



CMS Scheduling Goals in Simple Language

James Letts

on behalf of the Submission Infrastructure group
of the Offline and Computing coordination area
of the CMS Experiment at CERN



HTCondor Week - May 19, 2020

Abstract

In scientific computing, getting your stakeholders to write down in non-technical language how they want the system to behave seems like a simple exercise but is actually quite tricky. Staying away from discussions of technical implementations, we will try to describe CMS workflows and CMS scheduling policy goals and constraints in plain language. The coordinated provisioning of processing power, storage, and networking according to organizational goals requires a clear understanding of what you are trying to achieve and the constraints or boundary conditions on the system. Once the goals are understood, the technological roadmap to reaching them hopefully becomes clearer.



“Lasciate ogni speranza, voi ch'entrate.”

- Dante

CMS Workflows

Attempt to describe (most) CMS workflows in simple language:

- Different types of requests: data processing, Monte Carlo generation and physics simulation, detector reconstruction, physics analysis, as well as new types such as gridpacks, TensorFlow, interactive analysis, etc.
- Common resource requirements:
 - Processing (CPU and/or accelerators), Memory
 - Input data (disk and network) and conditions (calibrations, e.g.)
 - Significant (or not) disk I/O
 - Output data (disk and network)
- Some workflows are staged (and can be chained): output of one is input for the next.

Seems simple?

“Il diavolo si nasconde nei dettagli.”

- Italian Proverb

CMS Resource Landscape - Processing

- Not just Intel x86 anymore
- Sometimes hyperthreaded
- “Minimum” 2GB / logical CPU core
- ~200,000 “traditional” Grid-connected CPU batch cores, plus another ~100,000 opportunistic.
- Increasingly more allocations on HPC or Cloud resources, but no storage (and sometimes no WAN network, no calibration database access, etc.)
- With different accelerators in some cases (more common in the near future!)

CMS Resource Landscape - Storage

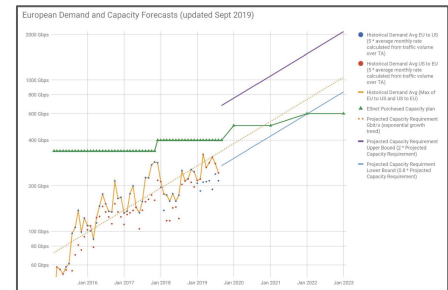
- Tiers of Grid-connected processing co-located with disk and tape storage are a relic of the original computing models from the late 1990's when network was “expensive”.
 - CMS jobs can now read data locally or remotely from disk or tape...
 - Network turned out to be really cheap.
- Caches and data lakes (reading over the WAN) a huge area of research.
- HPC & Cloud does not typically offer a storage allocation, so connectivity (over the WAN) to these storage solutions will be key.

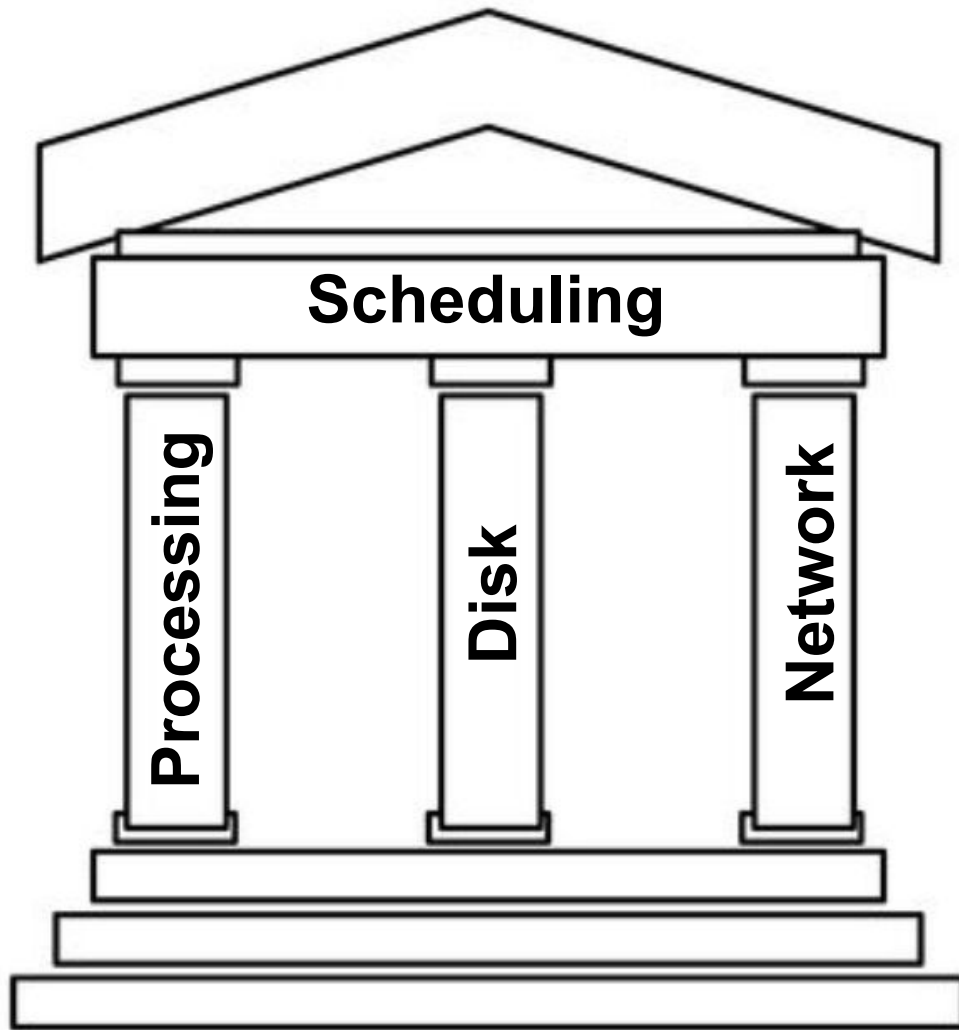
CMS Resource Landscape - Network

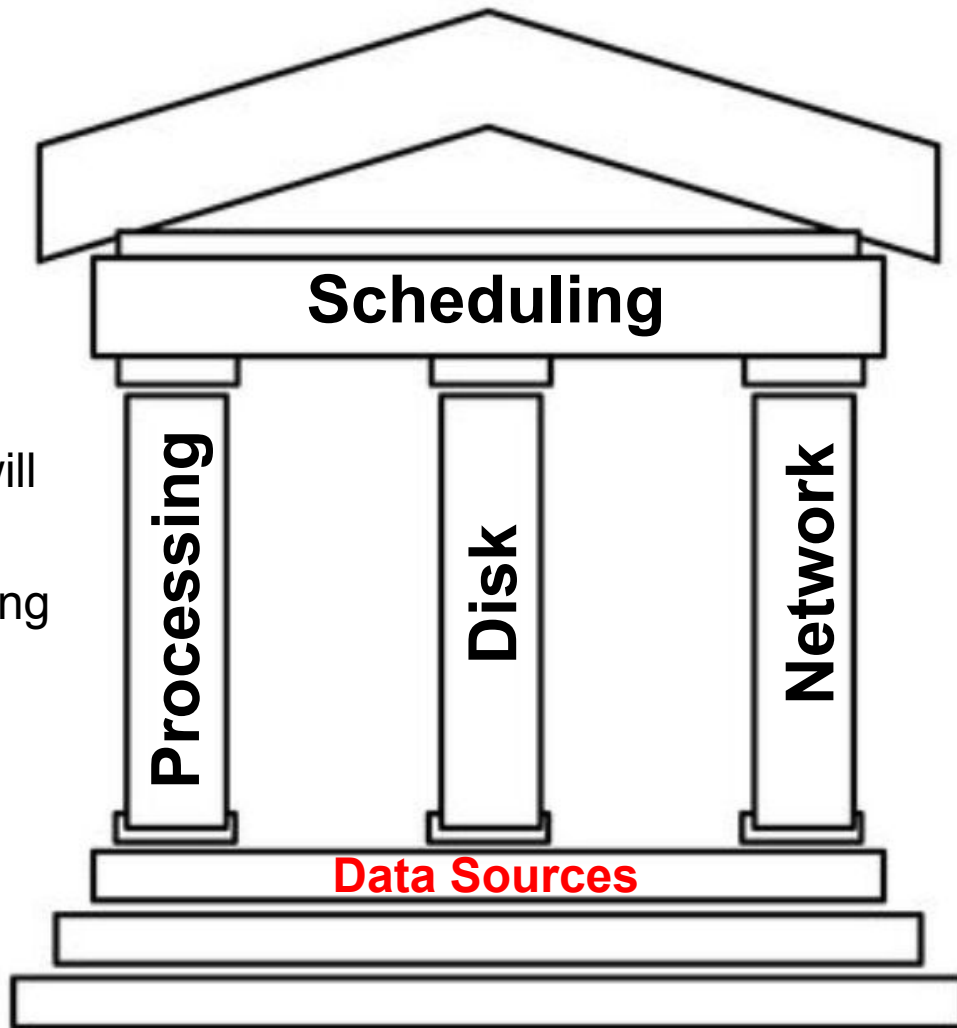
- Traditional Grid sites:
 - About ~60 in number world-wide
 - Typically at least 10 Gbps WAN connection, up to 100+ Gbps
 - LAN typically 1-10 Gbps
 - “Free” for us, and unmanaged
- HPC & Cloud sites typically have restrictions and costs associated with network (and storage).

Evolution towards HL-LHC Scales

- Increased data rates and event complexity will drive scales of computing at least an order of magnitude higher from today.
- Factor of 2~4 between resource needs and projections under flat budgets.
- Lots of work to improve the code base and data formats.
- We also need to be **smarter about how we schedule** workflows (processing), data storage (disk), and data staging (network).
 - Network is something we are used to treating as free and limitless.
 - Active areas of investigation:
 - Packet marking (IPv6)
 - Traffic shaping
 - Orchestration (bandwidth reservation)

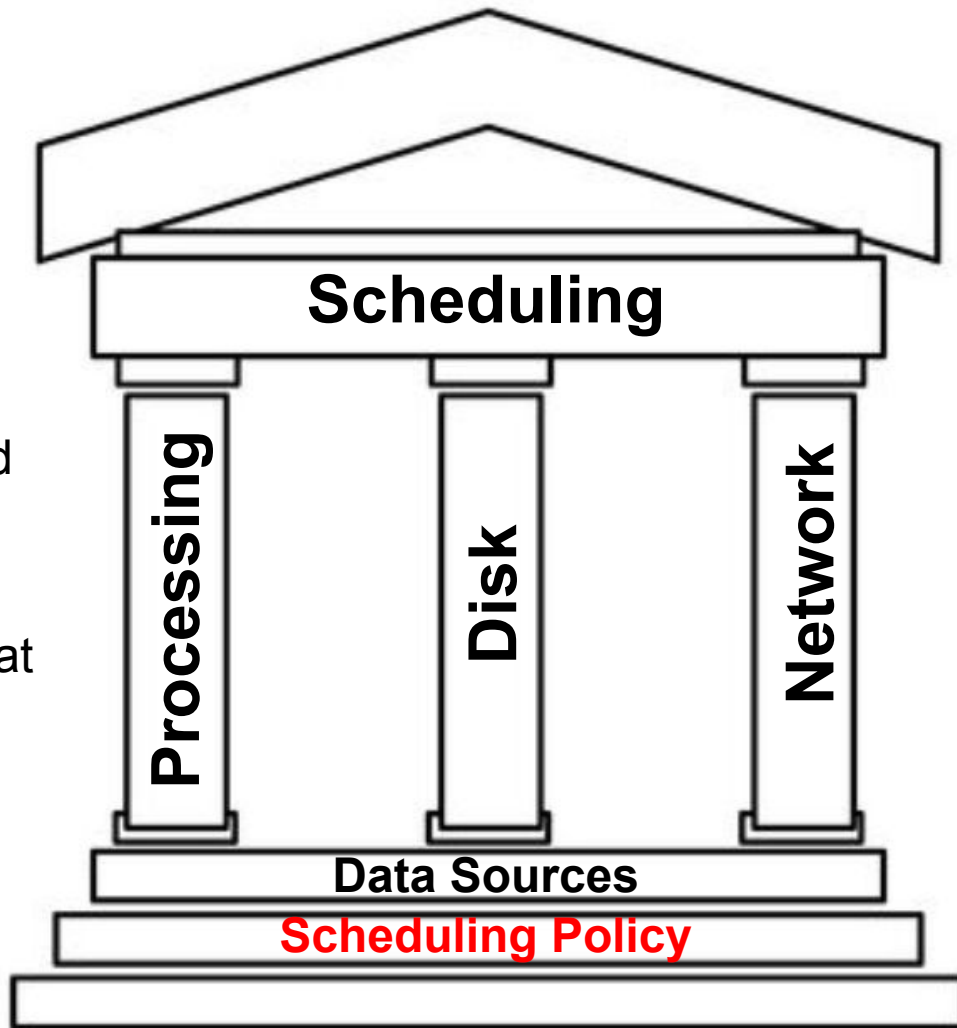






What data sources will we need in order to make smart scheduling decisions between these areas?

Now that we understand the broad strokes of CMS workflows and resources, let's look at writing down a scheduling policy.



“Il meglio è nemico del bene.”

- Italian proverb, quoted by Voltaire, and also by quite a few of my colleagues.**

What is “Good” Scheduling?

- Different **stakeholders** have varying ideas of what “good” is ...
- This is part of the challenge of writing down a “CMS” scheduling policy.

Funding Agencies (FA's):

- Like to see all of the resources used all the time efficiently for cutting edge science. By “resources” they mean processing (and accelerators) used with high efficiency, storage with little wastage on never-read files, and network as little as needed to move the data around (i.e. they don't want you to pin the shared WAN).
- Rationale: Why pay for a site you aren't using?

What is “Good” Scheduling?

Physics Analysis Users:

- Want a personal workflow (either centrally launched or personally) to complete in a predictable amount of time with limited manual intervention.
- Don't usually care about efficiency or cost, just time to completion and success rate.

What is “Good” Scheduling?

Central production:

We usually hear from them about what they don't want:

- A high-priority workflow is taking too long to complete.
- Lower-priority work is crowding out higher-priority work (priority inversion).
- Want to run a workflow at scale that requires one CPU core and 20GB of RAM. Workflow requirements come from upstream requests.
- Need to minimize job failures. Failures = manual cleanup and the various teams are effort-constrained. Tendency to request resources for the tails rather than the average so the outlier jobs don't crash.

What is “Good” Scheduling?

Central production:

- Need to co-schedule data transfers (disk & network) with processing job since inputs are staged on limited disk buffers. Major potential bottleneck to throughput.
- Does CMS want workflows with equal “priority” to finish at the same rate, or FIFO? To the exclusion of all other lower priority work? We have never settled on a clear (simple language) scheduling policy with respect to workflow prioritization.
- Data taking (“Tier-0”) workflows are the highest priority at CERN.

What is “Good” Scheduling?

Submission Infrastructure Team:

- Don't break the infrastructure by submitting impossible workflows:
 - Single CPU core and 70 GB of RAM user task ... Yes, this has happened ... and yes, we have machines that can run them. (We put limits on this subsequently.)
 - Hit pool scaling limits (many single-core, very short jobs ... “storms”)
 - Saturating **limited** resources that we do schedule (or should in the future):
 - Saturate network (WAN)
 - Saturate LAN at a site
 - Saturating memory on worker nodes: OOM incidents
- Fair-share between groups of stakeholders (central processing, analysis, Tier-0)
- We are one group looking at scheduling efficiency (FA concern).

Writing a CMS Scheduling Policy in Plain Language

Now try to boil this down into a COHERENT and SELF-CONSISTENT scheduling policy. My attempt last year [bold type added]:

*There are two **targets**, **number** of cores at a site and % of the total pool. In terms of ordered priority (highest to lowest):*

- *The **number** of cores at CERN for “Tier-0” workflows is the most important **target**.*
- *The total **number** of cores across all sites for “production” workflows would be the second-most important.*
- *Having a minimum **percentage** of cores for “analysis” workflows at any given site (assuming there is demand) would be the least priority. The minimum may be less than the **target percentage** that we would like to have ... i.e. the **target** at any given site may be 25% but we wouldn't like this to fluctuate under, say, 10% (then people complain).*

Writing a CMS Scheduling Policy in Plain Language

I only concentrated on the fair-share between groups ... What was important to “me” ... but there are at least 7 “aspects” that would go into a coherent SCHEDULING POLICY, since we are dealing with stakeholders with different ideas of what is important.

The aspects:

1. Maximization of utilization (only pay for what you need)
2. Minimization of wastage (poor CPU efficiency, job failures) (don't waste what you paid for)
3. Fair-share (get what you paid for)
4. Prioritization (some of us paid more than others)
5. Predictability (know when you are getting what you paid for)
6. Coordination between workflows, storage, and transfers (make a great supply chain)
7. Scalability (don't break the supply chain)

Attempt at a CMS Scheduling Policy

CMS wants to use efficiently all the resources at every Grid, HPC, and Cloud site pledged or allocated to us with a tunable fair-share balance between and among groups of users (Tier-0, production, analysis users) and a prioritization of workflows [insert relative scheduling priority policy here] with predictable workflow completion times. The submission infrastructure should work in an overall system where workflow management, data management, and network management are all coordinated to maximize throughput, minimize bottlenecks, and not overflow any of the supporting components of processing (RAM), disk, or network.

And now for something completely technical...

- Tier-0 **workflows** negotiated first, especially at CERN.
 - Done with accounting groups. Could be done with job sets.
- HEPCloud **resources** negotiated as a block?
 - Slow to spin up, so jobs often land elsewhere first.
 - Serial (rather than parallel) negotiator over resource sets? Give a chance to every resource set to match jobs.
 - Temporary work-around by routing sets of jobs to local schedd.
- CSCS resources another example: large Tier-2 with strictly 2GB RAM per core. Hard to compete in parallel with other sites.
- Analysis not crowded out at any single site (local stakeholders!)

And now for something completely technical...

- Throttling potentially destructive or embarrassing workflows:
 - Implement policies where malformed or extremely CPU-inefficient workflows could be limited in number (at a site)?
 - Very short jobs, or very high memory workflows.
 - WAN (remote reads) or LAN-heavy workflows scalable per site or region?
- **Central production often risks running out of input disk buffer at sites, which would idle processing**, i.e. input data is not where it needs to be (or conversely, the processing is at the wrong place). How to coordinate WM, DM, and NM with shared information?

“So long, and thanks for all the fish.” - Douglas Adams

CMS is very grateful for the close cooperation the HTCondor developers have given us, and the other groups we work with, for many years.

I have enjoyed giving this presentation that has taken a year to write ... since Miron asked me a very simple question.

Hopefully we have given you a good overall picture of what we are trying to accomplish with a scheduling policy and why.

Un(?)fortunately I will probably not be giving this talk next year, as I will take up the position of co-coordinator of CMS Offline & Computing at CERN for two years starting in September 2020.

Where to read more:

- Antonio's [talk](#) in the [HSF/WLCG Workshop](#) last week on integrating heterogeneous resources in the workflow management and submission infrastructure of CMS.
- CMS [presentation](#) at CHEP '19, "Evolution of the CMS Global Submission Infrastructure for the HL-LHC Era".