Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
terabit

Monitoring resources of a computing infrastructure with Redfish and SNMP

CCR 2024 - 24/05/2024

Palau (SS), Italy

N. Mosco, L. Lavezzi, M. Sadocco, L. Tabasso

INFN TO

Outline

- Motivation: current status of Turin INFN resources and future developments goals.
- Monitoring tools and techniques.
- Test application and current status: correlate power consumption to CPU load and number of HTCondor jobs.

What do we have in Turin?

HTC Cluster (WLCG Tier 2)

- BELLE
- ALICE

Kubernetes cluster

- jupyterhub
- Yoga: local cluster for ML

HPC cluster

- CTLab/TechZoo
- ET Italy – Einstein Telescope
- GraceHopper superchip (ETIC)

What do we have in Turin?

CTLab *(Computing Technology Lab)***:**

- New cluster dedicated to ET (*TechZoo*).
- Evaluate new technologies.
- Dedicated heterogeneous computing power on demand.

TeRABIT:

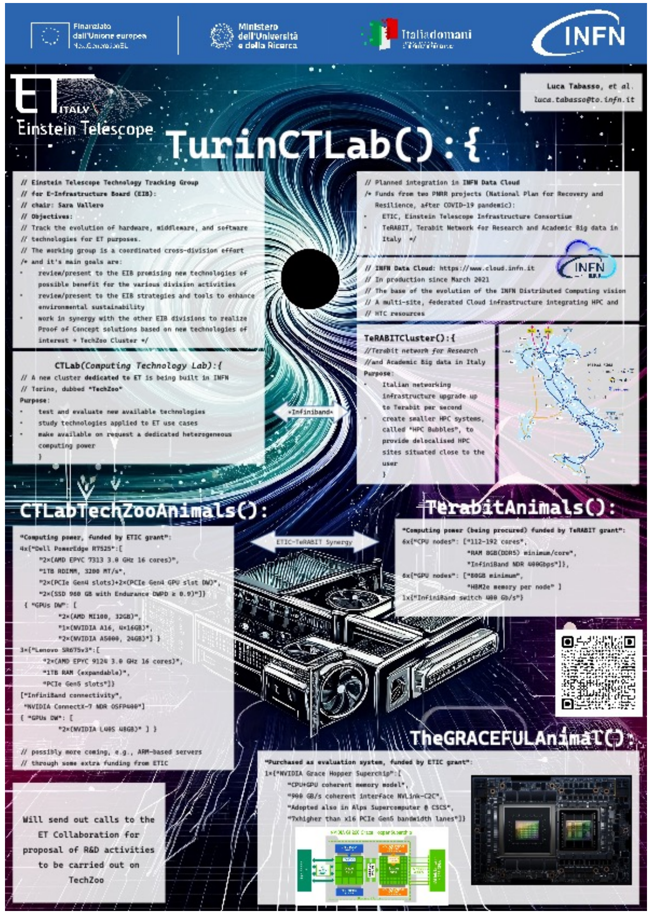
- Network for academic research.
- Upgrade current network to Tb/s.
- Infiniband connection high-bandwidth/low-latency.
- HPC bubbles: HPC resources available close to the user.

synergy

What do we have in Turin?

CTLab/TechZoo

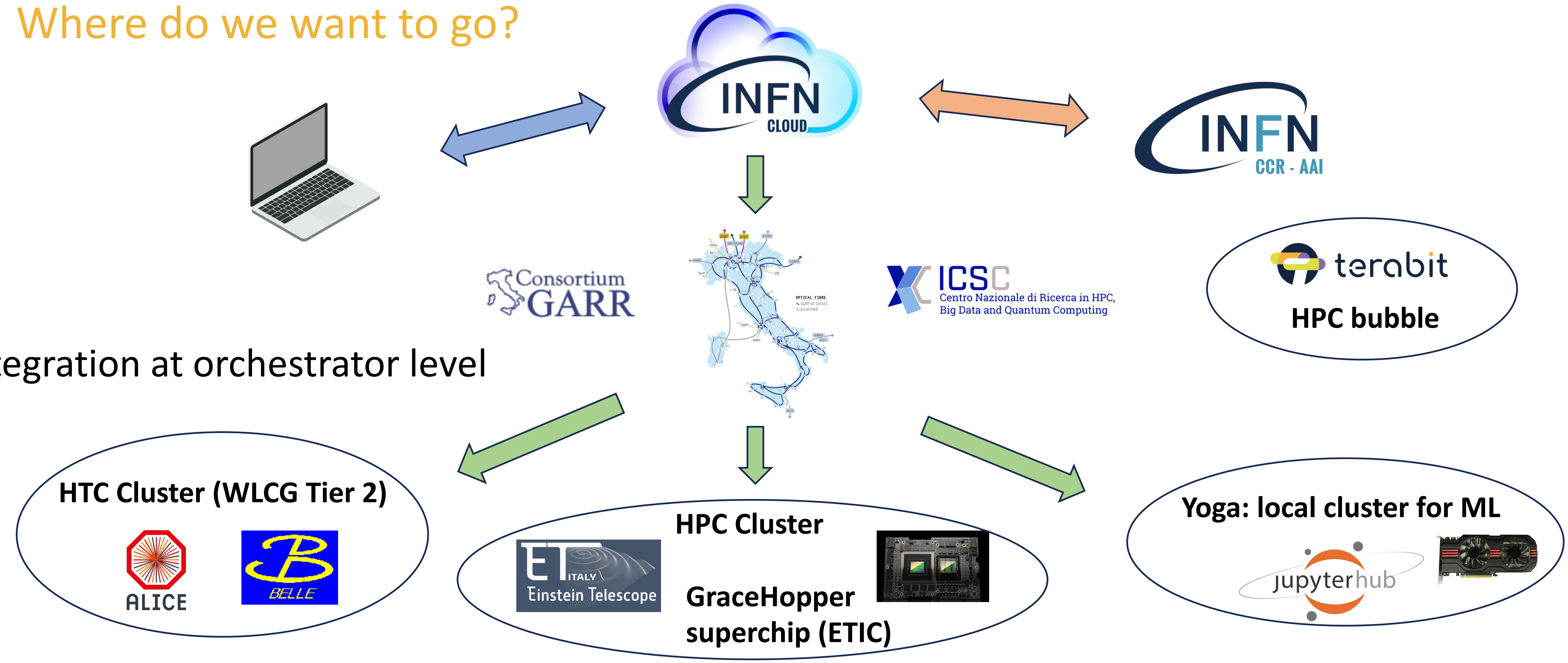
TeRABIT



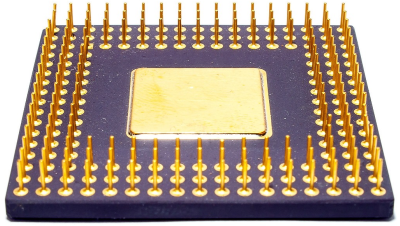
L. Tabasso

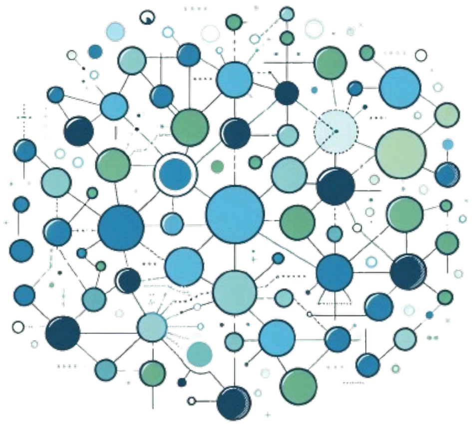
Where do we want to go?

Integration at orchestrator level



Monitoring a Computing Infrastructure





Resources:

- Computing nodes
- Network devices
- Storage
- (PDU/UPS)

Availability

Resource optimization

Issue detection

Tools and protocols for monitoring

Ganglia

Nagios®

SNMP

XDMoD
METRICS ON DEMAND

iCINGA

ZABBIX

Monitoring techniques

Agent-based monitoring

- Telegraf/InfluxDB
- Zabbix
- Datadog
- Nagios
- SNMP

Log monitoring

- ELK
- Graylog
- Redfish
- Loggly
- Splunk

Agent-less monitoring

- Telegraf
- SNMP
- Redfish

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
terabit

Redfish

- Open standard specification
- Agent-less
- RESTful API
- JSON-based schema
- Features: management, provisioning, monitoring
- Monitoring hardware health: temperature, power, disk I/O
- Logs collection

SNMP

- Industry standard protocol
- Manager-agent architecture
- Based on MIBs (Management Information Bases)
- Monitoring network devices, CPU/RAM usage, storage
- Real-time alerts for network faults and errors
- Mature and widely adopted in the industry

Prometheus

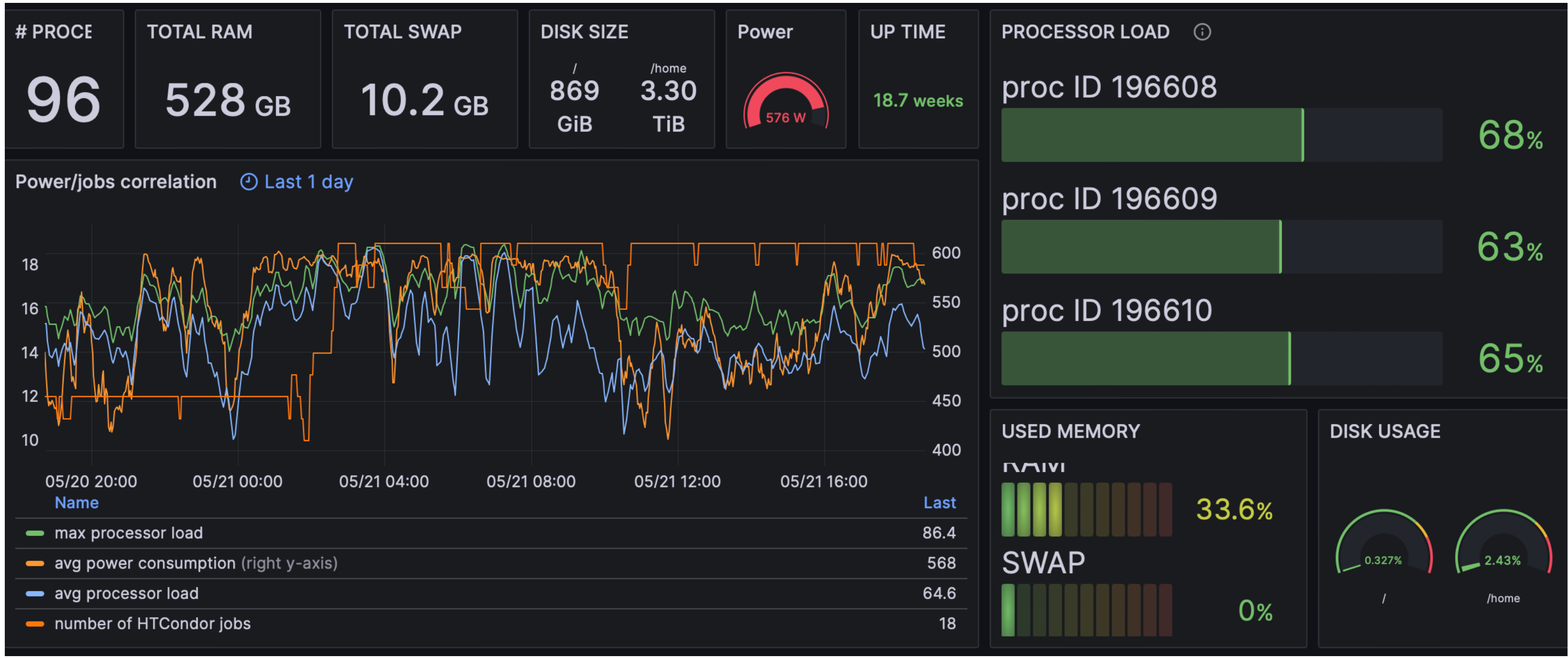
- Open source tool with alerting capabilities
- Time-series database
- Collection of data over HTTP (through exporters)
- Widely adopted for reliability and scalability

Grafana

- Open source platform for creating interactive dashboards
- Supports several data sources
- Notification when alerts are raised in Prometheus

Use case: Monitoring HTCondor jobs, CPU load, and power consumption

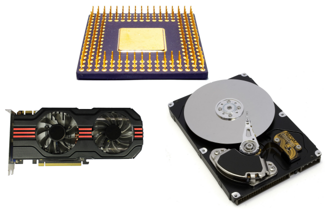
- SNMP tool installed on nodes to collect hardware metrics: CPU, disk, RAM; collect metrics about HTCondor jobs.
- Custom Redfish exporter to collect metrics about about power consumption and temperature.
- Data collected by Prometheus and displayed in Grafana: show correlation between power, CPU load, and number of jobs.



Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
terabit

HTCondor Software Suite
jupyterhub

OpenNebula



Application

Middleware

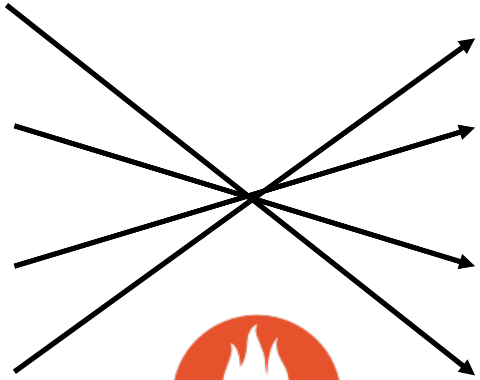
Hardware

Base infrastructure

Redfish

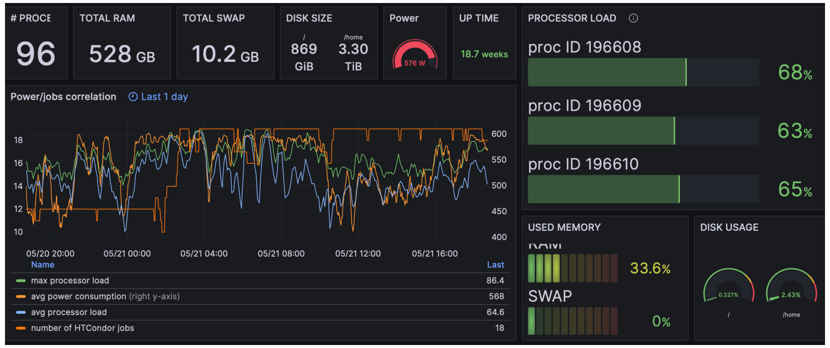
SNMP

Telegraf



Prometheus

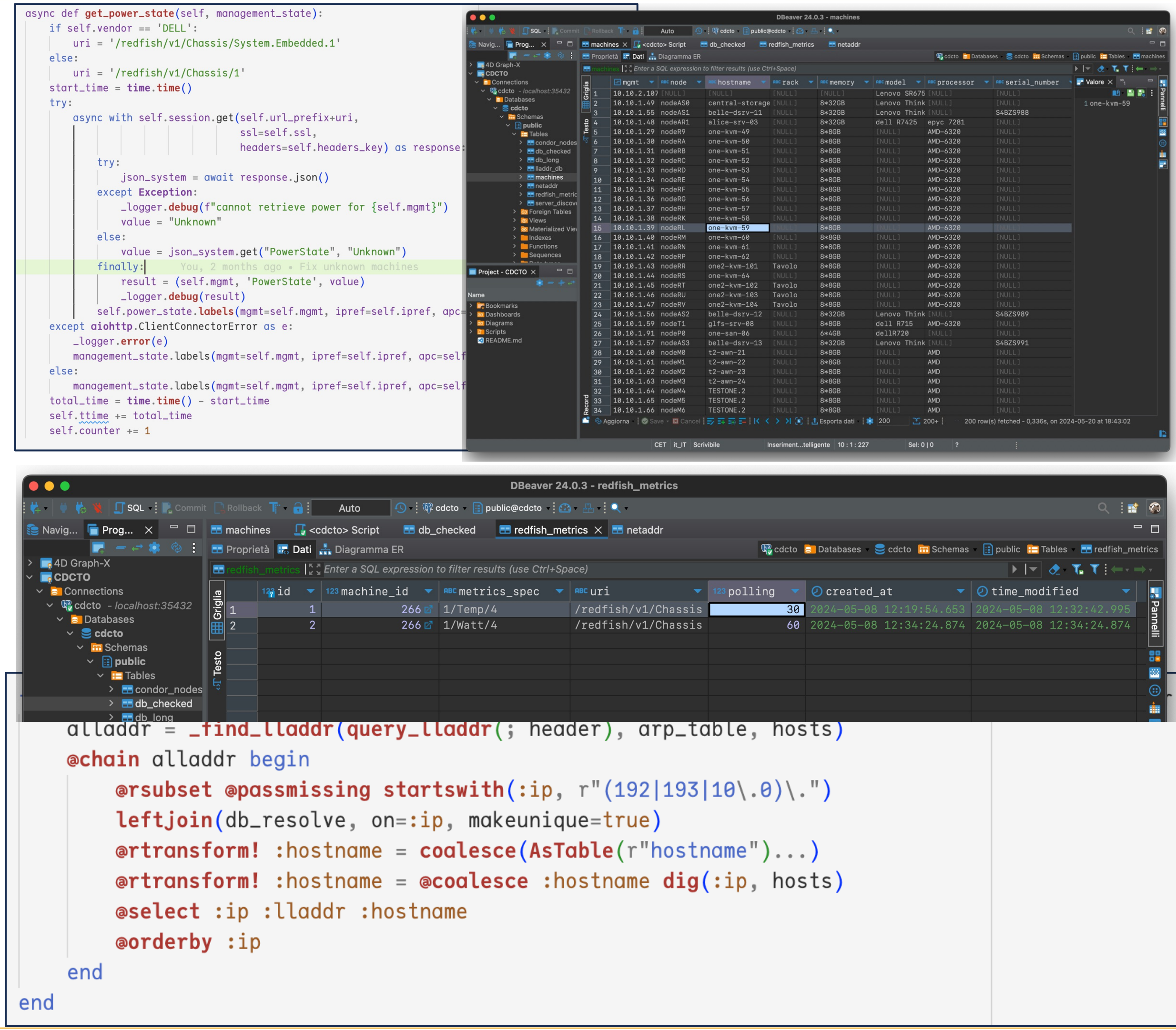
Grafana



Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
terabit

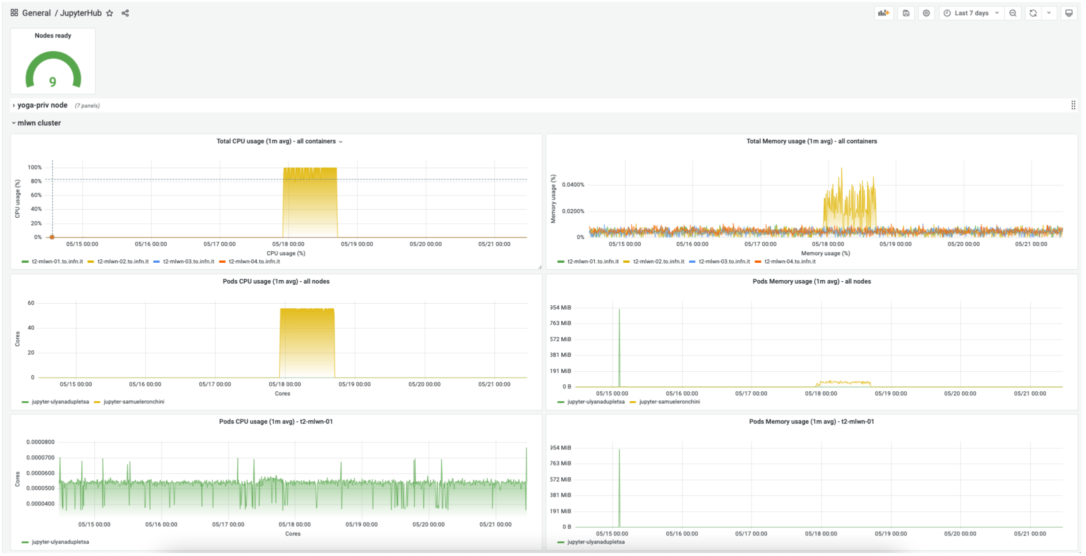
Future goals and development:

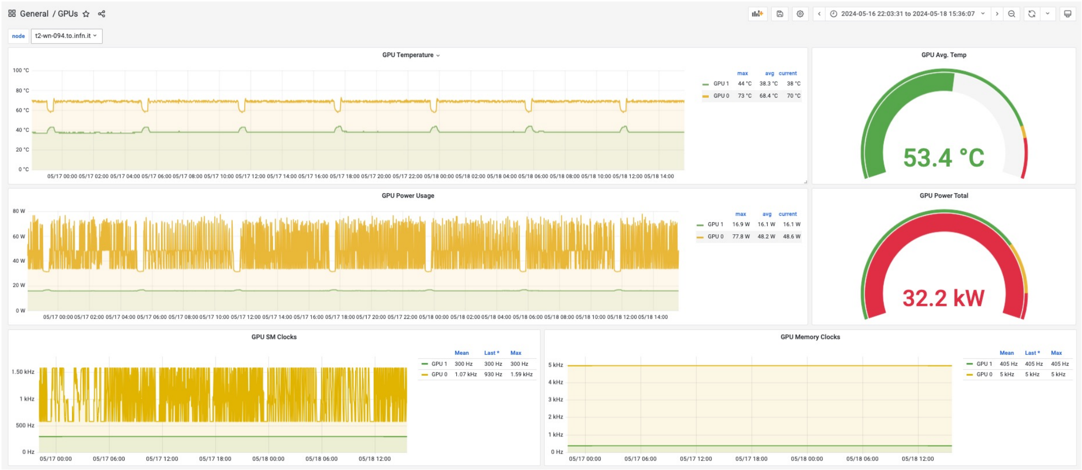
- Under development Python script for (semi)automatic discovery of metrics paths in Redfish.
- Metrics configuration stored in (Postgres) database.
- Configuration information processed by Julia scripts and loaded into the database.
- Redfish used to retrieve logs and collected by the ELK stack.



Yoga cluster

- Prototype for the resources that we intend to provide to the local community.
- It should be combined with INFN Cloud.
- Kubernetes cluster for ML applications.
- JupyterHub interface.
- CPU and GPU resources.
- Monitoring based on Prometheus and Grafana running in kubernetes nodes.





Conclusions

Thank you!

- Turin infrastructure is growing and new machines for different application clusters are being installed.
- Monitoring is important to keep track of resource usage, power consumption, fault detection, from all layers (infrastructure to software).
- Current development based on a custom solution using Redfish and SNMP for data collection; Prometheus and Grafana for data ingestion and visualisation.
- Custom tools and scripts written in different languages: *e.g.* the Redfish exporter and database manipulation scripts.

BACKUP SLIDES

What do we have in Turin?

CTLab *(Computing Technology Lab)***:**

- 4x Dell PowerEdge R7525:
  - 2×(AMD EPYC 7313 3.0 GHz 16 cores)
  - 1TB RDIMM, 3200 MT/s
  - 2×(PCIe Gen4 slots)+2×(PCIe Gen4 GPU slot DW)
  - 2×(SSD 960 GB with Endurance DWPD ≥ 0.9)
- GPUs DW:
  - 2×(AMD MI100, 32GB)
  - 1×(NVIDIA A16, 4×16GB)
  - 2×(NVIDIA A5000, 24GB)
- 3×Lenovo SR675v3
  - 2×(AMD EPYC 9124 3.0 GHz 16 cores)
  - 1TB RAM (expandable)
  - PCIe Gen5 slots

- InfiniBand connectivity
- NVIDIA ConnectX-7 NDR OSFP400
- GPUs DW
  - 2×(NVIDIA L40S 48GB)

Possibly more…
Extra ARM servers funded by ETIC

What do we have in Turin?

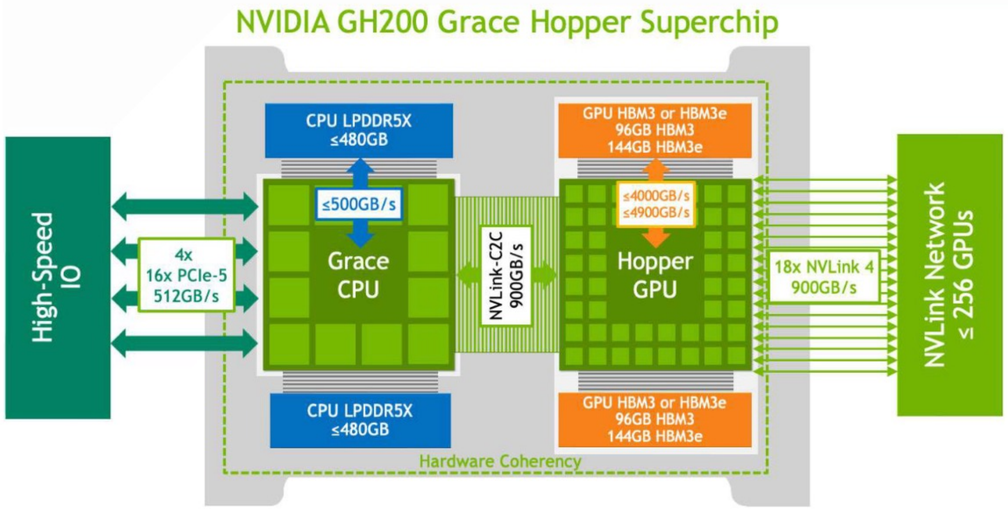
TeRABIT:

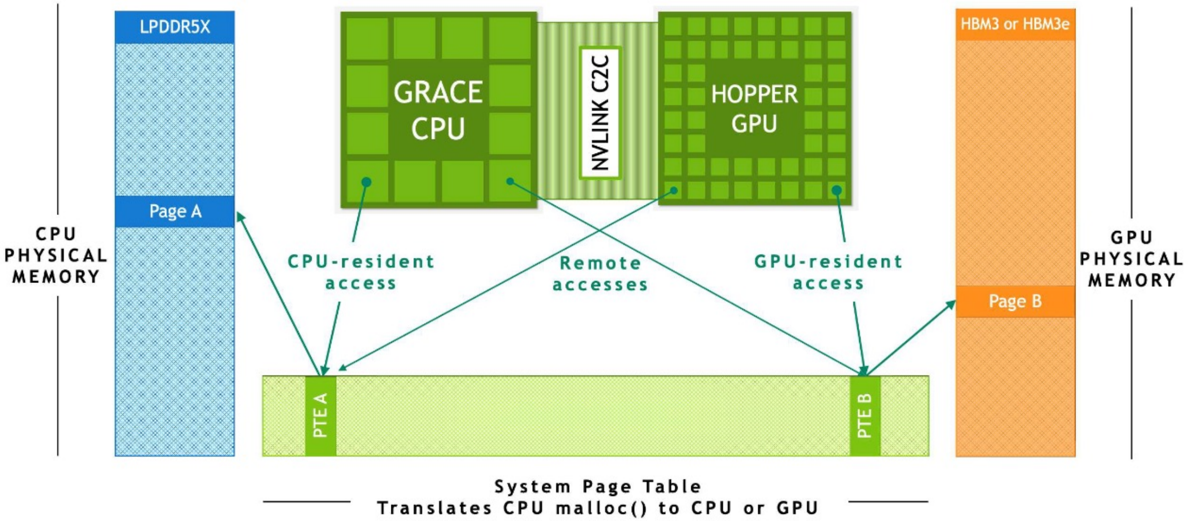
- 6xCPU nodes
  - 112-192 cores
  - RAM 8GB(DDR5) minimum/core
  - InfiniBand NDR 400Gbps
- 6xGPU nodes
  - 80GB minimum
  - HBM2e memory per node
- 1xInfiniBand switch 400 Gb/s

What do we have in Turin?

Evaluation system, funded by ETIC grant:

- 1×NVIDIA Grace Hopper Superchip
  - CPU+GPU coherent memory model
  - 900 GB/s coherent interface NVLink-C2C
  - Adopted also in Alps Supercomputer @ CSCS
  - 7x higher than x16 PCIe Gen5 bandwidth lanes



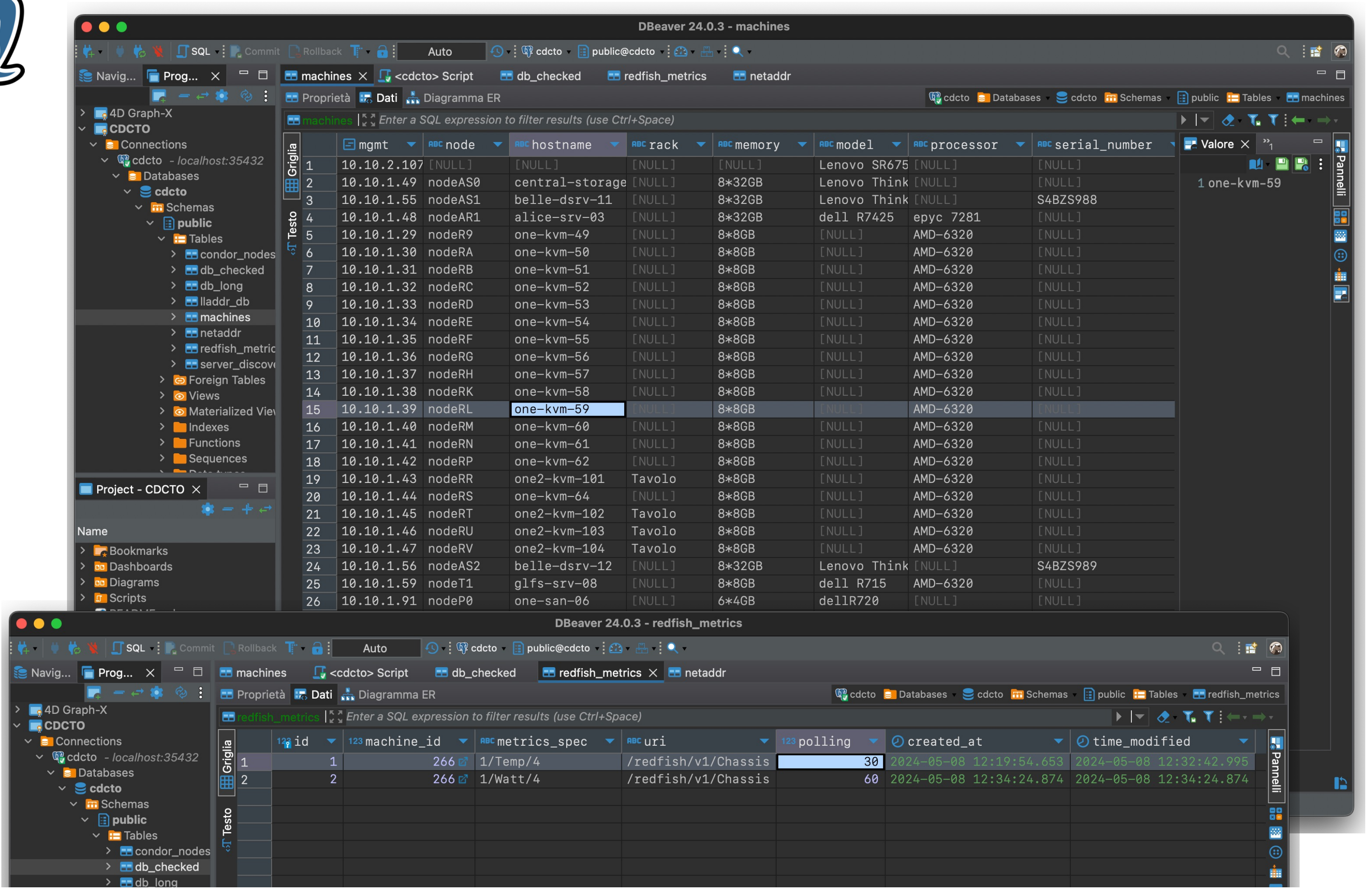


Redfish and Snmp

- **Benefits of Redfish and SNMP**
- **Scalability**: Redfish and SNMP support monitoring of large-scale infrastructures with ease.
- **Interoperability**: Both protocols are widely supported by hardware vendors, ensuring compatibility across different devices and platforms.
- **Real-time Monitoring**: Redfish and SNMP provide real-time insights into hardware and network performance, enabling proactive issue resolution.
- **Standardization**: Redfish and SNMP adhere to industry standards, simplifying integration and automation in infrastructure monitoring workflows.

Redfish configuration development

- Database initially built from preexisting sources, processing the information with Julia scripts.
- Julia script to verify the initial information.
- At this stage data is imported manually into the database.
- Prometheus configured by exporting the configuration *via* a script.



Future goals and development:

- Database of machines for configuration of Redfish exporter.
- Python script for (semi)automatic discovery of metrics paths.
- Redfish used to retrieve logs and collected by the ELK stack.

```
function find_multiple_addr!(df::AbstractDataFrame)
    addr_types(n) = map(("ip_$i" for i in 1:n)) do x
        replace(x, r"^ip_1" => "primary", r"^ip_2" => "seco
    end
    @chain df begin
        @rsubset begin
            :source in ["left_only", "both"] && startswith(
            @kwarg view=true
        end
        @groupby :lladdr
        transform!(nrow => :naddr)
        @rsubset(:naddr > 1; view=true)
        @aside @debug "naddr: $(length(_.naddr))"
        @transform! :type = addr_types(length(:naddr))
        @orderby :mgmt :ip
    end
end
```

```
async def get_power_state(self, management_state):
    if self.vendor == 'DELL':
        uri = '/redfish/v1/Chassis/System.Embedded.1'
    else:
        uri = '/redfish/v1/Chassis/1'
    start_time = time.time()
    try:
        async with self.session.get(self.url_prefix+uri,
                                    ssl=self.ssl,
                                    headers=self.headers_key) as response:
            try:
                json_system = await response.json()
            except Exception:
                _logger.debug(f"cannot retrieve power for {self.mgmt}")
                value = "Unknown"
            else:
                value = json_system.get("PowerState", "Unknown")
            finally:      You, 2 months ago • Fix unknown machines
                result = (self.mgmt, 'PowerState', value)
                _logger.debug(result)
            self.power_state.labels(mgmt=self.mgmt, ipref=self.ipref, apc=self.apc).state(result[2])
    except aiohttp.ClientConnectorError as e:
        _logger.error(e)
        management_state.labels(mgmt=self.mgmt, ipref=self.ipref, apc=self.apc).state("DOWN")
    else:
        management_state.labels(mgmt=self.mgmt, ipref=self.ipref, apc=self.apc).state("UP")
    total_time = time.time() - start_time
    self.ttime += total_time
    self.counter += 1
```

Monitoring a Computing Infrastructure

- Ensure availability
- Performance optimization
- Detect issues early
- Security

Monitoring a Computing Infrastructure

- Ensure availability
- Performance optimization
- Detect issues early
- Security

https://home.web.cern.ch/science/computing

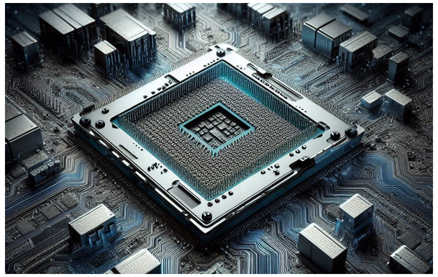
Finanziato dall'Unione europea
NextGenerationEU

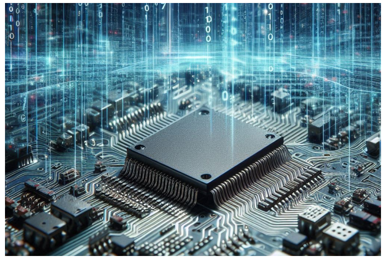
Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

terabit

Monitoring a Computing Infrastructure





Availability & Issue detection

Resource optimization