Improving CMS CPU Efficiency through Strategic Pilot Overloading

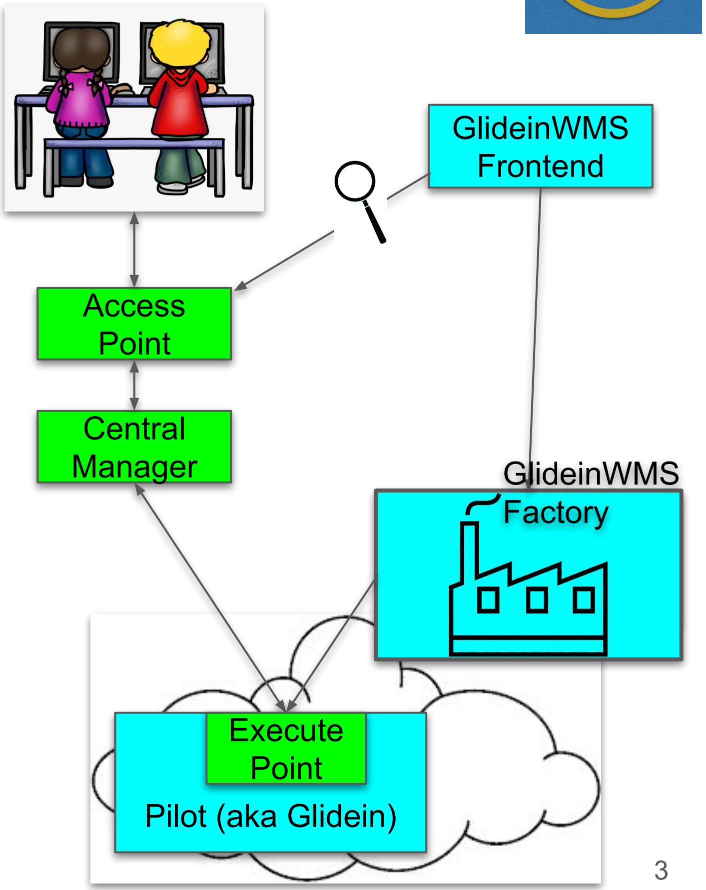
M. Mascheroni, A. Pérez-Calero Yzquierdo,
F. Von Cube, V. Zokaite, H. Kim, F. Würthwein
for the CMS Collaboration

Outline of the talk

- The CMS Submission Infrastructure
  - GlideinWMS and **pilot** approach
- Efficiency of CMS jobs
  - **Sources** of inefficiencies
  - Recovering CPU cycles through **pilot overloading**
- Overloading in action
  - Preliminary results
  - Overloading deployed in **production**
  - Impact on **event rates**

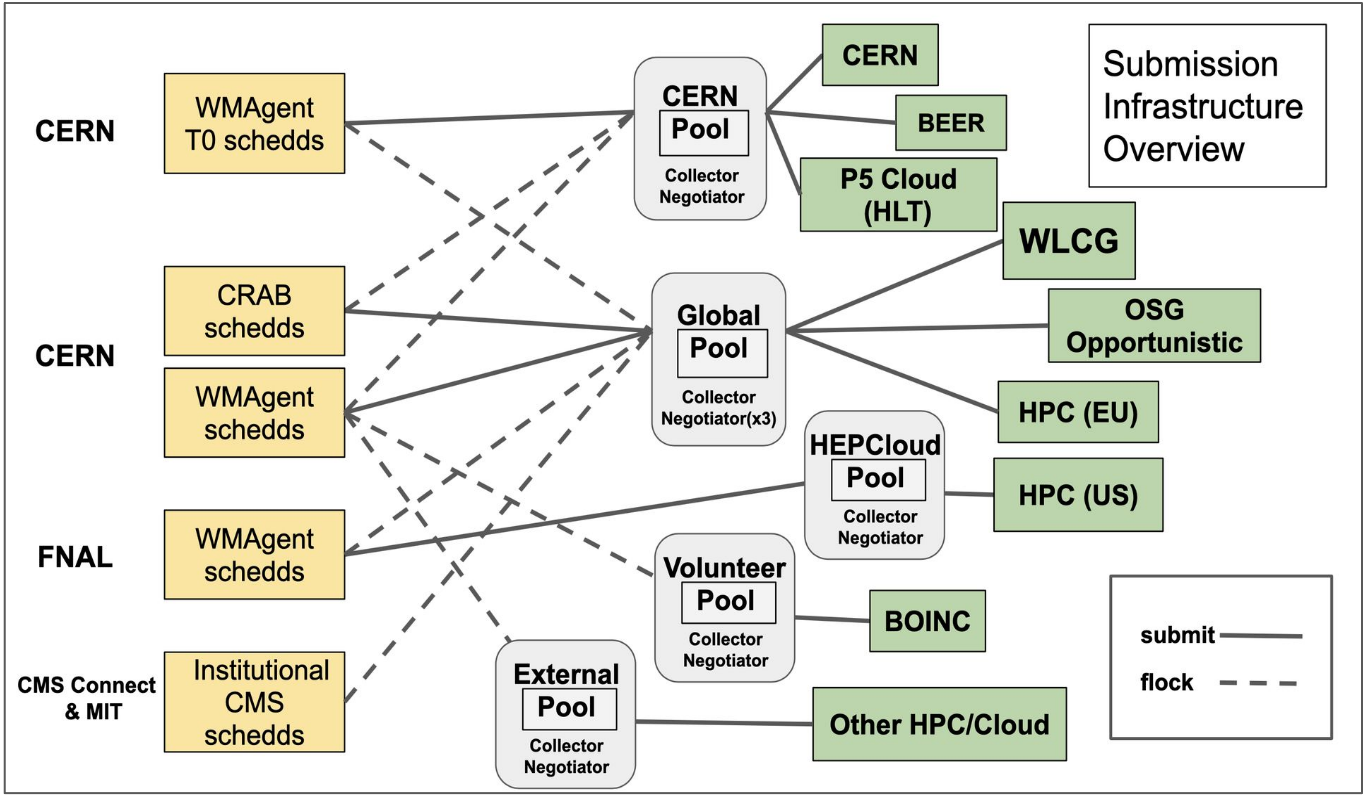
The CMS Submission Infrastructure Group

- Part of CMS Offline and Computing in **charge** of:
  - Organizing **HTCSS** and **GlideinWMS** **operations** in CMS, in particular of the **Global Pool**, an infrastructure where reconstruction, simulation, and analysis of physics data takes place
  - Communicate CMS **priorities to the development teams** of **GlideinWMS** and **Condor**
- In practice:
  - We operate a set of federated pool of resources **distributed over 70 Grid sites, plus non-Grid resources**
  - Join them into a Global Pool of resources managed by HTCondor



The CMS SI: federated HTCondor pools

- Types of access point
- Types of execution point

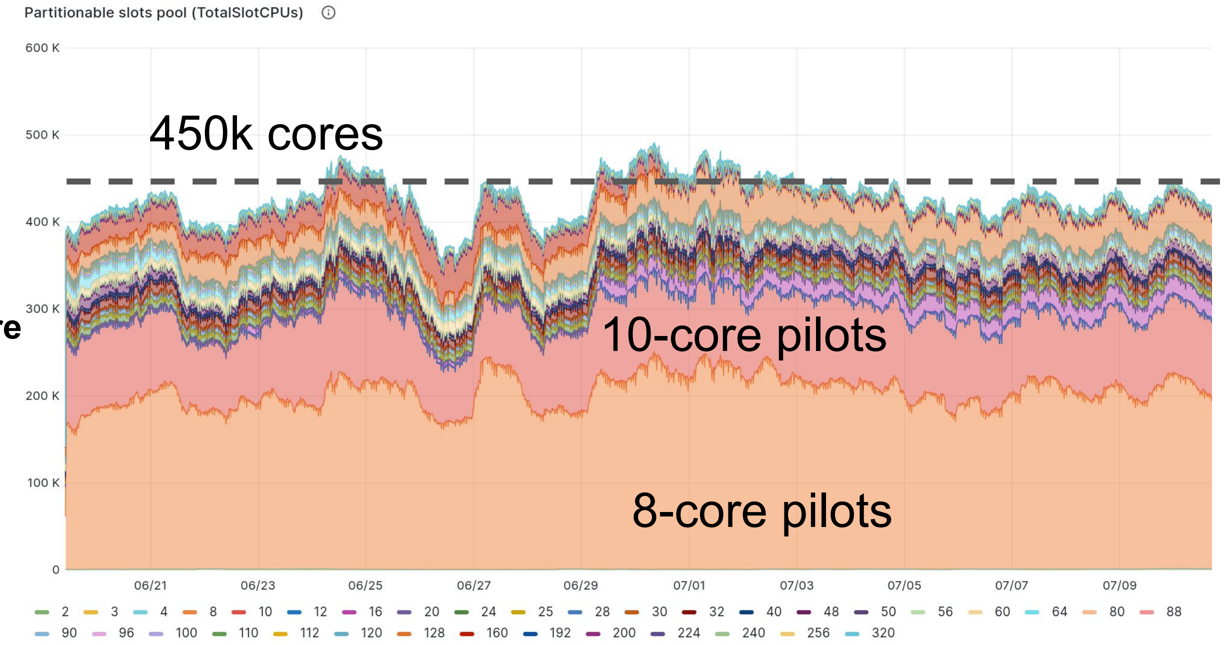


The CMS SI: multicore pilot model

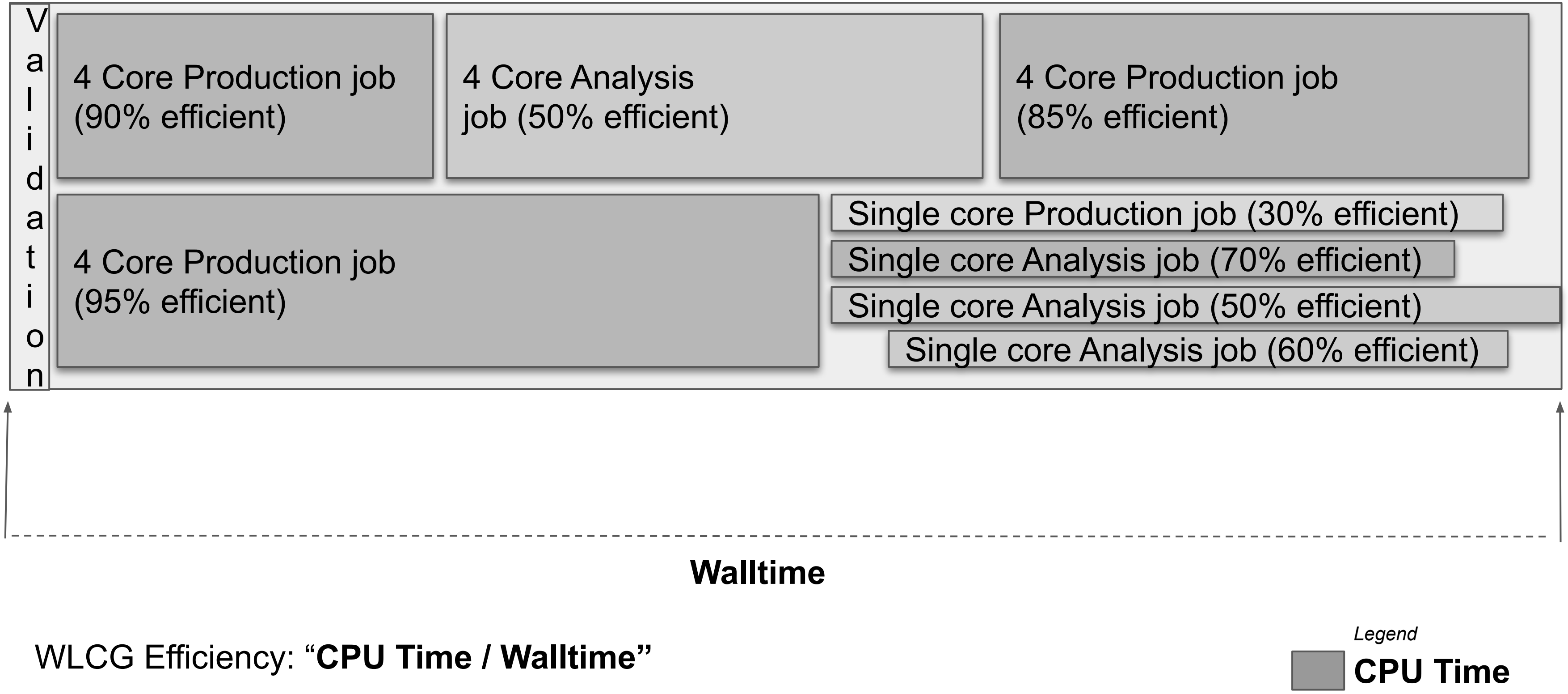
CMS Operates in a **late-binding** model

Acquiring resources for the CMS SI:

- Resources mainly acquired via **8-core pilot jobs** submitted to WLCG sites' CEs
- **Flexibility to use non-standard slots**, e.g.: high-mem, whole nodes, etc

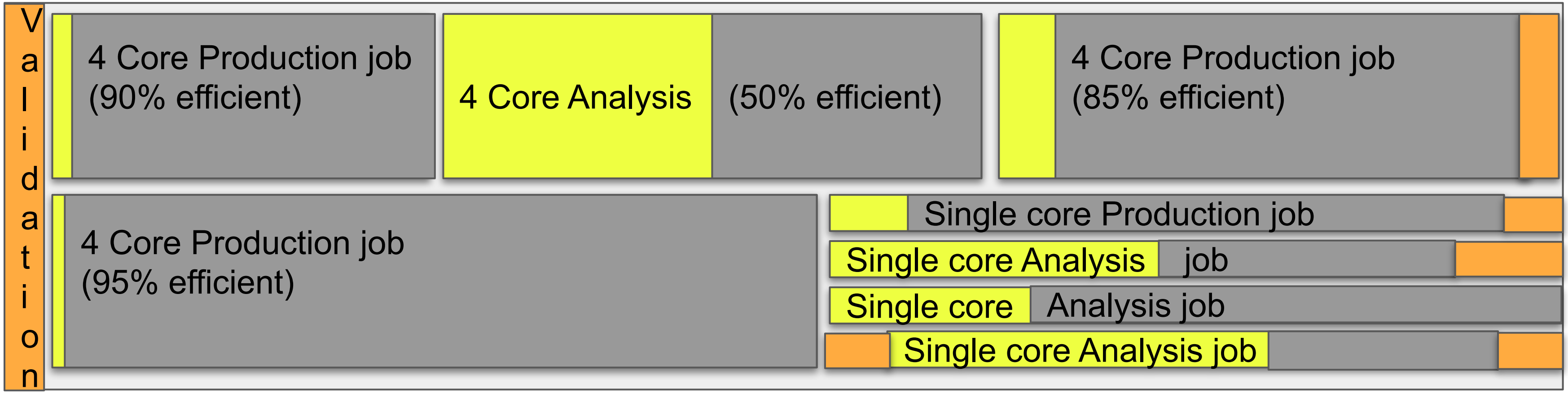


A typical CMS “**pilot** job”: 8-core 48h pilot job executing multiple **payloads**



WLCG Efficiency: “**CPU Time / Walltime**”

A typical CMS “job”: 8-core 48h pilot job executing multiple payloads



- **Efficiency results** observed and reported by our sites to the **EGI accounting portal**
  - That include **scheduling AND payload** inefficiencies
  - They can be factored and **measured independently**

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)**

Can we recover CPU cycles in some other way?

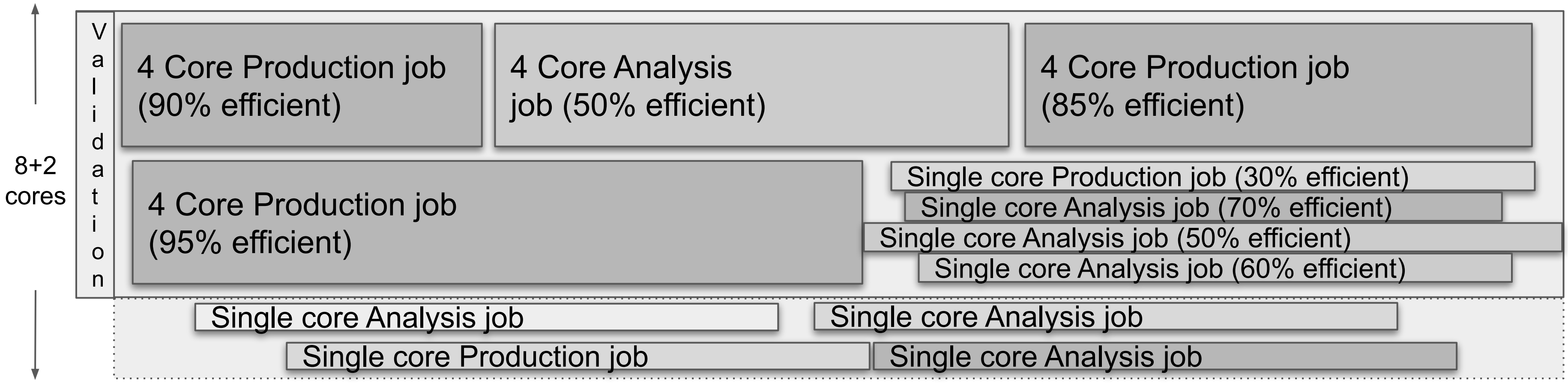
Strategy to recover unused CPU cycles: overloading pilots

Idea: Re-definition of the efficiency problem:

- **Improve CPU utilization efficiency by pushing more workload into the same pilot envelope**
  - Modify pilots so that **they accept more payload jobs into the same resources**
  - **Trivial to implement and test from CMS Submission Infrastructure side**

Principle: we want to recover unused CPU, **not gain opportunistic cycles!**

- **Moderate overloading**: add 25% extra CPU cores and memory to the nominal values of our standard 8-core pilot. Provides 2 extra cores, e.g. available to run additional CRAB or production payload

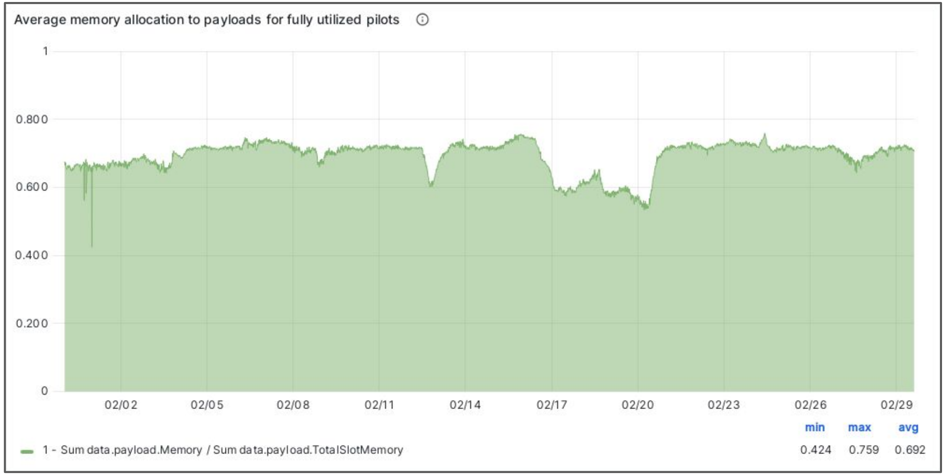


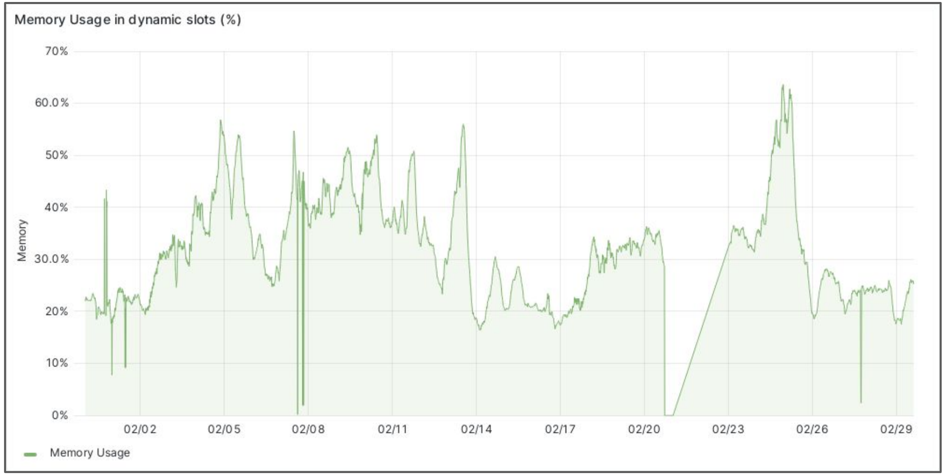
Available memory for overloading pilots

Do we have enough memory available in the pilots to make moderate overloading work? Analyse memory use for fully used pilots at Tier-1s (e.g. 30 day plots):

- Typically, at least 20% of the partitionable slot memory remains unscheduled for fully occupied pilots
- Then, for dynamic slots running the payload jobs, the average memory utilization is typically below 50%

There is no memory constraint for a moderate overloading strategy (e.g. +25%)

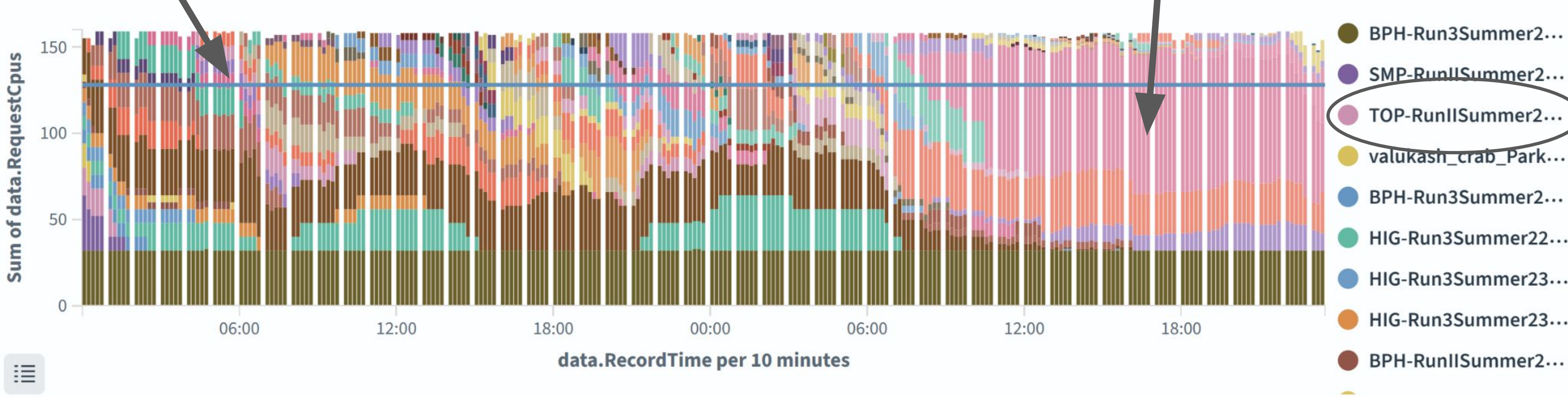




Overloading: whole node slot real example

	Links	Type	Workflow	WMAgent Request Name	Cpu Efficiency Outlier	Cpu Eff	Non Eviction Eff	Eviction Aware Eff Diff	Schedule Eff
+	McM - PMon - Unified	production	TOP-RunIISummer20UL16wmLHEGENAPV-00671	cmsunified_task_TOP-RunIISummer20UL16wmLHEGENAPV-00671__v1_T_240126_100411_2642	0	75.05%	76.41%	1.36%	98.19%

128 cores



128 cores pilot at FNAL overloaded up to 160 cores

Some results

Promising results already in early 2023

Initial test with **three sites** overloading only **one CE each**

[Link](#)

Resource Centre PIC — CPU Efficiency (%) by Submit Host and Month (Custom VOs)

Submit Host	Jan 2023	Feb 2023	Mar 2023	Apr 2023	May 2023	Jun 2023	Jul 2023	Aug 2023	Sep 2023	Oct 2023	Nov 2023	Dec 2023	Jan 2024	Total
ce13.pic.es:9619/ce13.pic.es-condor	71.58%	66.12%	77.66%	90.47%	86.59%	106.37%	95.59%	91.83%	71.93%	91.76%	113.82%	77.65%	88.38%	87.86%
ce14.pic.es:9619/ce14.pic.es-condor	71.58%	65.44%	75.97%	80.14%	74.64%	84.7%	80.64%	78.43%	58.3%	78.96%	91.25%	65.71%	73.25%	75.4%

[Link](#)

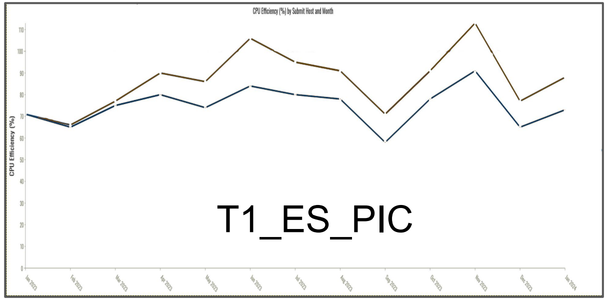
Resource Centre CIEMAT-LCG2 — CPU Efficiency (%) by Submit Host and Month (Custom VOs)

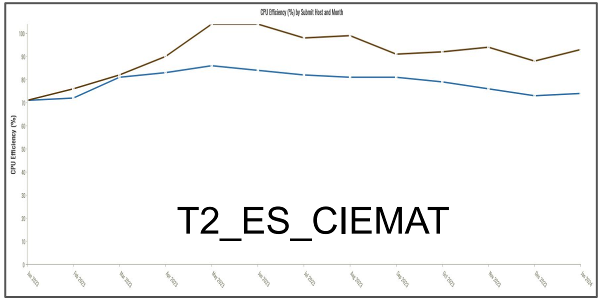
Submit Host	Jan 2023	Feb 2023	Mar 2023	Apr 2023	May 2023	Jun 2023	Jul 2023	Aug 2023	Sep 2023	Oct 2023	Nov 2023	Dec 2023	Jan 2024	Total
condorce1.ciemat.es:9619/condorce1.ciemat.es-condor	71.06%	76.15%	82.41%	90.45%	104.08%	104.05%	98.85%	99.25%	91.56%	92.59%	94.57%	88.52%	93.74%	92.94%
condorce2.ciemat.es:9619/condorce2.ciemat.es-condor	71.01%	72.86%	81.36%	83.56%	86.01%	84.36%	82.8%	81.33%	81.61%	79.01%	76.46%	73.13%	74.97%	79.72%

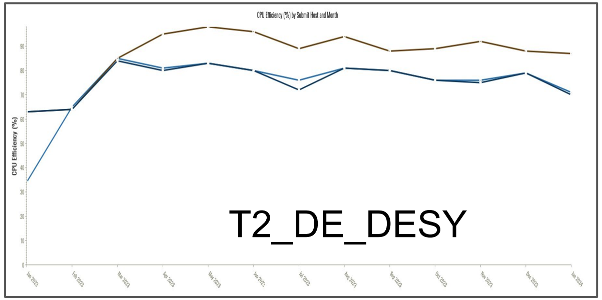
[Link](#)

Resource Centre DESY-HH — CPU Efficiency (%) by Submit Host and Month (Custom VOs)

Submit Host	Jan 2023	Feb 2023	Mar 2023	Apr 2023	May 2023	Jun 2023	Jul 2023	Aug 2023	Sep 2023	Oct 2023	Nov 2023	Dec 2023	Jan 2024	Total
grid-htcondorce0.desy.de:9619/grid-htcondorce0.desy.de-condor	63.75%	64.6%	84.83%	80.32%	83.63%	80.76%	72.91%	81.03%	80.66%	76.58%	75.03%	79.45%	70.95%	76.37%
grid-htcondorce1.desy.de:9619/grid-htcondorce1.desy.de-condor	63.01%	64.44%	85.31%	95.24%	98.57%	96.76%	89.58%	94.53%	88.8%	89.18%	92.19%	88.34%	87.73%	87.87%
grid-htcondorce2.desy.de:9619/grid-htcondorce2.desy.de-condor	34.23%	65%	85.19%	81.33%	83.52%	80.73%	76.68%	81.9%	80.86%	76.24%	76.28%	79.21%	71.67%	78.37%

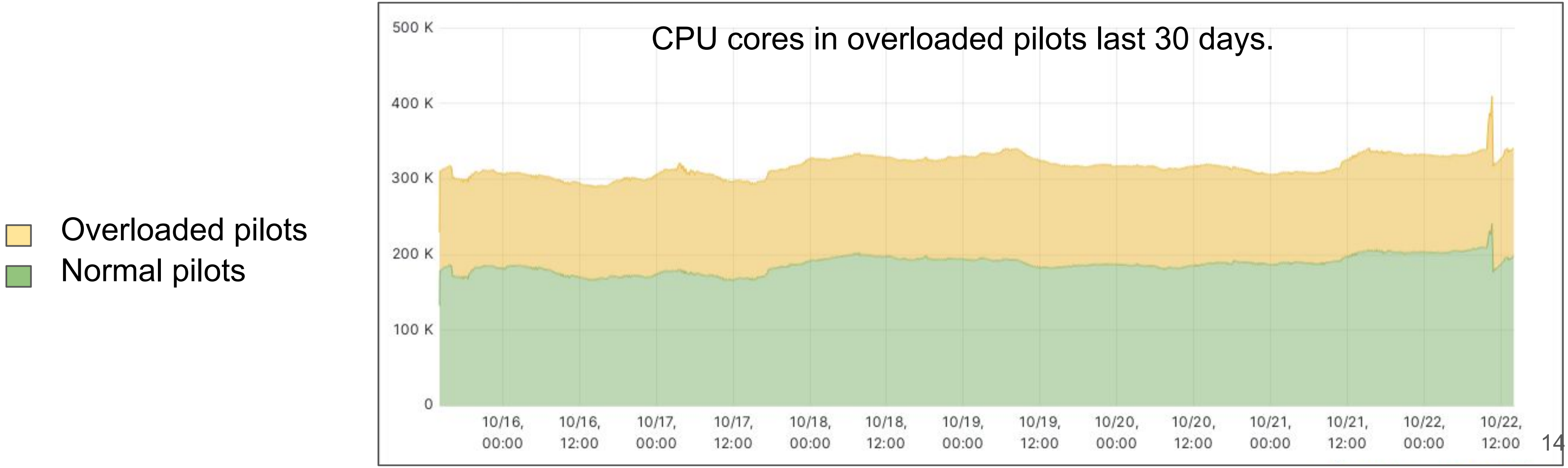






Overloading pilots expansion

- After promising initial results, CMS decided to enable overloading at more resource providers starting in January 2024:
  - All **Tier-1 sites**
  - **A set of good Tier-2s** (average scheduling efficiency already at 95%)
- Still kept ~50% unchanged for each site in order to compare results



Efficiency Improvements

- From the **pilot logs**, total walltime and used CPU time can be extracted.
    - **Calculate CPU efficiency** as measured and reported by the resource providers

- Performance difference is noticeable when comparing **overloaded and non-overloaded pilots**
    - **Significant improvement** in CPU utilization
    - No observed effect on job failure rates

Site Name	Normal Efficiency	Overloaded Efficiency	Efficiency increase
T1_FR_CCIN2P3	84.57%	95.78%	11.21%
T1_IT_CNAF	79.81%	84.62%	4.81%
T1_UK_RAL	72.41%	85.00%	12.60%
T2_BE_IIHE	77.21%	89.17%	11.96%
T2_DE_RWTH	55.68%	78.01%	22.33%
T2_EE_Estonia	73.81%	85.47%	11.66%
T2_ES_CIEMAT	72.65%	88.40%	15.75%
T2_IT_Bari	74.48%	78.87%	4.39%
T2_IT_Legnaro	75.77%	85.88%	10.11%
T2_IT_Rome	60.25%	73.38%	13.13%
T2_UK_London_IC	66.14%	72.94%	6.80%
T2_US_MIT	63.74%	70.61%	6.87%

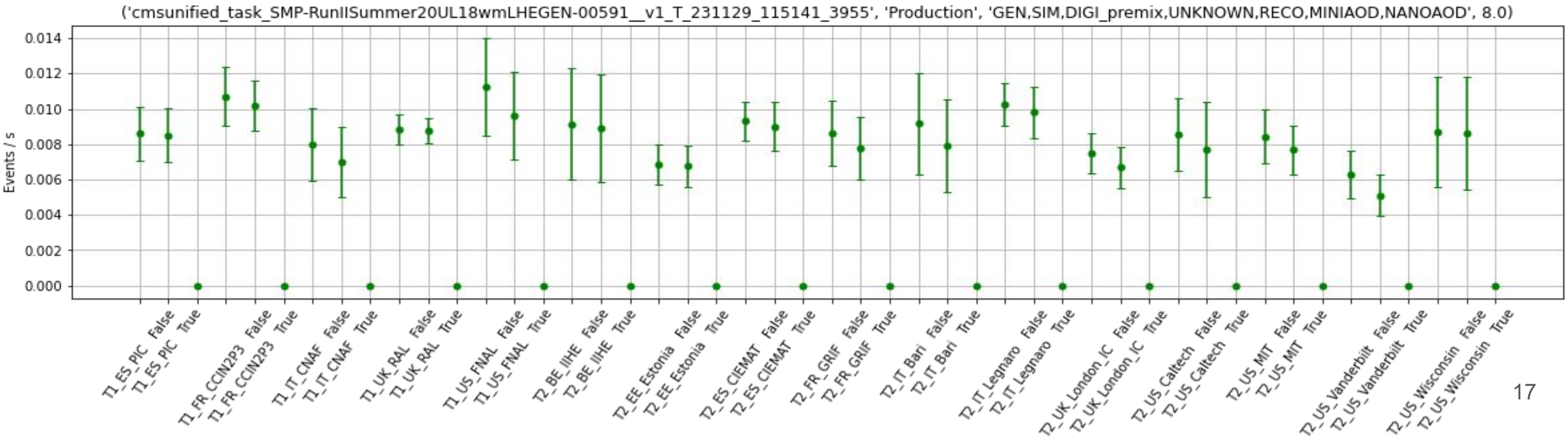
Past three months results

Impact on event rates (I)

Evaluated the impact of pilot overloading on event processing rates for diverse CMS workload types and sites. Results from **actual execution on the Grid**

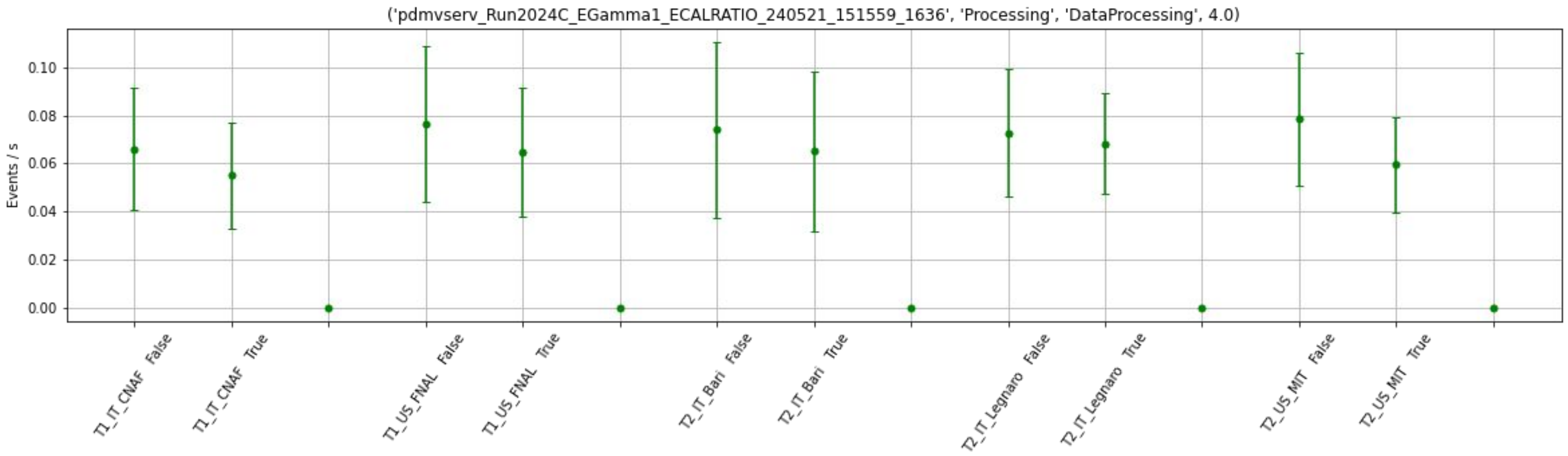
Event Rates comparison (I)

- 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



Event Rates comparison (II)

- Second example from a data processing workflow (~20k jobs)
  - Analogous results



Impact on event rates (II)

Evaluated event rates in a **controlled environment**: executed a **single type of workflow** (MinBias production) in fully loaded pilots for a variety of sites.

Overall impact on fully loaded pilots

Measured total number of events produced by a single pilot running on 8 physical cores as either 8 or 10 processes.

Increased total event throughput. Slower event rate per process is over compensated by running more processes.

Site	Processor	On (Evt/s)	Off (Evt/s)	Evt/s increment %
T2_BE_IIHE	AMD EPYC 7452 32-Core Processor	3,14	2,82	11,35
T2_DE_RWTH	AMD EPYC 7543 32-Core Processor	2,89	2,49	16,06
T2_ES_CIEMAT	Intel(R) Xeon(R) Gold 5318Y CPU @ 2.10GHz	4,31	3,58	20,39
T2_IT_Bari	AMD EPYC 7413 24-Core Processor	2,96	2,81	5,34
T2_IT_Legnaro*	AMD EPYC 7282 16|7413 24-Core Processor	2,98	2,74	8,76
T2_UK_London_IC	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz	1,70	1,61	5,59
T2_US_UCSD	AMD EPYC 7662 64-Core Processor	2,42	1,63	48,47
T2_US_Vanderbilt	Intel(R) Xeon(R) CPU E5-2420 0 @ 1.90GHz	1,30	1,05	23,81

Conclusions

- Moderately **overloading of our pilots** allows CMS to **recover** between 5% to 20% of idle **CPU cycles**
  - Extra 30k cores (re)gained using this strategy
- **No impact** observed from the site perspective on **job error rates**, CPU or memory (ab)use.
- Studies on **event processing rate** has shown:
  - Higher variability between jobs of the same workflow and between sites than an overloading true/false effect
  - Overall event rate increased in dedicated test

Acknowledgements

A. Pérez-Calero Yzquierdo's research supported by projects PID2020-113807RA-I00 and PID2022-142604OB-C21, funded by

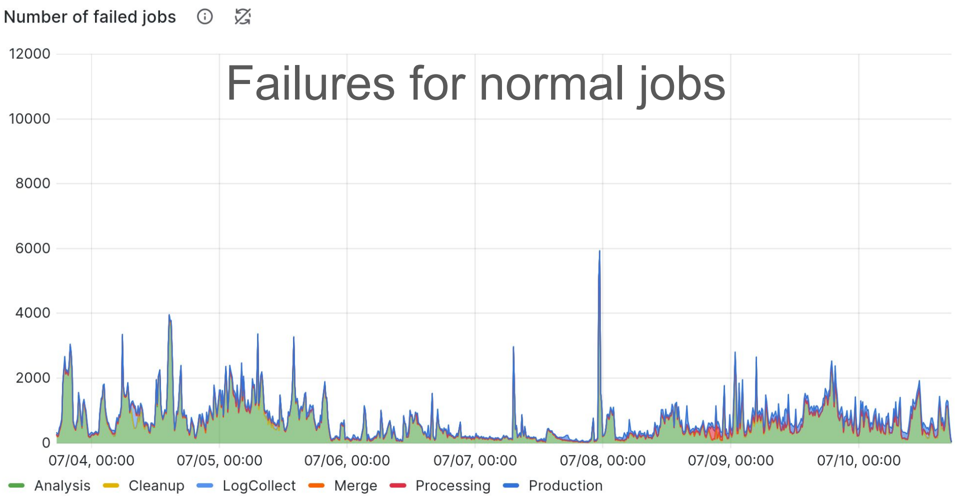
GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
Cofinanciado por la Unión Europea
AGENCIA ESTATAL DE INVESTIGACIÓN

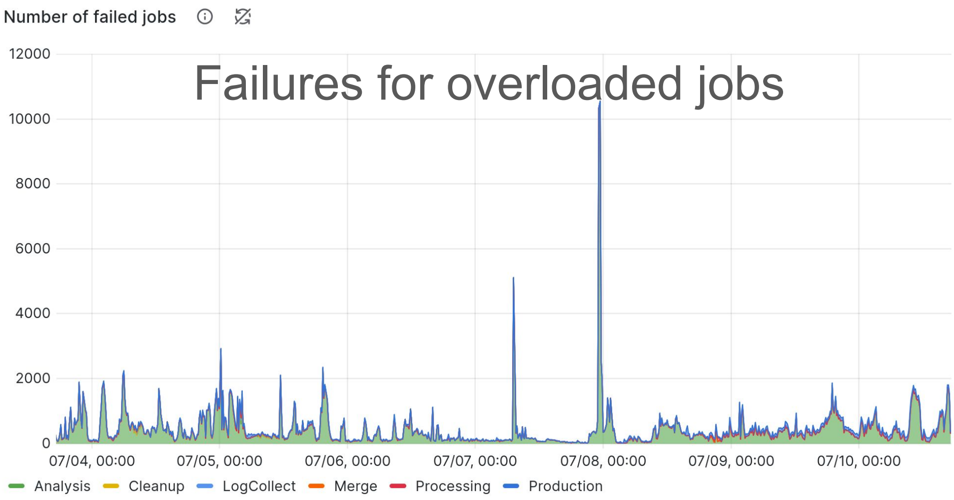
This work was partially supported by the US National Science Foundation under Grant No. 2121686 and under Grant No. 2030508

Backup Slides

Job Failures Comparison

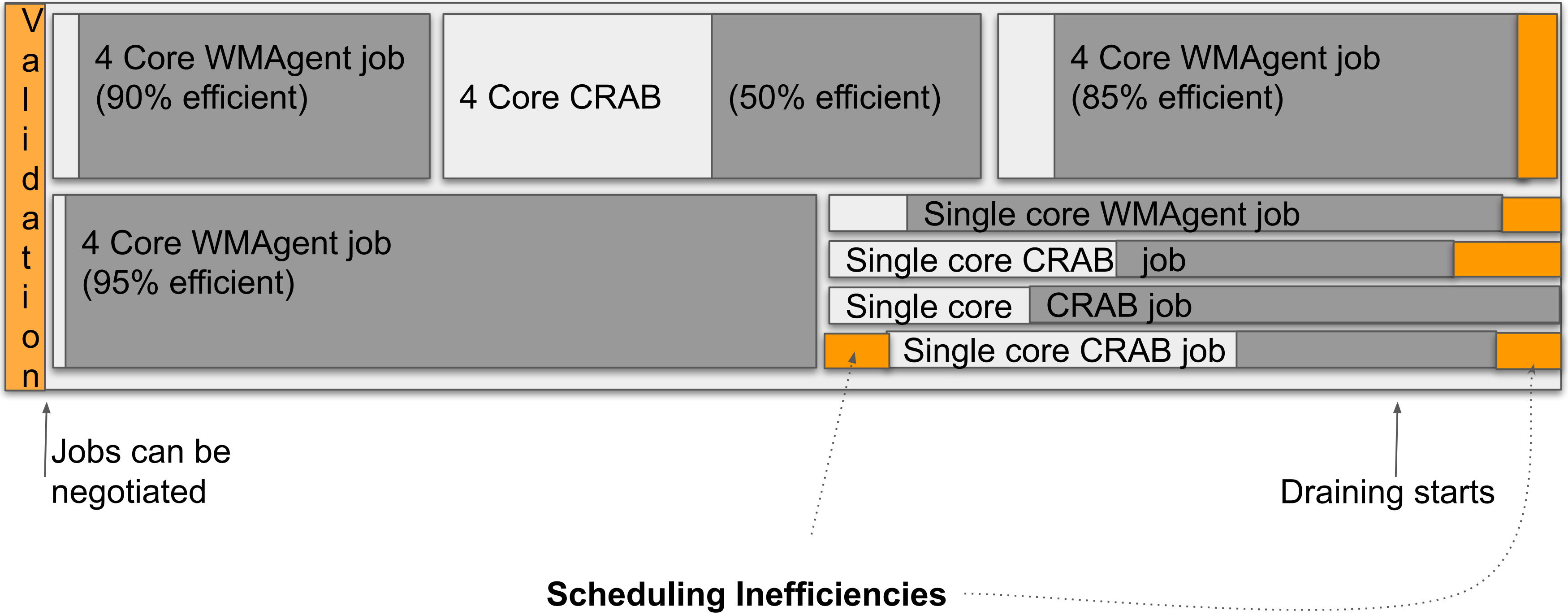
Absolute number of job failures in the last week grouped by job type



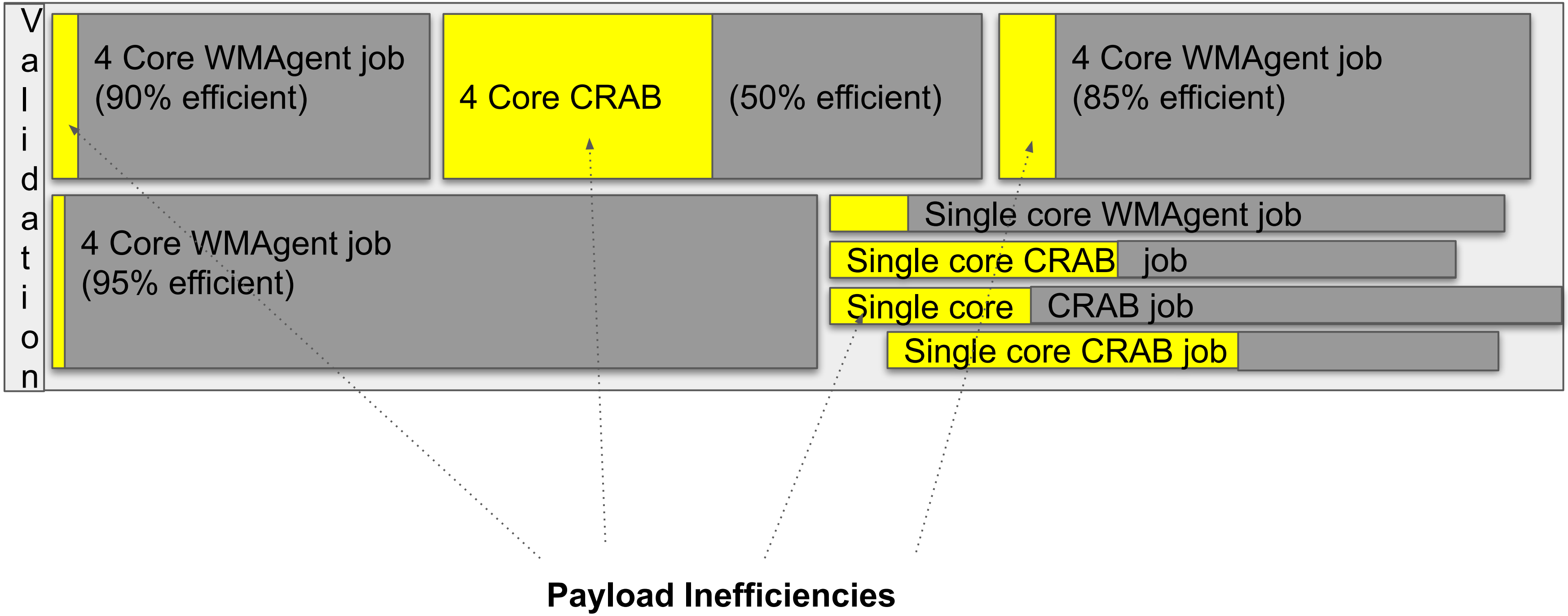


No impact in terms of job failures

A typical CMS "job": 8-core 48h pilot job executing multiple payloads

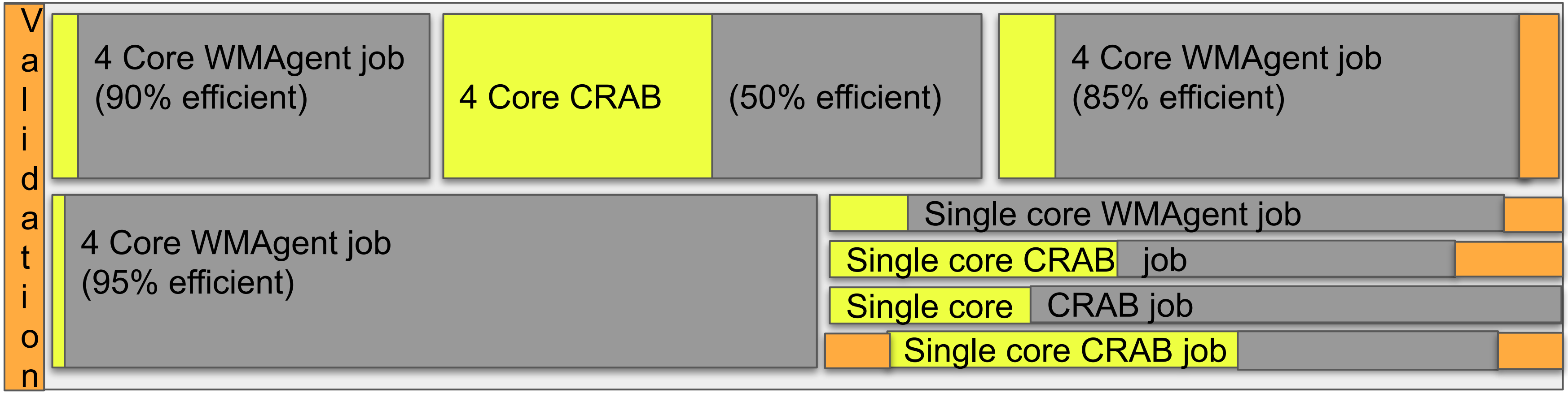


A typical CMS “job”: 8-core 48h pilot job executing multiple payloads



(Uses **payload walltime** as denominator)

A typical CMS “job”: 8-core 48h pilot job executing multiple payloads



- **Efficiency results** observed and reported by our sites to the **EGI accounting portal** include **scheduling AND payload** Inefficiencies

- **They can be factored and measured independently**