ATLAS DQ2 Deletion Service

While deletion might seem a typical operation on the first sight, for a complex distributed system like ATLAS DDM, it is far from trivial.

ATLAS DQ2 deletion service has been built as a part of ATLAS DDM system, for serving deletion requests across more than 100 ATLAS sites. It is a distributed service which interacts with GRID middleware and the DDM catalogs.

ATLAS DQ2 Deletion Service. System architecture.

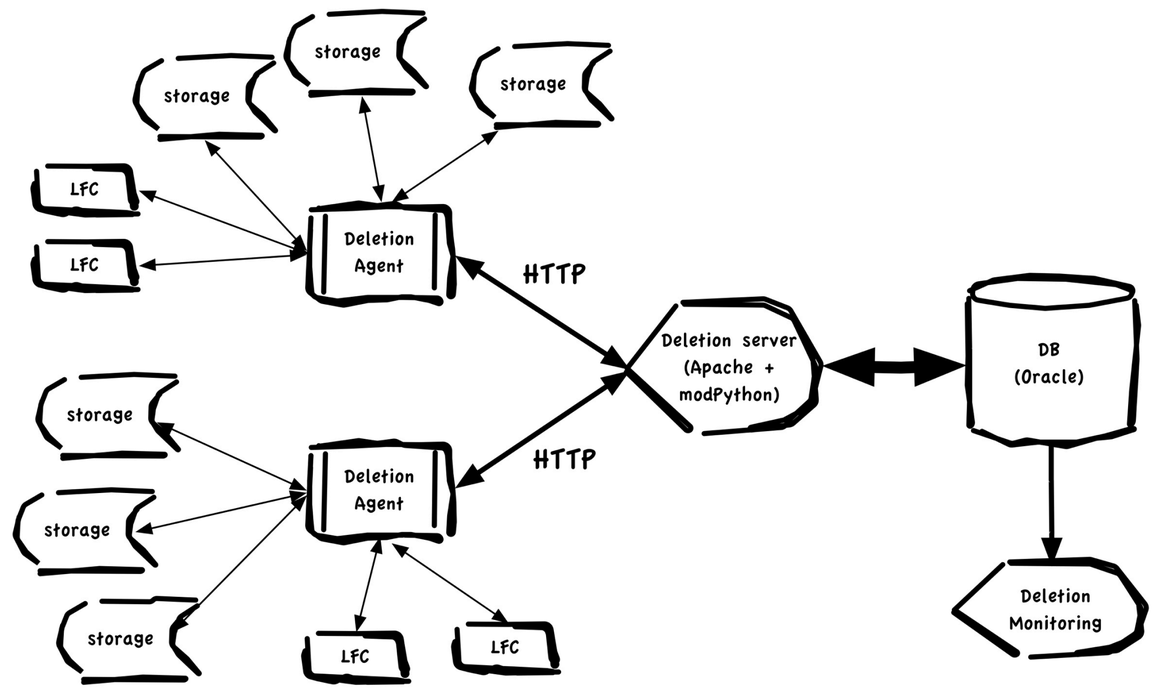
The Deletion service is divided into the following components:

__Server:__ collects deletion requests, stores detailed information about datasets, their contents (files) and the ongoing state of deletion requests.

__Client:__ provides communication between the server and other components of the service.

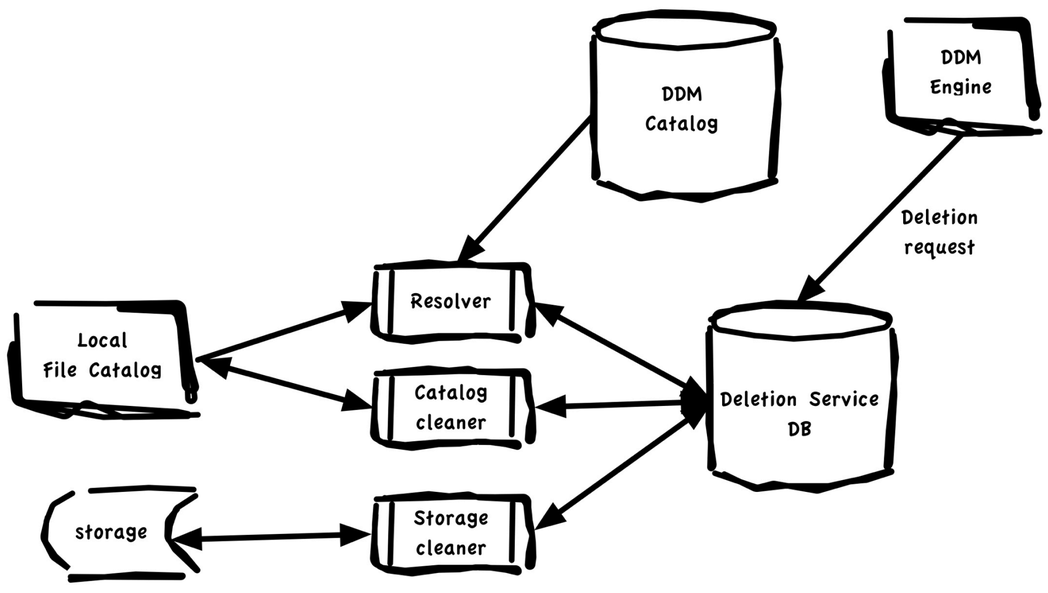
__Deletion Agent:__ performs actual deletion through intercommunication with the LFC servers and the storage system.

__Monitoring:__ displays the deletion status and errors.



Deletion agent

The workflow of the Deletion Agent: resolve list of files for deletion on site, unregister files from LFC catalog, delete files from mass storage systems. Deletion agent work s as three independent (parallel) process: resolver, catalog cleaner, storage cleaner.



__Resolver:__ resolves content of dataset (list of files) for deletion, store this content in database for future processing.

__Catalog cleaner:__ unregisters files on file catalogs(LFC), store results of operation with corresponded state in Deletion Service DB. In case process was unsuccessful operation will repeat.

__Storage cleaner:__ deletes files on mass storage systems, store results of operation in Deletion Service DB. Depending on the deletion status, files are set in the states: deleted or to repeat deletion again.

When all files in dataset marked as deleted on storage and deleted from LFC – this dataset is marked as successfully removed.

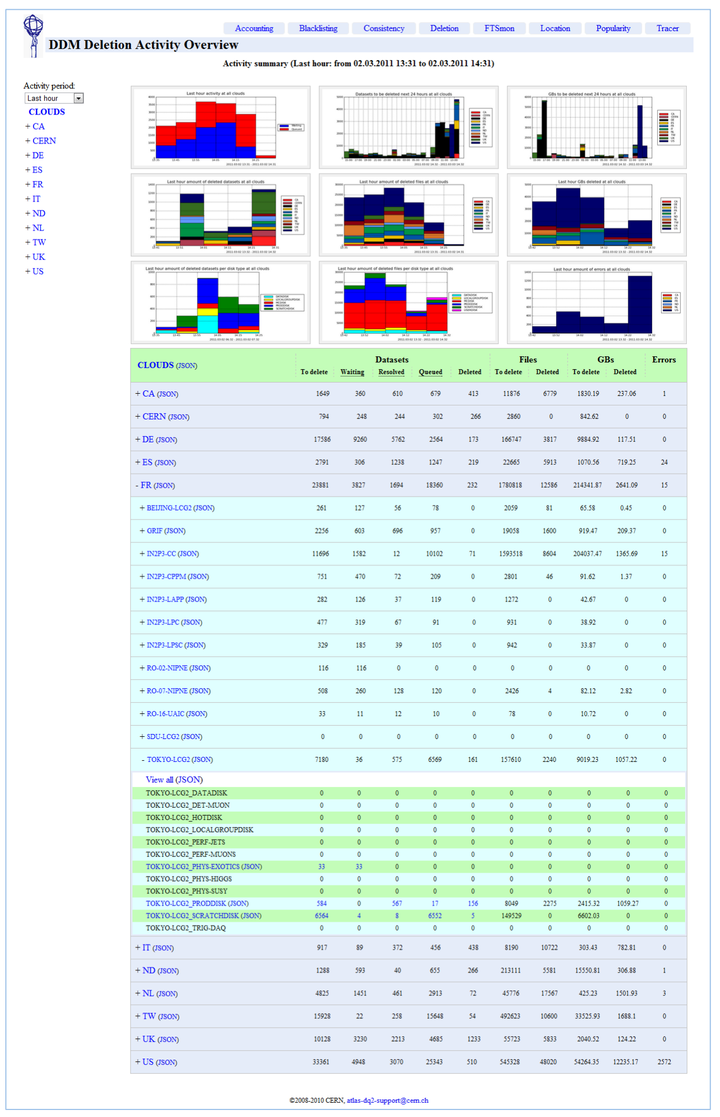
Features

__Productivity:__ To achieve performance, Deletion Agent is designed as multithread application. Each site served own copy of resolver, catalog and storage cleaner. For improving interaction with components, a bulk operation is implemented, and size of chunks configurable for sites and endpoints .

__Load throttling:__ Since Deletion service is highly aggressive for all the affected services, (LFC servers, storage systems, DB backend) the configurable delays are implemented as well. It give possibility to determine different deletion strategy for different sites.

__'Grace period':__ For safety reasons and to have some insurance in case of the human mistake, a grace period has been implemented in such a way that deletion requests in grace period can be easy canceled.

Deletion Monitoring



Deletion monitoring is a web application based on the Django framework, which provides live graphical and statistical reports about deletion process at ATLAS sites.

The information is available at the cloud/site/endpoint levels. It allows to select statistics at different periods.

In addition to graphical reports, monitoring generates table with info about waiting/resolved/ queued/deleted datasets, amount of files deleted, GBs deleted and amount of errors. Table is expandable. There are dataset and error browsers. The information is generated via jQuery AJAX calls and uses BBQ plug-in to maintain history and bookmarks.

http://bourricot.cern.ch/dq2/deletion/

Statistics

Deletion Service serves more than __100__ of sites with more than __700__ endpoints. In usual operation it deletes __2-2,5M__ of files per day which correspond to __250 - 300 Tb__ per day. During deletion campaigns, when deletion was carried out on most sites, deletion rate achieved more than 6M of files per day, reaching up to 300k files per hour.