Frank Winkler (frank.winkler@tu-dresden.de)
Robert Dietrich (now with NVIDIA)
Andreas Knüpfer (andreas.knuepfer@tu-dresden.de)
Sebastian Oeste (sebastian.oeste@tu-dresden.de)
Wolfgang E. Nagel (wolfgang.nagel@tu-dresden.de)

# I/O Aspects in the Center-Wide and Job-Aware Cluster Monitoring System PIKA

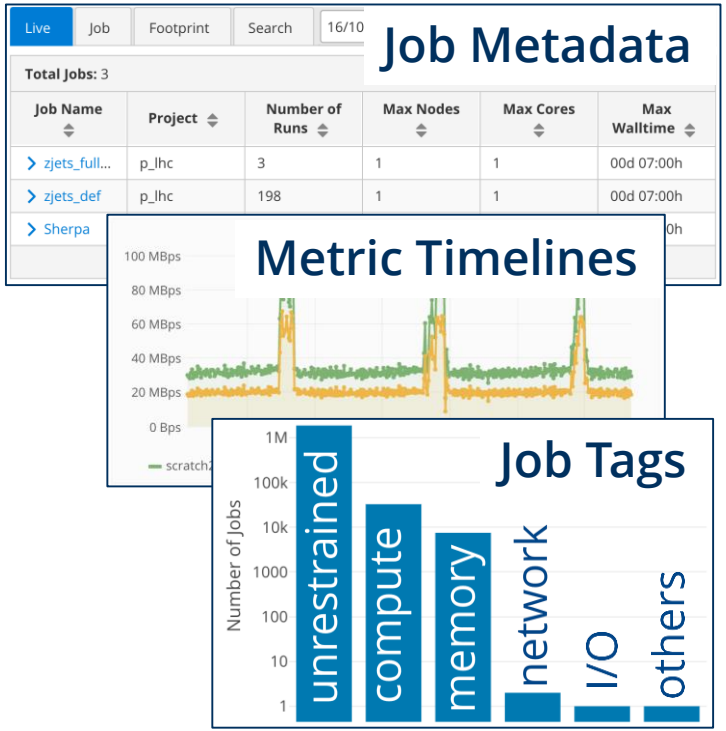
SC21 Analyzing Parallel I/O BoF

Center for Information Services & High Performance Computing

# PIKA: Center-Wide and Job-Aware Cluster Monitoring

— Non-intrusive data acquisition on all cluster nodes
— Continuous data collection

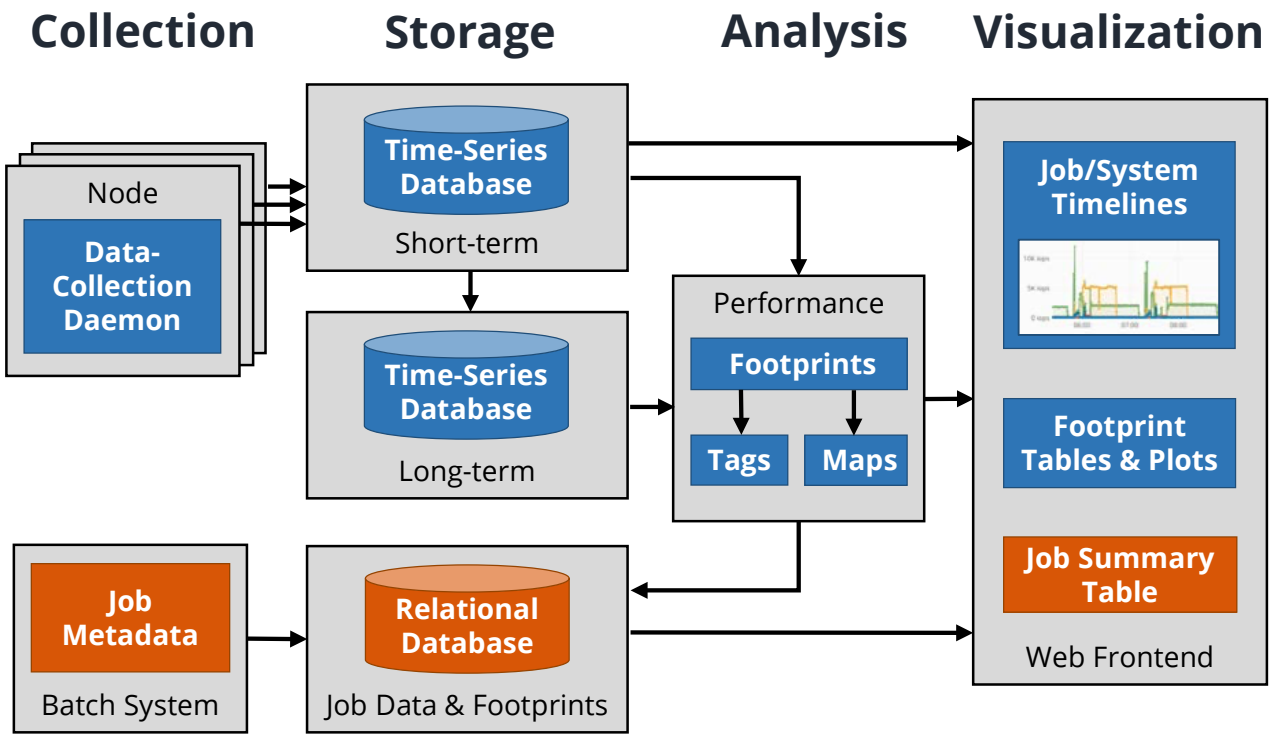
— Web frontend for live and post-mortem visualization
— Detection of pathological jobs
— Automatic job analysis and clasification
— Long-term data storage



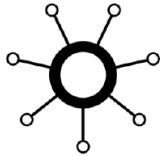

Funded by the DFG project ProPE

Continued as part of the NHR Center at ZIH

# PIKA Architecture Overview



# PIKA Performance Data Collection Daemon




Open-source collection daemon collectd [1]

- One collector/plugin for each metric source
- CPU counters are collected with LIKWID[2] every 60s*
  - Distinguish between per socket and per core
  - Multiplexing of event groups
- All other metrics are collected every 30s*
  - Lustre collector: read/write bandwidth and Lustre metadata [3]

[1] https://github.com/collectd/collectd

[2] https://github.com/RRZE-HPC/likwid

[3] https://gitlab.hrz.tu-chemnitz.de/pika/monitoring/-/blob/master/daemon/collectd/collectd-plugins/python/lustre_bw.py

* adjustable

# PIKA Metrics

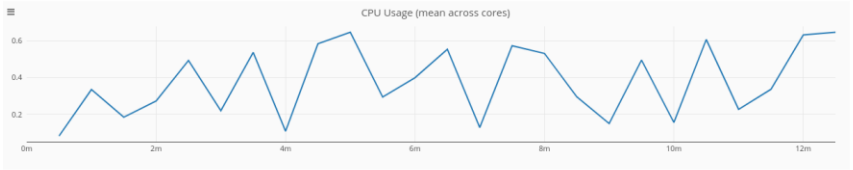
| Metric | Proposed Name | Data Source | Hardware Unit |
|---|---|---|---|
| CPU | | | |
| Usage | cpu_usage | /proc/stat | hardware thread |
| Main memory utilization | mem_used | /proc/meminfo | node |
| IPC | ipc | LIKWID | hardware thread |
| FLOPS (SP-normalized) | flops_any | LIKWID | hardware thread |
| Main memory bandwidth | mem_bw | LIKWID | CPU/socket |
| Power consumption | rapl_power | LIKWID | CPU/socket |
| Network bandwidth | net_bw | | |
| Infiniband bandwidth | ib_bw | /sys/class/infiniband/... | Infiniband device |
| Ethernet bandwidth | eth_bw | /sys/class/net/eth*/... | ethernet device |
| I/O bandwidth & metadata | | | |
| Local disk | read_bw, write_bw & read_ops, write_ops | /proc/diskstats | disk |
| Lustre | read_bw, write_bw & open, close, create, seek, fsync, read_requests, write_requests | /proc/fs/lustre/llite/*/stats | Lustre instance |
| GPU | | | |
| Usage | gpu_used | | |
| Memory Utilization | gpu_mem_used | NVML | GPU |
| Power Consumption | gpu_power | | |
| Temperature | gpu_temperature | | |

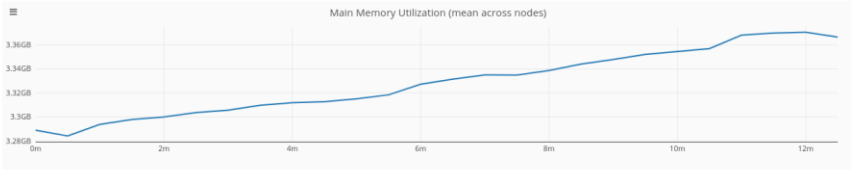
# Analysis and Visualization

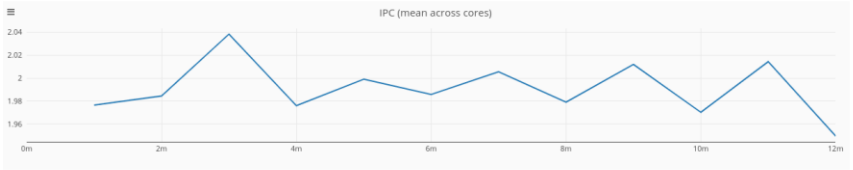
# PIKA Job Visualization

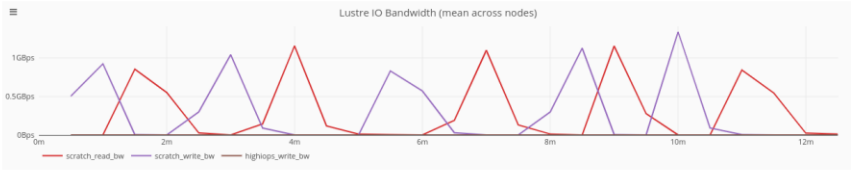
Search: JID=20837141 Job ID: 20837141

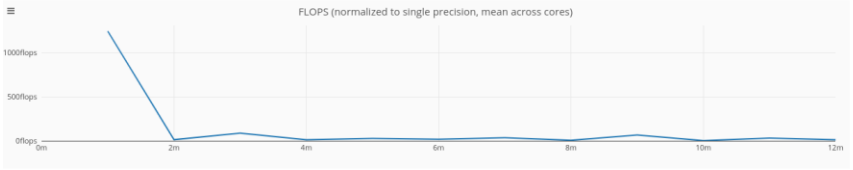
| Project | User | Start | End | State | #Nodes | #Cores | Exclusive | Walltime | Pending | Duration | Core Duration | Used Walltime | Partition |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dasupport | soeste | 16/11/21 17:09:10 | 16/11/21 17:21:43 | completed | 4 | 96 | 1 | 00d 01:30:00h | 00d 00:00:00h | 00d 00:12:33h | 0000y 000d 20:04h | 13.94% | haswell64 |

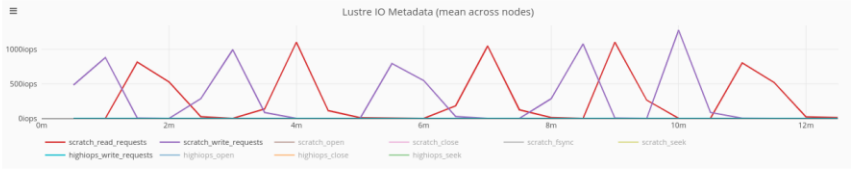


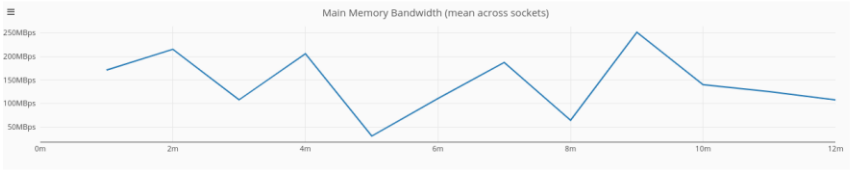



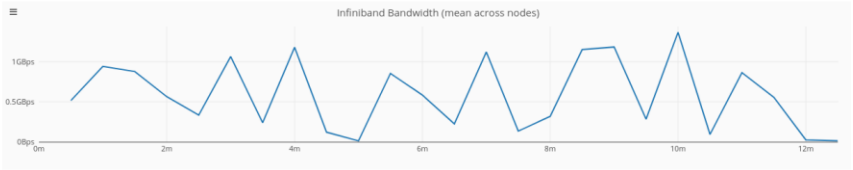



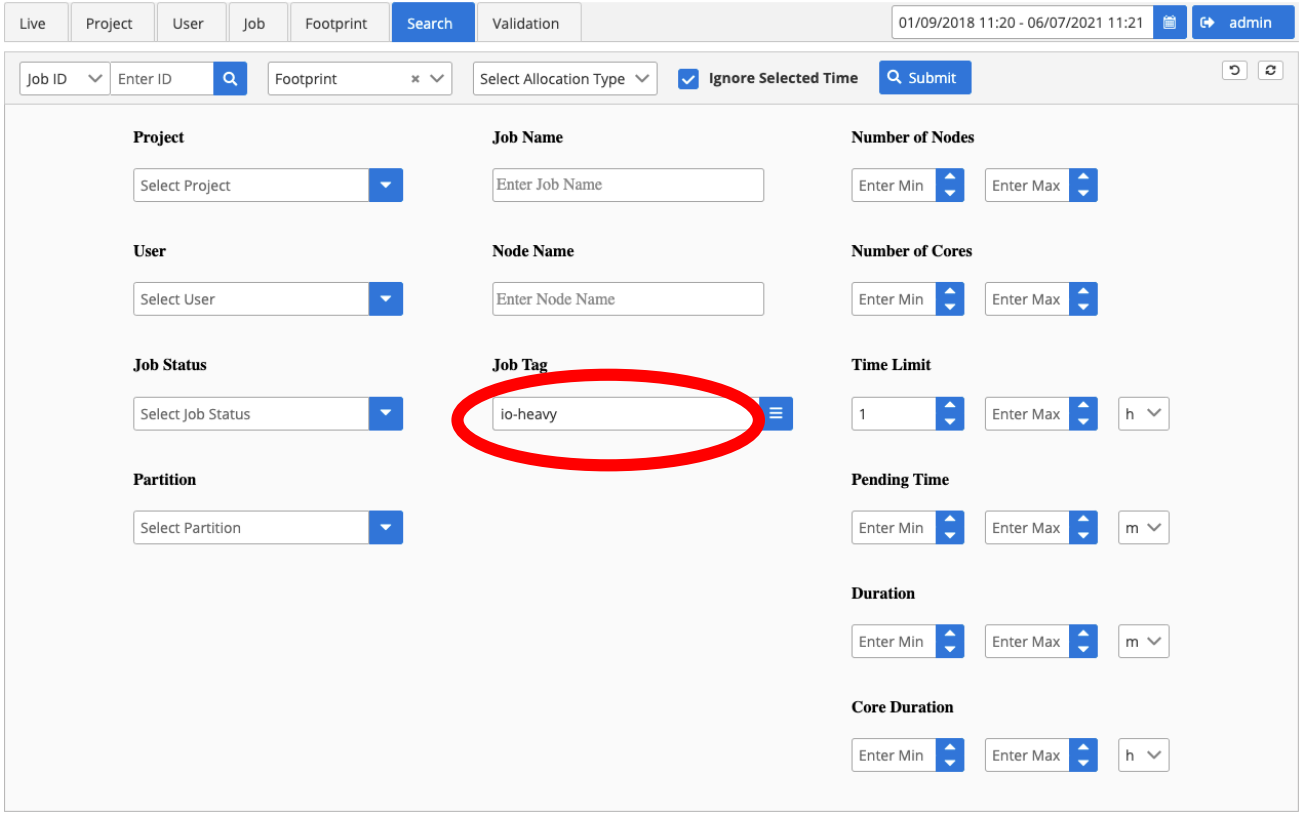



# PIKA Job Footprint Analysis - Tags

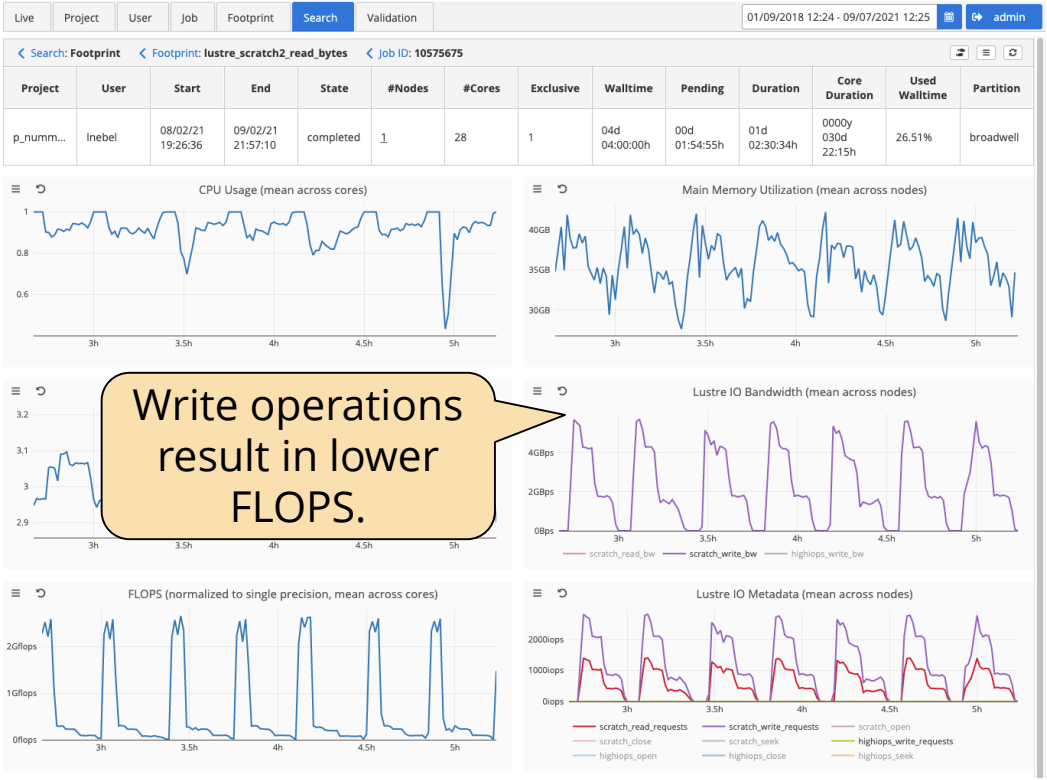


# PIKA Job Footprint Analysis - Tags

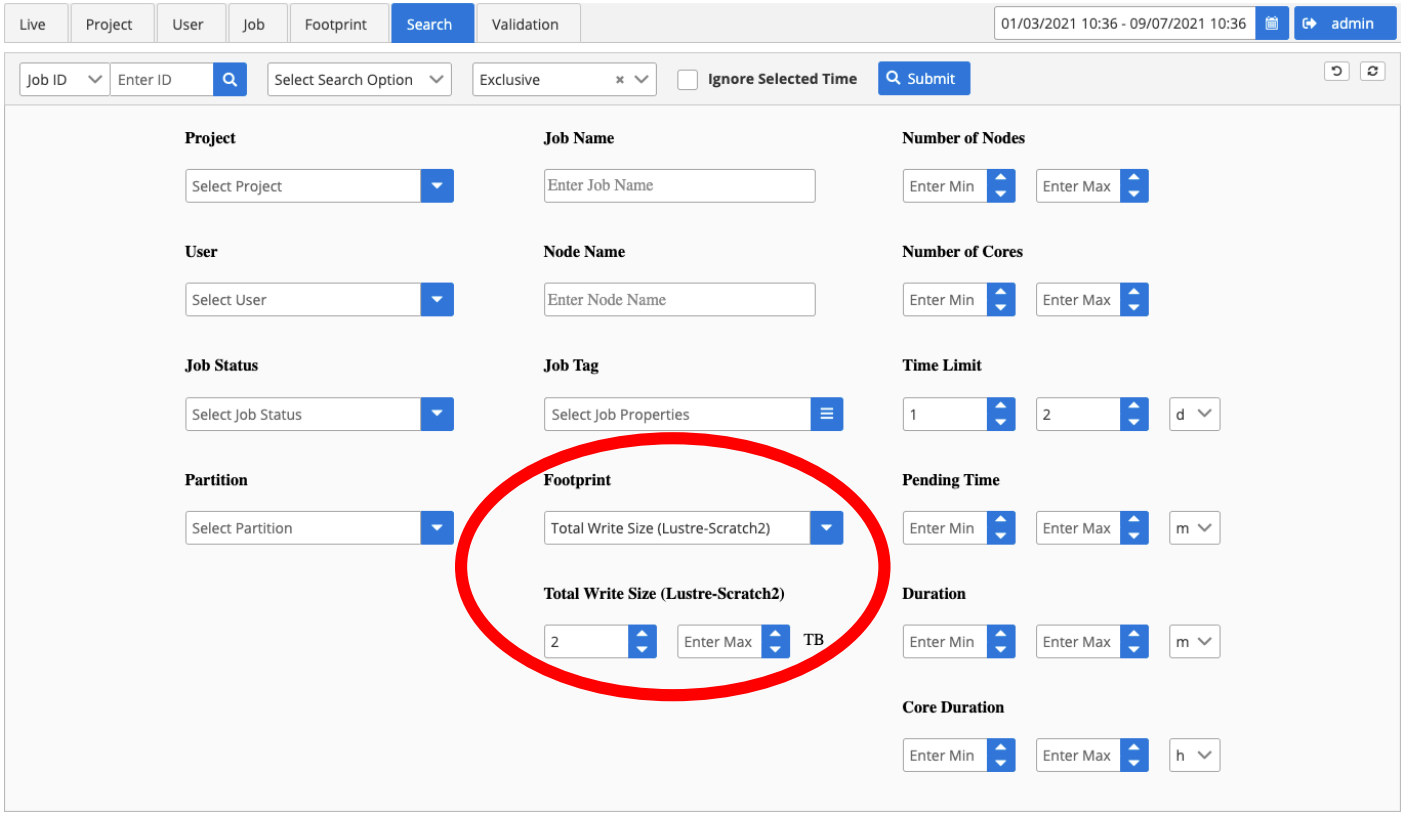
Automatic job tagging with heuristic thresholds

| 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 Job Visualization – Footprints



# PIKA Job Footprint Analysis – Search Jobs



# PIKA Job Footprint Analysis – Search Jobs

Sort jobs by largest write size from scratch2.

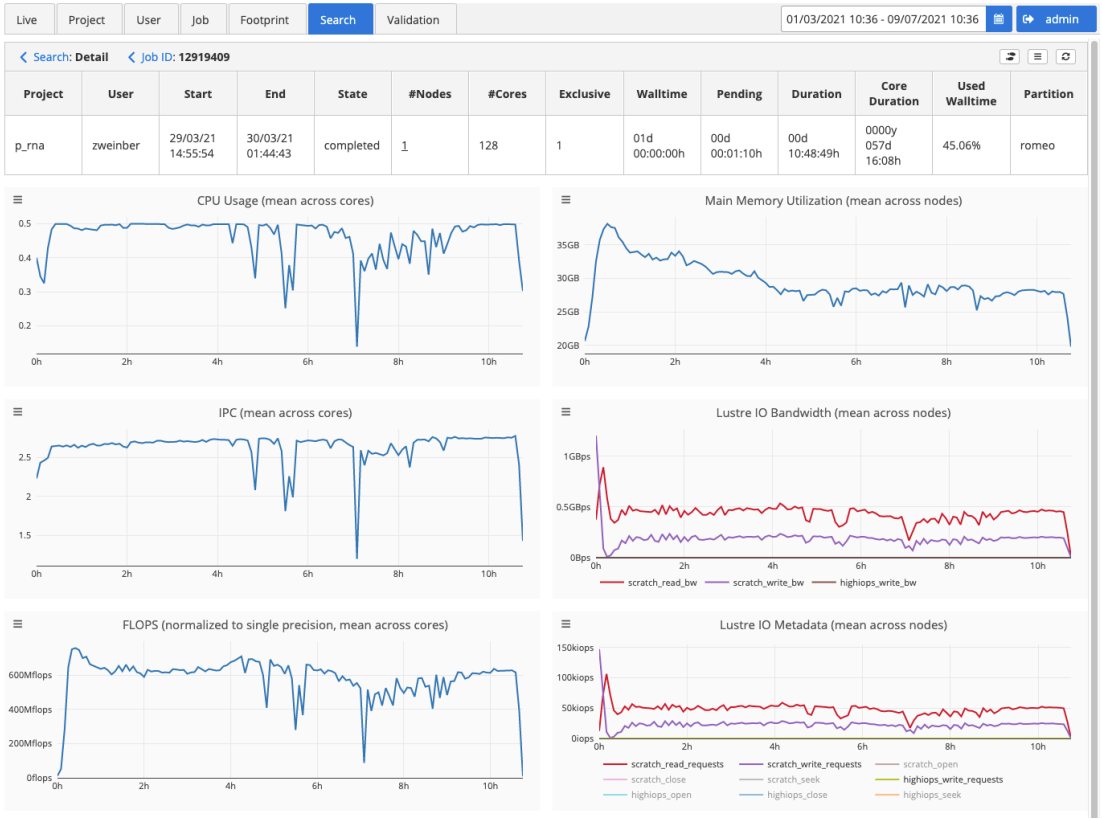
Live | Project | User | Job | Footprint | Search | Validation | admin

Search: **Detail** **Total Results:** 47

| Job ID | Project | User | Job Name | Start | End | State | #Nodes | #Cores | Exclusive | Walltime | Pending | Duration | Core Duration | Used Walltime | Partition | lustre scratch2 write bytes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 12919... | p_rna | zwein... | RunC... | 29/03/21 14:55:54 | 30/03/21 01:44:43 | compl... | 1 | 128 | 1 | 01d 00:00:00h | 00d 00:01:10h | 00d 10:48:49h | 0000y 057d 16:08h | 45.06% | romeo | 7.4e+12 |
| 12352... | p_func... | vankova | 1p90_... | 19/03/21 23:07:11 | 21/03/21 00:07:22 | timeout | 1 | 24 | 1 | 01d 00:00:00h | 00d 06:08:10h | 01d 01:00:11h | 0000y 025d 00:04h | 104.18% | haswe... | 7.2e+12 |
| 12352... | p_func... | vankova | Mst_sc... | 19/03/21 19:45:48 | 20/03/21 13:07:26 | compl... | 1 | 24 | 1 | 01d 00:00:00h | 00d 03:07:15h | 00d 17:21:38h | 0000y 017d 08:39h | 72.34% | haswe... | 6.0e+12 |
| 12352... | p_func... | vankova | rot-Ms... | 19/03/21 23:07:12 | 20/03/21 16:35:41 | compl... | 1 | 24 | 1 | 01d 00:00:00h | 00d 06:10:35h | 00d 17:28:29h | 0000y 017d 11:23h | 72.81% | haswe... | 6.0e+12 |
| 12768... | p_func... | vankova | St_sc-... | 27/03/21 07:55:59 | 28/03/21 09:56:24 | timeout | 1 | 24 | 1 | 01d 00:00:00h | 01d 15:40:54h | 01d 01:00:25h | 0000y 025d 00:10h | 104.2% | haswe... | 4.6e+12 |

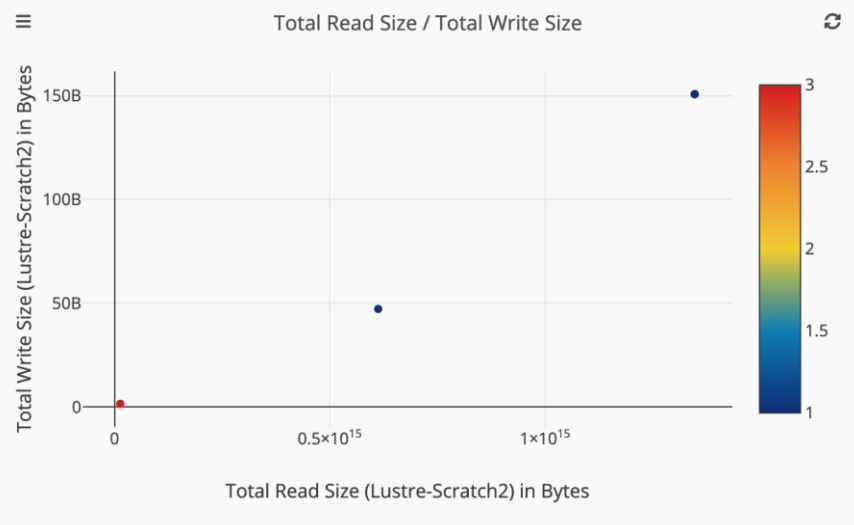
1 2 3 4 5 5

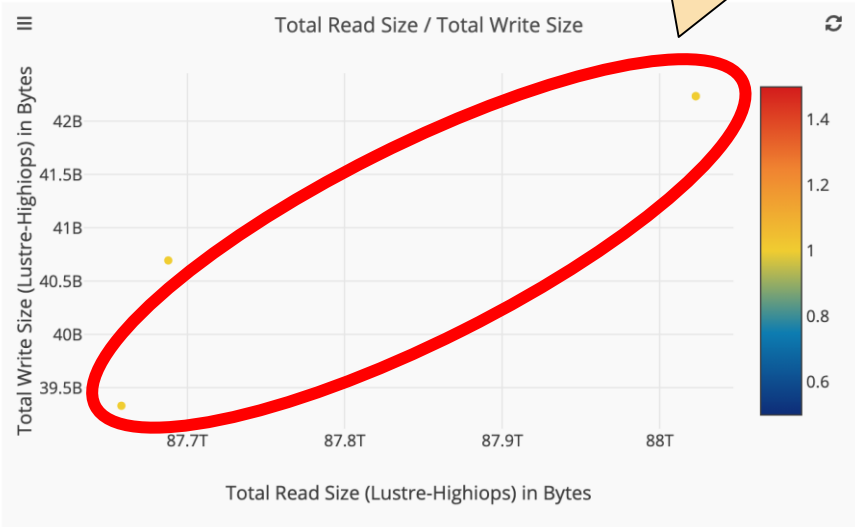
# PIKA Job Visualization



# PIKA Job Visualization – Compare Jobs

Different runs from the same code.



# PIKA Job Visualization – Read/ Write Distribution

Scatter plots (read/write) for all exclusive IO jobs that run at least one hour from September 2018 to present.


# Conclusion

We developed the **PIKA software stack** for center-wide and job-aware cluster monitoring

- Usage of open-source components
- Monitoring overhead is negligible
  - Can also be disabled (#SBATCH –no-monitoring)
- Automatic job categorization via job tagging (still WIP)
- Powerful interactive visualization (top-down approach)

# Roadmap

- Improve the simplistic „I/O-heavy“ classification
- Extend footprint analysis for metadata operation
- Record metrics also for other file systems e.g. BeeGFS
- Scan for jobs with “interesting” I/O behavior
- Heavy I/O phases might still be averaged out in long jobs – avg. BW not enough

# Thank you!

Software project available at https://gitlab.hrz.tu-chemnitz.de/pika

*“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.