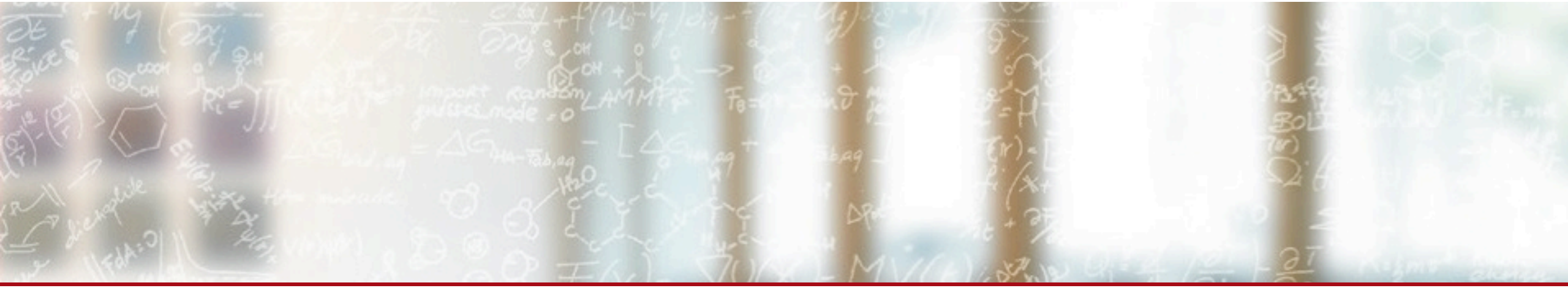
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Integrating Operational and Energy Datasets in HPC

hpc-ch forum on Business Continuity / Crises Management & Energy Savings and Efficiency

Michele Brambilla, CSCS

May 23, 2024

Outline

- Goal of the activity is to understand energy consumption in HPC jobs
- First step is to collect relevant and reliable data
- Compare measurements from sensors telemetry and software counters
  - Finally compare with manual measurements during real test runs
- Intrinsinc challenges due to the hardware infrastructure
- I will describe how a dynamic observability infrastructure can reduce the burden

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Hello, Alps

Alps & energy consumption

Introducing Alps, the new Swiss supercomputer

- 21 cabinets
- 128 nodes/cabinet
- 4 GraceHopper GH200 modules per node

Alps & energy consumption

Introducing Alps, the new Swiss supercomputer

- 21 cabinets
- 128 nodes/cabinet
- 4 GraceHopper GH200 modules per node

With a MaxPerformance of 270PF/s and 64GF/W (preAlps),
ranks 6th in the TOP500 chart and ranks 5th in the GREEN 500 chart

Alps & energy consumption

Introducing Alps, the new Swiss supercomputer

- 21 cabinets
- 128 nodes/cabinet
- 4 GraceHopper GH200 modules per node
- Nominal power 339 KW/cabinet

> 7 MW power (nominal) + network, storage, cooling

Observability is challenging

Alps requires to significantly scale up our observability platform

- ~ 10K GraceHoppers
- Heterogeneity of HW (CPU & GPU)
  - ~10K NVIDIA GraceHopper
  - ~2K AMD EPYC
  - ~700 AMD EPYC + NVIDIA A100
  - ~500 AMD MI300 CPU + GPU
  - 24 AMD EPYC + AMD MI250X GPU

Observability is challenging

Alps requires to significantly scale up our observability platform

- ~ 10K GraceHoppers
- Heterogeneity of HW (CPU & GPU)
- Multi-tenancy paradigm
- Not only logs and user data (accounting, SLURM jobs, storage, …) but also sensor data

Traditional approaches to observability do not scale well

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

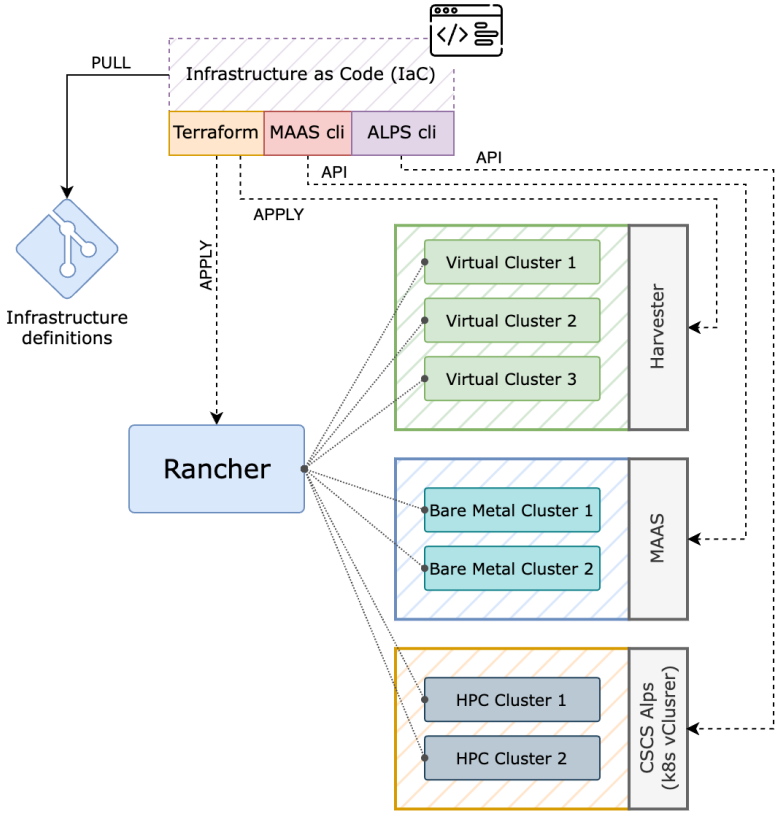
ETH zürich

EMOI: CSCS Extensible Monitoring and Observability Infrastructure

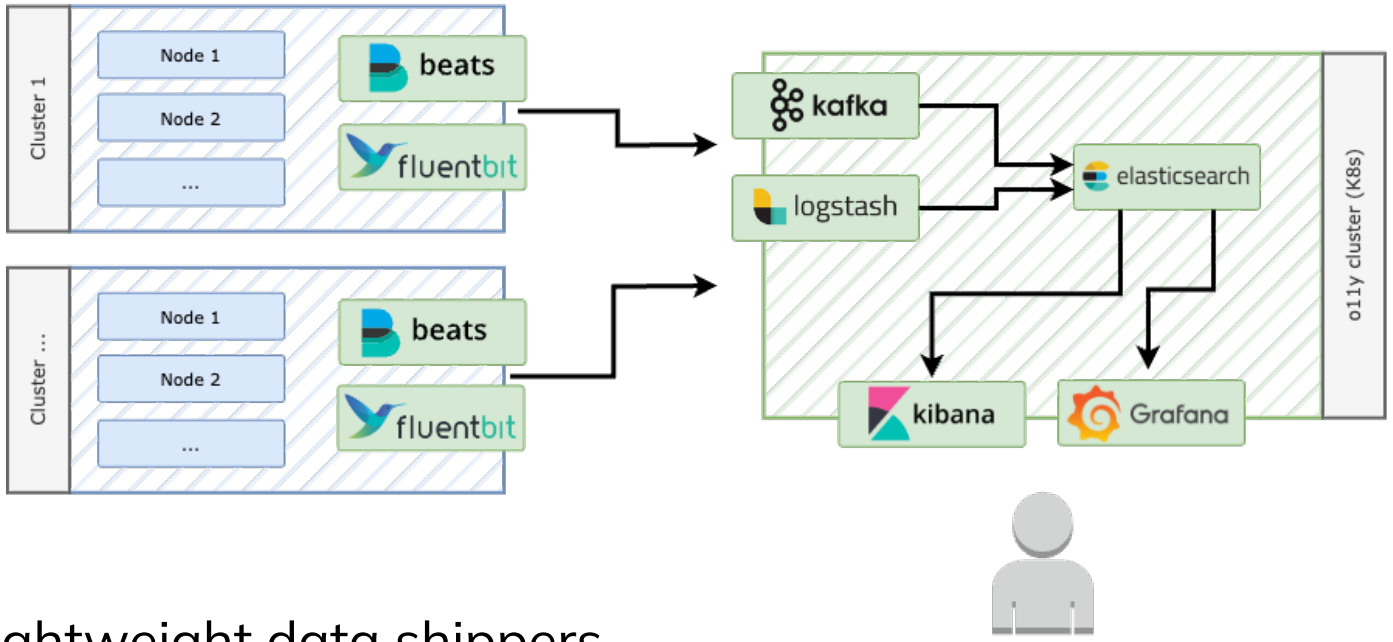
Setting our goals

Dynamic deployment of an Observability Cluster

- **Flexibility**: multiple physical or virtual clusters to accommodate custom workflows or external customers
- **Scalability**: provide horizontal scalability to meet changing demands
- **Automation**: apply IaaC principles and GitOps approach



Components

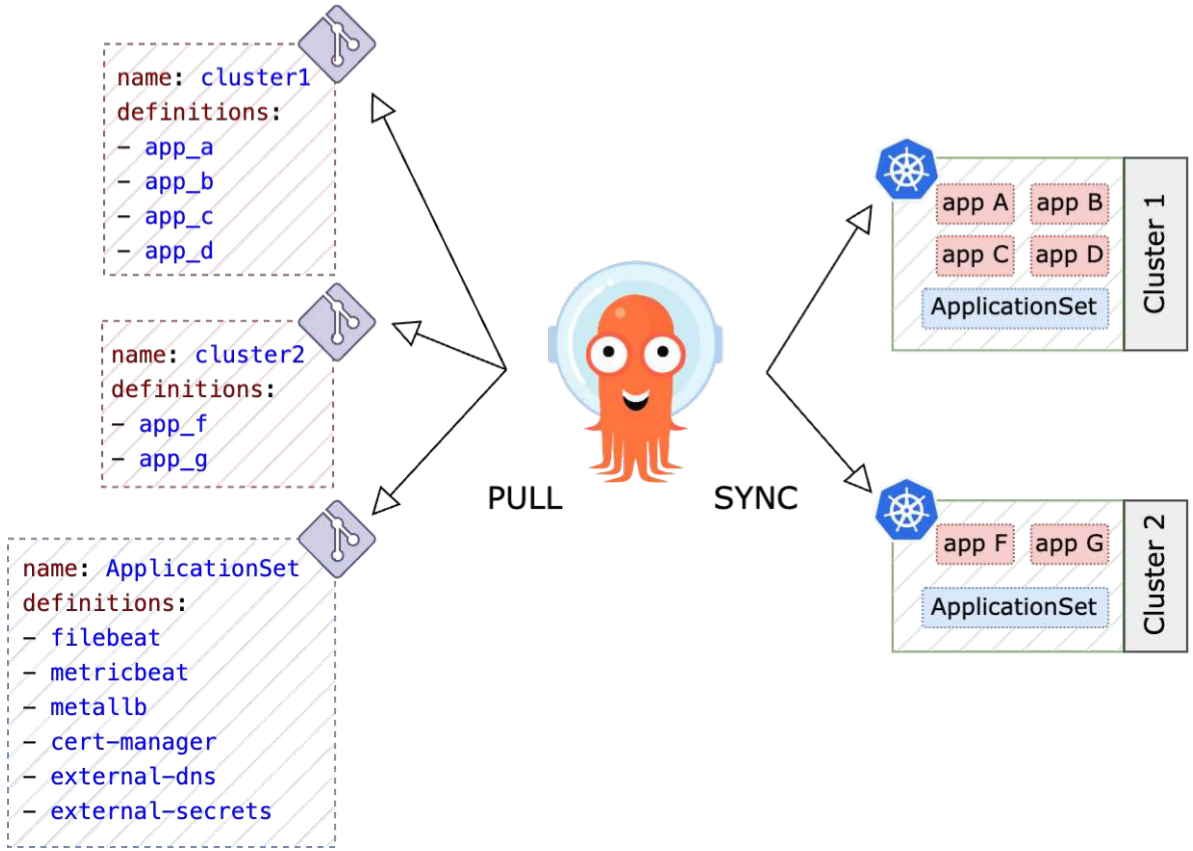


- **Beats, Fluentbit**: lightweight data shippers
- **Kafka:** message broker, implements data streaming, buffering and processing
- **Logstash**: data processing pipeline
- **Elasticsearch**: distributed search and analytics engine designed for storing large volumes of data
- **Kibana, Grafana**: visualization tools, build dashboards, view and analyze data

GitOps with ArgoCD

ArgoCD: GitOps continuous delivery tool for Kubernetes

- **Declarative**: deployments are declared using Kubernetes manifests or Helm charts
- **Version control**: applications and deployments are stored in Git repositories
- **Application definition**:for each cluster a separate git repo with all apps manifests
- **Graphical UI**: visualization of applications and differences between current and target state



Investigating the energy dataset

Goals

- Identify which component of the telemetry corresponds to the total energy of the node
- Compare telemetry energy data of the node with SLURM energy data of the jobs
- Verify telemetry data during the (cabinet) Grace-Hopper tests with ICON

Investigating the Energy Dataset

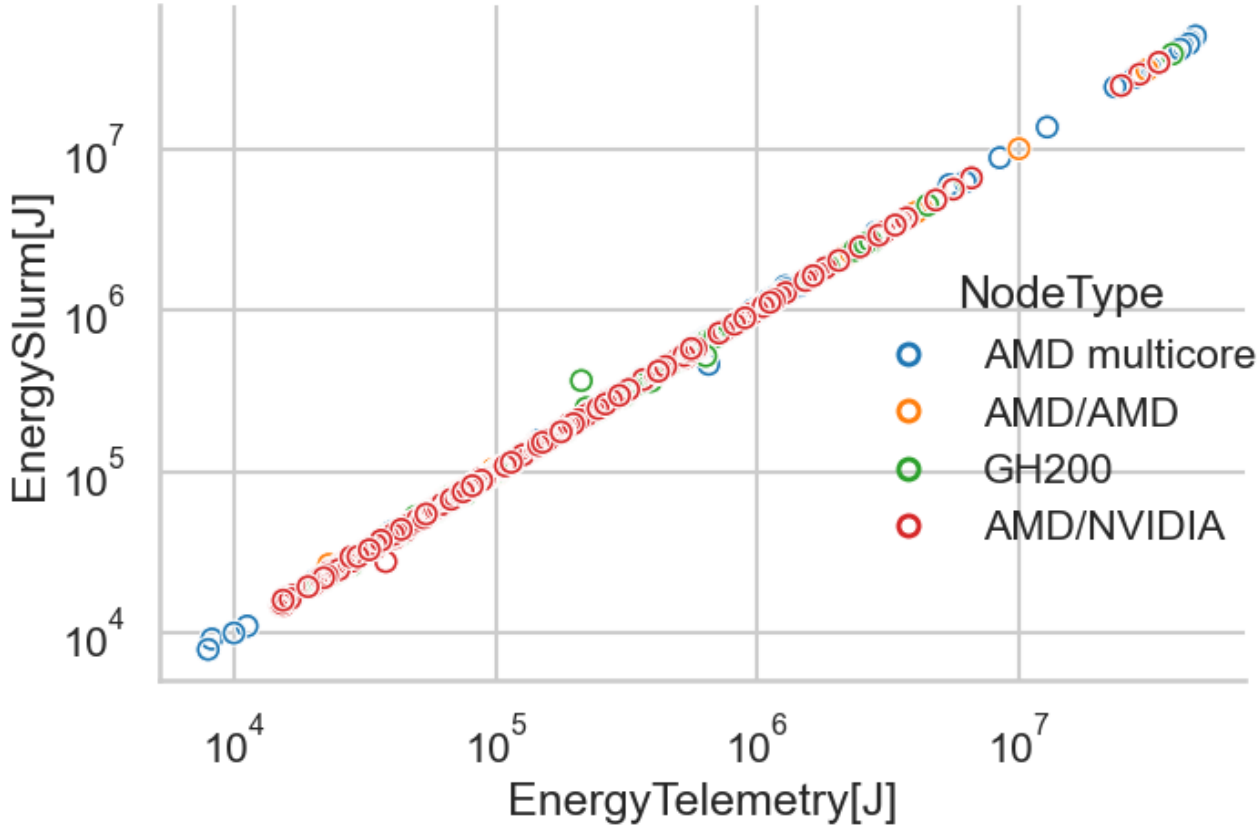
We can access energy consumption data from

- */sys/cray/pm_counters/* files on the node
  - 4 components: CPU, GPU, memory, total
  - Sample frequency ~10Hz
  - We can use them via SLURM jobAccounting
    - 1 report at the end of the job
- Redfish telemetry
  - Via the Alps management plane
  - Sample frequency < 1Hz

From the comparison of the measurements, the relevant field in the telemetry is VoltageRegulator

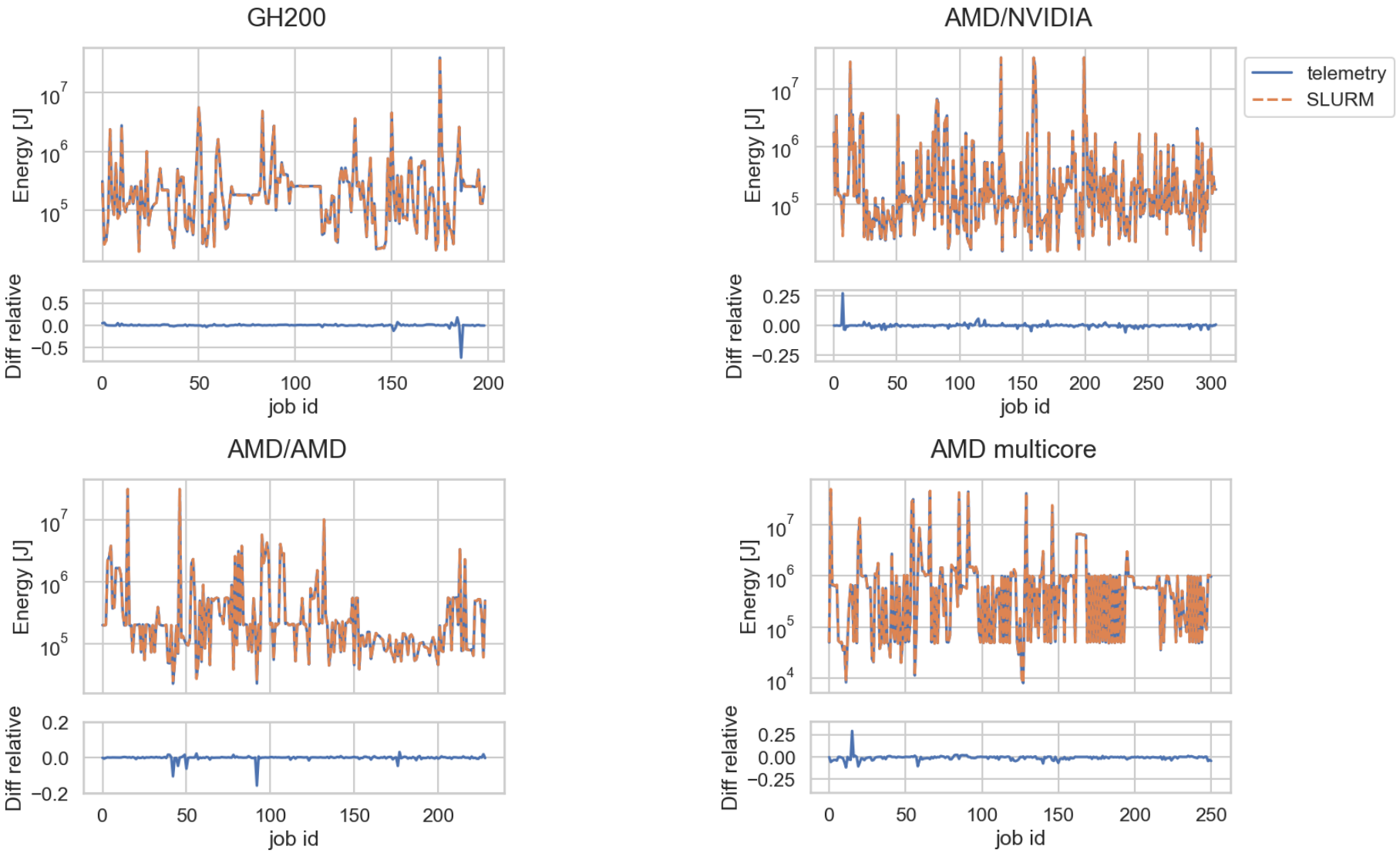
kind	task	start	end	CPU [J]	Memory [J]	VoltageRegulator [J]	node energy [J]
Telemetry	all	2024-01-29 T14:17:32	2024-01-29 T14:27:01	69'975	73'987	174'706	
PM file				69'686	72'185		170'259
Telemetry	none	2024-01-29 T14:17:38	2024-01-29 T14:18:28	2'670	6'011	10'285	
PM file				2'616	6'125		10'370
Telemetry	1CPU cores	2024-01-29 T14:18:29	2024-01-29 T14:19:29	3'138	7'202	12'661	
PM file				3'172	7'349		12'916

Comparing telemetry and SLURM

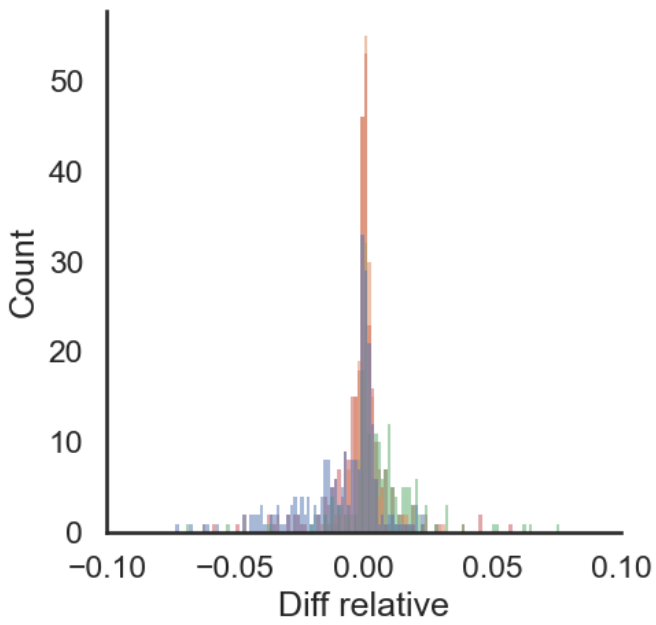


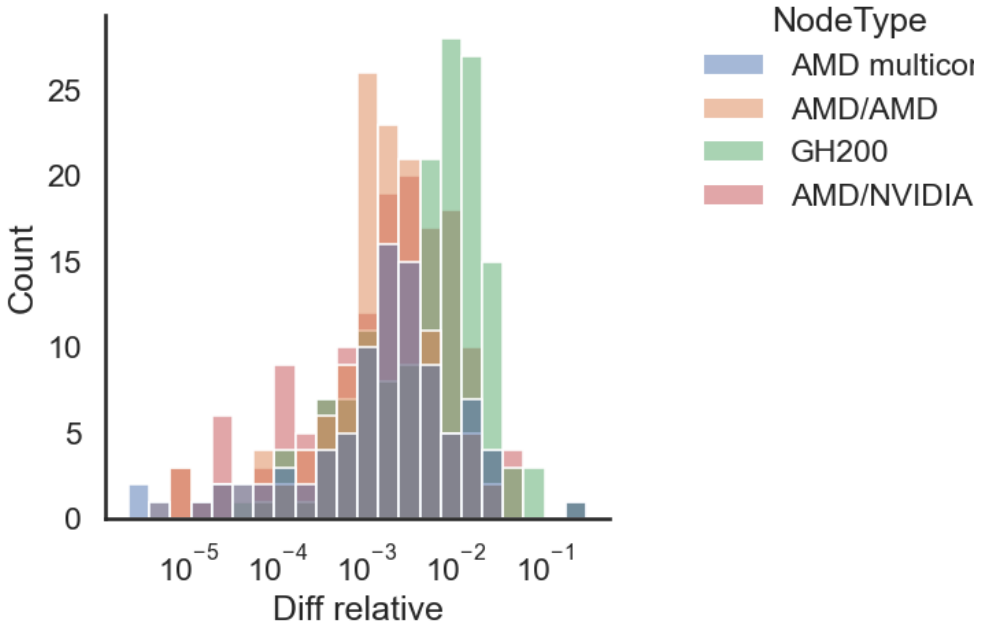
Almost all the points lay on a straight line, with minor deviations

Comparing telemetry and SLURM



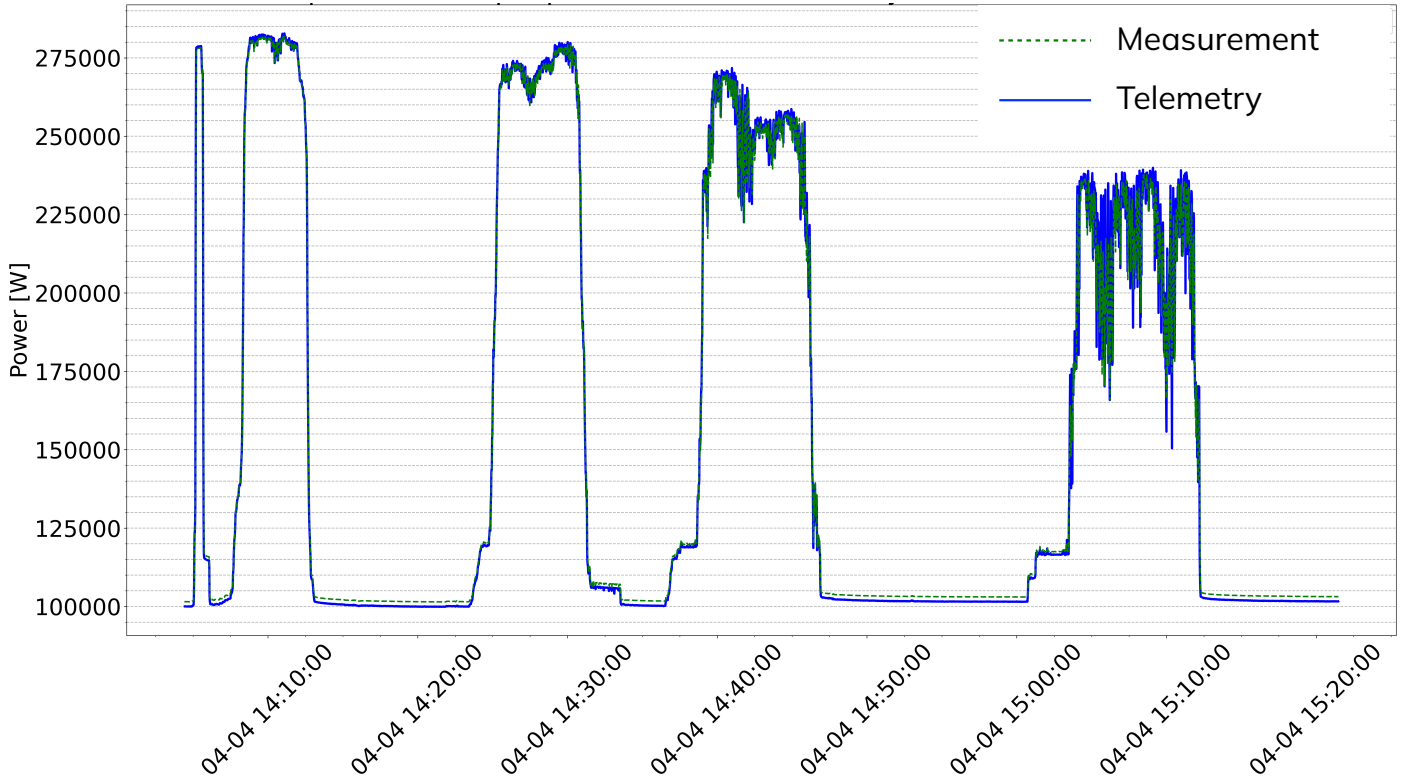
Comparing telemetry and SLURM





- Distribution of difference peaked around 0
- Most of the differences are < 10%

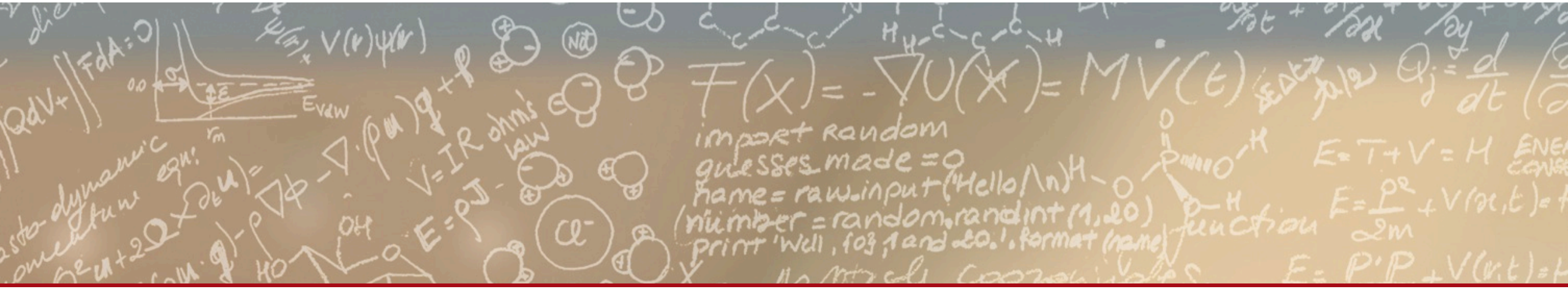
Comparing telemetry and FM measurements



- Telemetry has been validated against manual readings, with software used in production
- Good agreement between the two values

Conclusion

- Monitoring power consumption in HPC is a challenging task
  - We still have work to do, but we are on a good track
- Might require improving the observability infrastructure
- Automate as much as possible (and even more)
  - Our infrastructure can be interpreted as a “template” to be reused, integrated and where components can be replaced



Thank you for your attention.