Fermilab

Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

CMS Space accounting project and what we can do together

Natalia Ratnikova

WLCG pre-GDB meeting

12 April 2015

Outline

- CMS data management and operations
- Data storage at CMS sites
- CMS Space Monitoring project
  - Requirements
  - Components
  - Visualization options
- Summary of common areas

CMS data management and operations

- CMS data live in a **global name space**, addressed by a logical file name (LFN), e.g.:

  /store/data, /store/mc, /store/user, /store/group, …
- Data are accessed by physical file names (PFNs) according to the LFN to PFN translation rules specified in the trivial file catalogs provided by the sites
- Space monitoring allows to track the space occupied by each level under /store across the sites.
- CMS central Transfer Management Database keeps track of data maintained by PhEDEx.
- Information on other files, users data, temporary production and test data, is only available from the direct storage dumps.

Data storage at CMS sites

- Total over 100 sites
- Only Tier-1 and Tier-2 sites pledge storage space
- Storage technologies: Castor, dCache, DPM, EOS, Hadoop, LStore, Lustre, StoRM.
- CMS Tier 1 and 2 storage space requirements*

Year	2013	2014	2015	2016
Tier 1 Disk	26,000	26,000	26,000	33,000
Tier 1 Tape	50,000	55,000	74,000	100,000
Tier 2 Disk	26,000	27,000	29,000	38,000

•Increased pileup, higher HLT rate, data parking and scouting
•Volume will grow proportionally to LHC life time
•Phase 2 detector upgrade studies
➞CMS expects severe resource constraints

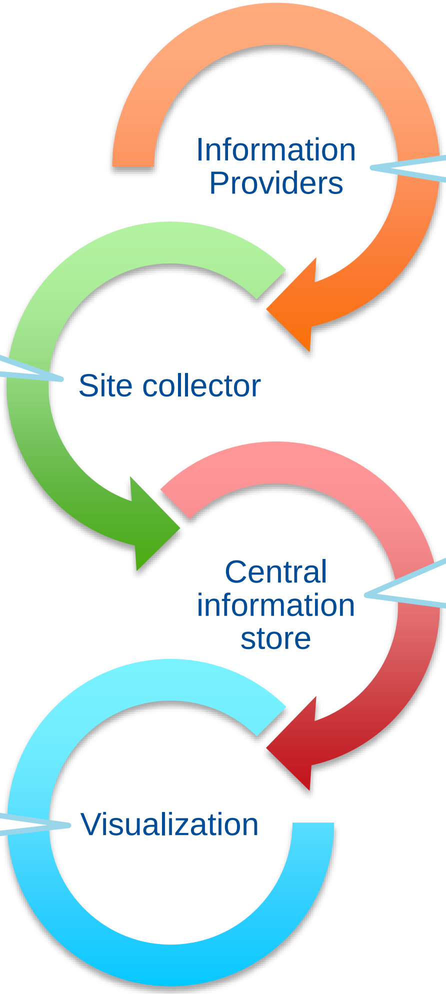
CMS Space Monitoring requirements

- Collect storage space usage information at the sites
  - including data not managed in the central file catalog, such as user data, temp areas, legacy data, orphaned files etc
- Aggregate space usage information
  - must reflect the CMS data organization
  - scalability and privacy considerations
- Update information at weekly intervals
  - unlike PhEDEx, which uses a white board approach, keep old records to monitor storage usage evolution over time

Additional desired features

- Collect information about available space
  - storage technology dependent,
  - usually includes (file system) overhead
- Keep track of other metrics
  - number of files and directories
  - file access modification time (overlap with popularity?)
- Legacy data
- Aggregation of the file sizes on the level of the storage bookkeeping database (e.g. chimera)
  - possible and preferred for some storage technologies
  - replaces part of client functionality

CMS Space Monitoring system overview



Tools for producing dumps of the site local storage catalogs

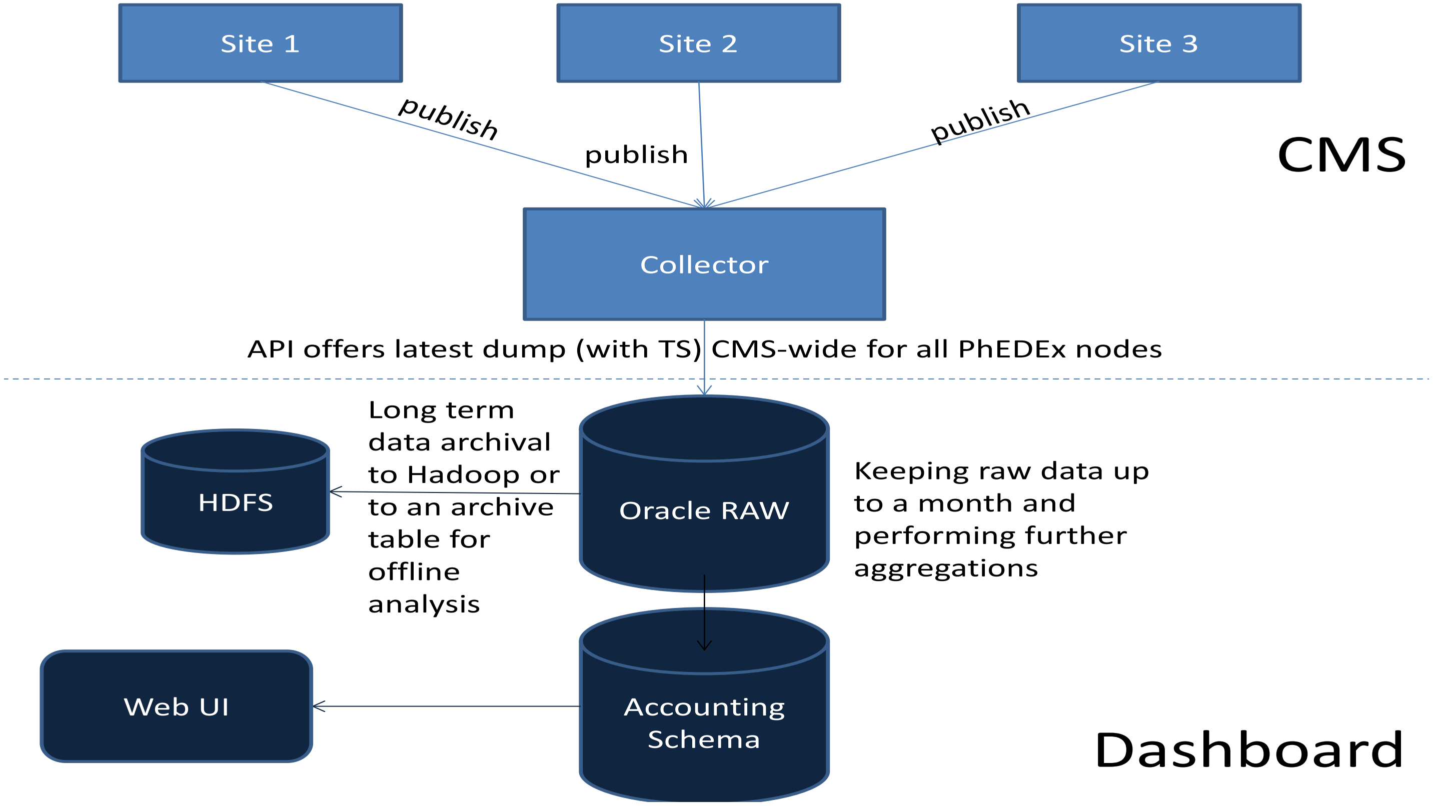
Client side tool
Parses and aggregates storage dump and uploads to central information store

DMWMMON database and interfaces
Data service APIs to insert or retrieve space usage records

Interface to retrieve DMWMMON info and presentation layer in CMS dashboard

Visualization

Proposal for visualization in CMS Dashboard based on ATLAS implementation


Site 1
Site 2
Site 3
publish
publish
publish
CMS
Collector
API offers latest dump (with TS) CMS-wide for all PhEDEx nodes
HDFS
Long term data archival to Hadoop or to an archive table for offline analysis
Oracle RAW
Keeping raw data up to a month and performing further aggregations
Web UI
Accounting Schema
Dashboard

We are currently also looking in Elasticsearch+Kibana based implementation

Summary

CMS specific	WLCG common
TFC (site configuration)	Storage technologies
Data storage namespace	Storage dump formats
Authenticated upload	Middleware software infrastructure
Monitoring configuration	Visualization infrastructure

- CMS SpaceMon will clearly benefit from WLCG common infrastructure and tools for storage information providers and visualization
- CMS specifics tasks, such as:
  - translating local storage areas to a global logical namespace
  - defining and maintaining aggregation parameters
  - site specific authentication and roles
  - monitoring configuration

  need to be done on the experiment side.

Backup slides

CMS Space Monitoring: data service

API name	Functionality	Options
auth	show authentication state and abilities	ability require_cert require_passwd
bounce	simple data service debugging tool	* die
dumpspacequery	return unprocessed sql results	none
getlastrecord	show latest record for a node	node
nodes	dump a list of nodes	node, noempty
storageinsert	insert aggregated node storage information	node timestamp dirinfo
storageusage	query storage info	node level rootdir time_since, time_until

CMS Space Monitoring: client implementation

Class (*type*)	Description and basic functionality
Record *(container)*	- holds aggregated node space usage info
StorageDump *(base class container)*	- machine representation of the storage dump information
Format::XML *(algorithm)*	- rules for parsing dump in XML format
Format::TXT *(algorithm)*	- rules for parsing dump in TXT format
RecordIO *(algorithm)*	- reads/writes Record from/to file - uploads/downloads Record to/from central database via a data service - closure test : write record to a file, read it back and compare
Aggregate *(algorithm)*	- converts StorageDump object into Record object
NamespaceConfig *(algorithm)*	- reads rules from the configuration file - converts rules into a Namespace tree - does LFN -> PFN conversion via local TFC file or database - provides aggregation parameters to the Aggregate algorithm
UserAgent.pm *(wrapper)*	- SpaceMon specific wrapper around LWP::UserAgent module that implements https interface to the data service