Managing a dynamic pool for CMS

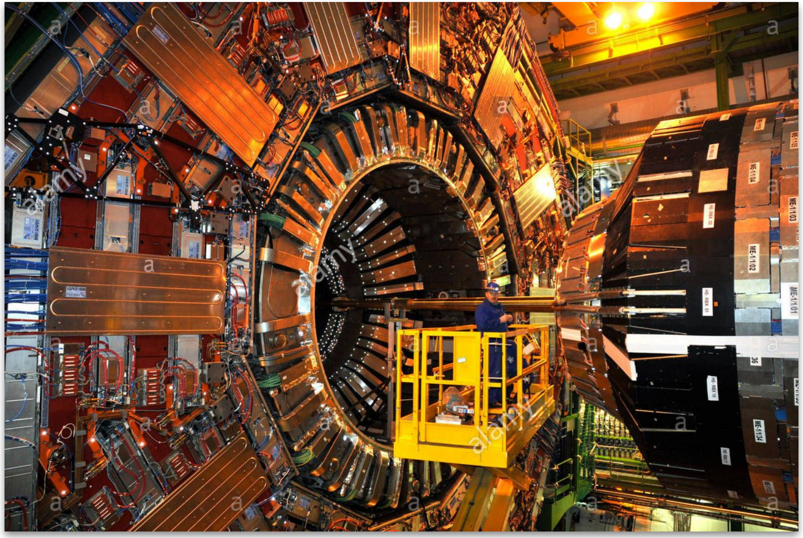
Marco Mascheroni, Antonio Perez-Calero Yzquierdo, Saqib Haleem, Edita Kizinevic
On Behalf of the CMS Submission Infrastructure team

HTCondor Workshop Spring 2021

Outline

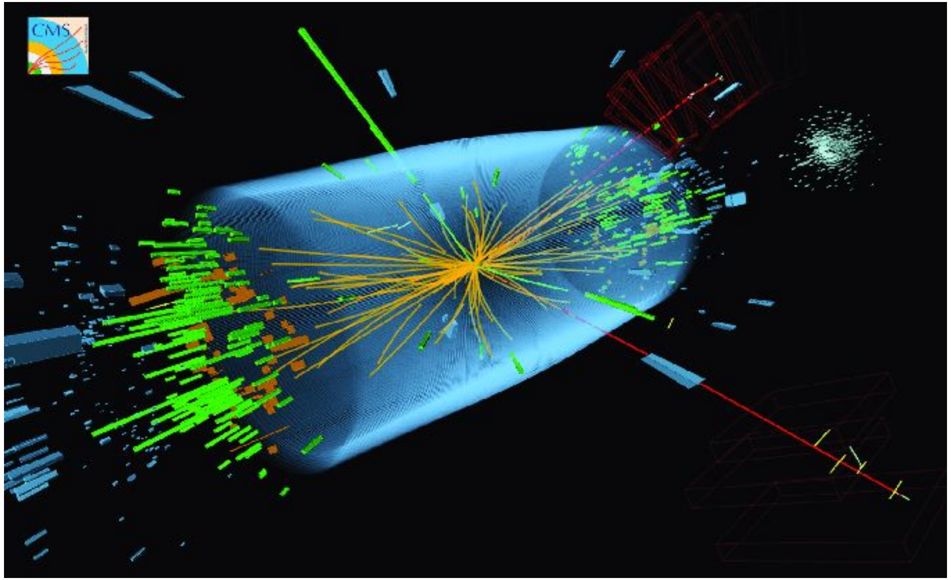
- The CMS experiment
- The CMS Submission Infrastructure
- Traditional Grid Resources and GlideinWMS
- Non Grid Resources and the vacuum model
- Conclusions

The CMS experiment at CERN



- **High Energy Physics general-purpose experiment** recording proton-proton collisions **at the LHC** at CERN

- Experimental data is stored, distributed, reconstructed, and analyzed, comparing to simulated data (Monte-Carlo)
    - **Hundreds of PBs per year**



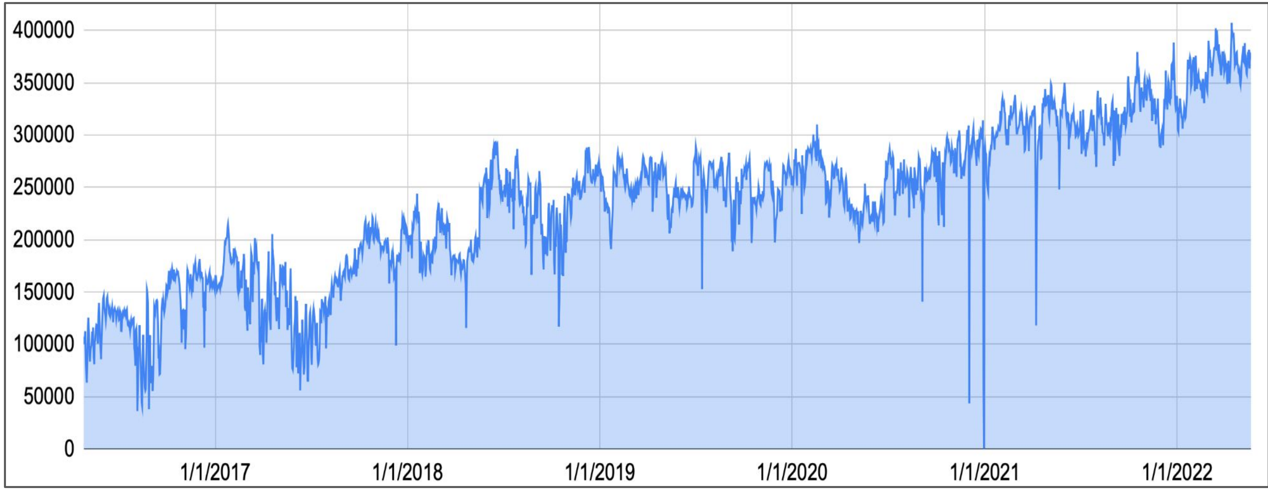
The computing landscape - the WLCG

- Data traditionally analyzed using **WLCG Grid resources**
  - Global collaboration of around 170 computing centers
  - Access based on dedicated resources (**pledges)**
- **Increasing** contributions from **HPC and** other types of **non-Grid** resources
  - Access often based on **allocations**
  - Also **opportunistic** and **volunteer**
  - **Non x86-64** and **GPUs** available

The CMS Submission Infrastructure Group

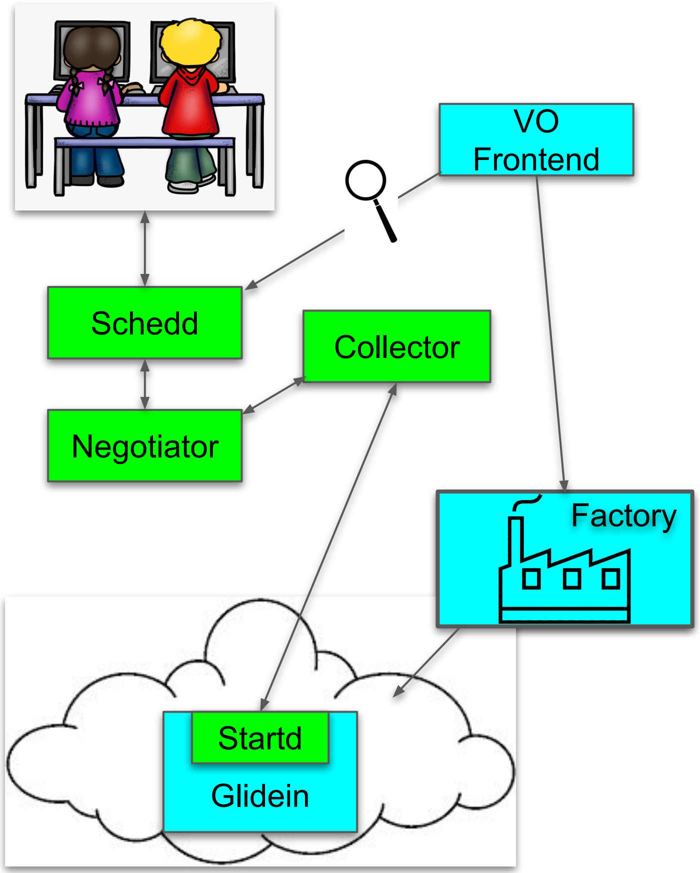
- Part of CMS Offline and Computing in **charge** of:
  - Organizing HTCondor and GlideinWMS operations in CMS, in particular of the **Global Pool**, an infrastructure where reconstruction, simulation, and analysis of physics data takes place
  - Communicate CMS **priorities to the development teams** of glideinWMS and HTCondor
- In practice:
  - We operate a set of federated HTCondor pools **distributed over 70 Grid sites, plus non-Grid resources**
  - We regularly hold **meetings with** HTCondor and glideinWMS **developers** where we discuss
    - Current operational limitations
    - Feature requests
    - Future scale requirements

CPU cores allocated to CMS over the past ~6 years (daily averages)



Traditional Grid Resources and glideinWMS

The Global Pool and GlideinWMS



- Ten years **proven approach**
- An HTCondor pool with startds running on the Grid is **dynamically** created
- GlideinWMS as a way to **adjust the size** of the pool and **configure** startds

How is the pool created is the focus of today

User Job Pressure

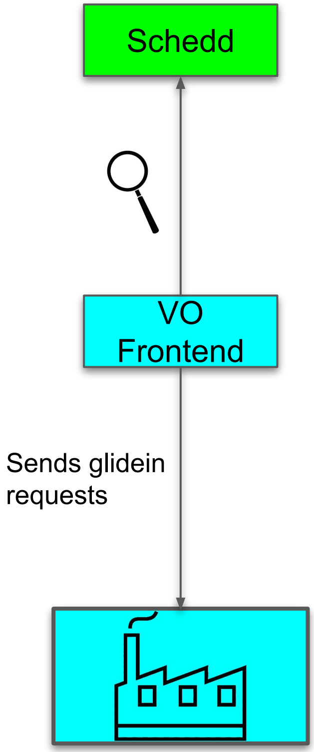
- Different workload management tools generates the **job pressure**
  - Tools used by **individual physicists** (CRAB, CMS Connect)
  - **Central production** organized centrally by small amount of people (WMAgent)
  - **Institutional** schedds
- Requirements for each **user job**
  - Classads for required memory, cpus, walltime, …
  - **DESIRED_Sites** classad: list of sites for data locality

Submit jobs and get results

Schedd

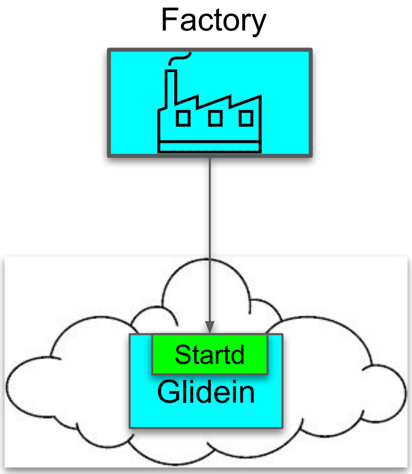
```
[mmascher@vocms0814 ~]$ condor_status -sched
Name                       Machine                    RunningJobs   IdleJobs   HeldJobs

SUBMIT04.MIT.EDU           SUBMIT04.MIT.EDU                   357         11          0
cmsgwms-submit6.fnal.gov   cmsgwms-submit6.fnal.gov         21027      53614          0
cmsgwms-submit7.fnal.gov   cmsgwms-submit7.fnal.gov         15138      55637          0
login-el7.uscms.org        login-el7.uscms.org               2561        195       6563
crab3@vocms0106.cern.ch    vocms0106.cern.ch                 3560      16982       6322
crab3@vocms0107.cern.ch    vocms0107.cern.ch                 2800      23241       6448
crab3@vocms0119.cern.ch    vocms0119.cern.ch                 4792      16152       6587
crab3@vocms0120.cern.ch    vocms0120.cern.ch                 5960      22475       6635
crab3@vocms0121.cern.ch    vocms0121.cern.ch                 3402      20237       6397
crab3@vocms0122.cern.ch    vocms0122.cern.ch                 3273      17519       6431
crab3@vocms0137.cern.ch    vocms0137.cern.ch                 4991      29076       6581
crab3@vocms0144.cern.ch    vocms0144.cern.ch                 2943      13379       6123
crab3@vocms0155.cern.ch    vocms0155.cern.ch                 4152      15688       9620
crab3@vocms0194.cern.ch    vocms0194.cern.ch                 3082      14119       6293
crab3@vocms0195.cern.ch    vocms0195.cern.ch                 3522      15289       6194
crab3@vocms0196.cern.ch    vocms0196.cern.ch                 3122      13834       6207
crab3@vocms0197.cern.ch    vocms0197.cern.ch                 4860      16342       6316
crab3@vocms0198.cern.ch    vocms0198.cern.ch                 3902      14763       5737
crab3@vocms0199.cern.ch    vocms0199.cern.ch                 4134      24767       6129
vocms0282.cern.ch          vocms0282.cern.ch                20444      54057          0
vocms0283.cern.ch          vocms0283.cern.ch                19876      54870          0
vocms0314.cern.ch          vocms0314.cern.ch                    7          2          0
vocms047.cern.ch           vocms047.cern.ch                     4          0          0
crab3@vocms059.cern.ch     vocms059.cern.ch                     1          0        222
```

The GlideinWMS frontend



- The CMS frontend looks at the pressure and sends **glidedin** requests to the factory
    - **First matchmaking** uses DESIRED_Sites and other classads to determine **required glideins** per site
- CMS specific configuration
    - **Singularity** images and bindmounts
    - Request adjustments based on job pressure (**idle removal**)
    - Condor version, slots layout, and other startd knobs
- Validation scripts
- Different groups for different types of jobs
    - Local users: some users have exclusive access to extra pledge resources
    - GPUs: requires different startd configuration and additional validation scripts
    - Overflow: ignore DESIRED_Sites and use xrootd under special conditions
    - …

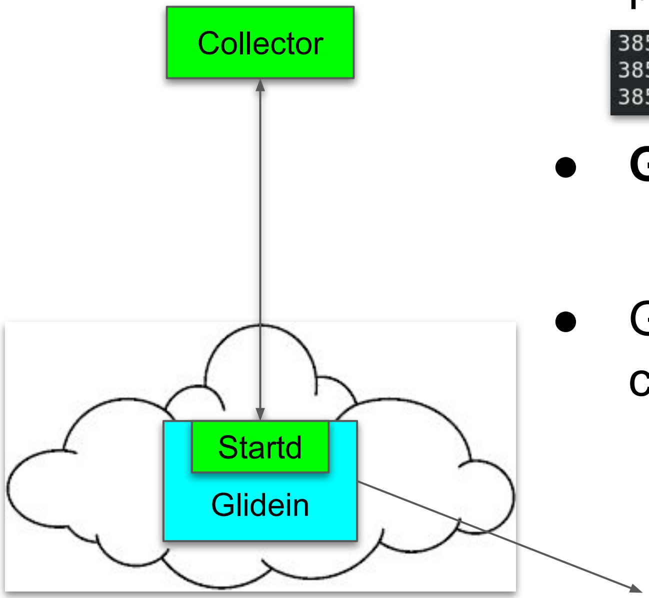
The GlideinWMS factory



- Based on frontend request factory is responsible of sending pilot jobs to Site’s Compute Elements
  - Factory configuration consists in static description of Grid resources
    - Factory shared with OSG and other experiments
  - Stores HTCondor tarballs to be downloaded on WN

```
<entry name="CMSHTPC_T1_US_FNAL_condce4_whole" auth_method="grid_proxy" gatekeeper="cmsosgce4.fnal.gov cmsosgce4.fnal.gov:9619" gridtype="condor" work_dir="Condor">
   <config>
      <max_jobs>
         <default_per_frontend glideins="50000" held="100" idle="5000"/>
         <per_entry glideins="50000" held="100" idle="5000"/>
      </max_jobs>
      <submit cluster_size="10" max_per_cycle="100" sleep="2">
         <submit_attrs>
            <submit_attr name="+maxWallTime" value="2880"/>
            <submit_attr name="+WantWholeNode" value="True"/>
         </submit_attrs>
      </submit>
   </config>
   <allow_frontends>
   </allow_frontends>
   <attrs>
      <attr name="GLIDEIN_CMSSite" const="True" glidein_publish="True" job_publish="True" parameter="True" publish="True" type="string" value="T1_US_FNAL"/>
      <attr name="GLIDEIN_CPUS" const="True" glidein_publish="False" job_publish="True" parameter="True" publish="True" type="string" value="auto"/>
      <attr name="GLIDEIN_ESTIMATED_CPUS" const="True" glidein_publish="False" job_publish="True" parameter="True" publish="True" type="string" value="64"/>
      <attr name="GLIDEIN_MaxMemMBs_Estimate" const="True" glidein_publish="True" job_publish="False" parameter="True" publish="True" type="string" value="True"/>
      <attr name="GLIDEIN_Max_Walltime" const="True" glidein_publish="False" job_publish="False" parameter="True" publish="True" type="int" value="171000"/>
      <attr name="GLIDEIN_REQUIRED_OS" const="True" glidein_publish="True" job_publish="False" parameter="True" publish="True" type="string" value="any"/>
      <attr name="GLIDEIN_Supported_VOs" const="True" glidein_publish="False" job_publish="False" parameter="True" publish="True" type="string" value="CMS"/>
   </attrs>
</entry>
```

Glidein jobs on the Grid



- **Factory** uses HTCondor-G to **send glidein jobs** (aka pilot jobs) to the site queues

```
3858108.1    fecmsglobal      5/23 05:00   0+00:35:21 R  0   1200 glidein_startup.sh
3858329.0    fecmsglobal      5/23 05:29   0+00:09:10 R  0    0.1 glidein_startup.sh
3858369.0    fecmsglobal      5/23 05:35   0+00:00:00 I  0    0.1 glidein_startup.sh
```

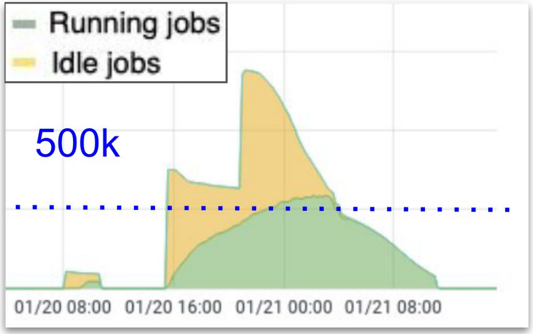
- **Glidein** eventually **runs**
  - Resource validation (scripts hosted on frontend)
  - Downloads HTCondor tarball from the factory
- Glidein launches a **properly configured startd** that connects to the Global pool
  - Platform selection
  - Singularity setup
  - Handling of drain (HTCondor CRON)
  - Custom START expression
  - CMSSW discovery
  - Frontier
  - Whole node setup

Central manager

- HTCondor startds available in the central manager for **second matchmaking**
  - Multicore and partitionable slots
  - User jobs script and sandbox transferred to worker nodes for execution

```
[mmascher@vocms0814 ~]$ condor_status -const 'slottype=="Partitionable"' -limit 10
Name                                                              OpSys      Arch   State     Activity LoadAv Mem   ActvtyTime

slot1@glidein_29049_520134706@CRUSH-OSG-C7-10-5-144-159           LINUX      X86_64 Unclaimed Idle      0.000  9248  1+05:11:02
slot1@glidein_29128_324146556@T2BAT0335.CMSAF.MIT.EDU             LINUX      X86_64 Unclaimed Idle      0.000  4368  0+08:44:05
slot1@glidein_72212_61286699@batch1202.desy.de                    LINUX      X86_64 Unclaimed Idle      0.000  3104  0+04:09:56
slot1@glidein_130356_671571298@c01-116-128.gridka.de              LINUX      X86_64 Unclaimed Idle      0.000  4608  1+02:12:56
slot1@glidein_141945_160403022@c01-118-127.gridka.de              LINUX      X86_64 Unclaimed Idle      0.000  4608  0+11:22:58
slot1@glidein_7_375411690@cmsgli-7139523-0-cmswn2279.fnal.gov     LINUX      X86_64 Unclaimed Idle      0.000 38460  1+10:51:15
slot1@glidein_1_446955000@cmspilot-12755300.0-red-c5019.unl.edu   LINUX      X86_64 Unclaimed Idle      0.000  1181  0+05:26:27
slot1@glidein_3849819_97457864@mh-epyc7662-4.t2.ucsd.edu          LINUX      X86_64 Unclaimed Idle      0.000  1808  0+02:03:46
slot1@glidein_43_199044715@ttcms032-4057885.0-lcg2237.gridpp.rl.ac.uk LINUX  X86_64 Unclaimed Idle      0.000  8704  0+16:17:11
slot1@glidein_15494_57338496@wl-07-22.lnl.infn.it                 LINUX      X86_64 Unclaimed Idle      0.000  3856  0+13:28:15
```

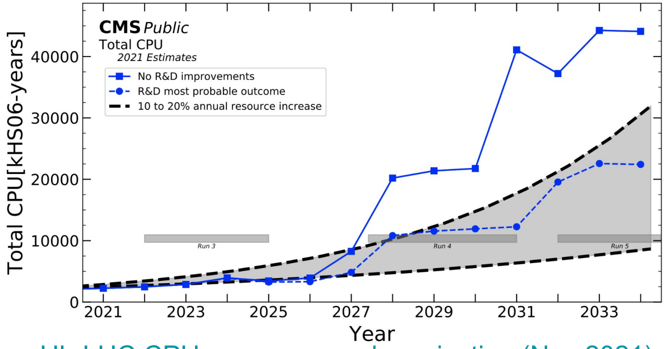
- 2021 tests showed **enough capacity for run3** (with some contingency)



Non Grid Resources and the vacuum model

Diversification of resources

- Access further resources in order to reach growing scales required by CMS (e.g. by HL-LHC era)
- HPC primary source of new resources
    - **Funding** for (pre-)Exascale infrastructures



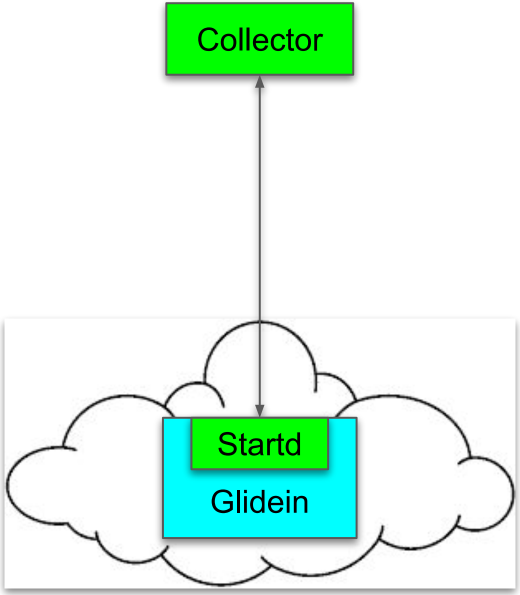
HL-LHC CPU resource needs projection (Nov 2021)

- CMS access HPC via a number of ways
    - **HEPCloud** as a portal to an ecosystem of diverse computing resources (see talk in the next session)
        - E.g.: NERSC Cori, TACC Frontera/Stampede2, PSC Bridges-2, SDSC Expanse, Purdue Anvil, ...
        - Routes to local or remote resources based on workflow requirements, cost, and efficiency
    - **Glideins in a vacuum** when it is hard to put a Compute Element in front of the resources
        - E.g.: CINECA Marconi100, BSC–CNS, Lancium, JSC-Jureca
    - …but we are also accessing HPCs via Compute Element + GlideinWMS glideins (i.e. **HPC a la Grid**)

- CMS@Home and High Level Trigger farm at CERN also exploited

Re-using glidein framework for resource validation and condor configuration **is key**

Glideins in a vacuum and manually configured startds



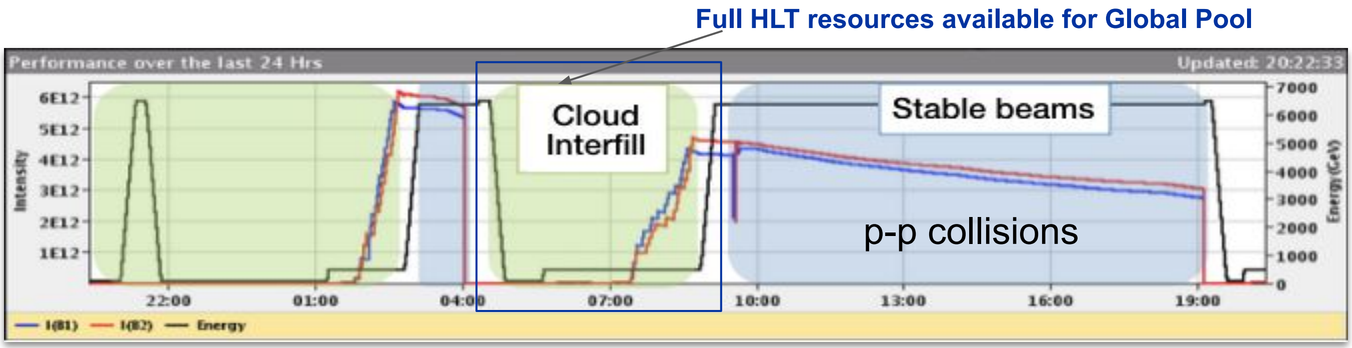
- Glideins ***spontaneously* started** on worker nodes
    - **Not requested** by the GlideinWMS Fontend
        - E.g. manually, by resource admins
    - **Tool developed to generate** the glidein command line **arguments**
        - It picks up latest changes (e.g.: need new filenames because of squids)

- Additionally, manually configured and launched startds are also supported
    - E.g.: the **split starter case at BSC**, which is not currently integrated with GlideinWMS
    - Missing resource/environment validation scripts included by default in GlideinWMS glideins
    - Need to maintain condor configuration as similar as possible to the one generated by GlideinWMS
        - monitor tools rely on standard set of classads

The HLT farms in 2022 and Run 3

- The **CMS Online Cloud** (deployed on the High Level Trigger farm resources) was commissioned during Run2 to dedicate HLT **resources for offline data processing** when not needed to support data taking (ref)
- Launch Openstack VMs running the startds when CPU is available (**Interfill + Fill modes**)
  - VMs suspended when HLT needs resources back (MaxHibernationTime of 24 hours)
- New farm will be available and will be used in Run 3
  - With lots of **GPUs**
  - Usage will be similar to the old one (interfill + fill modes)
  - Frees up the old farm: now a resource **fully dedicated to offline processing (~20k CPU cores)**

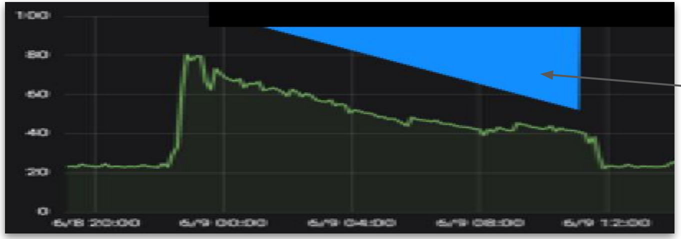
INTERFILL



LHC operation cycles

DURING LHC FILLS

HLT resources % required for DAQ



Opportunistic resources integration

Frequently managed as “extension” (or sub-Sites) of WLCG T1 or T2 centers with pledged resources

- **HPCs attached to sites**, e.g.: CINECA (T1_IT_CNAF), BSC (T1_ES_PIC), Jülich and HOREKA (T1_DE_KIT)
- **Dynamic Cloud extensions of grid sites**: CERN_Azure, PIC_AWS
- **Opportunistic** use of local clusters: CERN_BEER, KIT_T3, Purdue…

Most of these resources include **custom start** expressions, with **additional filters** to tune matchmaking of jobs to **non-standard resources constraints**

- E.g. **avoid analysis jobs** on HPCs, or just run **GEN-SIM jobs** (no remote storage access required) on opportunistic resources

Conclusions and future prospects

Conclusions and Future Prospects

- CMS Submission Infrastructure keeps the expansion toward a increasingly more **diverse set of resources**
  - Non x86 architectures, GPUs, HPCs, Cloud, etc

- **…while ensuring flexibility** in ways to connect new resources to the Global Pool

Conclusions and Future Prospects

Large Hadron Collider restarts

Beams of protons are again circulating around the collider's 27-kilometre ring, marking the end of a multiple-year hiatus for upgrade work

22 APRIL, 2022

Additional motivation as the LHC is restarting operations (Run3, 2022-25)

…at **unprecedented collision energy** (13.6 TeV) providing new territory for Physics exploration

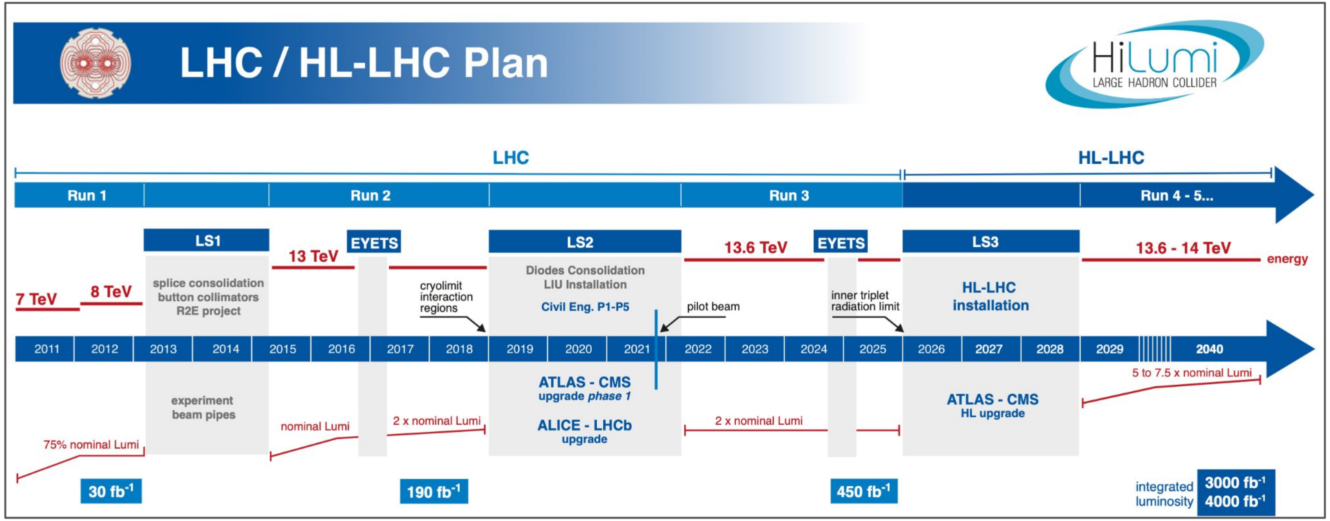
Time to accumulate loads of new data with CMS!

[CERN news]

Conclusions and Future Prospects

…and the LHC program extends well into the future so we need to:

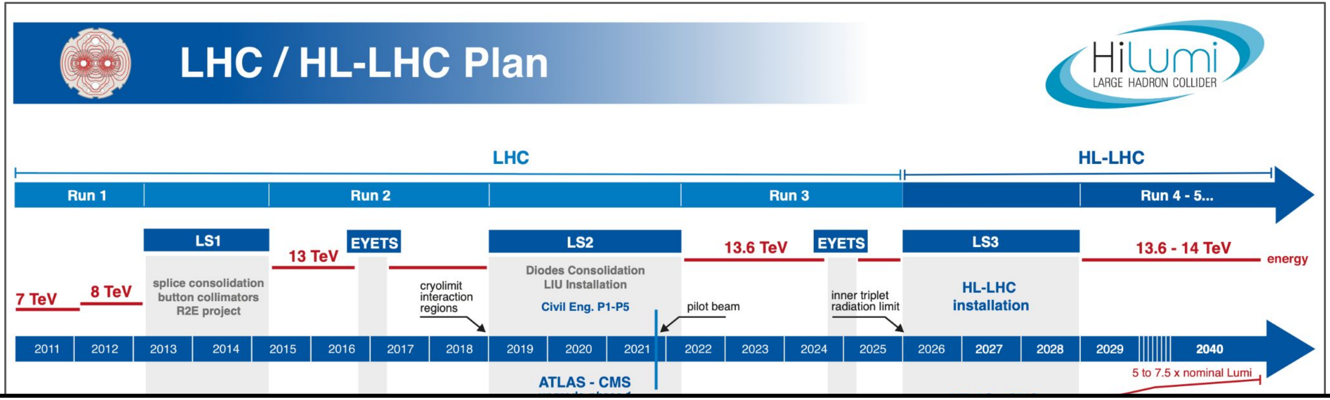
- continue pushing for **higher scales**, as required by CMS needs…
- …while maintaining **stability** and **efficiency**
- remain relevant by **adapting** to tools/technology changes
    - Recently completed **token migration** (next talk by Saqib)



Conclusions and Future Prospects

…and the LHC program extends well into the future so we need to:

- continue pushing for **higher scales**, as required by CMS needs…
- …while maintaining **stability** and **efficiency**
- remain relevant by **adapting** to tools/technology changes
    - Recently completed **token migration** (next talk by Saqib)



Thanks to the HTCondor team for the great support over many years!

Abstract

Our presentation will cover the current status of the CMS Submission Infrastructure, which is undergoing a diversification of the type of resources and the methods to access them. In particular, we have been incorporating multiple HPC resources that are treated as dynamic extensions of existing grid sites, while the so-called "vacuum model" for resource allocation is also becoming popular for acquisition of cloud resources and also at other supercomputers.

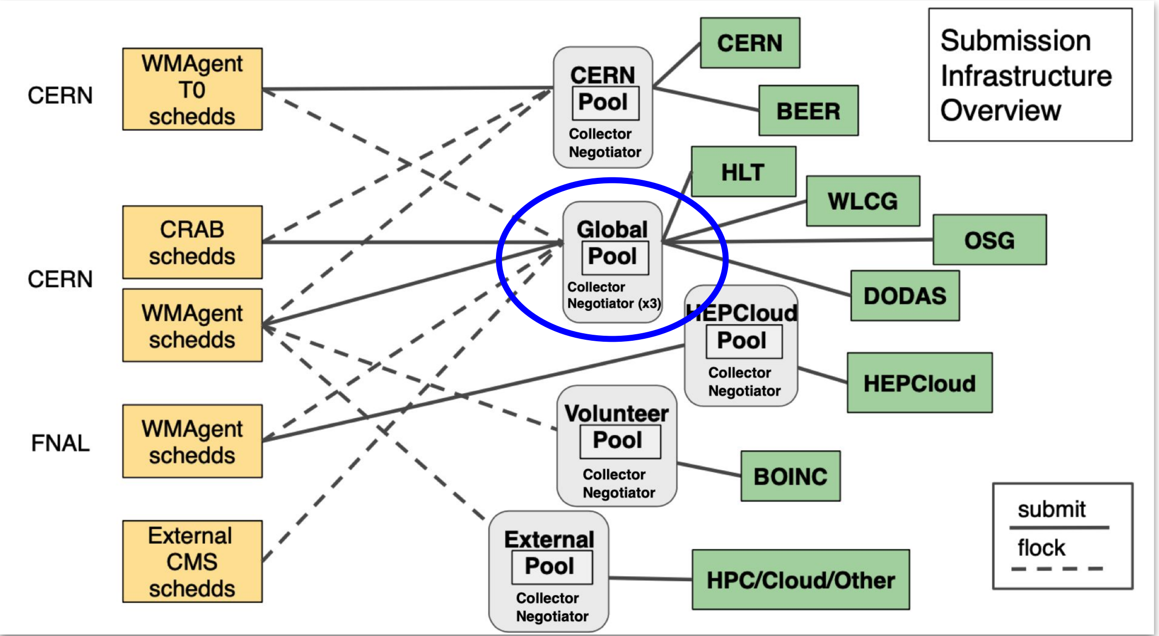
Backup Slides

A Federation of pools

- **The CMS SI model evolved to use federated pools,** with extensive use of **flocking**

- Two sets of workflow managers: CRAB + WMAgent

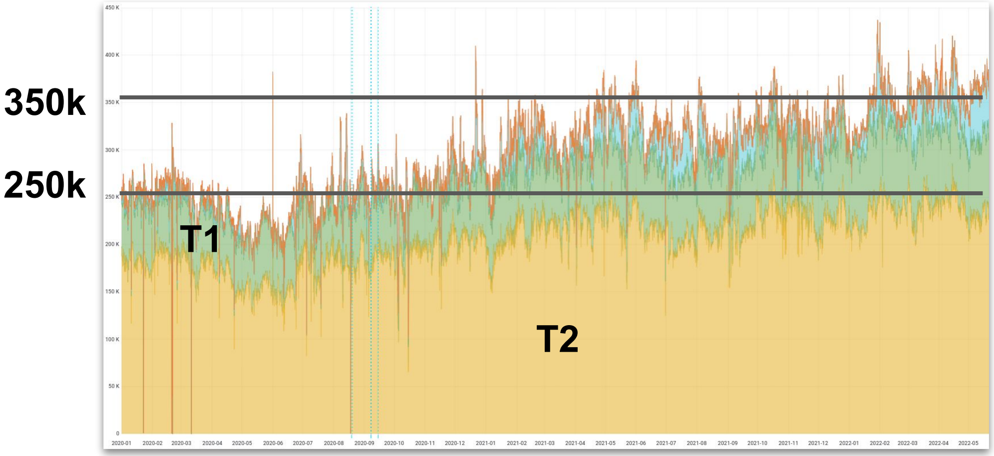
- **The Global Pool is the biggest** and most important one:
  - **~300k CPU cores**
  - **100k to 150k running jobs**
  - 50+ schedds
  - 3 negotiators
- Redundant infrastructure for HA



- Resources **mainly** acquired with GlideinWMS pilots
- Vacuum-like instantiated: slots (DODAS), BOINC(CMS@Home), opportunistic (HLT)...

The CMS Submission Infrastructure Group

- Part of CMS Offline and Computing in **charge** of:
  - Organizing HTCondor 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 HTCondor
- In practice:
  - We operate an HTCondor pool peaking at **300k/350k cores distributed over 70 Grid sites, plus non-Grid resources**
  - We regularly hold **meetings with** HTCondor and glideinWMS **developers** where we discuss
    - Current operational limitations
    - Feature requests
    - Future scale requirements



Cores running in the Global Pool in the past 2 years