

Accessing Data Federations with CVMFS

Thursday, 13 October 2016 11:45 (15 minutes)

Data federations have become an increasingly common tool for large collaborations such as CMS and Atlas to efficiently distribute large data files. Unfortunately, these typically come with weak namespace semantics and a non-POSIX API. On the other hand, CVMFS has provided a POSIX-compliant read-only interface for use cases with a small working set size (such as software distribution). The metadata required for the CVMFS POSIX interface distributed through a caching hierarchy, allowing it to scale to the level of about a hundred thousand hosts. In this paper, we will describe our contributions to CVMFS that merges the data scalability of XRootD-based data federations (such as AAA) with metadata scalability and POSIX interface of CVMFS. We modified CVMFS so it can serve unmodified files without copying them to the repository server. CVMFS 2.2.0 is also able to redirect requests for data files to servers outside of the CVMFS content distribution network. Finally, we added the ability to manage authorization and authentication using security credentials such as X509 proxy certificates. We combined these modifications with the OSG's StashCache regional XRootD caching infrastructure to create a cached data distribution network. We will show performance metrics accessing the data federation through CVMFS compared to direct data federation access. Additionally, we will discuss the improved user experience of providing access to a data federation through a POSIX filesystem.

Tertiary Keyword (Optional)

Storage systems

Secondary Keyword (Optional)

Computing middleware

Primary Keyword (Mandatory)

Distributed data handling

Primary author: WEITZEL, Derek John (University of Nebraska (US))

Co-authors: BOCKELMAN, Brian Paul (University of Nebraska (US)); DYKSTRA, Dave (Fermi National Accelerator Lab. (US)); BLOMER, Jakob (CERN); MEUSEL, Rene (CERN)

Presenter: BOCKELMAN, Brian Paul (University of Nebraska (US))

Session Classification: Track 4: Data Handling

Track Classification: Track 4: Data Handling