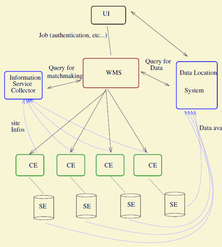
Evolution of the Distributed Computing Model of the CMS experiment at the LHC

Abstract

The Computing Model of the CMS experiment was prepared in 2005 and described in detail in the CMS Computing Technical Design Report. With the experience of the first years of LHC data taking and with the evolution of the available technologies, the CMS collaboration identified areas where improvements were desirable. In this work we describe the most important modifications that have been, or are being implemented in the Distributed Computing Model of CMS. The Worldwide LHC computing Grid (WLCG) Project acknowledged that the whole distributed computing infrastructure is impacted by this kind of changes that are happening in most LHC experiments and decided to create the Technical Evolution Group (TEG) aiming at assessing the situation and developing a strategy for the future. In this work we describe the CMS view on the TEG activities as well.

Workload Management



Baseline model

Classical push model, uses the workload management systems provided by the grid projects (currently the EMI-WMS)

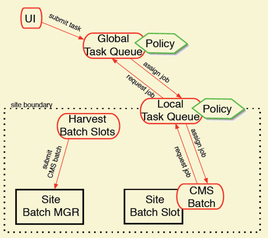
Intrinsically scalable but heavily dependent on the Information System.

Proposed in the Computing TDR in 2005 and still in use.

2-Layers pull model

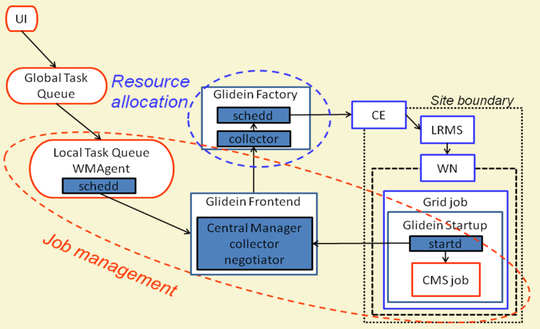
Consists of a global task queue and local task queues at the sites. Resource allocation is done at the site by CMS components. Also presented in the Computing TDR in 2005 but never implemented.

Independent of the Information system but requiring CMS specific services at the sites.



Evolution

Based on the 2-layer pull model but the Local Queues and the resource harvesting services are outside of the sites. Relies on the glidein-WMS.



The use of the new architecture is transparent for the sites but the splitting of resource allocation and job management gives much more flexibility, making the job management chain independent of the infrastructure.

Security

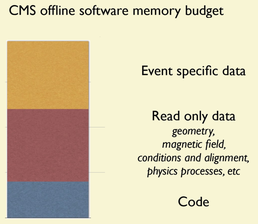
Grid Authentication Authorization and Auditing (AAA) happens on the Grid Computing Element. In the new model an additional AAA step is required on the Worker Node.

The glidein and CMS job execution environment need to be separated to protect the glidein form malicious CMS jobs.

Glexec is a piece of Grid middleware providing the needed functionality. The WLCG Technical Evolution Group recommended its deployment on the WLCG infrastructure.

Multi-core Jobs

A substantial fraction of the memory of a job is common to all events processed. Instead of analyzing events in individual single-core jobs it is more convenient to process events in parallel in the same multi-core jobs sharing the common part of the memory (Code and read-only data).



A typical CMS job executed in parallel on 8 cores uses about 25% less memory than 8 individual single-core jobs. For this reason CMS wants to move to a multi-core resource allocation model.

The most natural multi-core allocation method is delivering whole nodes. Advantages are:

- reduced complexity and scale;
- no interference among users sharing the same host;
- naturally evolving to a cloud allocation model.

Virtualization and clouds

The Cloud resource allocation model does not add much to the current Grid model with whole nodes allocation. CMS will not have difficulties adapting to such a model if needed.

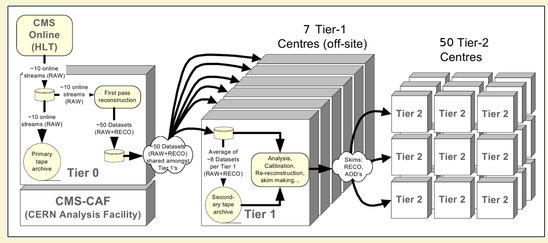
Virtualization is a particular aspect of clouds that is actually independent of the use of a cloud interface

CMS acknowledges that virtualization is a useful technique to efficiently manage multi-VO sites. As far as the provision of resources is transparent to CMS and with no significant performance penalization, it is acceptable.

Storage and Data Management

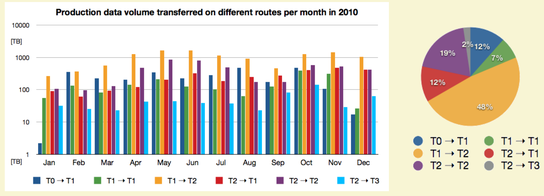
Data Distribution

The original CMS data distribution followed a hierarchical model proposed by the MONARC project.



Already in 2009/2010 CMS decided to implement the full mesh allowing transfers from any Tier1/2 to any Tier1/2.

A significant amount of traffic was observed on the Tier2-Tier2 routes already in 2010.

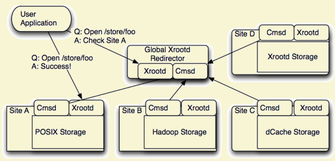


Automation of data placement/deletion is now under development to optimize storage resource usage. A Data Popularity system is already in production to monitor user activities.

Storage Federations

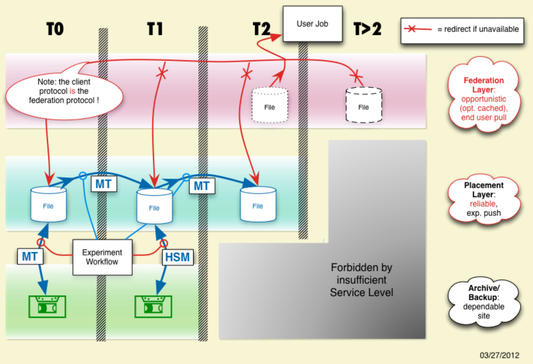
The CMS original model requires that running jobs only access data stored at the site where they run. Reduced cost and increased reliability of networks allows to access data remotely. CMS decided to allow remote data access in a few cases:

1. unavailability of the local copy of a file (e.g. for corruption) and transparent fall back on a remote copy;
2. using low I/O processes, i.e. visualization programs;
3. to address site congestion, when the available copies of a dataset are at overloaded sites;
4. to increase the utilization of CPU power at sites where proper data management is not possible (e.g. Tier3s)



CMS used xrootd to implement this behavior. A network of xrootd redirectors implement the federation of site storage.

The concept of federation layer was formalized during the work of the WLCG Storage and Data Management TEGs.



The Federation layer works on top of the already existing Archive Layer and Placement Layer and provides extra data availability capabilities.

Tape-Disk separation

The CMS model assumes only one visible storage end point per site. Tier1s provide access to their MSS system transparently through a T1D0 SRM service class.

CMS foresees to explicitly distinguish the disk and tape resources in order to:

- better protect the MSS from unwanted access (e.g. remote access, user analysis);
- be more flexible in the definition of what needs to be archived;
- be more efficient in the migration procedure (bulk migration).

Non-event data distribution

Traditionally non-event data include:

- Construction and geometry/alignment data
- Hardware configuration data, calibration data
- Conditions data

These are stored in databases and distributed via Frontier and a set of squid caches deployed at sites. The squid caches are currently used by CMS to distribute also small files needed by the event generators.

The WLCG TEGs are now supporting the deployment of a network of squid caches to support file distribution via CVMFS. CMS will use CVMFS to distribute software and other small files like the glidein wrappers.

Authors

C.Grandi (INFN-Bologna); B. Bockelman (U.Nebraska); D. Bonacorsi (U.Bologna & INFN-Bologna); G. Donvito (INFN-Bari); D.Dykstra (FNAL); I. Fisk(FNAL); J.Hernandez (CIEMAT); S.Metson (U.Bristol); I.Sfiligoi (UCSD); S. Wakefield (I.C. London)