

Federation of compute resources available to the German CMS community

R. Florian von Cube*, Manuel Giffels, Christoph Heidecker, Günter Quast, Martin Sauter, Matthias J. Schnepf

Computing in Particle Physics

The challenges ahead

- LHC's luminosity will increase drastically in the future. Hence, the increase in data and required simulation will result in shortcomings in computing resources.
- Assuming flat funding, progress in technology is expected to advance computing power by 20%.

Modern Technologies

- Remaining shortcomings have to be overcome by using efficient, modern technologies in analysis techniques and by exploiting resources from different providers (see poster #467).
- Data lakes and modern caching concepts (see poster #464) enable intelligent data distribution.

Distributed Heterogeneous Resources

- Results in a more heterogeneous resource environment in the future.
- Managing this poses a huge challenge for computing operations of experiments.

In order to facilitate the computing operations of experiments, the idea is to *federate national computing resources with a single point of entry*. This allows computing operations of an experiment to submit jobs to a single national submission infrastructure and the nationwide distribution of the jobs on specific resources is done locally.

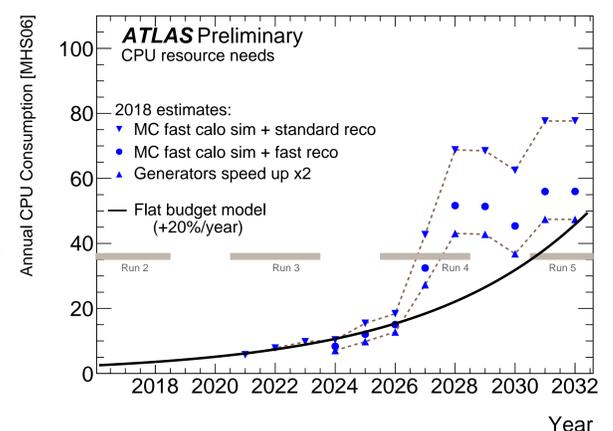
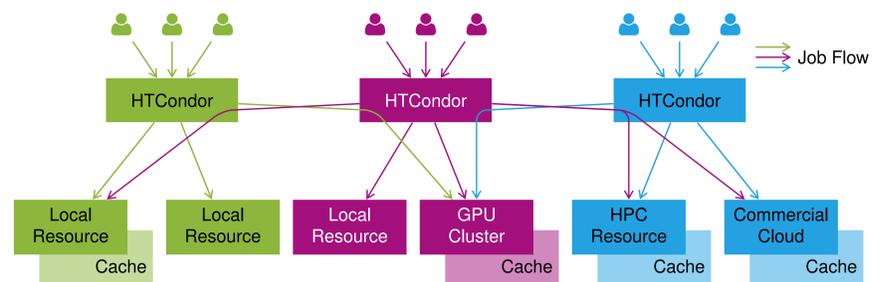


Figure: <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults>

Federate computing resources using HTCondor's flocking and routing mechanisms

- In order to utilize available resources most efficiently, connect resource pools and send jobs to the most fitting resource.
- The pools flock to each other, allowing jobs to move freely.
- Users and hosts are authenticated and authorized using already deployed grid security infrastructure (GSI) X.509-certificates. This allows easy implementation for participating sites.
- Opportunistic resources are used to expand the corresponding local resource pool.



HTCondor [1]

HTCondor is a batch-software for managing the distribution (scheduling) of computing payloads onto resources. While being highly dynamic, it is able to scale to large pools.

Flocking in HTCondor

Flocking allows for connecting HTCondor resource pools in order to send jobs from one pool to another. A job becomes eligible for flocking, if in the local pool, no match is found.

Routing in HTCondor

Every entity of the HTCondor universe is described by a list of so-called ClassAds. A router allows to modify ClassAds of submitted jobs. This allows to append requirements to run only on specific resources, or be sent to another resource pool.

Example Setup: Including a High-Throughput-Cluster

GridKa High Throughput Cluster

- Throughput optimized analysis system designed for different user groups at GridKa.
- 1 PB integrated cache volume.
- 100 Gbit/s to LHCONE.

Current status

- Transparent integration of high throughput cluster for ETP users via flocking in progress.
- Authentication and authorization is done using X.509-certificates.

Next Steps

- Test setup with external groups and resources, collaborations already established.
- Implement accounting for resource providers.

[1] HTCondor: <http://research.cs.wisc.edu/htcondor/>