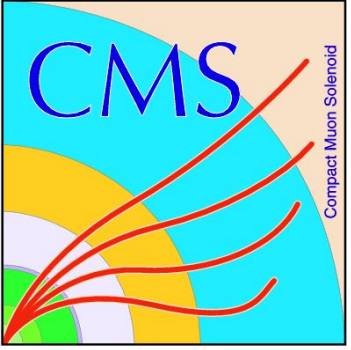


Improving the Scheduling Efficiency and Scalability of a Global Multi-core HTCondor Pool in CMS

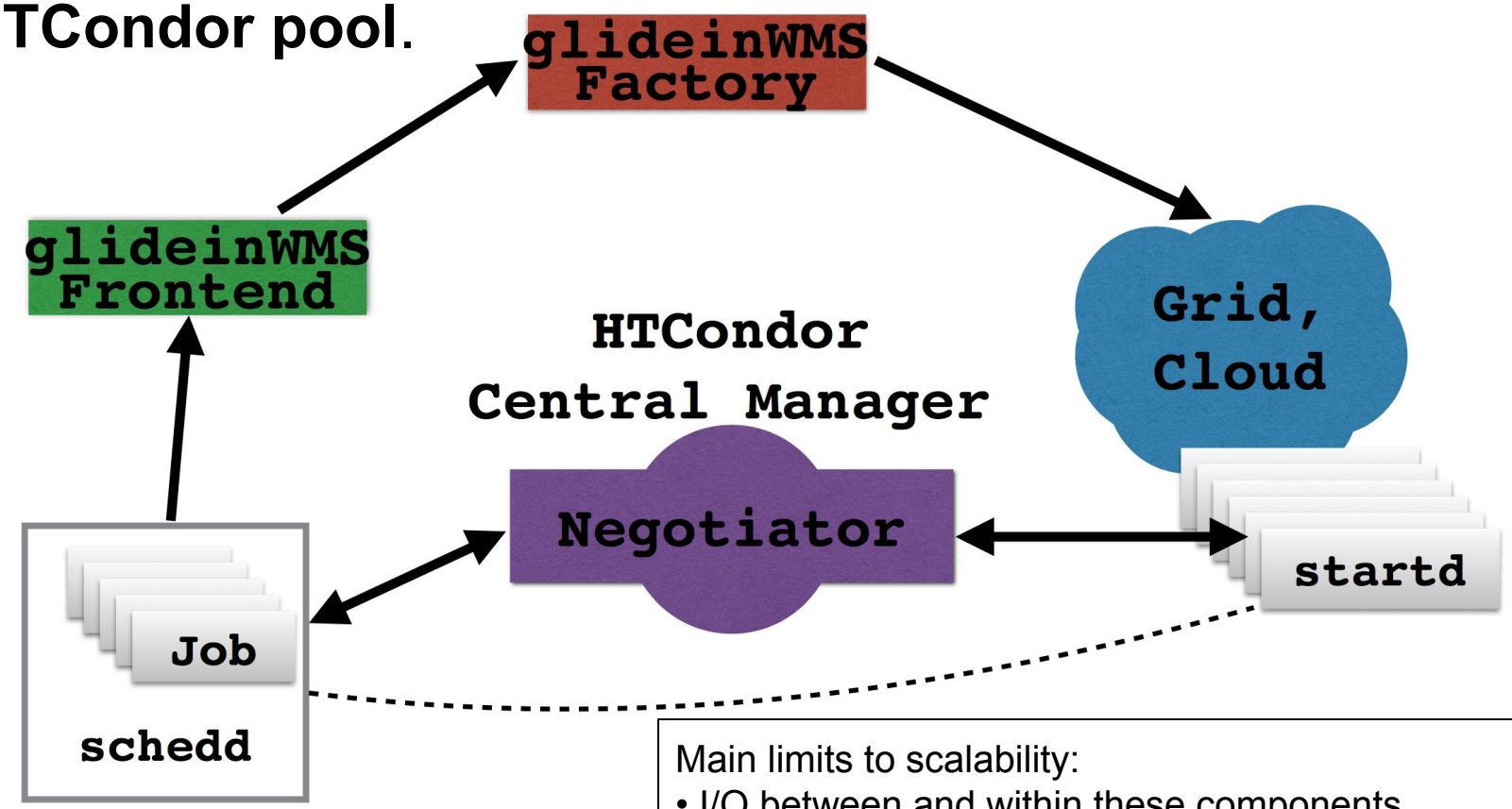
James Letts
on behalf of the CMS Submission Infrastructure Group
HTCondor Week
May 22, 2018

SI Group Charge

- The charge of the Submission Infrastructure Group with the CMS experiment at CERN is to:
  - **Organize** glideinWMS and HTCondor pool **operations** in CMS, in particular of the Global and Tier-0 HTCondor Pools
  - **Communicate** CMS **priorities** to the development teams of glideinWMS and HTCondor
- SI activities broadly fall into several categories:
  - Overcoming current operational limitations or problems
  - Preparing for future scales or feature requirements (i.e. next year's problems)
  - Integration of new, diverse resource types and submission methods
- We regularly contribute to the HTCondor Workshops in the U.S. and Europe as well as to international conferences such as CHEP.

Global Pool

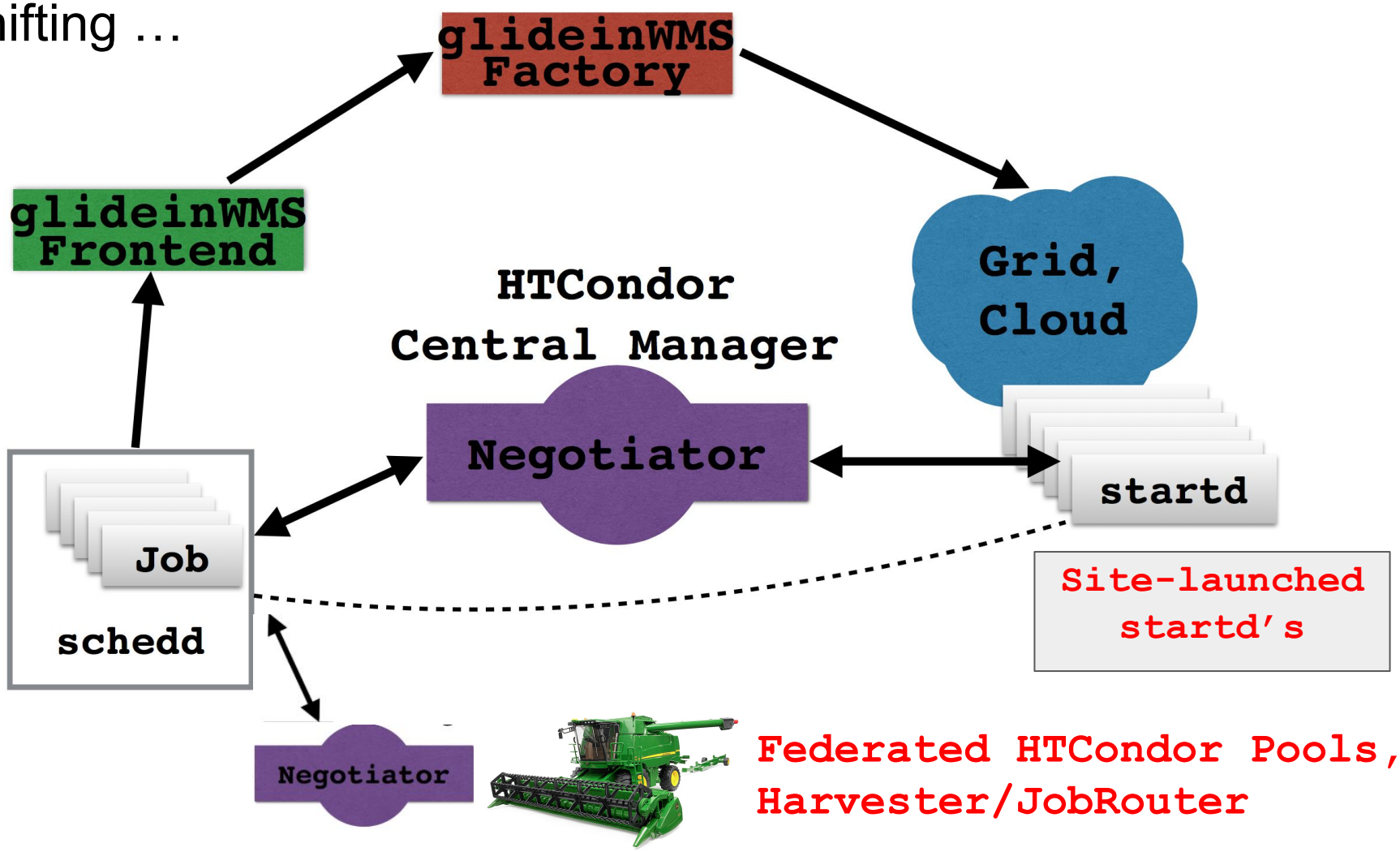
The Global Pool is both a **glideinWMS instance** and a **HTCondor pool**.



Main limits to scalability:
- I/O between and within these components
- Combinatorics of Negotiator (RRL's x pilot startd's)
- Speed and RAM usage of individual components
- Ability to scale horizontally

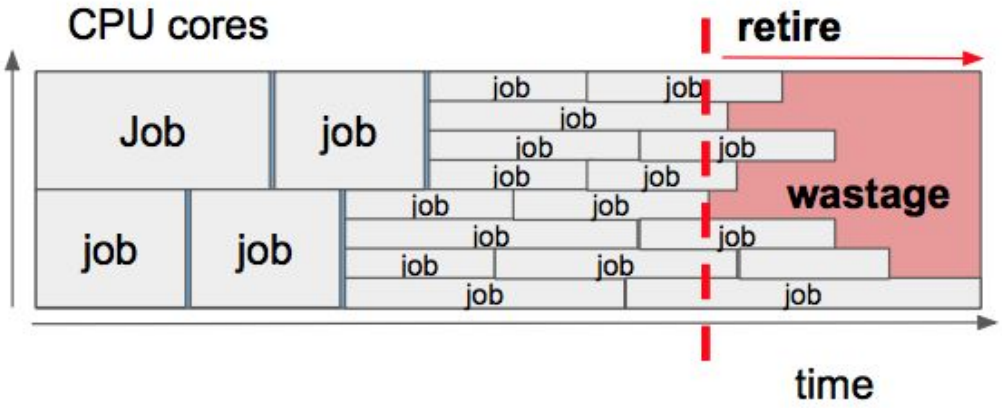
Global Pool

Meanwhile, the resource and submission landscape is shifting …



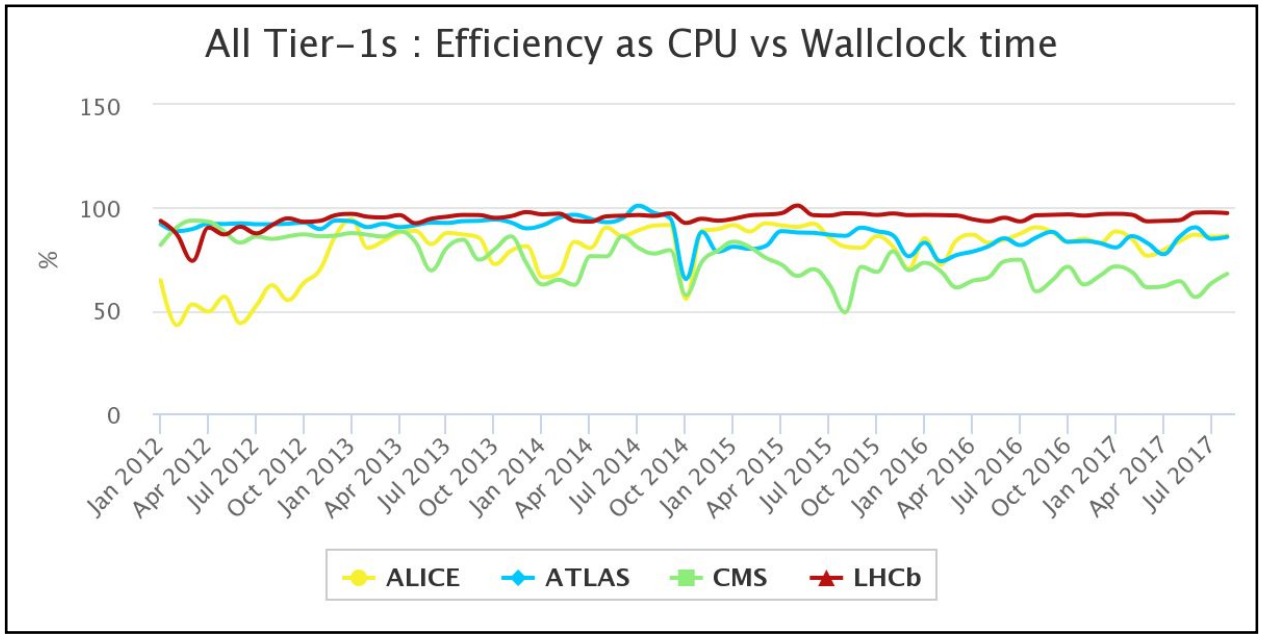
Multi-core Pool

- The Global Pool is also a multi-core pool with dynamic provisioning of slots (CPU, Memory, Disk, eventually I/O).
- Relevant parameters are the glidein lifetime and glidein retire time (constrained by accuracy of wall clock time estimation of jobs).
- Partitionable slots can become fragmented into lower core count dynamic slots over time - pilot renewal counters the fragmentation effect. Also explicit defragmentation (daemon).



CPU Efficiency

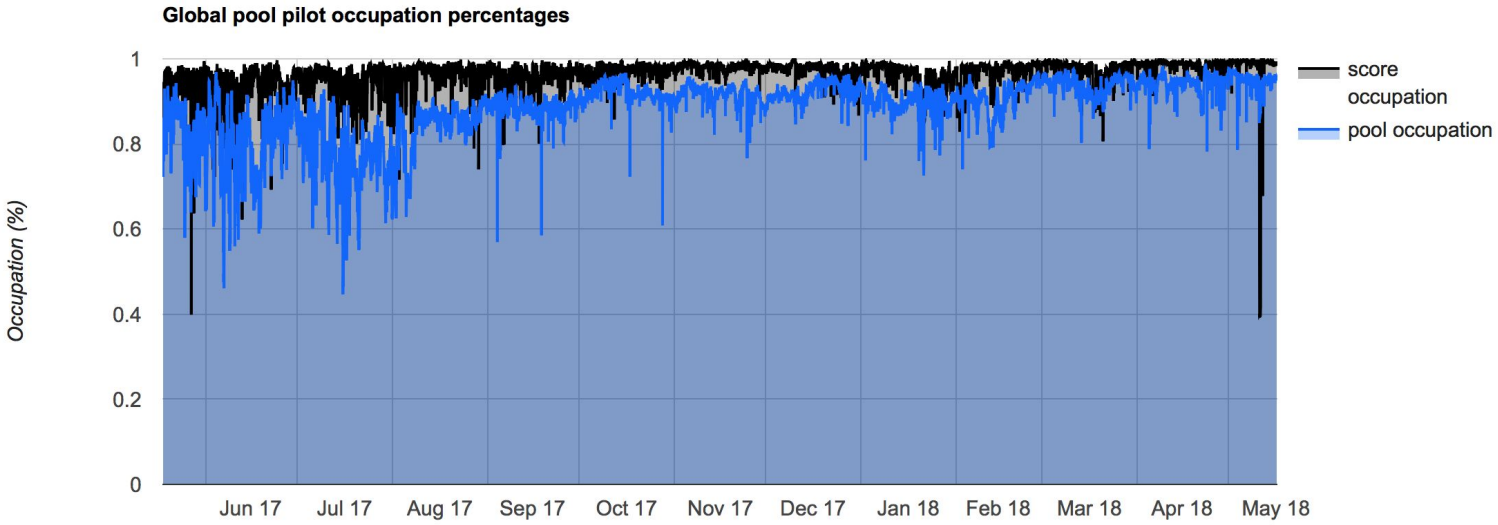
- CMS has been studying CPU efficiency in a dedicated task force for the past year. Motivated by poor comparisons with other LHC experiments.
- We were the first large experiment to use multi-core extensively.
- CPU efficiency is factorable between the pilot scheduling efficiency and the intrinsic CPU efficiency of the application.



Source: [EGI Accounting Portal]

Scheduling Efficiency

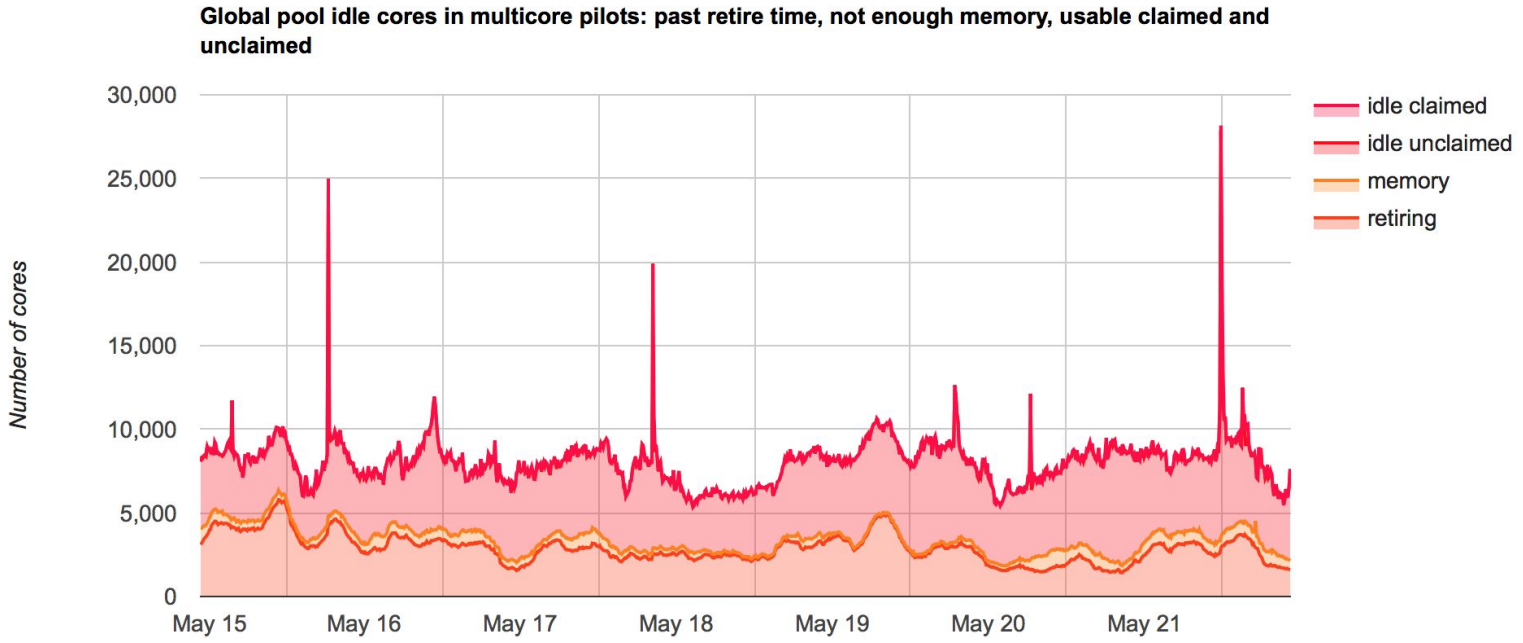
- Scheduling efficiency is % of occupied cores relative to the total number.
- This time last year, multi-core pilot occupancy was a problem, often ~80%
- Single-core glidein occupancy has always been around 97% or better, with losses of the order of the negotiation cycle length relative to the pilot length.
- We fixed multi-core scheduling efficiency in 2017. The next few slides show how.
- Current efficiency close to that of single-core, ~96% or better, after legitimate use cases are taken into account.



Sources of Inefficiency

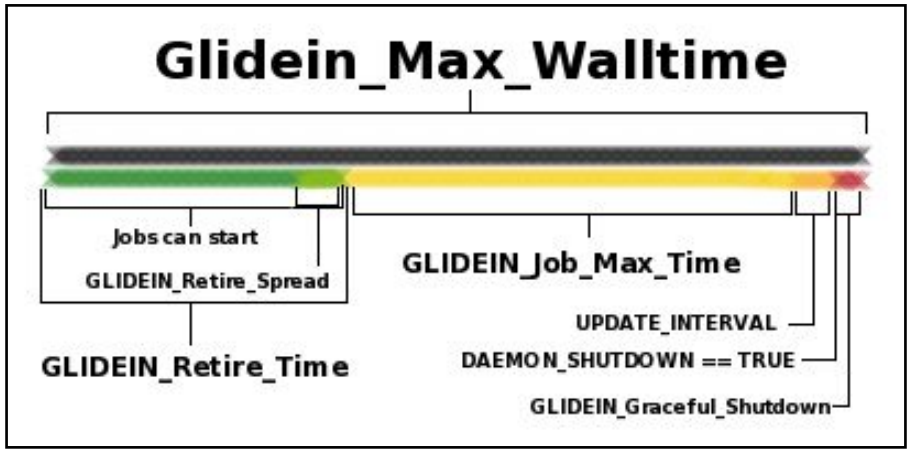
In multi-core glideins, there are several sources of inefficiency:

- Retiring glideins (exit when last job finishes - not a problem with single core)
- Memory starvation (p-slots use up all memory before CPUs) - legitimate use case when a workflow needs more memory/core
- CPU cores for which there is no work to match



Retiring Glideins

- Glidein retire time - stop accepting new jobs. Was set to 10h years ago.
- Driven by the accuracy of the job estimated wall clock time, i.e. we need to give jobs time to actually finish if they habitually overshoot their run time by N hours, else we increase badput.
- A 2017 study found that over 99% of jobs complete within 3h of the time that they ask for, so we lowered the retire time to 4h. This, however, was after several months of effort both from the analysis and production sides to increase accuracy of MaxWallClockMins.
- Reduced the inefficiency due to retirement from ~5% to less than 2%.



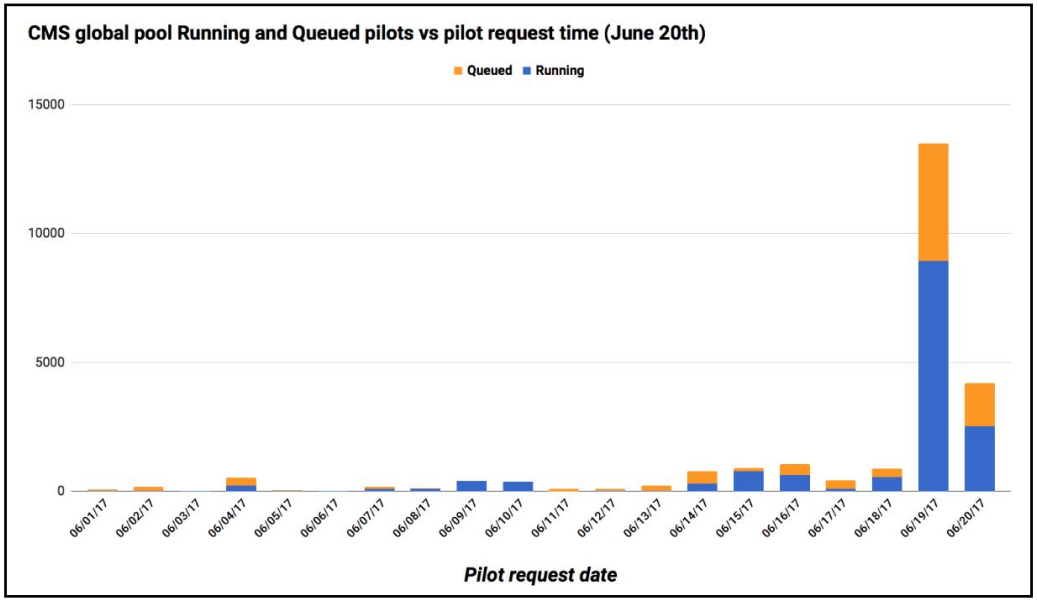
No Work to Match

Several reasons why a slot may have no work to match it:

- Slots are long-lived and held idle waiting for work (e.g. HLT farm, CAF) - **legitimate use case**.
- Over-provisioning:
  - Too many glideins are requested for a particular workflow and the work is spread thinly among them.
  - Slots could be restricted to particular kind of workflow or user.
  - Glidein was requested long enough ago that the work completed elsewhere.
- Pilots become fragmented over time and cannot match higher core count workflows.
- Frequent expansion and contraction of the pool - draining glideins before retirement because of a lack of work.

Removing Stale Pilots

- We discovered that pilots existed in the batch queues at sites long after they were requested - sometimes up to two weeks
- In an environment where job pressure is changing rapidly, we needed tighter coupling of resource provisioning to job pressure.
- First order solution: remove idle (queued) glideins after 1h
- Improved overall scheduling efficiency by ~10%
- Downside: pilot churn at sites.



HTCondor Improvements

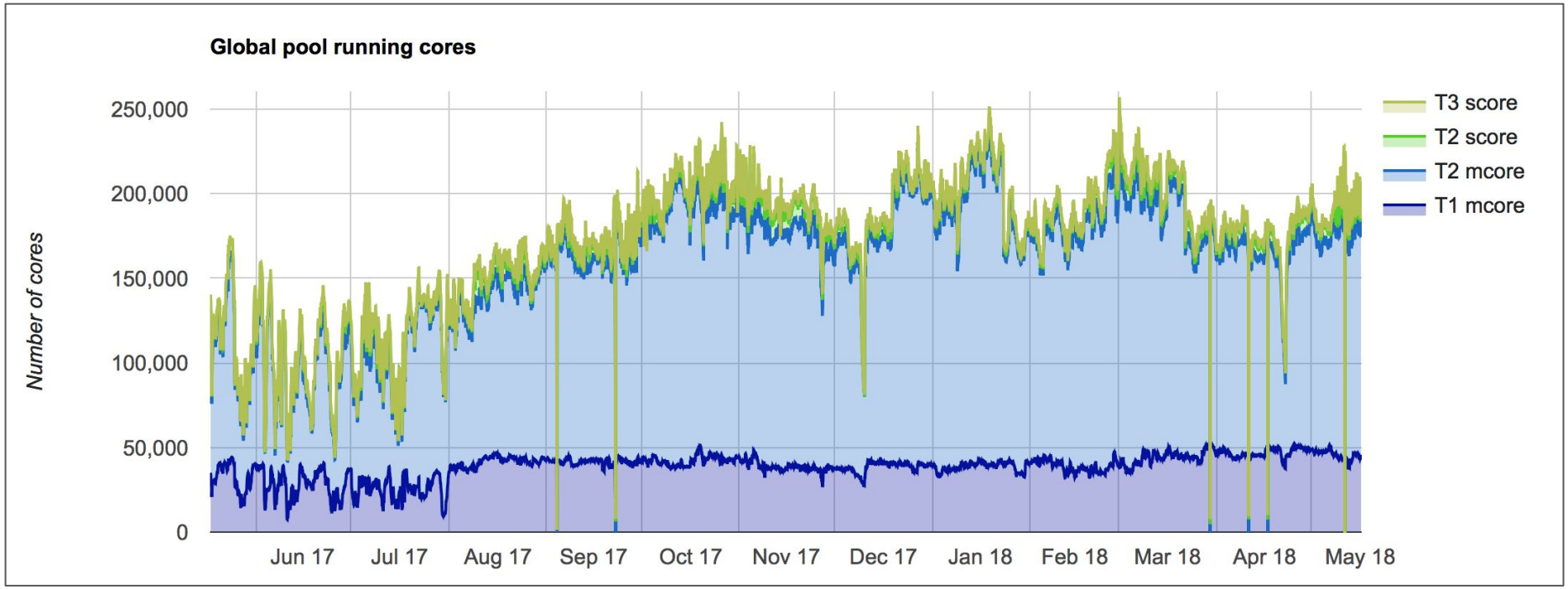
We also worked closely with the HTCondor developers to improve mostly the functioning of the Central Manager.

- Depth-wise filling of pilots / Slot weight expression
- Moving the CCB's onto separate hardware
- Negotiator on most powerful VM's available at CERN - scaling vertically
- Queuing & prioritization of central manager queries - less blocking

We have not focused much on schedd improvements since we can more easily scale horizontally (for now).

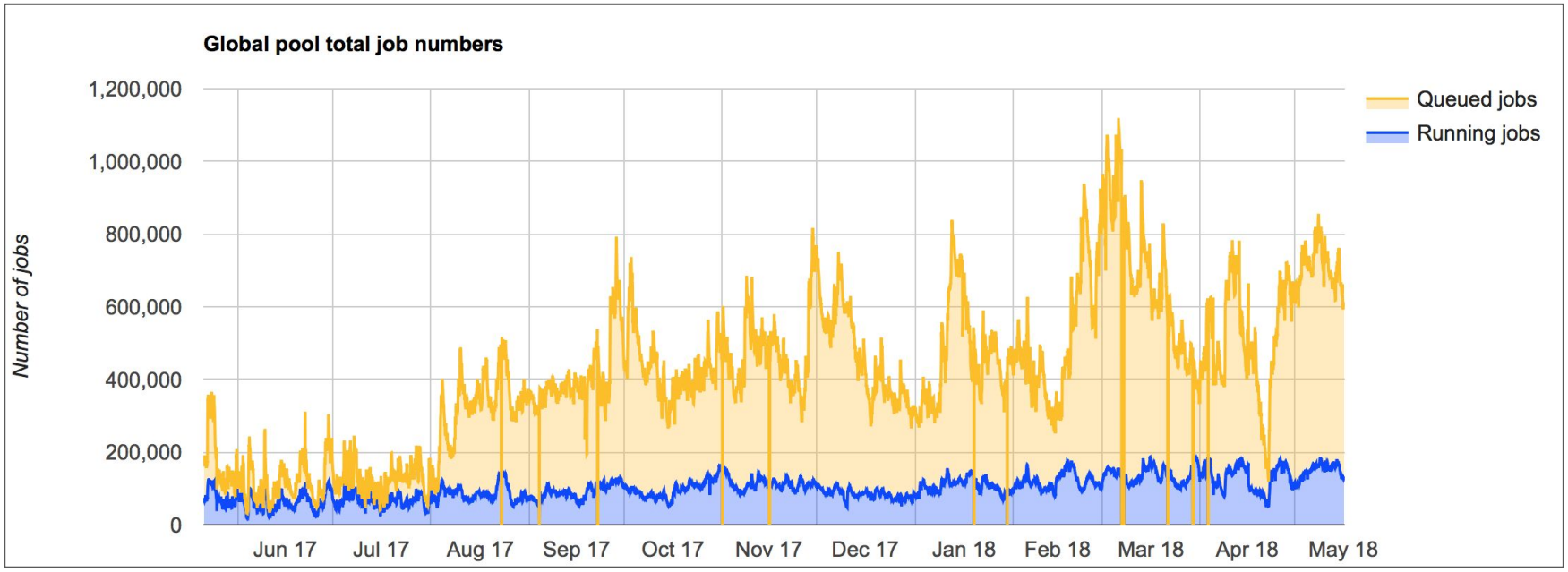
Recent Scaling

- Since last year's HTCondor Week, the CMS Global Pool has grown from ~100K CPU cores to over 200K, with peaks of 250K.
- Largely driven by resource deployments, which tend to happen later in the calendar year, and the availability of opportunistic cores.



Demand for Resources

- Demand for resources more constantly increasing, and increasing in core count.
- There are seasonal peaks and random noise.
- Challenge of SI is to provision resources **stably** and **efficiently** in whatever demand environment is facing us.



HL-LHC Challenge

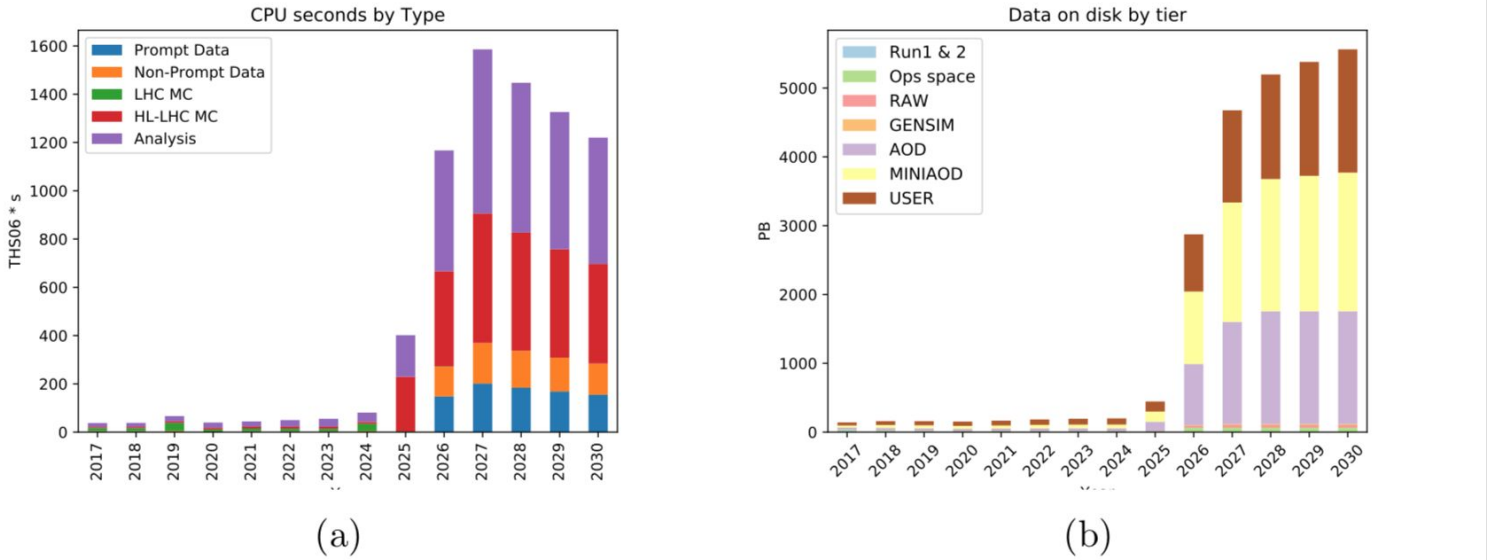


Figure 3: CMS estimated CPU (3a) and disk space (3b) resources required into the HL-LHC era, using the current computing model with parameters projected out for the next 12 years.

- Plot of CMS HL-HLC processing and data scales, from HSF.
- Factor of ~20 jump in scale (jobs and cores) around 2025.
- Expect the resource landscape to accelerate move to HPC & Cloud.

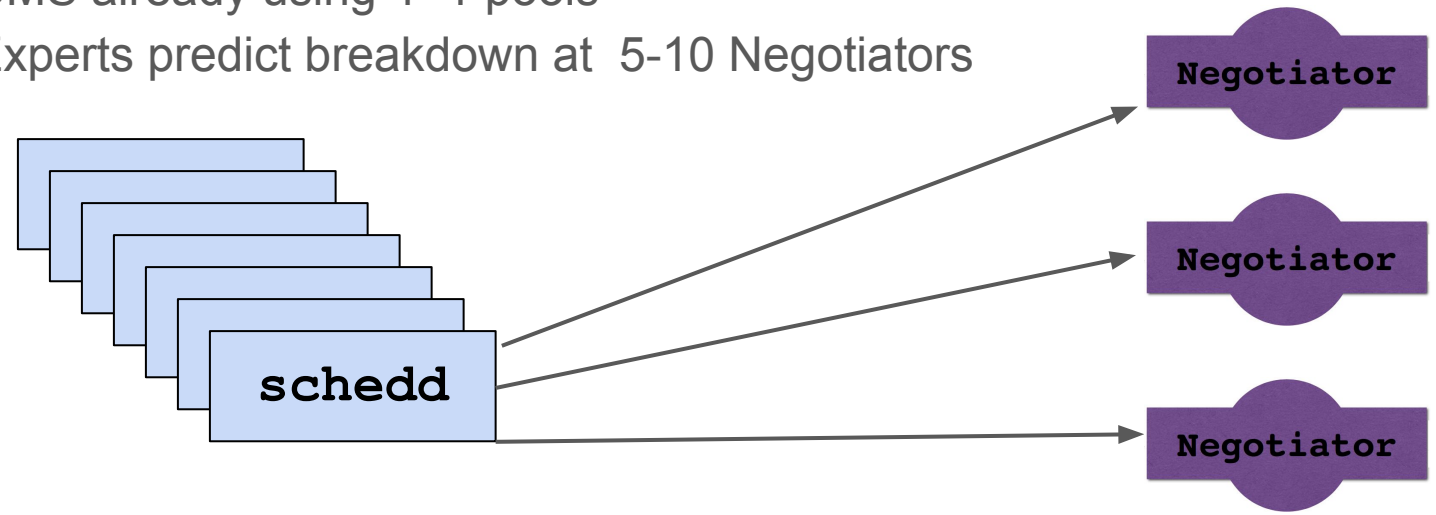
Evolution or Revolution?

- We expect a factor 20 increase in the numbers of jobs and cores.
- Only some evolution in parallelization expected. Current mix is 1/4/8-threaded jobs … perhaps a factor of 2-4 more only.
- Still expect a significant fraction of single-core workflows.

- The pilot model works best when all of your resources are more or less the same: Grid resources based on similar architectures and OS, sitting behind a small number of different Computing Elements (i.e. HTCondor-CE, ARC, CREAM).
- The resource landscape is already changing ...

Talking to Multiple Pools

- However, with larger scale use of HPC, local admins want more control over what (and when) can run on their clusters.
- Breaking of the Global Pool model already underway.
  - CERN Pool serves Tier-0: CMS data taking - specialized policies
  - HEPCloud in the U.S. - more specialized policies depending on resource capabilities & costs
  - Other (many) national Clouds coming soon in Europe.
- Question (to be answered by our 2018 round of scale testing): How many pools (Negotiators) can a schedd talk to before things break down?
  - CMS already using 4+1 pools
  - Experts predict breakdown at 5-10 Negotiators



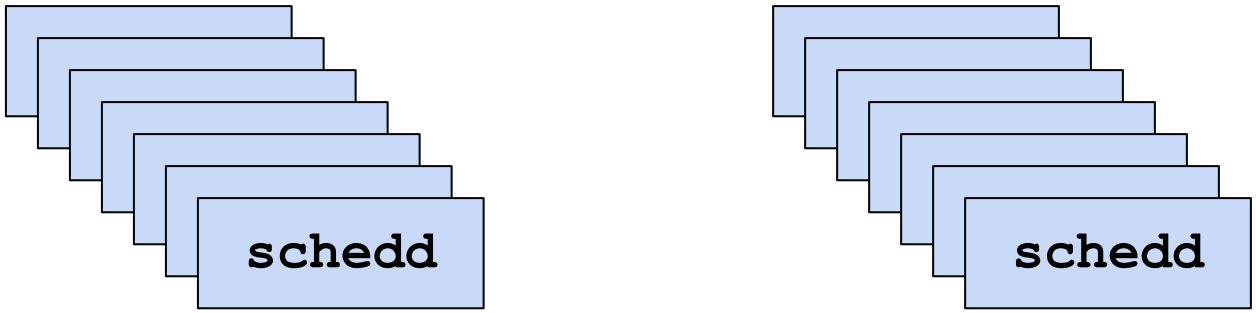
Accounting

Could a centralized accounting service that talks to multiple pools mitigate scaling issues as the number of pools increases without bound?

We are interested in developments coming from HEPCloud, such as the Decision Engine (as a glideinWMS frontend replacelemt) and Acquisition Engine - very different models of managing information (caching) from the current glideinWMS and HTCondor, which rely mostly on real-time queries (which can often fail).

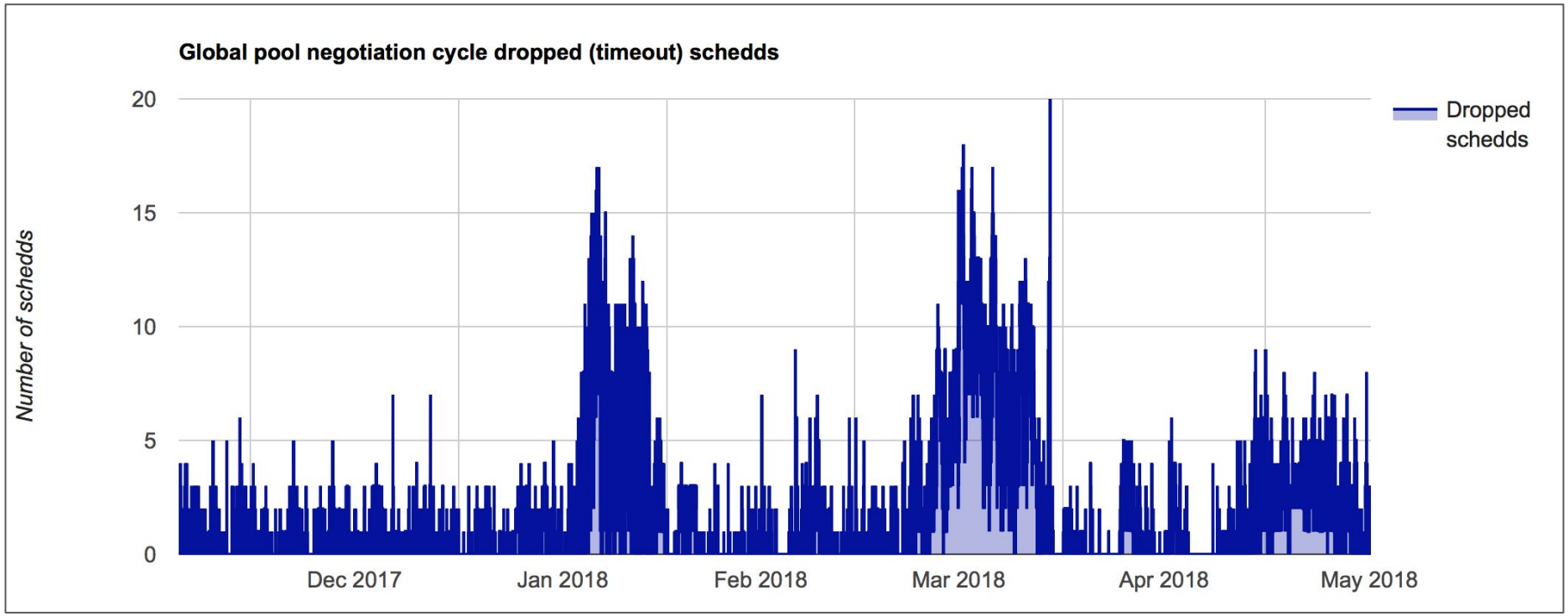
Schedulers

- CMS Production and Analysis workflows are submitted to HTCondor schedd's at CERN and Fermilab.
- Job submission can scale horizontally - currently ~5 active production and ~15 active analysis schedd's.
- Going to O(100) schedulers on stronger hardware in the HL-LHC era should handle the load? To be studied in our 2018 scale tests. Are there limitations?



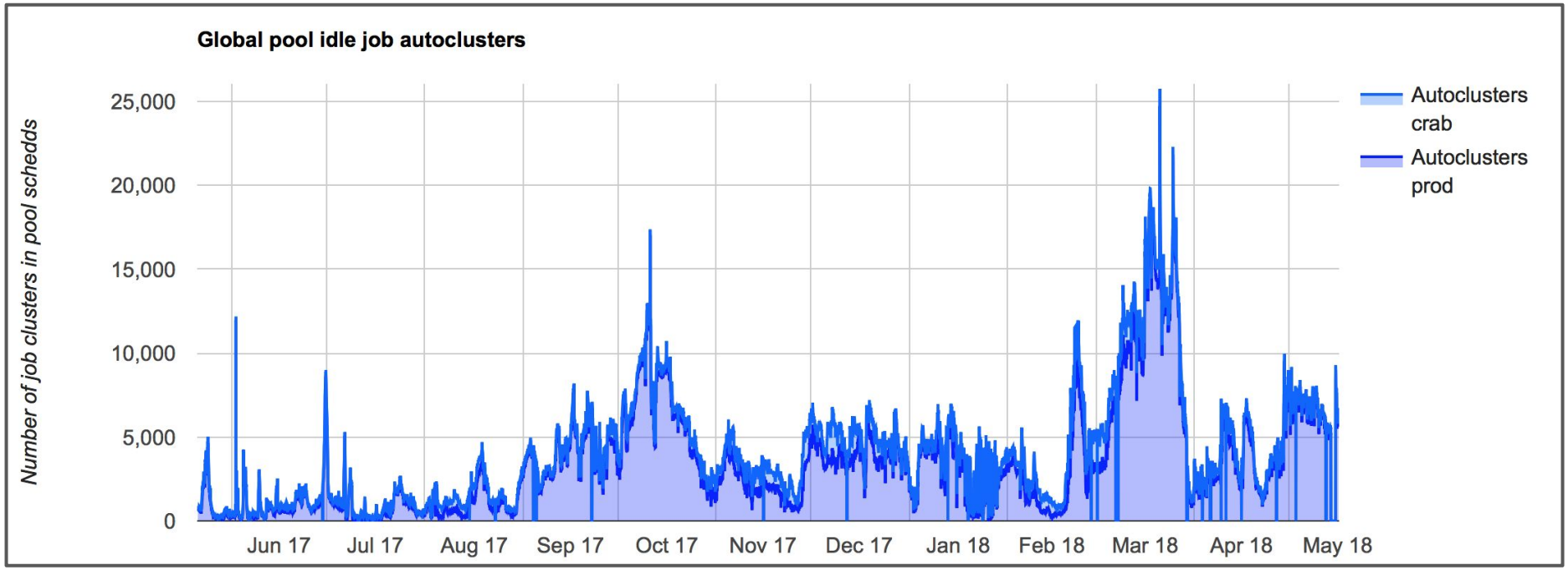
Dropped Schedd's

- Sometimes we overload schedd's, especially when submissions are dominated by single-core jobs and the job numbers explode…
- We drop queries from unresponsive schedd's in the central manager after 60s.



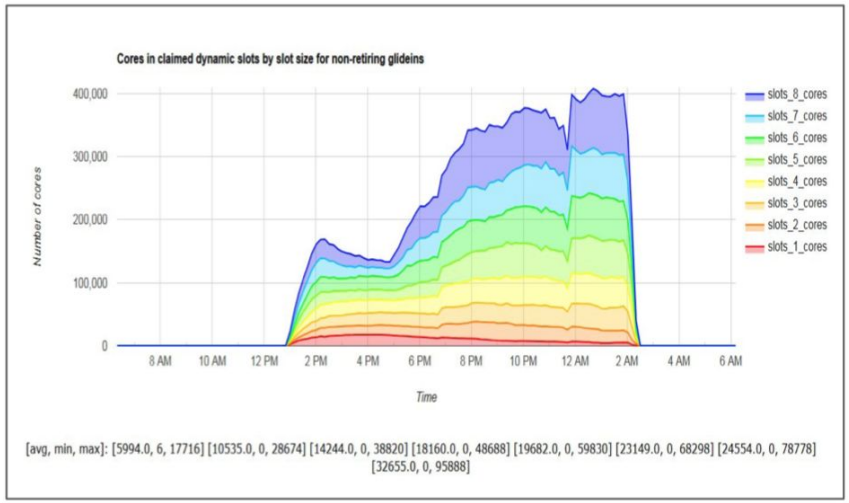
Auto-clusters

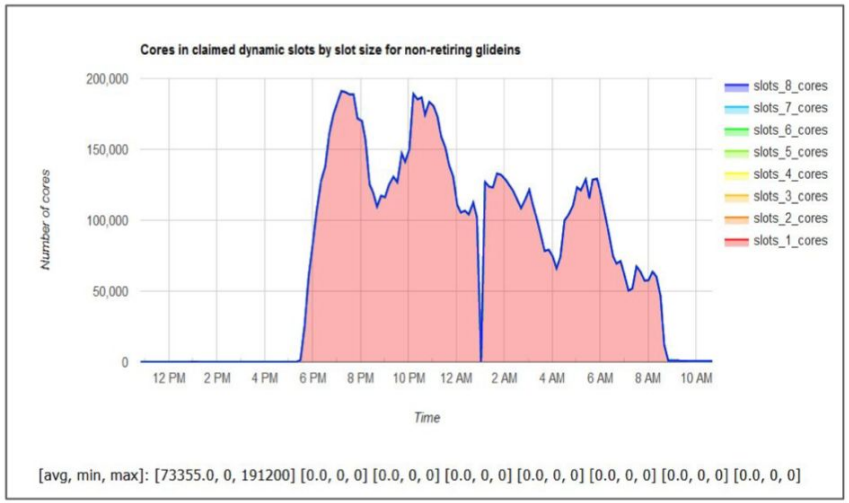
The pool has been able to handle large numbers of diverse resource request lists …



Central Manager Scalability

- In our late 2017 scale round, we pushed the size of a multi-core pool to over 400,000 CPU cores, so we are not worried about the next couple of years.
- Typical peaks in early 2018 are ~250,000 CPU cores.
- However. we still see scaling limitations ~150,000 dynamic slots, for example when production submits a “storm” of single-threaded workflows.





Conclusions

During 2018 we will be studying the scalability of federated pools at large scales. The resource landscape is changing rapidly as we move to HPC & Cloud.

Longer term, we will be looking to HL-LHC scales and the sustainability of HTCondor and glideinWMS to serve our workload management needs in the future.

We wish to thank the development communities of HTCondor and glideinWMS for their continued close collaboration with CMS.

Abstract

CMS has increased the scheduling efficiency of workflows in reusable multi-core pilots by various improvements to the configuration of the glideinWMS pilots, accuracy of resource requests, efficiency and speed of the HTCondor infrastructure, and job matching algorithms. We also continue to study the scalability of HTCondor pools for Run III of the LHC and beyond.