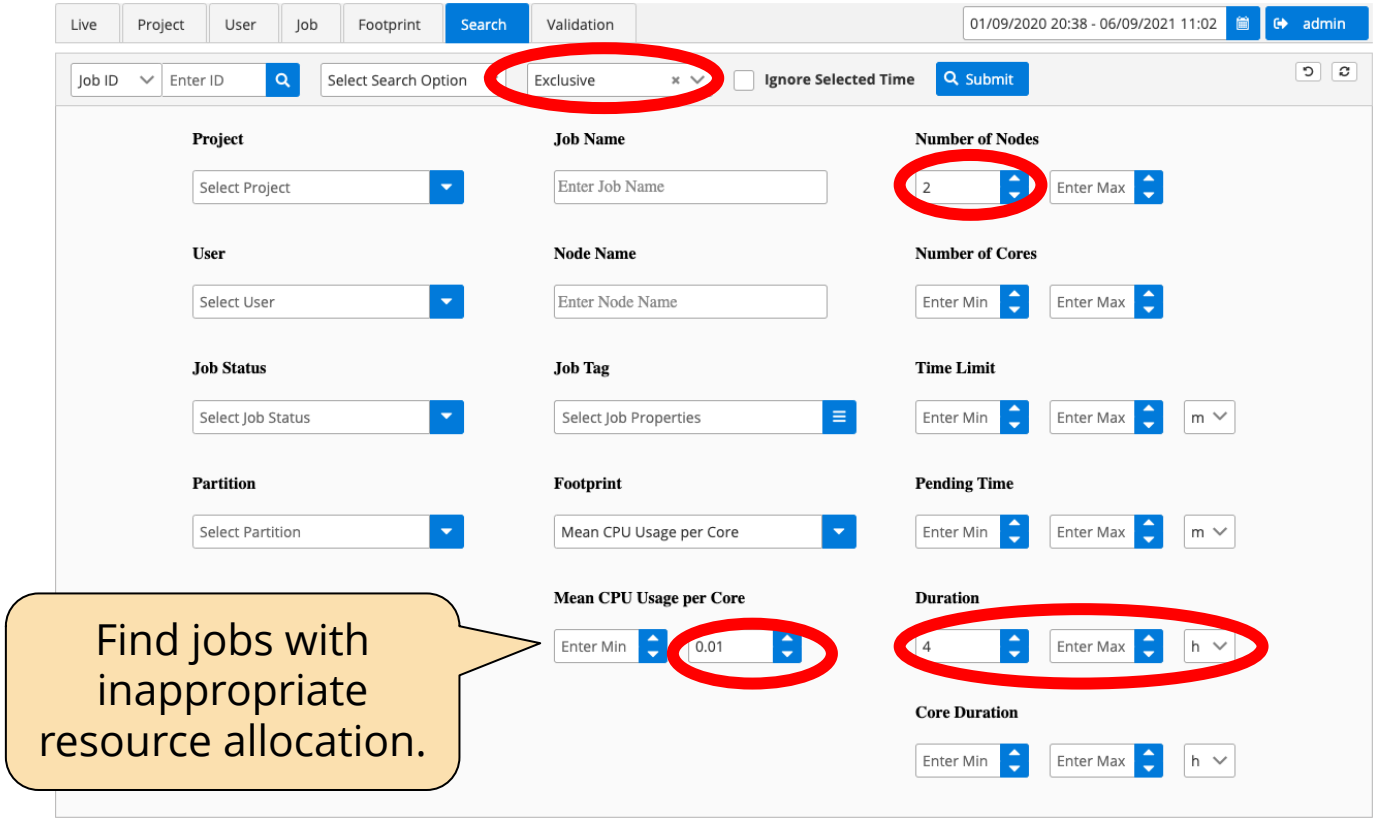
PIKA Job Visualization – Footprints



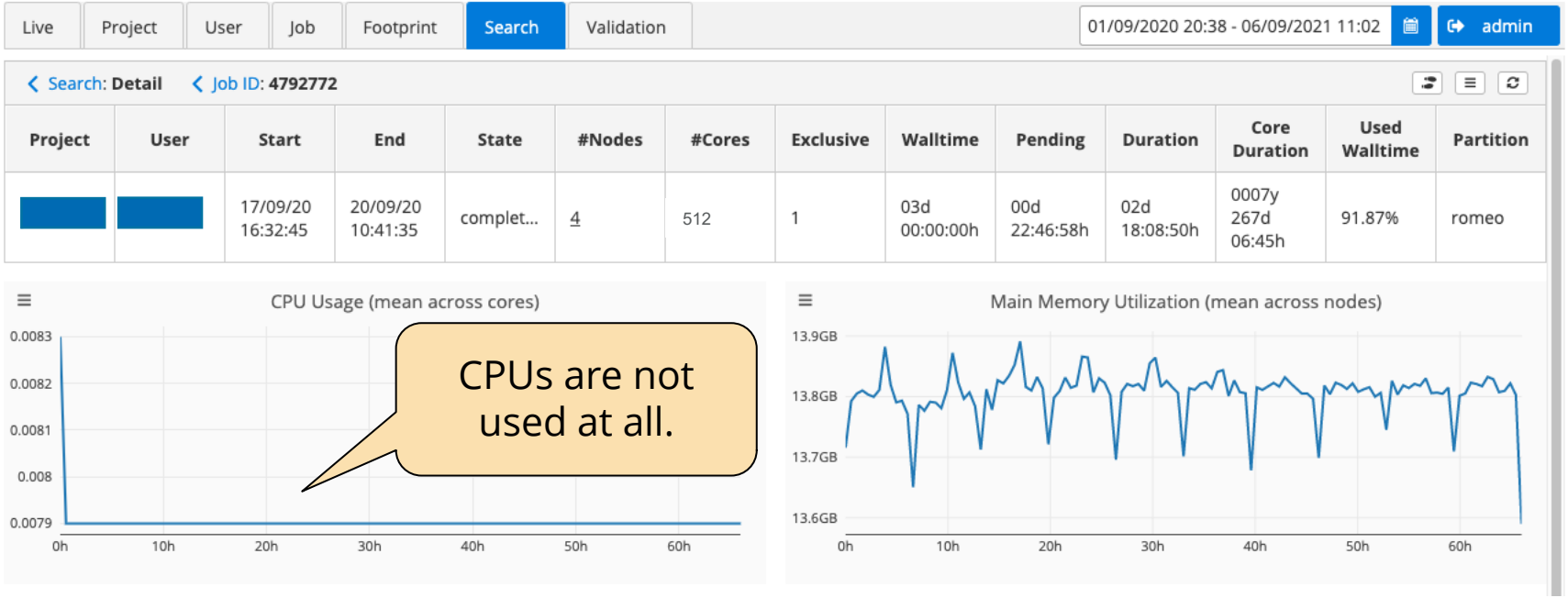
PIKA Job Visualization – Footprints



PIKA Filter Mask



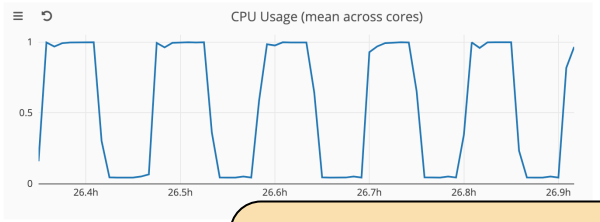
PIKA Filter Mask

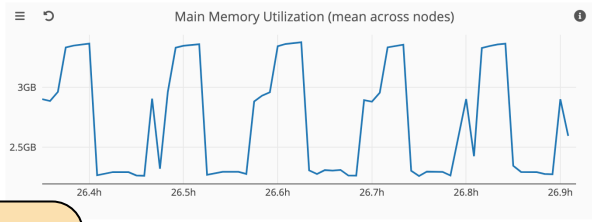


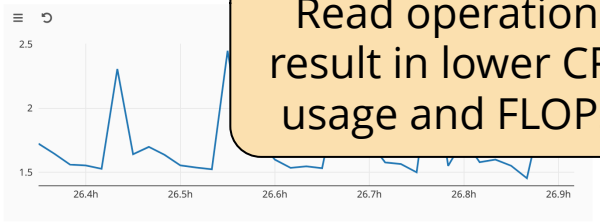
PIKA Job Performance Issues

Job Name: **case_3** Job ID: **21087795**

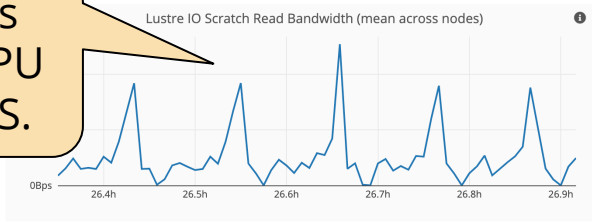
Project	Start	End	State	#Nodes	#Cores	Exclusive	Walltime	Pending	Duration	Core Time	Used Walltime	Partition
p_nhr_pika	01/12/21 22:33:21	04/12/21 22:33:52	timeout	1	24	0	03d 00:00:00h	06d 06:52:05h	03d 00:00:31h	0000y 072d 00:12h	100.01%	haswell64...

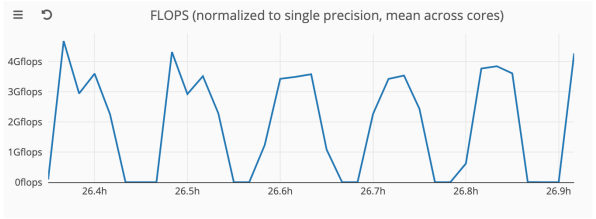


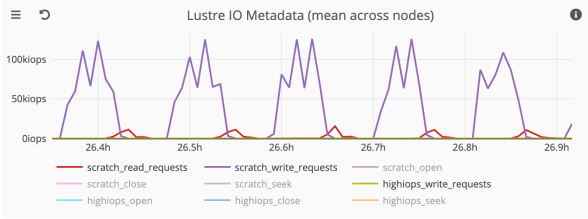




Read operations result in lower CPU usage and FLOPS.





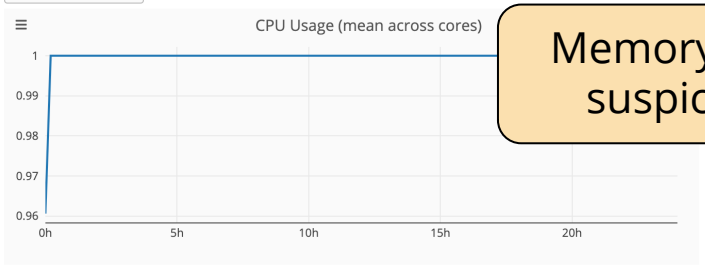


PIKA Job Performance Issues

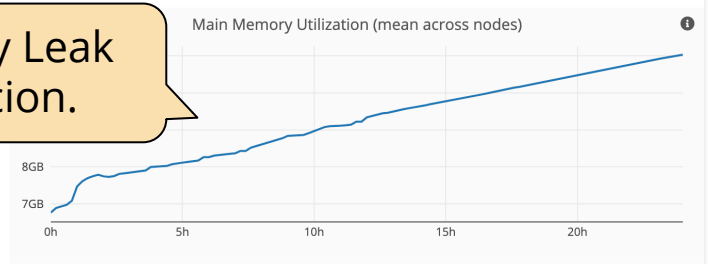
Job Name: **case_2** Job ID: **23482120**

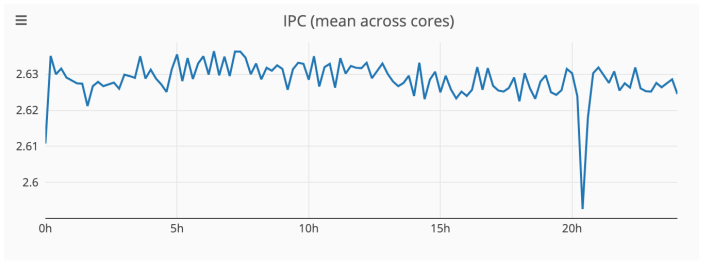
Project	Start	End	State	#Nodes	#Cores	Exclusive	Walltime	Pending	Duration	Core Time	Used Walltime	Partition
p_nhr_pika	09/02/22 13:03:42	10/02/22 13:04:00	timeout	4	96	0	01d 00:00:00h	00d 00:14:04h	01d 00:00:18h	0000y 096d 00:28h	100.02%	haswell64

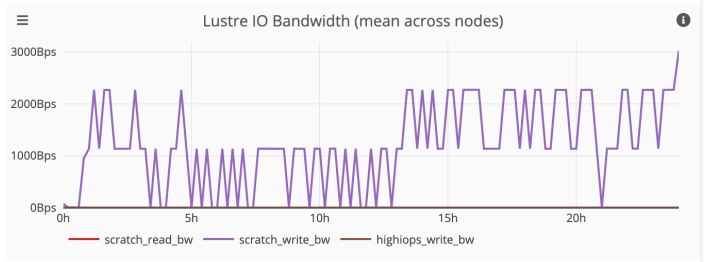
compute-bound



Memory Leak suspicion.







PIKA Demo

https://hpcmon.zih.tu-dresden.de/nhr

User / Password: TBA

Conclusion

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

- **Metric Data Collector:** An extension of the collection daemon collectd to record metric data on each compute node
- **Job Metadata Collector:** Centralized capture of job metadata for both exclusive and shared jobs using a Slurm PrEp Plugin
- **Frontend:** Powerful interactive GUI with top-down approach
- **Post-processing:** Python analysis scripts for job tagging and automatic detection of job performance issues

PIKA uses **InfluxDB** to store metric data and **MariaDB** to store job metadata. Most of the above components are available as **docker containers** and **RPM packages**.

Conclusion

— Continuous job performance monitoring is important
— Easy access to prepared performance data for projects, users and jobs
— Performance overview for admins, easy scan for pathological / suboptimal jobs
— Allows targeted consulting offers

Paper: 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.

PIKA available as open source software at
https://gitlab.hrz.tu-chemnitz.de/pika