

CMS Space Monitoring

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2014 J. Phys.: Conf. Ser. 513 042036

(<http://iopscience.iop.org/1742-6596/513/4/042036>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 131.225.23.169

This content was downloaded on 01/08/2014 at 20:14

Please note that [terms and conditions apply](#).

CMS Space Monitoring

N Ratnikova^{1†}, C-H Huang¹, A Sanchez-Hernandez², T Wildish³ and X Zhang⁴

¹Fermi National Accelerator Laboratory, Batavia, Illinois, USA

²Centro Invest. Estudios Avanz, Mexico City, Mexico

³Princeton University, Princeton, New Jersey, USA

⁴Institute of High Energy Physics, Beijing, People Republic of China

E-mail: [†]natasha@fnal.gov

Abstract.

During the first LHC run, CMS stored about one hundred petabytes of data. Storage accounting and monitoring help to meet the challenges of storage management, such as efficient space utilization, fair share between users and groups and resource planning. We present a newly developed CMS space monitoring system based on the storage metadata dumps produced at the sites. The information extracted from the storage dumps is aggregated and uploaded to a central database. A web based data service is provided to retrieve the information for a given time interval and a range of sites, so it can be further aggregated and presented in the desired format. The system has been designed based on the analysis of CMS monitoring requirements and experiences of the other LHC experiments. In this paper, we demonstrate how the existing software components of the CMS data placement system, PhEDEx, have been re-used, dramatically reducing the development effort.

1. Introduction

The CMS experiment computing [1] infrastructure spans a large number of geographically dispersed sites that provide both computational and data storage resources. Storage capacity at the sites varies from hundreds of terabytes to several petabytes, total data volume after the first several years of LHC collisions reaching one hundred petabytes.

The central data operations team [2] is responsible for efficient resource utilization. The operators monitor space usage and extrapolate it to estimate resource needs for all central computing tasks, including coordinated data production and processing, transfers to other sites for custodial storage and replication of popular data. They work with the sites to enforce reliable data delivery and to keep data and meta-data consistent at all sites and both on disk and on tape. Data transfers between the sites are managed by the CMS data placement and location system PhEDEx [3]. PhEDEx automates data operations tasks, executed by a set of agents running at the sites and communicating with the central Transfer Management Database (TMDB) to retrieve their workload and report back the results. TMDB is based on an Oracle database cluster hosted at CERN. PhEDEx also provides a 'Namespace framework' [4] to perform operations on the sites' local storage system via storage-technology specific plugins and a web data service [5] interface to retrieve information from TMDB in machine-readable formats for monitoring data transfers and operations.



PhEDEx knows about centrally managed data at sites. However it does not know about temporary production files or data produced by users. Some sites have their own storage space monitoring, including users and group data. Still, there is no system for monitoring all CMS data across all sites. The CMS space monitoring system has been designed to provide a global view of the distributed storage based on the sites local storage information.

2. Space monitoring project

The space monitoring project was initiated following the recommendations of the CMS monitoring task force [6]. The first prototype realized at the end of 2011 demonstrated a proof of concept for a global storage accounting and monitoring system based on ‘storage dumps’ [7]. A ‘storage dump’ is a file containing the metadata information about files, their sizes and checksums, that exist in the storage element. Typically they can be produced by reading the metadata database associated with the storage element, without the need to exhaustively search the storage element itself.

The second prototype, based on an Oracle database, kept the original architectural design as shown in Fig. 1, but the system was fully re-implemented using PhEDEx code base and components [4]. These components provide safe and efficient interfaces to the database and various types of storage, and common solutions to authentication, security, documentation, and system deployment.

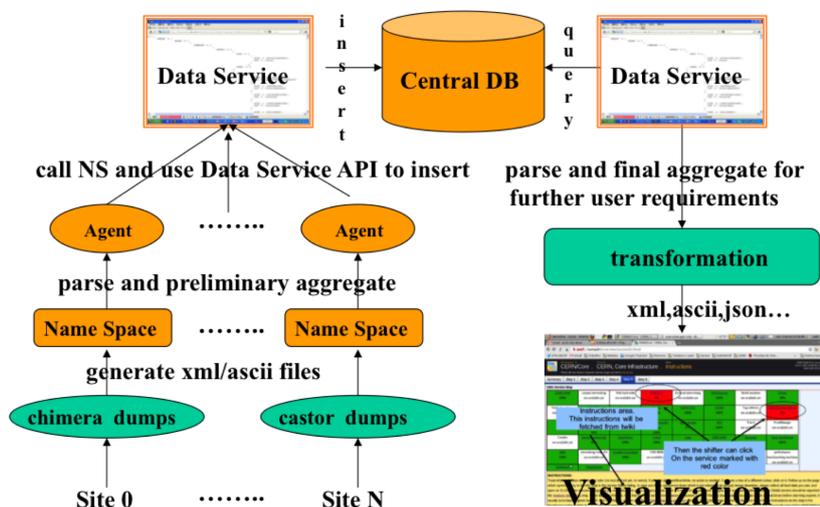


Figure 1. General architecture of CMS space monitoring system. ‘NS’ refers to the PhEDEx ‘NameSpace’ tools, a lightweight framework to encapsulate interaction with a storage element.

The second prototype has been released as part of the CMS web services, allowing a few pilot sites to test the new application. Testing revealed limitations due to several assumptions made in the prototype. In particular, we had assumed that files of a given ‘type’ (real data, monte carlo, user-files etc) would all be stored in a hierarchy with a single root per type, per site. This

turns out not to be true everywhere, some sites have more complex setups. The schema was enhanced and the APIs extended to resolve these limitations.

3. Components, interfaces, and information flow

The CMS space monitoring takes storage content information from the storage dumps produced at the sites. Format and requirements to storage dumps and tools recommended for producing them, are presented in [7], [8].

The diagram in Fig. 2 illustrates the information flow for one site.

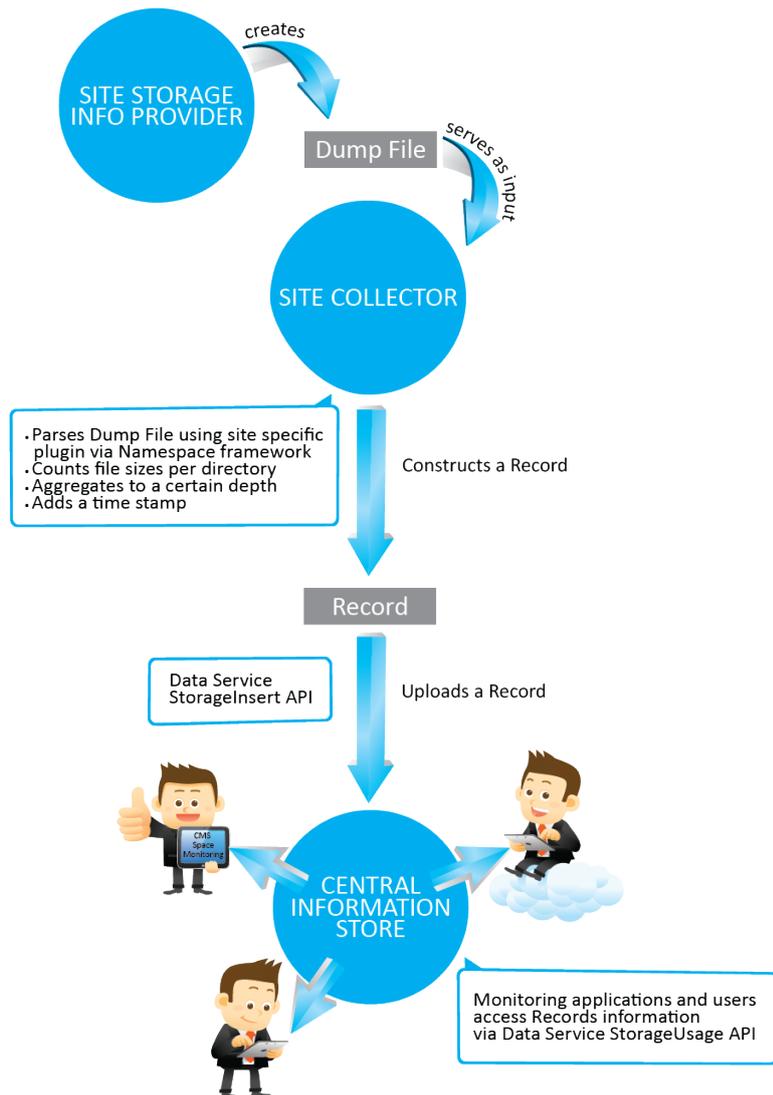


Figure 2.

Site Storage Info Provider is a site specific and storage specific service that produces storage dumps at regular time intervals.

Site Collector is a process running locally on the site, which checks whenever a new storage *Dump File* is available from the Site Information Provider. The Site Collector parses and aggregates information from the *Dump File* into a *Record*, and uploads the *Record* to a *Central Information Store*.

Each site provides its own storage information provider service responsible for producing storage dumps. For example, dCache [9] storage system can use either the chimera-dump [10] or pnfs-dump [11] utility executed at regular time intervals. The resulting dump file is then placed in a shared area accessible by the site collector.

The site collector is a process running locally on the site. It checks if a new storage dump file is available, and passes it as an input to the storage insert utility, which serves as an interface to the central information store.

The storage insert utility parses the dump, counting file sizes per directory, aggregates the sizes to a certain level of depth defined in the sites' configuration, constructs a compact record,

including the site name and the time stamp, and uploads that record to the central information store.

The storage insert utility is provided centrally to the sites as part of the space monitoring package. It comes with a set of plugins for handling different formats of the storage dump file.

Interaction with the information store, both to store and to retrieve information, is realized via web data service APIs. The storage information provider and site collector do not communicate directly.

4. Data organization

The CMS data management system [12] maintains central data catalogs representing the experiment's knowledge of the storage content at each site. The file information in the central data catalog is stored in terms of the Logical File Names, LFNs. The data are named, grouped together by types and organized in a tree-like LFN namespace structure [13] according to conventions developed by the CMS data and workload management policy group. Examples of top level namespaces, or data types, include raw, prompt reconstruction, heavy-ion data, MC simulated and generators data samples, unmerged production files, load test files, users data and other data types.

The real data on storage elements are addressed via Physical File Names, the PFNs, identified via a Trivial File Catalog, TFC, which provides rules to convert LFNs to PFNs. Each site maintains its own TFC. In the process of evolution of storage technology and infrastructure, the data at the site may be migrated to a new storage system with different physical names. If the TFC is updated properly, such transitions are transparent for the computing operations [14]. In some cases such changes may lead to the occurrence of blind data, i.e. data that are no longer accessible via the updated TFC, while they still occupy storage space at the site. To be able to deal with such situations, the CMS space monitoring system keeps track of full PFNs, i.e. direct paths to the data on the site local file system, as provided in the storage dumps.

In contrast to the central data catalogs, the space monitoring system does not keep information about individual files. It only stores the aggregated space usage for all files that reside under a certain level of depth in the directory structure, to reduce the database storage requirements. The schema allows to store records with varying level of detail. The aggregation depth may vary depending on data type, defined by a special set of tags. For example, sites may have privacy policies that do not allow to expose the contents of the users data areas. In practice, aggregation levels are defined by the local site specific configuration, standard tags based on CMS Namespace conventions, and general application defaults, in this order of precedence.

In addition to providing a mechanism to define the initial aggregation level, tags allow to query, aggregate and compare data by types across the sites.

5. Central infrastructure, support and operations

While Space Monitoring and PhEDEx use the same code base, they do not share the infrastructure: they use separate databases and data service instances.

The Space Monitoring code is maintained under the PhEDEx umbrella for practical reasons, but is packaged and distributed separately, it is not coupled with PhEDEx release cycles.

The Central Information Store is an Oracle database, supported by CERN IT division. Schema initialization scripts are included in the application code. Sites need to be registered as an additional step. Also the tags for supported data types need to be initialized by an authorized administrator.

Site access to Central Information Store is realized via web data service APIs based on Grid certificates [15] authentication.

6. The deployment procedure

Our aim is to make it easy for a site to join the CMS space monitoring system. These are roughly the deployment steps for the site admin:

- (i) Set up service for producing storage dumps using recommended tools or your site's preferred solution.
- (ii) Install space monitoring package on the system where storage dumps can be accessed.
- (iii) Make sure site is registered in the central information store.
- (iv) Configure site collector to use one of the provided parsers, or write your own if needed.
- (v) Provide mapping between data types and storage locations for the configuration; part of this can be done automatically using the sites TFC.
- (vi) Adjust levels of aggregation as necessary.
- (vii) Start an agent or your own site collector scheduler to collect and feed the information to the central information store.

Most of these steps rely on procedures and tools already familiar to CMS site administrators from their experience with PhEDEx deployment, site registration, management of site agents and configuration. Most of the sites are already producing full storage dumps for the data consistency checking purpose.

7. Summary

The CMS space monitoring system based on local storage dumps collects information about all data on storage, not only official data - to give a realistic view of local storage usage at the participating sites. Unlike other central data catalogs, it stores physical locations of the data.

We aggregate information to a certain level of depth, and provide a dynamic view of space usage at the sites, not just current values. Data tags have been introduced to allow for a customizable aggregation of the space usage information for different data types.

Reuse of the generalized solutions and of the code base from PhEDEx helped to dramatically reduce our development effort, while gaining all the benefits of one of CMS most mature computing projects. From the site admin perspective, the existing expertise with the PhEDEx family of tools should help the sites to join and operate the system more smoothly.

References

- [1] Bonacorsi D 2007 "The CMS computing model" *Nucl. Phys. B (Proc. Suppl.)* **172** 53-56
- [2] Kaselis R *et al.*, 2012 "CMS Data transfer operations after the first years of LHC collisions," *J. Phys.: Conf. Ser.* **396**, 042033 (2012).
- [3] Egeland R, Metson S and Wildish T 2008 "Data transfer infrastructure for CMS data taking" *XII Advanced Computing and Analysis Techniques in Physics Research* (Erice, Italy: Proceedings of Science)
- [4] Sanchez-Hernandez A, Egeland R, Huang C-H, Ratnikova N, Magini N and Wildish T 2012 "From toolkit to framework: The past and future evolution of PhEDEx," *J. Phys.: Conf. Ser.* **396**, 032118 (2012).
- [5] Egeland R, C-H Huang and Wildish T, 2010 "PhEDEx data service," *J. Phys.: Conf. Ser.* **219**, 062010 (2010).
- [6] Bauerdick L A T and Sciaba A 2012 "Towards a global monitoring system for CMS computing operations," *J. Phys. Conf. Ser.* **396**, 032099 (2012).
- [7] Huang C-H, Lanciotti E, Magini N, Ratnikova N, Sanchez-Hernandez A, Serfon C, Wildish T and Zhang X 2012 "Data storage accounting and verification at LHC experiments," *J. Phys. Conf. Ser.* **396**, 032090 (2012).
- [8] Lanciotti E 2011 "Storage elements dumps and consistency checks versus file catalogues", <https://twiki.cern.ch/twiki/bin/view/LCG/ConsistencyChecksSEsDumps>
- [9] dCache, <http://www.dcache.org>
- [10] chimera-dump: a python script to get information from the chimera postgresql DB, <http://trac.dcache.org/wiki/contributed/chimeraDump>

- [11] PNFS utilities, <http://www.dcache.org/downloads/pnfs>
- [12] Giffels M, Guo Y Kuznetsov V, Magini N and Wildish T 2013 The CMS Data Management System, *these proceedings* CHEP 2013
- [13] CMS LFN Namespace, internal documentation, <https://twiki.cern.ch/twiki/bin/viewauth/CMS/DMWMPG.Namespace>
- [14] Gutsche O *et al.* 2013 CMS Computing Operations During Run1, *these proceedings* CHEP 2013
- [15] Globus toolkit online documentation: Overview of the Grid Security Infrastructure, <http://www.globus.org/security/overview.html>