

Table of Contents

Proposal for using CVMFS for Conditions data for LHC Run 3.....	1
---	---

Proposal for using CVMFS for Conditions data for LHC Run 3

ATLAS has been prototyping a new Conditions database architecture based on a RESTful http interface and including a higher level interface on the database server side than the current low-level SQL used by ATLAS & CMS over Frontier. Here is some motivation for the project. Belle II has another architecture that uses CVMFS for the bulk of the data but has a direct-access database server (without a squid hierarchy) that all worker nodes connect to in order to find out where the data is in CVMFS. I don't believe that will scale to the levels that ATLAS & CMS need.

I propose to instead distribute conditions data fully over CVMFS. Given that CVMFS is already installed at most grid sites, distributing conditions data with the same mechanism would greatly reduce the operational load by eliminating specialized servers and reducing software maintenance needs. It would eliminate the need for Frontier servers, an oracle (or other) database, the frontier client, and CORAL. In addition, since CVMFS has client side caching, it should significantly reduce load on the squid infrastructure.

The basic idea is to map the CMS conditions data model with IOVs, tags, and global tags onto a filesystem hierarchy. CVMFS behind the scenes generates sqlite catalogs for subtrees of the filesystem. The publisher has complete control over where the subtrees are split up, by simply creating a '.cvmfscatalog' file or by listing wildcard expressions of directory paths in a '.cvmfsdirtab' file at the top of a repository tree. The key would be to arrange the filesystem hierarchy in such a way so that the catalogs do not get very large and so that there does not get to be continuous churn of large numbers of catalogs every time new data is added for IOVs. Payloads could continue to be in large blobs, and CVMFS stores every file as a hash of its contents so the payloads can just be referred to by well-known names.

Some work would need to be done on CVMFS, in particular to reduce publish delays for the CMS HLT. There has already been a lot of technical discussions with the CVMFS development team on doing that, and there are many things that can be done, especially in well-controlled installations like the HLT. When very low latency at small scales is needed, such as for creating global tags, the data can initially be stored on an ordinary filesystem. From there it can be published on CVMFS for scaling the number of clients.

The code that creates the filesystem layout and reads it back should be written as a generic library that can be shared between experiments.

-- DaveDykstra - 2016-04-25

This topic: [Frontier > ReplaceWithCVMFSProposal](#)

Topic revision: r1 - 2016-04-26 - DaveDykstra



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)