CEEMS: A Resource Manager Agnostic Energy & Performance Monitoring Stack

Mahendra Paipuri

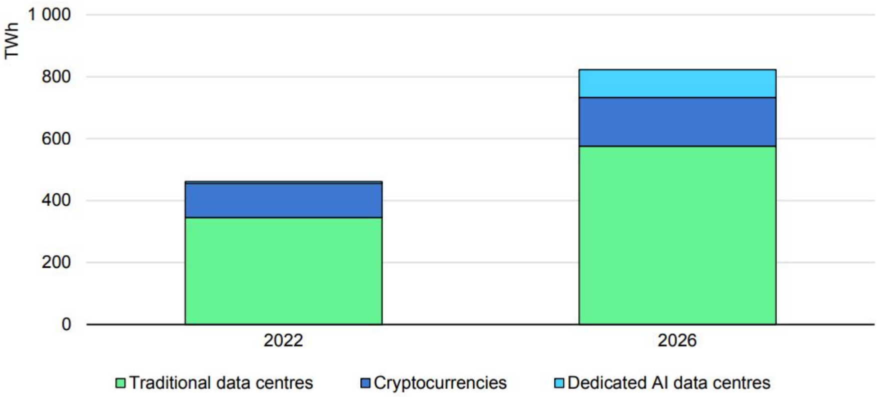
IDRIS, CNRS

27th March 2025

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

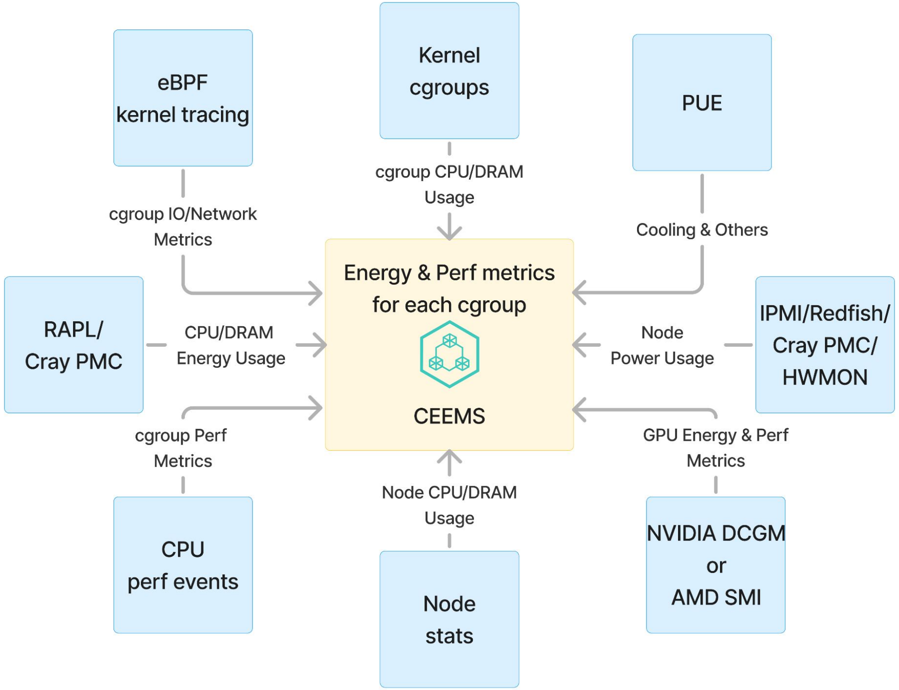
Compute Energy & Emissions Monitoring Stack (CEEMS)

CEEMS

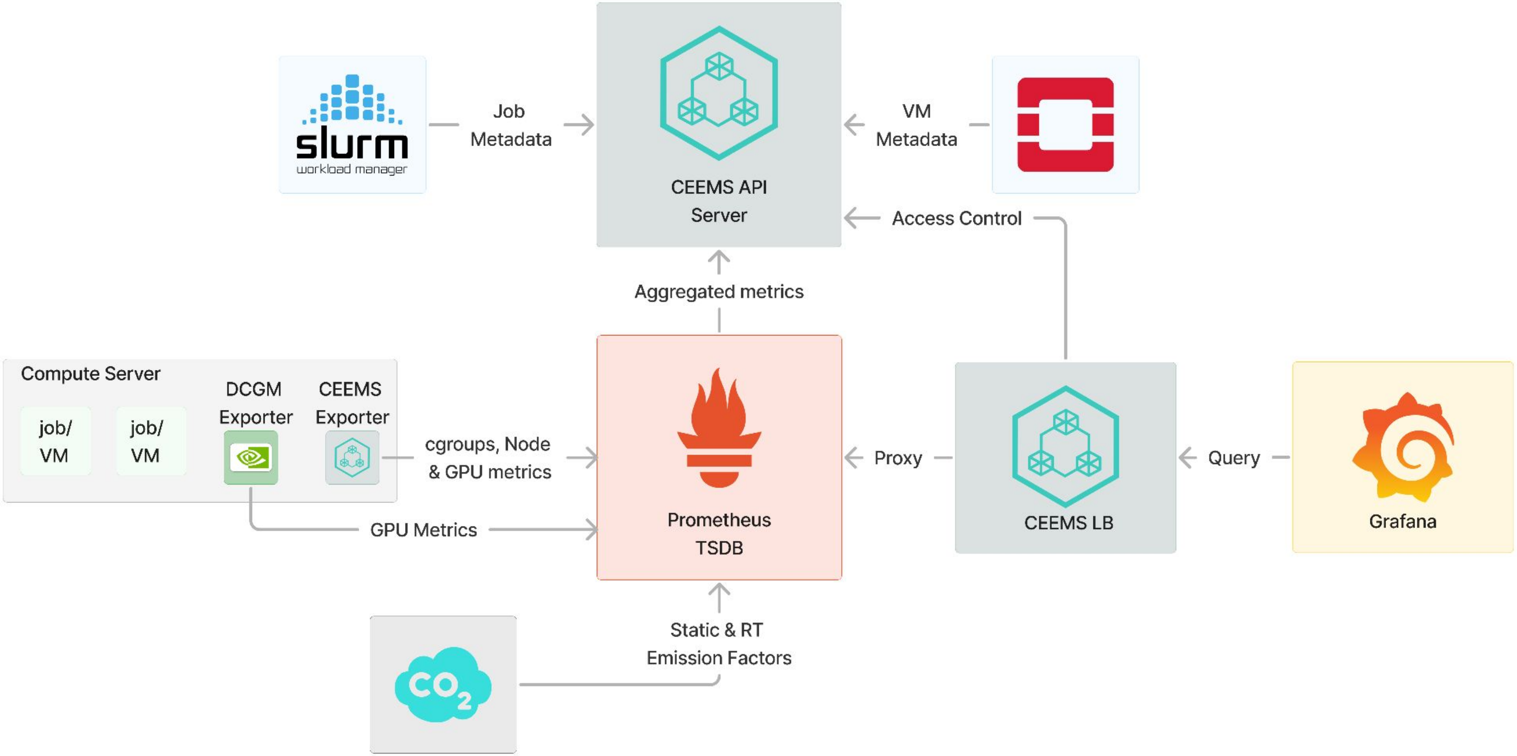
- Started as a tool to estimate energy consumption and equivalent emissions for HPC workloads
- Extended the stack to support Openstack. Currently adding k8s support.
- A system level stack
- cgroups, perf subsystem, eBPF are at the heart of CEEMS
- Based on CNCF Opensource components. Prometheus as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



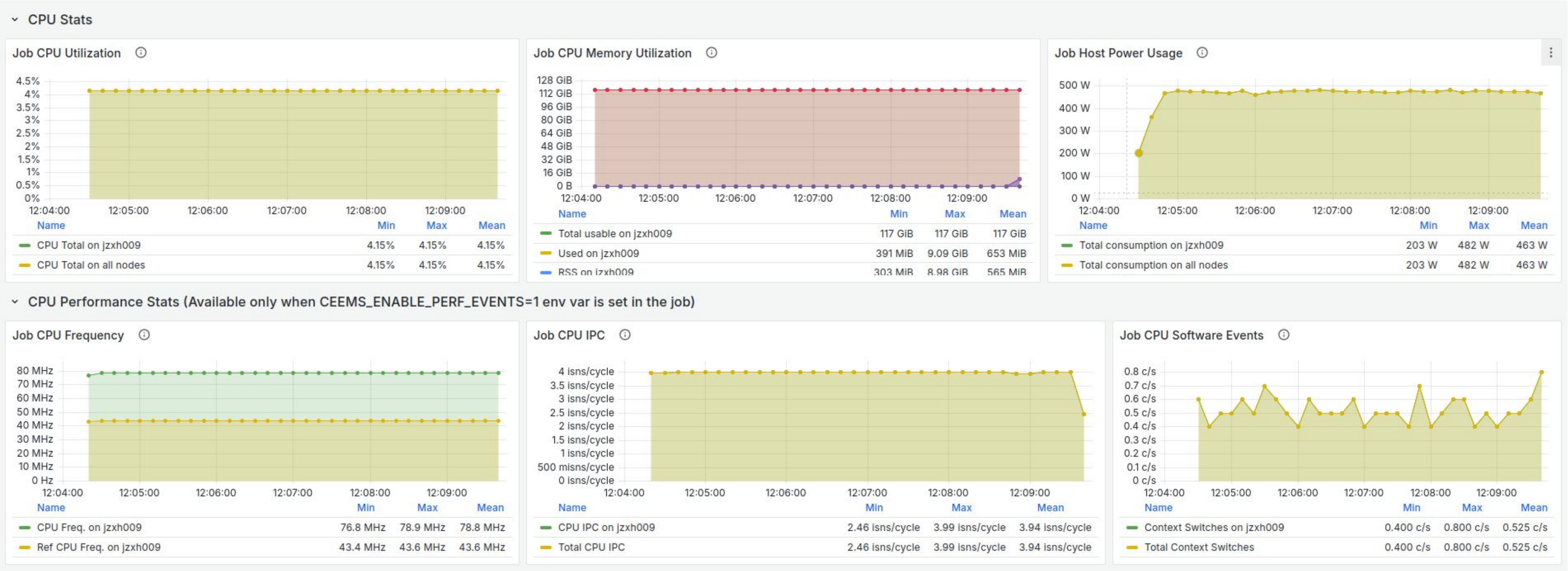
CEEMS Architecture



Features

- Monitors energy, performance, IO and network metrics for different types of resource managers
- Supports different energy sources like RAPL, HWMON, Cray's PM Counters and BMC via IPMI or Redfish
- Supports NVIDIA (MIG and vGPU) and AMD 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

Name	Min	Max	Mean
CPU Total on jzxh009	4.15%	4.15%	4.15%
CPU Total on all nodes	4.15%	4.15%	4.15%

Job CPU Memory Utilization

Name	Min	Max	Mean
Total usable on jzxh009	117 GiB	117 GiB	117 GiB
Used on jzxh009	391 MiB	9.09 GiB	653 MiB
RSS on jzxh009	303 MiB	8.98 GiB	565 MiB

Job Host Power Usage

Name	Min	Max	Mean
Total consumption on jzxh009	203 W	482 W	463 W
Total consumption on all nodes	203 W	482 W	463 W

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

Job CPU Frequency

Name	Min	Max	Mean
CPU Freq. on jzxh009	76.8 MHz	78.9 MHz	78.8 MHz
Ref CPU Freq. on jzxh009	43.4 MHz	43.6 MHz	43.6 MHz

Job CPU IPC

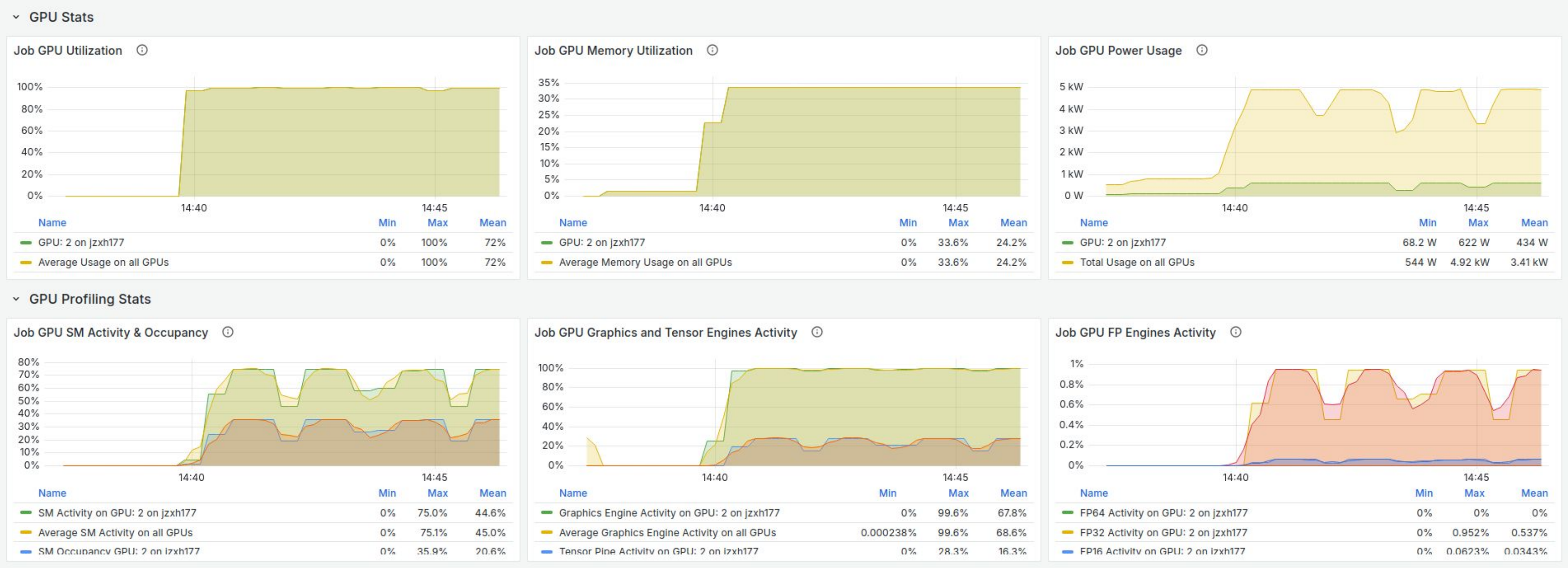
Name	Min	Max	Mean
CPU IPC on jzxh009	2.46 isns/cycle	3.99 isns/cycle	3.94 isns/cycle
Total CPU IPC	2.46 isns/cycle	3.99 isns/cycle	3.94 isns/cycle

Job CPU Software Events

Name	Min	Max	Mean
Context Switches on jzxh009	0.400 c/s	0.800 c/s	0.525 c/s
Total Context Switches	0.400 c/s	0.800 c/s	0.525 c/s

CPU Stats

User Dashboards



GPU Stats

Job GPU Utilization

Name	Min	Max	Mean
🟩 GPU: 2 on jzxh177	0%	100%	72%
🟨 Average Usage on all GPUs	0%	100%	72%

Job GPU Memory Utilization

Name	Min	Max	Mean
🟩 GPU: 2 on jzxh177	0%	33.6%	24.2%
🟨 Average Memory Usage on all GPUs	0%	33.6%	24.2%

Job GPU Power Usage

Name	Min	Max	Mean
🟩 GPU: 2 on jzxh177	68.2 W	622 W	434 W
🟨 Total Usage on all GPUs	544 W	4.92 kW	3.41 kW

GPU Profiling Stats

Job GPU SM Activity & Occupancy

Name	Min	Max	Mean
🟩 SM Activity on GPU: 2 on jzxh177	0%	75.0%	44.6%
🟨 Average SM Activity on all GPUs	0%	75.1%	45.0%
🟦 SM Occupancy GPU: 2 on jzxh177	0%	35.9%	20.6%

Job GPU Graphics and Tensor Engines Activity

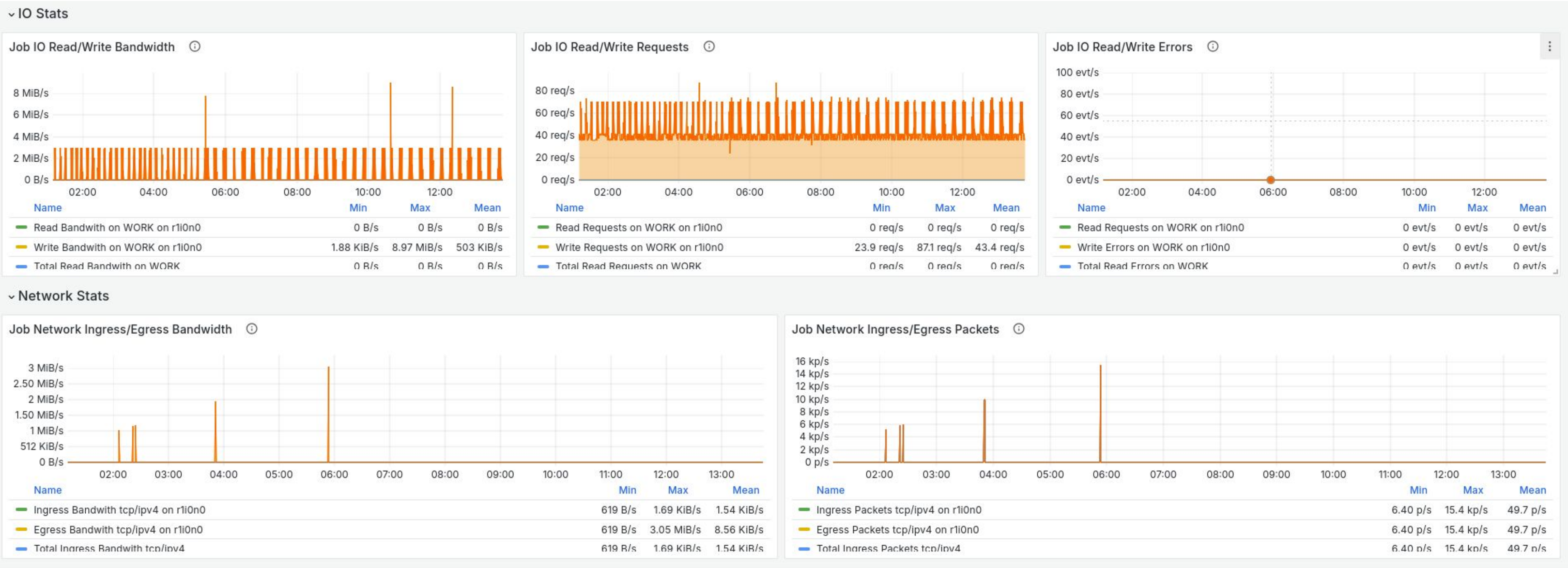
Name	Min	Max	Mean
🟩 Graphics Engine Activity on GPU: 2 on jzxh177	0%	99.6%	67.8%
🟨 Average Graphics Engine Activity on all GPUs	0.000238%	99.6%	68.6%
🟦 Tensor Pipe Activity on GPU: 2 on jzxh177	0%	28.3%	16.3%

Job GPU FP Engines Activity

Name	Min	Max	Mean
🟩 FP64 Activity on GPU: 2 on jzxh177	0%	0%	0%
🟨 FP32 Activity on GPU: 2 on jzxh177	0%	0.952%	0.537%
🟦 FP16 Activity on GPU: 2 on jzxh177	0%	0.0623%	0.0343%

GPU Stats

User Dashboards



⌄ IO Stats

Job IO Read/Write Bandwidth ⓘ

Name	Min	Max	Mean
🟩 Read Bandwith on WORK on r1i0n0	0 B/s	0 B/s	0 B/s
🟨 Write Bandwith on WORK on r1i0n0	1.88 KiB/s	8.97 MiB/s	503 KiB/s
🟦 Total Read Bandwith 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	87.1 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 ⓘ

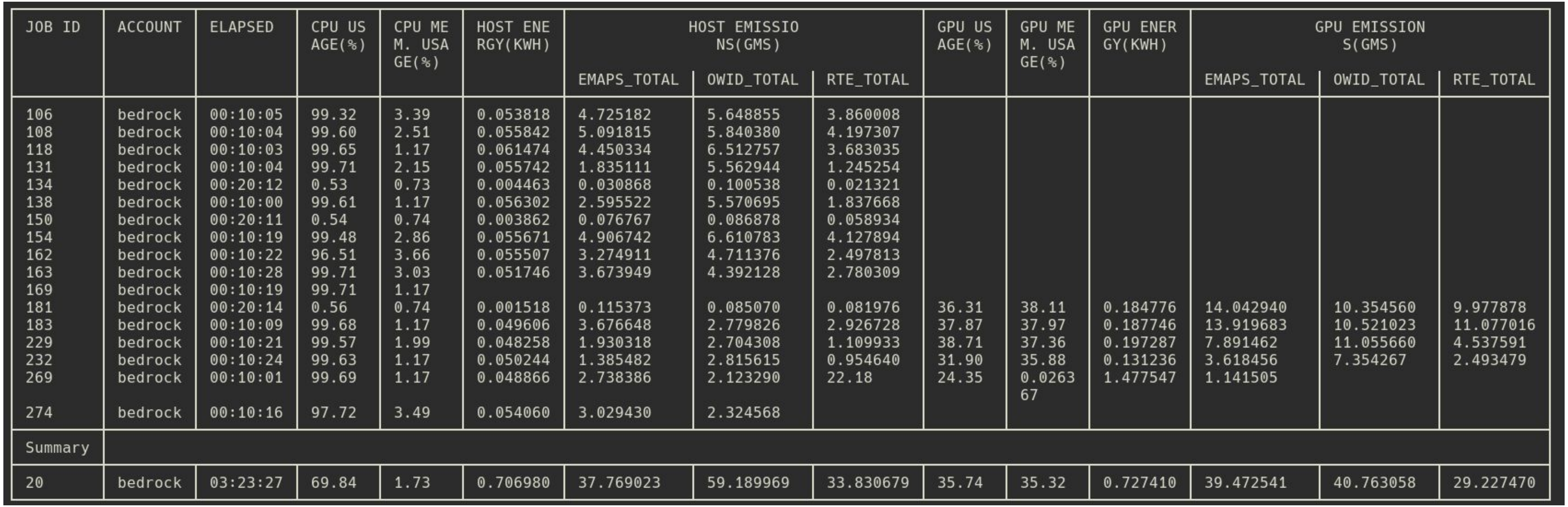
Name	Min	Max	Mean
🟩 Ingress Bandwith tcp/ipv4 on r1i0n0	619 B/s	1.69 KiB/s	1.54 KiB/s
🟨 Egress Bandwith tcp/ipv4 on r1i0n0	619 B/s	3.05 MiB/s	8.56 KiB/s
🟦 Total Ingress Bandwith tcp/ipv4	619 B/s	1.69 KiB/s	1.54 KiB/s

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 p/s	15.4 kp/s	49.7 p/s

IO/Network Stats

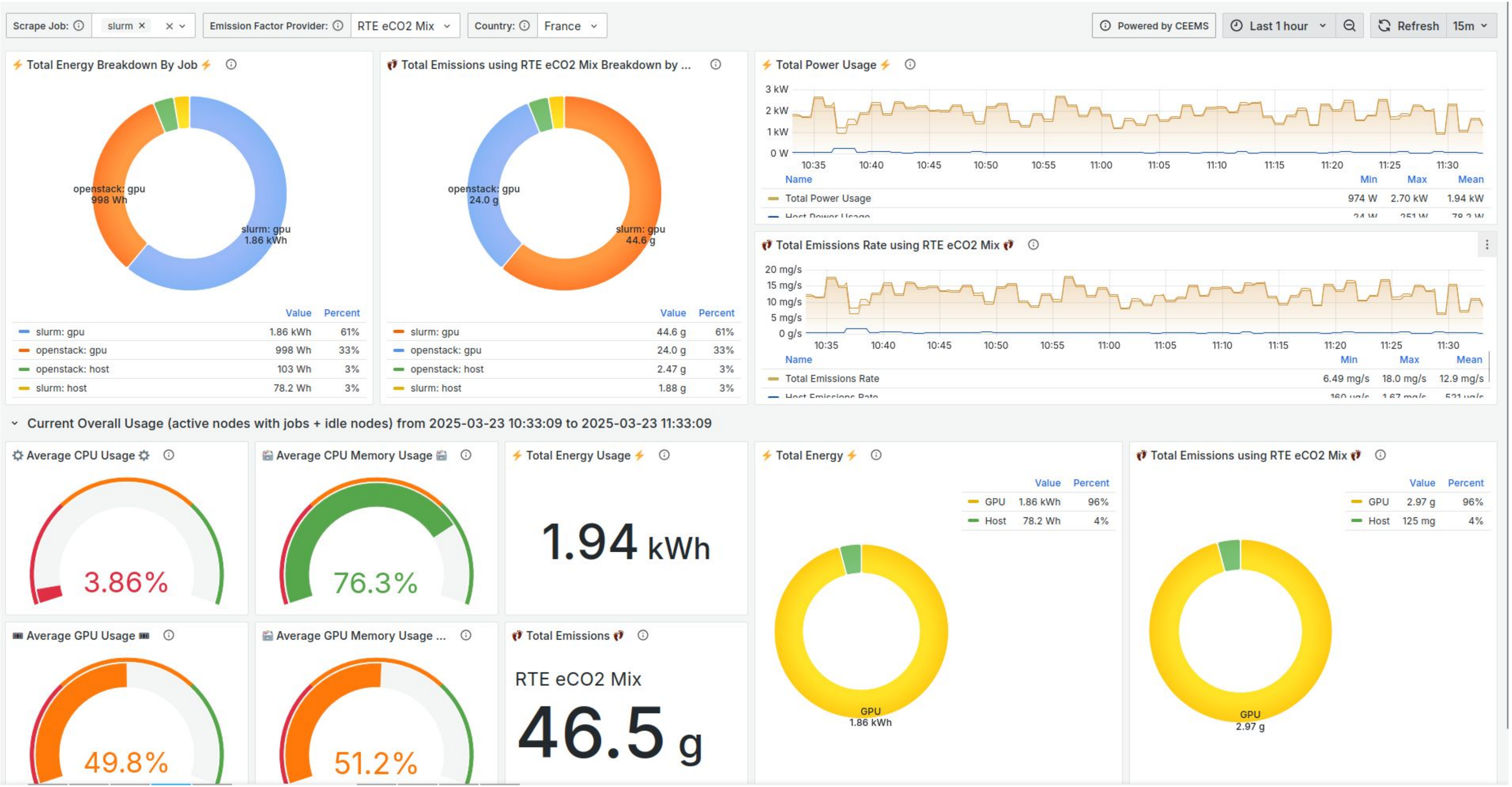
CLI Client Tool



JOB ID	ACCOUNT	ELAPSED	CPU USAGE(%)	CPU MEM. USAGE(%)	HOST ENERGY(KWH)	HOST EMISSIONS(GMS)			GPU USAGE(%)	GPU MEM. USAGE(%)	GPU ENERGY(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



⚡ Total Energy Breakdown By Job ⚡ ⓘ

	Value	Percent
slurm: gpu	1.86 kWh	61%
openstack: gpu	998 Wh	33%
openstack: host	103 Wh	3%
slurm: host	78.2 Wh	3%

Total Emissions using RTE eCO2 Mix Breakdown by ... ⓘ

	Value	Percent
slurm: gpu	44.6 g	61%
openstack: gpu	24.0 g	33%
openstack: host	2.47 g	3%
slurm: host	1.88 g	3%

⚡ Total Power Usage ⚡ ⓘ

Name	Min	Max	Mean
Total Power Usage	974 W	2.70 kW	1.94 kW
Host Power Usage	24 W	251 W	78.2 W

Total Emissions Rate using RTE eCO2 Mix ⓘ

Name	Min	Max	Mean
Total Emissions Rate	6.49 mg/s	18.0 mg/s	12.9 mg/s
Host Emissions Rate	160 µg/s	1.67 mg/s	521 µg/s

Current Overall Usage (active nodes with jobs + idle nodes) from 2025-03-23 10:33:09 to 2025-03-23 11:33:09

Average CPU Usage ⓘ
3.86%

Average CPU Memory Usage ⓘ
76.3%

⚡ Total Energy Usage ⚡ ⓘ
1.94 kWh

⚡ Total Energy ⚡ ⓘ

	Value	Percent
GPU	1.86 kWh	96%
Host	78.2 Wh	4%

Total Emissions using RTE eCO2 Mix ⓘ

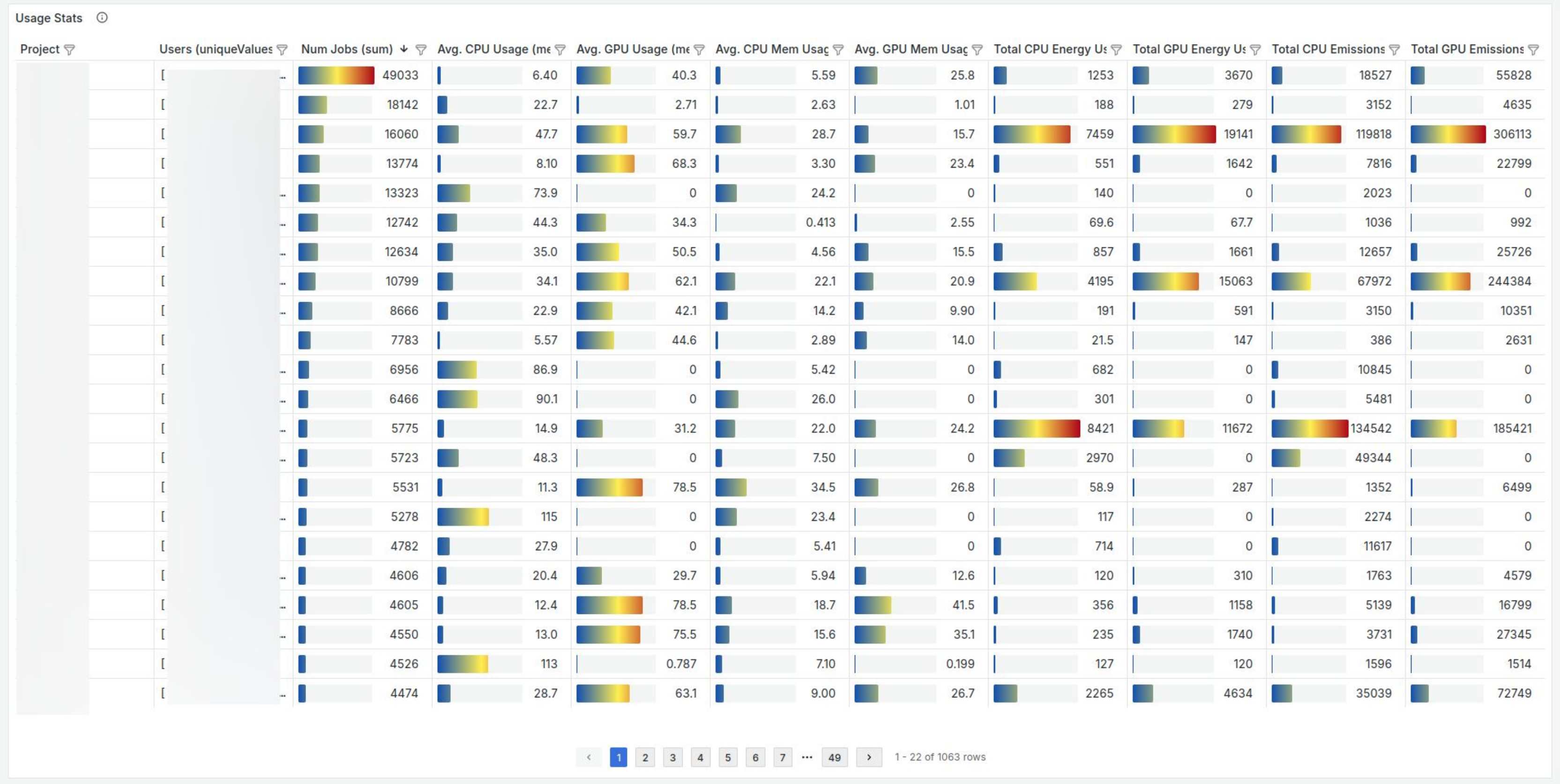
	Value	Percent
GPU	2.97 g	96%
Host	125 mg	4%

Average GPU Usage ⓘ
49.8%

Average GPU Memory Usage ... ⓘ
51.2%

Total Emissions ⓘ
RTE eCO2 Mix
46.5 g

Cluster Dashboards - Operators



Usage Stats ⓘ

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

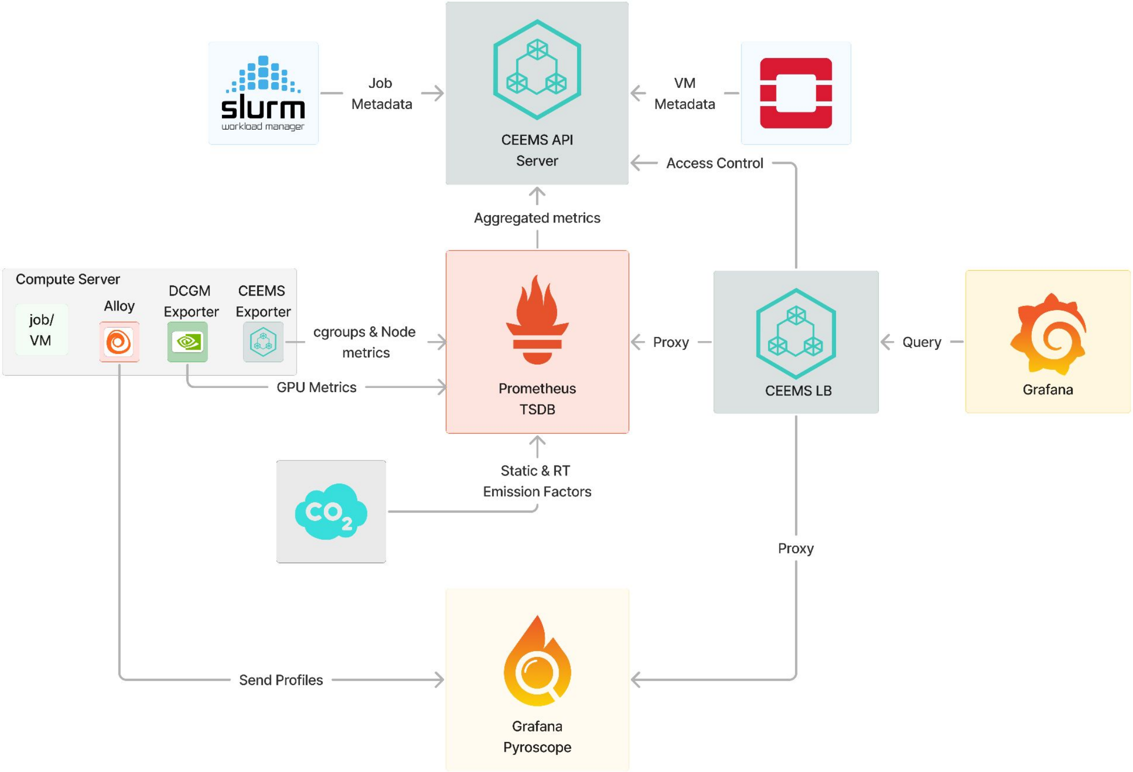
Continuous Profiling

- Continuous profiling: Statistical profiling based on sampling call stack
  - eBPF based
  - No instrumentation needed
  - Very low overhead
  - “Always On” in production

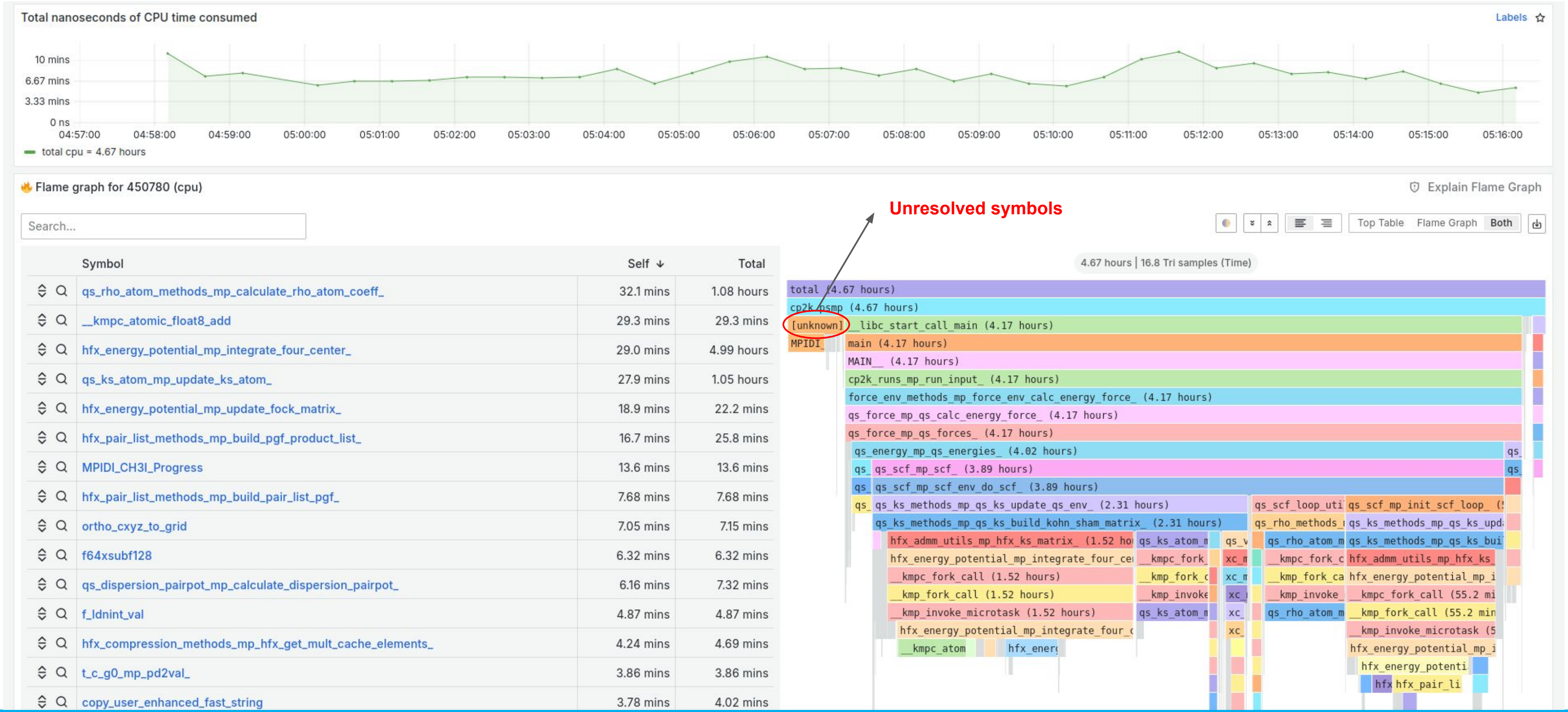
- Works out-of-the-box for compiled languages like C, C++, FORTRAN, Go,...
- Championed by Google and heavily used in cloud native eco-system
- Grafana, Splunk, Datadog, Amazon, Polar signals offer Open Source profilers

CEEMS Exporter supports Grafana Alloy and Pyroscope

CEEMS Architecture with Continuous Profiling



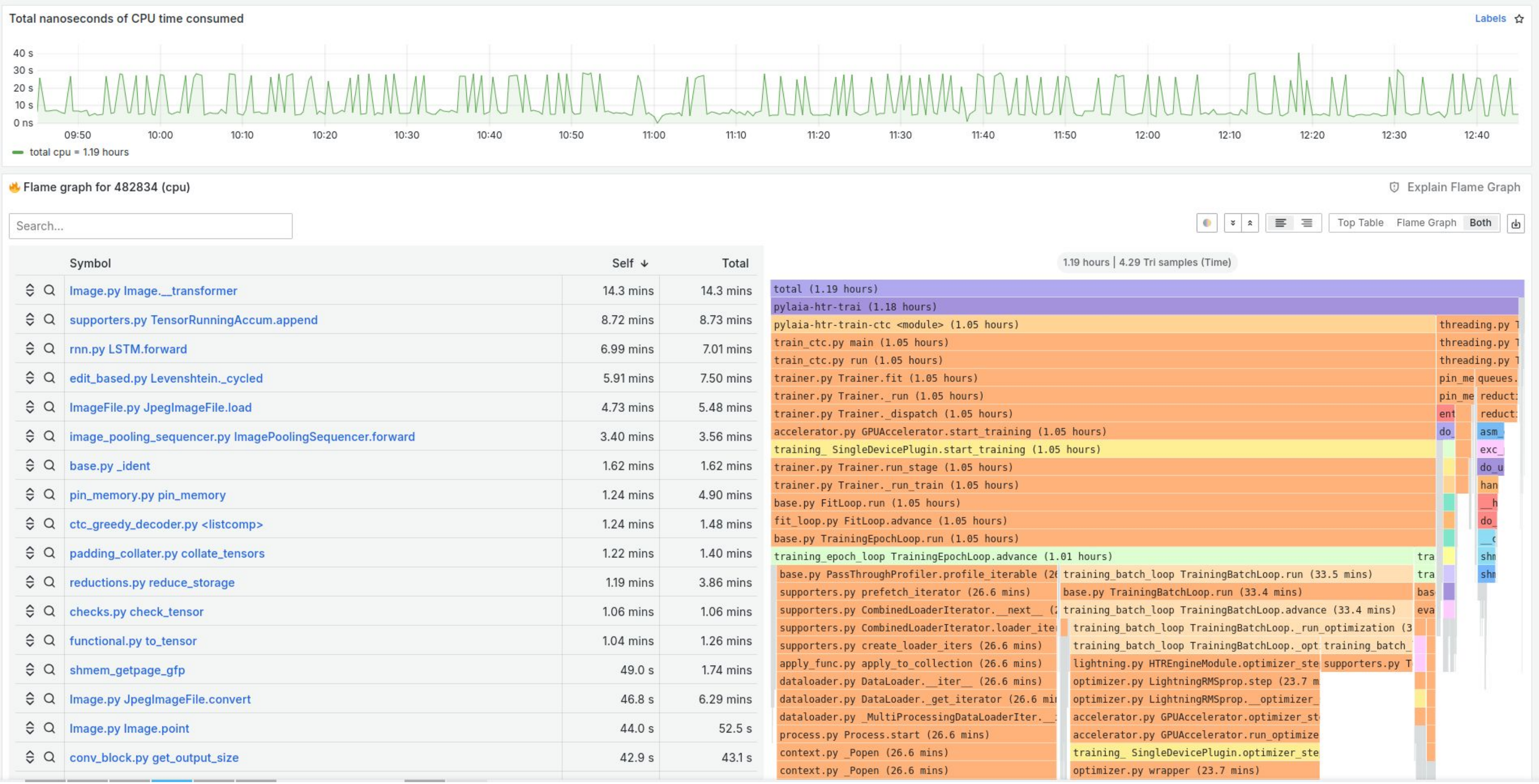
Continuous Profiling of SLURM Jobs


Total nanoseconds of CPU time consumed
Labels
10 mins
6.67 mins
3.33 mins
0 ns
04:57:00
04:58:00
04:59:00
05:00:00
05:01:00
05:02:00
05:03:00
05:04:00
05:05:00
05:06:00
05:07:00
05:08:00
05:09:00
05:10:00
05:11:00
05:12:00
05:13:00
05:14:00
05:15:00
05:16:00
total cpu = 4.67 hours
Flame graph for 450780 (cpu)
Explain Flame Graph
Search...
Top Table
Flame Graph
Both
4.67 hours | 16.8 Tri samples (Time)
total (4.67 hours)
cp2k.psmp (4.67 hours)
[unknown]
__libc_start_call_main (4.17 hours)
MPIDI_
main (4.17 hours)
MAIN__ (4.17 hours)
cp2k_runs_mp_run_input_ (4.17 hours)
force_env_methods_mp_force_env_calc_energy_force_ (4.17 hours)
qs_force_mp_qs_calc_energy_force_ (4.17 hours)
qs_force_mp_qs_forces_ (4.17 hours)
qs_energy_mp_qs_energies_ (4.02 hours)
qs_scf_mp_scf_ (3.89 hours)
qs_scf_mp_scf_env_do_scf_ (3.89 hours)
qs_ks_methods_mp_qs_ks_update_qs_env_ (2.31 hours)
qs_ks_methods_mp_qs_ks_build_kohn_sham_matrix_ (2.31 hours)
hfx_admm_utils_mp_hfx_ks_matrix_ (1.52 ho
hfx_energy_potential_mp_integrate_four_cei
__kmpc_fork_call (1.52 hours)
__kmp_fork_call (1.52 hours)
__kmp_invoke_microtask (1.52 hours)
hfx_energy_potential_mp_integrate_four_c
__kmpc_atom
hfx_energ
qs_scf_loop_uti
qs_scf_mp_init_scf_loop_
qs_rho_methods_
qs_ks_methods_mp_qs_ks_upd
qs_rho_atom_m
qs_ks_methods_mp_qs_ks_bui
__kmpc_fork_c
hfx_admm_utils_mp_hfx_ks_
__kmp_fork_ca
hfx_energy_potential_mp_i
__kmp_invoke
__kmpc_fork_call (55.2 mi
__kmp_fork_call (55.2 min
__kmp_invoke_microtask (5
hfx_energy_potential_mp_i
hfx_energy_potenti
hfx hfx_pair_li
qs_ks_atom_m
__kmpc_fork
__kmp_fork_c
__kmp_invoke
xc_
qs_

Unresolved symbols

	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_CH3I_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

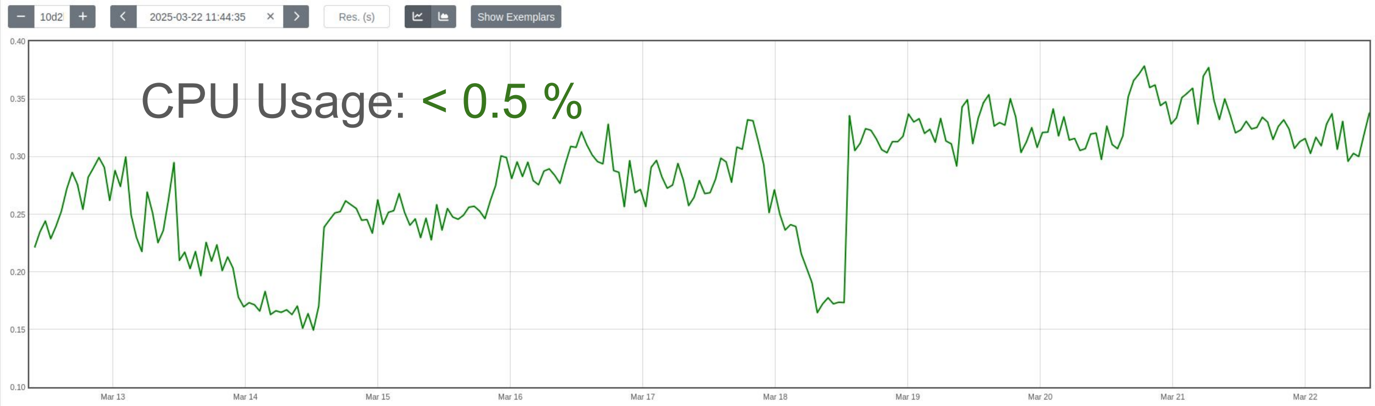
Continuous Profiling of SLURM Jobs


Total nanoseconds of CPU time consumed
Labels
40 s
30 s
20 s
10 s
0 ns
09:50
10:00
10:10
10:20
10:30
10:40
10:50
11:00
11:10
11:20
11:30
11:40
11:50
12:00
12:10
12:20
12:30
12:40
total cpu = 1.19 hours
Flame graph for 482834 (cpu)
Explain Flame Graph
Search...
Top Table
Flame Graph
Both
1.19 hours | 4.29 Tri samples (Time)
total (1.19 hours)
pylaia-htr-trai (1.18 hours)
pylaia-htr-train-ctc <module> (1.05 hours)
train_ctc.py main (1.05 hours)
train_ctc.py run (1.05 hours)
trainer.py Trainer.fit (1.05 hours)
trainer.py Trainer._run (1.05 hours)
trainer.py Trainer._dispatch (1.05 hours)
accelerator.py GPUAccelerator.start_training (1.05 hours)
training_ SingleDevicePlugin.start_training (1.05 hours)
trainer.py Trainer.run_stage (1.05 hours)
trainer.py Trainer._run_train (1.05 hours)
base.py FitLoop.run (1.05 hours)
fit_loop.py FitLoop.advance (1.05 hours)
base.py TrainingEpochLoop.run (1.05 hours)
training_epoch_loop TrainingEpochLoop.advance (1.01 hours)
base.py PassThroughProfiler.profile_iterable (26
supporters.py prefetch_iterator (26.6 mins)
supporters.py CombinedLoaderIterator.__next__ (2
supporters.py CombinedLoaderIterator.loader_ite
supporters.py create_loader_iters (26.6 mins)
apply_func.py apply_to_collection (26.6 mins)
dataloader.py DataLoader.__iter__ (26.6 mins)
dataloader.py DataLoader._get_iterator (26.6 mi
dataloader.py _MultiProcessingDataLoaderIter.__
process.py Process.start (26.6 mins)
context.py _Popen (26.6 mins)
context.py _Popen (26.6 mins)
training_batch_loop TrainingBatchLoop.run (33.5 mins)
base.py TrainingBatchLoop.run (33.4 mins)
training_batch_loop TrainingBatchLoop.advance (33.4 mins)
training_batch_loop TrainingBatchLoop._run_optimization (3
training_batch_loop TrainingBatchLoop._opt
lightning.py HTREngineModule.optimizer_ste
optimizer.py LightningRMSprop.step (23.7 m
optimizer.py LightningRMSprop.__optimizer_
accelerator.py GPUAccelerator.optimizer_st
accelerator.py GPUAccelerator.run_optimize
training_ SingleDevicePlugin.optimizer_ste
optimizer.py wrapper (23.7 mins)
training_batch_
supporters.py T
tra
tra
bas
eva
threading.py T
threading.py T
threading.py T
pin_me
pin_me
ent
do_
queues.
reduct:
reduct:
asm_
exc_
do_u
han
__h
do_
__c
shn
shn

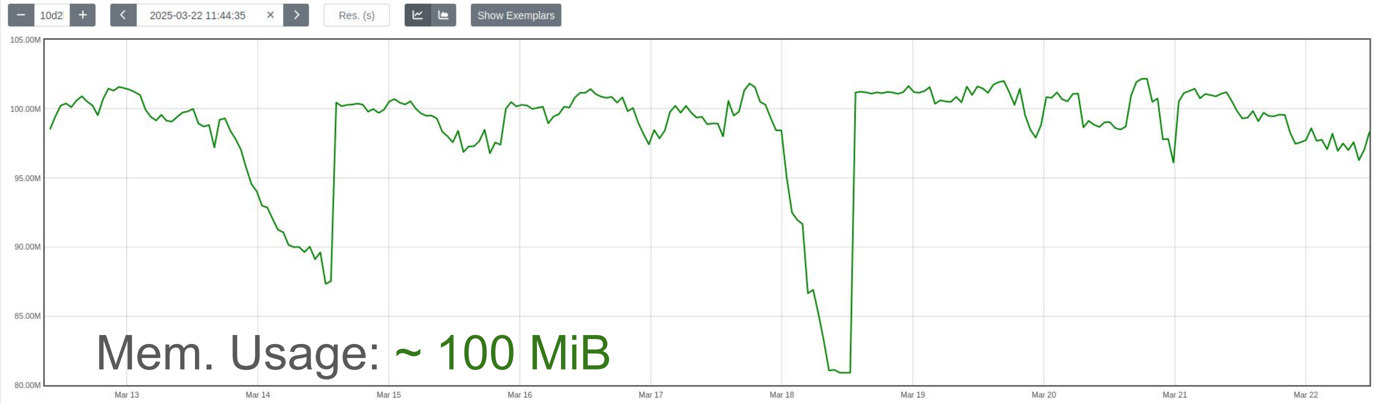
Symbol	Self ↓	Total
Image.py Image.__transformer	14.3 mins	14.3 mins
supporters.py TensorRunningAccum.append	8.72 mins	8.73 mins
rnn.py LSTM.forward	6.99 mins	7.01 mins
edit_based.py Levenshtein._cycled	5.91 mins	7.50 mins
ImageFile.py JpegImageFile.load	4.73 mins	5.48 mins
image_pooling_sequencer.py ImagePoolingSequencer.forward	3.40 mins	3.56 mins
base.py _ident	1.62 mins	1.62 mins
pin_memory.py pin_memory	1.24 mins	4.90 mins
ctc_greedy_decoder.py <listcomp>	1.24 mins	1.48 mins
padding_collater.py collate_tensors	1.22 mins	1.40 mins
reductions.py reduce_storage	1.19 mins	3.86 mins
checks.py check_tensor	1.06 mins	1.06 mins
functional.py to_tensor	1.04 mins	1.26 mins
shmem_getpage_gfp	49.0 s	1.74 mins
Image.py JpegImageFile.convert	46.8 s	6.29 mins
Image.py Image.point	44.0 s	52.5 s
conv_block.py get_output_size	42.9 s	43.1 s

Exporter Overhead

CPU and Memory Usage averaged over ~360 nodes.



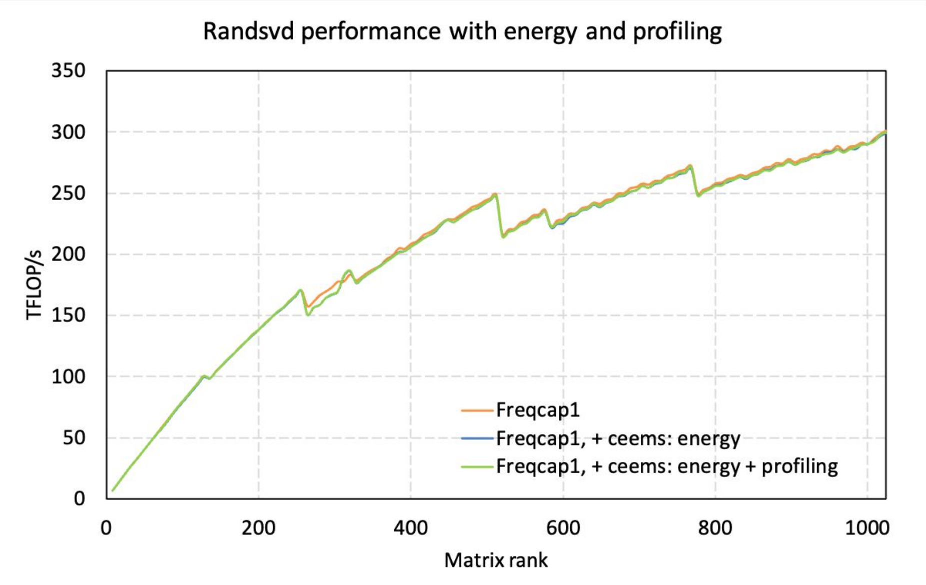
CPU Usage: < 0.5 %

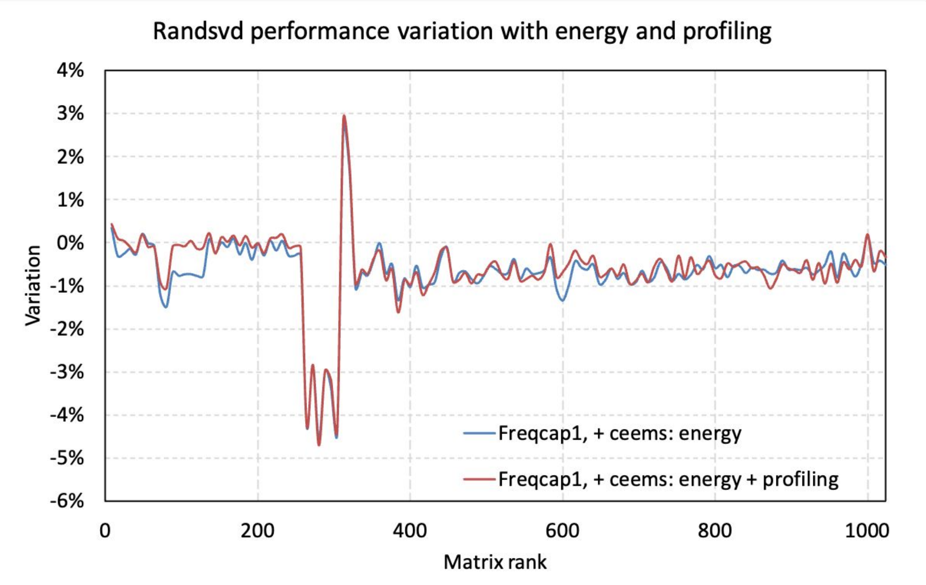


Mem. Usage: ~ 100 MiB

Benchmarks

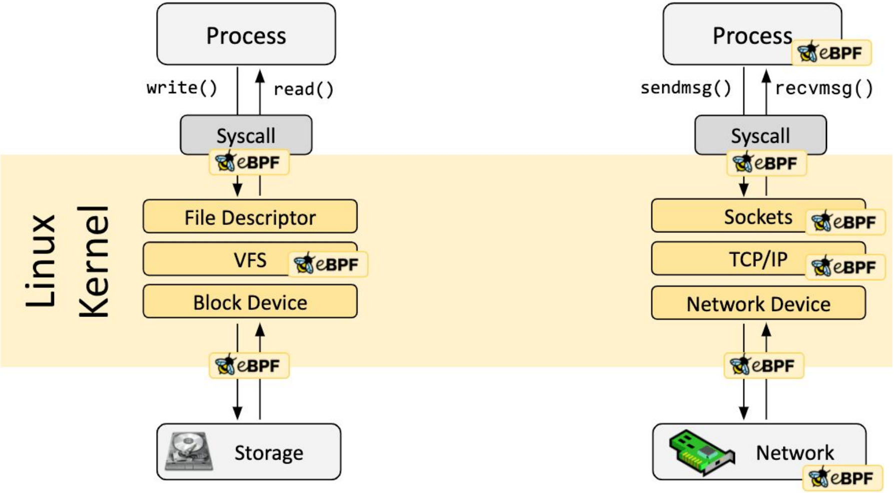
Randomised SVD with varying matrix size





Technical Details

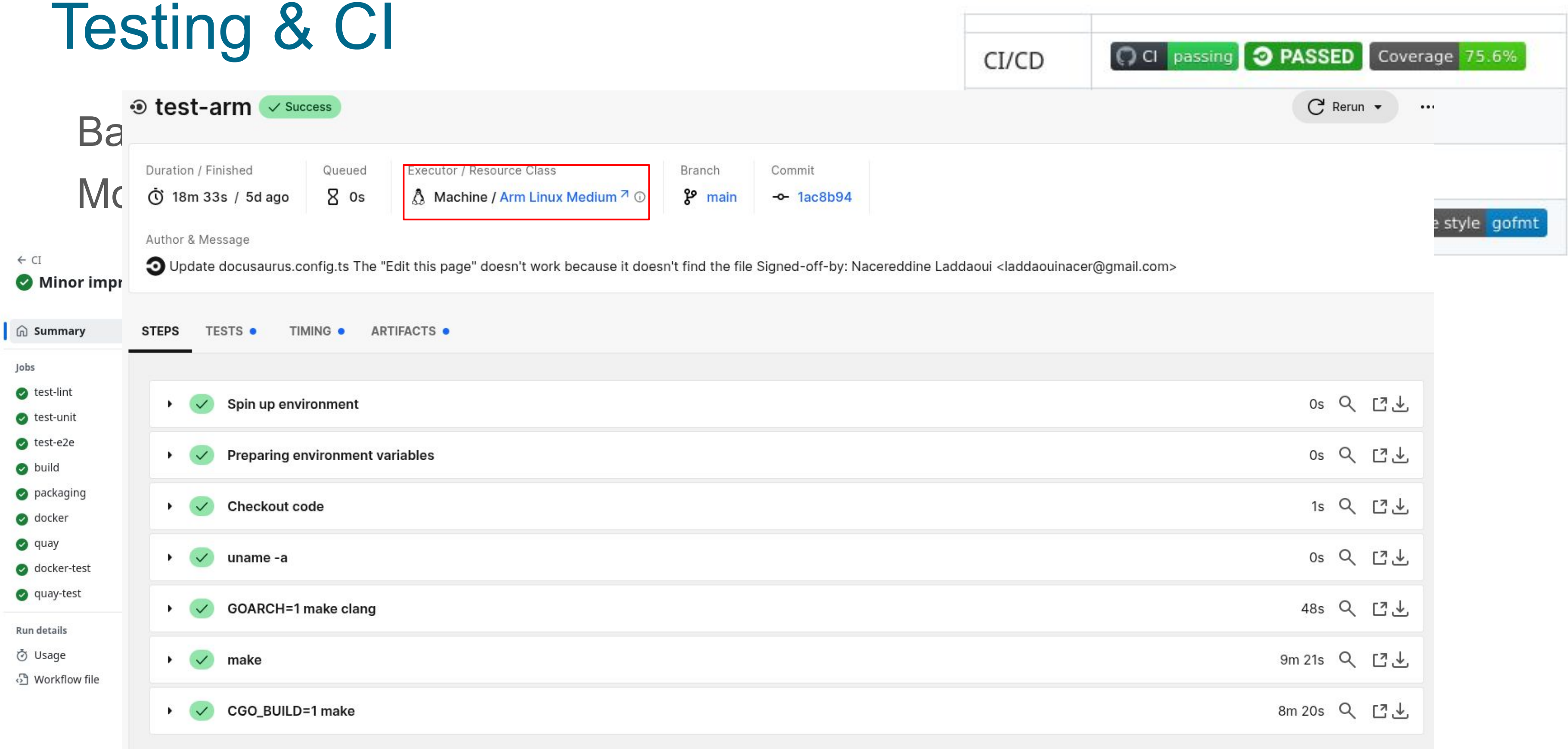
- 100 % Go (except the bpf programs which are in C)
- CEEMS apps are **Capability Aware**
- Uses eBPF for IO and Network metrics



Testing & CI

Ba

Mc



CI/CD	CI passing ✓ PASSED Coverage 75.6%

style gofmt

test-arm ✓ Success

Rerun ▾ ···

Duration / Finished: 18m 33s / 5d ago

Queued: 0s

Executor / Resource Class: Machine / Arm Linux Medium ↗ ⓘ

Branch: main

Commit: 1ac8b94

Author & Message

Update docusaurus.config.ts The "Edit this page" doesn't work because it doesn't find the file Signed-off-by: Nacereddine Laddaoui <laddaouinacer@gmail.com>

← CI

✓ Minor impr

Summary

Jobs

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

Run details

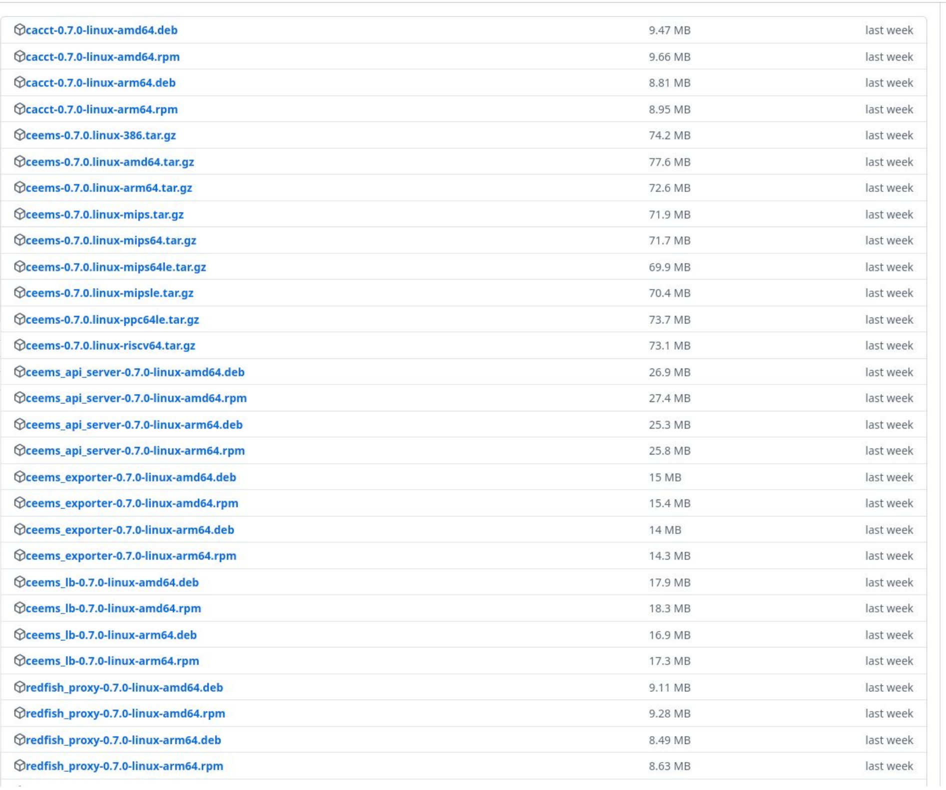
- Usage
- Workflow file

STEPS | TESTS ● | TIMING ● | ARTIFACTS ●

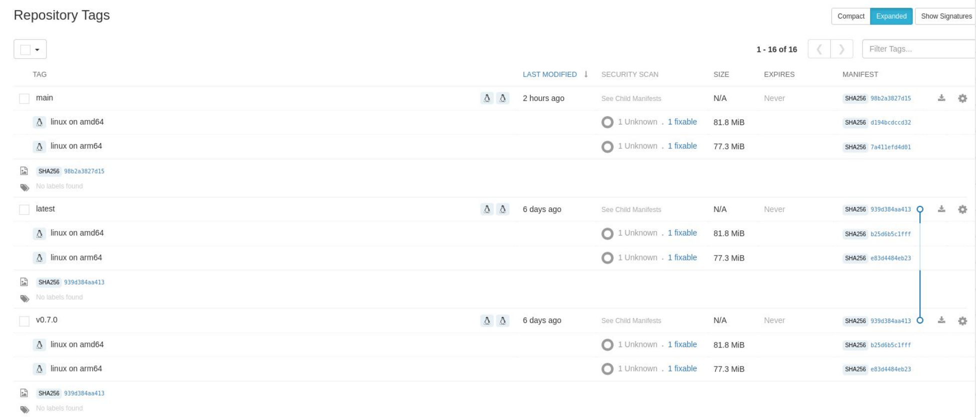
Step	Time
✓ Spin up environment	0s
✓ Preparing environment variables	0s
✓ Checkout code	1s
✓ uname -a	0s
✓ GOARCH=1 make clang	48s
✓ make	9m 21s
✓ CGO_BUILD=1 make	8m 20s

Packaging

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



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



Repository Tags

TAG	LAST MODIFIED	SECURITY SCAN	SIZE	EXPIRES	MANIFEST
main	2 hours ago	See Child Manifests	N/A	Never	SHA256 98b2a3827d15
linux on amd64		○ 1 Unknown . 1 fixable	81.8 MiB		SHA256 d194bcdccd32
linux on arm64		○ 1 Unknown . 1 fixable	77.3 MiB		SHA256 7a411efd4d01
SHA256 98b2a3827d15 No labels found					
latest	6 days ago	See Child Manifests	N/A	Never	SHA256 939d384aa413
linux on amd64		○ 1 Unknown . 1 fixable	81.8 MiB		SHA256 b25d6b5c1fff
linux on arm64		○ 1 Unknown . 1 fixable	77.3 MiB		SHA256 e83d4484eb23
SHA256 939d384aa413 No labels found					
v0.7.0	6 days ago	See Child Manifests	N/A	Never	SHA256 939d384aa413
linux on amd64		○ 1 Unknown . 1 fixable	81.8 MiB		SHA256 b25d6b5c1fff
linux on arm64		○ 1 Unknown . 1 fixable	77.3 MiB		SHA256 e83d4484eb23
SHA256 939d384aa413 No labels found					

Demo

Final Remarks

- CEEMS provide a “complete” monitoring solution
- Running on Jean Zay since ~ 1 year with a scrape frequency of 10s
- Currently working on eBPF based monitoring for CUDA applications
- Add support to k8s
- 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.

Thank you

Resources:

- CEEMS GitHub Repo
- CEEMS Docs
- CEEMS API Server Docs
- CEEMS Exporter Metrics List
- CEEMS Demo