

Track Energy & Emissions of User Jobs on HPC/AI Platforms using CEEMS

Mahendra Paipuri

CDSP, SciencesPo Paris, CNRS

FOSDEM 2026 - HPC, Big Data and Data Science Devroom

01st February 2026

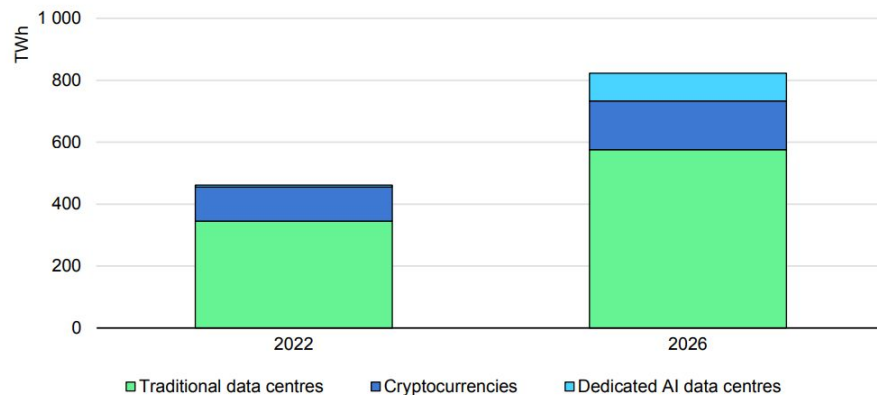
SciencesPo
CENTRE DE DONNÉES SOCIO-POLITIQUES



Context

- 40 % of DC consumption is due to servers
- Exploding usage of accelerators (GPUs) will only “accelerate” this snowball effect
- “Practical” solution is to engage the end users to optimize their workflows
- Need to provide relevant metrics and tools to encourage optimization

Estimated electricity demand from traditional data centres, dedicated AI data centres and cryptocurrencies, 2022 and 2026, base case



IEA 2024

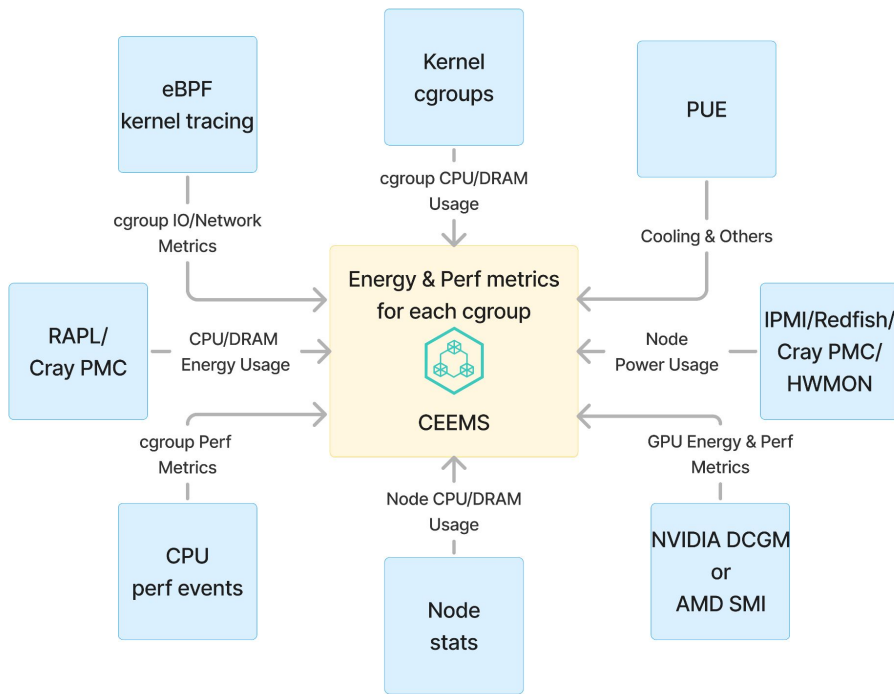
Compute Energy & Emissions Monitoring Stack (CEEMS)

CEEMS

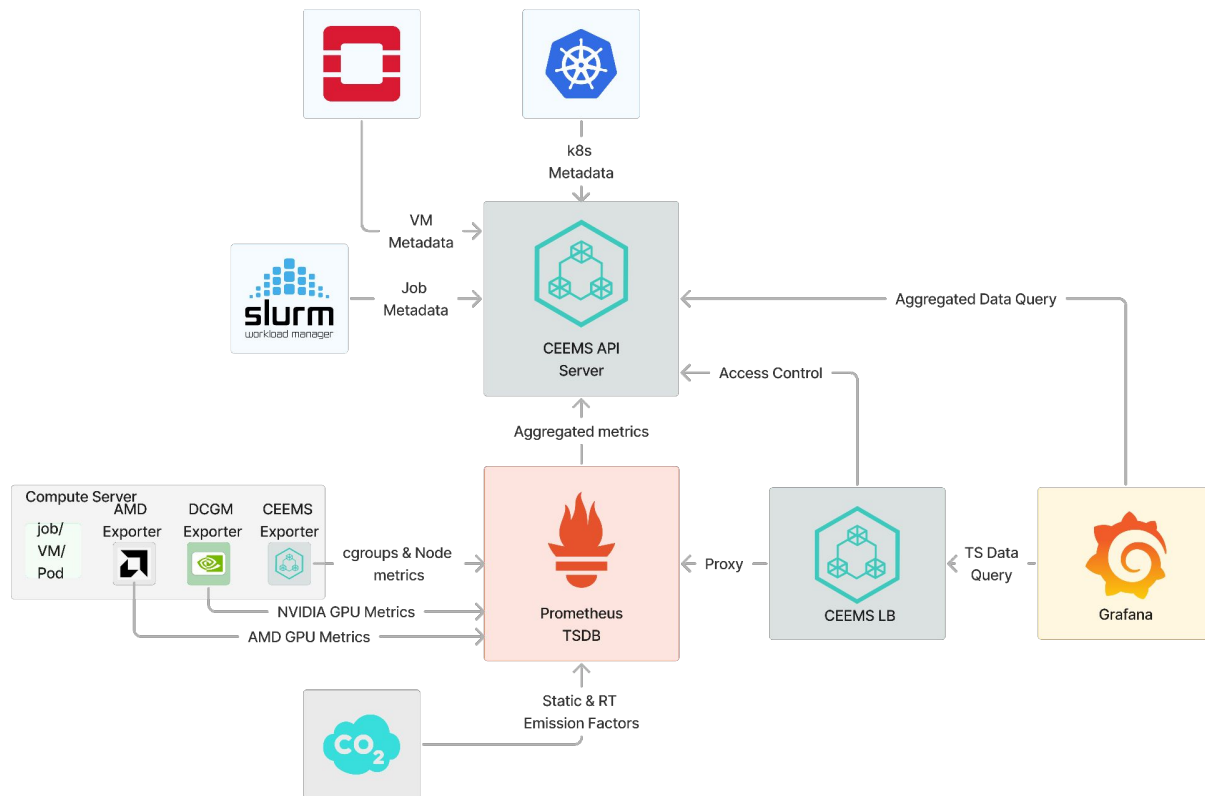
- Started as a tool to estimate energy consumption and equivalent emissions for HPC workloads
- Extended the stack to support Openstack and k8s
- A system level stack
- cgroups, perf subsystem, eBPF are at the heart of CEEMS
- Based on CNCF Opensource components. Prometheus as TSDB and Grafana for visualization. CLI client also available

CEEMS

Control Groups (cgroups) provide a mechanism for aggregating/partitioning sets of tasks, and all their future children, into hierarchical groups with specialized behaviour. For Linux, a SLURM job, an Openstack VM or a k8s pod is effectively a cgroup



CEEMS Architecture



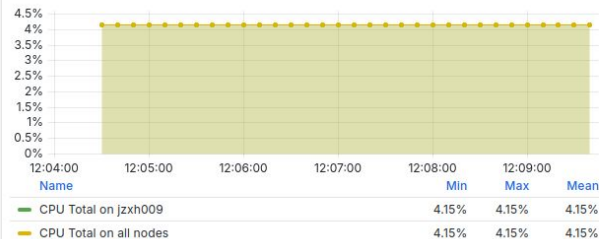
Features

- Monitors energy and performance metrics for different types of resource managers
- IO metrics are monitored in a file system agnostic manner using eBPF
- Supports different energy sources like RAPL, HWMON, Cray's PM Counters and BMC via IPMI or Redfish
- Supports NVIDIA (MIG and vGPU) and AMD (QPX, TPX,...) GPUs
- Realtime access to metrics via Grafana dashboards or using a CLI client tool
- Access control to Prometheus datasource in Grafana

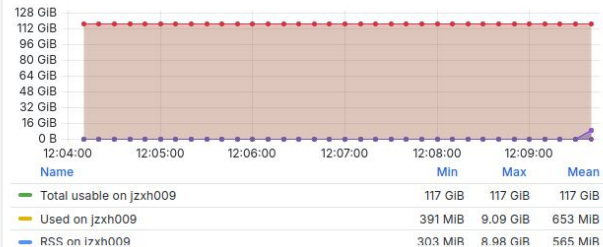
User Dashboards

▼ CPU Stats

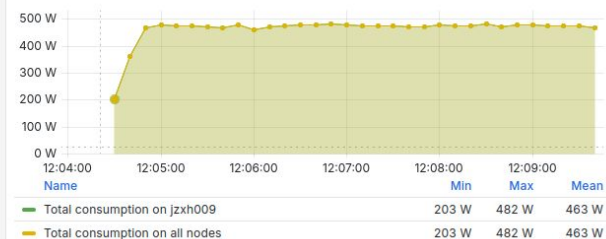
Job CPU Utilization ⓘ



Job CPU Memory Utilization ⓘ

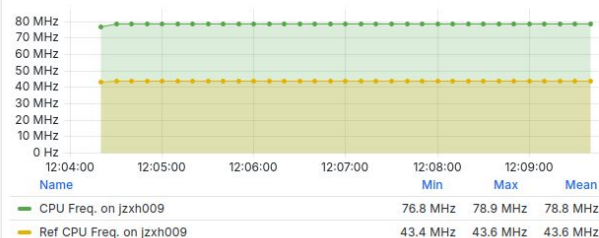


Job Host Power Usage ⓘ

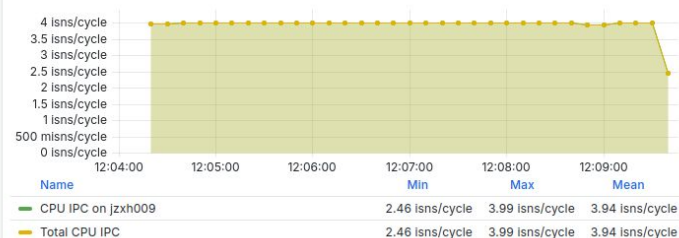


▼ CPU Performance Stats (Available only when CEEMS_ENABLE_PERF_EVENTS=1 env var is set in the job)

Job CPU Frequency ⓘ



Job CPU IPC ⓘ



Job CPU Software Events ⓘ

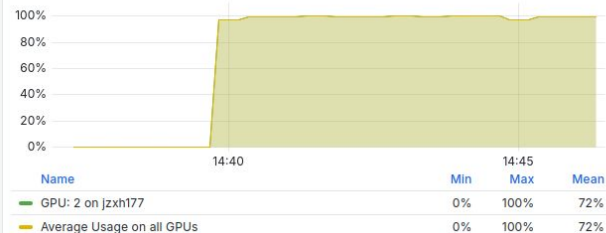


CPU Stats

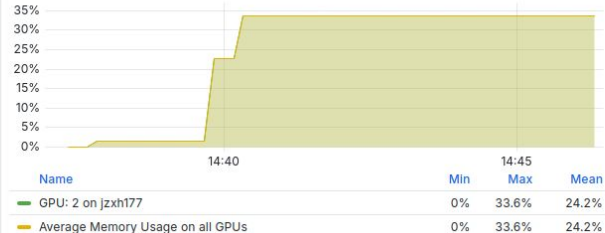
User Dashboards

GPU Stats

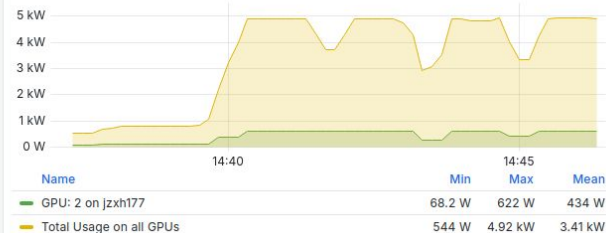
Job GPU Utilization ⓘ



Job GPU Memory Utilization ⓘ

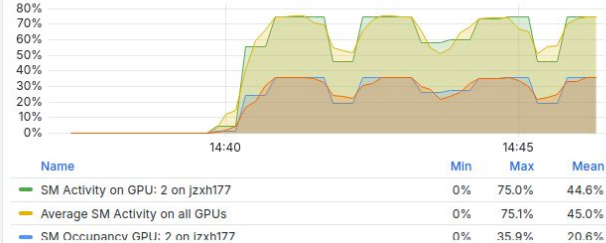


Job GPU Power Usage ⓘ

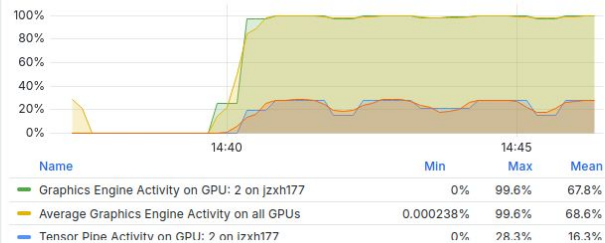


GPU Profiling Stats

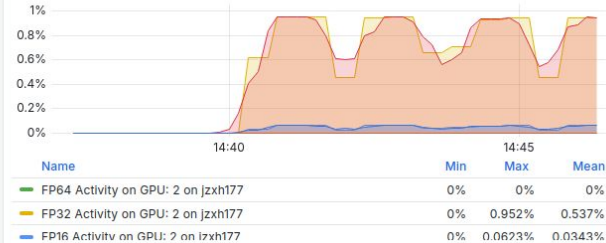
Job GPU SM Activity & Occupancy ⓘ



Job GPU Graphics and Tensor Engines Activity ⓘ



Job GPU FP Engines Activity ⓘ

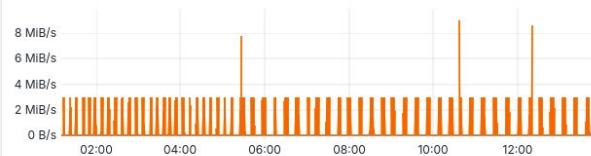


GPU Stats

User Dashboards

~ IO Stats

Job IO Read/Write Bandwidth ⓘ



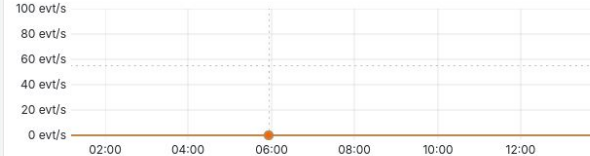
| Name | Min | Max | Mean |
|-----------------------------------|------------|------------|-----------|
| Read Bandwidth on WORK on r1i0n0 | 0 B/s | 0 B/s | 0 B/s |
| Write Bandwidth on WORK on r1i0n0 | 1.88 KIB/s | 8.97 MiB/s | 503 KIB/s |
| Total Read Bandwidth on WORK | 0 B/s | 0 B/s | 0 B/s |

Job IO Read/Write Requests ⓘ



| Name | Min | Max | Mean |
|----------------------------------|------------|-----------|------------|
| Read Requests on WORK on r1i0n0 | 0 req/s | 0 req/s | 0 req/s |
| Write Requests on WORK on r1i0n0 | 23.9 req/s | 871 req/s | 43.4 req/s |
| Total Read Requests on WORK | 0 req/s | 0 req/s | 0 req/s |

Job IO Read/Write Errors ⓘ



| Name | Min | Max | Mean |
|---------------------------------|---------|---------|---------|
| Read Requests on WORK on r1i0n0 | 0 evt/s | 0 evt/s | 0 evt/s |
| Write Errors on WORK on r1i0n0 | 0 evt/s | 0 evt/s | 0 evt/s |
| Total Read Errors on WORK | 0 evt/s | 0 evt/s | 0 evt/s |

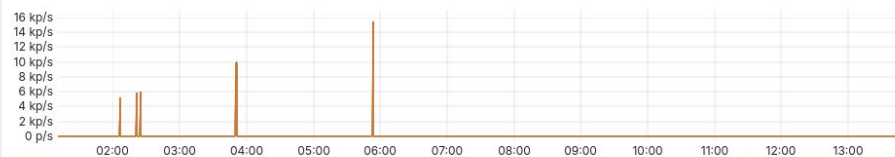
~ Network Stats

Job Network Ingress/Egress Bandwidth ⓘ



| Name | Min | Max | Mean |
|--------------------------------------|---------|------------|------------|
| Ingress Bandwidth tcp/ipv4 on r1i0n0 | 619 B/s | 1.69 KIB/s | 1.54 KIB/s |
| Egress Bandwidth tcp/ipv4 on r1i0n0 | 619 B/s | 3.05 MiB/s | 8.56 KIB/s |
| Total Ingress Bandwidth tcp/ipv4 | 619 B/s | 1.69 KIB/s | 1.54 KIB/s |

Job Network Ingress/Egress Packets ⓘ



| Name | Min | Max | Mean |
|------------------------------------|----------|-----------|----------|
| Ingress Packets tcp/ipv4 on r1i0n0 | 6.40 p/s | 15.4 kp/s | 49.7 p/s |
| Egress Packets tcp/ipv4 on r1i0n0 | 6.40 p/s | 15.4 kp/s | 49.7 p/s |
| Total Ingress Packets tcp/ipv4 | 6.40 n/s | 15.4 kn/s | 49.7 n/s |

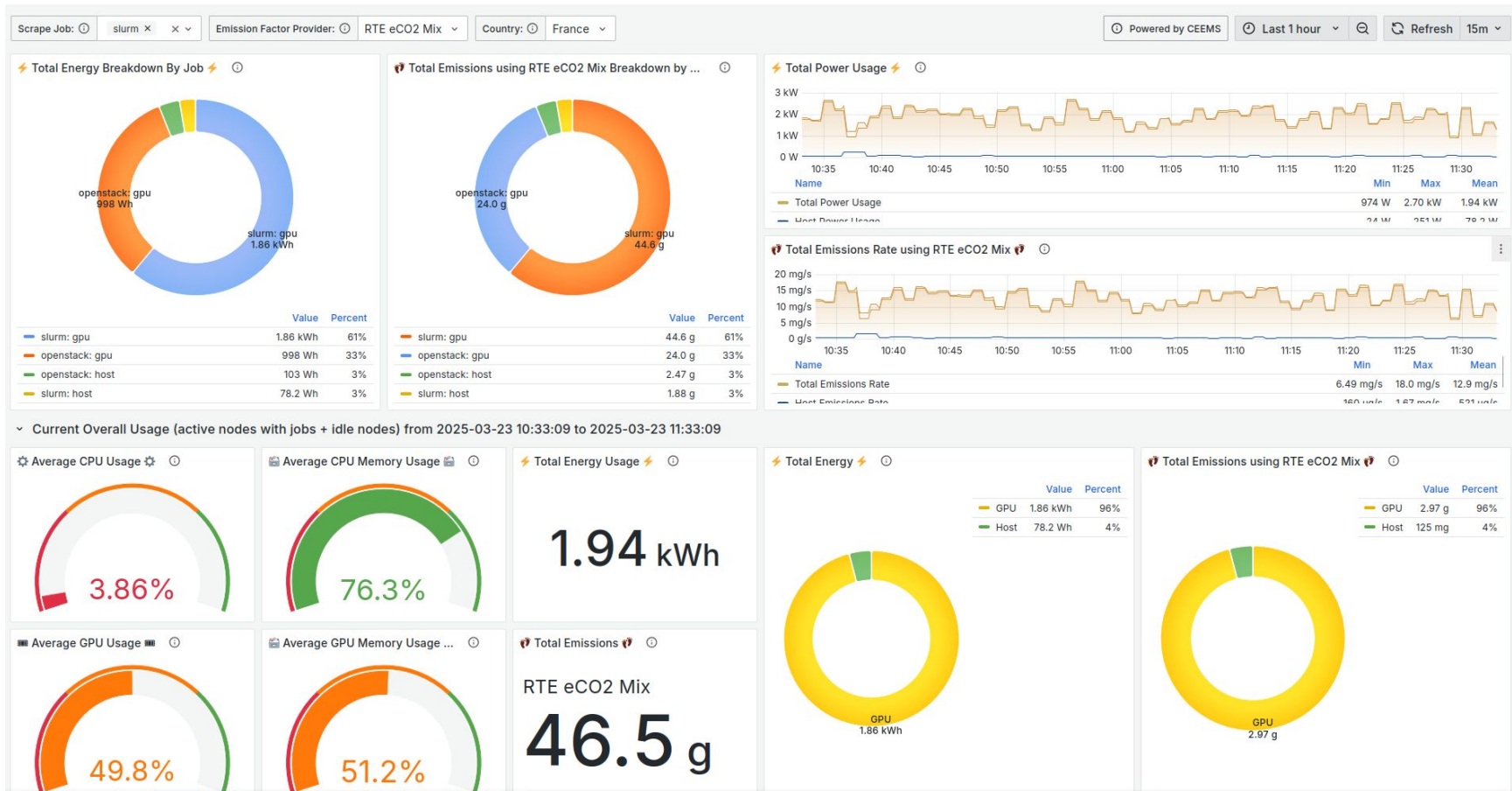
IO/Network Stats

CLI Client Tool

| JOB ID | ACCOUNT | ELAPSED | CPU US AGE(%) | CPU ME M. USA GE(%) | HOST ENE RGY(KWH) | HOST EMISSIONS(GMS) | | | GPU US AGE(%) | GPU ME M. USA GE(%) | GPU ENER GY(KWH) | GPU EMISSIONS(GMS) | | |
|---------|---------|----------|------------------|---------------------------|----------------------|---------------------|------------|-----------|------------------|---------------------------|---------------------|--------------------|------------|-----------|
| | | | | | | EMAPS_TOTAL | OWID_TOTAL | RTE_TOTAL | | | | EMAPS_TOTAL | OWID_TOTAL | RTE_TOTAL |
| 106 | bedrock | 00:10:05 | 99.32 | 3.39 | 0.053818 | 4.725182 | 5.648855 | 3.860008 | | | | | | |
| 108 | bedrock | 00:10:04 | 99.60 | 2.51 | 0.055842 | 5.091815 | 5.840380 | 4.197307 | | | | | | |
| 118 | bedrock | 00:10:03 | 99.65 | 1.17 | 0.061474 | 4.450334 | 6.512757 | 3.683035 | | | | | | |
| 131 | bedrock | 00:10:04 | 99.71 | 2.15 | 0.055742 | 1.835111 | 5.562944 | 1.245254 | | | | | | |
| 134 | bedrock | 00:20:12 | 0.53 | 0.73 | 0.004463 | 0.030868 | 0.100538 | 0.021321 | | | | | | |
| 138 | bedrock | 00:10:00 | 99.61 | 1.17 | 0.056302 | 2.595522 | 5.570695 | 1.837668 | | | | | | |
| 150 | bedrock | 00:20:11 | 0.54 | 0.74 | 0.003862 | 0.076767 | 0.086878 | 0.058934 | | | | | | |
| 154 | bedrock | 00:10:19 | 99.48 | 2.86 | 0.055671 | 4.906742 | 6.610783 | 4.127894 | | | | | | |
| 162 | bedrock | 00:10:22 | 96.51 | 3.66 | 0.055507 | 3.274911 | 4.711376 | 2.497813 | | | | | | |
| 163 | bedrock | 00:10:28 | 99.71 | 3.03 | 0.051746 | 3.673949 | 4.392128 | 2.780309 | | | | | | |
| 169 | bedrock | 00:10:19 | 99.71 | 1.17 | | | | | | | | | | |
| 181 | bedrock | 00:20:14 | 0.56 | 0.74 | 0.001518 | 0.115373 | 0.085070 | 0.081976 | 36.31 | 38.11 | 0.184776 | 14.042940 | 10.354560 | 9.977878 |
| 183 | bedrock | 00:10:09 | 99.68 | 1.17 | 0.049606 | 3.676648 | 2.779826 | 2.926728 | 37.87 | 37.97 | 0.187746 | 13.919683 | 10.521023 | 11.077016 |
| 229 | bedrock | 00:10:21 | 99.57 | 1.99 | 0.048258 | 1.930318 | 2.704308 | 1.109933 | 38.71 | 37.36 | 0.197287 | 7.891462 | 11.055660 | 4.537591 |
| 232 | bedrock | 00:10:24 | 99.63 | 1.17 | 0.050244 | 1.385482 | 2.815615 | 0.954640 | 31.90 | 35.88 | 0.131236 | 3.618456 | 7.354267 | 2.493479 |
| 269 | bedrock | 00:10:01 | 99.69 | 1.17 | 0.048866 | 2.738386 | 2.123290 | 22.18 | 24.35 | 0.026367 | 1.477547 | 1.141505 | | |
| 274 | bedrock | 00:10:16 | 97.72 | 3.49 | 0.054060 | 3.029430 | 2.324568 | | | | | | | |
| Summary | | | | | | | | | | | | | | |
| 20 | bedrock | 03:23:27 | 69.84 | 1.73 | 0.706980 | 37.769023 | 59.189969 | 33.830679 | 35.74 | 35.32 | 0.727410 | 39.472541 | 40.763058 | 29.227470 |

cacct - Exports time series data of metrics in CSV format

Cluster Dashboards - Operators



Cluster Dashboards - Operators

| Usage Stats ⓘ | | | | | | | | | | | | |
|---------------|------------------------|------------------|-----------------------|-----------------------|----------------------|----------------------|-----------------------|-----------------------|-----------------------|-----------------------|--|--|
| Project ⓘ | Users (uniqueValues) ⓘ | Num Jobs (sum) ⓘ | Avg. CPU Usage (m%) ⓘ | Avg. GPU Usage (m%) ⓘ | Avg. CPU Mem Usage ⓘ | Avg. GPU Mem Usage ⓘ | Total CPU Energy Us ⓘ | Total GPU Energy Us ⓘ | Total CPU Emissions ⓘ | Total GPU Emissions ⓘ | | |
| | | 49033 | 6.40 | 40.3 | 5.59 | 25.8 | 1253 | 3670 | 18527 | 55828 | | |
| | | 18142 | 22.7 | 2.71 | 2.63 | 1.01 | 188 | 279 | 3152 | 4635 | | |
| | | 16060 | 47.7 | 59.7 | 28.7 | 15.7 | 7459 | 19141 | 119818 | 306113 | | |
| | | 13774 | 8.10 | 68.3 | 3.30 | 23.4 | 551 | 1642 | 7816 | 22799 | | |
| | | 13323 | 73.9 | 0 | 24.2 | 0 | 140 | 0 | 2023 | 0 | | |
| | | 12742 | 44.3 | 34.3 | 0.413 | 2.55 | 69.6 | 67.7 | 1036 | 992 | | |
| | | 12634 | 35.0 | 50.5 | 4.56 | 15.5 | 857 | 1661 | 12657 | 25726 | | |
| | | 10799 | 34.1 | 62.1 | 22.1 | 20.9 | 4195 | 15063 | 67972 | 244384 | | |
| | | 8666 | 22.9 | 42.1 | 14.2 | 9.90 | 191 | 591 | 3150 | 10351 | | |
| | | 7783 | 5.57 | 44.6 | 2.89 | 14.0 | 21.5 | 147 | 386 | 2631 | | |
| | | 6956 | 86.9 | 0 | 5.42 | 0 | 682 | 0 | 10845 | 0 | | |
| | | 6466 | 90.1 | 0 | 26.0 | 0 | 301 | 0 | 5481 | 0 | | |
| | | 5775 | 14.9 | 31.2 | 22.0 | 24.2 | 8421 | 11672 | 134542 | 185421 | | |
| | | 5723 | 48.3 | 0 | 7.50 | 0 | 2970 | 0 | 49344 | 0 | | |
| | | 5531 | 11.3 | 78.5 | 34.5 | 26.8 | 58.9 | 287 | 1352 | 6499 | | |
| | | 5278 | 115 | 0 | 23.4 | 0 | 117 | 0 | 2274 | 0 | | |
| | | 4782 | 27.9 | 0 | 5.41 | 0 | 714 | 0 | 11617 | 0 | | |
| | | 4606 | 20.4 | 29.7 | 5.94 | 12.6 | 120 | 310 | 1763 | 4579 | | |
| | | 4605 | 12.4 | 78.5 | 18.7 | 41.5 | 356 | 1158 | 5139 | 16799 | | |
| | | 4550 | 13.0 | 75.5 | 15.6 | 35.1 | 235 | 1740 | 3731 | 27345 | | |
| | | 4526 | 113 | 0.787 | 7.10 | 0.199 | 127 | 120 | 1596 | 1514 | | |
| | | 4474 | 28.7 | 63.1 | 9.00 | 26.7 | 2265 | 4634 | 35039 | 72749 | | |

< 1 2 3 4 5 6 7 ... 49 > 1 - 22 of 1063 rows

Supported Metrics

- CPU and GPU Energy Usages and Emissions
- CPU and GPU Usages and Memory Usages
- CPU Hardware/Software/Cache Perf Metrics
- GPU Profiling Metrics (for NVIDIA GPUs)
- IO (Read/Write bytes, bandwidth, requests, errors)
- Network (TCP/UDP, IPv4/IPv6, Ingress and Egress)
- Selected RDMA Metrics (QPs, MRs, requests)

All metrics are *per* cgroup (SLURM job, Openstack VM, k8s pod)

Metrics alone are not enough...

- Usage and perf metrics give a rudimentary idea of how application is behaving
- Need to profile the application to figure out the bottlenecks and hotspots
- Deterministic Profiling: Record call stack & memory stats, investigate and iterate
- Limitations of deterministic profiling:
 - Overhead
 - Hard to recreate problematic scenarios
 - Distributed systems make these only worst

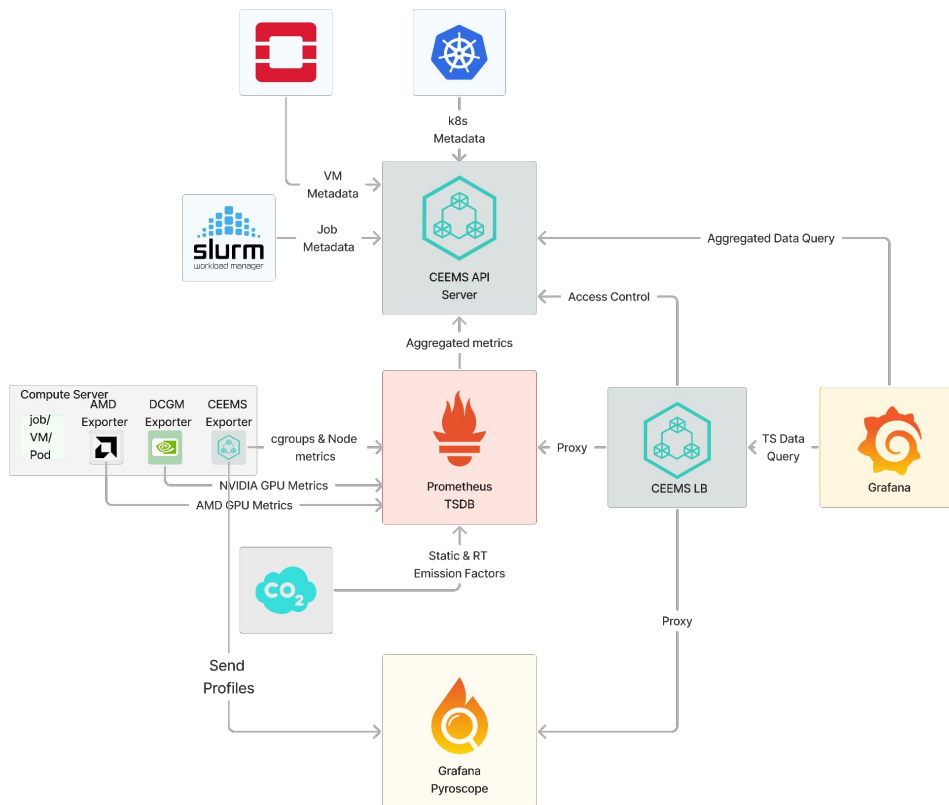
Continuous Profiling

Continuous Profiling

- Continuous profiling: Statistical profiling based on sampling call stack
 - eBPF based
 - No instrumentation needed
 - Very low overhead
 - “Always On” in production
- Works out-of-the-box for compiled languages like C, C++, FORTRAN, Go,...
- Championed by Google and heavily used in cloud native eco-system
- Grafana, Splunk, Datadog, Amazon, Polar signals offer Open Source profilers

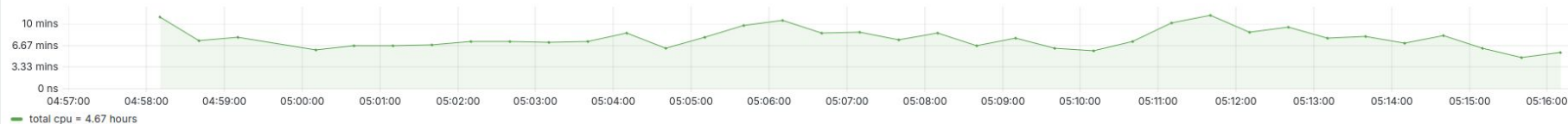
CEEMS Exporter supports Grafana Pyroscope

CEEMS Architecture with Continuous Profiling



Continuous Profiling of SLURM Jobs

Total nanoseconds of CPU time consumed



Flame graph for 450780 (cpu)

Explain Flame Graph

Search...

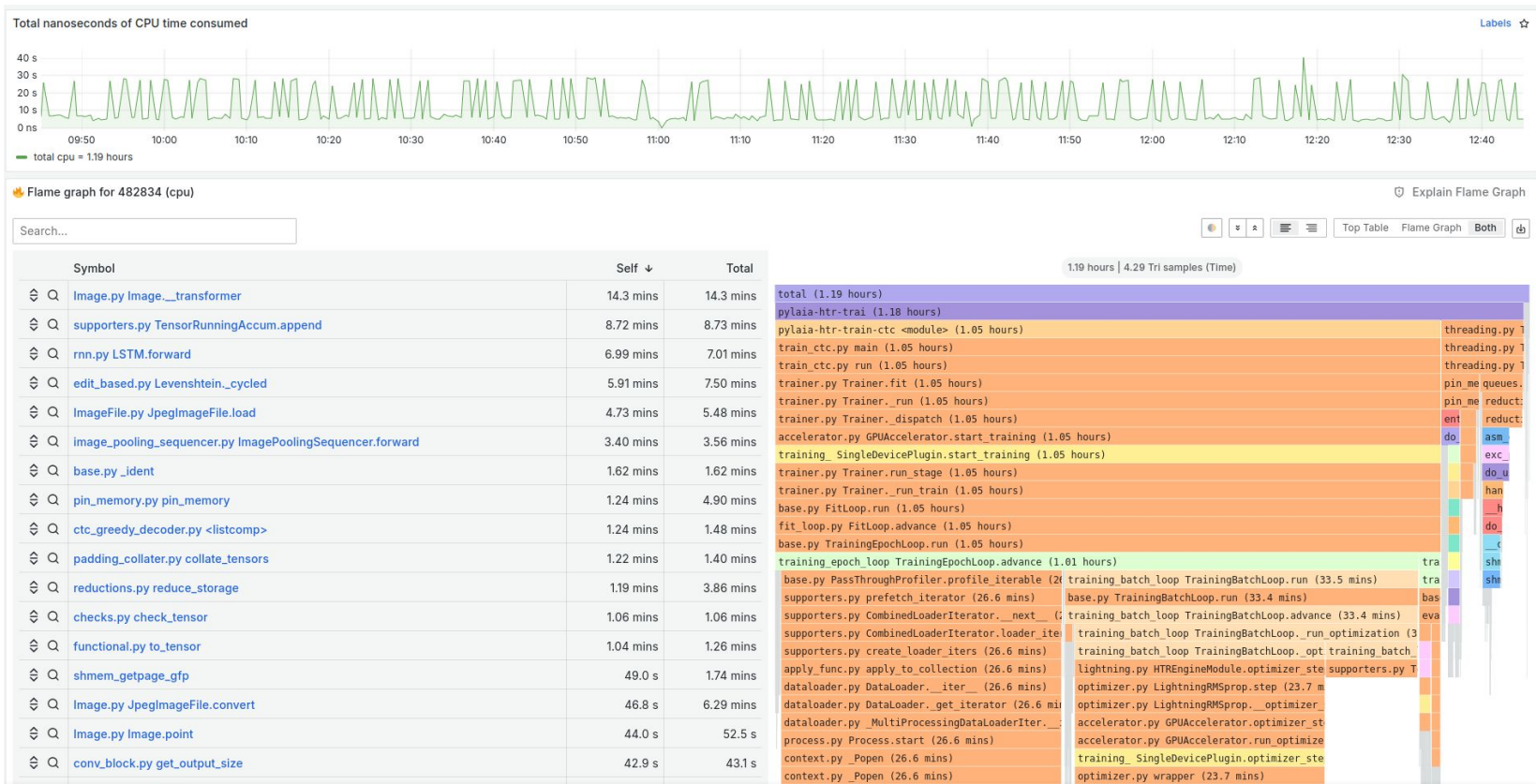
Unresolved symbols

Top Table Flame Graph Both

| Symbol | Self ↓ | Total |
|---|-----------|------------|
| qs_rho_atom_methods_mp_calculate_rho_atom_coeff_ | 32.1 mins | 1.08 hours |
| __kmpc_atomic_float8_add | 29.3 mins | 29.3 mins |
| hfx_energy_potential_mp_integrate_four_center_ | 29.0 mins | 4.99 hours |
| qs_ks_atom_mp_update_ks_atom_ | 27.9 mins | 1.05 hours |
| hfx_energy_potential_mp_update_fock_matrix_ | 18.9 mins | 22.2 mins |
| hfx_pair_list_methods_mp_build_pgf_product_list_ | 16.7 mins | 25.8 mins |
| MPIDI_CH3L_Progress | 13.6 mins | 13.6 mins |
| hfx_pair_list_methods_mp_build_pair_list_pgf_ | 7.68 mins | 7.68 mins |
| ortho_cxyz_to_grid | 7.05 mins | 7.15 mins |
| f64xsubf128 | 6.32 mins | 6.32 mins |
| qs_dispersion_pairpot_mp_calculate_dispersion_pairpot_ | 6.16 mins | 7.32 mins |
| f_ldnint_val | 4.87 mins | 4.87 mins |
| hfx_compression_methods_mp_hfx_get_mult_cache_elements_ | 4.24 mins | 4.69 mins |
| t_c_g0_mp_pd2val_ | 3.86 mins | 3.86 mins |
| copy_user_enhanced_fast_string | 3.78 mins | 4.02 mins |

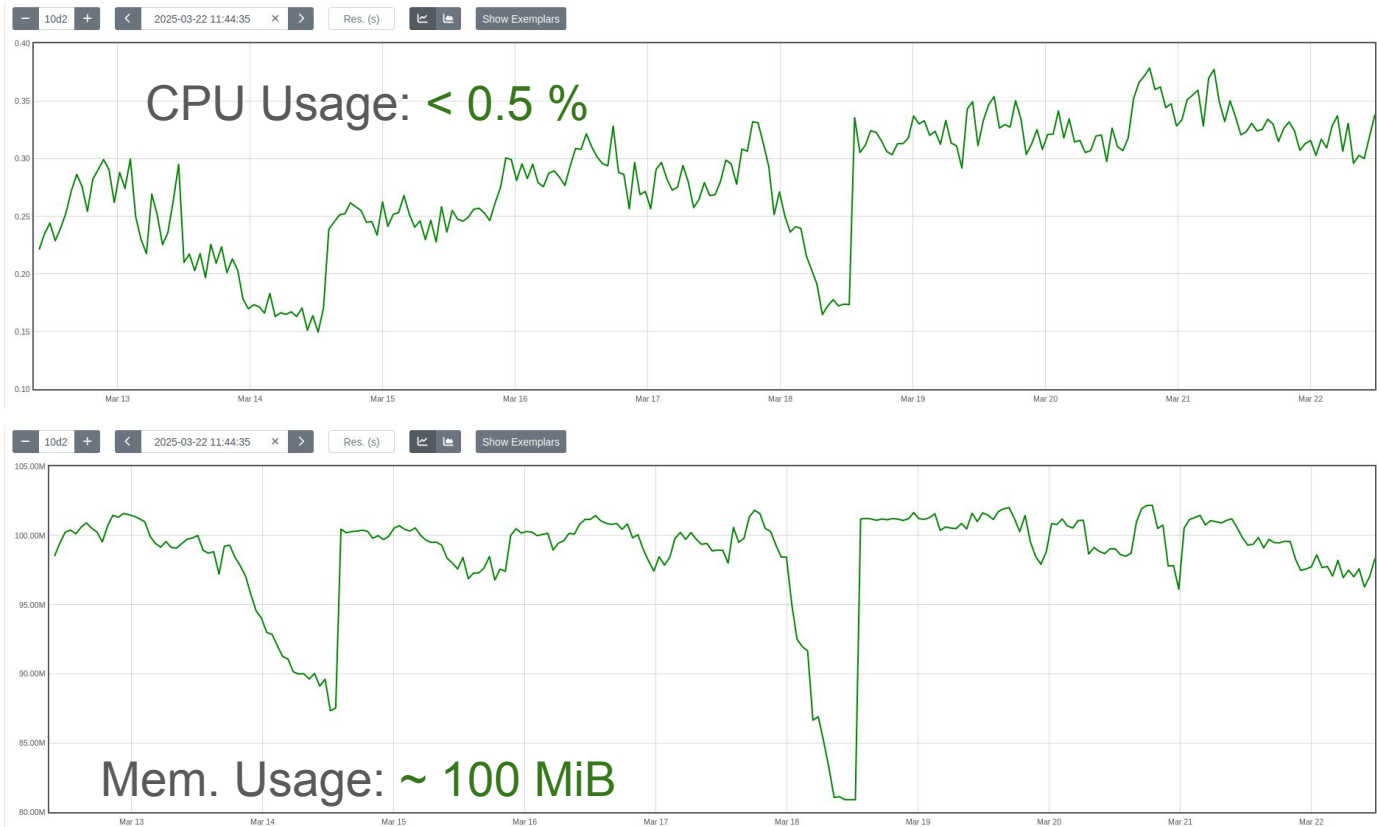


Continuous Profiling of SLURM Jobs



Exporter Overhead

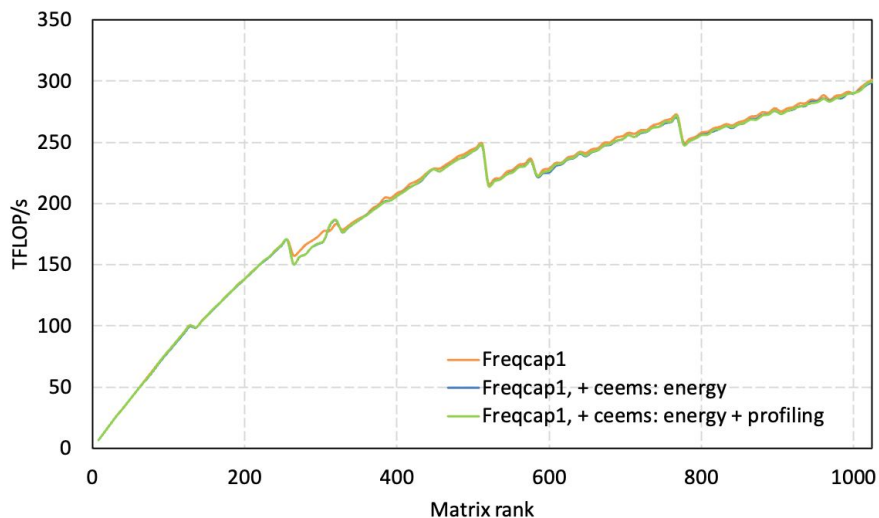
CPU and Memory Usage averaged over ~360 nodes.



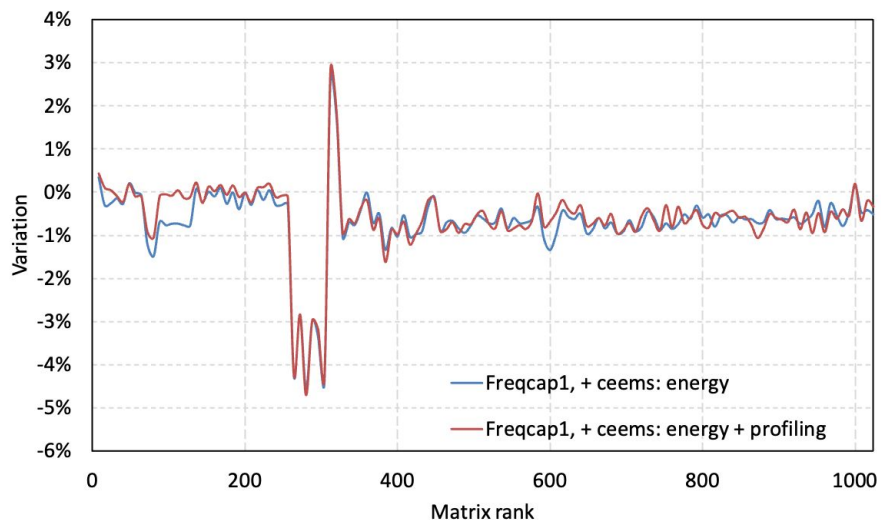
Benchmarks

Randomised SVD with varying matrix size

Randsvd performance with energy and profiling

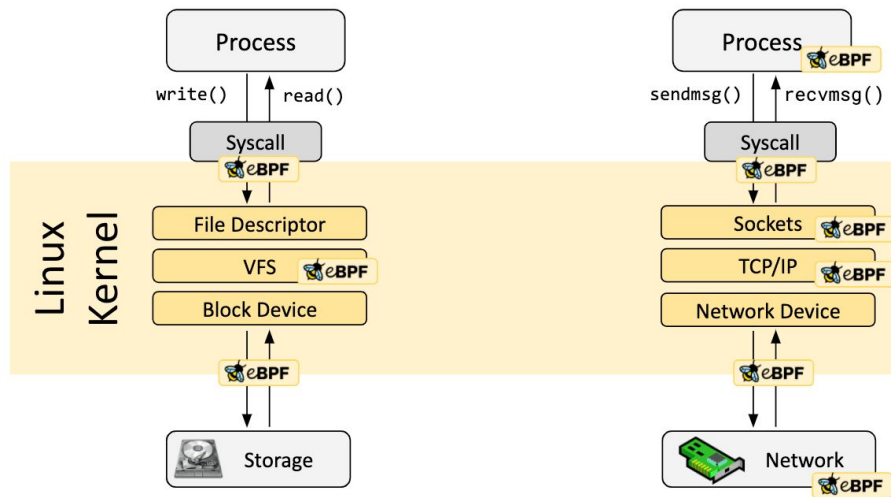


Randsvd performance variation with energy and profiling



Technical Details

- 100 % Go (except the bpf programs which are in C)
- CEEMS apps are **Capability Aware**
- Uses eBPF for IO and Network metrics



Testing & CI

Battle tested in CI. > 75 % of unit test coverage.

More than 60 e2e tests

| | |
|---------|---|
| CI/CD | CI passing PASSED Coverage 75.6% |
| Docs | docs passing go reference |
| Package | release v0.7.0 |
| Meta | license GPL-3.0 go report A+ code style gofmt |

← CI

✓ Minor improvements in power usage collectors #747

Re-run all jobs ...

Summary

Jobs

- ✓ test-lint
- ✓ test-unit
- ✓ test-e2e
- ✓ build
- ✓ packaging
- ✓ docker
- ✓ quay
- ✓ docker-test
- ✓ quay-test

Run details

Usage

Workflow file

Triggered via pull request 26 minutes ago

mahendrapaipuri opened #330 [power_coll_improvs](#)

Status

Success

Total duration

17m 46s

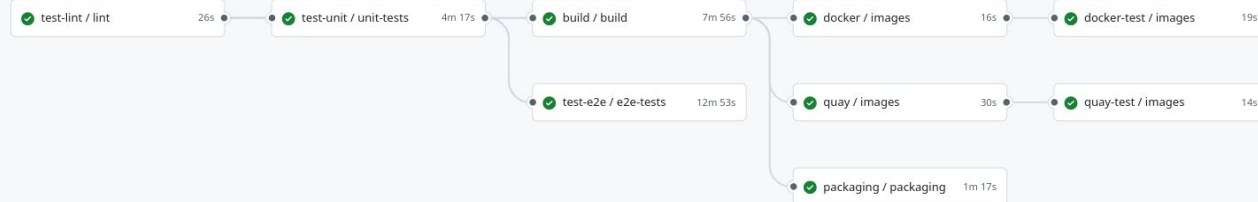
Artifacts

2

ci.yml

on: pull_request

Take a New Screenshot



Packaging

Pre-compiled binaries, RPM/DEB packages and OCI images are available for different archs.

Helm chart is also available

Repository Tags

| TAG | LAST MODIFIED | SECURITY SCAN | SIZE | EXPIRES | MANIFEST |
|----------------|---------------|-----------------------|---------|---------|----------------------|
| main | 2 hours ago | See Child Manifests | N/A | Never | SHA256: 9852a3827d15 |
| linux on amd64 | | 1 Unknown - 1 fixable | 81.8 MB | | SHA256: 45948a4c4d32 |
| linux on arm64 | | 1 Unknown - 1 fixable | 77.3 MB | | SHA256: 7d4121f49d81 |
| latest | 6 days ago | See Child Manifests | N/A | Never | SHA256: 939d384aa413 |
| linux on amd64 | | 1 Unknown - 1 fixable | 81.8 MB | | SHA256: 625d85c1ffff |
| linux on arm64 | | 1 Unknown - 1 fixable | 77.3 MB | | SHA256: e83d4d84d023 |
| v0.7.0 | 6 days ago | See Child Manifests | N/A | Never | SHA256: 939d384aa413 |
| linux on amd64 | | 1 Unknown - 1 fixable | 81.8 MB | | SHA256: 625d85c1ffff |
| linux on arm64 | | 1 Unknown - 1 fixable | 77.3 MB | | SHA256: e83d4d84d023 |

| | | |
|--|---------|-----------|
| cacct-0.7.0-linux-amd64.deb | 9.47 MB | last week |
| cacct-0.7.0-linux-amd64.rpm | 9.66 MB | last week |
| cacct-0.7.0-linux-arm64.deb | 8.81 MB | last week |
| cacct-0.7.0-linux-arm64.rpm | 8.95 MB | last week |
| ceems-0.7.0.linux-386.tar.gz | 74.2 MB | last week |
| ceems-0.7.0.linux-amd64.tar.gz | 77.6 MB | last week |
| ceems-0.7.0.linux-arm64.tar.gz | 72.6 MB | last week |
| ceems-0.7.0.linux-mips.tar.gz | 71.9 MB | last week |
| ceems-0.7.0.linux-mips64.tar.gz | 71.7 MB | last week |
| ceems-0.7.0.linux-mips64le.tar.gz | 69.9 MB | last week |
| ceems-0.7.0.linux-mipsle.tar.gz | 70.4 MB | last week |
| ceems-0.7.0.linux-ppc64le.tar.gz | 73.7 MB | last week |
| ceems-0.7.0.linux-riscv64.tar.gz | 73.1 MB | last week |
| ceems_api_server-0.7.0-linux-amd64.deb | 26.9 MB | last week |
| ceems_api_server-0.7.0-linux-amd64.rpm | 27.4 MB | last week |
| ceems_api_server-0.7.0-linux-arm64.deb | 25.3 MB | last week |
| ceems_api_server-0.7.0-linux-arm64.rpm | 25.8 MB | last week |
| ceems_exporter-0.7.0-linux-amd64.deb | 15 MB | last week |
| ceems_exporter-0.7.0-linux-amd64.rpm | 15.4 MB | last week |
| ceems_exporter-0.7.0-linux-arm64.deb | 14 MB | last week |
| ceems_exporter-0.7.0-linux-arm64.rpm | 14.3 MB | last week |
| ceems_lb-0.7.0-linux-amd64.deb | 17.9 MB | last week |
| ceems_lb-0.7.0-linux-amd64.rpm | 18.3 MB | last week |
| ceems_lb-0.7.0-linux-arm64.deb | 16.9 MB | last week |
| ceems_lb-0.7.0-linux-arm64.rpm | 17.3 MB | last week |
| redfish_proxy-0.7.0-linux-amd64.deb | 9.11 MB | last week |
| redfish_proxy-0.7.0-linux-amd64.rpm | 9.28 MB | last week |
| redfish_proxy-0.7.0-linux-arm64.deb | 8.49 MB | last week |
| redfish_proxy-0.7.0-linux-arm64.rpm | 8.63 MB | last week |

Final Remarks

- CEEMS provide a “complete” monitoring solution
- Running on Jean Zay since ~ 1,5 years with a scrape frequency of 10s
- Currently working on support for cloud VMs using [SPEC](#) Power database.
- A [demo instance](#) is available to play around

Grid5000/SLICES-FR platform has been of immense use
during the development of this stack.
A huge thanks to Grid5000/SLICES-FR team.



Demo

Thank you

Resources:

- [CEEMS GitHub Repo](#)
- [CEEMS Docs](#)
- [CEEMS API Server Docs](#)
- [CEEMS Helm Chart](#)
- [CEEMS Exporter Metrics List](#)
- [CEEMS Demo](#)