Opportunistic usage of the LHCb Trigger Farm

Antonio Falabella, Francesco Sborzacchi, Alexandre Boyer
on behalf of Online and Offline groups

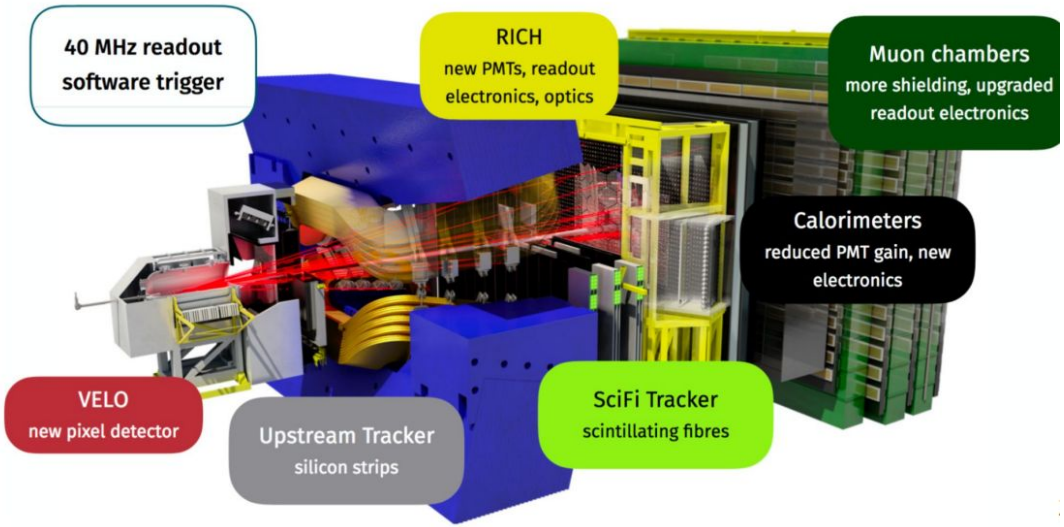
Outline

- The LHCb Experiment and its Run 3 DAQ
- Opportunistic usage of the trigger farm
- HTCondor
- Implementation
- Results and Operations
- Conclusions

LHCb Experiment

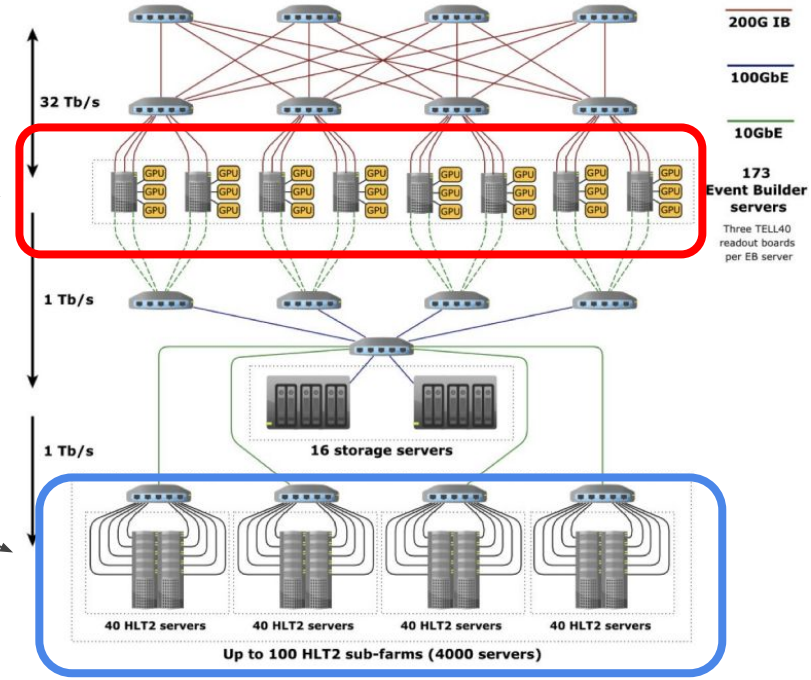
- Experiment dedicated to flavour physics
- **Major upgrade** of all subtectors for Run 3 to collect data at L = 2 X 10^{33} cm^{-2} s^{-1}
- **Average pile up ~ 5**
- Reconstruction at **30 MHz** LHC pp collision rate for the High Level Trigger (HLT)
- **New data acquisition system** and data center

CERN-LHCC-2012-007



Run 3 DAQ

- LHCb raw data (4 TB/s) at 30 MHz
- Level 1 Trigger ~ 500 GPUs NVIDIA RTX A5000 GPUs on EB nodes **:** **30MHz → 1MHz**
- Data sent to 40PB buffer
- **Level 2 ~3000 trigger lines run on 4000 nodes farm**



J. Phys.: Conf. Ser. 878 012012
https://arxiv.org/abs/2105.04031

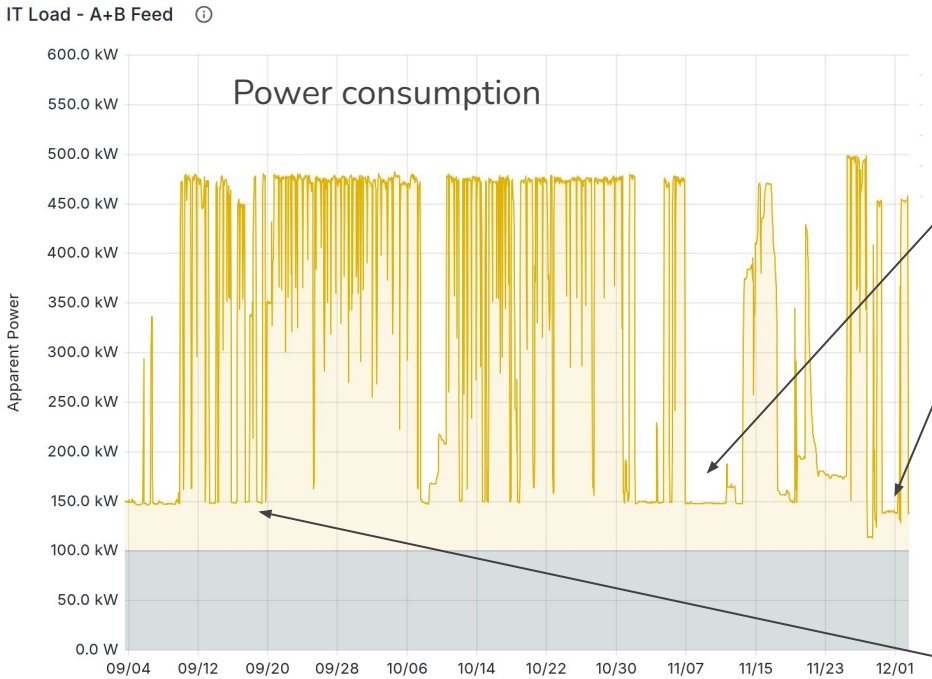
Level 2 Trigger Farm @ Interaction point (P8)

- Level 2 trigger farm consists of ~4.5k nodes
- **~260k** cores of different HS23[3]
- To be compared with Tier0[1] (not pledged to LHCb only)
  - 11k nodes
  - 344k cores
  - 3.2M HS23[2]
- **Level 2 trigger farm more than the whole LHCb Grid CPU resources[3]**

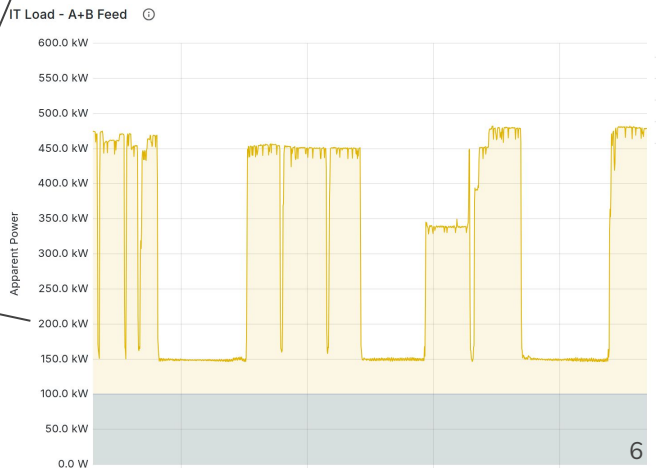
Num nodes	Num cores	HS23 (HS23 per core)
1522	40	382 (9,55)
2090	56	625 (11,2)
565	48	468 (9,75)
44	96	1851 (19,3)
203	256	3819 (14,9)
4424	261232	**3031689**

1) https://monit-grafana.cern.ch/d/000000473/it-overview?orgId=1
2) http://wlcg-cric.cern.ch/
3) https://cds.cern.ch/record/2924639/files/LHCb-PUB-2025-007.pdf
4) https://w3.hepix.org/benchmarking.html

Opportunistic usage



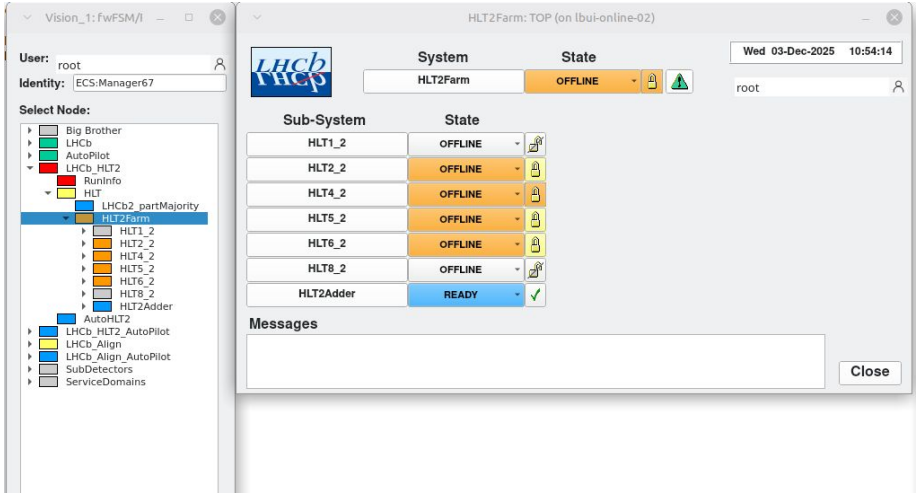
Profit from the time slots when the farm is idle for **few days** to submit other workloads i.e. Grid jobs



Level 2 trigger farm

- The farm consists of ~120 subfarms (40 servers each)
- From the HLT2 operations point of view a STATE is assigned to them
- Transitions between different states are related to HLT2 processing activities
- Possible states are OFFLINE, PAUSED, ERROR, UNKNOWN, PAUSING, RUNNING, READY
- States can be grouped into **Idle states** and **Not Idle states**

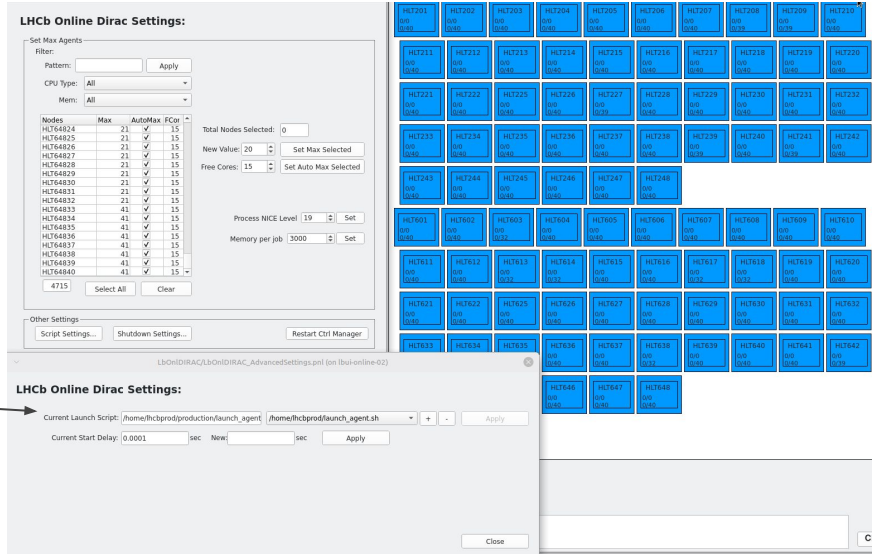
Example of a subfarm



Idle states*	Not Idle states
OFFLINE, UNKNOWN * Green light from RTA outside MD and TS	RUNNING, PAUSED, READY, PAUSING, ERROR

Opportunistic usage in the past

- The usage of the farm to run Grid jobs is not new
- A special **LHCbDirac pilot**[1] code has been developed by offliners to submit simulation jobs on level 2 trigger nodes
- The custom pilot code was started on idle subfarms when a large enough time window was foreseen (i.e. MD) and manually enabled



1) Note: A pilot is script that create the proper environment to run a job and contacts LHCbDirac central services if there are jobs to run. This is the same mechanism used on other grid sites

Improvements of this task

- The **PRIORITY** of the farm is to execute level 2 trigger jobs
  - Reallocate resources as soon as needed by the experiment
- Automate the previous mechanism, face future requirements of job matching (i.e. multicore jobs, user analysis, shared storage)
- Add a LHCbDirac service (Site Director[1]) to submit jobs automatically
- Scale up to the size of the farm
- Enhance monitoring of the opportunistic usage of the farm

1) Site Director: LHCbDirac central service use to submit pilot to batch systems (https://cds.cern.ch/record/2806286)

Improvements of this task

- HTCondor[1] fullfills flexible and automatic resource allocator requests
- Added a 4PB Ceph[2] storage exposed through an xrootd[3] instance (thanks to Alexander Rogovskiy)
- In terms of Grid jargon it is Tier2D site

1) https://htcondor.org/ : Workload management
2) https://docs.ceph.com : A distributed POSIX opensource filesystem
3) https://xrootd.org/ : Low latency and scalable data access

HTCondor

- HTCondor version 24.0.1 (latest as of installation time) on RH9
- An HTCondor cluster is made of different components: *master* (control the status of every components), *manager* (decides where to run a job[1]), *execution point* (nodes that run the payload)
- Cluster configuration using puppet -> thanks to Pierfrancesco Cifra
- Inclusion of the subfarms easily controlled by puppet configuration
- Monitoring using prometheus exporter (tool to publish metrics: i.e. num jobs, cpu load ...)

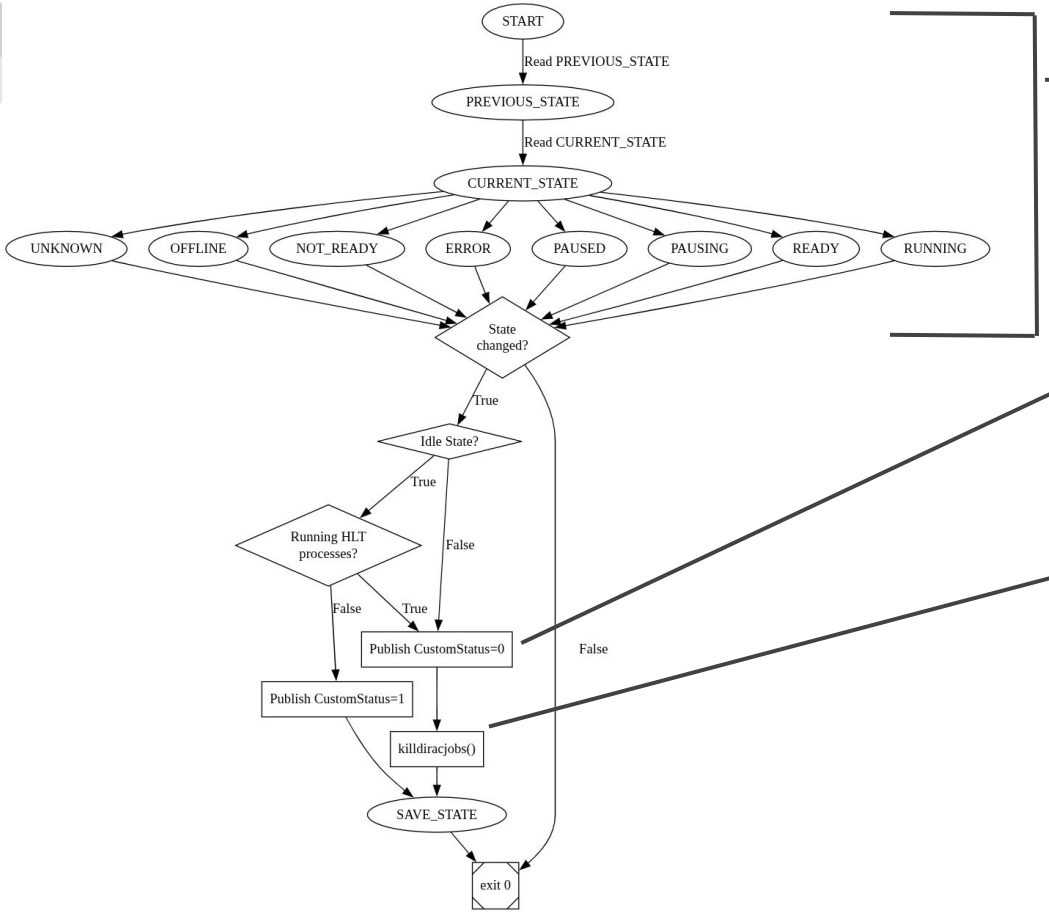
https://https://htcondor.readthedocs.io/en/main/admin-manual/introduction-admin-manual.html.org/

1) In HTCondor terms a job is pilot in LHCbDirac terms

ClassAds

- To match resources to submitted jobs HTCondor implements ClassAds (*Classification Advertisements*):
  - Characteristic (owner of the job, memory required) or constraints (num cores...) of the job and resources are exposed through *ClassAds*
- HTCondor uses a built-in ClassAd (STARTD) to decide when a node can run a job
- If the STARTD of the nodes evaluates to TRUE than it can run jobs
- ClassAds can also be a custom feature defined by the administrator, for example by mean of a script or a program
  - Documentation: https://htcondor.readthedocs.io/en/23.0/admin-manual/daemon-cron.html
- In this task I developed a script (**CustomClassAd**) that produces a customised ClassAd (**CustomStatus = 1 or 0**) that extracts additional information in order to determine if the Hlt2 STATE is in an **Idle** or **Not Idle state**)
- The final condition to enable job submission is ($(STARTD) && CustomStatus == 1) == TRUE
  - This is evaluated on every node of the cluster

CustomClassAd logic



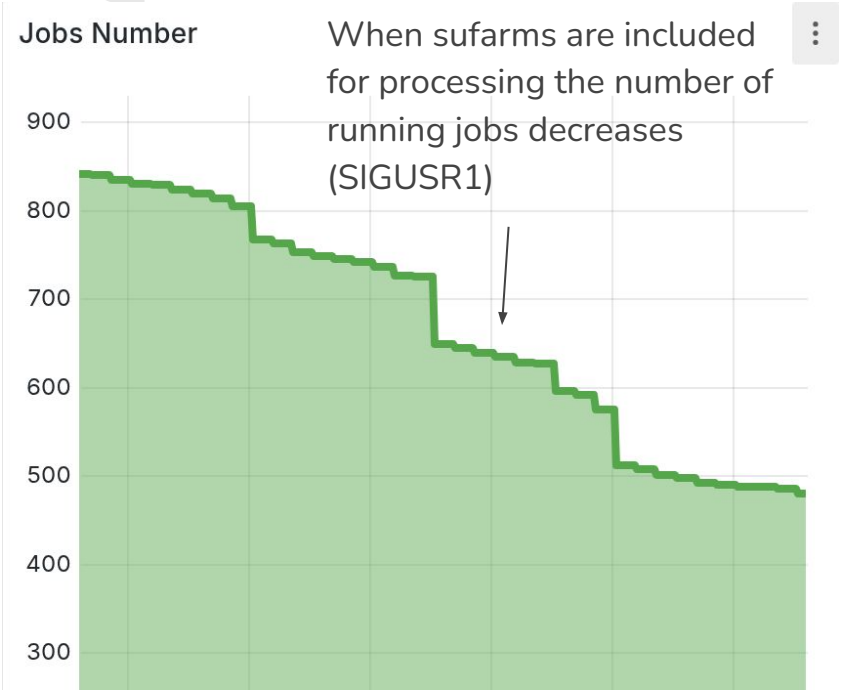
- **CustomClassAd** runs inside HTCondor as an external module and determine the **transition among** Hlt2 statuses of the machine
- if the state **transitioned** for **Idle** to **Not Idle** or viceversa then CustomStatus is evaluated to 0 or 1
- If the state switches to a not Idle state grid jobs are sent a linux **SIGUSR1** signal that is understood by gaudi and let current event complete (killdiracjobs function)

LHCbDirac integration

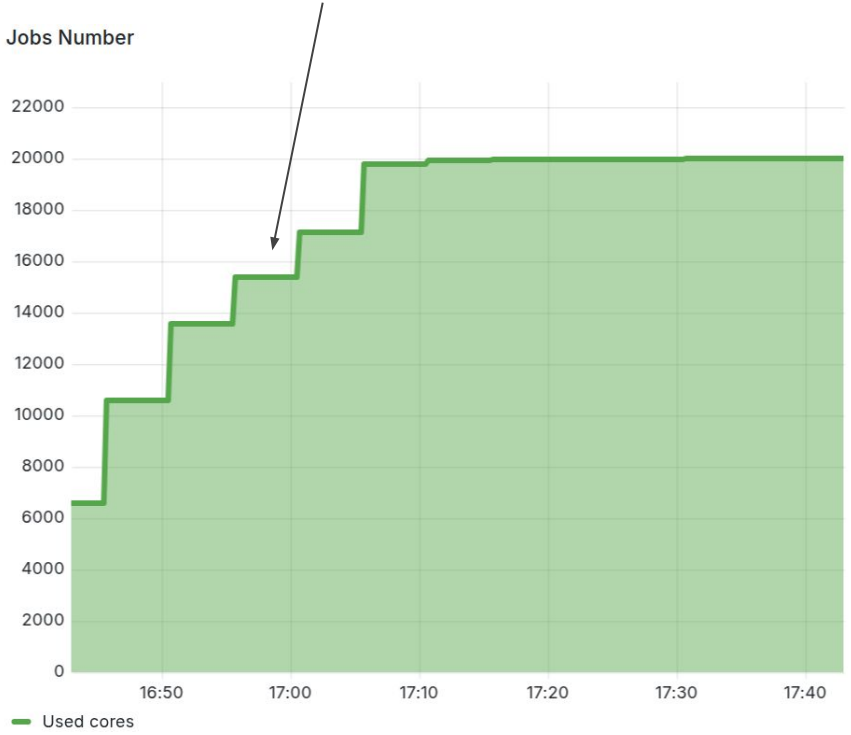
- LHCbDirac Site Director to submit pilots automatically (thanks to Alexandre Boyer)
- The number of jobs that can be submitted is customisable
  - At the beginning we configured the Site Director to submit 1 job per pilot but after some tuning we chose 8 single core per pilot to fill the farm faster
  - A fraction of the farm is dedicated so the minimum number of pilots that can be submitted is set to 2000[1]

1) the number of dedicated nodes is 335 (48 cores each: total 16080)

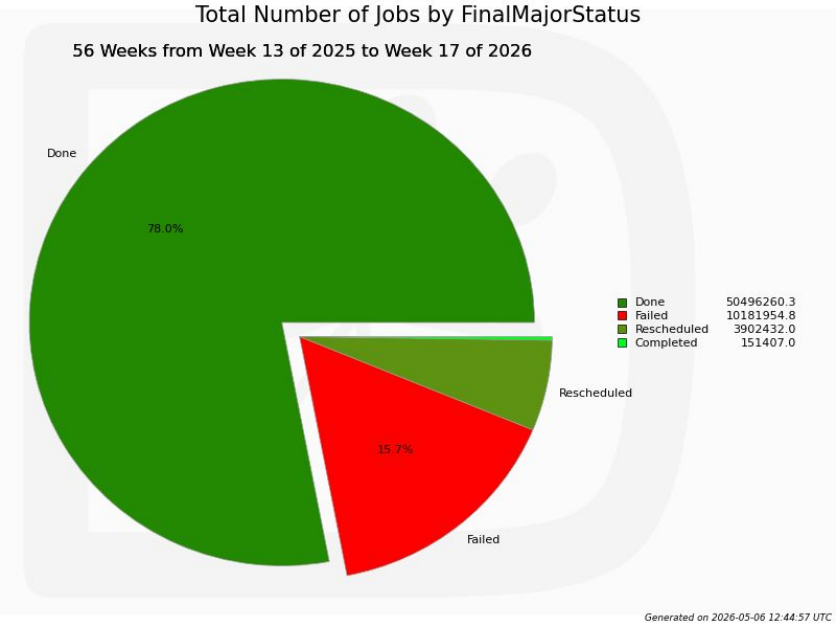
Operations

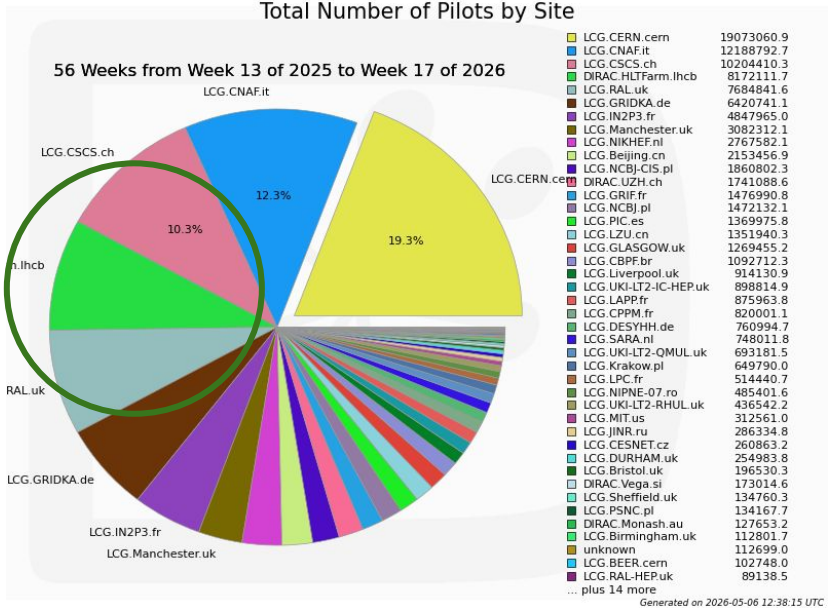


When a subfarm becomes idle the number of submitted pilots increases



The new setup in action





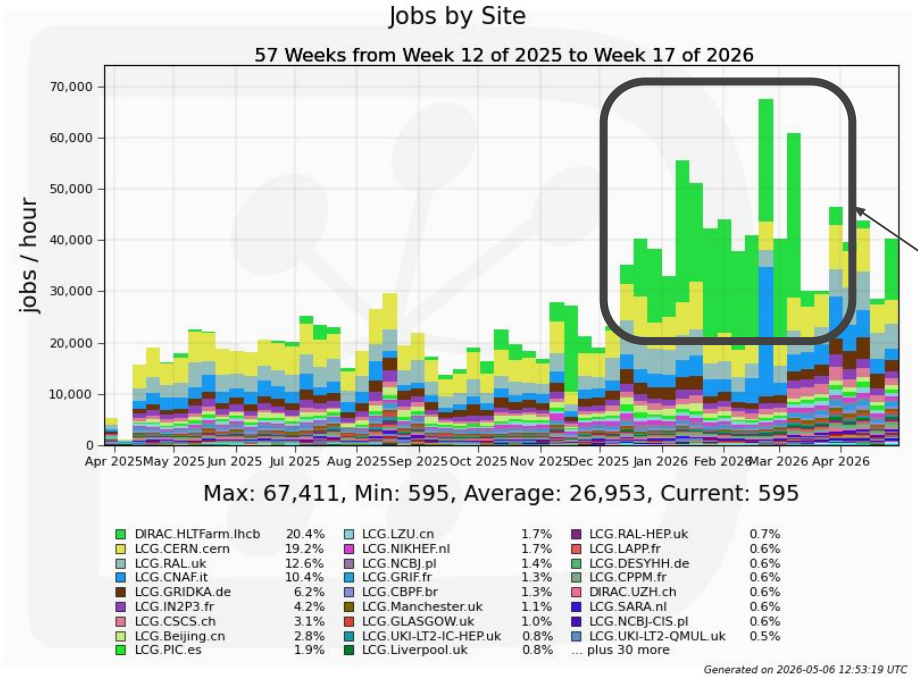
Total job run since April (Done : **50M**, Failed **10M** due to HLT2 priority)

Grid jobs by site since April (~ **10%** at HLT2Farm)

Conclusion and next steps

- Setup a HTCondor cluster that exposes opportunistically the HTL2 Farm resources
- Automated the inclusion of the nodes through custom ClassAds (cron like scripts)
- Avoid sending SIGUSR1 signal to the pilot or properly catch it (to be followed up with offline group)
- Plan to use farm during long shutdown
- Collecting use cases for HLT1 farm usage

Backup - Job Flow



winter shutdown

HLT2 contribution during recent MD