Abstract. With the continuously increasing volume of data produced by ATLAS and stored on the WLCG sites, the probability of data corruption or data losses, due to software and hardware failures is increasing. In order to ensure the consistency of all data produced by ATLAS a Consistency Service has been developed as part of the DQ2 Distributed Data Management system. This service is fed by the different ATLAS tools, i.e. the analysis tools, production tools, DQ2 site services or by site administrators that report corrupted or lost files. It automatically corrects the errors reported and informs the users in case of irrecoverable file loss.

1. Introduction

Data losses and data corruption is an important issue on a distributed environment as the grid. The more the volume of data stored on disk servers is increasing, the higher the probability of data corruption or data losses. In the last months about one big loss is reported each month on the sites of WLCG [1] supporting ATLAS, which usually means $O(10^4)$ or even more files that are lost. These losses are very frequently due to hardware failures, with disks that cannot be rebuilt despite they are used in RAID mode. Till recently, when bad files were reported, all operations to fix them were done manually and were putting a high load on the sites and on the ATLAS Distributed Data Management (DDM) operation team. More automation was needed. Moreover it was difficult to keep track of all incidents and of all operations done for the recovery. The Consistency Service was developed to address these issues. After a detailed description of the service in chapter 2, some statistics regarding the files reported to it will be presented in chapter 3.

2. The Consistency Service

2.1. Overview

The consistency service collects all bad files (i.e. file losses, data corruption) that can be reported either by ATLAS tools, or people such as site admins. Once bad files have been reported, a complete recovery procedure is run that consists in deleting them from the catalogues (Storage namespace, LFC, DQ2 catalogues), identifying the datasets affected, retransferring them if possible or notifying the owner of the files/datasets if some of them are definitively lost. The service also ensures book-keeping: the list of files that were reported is kept as well as all actions done to recover them (typically $\sim 6-7$ actions per file). The technical implementation consists in:

- A server: To collect bad files (lost or corrupted) reported via a client.
A client: To report bad files.
An agent: To correct the bad files reported.
A monitoring system: To monitor the evolution of the recovery/provide statistics, etc.

As the rest of the DQ2 [2] system, the service is written in python. Its different parts are detailed hereafter.

2.2. Server/client
The Server is based on an Apache server that interacts with an Oracle backend. The client sends requests to this server. Both POST and GET requests are available to report suspicious/bad files and get information about the files declared. Since files that are declared bad will trigger some recovery procedure a protection is put at the server level so that only a restricted amount of people (based on VOMS attribute) are able to post a list of bad files. To report a file corruption/loss, only a list of SURLs and a 64 characters field called "reason" detailing the reason of the problem (e.g. pool failure, file corrupted, etc.) need to be provided. For each file declaration, a timestamp is also automatically recorded as well as the DN of the submitter and the IP of the machine where the client was run. The server supports bulk submission of files, which allow to declare 10 000s files in only a few seconds.

At the database level, the tables are designed to allow scalability in the long term (use of partitioning, indexes, etc.).

2.3. Agent
Figure 1 shows the functional schema of the consistency agent. This one obtains from the server the list of bad files (SURLs) to treat. The recovery procedure includes many steps. The agent first queries TierOfATLAS\(^1\) to identify the sites where the bad files are located and the LFC in which they are recorded. A GUID is then obtained from this LFC for each SURL. With the GUID, a reverse look-up is done on the DQ2 Central Catalogues to identify the list of datasets that contain the file. The next step is to delete the bad files from the Storage Element and from the LFC. This is done by generating a "trash" dataset. This dataset contains all the files that need to be deleted, and is submitted to the ATLAS DDM deletion service [3]. Once the files are physically and logically deleted, the information of the DQ2 location catalogue is refreshed. At this point, the consistency between Storage Element and the catalogues is restored. The end of the recovery consists in checking if the file is still available in another site. This is done by querying all LFCs used by ATLAS. In the case the file is available in another site, the service replicates it. If the file cannot be found anywhere, the service removes it from the dataset definition and notifies the dataset owner.

2.4. Monitoring
The monitoring [4], which is developed under the Django framework [5], has two functions. The first one is to monitor the work of the agent, to see at which step of the recovery it is. The second function is to access informations of bad files previously reported. In particular, a search interface is available that allows to get the list of incidents\(^2\) filtered on different criteria such as time, site. For each of them, a summary page is available that provides in particular the reason, the site, identity (DN, IP) of the person that declared, final status of the recovery, plots of recovery time. For each file, the list of actions done to recover it is also available.

---

1 TierOfATLAS is a dictionary where static information used by ATLAS tools about sites/clouds in ATLAS is recorded. This will be replaced in a near future by AGIS (ATLAS Grid Information System).
2 What is called incident is a triplet (date, reason, endpoint) identifying a list of bad files.
Figure 1. Functional schema of the consistency agent. The different steps of the recovery are shown from 0 to 6. At step 6, if files are recoverable from another site, DQ2 will retrieve them (7a); if they are not, files will be removed from dataset definition (7b) and users notified (8b). For all steps retry mechanisms are implemented in case of some of the services (e.g. LFC, DQ2 catalogues) being unavailable.

3. Statistics
The service is running since beginning of February 2010. Since that date, reports of bad files are done almost every day. Beginning December 2010, the number of files declared to the system is $\sim 530\,000$ corresponding to 261 incidents affecting 38 physical sites and 130 endpoints (each site publishes different endpoints). Figure 2 shows the number of files reported for each incident. The mean number of files per incident is 2,000 with an RMS of 6,000. 15 incidents with more than 10,000 files were reported. More than 95% of the errors are due to hardware problems (disk crash, controller error, etc.). The recovery time for these incidents is in general fast but depends on the size of the files, the bandwidth to the site, the activity of other transfers going-on. Figure 3 shows the number of files recovered versus time for one of the big losses (44,000 files). In this case, the complete recovery occurred in less than 2 days which is the case for the majority of incidents.

It is also interesting to have a look at the final states of the consistency agent. The files can be classified in 5 different categories depending on the final state. The first two states correspond to files that cannot be found in one of the 2 catalogues used by DDM (LFC and DQ2 catalogue) and therefore no recovery procedure can be initiated. An other state corresponds to files that were recovered from somewhere else. The last two states correspond to files that are definitively lost, and files that are in datasets that disappeared between the beginning and the end of the recovery. Figure 4 shows the evolution of the number of files of the different states versus time.
Number of files per incidents

Entries: 261
Mean: 2046
RMS: 5945

Number of files declared

Figure 2. Number of files reported by incidents. Most of the incidents reported consist only of a few files. But incidents with more than 10,000 are not so rare (15 in 9 months).

Recovery time for a loss of 44,000 files. All the files were recovered in less than 2 days.

Figure 3. Recovery time for a loss of 44,000 files. All the files were recovered in less than 2 days.

It can be noticed that the number of files treated is increasing with time due to the fact that more and more sites use the service. 54% of the files declared and treated by the service could be recovered. The number of files definitively lost is quite high: 24%, which means 130,000 files. The rest is mainly due to files not registered in one of the catalogues (20%). These unregistered files are coming mainly from failed transfer attempts and should be identified and removed regularly by the sites [6].

Figure 4. Evolution of the number of files declared to the consistency service split by the different final states of the consistency agent versus time. Red (resp. green) indicates files that are not in LFC (resp. in DQ2 catalogs). Blue represents files that were successfully recovered. In light blue are the files belonging to datasets that disappeared between the beginning and the end of the recovery. Yellow are files definitively lost. The global increase of the number of files correspond to the increasing number of sites using the service.
4. Conclusions
The consistency service allows to efficiently fix bad files reported by ATLAS tools and by users. It alleviates the load on operation teams by taking in charge of the whole recovery procedure. It also provides book-keeping that is needed to analyse the evolution of the incidents over the time. The analysis of these incidents reported since the service has been put into production shows that the loss of files is a very frequent phenomenon. Although they are not the most frequent, the big losses (more than 10 000 files) are not rare events. The redundancy (i.e. multiples copies of each dataset) is crucial to ensure that no files will be lost.

References
[1] The Worldwide LHC Computing Grid (WLCG) web portal: http://lcg.web.cern.ch/LCG
[2] M. Lassnig, V. Garonne, M. Branco and A. Molfetas, “Dynamic And Adaptive Data-Management In Atlas,” J. Phys. Conf. Ser. 219, 062054 (2010).
[3] V. Garonne et al., Grid data deletion service, CHEP 2010
[4] http://bourricot.cern.ch/dq2/consistency/
[5] http://www.djangoproject.com/
[6] C. Serfon, “Data management tools and operational procedures in ATLAS: Example of the German cloud,” J. Phys. Conf. Ser. 219, 042053 (2010).