HTCondor Solutions for the Future of the CMS Submission Infrastructure

James Letts
on behalf of the CMS Submission Infrastructure Group

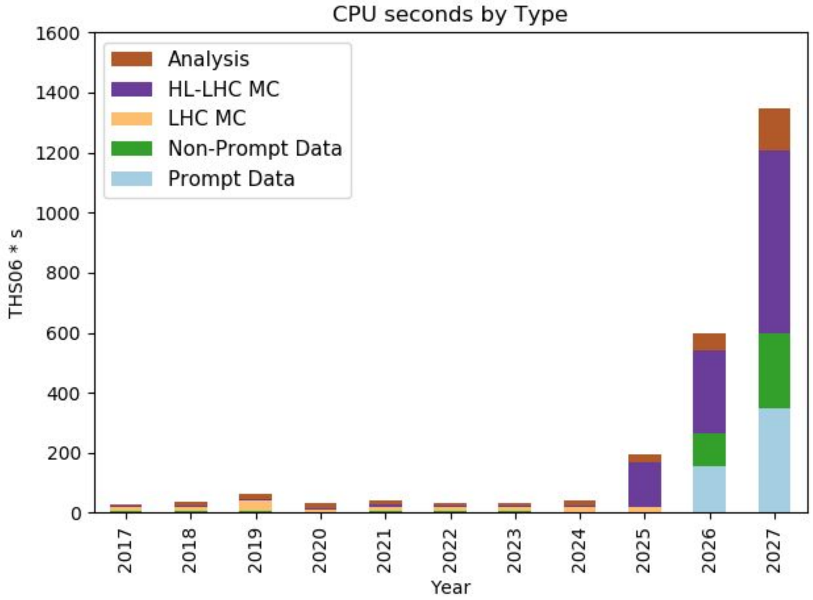
HTCondor Week - Madison, WI (USA)
May 22, 2019

Abstract

The resource landscape for high energy physics experiments in the HL-LHC era is expected to be more dominated by Cloud and HPC rather than traditional Grid sites, reaching scales much higher than current capacity. HPC and Cloud resources are much more diverse and heterogeneous than traditional Grid resources, each with special capabilities or limitations that must be handled by the submission infrastructure. The CMS experiment at the LHC at CERN is actively focusing on the functionality necessary to adapt the HTCondor infrastructure to fulfil these needs and requirements, with enhanced customization and control over resource allocation and usage while maintaining global flexibility. The stability and scalability of HTCondor pools to HL-LHC scales is thought to be tractable, with the increasing complexity of the pool architecture driving limitations.

HL-LHC Scales

- Currently (2019) LHC is in shutdown (LS2)
- Run 3 processing capacity scale: +~50%
- Run 4 (2027) processing capacity scales: **22x current levels**



LS2 Run 3 Run 4

Image Source: CMS Offline and Computing Public Results

The Road Ahead

The resource landscape will change a lot between now and Run 4:

- Cloud and HPC will become the norm, not the exception.
- Existing sites will never be allowed to expand 22x from current capacities.

The submission infrastructure will have to evolve accordingly, both in scale and complexity:

- Trend for large Cloud resources to have dedicated (federated) pools - which fragments (horizontally scales?) the submission infrastructure
- Avoid scaling up 22x in the number of jobs - increase thread count
- Job Sets in scheduling? Schedule larger blocks of jobs.
- We can already schedule GPU’s.

Group Charge

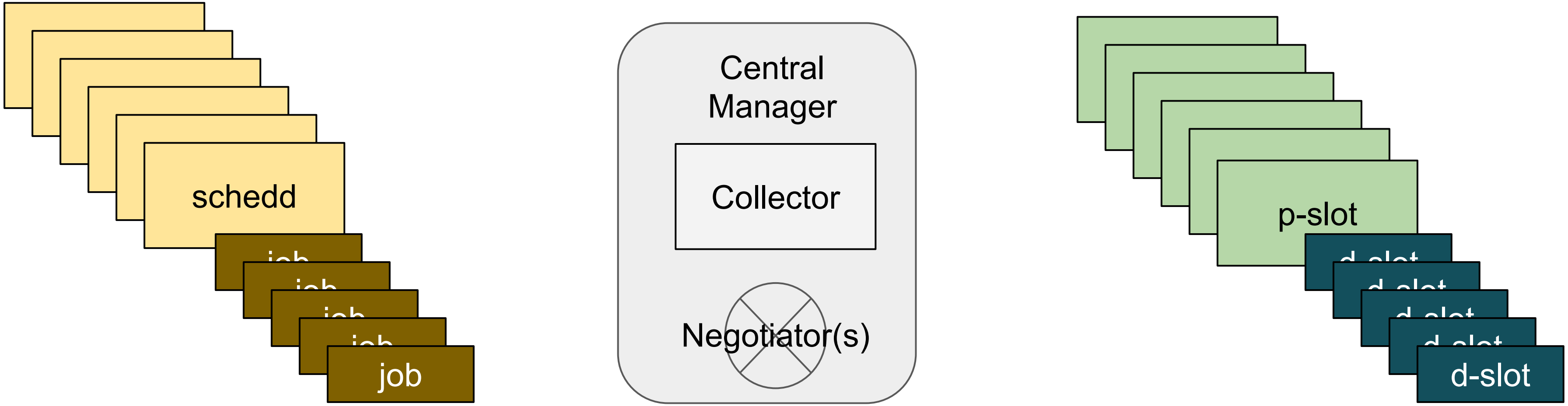
The charge of the CMS Submission Infrastructure Group (part of CMS Offline & Computing) is to:

- Organize GlideinWMS and HTCondor pool **operations** in CMS
- Communicate CMS **priorities** to the development teams

SI activities fall broadly into 3 categories:

- Overcoming current operational limitations or problems
- Integration of new, diverse resource types and submission methods
- Preparing for future scales or feature requirements

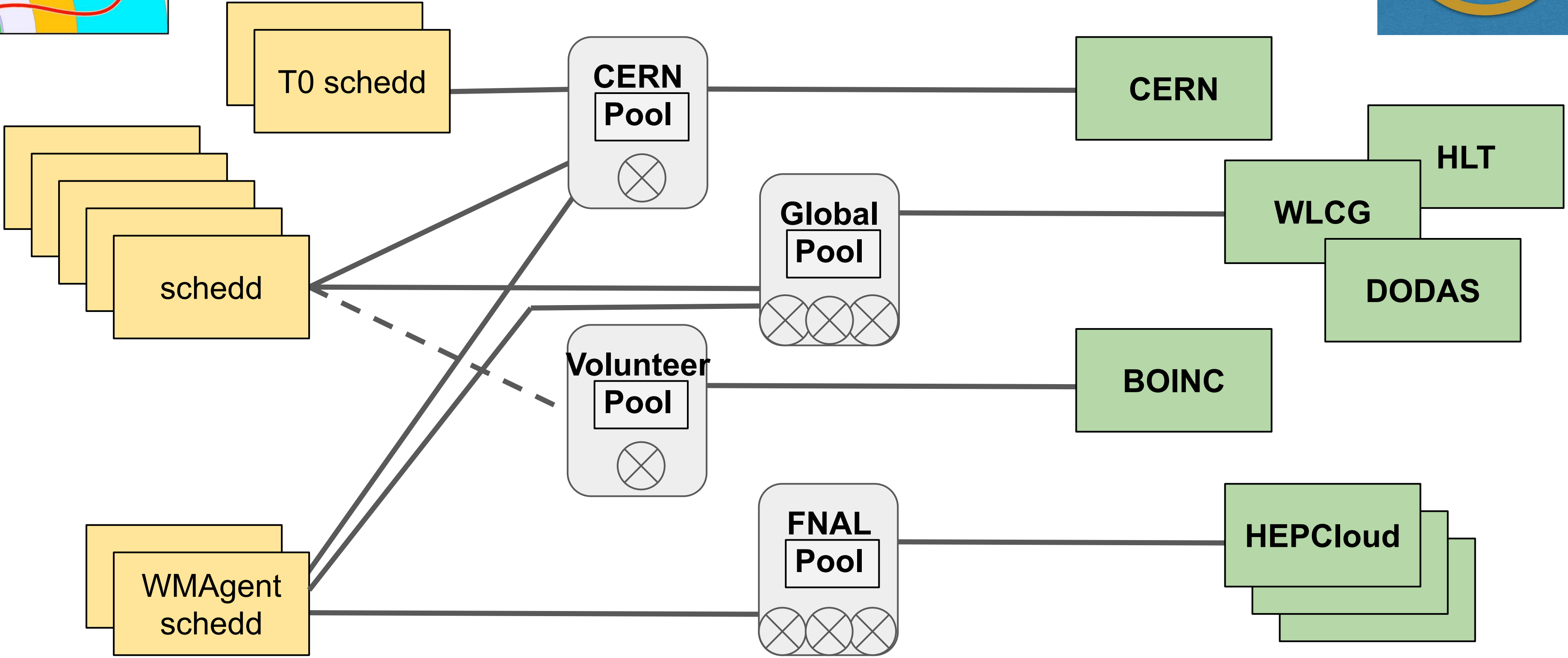
HTCondor Pool



Schedd's can connect to more than one Central Manager (federated pools).

Machines (startd's) can connect to only a single Negotiator.

HTCondor Pools - Current Overview



Federated Pools

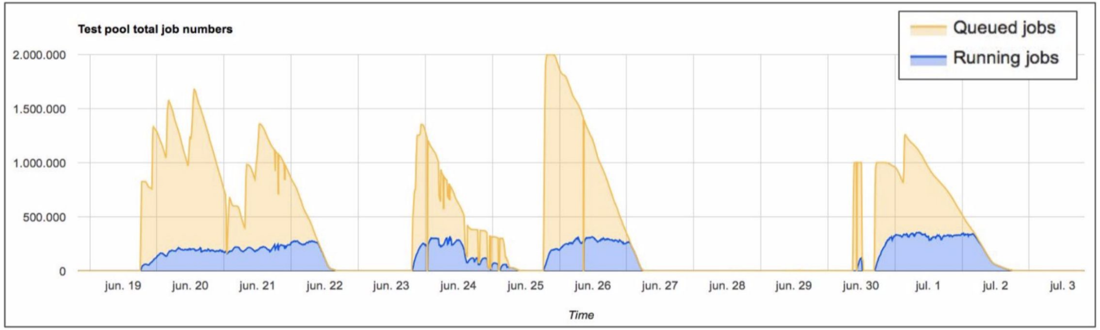
Trend for large HPC or Cloud resources to have dedicated (federated) pools, which fragments (horizontally scales?) the submission infrastructure on the resource side.

- HEPCloud in the Fermilab Pool - schedd’s can flock to the Global, CERN pools.
- Other Cloud sites may have local HTCondor pools or infrastructure.
- Some motivations are to have more control over costs or on types of workflows that can run on the Cloud or HPC resource.
- CMS has made efforts on site customization of start expressions (later in 2019).
- Risk: To how many pools can a schedd flock work? CMS wants to study this in 2019 scale tests.

Current Scales

- Typically ~700 analysis users, single production user.
- WMAgents and CRAB TaskWorkers materialize jobs - not yet using late materialization (but this is in the works)
    - ~30 active schedd's talking to 1-4 pools, some with up to 3 Negotiators.
    - Typically 1-2M jobs materialized in the queues, O(100) tasks/schedd
- JobRouter used extensively on the schedd's to overflow jobs to 'nearby' processing sites (and read input data remotely), adjust run time or memory:
    - We have to be careful to not overwhelm the LAN or WAN for any sites.
    - Sometimes JobRouter overloads the (already overloaded) schedd's
    - Largely flying blind, however - no network monitoring

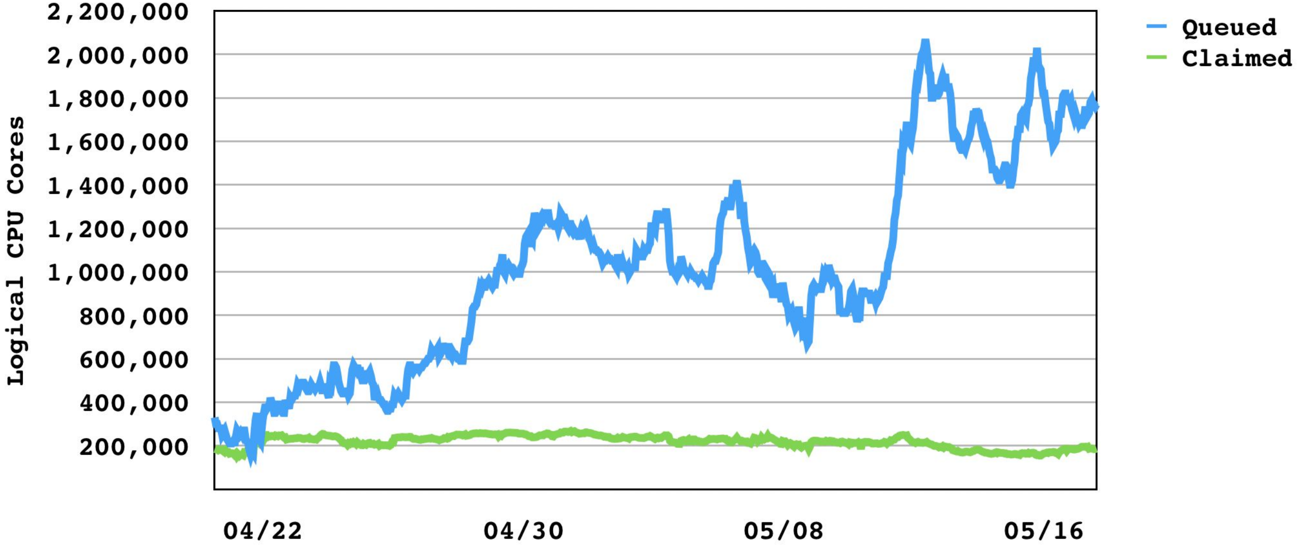
Scalability & Stability



- Maximums achieved in the scale tests:
    - 350K dynamic slots number of jobs)
    - This is ~2x current levels.
    - Equivalent to a ~600K CPU core pool with current thread mixture.
- Scaling limitations driven by the **number running jobs** and **number of idle CPU cores**.
- Rate at which schedd's can launch new jobs was an issue in early rounds.
- Likely safe through the end of LHC Run 3, +50% purchased resources.
- Bursting into the Cloud at large scale might be challenging in a centralized pool.

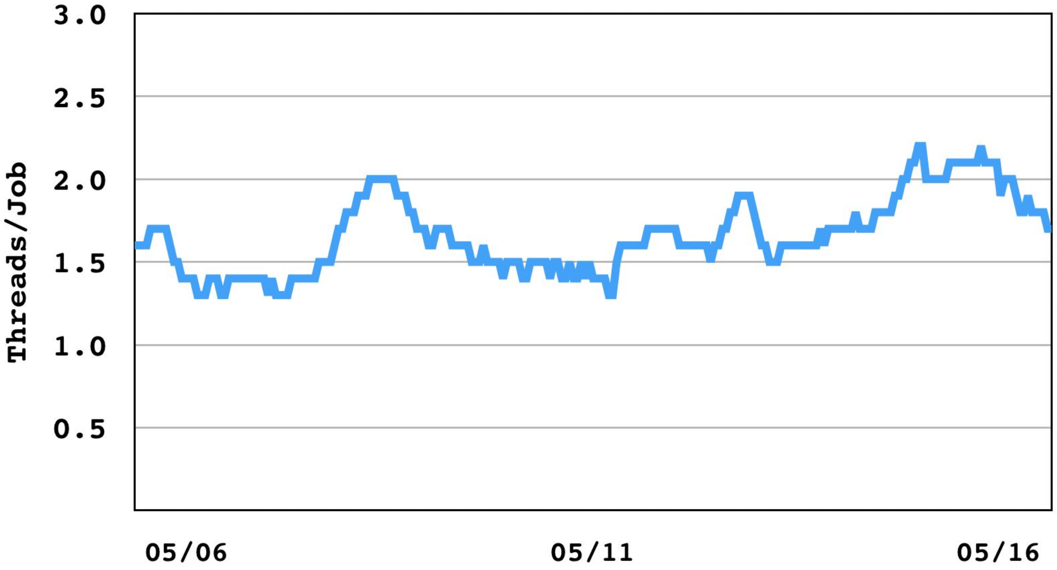
Scales - CPU Cores

- Up to ~200,000 dynamic slots running on up to ~300,000 logical CPU cores.
- We typically schedule on CPUs and Memory, but not network.



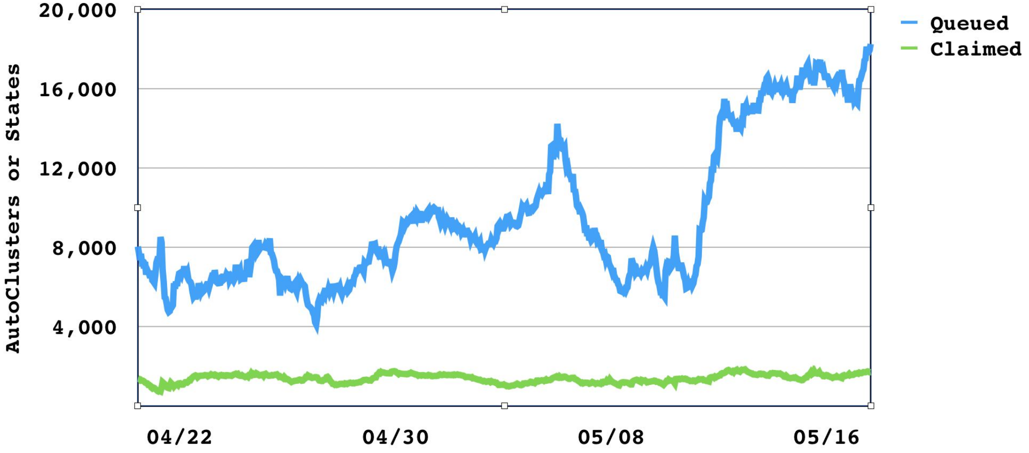
Thread Count

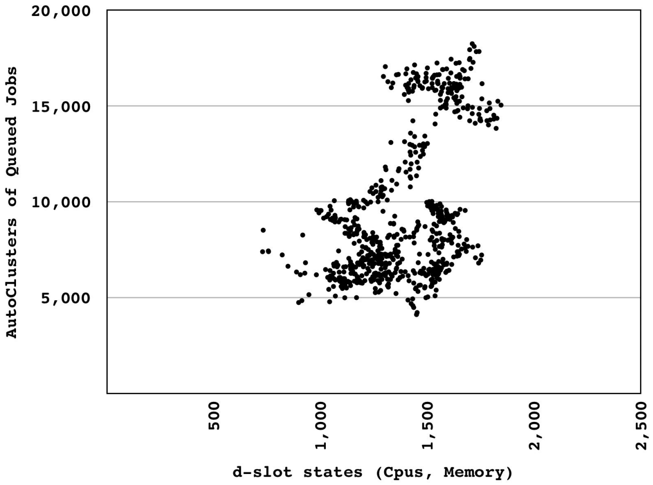
- Multi-core pilots enable increase in the thread count of payload jobs, reduce combinatorics of matching.
- Still running mostly single-threaded jobs
- Improving this would help reach HL-LHC scales by reducing the number of jobs.



Complexity

- Typically ~1-2K distinct combinations of CPU count and memory available in d-slots (green)
- Driven somewhat by complexity of resource demands (AutoClusters) of queued jobs (blue)
- More diverse requests result in somewhat more fragmented resources.

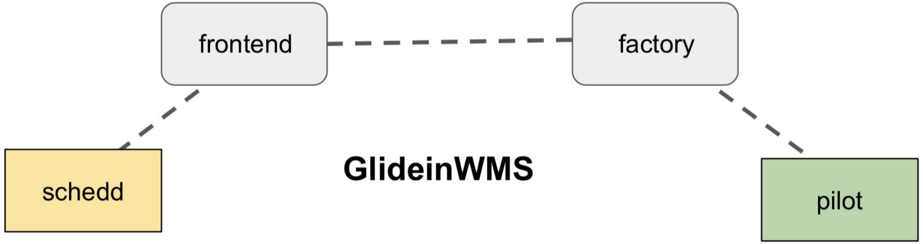




Resource Provisioning

Several resource provisioning systems in production currently. As the resource landscape becomes more diverse, provisioning mechanisms become more complex. One-size-fits-all pilot model less applicable.

- **GlideinWMS** (pilot job model): frontend queries schedd’s, requests pilot submission from factory to Grid-accessible site based on need (“pressure”).
- **BOINC**: central server queries schedd’s; client running on remote machine; startd connects to HTCondor pool (cms@Home).
- **Vacuum** model: DODAS, e.g. (machines bring up startd’s if others are claimed)



Scalability & Stability

- We do not know where the next scaling limit is.
- **Some 2019 Test Goals:**
    - Re-checking the 2018 with improved details:
        - Effect on job start rate of schedd sandboxes
        - Recent improvements in glideinWMS frontend matchmaking cycle
    - Multi-threaded Negotiator promises to give us even more headroom.
    - Where are the schedd’s spending their time?
    - What are the limitations on flocking or the number of federated pools?

Reaching HL-LHC Scales

- How do we get a factor of 22 improvement in scale by HL-LHC?
- Increase thread count from 2 to 8: x4 or more ?
  - Step-chains may help - unite all the data processing steps into a single multi-threaded application instead of individual jobs of varying core count.
- Unimplemented improvements such as the multi-threaded Negotiator: x2 ?
- Another factor of 2-3 should be possible to find before 2027.
- Would Job Sets allow us to horizontally scale without limit, ability to move entire tasks to other schedd's, pools?
  - Points us to the next frontier for improvements: **workflow management**

Workload & Data Management and Monitoring

HTCondor pools don’t exist in a vacuum. Upstream:

- Workflows submitted from **CRAB TaskWorker** (physics analysis users) and **WMAgent**’s (central processing and reconstruction) - CMS-only projects.
  - N.B. One WMAgent currently tied to a single schedd.
- **Unified** is an operations tool that reads monitoring, coordinates data placement, and manipulates the schedd queues accordingly.
  - Implements policies by querying various data sources: PhEDEx (DM), monitoring of the pools and sites (cms-gwmsmon), etc.
  - In this way it is like a policy engine.

Workload & Data Management and Monitoring

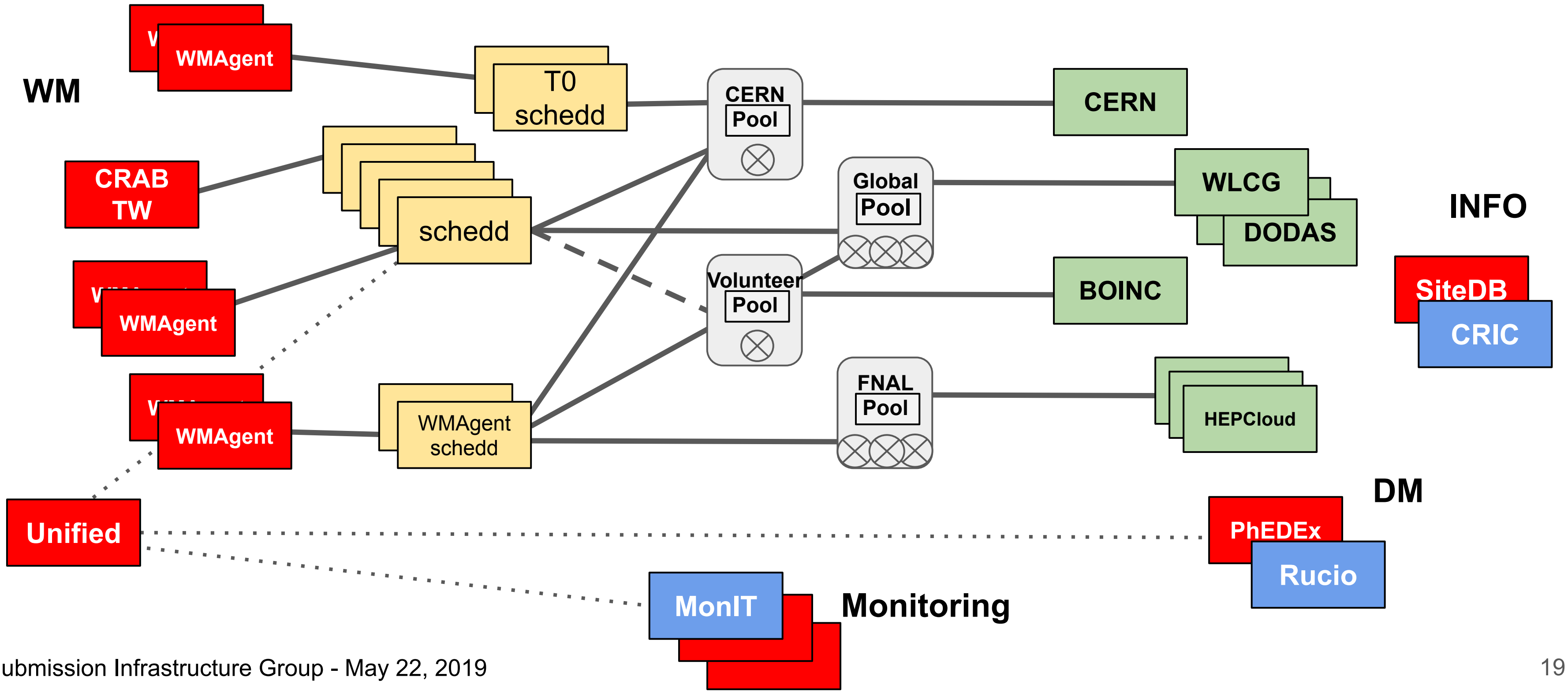
CMS trend to adopt community software projects to replace legacy services:

- **Rucio** - community project for **data management** (input files and output data of jobs) - replacement for PhEDEx
- **MonIT** - community project for **monitoring** - job, HTCondor pool, site monitoring - replacement for cms-gwmsmon, Dashboard, etc.
- **CRIC** - community project for **information services**, e.g. site attributes

Each of these could become a data service for a policy engine.

CMS batch (HTCondor) and resource provisioning (glideinWMS) software has been developed externally for many years.

Workload Management (WM), Data Management (DM), and Monitoring



The Bottom Line

- Trend in CMS is to move to community projects.
  - Outcome of last year’s Data Management Review was to adopt Rucio.
  - We are very interested in Decision Engine.
- We also carried out a Workload Management Review. There is no obvious community project that can satisfy our needs.
  - But is our problem so unique? Involves coordinated scheduling of workloads, data movement, networking, fair-shares at the site level, etc. ⇨ *A multi-dimensional scheduling problem which would need input from multiple data sources.*
  - We tried to adopt PanDA in 2013 and failed.
- **We don’t have the tools in the SI to make these kind of scheduling decisions, or let sites or WM system make them, even if the job’s requirements could be written down.**