Evolving the CMS Submission Infrastructure

M. Mascheroni, A. Pérez-Calero Yzquierdo, F. Von Cube, V. Zokaite, H. Kim, L. Simas

Outline

CMS computing at scale
CERN, CMS and the WLCG ecosystem

The CMS Submission Infrastructure
Supporting Tier-0, production and user analysis

Improving resource utilization
Group surplus, IO-slots and pilot overloading

Towards HL-LHC-scale workflow management
Preparing for the future of high-luminosity experiments

The CMS experiment at CERN

Major LHC Experiment

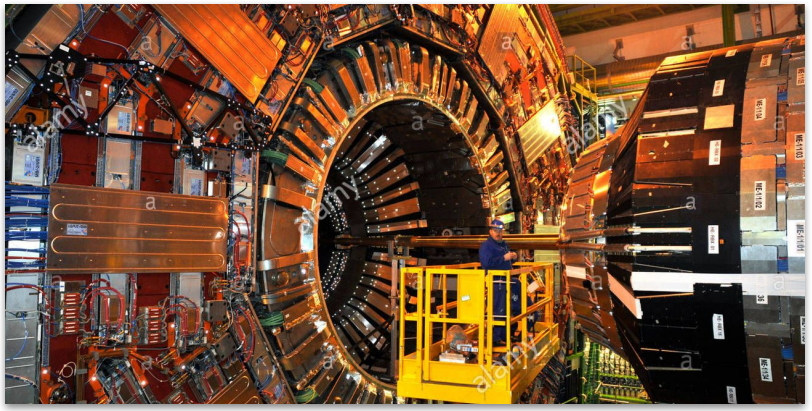
Proton-proton interactions at the world's **highest collision energy**

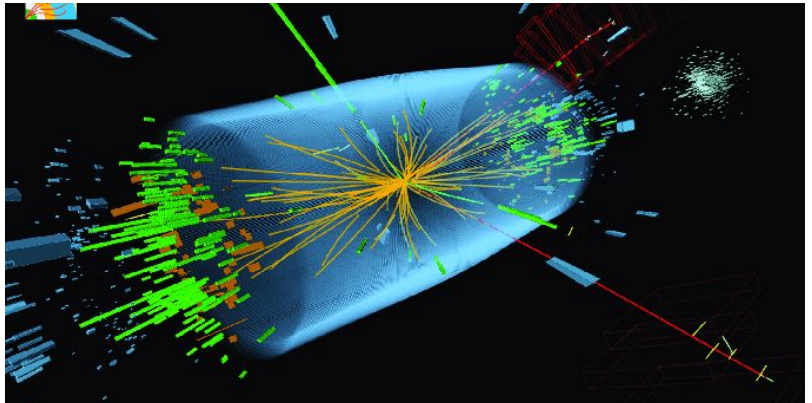
Fundamental Physics

Studies elementary particles and fundamental forces in broad range of processes

Massive Data Processing

Hundreds of PB of data processed every year from reconstructions and MC simulations



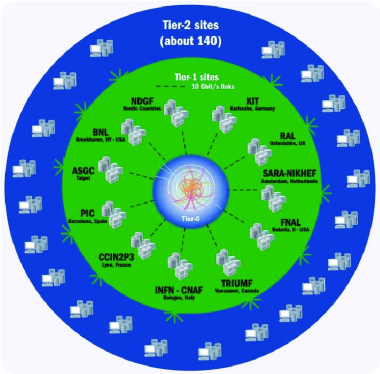


The Worldwide LHC Computing Grid

WLCG Overview

CMS data is processed and analyzed using the **Worldwide LHC Computing Grid (WLCG)**.

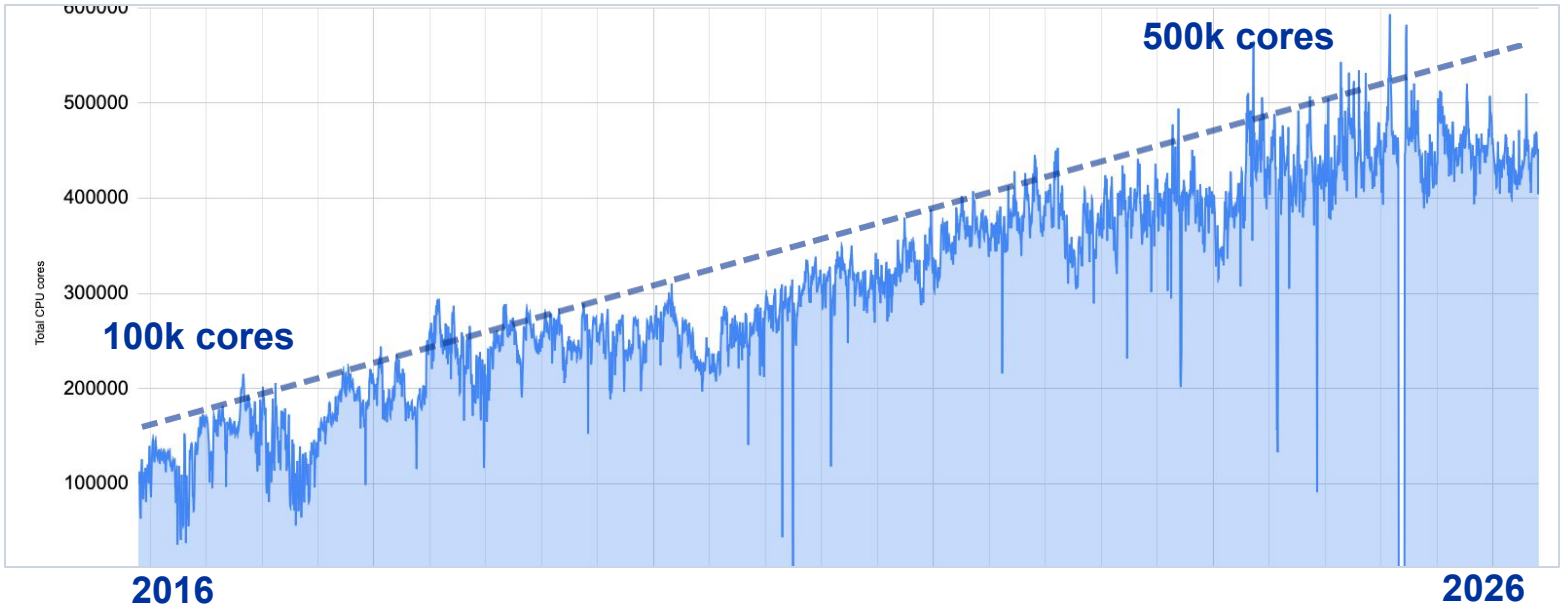
- **~170 computing centers distributed worldwide**
- **~1.4 million CPU cores and ~2 EB of storage**
- **Resources organized in Tier-0, Tier-1 and Tier-2 facilities**



Evolution of CMS CPU Capacity

Continuous growth from **~100k to ~500k cores** over the last decade.

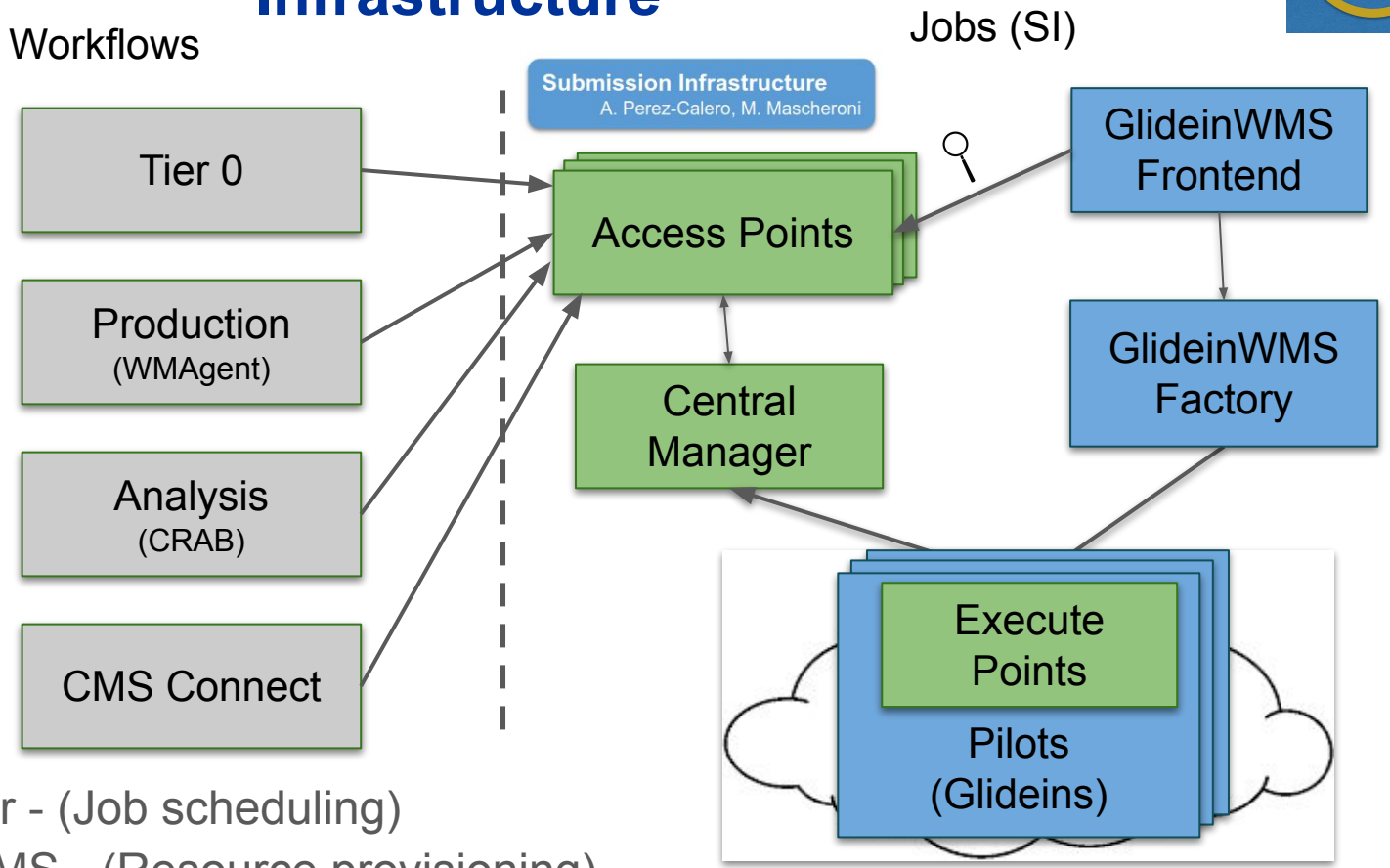
Single federated pool spanning **WLCG, HPC, and cloud resources** via HTCondor.



The CMS Submission Infrastructure

Supporting Tier0, production and analysis

Architecture of the CMS Submission Infrastructure



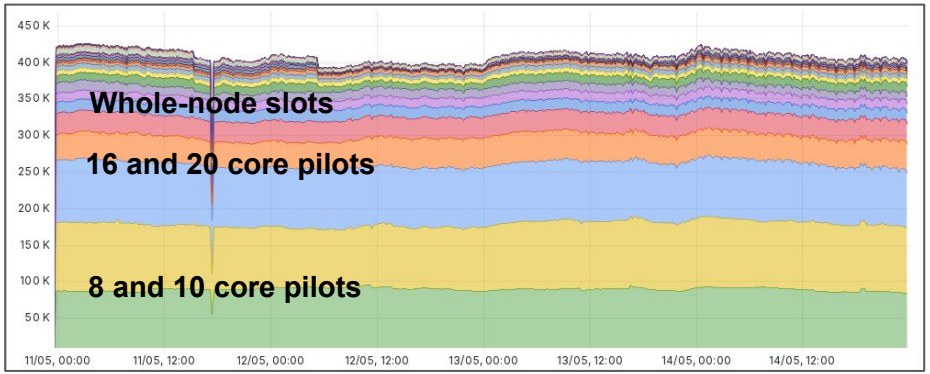
HTCondor - (Job scheduling)

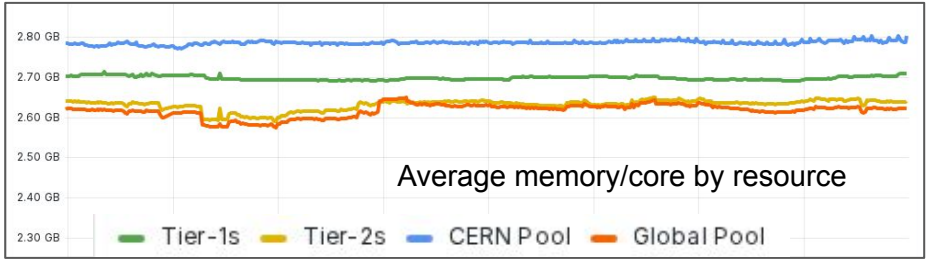
GlideinWMS - (Resource provisioning)

Resource Provisioning & Partitioning

Acquiring resources:

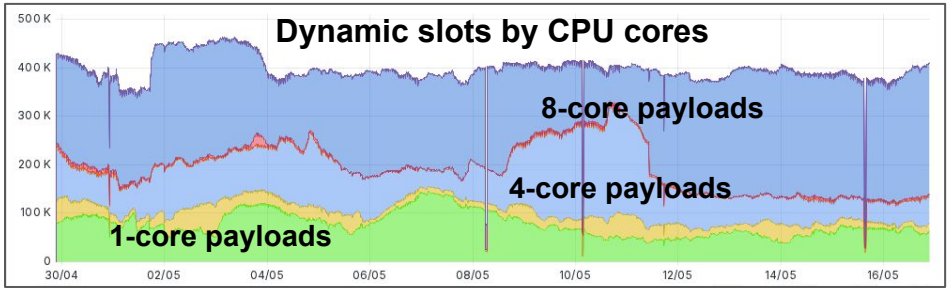
- Mostly **16-core and 8-core pilots**
- Also **whole-node** and **HPC** resources
- No need for dedicated **high-memory slots**

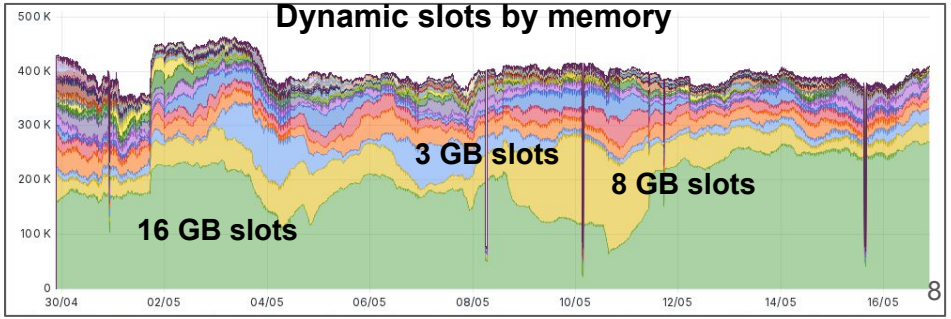




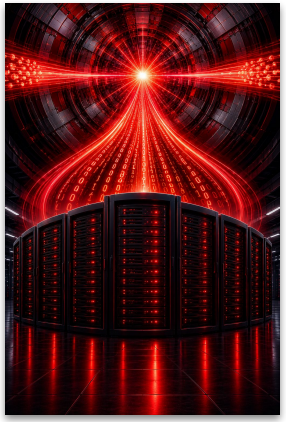
Using resources:

- Dynamic partitioning by **CPU and memory**
- Common payloads: **1-, 4- and 8-core**
- **Mixed workloads**: Prod, Analysis, and Tier 0





Tier 0 use case



Tier 0

Production
(WMAgent)

Analysis
(CRAB)

CMS Connect

- **RAW event** data must be formatted and reconstructed
  - **Repack** jobs (single-core, highest priority)
  - **Prompt Reconstruction** jobs (multicore)
- **Latency-critical**
  - **~1 PB/day** during data taking
  - Delays risk filling online buffers

Production use case

Tier 0

Production
(WMAgent)

Analysis
(CRAB)

CMS Connect

- **Large-scale campaign processing**
  - **Monte Carlo production** campaigns
  - Reprocessing and re-reconstruction of **data**
  - **Creation of** MiniAOD and NanoAOD **datasets**
- **Throughput-oriented**
  - Flexible workloads **optimized** for high resource utilization

Analysis use case

Tier 0

Production
(WMAgent)

Analysis
(CRAB)

CMS Connect

- **User-driven** workflows
  - Private Monte Carlo productions
  - Detector studies
  - Final physics analyses
- **Highly diverse**
  - Histograms from NanoAOD
  - Ntuplizers, skims and custom workflows
  - Many specialized use cases
  - Mostly **single-core**

Improving resource utilization

Group surplus, IO-slots and pilot overloading

Sharing Resources Across Tier-0, Production, and Analysis

Priority Resources

- Most **CERN** resources are reserved for **Tier0 workloads**

Flexible Sharing

- Production and analysis can **temporarily exceed their quota**
- Enabled through `GROUP_ACCEPT_SURPLUS`
- During **shutdown periods**, resources evolve toward a **production–analysis balance** (e.g. 50/50 split at CERN)

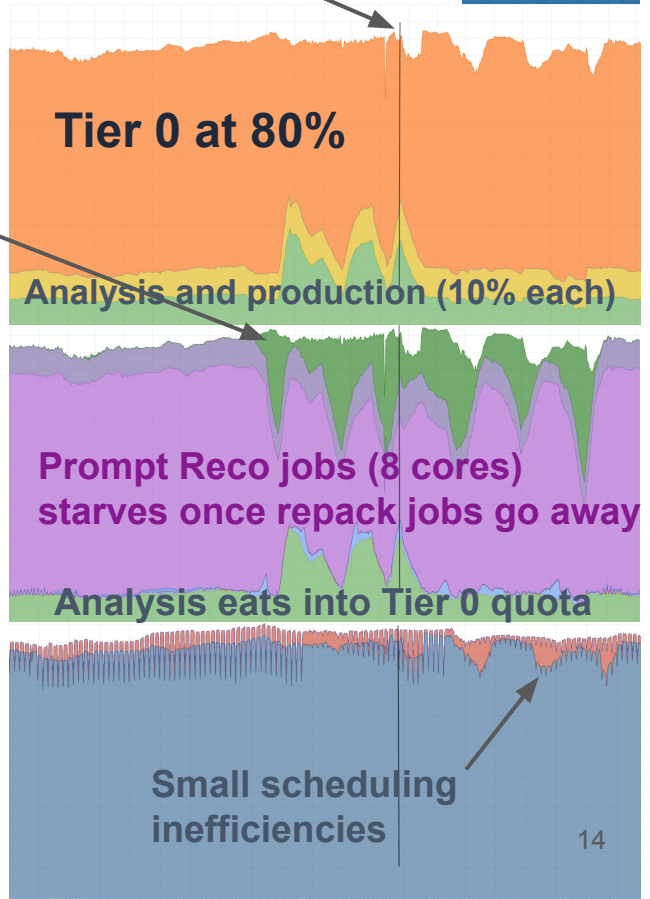
Group	Tier0	Production	Analysis
CERN	80%	10%	10%
T1	30%	63%	7%
US T2	10%	63%	27%
non-US T2 + T3	0%	60%	40%

Divisions in group possible thanks to **pool federation** and job's **flocking**

Single-core bursts create fragmentation and Tier-0 starvation

- **Fragmentation**
  - Bursty **single-core repack jobs** create internal fragmentation
  - Single-core **analysis** jobs opportunistically **fill the gaps**
- **Consequences**
  - **PromptReco multicore jobs** may temporarily **starve**
  - Disabling `GROUP_ACCEPT_SURPLUS` **restores Tier0 quotas**
  - Small scheduling **inefficiencies** introduced by **repack jobs retraction** once analysis cannot fill up the gaps

Tradeoff: efficiency versus Tier 0 latency



Dedicated IO-slots reduce fragmentation caused by auxiliary jobs

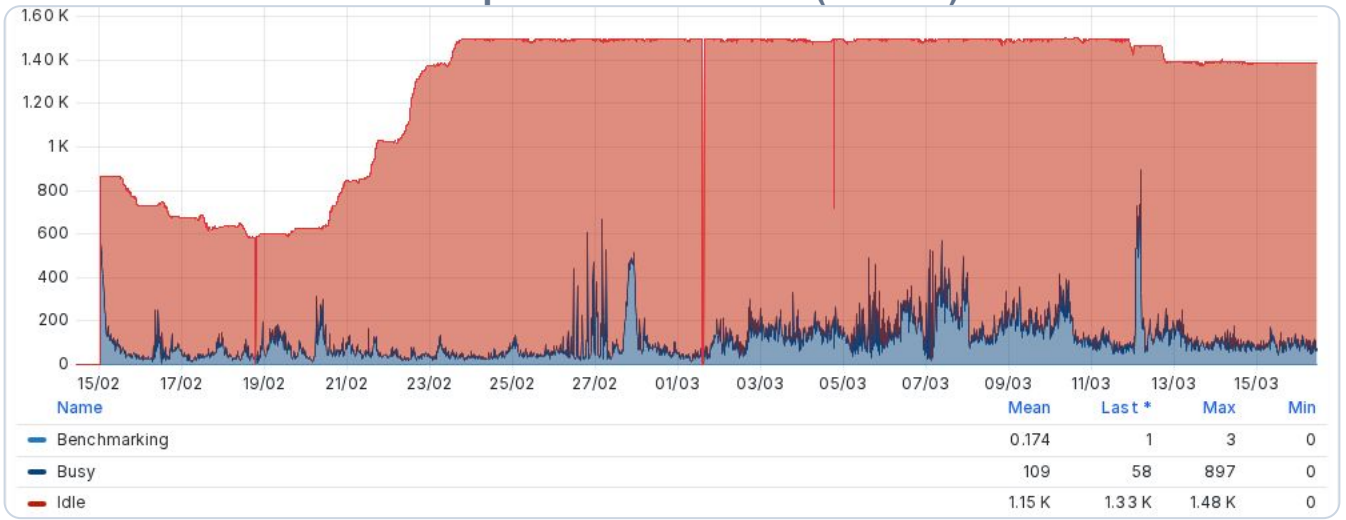
Fragmentation

- Short single-core auxiliary jobs (merge, log collect, …) arrive in bursts
- Internal fragmentation impacts prioritization and efficiency
- High-priority multicore jobs may not find large slots

Dedicated IO-slots

- CMS SI introduces an additional **IO-slot** per pilot
- Reserved exclusively for auxiliary payloads
- The slot is **virtually added** and idle by design
- Large slots are preserved and fragmentation is reduced

Example IO-slots in use (Tier-1s)



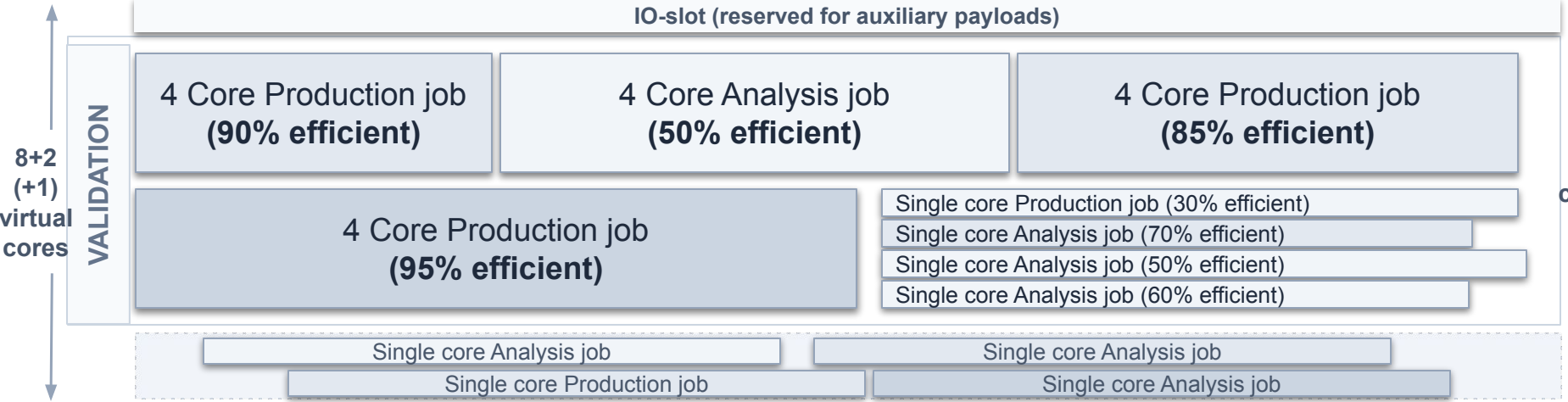
Recover Unused CPU Cycles: Overloading Pilots

Principle

- **Recover** unused CPU cycles, not new resources
- Push more workload into the same pilot envelope
- Moderate overloading (**+25% CPU and memory**)
- Additional payload jobs match extra **virtual cores**

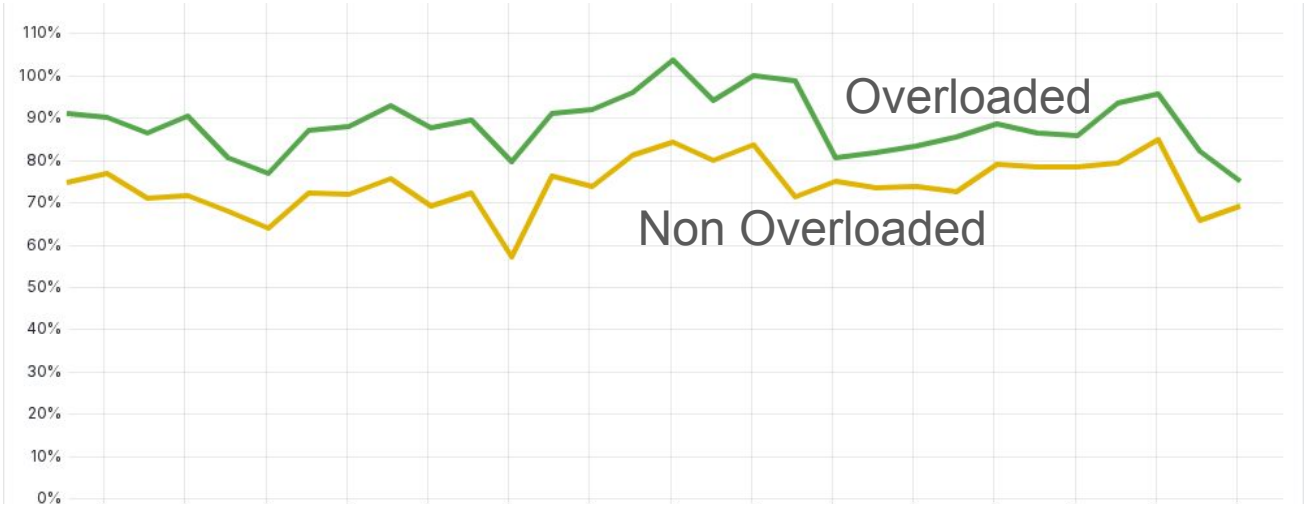
Benefits

- Improve overall CPU utilization
- No workflow changes required
- Implemented entirely from the SI side
- Easily deployed and tested



Overloading results

- **Introduced in 2023:** Started as a test and progressively extended to most resources.
  - Deployed on approximately ⅔ **of suitable resources**.
  - CERN resources hosting **Tier-0 tasks** are excluded for stability.
- **Performance Metric:** Pilot CPU utilization measured as (Sys + User CPU) / (Walltime × Ncores).
  - Enables precise computation of **full pilot CPU efficiency**.



Daily CPU efficiency (last month): Tier-1s

Significant and systematic improvement observed

Towards HL-LHC-scale workflow management

Preparing for the future

Towards a Sustainable Workflow Management for HL-LHC

Workflows

Tier 0

Production
(WMAgent)

Analysis
(CRAB)

CMS Connect

Submission Infrastructure
A. Perez-Calero, M. Mascheroni

Access Points

Central
Manager

Jobs (SI)

GlideinWMS
Frontend

GlideinWMS
Factory

Execute
Points

Glideins

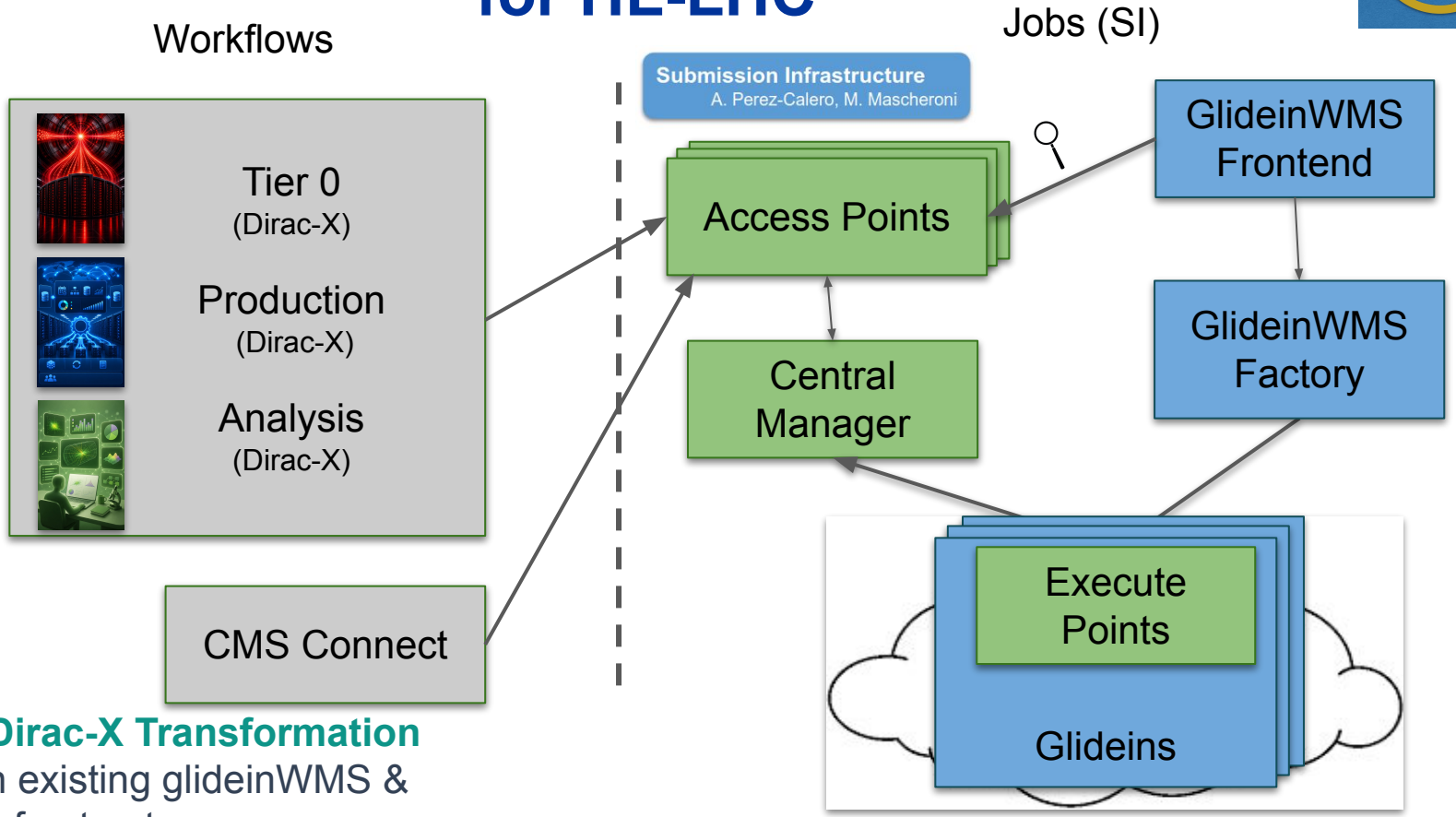
Resource constraints drive the **convergence** toward a **unified workflow solution** for future operations.

Evaluating future Workflow Management solutions

- **WM Review** in 24/25 concluded that the current system is not viable for HL-LHC
  - Recommended to consider PanDA, DIRAC-X, or revamping the current system
- Several round of investigations with various group and people involved
- **Consensus emerged** during Spring 2026 Offline and Computing week

DIRAC-X	Revamped WM	PanDA
• Community solution	• Micro-agent architecture	• Evaluated during review
• Preferred direction	• Fallback solution	• Not actively pursued

Submission Infrastructure as a foundation for HL-LHC



Integrating **Dirac-X Transformation system** with existing glideinWMS & HTCondor infrastructure.

Conclusions

10 Years of SI Evolution

- Integrating **heterogeneous resources**
- Improved utilization: **IO-slots** & **pilot overloading**
- Workloads in **common HTCondor pools**

Proven at Scale

- Grew from **100k to 500k cores**
- Federates WLCG, HPC, and cloud
- Supports Tier0, production, & analysis **simultaneously**

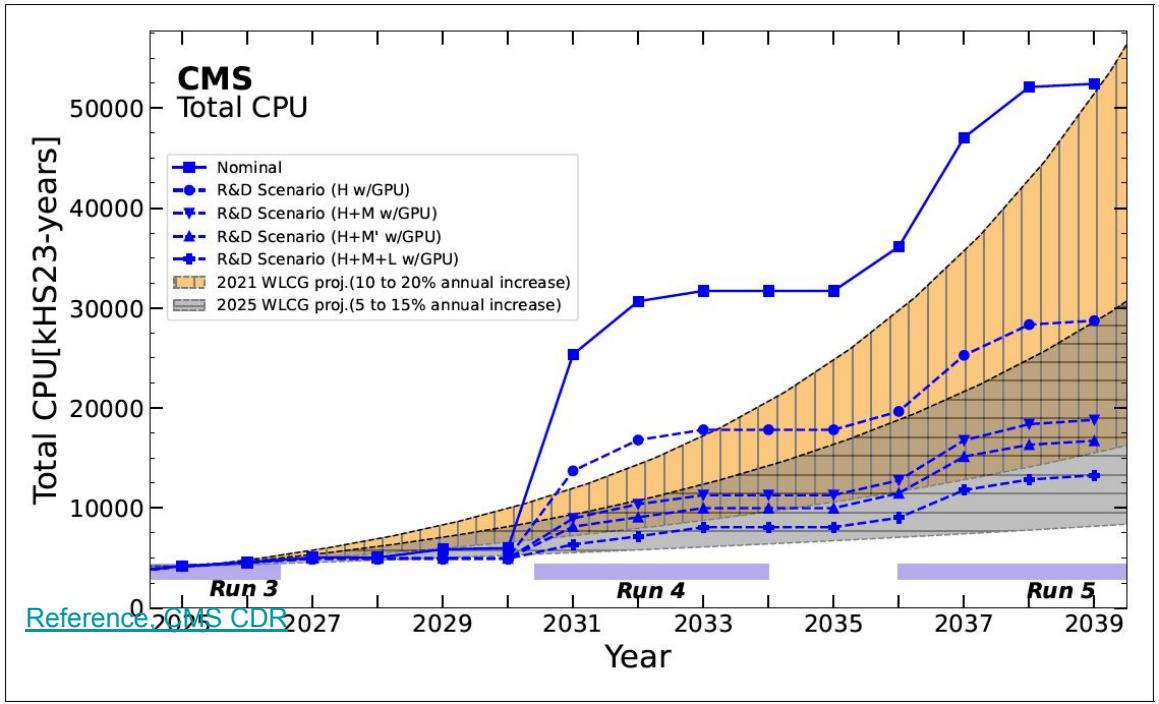
Towards HL-LHC

- Preserve **Global Pool** model
- Build on **community** solutions leveraging **Dirac-X** transformation system
- Next-gen WMS via **GlideinWMS & HTCondor**

Additional slides

The HL-LHC Challenge for CMS Submission Infrastructure

The HL-LHC challenge for the CMS SI: continuous evolution enabling further **scalability** and **resource heterogeneity**, while maintaining the demonstrated stability, flexibility and **efficiency**



Reference: CMS CDR

The SI vision for HL-LHC:

- **Build from the current system** that has **served CMS effectively for more than a decade**
- **Expand and adapt** to **more and new resource types**, diverse **allocation policies** and **tools**, etc

Sources of CPU Inefficiency

- Payload Inefficiencies
  - Bootstrapping and staging
  - I/O-bound jobs
    - Either **heavy I/O** jobs or jobs that use **remote reads**
  - User code (CRAB jobs)
  - StepChain (vs TaskChain): Multiple executables linked together as a single payload job
    - Pro: less jobs to manage, reduce intermediate data storage and transfers. 10x **faster turnaround**.
    - Con: diverse resource needs leading to **inefficiencies**
- Scheduling Inefficiencies
  - Non-standard requirements for jobs
    - System optimized for **2GB per core of RAM and 8 hours** of walltime
  - **Limited pilot lifetime**: draining and defragmentation

Valid reasons for inefficiencies, hard to reduce often.

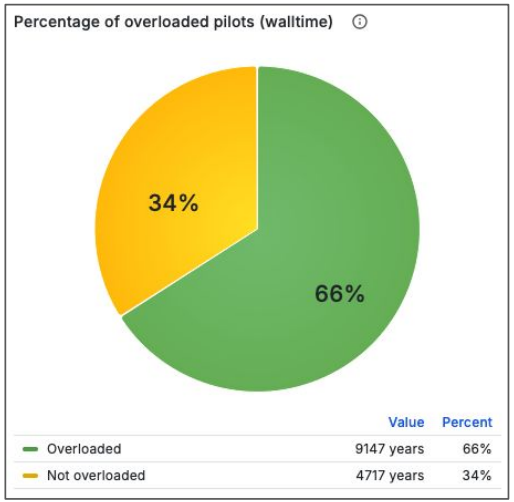
- **Scheduling efficiency typically >95% level for stable sites (T1s and big T2s)**

CMS Connect

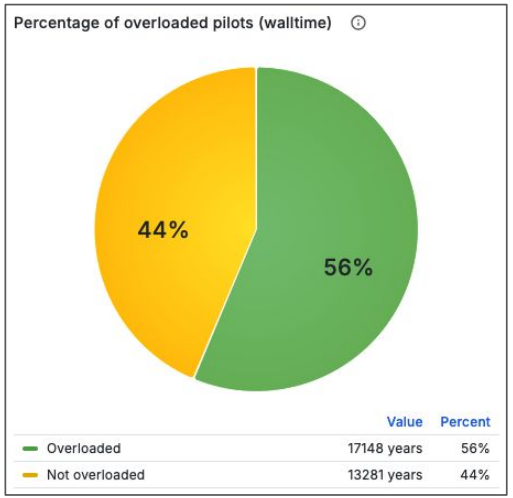
- Light interface for end user that allows to send plain condor jobs
- E.g.: Detector calibration studies not suitable for CRAB

Overloading results

Overloading results



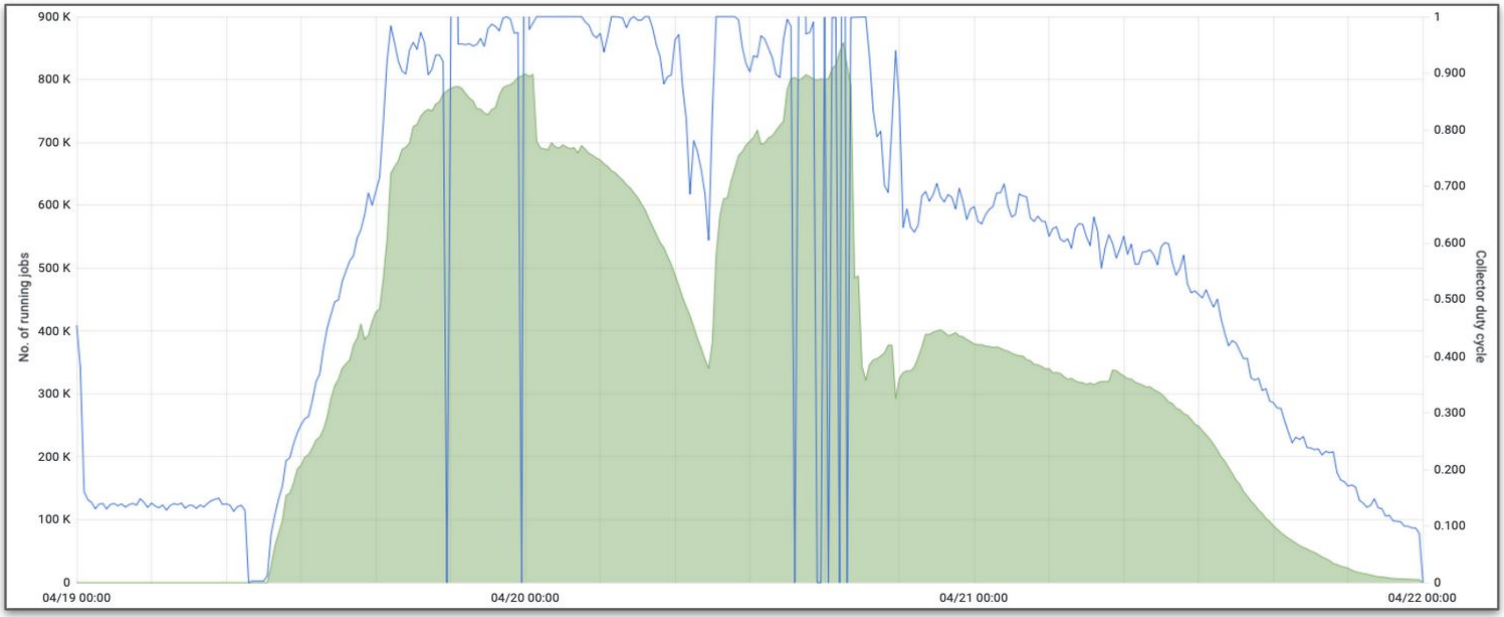
Percentage of overloaded pilots for T1s



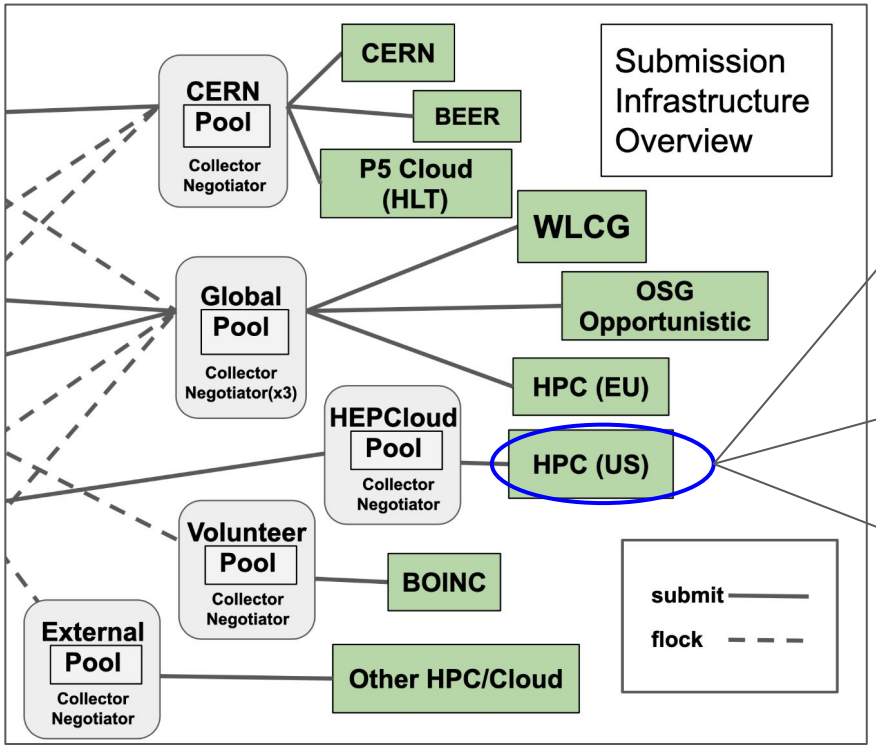
Percentage of overloaded pilots for all sites

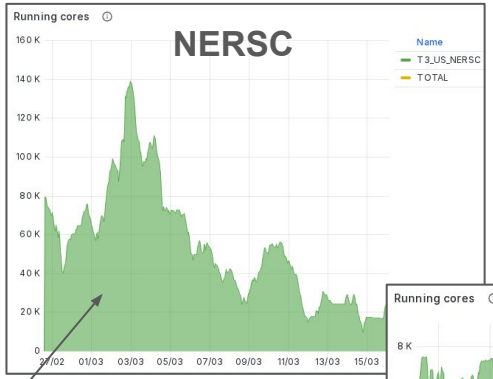
Scalability tests

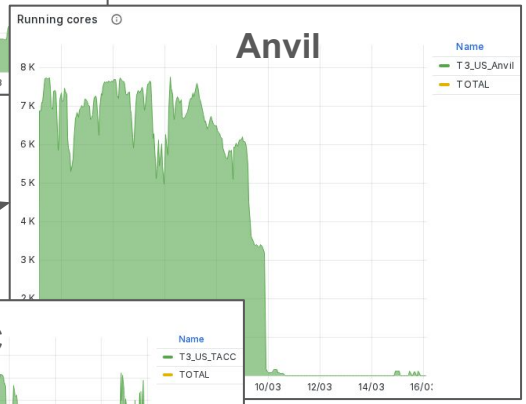
- Pushed the scalability of our Global Pool to about **800k simultaneously running jobs**
  - Factor of 8 away of current pool size considering 4 core jobs

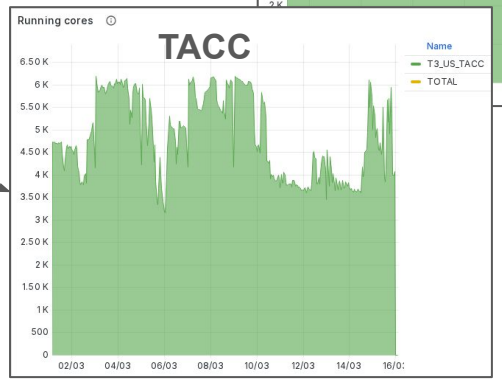


HPC Integration









Integration of GPU resources in SI

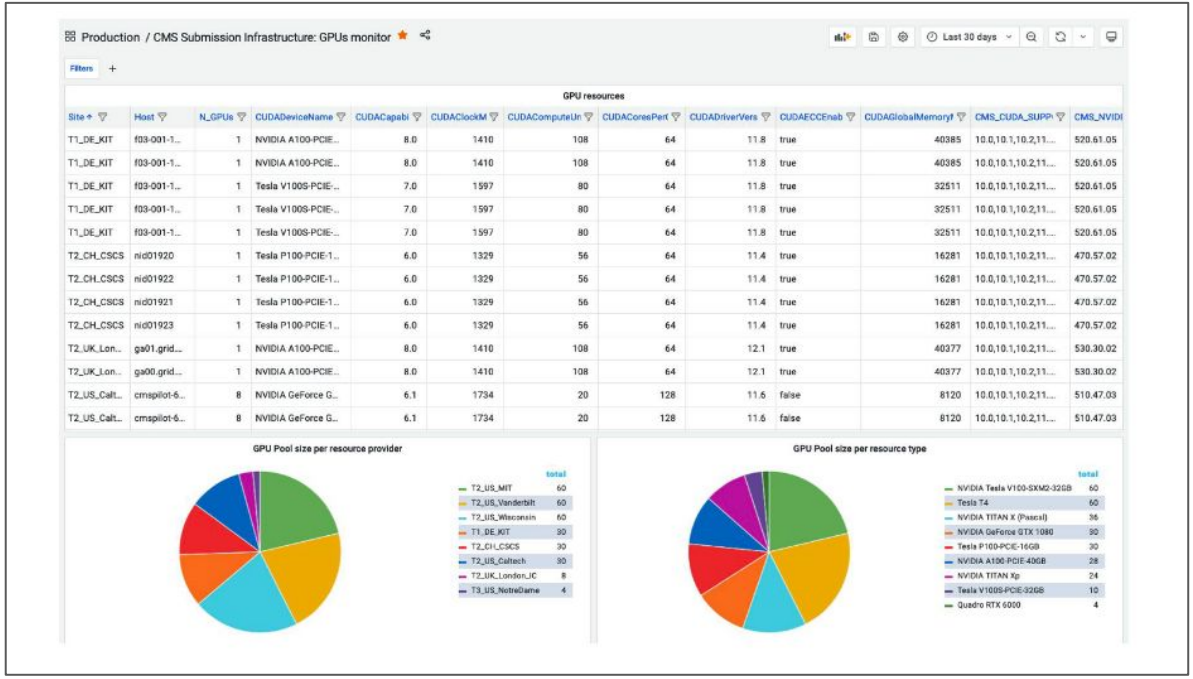
- **GPU resources** in the CMS global pool since ~**2022** (see CHEP23 ref)

The Global Pool FE submits **scout pilots** to GPU-like CEs present in our pilot factories

HTCondor GPU discovery tool allows to **gather information** about the devices

The SI team has built an internal catalogue of GPU slots to inform users about “what is available and where”

Allows users to perform a fine-grained matchmaking based on device properties



Event Rates comparison (I)

- Compared event rate results for all workflows for several months, **classifying jobs by execution site** and workflow type.
- First example, notice this full **StepChain simulation workflow** (~450k jobs in total)
  - Results: event processing rates present **high variability**, with **overloading effect** on throughput **smaller than dispersion** between jobs at the same site, and across sites