2018

MONITORING OF SWT2 DATA CLUSTERS FOR THE ATLAS EXPERIMENT

Antara Ray

MONITORING OF SWT2 DATA CLUSTERS FOR THE ATLAS EXPERIMENT

by

ANTARA RAY

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2018

Abstract

MONITORING OF DATA CLUSTERS FOR THE ATLAS EXPERIMENT

Antara Ray, MS

The University of Texas at Arlington, 2018

Supervising Professor: David Levine

Monitoring of the South West Tier 2 RSEs is done by CERN with the help of Rucio. The challenge faced by the team monitoring the servers at the University of Texas site was that the monitoring data is pictorially represented and provided to them in GIF format.

In this work we focus on creating an interactive site that will not only monitor the data at the local RSEs but also create a platform to analyze the data storage systems. In turn it will also create alerts whenever during monitoring an aberration from expected behavior is noticed either in the storage at RSEs or transfer and deletion of data from the RSEs.

To my husband

Acknowledgements

I would like to express my sincere gratitude to Mr. David Levine my thesis supervisor who has guided me in the course of this thesis. His continuous support and patience with me has helped me in contributing my research work in this thesis.

I would also like to thank Mr. Patrick McGuigan, my mentor for his help, support and guidance throughout my research for the ATLAS experiment and the monitoring system. It was his support, continuous feedback and invaluable suggestions that have helped me at each stage during this research work.

I would like to acknowledge and thank my committee members, Dr. Ramez A Elmasri and Dr. Elizabeth D Diaz for their co-operation.

October 30, 2018

Table of Contents

List of Illustration

List of Tables

Chapter 1

Introduction

The Atlas experiment at the CERN Large Hadron Collider, the world's largest and most powerful particle collider and the largest machine in the world [13], generates humongous volumes of data that flows down to different computing clusters across the globe for analysis. One such data cluster is the South West Tier 2 cluster. The South West Tier 2 cluster comprise of the computing clusters present at the University of Oklahoma , University of Texas Arlington and Langston University.

The Atlas South West Tier 2 computing clusters present at UT Arlington campuses at Arlington and Fort Worth are maintained by the people of UT Arlington. Whenever if there is any kind of issue or surge in the data centers the people responsible for the centre here at UT Arlington are notified by CERN which monitor all its clusters using Rucio (a Distributed Data Management system used by CERN). Once notified the people at UT Arlington start investigating the root cause of the issue.

UT Arlington has access to all the data that Rucio uses to monitor the clusters here at UT Arlington campus. Thus our goal was to build a scalable localized monitoring system that would track the health of the clusters under UT Arlington and the clusters at University of Oklahoma and Langston University where it is required. Our system will not only monitor health of the storage servers it will also generate alerts when it notices a surge in the storage units based on predetermined threshold values.

Chapter 2

Background

Data center monitoring encompasses maintaining regular surveillance over the entire data center infrastructure. Not just the data centre but the channels through which data flows to and from these centers also need to be constantly monitored for a fault free system.

Worldwide LHC Computing Grid

The Worldwide LHC Computing Grid (WLCG) is a global collaboration of computer centers. It was launched in 2002 to provide a resource to store, distribute and analyze the 15 petabytes of data generated every year by the Large Hadron Collider (LHC)[5].

When work began on the design of a computing system for LHC data analysis it was discovered that the required computing power was more than the fund available to CERN. Most of the laboratories and universities collaborating on the LHC had access to national or regional computing facilities. These computing facilities were integrated into a single LHC computing service – the Grid – in 2002. It now links thousands of computers and storage systems in over 170 centers across 41 countries. These computer centers are arranged in "Tiers", and together serve a community of over 8000 physicists with near real-time access to LHC data. The Grid gives users the power to process, analyze and store LHC data. The grid-based infrastructure is an effective solution to the data-analysis challenge of the LHC offering many advantages-multiple copies of data can be kept at different sites ensuring access for all scientists independent of geographical location, no single point of failure, round the clock monitoring of data.[5]

Tier 0 is the CERN Data Centre and all the raw data from the LHC passes through this central hub. CERN is responsible for this data and does the work of reconstructing the raw data into meaningful information. The data from Tier 0 – some raw and some reconstructed flows to Tier 1 data centers.

Tier 1 consists of 13 computer centers which stores the LHC data. Tier 1 data centers support the Tier 0 center round the clock. Apart from storing the data it receives from Tier 0 it also performs large-scale processing of data and storage of output. It then distributes the data to Tier 2 datacenters and stores the output data that the Tier 2s produce. Optical-fiber connect CERN to each of the 13 major Tier 1 centers around the world. This network is called the LHC Optical Private Network (LHCOPN)[12].

Country	Tier 1 Grid site
Canada	TRIUMF
Germany	KIT
Spain	PIC
France	IN2P3
Italy	INFN
Nordic countries	Nordic Datagrid Facility
Netherlands	NIKHEF / SARA
Republic of Korea	GSDC at KISTI
Russian Federation	RRC-KI and JINR
Taipei	ASGC
United Kingdom	GridPP
US	Fermilab-CMS
US	BNL ATLAS

Table 2-1 Tier 1 Grid Sites

Tier 2 datacenters are typically universities and other scientific institutes that can store sufficient data and provide adequate computing power for specific analysis tasks. Additionally, they are also involved in production and reconstruction of simulated events. There are around 155 Tier 2 sites around the world.

The physics department at the University of Texas at Arlington(UTA) has collaborated with the University of Oklahoma (OU) and Langston University (LU) to form the Southwest Tier 2 distributed computing facility to support the ATLAS experiment.

A little detail about the hardware:

• SWT2– UTA Tier 2 Hardware

• <u>SWT2 CPB</u>

◦ 465Nodes(9848Slots)–2-3GB RAM per Slot

◦ 45 Support Nodes(15head,30storage)

◦ 5500TB of usable xrootd storage (MD1000/MD1200/MD3X60)

◦ ROCKS6.1(SL6.3),PBS/Torque,OSG3.2

• UTA SWT2
◦ 110Nodes(2360Slots)–2GB RAM per slot
◦ 15 Support Nodes(9 head,6 storage)
◦ 440 TB of usable xrootd storage (MD1000/MD1200)
◦ ROCKS 6.1(SL 6.3), PBS/Torque, OSG 3.2

SWT2 – Lucille Hardware

• 30 Nodes(960 Slots)–4GB RAM per slot
• 5 Support Nodes(2 head,3 storage)
• 110 TB of usable XFS storage(MD 1200/MD 3200)
• ROCKS 6.1(CentOS 6.9), HTCondor, OSG 3.3
• Cluster also has some GPU nodes that have been used for other OSG testing

SWT2- OU OSCER Hardwar

• 41 Nodes(1800 Slots)–2-4 GB RAM per slot
• 8 Support Nodes(2 head,6 storage)
• 500 TB of usable xrootd storage(5 T630s with 16 8 TB drives)
• SALT(CentOS 7.3), SLURM 17.02, OSG3.3
• Part of generic OSCER HPC cluster
• Rest of OSCER Schooner Hardware
  ◦ 510 Nodes(about 10k Slots)–2-4 GB RAM per Slot
  ◦ Opportunistically available for ATLAS production [19]

[Updated in accordance to data from Patrick McGuigan ,FACULTY RESEARCH ASSOCIATE, Physics,UTA]

Individual scientists can access the Grid through local (or Tier 3) computing resources.

Atlas Experiment:

ATLAS (**A Toroidal LHC ApparatuS**) is one of the particle collision detectors used at the Large Hadron Collider (LHC) operated by the European Organization for Nuclear Research (CERN). ATLAS explores a range of physics topics, with the primary focus of improving our understanding of the fundamental constituents of matter. ATLAS was one of the two LHC experiments involved in the discovery of the Higgs boson in July 2012. It was also designed to search for evidence of theories of particle physics beyond the Standard Model. The experiment is a collaboration involving roughly 3,000 physicists from over 175 institutions in 38 countries.[4]

The ATLAS experiment is exploring the effects of collisions of protons of extraordinarily high energy. It is trying to learn about the basic forces that have shaped our universe since the beginning of time and its future. Among the possible unknowns are extra dimensions of space, unification of fundamental forces, and evidence for dark matter candidates in the Universe. Following the discovery of the Higgs boson, further data will allow in-depth investigation of the boson's properties and thereby of the origin of mass.[3]

Chapter 3

Atlas South West Tier 2

Figure 3-1 ATLAS SWT2 Logo

The Tier 1 facility for the United States of America is located at the Brookhaven National Laboratory. There are five Tier 2 facilities in United States of America. Following are the five sites:

- ATLAS Southwest Tier 2
- Northeast Tier2
- ATLAS Great Lakes Tier 2
- Midwest Tier 2
- Western Tier 2 [This site no longer exists as of October,2018]
  [Updated in accordance to data from Patrick McGuigan ,FACULTY RESEARCH ASSOCIATE, Physics,UTA]

Atlas SWT2 is a collaborative effort amongst the physics departments of the University of Texas at Arlington (UTA), the University of Oklahoma (OU) and Langston University (LU) to support the processing of data generated in the Large Hadron Collider (LHC) for the Atlas experiment.

Following are the computing clusters present for Atlas SWT2

Cluster	Job Slots	Storage
UTA_SWT2	2360	440 TB
SWT2_CPB	9848	5500 TB
OU_OCHEP_SWT2	1800	500 TB
OU_OSCER	10000	
LUCILLE	960	110 TB

Table 3-1 Computing Clusters for Atlas SWT2

Of the above clusters the UTA_SWT2 and SWT2_CPB are being operated by University of Texas at Arlington as of summer of 2018.

Location of the data centers

- University of Oklahoma
  - Oklahoma Center for High Energy Physics (OCHEP)
  - OU Supercomputing Center for Education and Research (OSCER)
- University of Texas Arlington
  - Chemistry and Physics Building (CPB)
  - Arlington Regional Data Center (ARDC) in Fort Worth
- Langston University
  - Lucille [7]

Figure 3-2 US ATLAS SWT2 Center

Chapter 4

Rucio

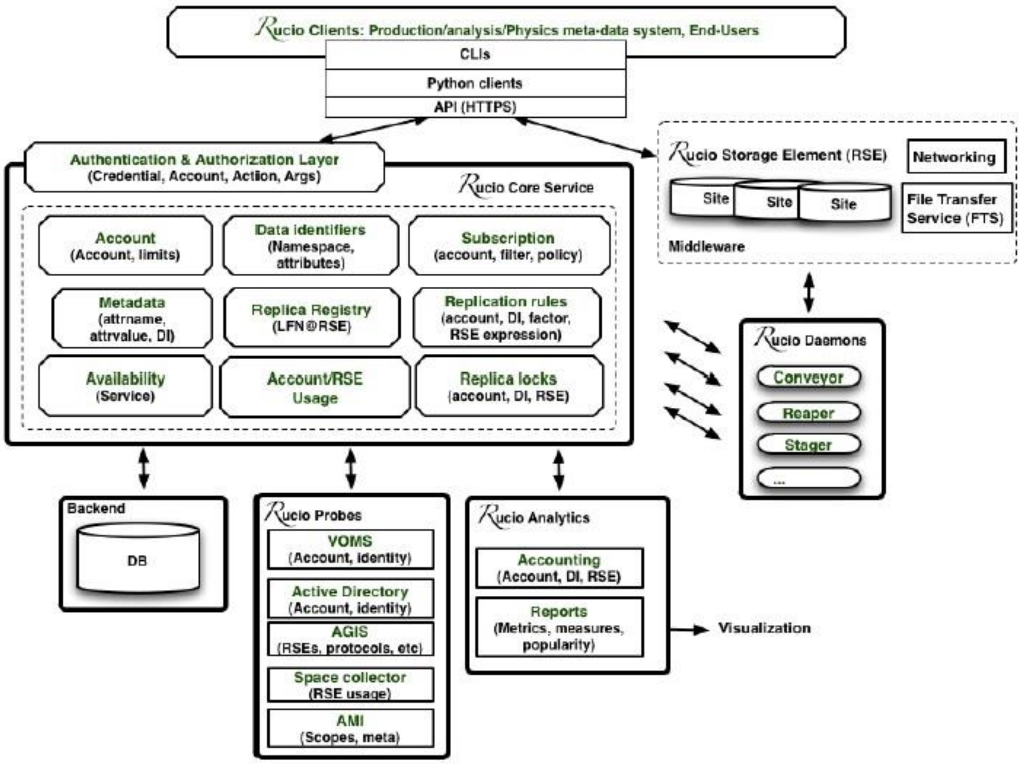


Figure 4-1 The Rucio Architecture

Image courtesy:
https://www.sciencedirect.com/science/article/pii/S2405601415006409

A brief about Rucio:

1) Rucio is the next-generation of Distributed Data Management (DDM) system benefiting from recent advances in cloud and ”Big Data” computing to address HEP experiments scaling requirements. Rucio is an evolution of the ATLAS DDM system Don Quixote 2 (DQ2).

2) Rucio is written in python. It is based on a distributed architecture and can be decomposed into different components as shown in the figure above. The main part is the core. It is composed of a set of frontend servers that interact with a database backend.
3) There are two kind of servers - the Rucio authentication nodes that get the user credentials and generate tokens and the Rucio servers that allow to create/list/modify Data Identifiers, rules, replicas, metadata, subscriptions.

4) The interface to the database is done through the use of an ORM2 layer (sqlalchemy) that allows to support multiple database type (Oracle , MySQL, PostgreSQL). The database backend used in ATLAS is Oracle .

5) To interact with the servers, Rucio provides a RESTful interface that can be used by every HTTP client. In addition to this REST interface, python clients are also available.

6) When a user wants to interact with Rucio, the client first contacts the authentication servers to get a token. The token can then be used to access the Rucio servers.

7) The daemons are a set of lightweight agents dedicated to very specific tasks like evaluating a rule or subscription, transferring or deleting files, recovering corrupted or lost files. They interact directly with the core part of Rucio, with the Rucio Storage Elements or some external service as the File Transfer Service (FTS). Each daemon is under the control of a supervisor process that allows to switch it on/off and restart it automatically in case of crash.

8) Sharding is used to distribute the load between different workers, which allows to scale horizontally.

9) A Nagios server is used to monitor the system and to synchronize it with external services like VOMS(Virtual Organization Management System), AGIS(ATLAS Grid Information System). Moreover the servers and

daemons also generate time series that are sent to a graphite server which allows to have a good overview of the performances of the system.

10) Finally Rucio includes an analytic part based on Hadoop. All the log files of the system are daily moved on a Hadoop Distributed File System (HDFS) and can be analyzed via map/reduce jobs. This analytic part is also used to provide extra information like the list of file replicas on a site. [6]

Chapter 5

About the storage servers that are to be monitored

There are about 30 storage servers There is a top level directory called Storage. Then under this directory are three directories namely :
ATLAS Data disk
ATLAS Scratch disk
ATLAS Local group disk
Space is reported by tokens i.e ATLAS Data disk, ATLAS Scratch disk, ATLAS Local group disk are called space tokens . Periodically these space tokens are queried to find out how much space is used in a particular area .This data is taken and written into a special JSON file.The JSON is stored in known path for each of the token areas.Each of the space tokens is associated with a path which is registered with Rucio which in turn accesses these paths to find out how much space is actually used by the data present at the servers.

The division of ATLAS Data disk, ATLAS Scratch disk, ATLAS Local group disk is nothing but the way ATLAS chooses to segregate its data and create space for different research activities on its data .
Data disk stores data that ATLAS has sent to the servers for processing and storage.
Scratch disk is a temporary storage area. This short term storage area is for users who work on ATLAS's data and store the results of the analysis of the data and results their experiments.

In the JSON file- http://monitor.atlas-swt2.org/rucio/swt2.json , from where we get the data for monitoring the storage servers we will be monitoring the following RSEDATA tokens:
SWT2_CPB_DATADISK
SWT2_CPB_SCRATCHDISK
SWT2_CPB_LOCALGROUPDISKS
UTA_SWT2 _DATADISK
These are the space tokens for the servers at the UT Arlington sites.

RSE here stands for Rucio Storage Element and each of the following - SWT2_CPB_DATADISK, SWT2_CPB_SCRATCHDISK, SWT2_CPB_LOCALGROUPDISKS, UTA_SWT2 _DATADISK – is an RSE to Rucio.

(As per communication with Patrick McGuigan ,FACULTY RESEARCH ASSOCIATE, Physics,UTA)

The data in each of the above mentioned disk spaces tell us about the data that Rucio has gathered from the space token paths registered at Rucio and what Rucio has calculated -the actual disk space that should have been occupied by the data transferred to the datadisks and is being tracked by Rucio.It also gives us the data that Rucio considers is expired and obsolete. Rucio as a data management system to keep track of the data flowing into the system and how much space that data is occupying in the servers.

Rucio: This parameter gives us the count of data that is already present in the disk as well as the data is in the process of getting replicated to the disk.

Unavailable: This parameter gives us the count of data being copied to the site or files declared bad and being recovered.

Expired: This parameter gives us the count of files that can be deleted if space is needed.

Obsolete: This parameter gives us the count of files that must be deleted instantly.

(The above data is as per a mail from Cedric Serfon [cedric.serfon@cern.ch] from CERN)

Of all the given data the “rucio” and “json” data points are of utmost importance to us. The data point ”rucio” as has been mentioned earlier is the calculation on Rucio’s part as to much space should have been occupied by the data it is tracking in the servers at UT Arlington site and “json” is the data which is calculated on running services on the servers at UT Arlington sites to determine how much space has actually been occupied by the data tracked by Rucio.”gsiftp” is another value calculated by running gsift service at the local servers and point to the space occupied by the data at the UTA site servers just like “json”.
Space occupancy percentage is calculated based on the total space occupied indicated by “json”.
We are plotting a set of graphs for Datadisk,Local Groupdisk and Scratch Disk.
For Transfer and Deletion Luccile(Langston University) and OU Oscer (University of Oklahoma) will be tracked as well as Datadisk ,Local Groupdisk and Scratch Disk.

Chapter 6

Motivation behind the thesis

In the lowest level Rucio deals with files. It not only manages the data it also tracks the data (i.e. it also keeps a track of the number of copies of that data). Rucio is also responsible for moving the data across locations i.e. instances. For moving of data Rucio relies on the FTS(File Transfer Service). FTS is responsible for aggregating the files that needs to be transferred and affects those transfers. The FTS communicates with the SRM which in turn communicates with GRID FTP to make the final transfers.
(As per communication with Patrick McGuigan ,FACULTY RESEARCH ASSOCIATE, Physics)

The GRID FTP (Worldwide LHC Computing Grid) provides continuous access to computing resources. Users make a job request from any of the entry points into the system. It is the responsibility of the Grid to establishes the identity of the user, checks their credentials, and searches for available sites that can provide the resources requested. Users do not have to worry about where the resources are coming from.[5]

Currently the health (the actual amount of raw storage used with respect to the overall storage capacity of the cluster) of the storage clusters also known as Rucio Storage Element (RSE) are monitored by Rucio. Rucio probes the RSEs present at the University campus as well as the Fort Worth location and gathers information about:

1) The capacity of the RSE that is being currently used by the data which has been transferred from the ATLAS Tier 1 resources so as to monitor whether optimal usage of resources is being done.
2) Monitoring deletions at the UTA cluster sites.
3) Monitoring transfers from the UTA sites.

Results:

A graphic representation of all this data and transfer of data is available to UTA in the form of:

Figure 6-1 System health report provided by Rucio

Figure 6-2 System health report provided by Ganglia

* Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids.

These reports though provide a fair idea about the data but since these images are not interactive and just a snapshot of the storage systems it is not convenient as a monitoring system for the people responsible here at UTA.

Distributed computing requires the use of geographically separated hosts, services, and software. Thus, monitoring these hosts become a very important aspect.

Monitoring RSEs at UTA sites:

CERN monitors all the sites to which it's data flows by using Rucio which is Distributed Data Management (DDM) system. This means that CERN at any given point of time has complete knowledge about the health of all the sites at which it's data from the Atlas experiment is being processed and stored. This is their way to ensure that all the work on the data goes on as planned. If there is any discrepancy between what is expected and what is actually happening at a RSE the responsible authorities can be alerted so that necessary action can be taken.

Monitoring deletions at the UTA cluster sites:

From time to time data at the remote sites are marked for deletion by Rucio. Rucio maintains its own database through which it can track the existence of data files at the different locations. It also maintains the list of duplicate data files that are present at the various sites. Data files are marked for deletion when their usage is complete and they have to be removed to create more space for new data.

This data deletion activity is monitored by Rucio as well to track successful deletion of data and failure in deletion of data. This is an important activity since deletion would create the required space that is planned for so that new data files can be transferred into the RSEs.

Monitoring transfers from the UTA cluster sites:

Rucio not only monitors data it also maintains all the details about the data files that it monitors. Thus when required it also marks data for transfer from the remote RSEs to another remote RSE for optimal usage of disk space and computing capabilities.

GridFTP (Grid File Transfer Protocol) services, as the name suggests allows for the transfer of files between hosts within the grid, does the job of moving data into and out of the clusters. The data transfer activity is monitored by Rucio so as to track successful transfer of all the intended data files. In case of failures the responsible authorities at CERN are alerted.

(As per communication with Patrick McGuigan ,FACULTY RESEARCH ASSOCIATE, Physics)

Chapter 7

Goals to be achieved by the thesis

If there is a monitoring system already in place why do we need another monitoring system?

Here is the reason. Though there is a highly effective monitoring system in place it primarily serves CERN to monitor CERN's data and the data's movement from CERN data centers to data centers around the world. Rucio prods the local RSEs, for example, at an University campus which is working as a Tier 2 resource for CERN to gather information about their health . It in-turn updates the authorities at CERN about the status of these remote RSEs and if there is an issue, alerts and panic calls originate from CERN to the local authorities. The local authorities then investigate their local Storage and Compute Elements (CE) to determine the cause of issue or failure. Thus, since this is a lengthy process which spans across continents naturally the time between the occurrence of an issue to the time it gets reported to the concerned people on site of the affected RSE is way more than it should be. Thus, we needed a new system that would work on the same resources that Rucio uses to gain information about the health of the RSEs concerned, in our case it is the two clusters present at UT Arlington campus and Fort Worth site, and create a parallel monitoring system that would:

1) Gather information from the local systems at regular intervals and create an alert if it finds any surge in the storage units.
2) Track failure in deletions from the SWT2 (South West Tier 2) sites and alert the system administrators if the percentage of failure is above a pre-determined threshold value.
3) Track failure in transfer of files from the SWT2 sites to other remote clouds and transfer of files from remote clouds to the local sites and alert the system administrator if the percentage of failure in transfer goes beyond a pre-determined threshold value.

Chapter 8

Problem Statement

There is essentially two parts to the problem:

1) Health tracking of Rucio Storage Elements(RSE)
2) DDM monitoring service

1) Health tracking of Rucio Storage Elements(RSE)

Currently the Rucio monitoring system tracks the data disk usage at all the Southwest Tier 2 clusters. For the people responsible for the storage and computing facilities at UTA it is very essential to know about the health of the systems. All the monitoring data is available through a JSON file which is accessible to us through the following file :

http://monitor.atlasswt2.org/rucio/swt2.json

The above mentioned json is not the json Rucio uses for monitoring but has been created by the UT Arlington system monitoring people since due to security reasons the json Rucio uses can not be access by anyone not authorized by CERN.This json contains the same data as the once that Rucio uses.

The JSON file provides a snapshot of the system health every four hours. In the JSON file the field “rucio” under SWT2_CPB_DATADISK, SWT2_CPB_SCRATCHDISK, SWT2_CPB_LOCALGROUPDISK and UTA_SWT2_DATADISK gives the amount of storage space the Rucio system believes is used by the currently available data at the site and the field “json” gives the amount of space that is actually occupied by the data available at the sites. The values that are provided for “json” are obtained by the Rucio monitoring system by running a job on the actual storage sites. Thus, essentially the data pointed to by “rucio” and “json” are same and thus they should be same as well. This seldom happens. There is always a difference in the values of “used” and “free” among “rucio” and “json”. Also at some times the difference between the data becomes so high that it becomes hard to determine whether there is actually a fault in the system or the jobs.

Deletion of data from the University of Arlington clusters are tracked by Rucio and the data generated is available to us through the Deletion JSON file(page 49).

2) DDM monitoring service

The files stored at the storage systems are monitored by the DDM monitoring service and deletions happen based on availability of space on the RSEs (Rucio Storage Element).

Two situations can occur in the RSEs:

1) Any RSE can be the destination of transfer for data files.
2) Any RSE can be the source for transfer of data files.

1) Deletions from an RSE (RSE as destination of transfer) : Deletions from a RSE typically happens when a particular RSE is using around 90% (this value has been hard coded in the code and if in future the value needs to be changed the code needs to be altered to accommodate the change) of its total capacity of storage. Rucio monitors its RSEs and once data is to be transferred to a particular RSE and its capacity to store data has been reached, data from that particular RSE is marked for deletion and is deleted until its storage capacity reaches acceptable levels.

We are interested in tracking the transfer of data to the following RSEs as well as track the rate of deletions happening at these RSEs:

US LUCILLE
US OU_OSCER_ATLAS
US SWT2_CPB
US UTA_SWT2

A snapshot of the DDM dashboard is given below:

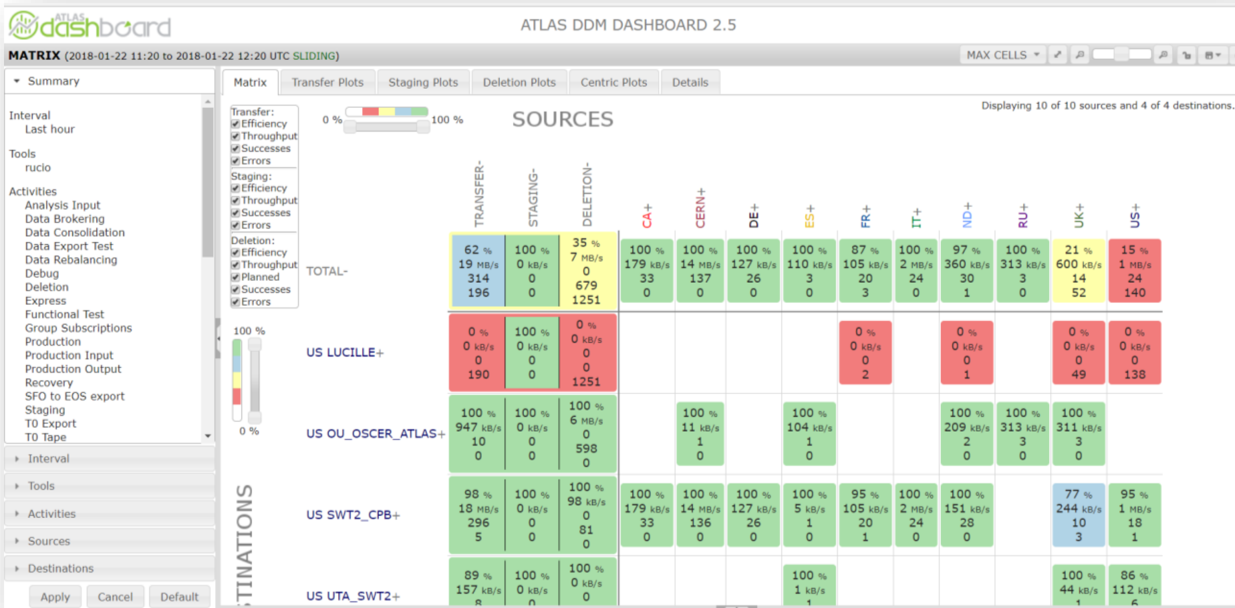
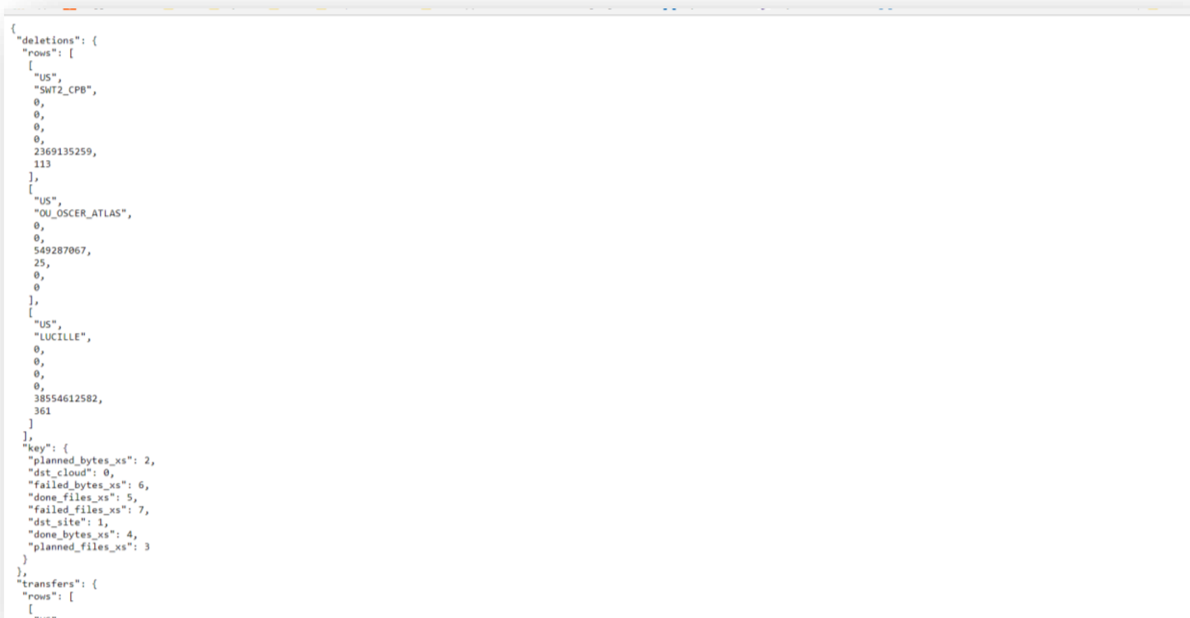


Figure 8-1 DDM TABLE view for our sites as a destination and

deletions that are to happen (Last Hour)

As seen in the dashboard the columns and rows give us the pictorial view of the data that is to be monitored. Currently though a color scheme is used to show the storage health we need a more efficient monitoring system that would not only indicate the exact percentage of transfer happening at a given point in time and raise an alarm once the transfer percentage falls below a pre-determined value and deletion of data falls below a pre-determined value.

To capture the data for the above-mentioned system we will be using the following JSON:




```
{
 "deletions": {
  "rows": [
   [
    "US",
    "SWT2_CPB",
    0,
    0,
    0,
    0,
    2369135259,
    113
   ],
   [
    "US",
    "OU_OSCER_ATLAS",
    0,
    0,
    549287067,
    25,
    0,
    0
   ],
   [
    "US",
    "LUCILLE",
    0,
    0,
    0,
    0,
    38554612582,
    361
   ]
  ],
  "key": {
   "planned_bytes_xs": 2,
   "dst_cloud": 0,
   "failed_bytes_xs": 6,
   "done_files_xs": 5,
   "failed_files_xs": 7,
   "dst_site": 1,
   "done_bytes_xs": 4,
   "planned_files_xs": 3
  }
 },
 "transfers": {
  "rows": [
   [
```


Figure 8-2 DDM JSON URL for our sites as a destination of transfer and deletions

(LAST HOUR) [15]

Here it can be noted that the "key" field is being used as an index which interprets the meaning of the data above i.e. for example

```
"key": {
   "planned_bytes_xs": 2
   "dst_cloud": 0,
   "failed_bytes_xs": 6,
   "done_files_xs": 5,
   "failed_files_xs": 7,
   "dst_site": 1,
   "done_bytes_xs": 4,
   "planned_files_xs": 3

}
```

Would essentially be interpreted as

```
"deletions": {
  "rows": [
   [
    "US",
    "SWT2_CPB",
    0,
    0,
    0,
    0,
    2369135259,
    113
   ],
```

"dst_cloud" = "US"

```
"dst_site"= "SWT2_CPB"
"planned_bytes_xs"= 0
"planned_files_xs": 0
"done_bytes_xs": 0
"done_files_xs": 0
"failed_bytes_xs": 2369135259
"failed_files_xs": 113
```

Note: the index here starts from 0

Data can originate from any of the following (country based clouds except for CERN) clouds :
CA,CERN,DE,ES,FR,IT,ND,NL,RU,TW,UK ,US

Following are fields that would be extracted from the JSON and saved in the database for analysis:

dst_site
done_files_xs
failed_files_xs
done_bytes
failed_bytes
planned_files
planned_bytes

The done_bytes will be plotted in a chart against the field to_date captured from the JSON.

The failed_bytes will be plotted in a chart against the field to_date captured from the JSON.
The planned_bytes will be plotted in a chart against the field to_date captured from the JSON.
The done_files_xs will be plotted in a chart against the field to_date captured from the JSON.
The failed_files_xs will be plotted in a chart against the field to_date captured from the JSON.
The planned_files will be plotted in a chart against the field to_date captured from the JSON.

The above-mentioned charts will give the user a pictorial view of the amount of data that is being deleted and the amount of data that is being planned for deletion and the amount of data that is failing deletion.

In the present monitoring system there is no provision for the user at UT Arlington site to get to know if the storage system is in critical condition as far as the capacity of the storage system(health) is concerned. There is no provision for apprise the concerned parties if deletion of files from a particular RSE is failing. The system that will be built would raise an alarm through a mailing system before an RSE completely goes down and no more transfer of files to the RSE is possible.

2) Transfer of files from an RSE (RSE as source of transfer) : Similarly, files are transferred from our RSEs to the clouds of data mentioned above. A snapshot below of the DDM dashboard gives us an idea about the number of files that are marked for transfer from the SWT2 RSEs to the data clouds.

The data for these transfers is captured from the JSON below:

```
    9439637665,
    13,
    138
   ],
   [
    "TW",
    5394359488,
    4,
    1
   ],
   [
    "US",
    95499525740,
    1774,
    158
   ],
   [
    "CA",
    9980666630,
    18,
    49
   ],
   [
    "UK",
    19754444366,
    18,
    29
   ],
   [
    "DE",
    149305171931,
    120,
    165
   ],
   [
    "IT",
    2881610437,
    3,
    5
   ]
  ],
  "key": {
   "errors_xs": 3,
   "files_xs": 2,
   "dst_cloud": 0,
   "bytes_xs": 1
  }
 },
```


Figure 8-3 JSON for transfer of files from an RSE [17]


```
{
 "deletions": {
  "rows": [
   [
    "RU",
    0,
    0,
    13538045287,
    581,
    0,
    0
   ],

  "key": {
   "planned_bytes_xs": 1,
   "dst_cloud": 0,
   "done_files_xs": 4,
   "failed_files_xs": 6,
   "planned_files_xs": 2,
   "done_bytes_xs": 3,
   "failed_bytes_xs": 5
  }
```

Here it can be noted that the “key” field is being used as an index which interprets the meaning of the data above i.e.

```
"planned_bytes_xs": 1 =0,
"dst_cloud": 0 ="RU",
"done_files_xs": 4 =581,
"failed_files_xs": 6 =0,
"planned_files_xs": 2 =0,
"done_bytes_xs": 3 =13538045287,
"failed_bytes_xs": 5 =0
```

Which essentially interprets to
planned_bytes_xs = number of bytes that are planned for deletion.
dst_cloud = destination cloud
done_files_xs= number of files that are deleted
failed_files_xs= number of files that failed to get deleted
planned_files_xs=number of files planned for deletion in recent future
done_bytes_xs= number of bytes already deleted
failed_bytes_xs= number of bytes that failed deletion

The snapshot below shows the transfers of data from SWT2 clusters to the various clouds across the globe

Figure 8-4 transfers of data from SWT2 clusters to the various clouds across the globe

Chapter 9

Analysis and Implementation of the solution

The project be broadly divided into two portion:

1) The data extraction and analysis part.
2) The data visualization part.

Data extraction and analysis:

For data extraction and analysis python is used as our programming language and for database MongoDB is used.

MongoDB was chosen as the database because it is a free and open-source cross-platform NoSQL database which is stores data as JSON-like documents and is highly scalable (MongoDB scales horizontally using sharding) and easy to maintain. Also tables can be easily created from the scripts. Though it is very user friendly it does not compromise on performance. It supports field, range query, and regular expression searches.[18]

The data visualization:

To visualize and pictorially interpret the data the powerful visualization library of d3js has been used.

D3JS is a powerful JavaScript library bring data to life using HTML, SVG, and CSS. [11]

Now for the actual application:

Data extraction:

Initially all the JSON files are studied to determine the data that needs to be extracted to create our monitoring system. The deletion and transfer JSON files are updated every hour with fresh data where as the storage JSON is expected to get fresh data every four hours. The following JSON (storage) file is used to track the health of the RSEs at UTA sites:

http://monitor.atlas-swt2.org/rucio/swt2.json

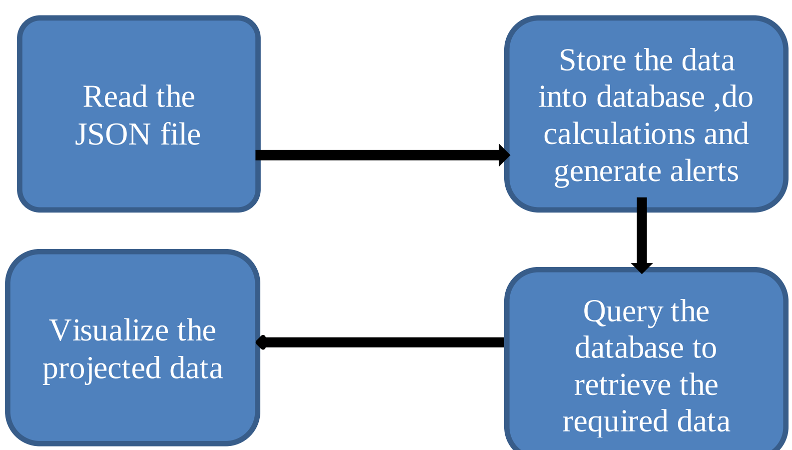
This json is used for monitoring the health of the RSEs at the UTA sites as viewed by the Rucio Monitoring System. As per the Rucio monitoring system - for the SWT2_CPB_DATADISK, SWT2_CPB_SCRATCHDISK, SWT2_CPB_ LOCALGROUPDISK and the UTA_SWT2_DATADISK RSEs the computed space that should be occupied by the data tracked by RUCIO at these sites is given by the value "used" for the field "rucio". The Rucio Monitoring System also prods the RSEs to retrieve the actual space occupied by the data present in the disks .

This value is reflected in the value "used" for the field "json":

rucio:
used: 3885234453400617
rse: "SWT2_CPB_DATADISK"
updated_at: "Tue, 03 Apr 2018 10:04:36 UTC"
free: null
source: "rucio"
total: 3885234453400617
json:
used: 3899977295239168
rse: "SWT2_CPB_DATADISK"
updated_at: "Tue, 03 Apr 2018 09:55:38 UTC"
free: 455022704760832
source: "json"
total: 4355000000000000

In an ideal situation both the fields : rucio "used" and json "used" should have reflected the exact same values for "used".This does not happen. Most of the time the usage of diskspace reflected in "rucio" and "json" are different thus indicating toward issues either in the jobs running on the system or faults in the system itself.

To better interpret this situation our solution will calculate the difference in the values of "rucio" and "json" used space and based on whether the percentage by which they differ from each other is of concern (depending on a pre-determined value) an alert will be generated. The alert would let the people responsible for the system know of a possible fault that is about to happen. Our solution can be broadly divided into four parts:



1) Collection of data and alert system:

Storage:

The most important part of the solution is the collection of the data as well as creating alerts when an aberration from expected result is noticed. Expected result here is:

The data we require to analyze the clusters is obtained from the following JSON : http://monitor.atlas-swt2.org/rucio/swt2.json.

This json is updated every four hours . Thus we can expect to get fresh set of data every four hours. Thus a script will extract the data from the json every four hours and store the data is the database.

The amount of storage occupied by the data in the SWT2_CPB_DATADISK,SWT2_CPB_LOCALGROUPDISK,SWT2_CPB_SCRATCHDISK and UTA_SWT2_DATADISK should not exceed 95% (this data has been hard coded into the code and is the acceptable level of data storage. Setting the system storage below 95% would generate extremely frequent alerts for failure and is not desireable) of the total storage space available to each to these space tokens. The script fetchData_Storage.py runs every four hour to determine the amount of space occupied at each of these space tokens. Once it is observed that any of the space tokens have data more than 95% of the total storage space assigned to it a mail is triggered to the concerned people letting them know of the situation of the storage. The total disk space is always indicated by the total disk space indicated by json total for each of the space tokens.

The captured data is stored in the Mongo database. The data will range up to a year so that the historical data can be analyzed when required. Below is an example of json total for SWT2_CPB_DATADISK space token from the json file:

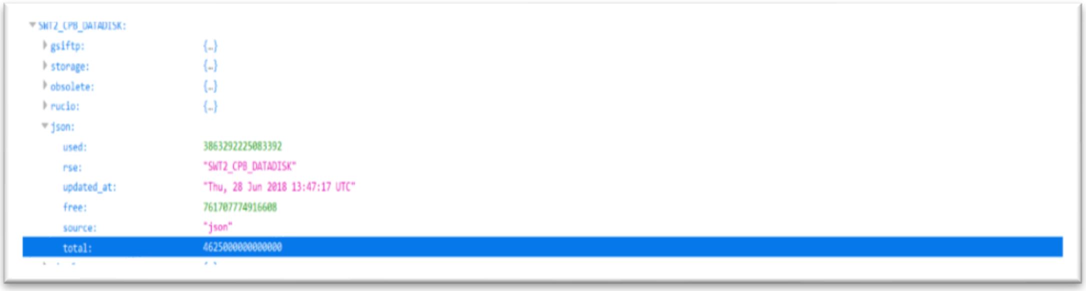


Figure 9-1 SWT2_CPB_DATADISK space token

Similarly we will get different values for json total for each of the space tokens : SWT2_CPB_LOCALGROUPDISK, SWT2_CPB_SCRATCHDISK and UTA_SWT2_DATADISK. Every value of "used" will be stored against a corresponding "updated_at" value. Which would look like the following in the MongoDB.

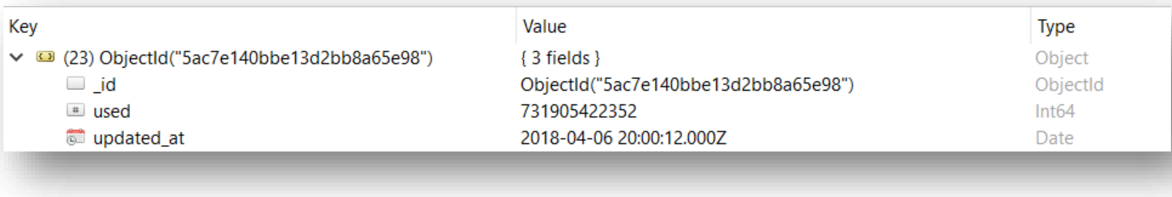


Figure 9-2 Data in MongoDB

Every time the data is captured from the json we will also calculate the discrepancy between "rucio" and "json" values using the following formula:

((((rucio_used - json_used) / json_SWT2_CPB_DATADISK) * 100 > 10) or (((rucio_used - json_used) / 1024^4) > datadisk_offby)

Note : The above formula has been copied from the python code

Where json_SWT2_CPB_DATADISK = total space indicated by json for the SWT2_CPB_DATADISK space token
rucio_used=space occupied by data indicated by rucio
json_used= space occupied by data indicated by json
datadisk_offby=100 TB (set in the config file)

If the discrepancy between the two data points – rucio_used and json_used is found to be above 10%(the predetermined value) of total disk space[calculated by ((rucio_used - json_used) / json_SWT2_CPB_DATADISK) * 100] or by 100 TB [calculated by (rucio_used - json_used) / 1024[4]] then an mail will be sent to the concerned people here at UT Arlington. This will give the people responsible for the UTA sites to take possible measure to figure out the reason for the discrepancy and enough time to analyze the issue that is causing the surge in the space taken by the data and analyze the reason for discrepancy in expected values of storage space being used by the data.

Deletion:

In fetchData_Deletion : Deletion of data files is monitored for the following sites:

SWT2_CPB

UTA_SWT2

OU_OSCER_ATLAS

LUCILLE

For all the above sites the number of files and bytes successfully deleted – done_files_xs, done_bytes and the number of files and bytes which failed to get deleted successfully -failed_files_xs and failed_bytes_xs are monitored every hour.

Alert mails get generated on two conditions :

1) If the number of deleted files (done_files_xs) is not zero using the following formula the percentage of failure is calculated:

$$n = (failed\_bytes / done\_bytes) * 100$$

If the value of n is found to be greater than 10% an alert email is generated and sent to the concerned people.

2) Else if the number of deleted files (done_files_xs) is zero and the number of failed bytes (failed_files_xs) is not zero an alert email is generated and sent to the concerned people with the amount of failed files and done files .

Transfer:
In fetchData_Transfer_UTA_Dest and fetchData_Transfer_UTA_Src: For transfer of data files from source to destination the number of files that were marked for transfer but failed to get transferred (error_xs) and the total number of files that were marked for transfer and got transferred(files_xs) in the last hour is monitored.
Using the following formula the percentage of files that failed to get transferred is calculated:

error_prcnt= [errors_xs / (errors_xs + files_xs)] x 100

If the error percentage is above 70% (pre-determined value hard coded in the code) a mail gets generated alerting the concerned people.

The following four scripts do the collection of data for the project:

<u>Storage Space monitoring data</u> is collected by the script **<u>fetchData_Storage.py</u>**. This script is executed by cron jobs setup in the server once every four hours assuming that a fresh set of data is received every four hours via the json **<u>http://monitor.atlasswt2.org/rucio/swt2.json</u>** .The data is stored in the tables :

For the SWT2_CPB DATADISK

- rucio_SWT2_CPB_DATADISK
- json_SWT2_CPB_DATADISK
- expired_SWT2_CPB_DATADISK
- unavailable_SWT2_CPB_DATADISK
- gsiftp_SWT2_CPB_DATADISK
- obsolete_SWT2_CPB_DATADISK

For the SWT2_CPB_LOCALGROUPDISK

- rucio_SWT2_CPB_LOCALGROUPDISK
- json_SWT2_CPB_LOCALGROUPDISK
- expired_SWT2_CPB_LOCALGROUPDISK
- unavailable_SWT2_CPB_LOCALGROUPDISK
- gsiftp_SWT2_CPB_LOCALGROUPDISK
- obsolete_SWT2_CPB_LOCALGROUPDISK

For the SWT2_CPB_SCRATCHDISK

- rucio_SWT2_CPB_SCRATCHDISK
- json_SWT2_CPB_SCRATCHDISK
- expired_SWT2_CPB_SCRATCHDISK
- unavailable_SWT2_CPB_SCRATCHDISK

- gsiftp_SWT2_CPB_SCRATCHDISK
- obsolete_SWT2_CPB_SCRATCHDISK

For the UTA_SWT2_DATADISK

- rucio_UTA_SWT2_DATADISK
- json_UTA_SWT2_DATADISK
- expired_UTA_SWT2_DATADISK
- unavailable_UTA_SWT2_DATADISK
- gsiftp_UTA_SWT2_DATADISK
- obsolete_UTA_SWT2_DATADISK

Duplicate data entry in each of the above tables is prevented by using "updated_at" value for each table as a unique key. Alerts are generated in the form of a mail for each of the above data values : rucio , json, expired, unavailable, gsiftp and obsolete for their values "used" – for storage space used by each as compared to the total storage space as indicated by "json".If the value exceeds some predefined percentage : currently set to 95%(hard coded into the code) - a mail is sent by the system to the intended recipients.

Snapshot of rucio_SWT2_CPB_DATADISK in MongoDB

```
Type "it" for more
> db.rucio_SWT2_CPB_DATADISK.find()
{ "_id" : ObjectId("5ad137cff95a8d1a39f0f221"), "used" : NumberLong("3928729964416747"), "updated_at" : ISODate("2018-04-13T22:04:37Z") }
{ "_id" : ObjectId("5ad1b867f95a8d4a390ffb8a"), "used" : NumberLong("3933703077242764"), "updated_at" : ISODate("2018-04-14T07:04:36Z") }
{ "_id" : ObjectId("5ad22deef95a8d6d57501efc"), "used" : NumberLong("3940187938212891"), "updated_at" : ISODate("2018-04-14T14:04:36Z") }
{ "_id" : ObjectId("5ad2ce9df95a8d1f7ae31b0d"), "used" : NumberLong("3931281883699295"), "updated_at" : ISODate("2018-04-15T02:04:36Z") }
{ "_id" : ObjectId("5ad30b6af95a8d3230072cc3"), "used" : NumberLong("3934038805775318"), "updated_at" : ISODate("2018-04-15T07:04:36Z") }
{ "_id" : ObjectId("5ad3e540f95a8d754c09c069"), "used" : NumberLong("3935884583725324"), "updated_at" : ISODate("2018-04-15T22:04:36Z") }
{ "_id" : ObjectId("5ad41c0ff95a8d076b250067"), "used" : NumberLong("3936531301779032"), "updated_at" : ISODate("2018-04-16T02:04:36Z") }
{ "_id" : ObjectId("5ad610f0f95a8d244b534f9b"), "used" : NumberLong("3905443136498046"), "updated_at" : ISODate("2018-04-17T14:04:36Z") }
{ "_id" : ObjectId("5ad65de6f95a8d3ba482917c"), "used" : NumberLong("3909362757017542"), "updated_at" : ISODate("2018-04-17T18:04:36Z") }
{ "_id" : ObjectId("5ad6c9cdf95a8d5e93d9fc5b"), "used" : NumberLong("3913217758831497"), "updated_at" : ISODate("2018-04-18T02:04:36Z") }
{ "_id" : ObjectId("5ad6e5ecf95a8d672a1e0263"), "used" : NumberLong("3921898959976103"), "updated_at" : ISODate("2018-04-18T06:04:36Z") }
{ "_id" : ObjectId("5ad6f3fdf95a8d6b5e798641"), "used" : NumberLong("3922662227961910"), "updated_at" : ISODate("2018-04-18T07:04:36Z") }
{ "_id" : ObjectId("5ad71e2bf95a8d7a4f46b459"), "used" : NumberLong("3925393954167495"), "updated_at" : ISODate("2018-04-18T10:04:36Z") }
{ "_id" : ObjectId("5ad7566df95a8d0cbffe12b9"), "used" : NumberLong("3927574499286605"), "updated_at" : ISODate("2018-04-18T14:04:36Z") }
{ "_id" : ObjectId("5ad78eacf95a8d1dd72b0b6b"), "used" : NumberLong("3935124075293536"), "updated_at" : ISODate("2018-04-18T18:04:36Z") }
{ "_id" : ObjectId("5ad7c6ecf95a8d2eef72b42c"), "used" : NumberLong("3937210716040262"), "updated_at" : ISODate("2018-04-18T22:04:37Z") }
{ "_id" : ObjectId("5ad7ff2df95a8d401521fc8f"), "used" : NumberLong("3939835987723690"), "updated_at" : ISODate("2018-04-19T02:04:37Z") }
{ "_id" : ObjectId("5ad8376cf95a8d514720f91f"), "used" : NumberLong("3921672273101183"), "updated_at" : ISODate("2018-04-19T06:04:36Z") }
{ "_id" : ObjectId("5ad8457df95a8d558cd66a11"), "used" : NumberLong("3921811044648484"), "updated_at" : ISODate("2018-04-19T07:04:36Z") }
{ "_id" : ObjectId("5ad86fbef95a8d64275556db"), "used" : NumberLong("3922562264359688"), "updated_at" : ISODate("2018-04-19T10:04:36Z") }
Type "it" for more
>
```

Table 9-1 rucio_SWT2_CPB_DATADISK Table:

_id	used	updated_at
5ad137cff95a8d1a39f0f221	3928729964416740	2018-04-13T22:04:37Z
5ad1b867f95a8d4a390ffb8a	3933703077242760	2018-04-14T07:04:36Z
5ad2ce9df95a8d1f7ae31b0d	3940187938212890	2018-04-14T14:04:36Z
5ad30b6af95a8d3230072cc3	3931281883699290	2018-04-15T02:04:36Z
5ad3e540f95a8d754c09c069	3935884583725320	2018-04-15T22:04:36Z
5ad41c0ff95a8d076b250067	3936531301779030	2018-04-16T02:04:36Z
5ad610f0f95a8d244b534f9b	3905443136498040	2018-04-17T14:04:36Z
5ad65de6f95a8d3ba482917c	3909362757017540	2018-04-17T18:04:36Z
5ad6c9cdf95a8d5e93d9fc5b	3913217758831490	2018-04-18T02:04:36Z
5ad6e5ecf95a8d672a1e0263	3921898959976100	2018-04-18T06:04:36Z
5ad6f3fdf95a8d6b5e798641	3922662227961910	2018-04-18T07:04:36Z
5ad71e2bf95a8d7a4f46b459	3925393954167490	2018-04-18T10:04:36Z
5ad7566df95a8d0cbffe12b9	3927574499286600	2018-04-18T14:04:36Z
5ad78eacf95a8d1dd72b0b6b	3935124075293530	2018-04-18T18:04:36Z
5ad7c6ecf95a8d2eef72b42c	3937210716040260	2018-04-18T22:04:37Z
5ad7ff2df95a8d401521fc8f	3939835987723690	2018-04-19T02:04:37Z
5ad8376cf95a8d514720f91f	3921672273101180	2018-04-19T06:04:36Z
5ad8457df95a8d558cd66a11	3921811044648480	2018-04-19T07:04:36Z
5ad86fbef95a8d64275556db	3922562264359680	2018-04-19T10:04:36Z

Where _id = unique identifier for each document in the MongoDB

Deletion of files monitoring data is collected by the script **fetchData_Deletion.py** This script is executed by cron jobs setup in the server once every hour assuming that a fresh set of data is received every hour. The data are stored in the following tables:

For Lucille-

- del_LUCILLE

For OU_OSCER_ATLAS-

- del_ OU_OSCER_ATLAS

For SWT2_CPB-

- del_ SWT2_CPB

For UTA_SWT2-

- del_ UTA_SWT2

Table 9-2 del_SWT2_CPB Table:

_id	planned_bytes _xs	done_files _xs	failed_files _xs	failed_bytes _xs	done_bytes_xs	planned_ files_xs	to_date
5b43f702f95a8d47e3d056fe	0	2354	0	0	18574881394	0	2018-07-10T00:00:00Z
5b43e8f3f95a8d435270c823	0	2359	0	0	31041557144	0	2018-07-09T23:00:00Z
5b43dae2f95a8d3ec9305cd8	0	141	0	0	3427592636	0	2018-07-09T22:00:00Z
5b43ccd3f95a8d3a758f1763	0	454	0	0	9333218373	0	2018-07-09T21:00:00Z
5b43bec2f95a8d362b415573	0	570	0	0	11663388280	0	2018-07-09T20:00:00Z
5b43b0b2f95a8d31e5debe26	0	1520	0	0	55343667736	0	2018-07-09T19:00:00Z
5b43a2a2f95a8d2d9bb7bdbd	0	628	0	0	7036372155	0	2018-07-09T18:00:00Z
5b439493f95a8d29502525f3	0	843	0	0	2297626458	0	2018-07-09T17:00:00Z
5b438683f95a8d24ee17c35e	0	80	0	0	27187478419	0	2018-07-09T16:00:00Z
5b437873f95a8d208498d067	0	296	0	0	5966362499	0	2018-07-09T15:00:00Z
5b436a62f95a8d1c3ea20e6f	0	263	0	0	8563922176	0	2018-07-09T14:00:00Z
5b435c53f95a8d17ce7bf9b8	0	192	0	0	4236779049	0	2018-07-09T13:00:00Z
5b434e42f95a8d137846f8ae	0	67	0	0	62910321	0	2018-07-09T12:00:00Z
5b433223f95a8d0aef32c341	0	164	0	0	948881939	0	2018-07-09T10:00:00Z
5b432412f95a8d063c4b47c6	0	102	0	0	318748736	0	2018-07-09T09:00:00Z
5b431603f95a8d7e6703744d	0	256	0	0	1543026467	0	2018-07-09T08:00:00Z
5b4307f3f95a8d7a070df44b	0	22	0	0	67924158	0	2018-07-09T07:00:00Z
5b42f9e3f95a8d75b1a1e916	0	30	0	0	1735134629	0	2018-07-09T06:00:00Z
5b42ebd3f95a8d714f50f521	0	76	0	0	630584053	0	2018-07-09T05:00:00Z
5b42ddc2f95a8d6cfce90484	0	51	0	0	92084927	0	2018-07-09T04:00:00Z

Where _id = unique identifier for each document in the MongoDB

Snapshot of del_SWT2_CPB in MongoDB:

```
> db.del_SWT2_CPB.find().sort({"to_date":-1})
{ "_id" : ObjectId("5b43f702f95a8d47e3d056fe"), "planned_bytes_xs" : 0, "done_files_xs" : 2354, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("18574881394"), "p
lanned_files_xs" : 0, "to_date" : ISODate("2018-07-10T00:00:00Z") }
{ "_id" : ObjectId("5b43e8f3f95a8d435270c823"), "planned_bytes_xs" : 0, "done_files_xs" : 2359, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("31041557144"), "p
lanned_files_xs" : 0, "to_date" : ISODate("2018-07-09T23:00:00Z") }
{ "_id" : ObjectId("5b43dae2f95a8d3ec9305cd8"), "planned_bytes_xs" : 0, "done_files_xs" : 141, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("3427592636"), "pla
nned_files_xs" : 0, "to_date" : ISODate("2018-07-09T22:00:00Z") }
{ "_id" : ObjectId("5b43ccd3f95a8d3a758f1763"), "planned_bytes_xs" : 0, "done_files_xs" : 454, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("9333218373"), "pla
nned_files_xs" : 0, "to_date" : ISODate("2018-07-09T21:00:00Z") }
{ "_id" : ObjectId("5b43bec2f95a8d362b415573"), "planned_bytes_xs" : 0, "done_files_xs" : 570, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("11663388280"), "pl
anned_files_xs" : 0, "to_date" : ISODate("2018-07-09T20:00:00Z") }
{ "_id" : ObjectId("5b43b0b2f95a8d31e5debe26"), "planned_bytes_xs" : 0, "done_files_xs" : 1520, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("55343667736"), "p
lanned_files_xs" : 0, "to_date" : ISODate("2018-07-09T19:00:00Z") }
{ "_id" : ObjectId("5b43a2a2f95a8d2d9bb7bdbd"), "planned_bytes_xs" : 0, "done_files_xs" : 628, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("7036372155"), "pla
nned_files_xs" : 0, "to_date" : ISODate("2018-07-09T18:00:00Z") }
{ "_id" : ObjectId("5b439493f95a8d29502525f3"), "planned_bytes_xs" : 0, "done_files_xs" : 843, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("2297626458"), "pla
nned_files_xs" : 0, "to_date" : ISODate("2018-07-09T17:00:00Z") }
{ "_id" : ObjectId("5b438683f95a8d24ee17c35e"), "planned_bytes_xs" : 0, "done_files_xs" : 80, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("27187478419"), "pla
nned_files_xs" : 0, "to_date" : ISODate("2018-07-09T16:00:00Z") }
{ "_id" : ObjectId("5b437873f95a8d208498d067"), "planned_bytes_xs" : 0, "done_files_xs" : 296, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("5966362499"), "pla
nned_files_xs" : 0, "to_date" : ISODate("2018-07-09T15:00:00Z") }
{ "_id" : ObjectId("5b436a62f95a8d1c3ea20e6f"), "planned_bytes_xs" : 0, "done_files_xs" : 263, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("8563922176"), "pla
nned_files_xs" : 0, "to_date" : ISODate("2018-07-09T14:00:00Z") }
{ "_id" : ObjectId("5b435c53f95a8d17ce7bf9b8"), "planned_bytes_xs" : 0, "done_files_xs" : 192, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : NumberLong("4236779049"), "pla
nned_files_xs" : 0, "to_date" : ISODate("2018-07-09T13:00:00Z") }
{ "_id" : ObjectId("5b434e42f95a8d137846f8ae"), "planned_bytes_xs" : 0, "done_files_xs" : 67, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : 62910321, "planned_files_xs" :
0, "to_date" : ISODate("2018-07-09T12:00:00Z") }
{ "_id" : ObjectId("5b433223f95a8d0aef32c341"), "planned_bytes_xs" : 0, "done_files_xs" : 164, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : 948881939, "planned_files_xs"
: 0, "to_date" : ISODate("2018-07-09T10:00:00Z") }
{ "_id" : ObjectId("5b432412f95a8d063c4b47c6"), "planned_bytes_xs" : 0, "done_files_xs" : 102, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : 318748736, "planned_files_xs"
: 0, "to_date" : ISODate("2018-07-09T09:00:00Z") }
{ "_id" : ObjectId("5b431603f95a8d7e6703744d"), "planned_bytes_xs" : 0, "done_files_xs" : 256, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : 1543026467, "planned_files_xs"
 : 0, "to_date" : ISODate("2018-07-09T08:00:00Z") }
{ "_id" : ObjectId("5b4307f3f95a8d7a070df44b"), "planned_bytes_xs" : 0, "done_files_xs" : 22, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : 67924158, "planned_files_xs" :
0, "to_date" : ISODate("2018-07-09T07:00:00Z") }
{ "_id" : ObjectId("5b42f9e3f95a8d75b1a1e916"), "planned_bytes_xs" : 0, "done_files_xs" : 30, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : 1735134629, "planned_files_xs"
: 0, "to_date" : ISODate("2018-07-09T06:00:00Z") }
{ "_id" : ObjectId("5b42ebd3f95a8d714f50f521"), "planned_bytes_xs" : 0, "done_files_xs" : 76, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : 630584053, "planned_files_xs" :
 0, "to_date" : ISODate("2018-07-09T05:00:00Z") }
{ "_id" : ObjectId("5b42ddc2f95a8d6cfce90484"), "planned_bytes_xs" : 0, "done_files_xs" : 51, "failed_files_xs" : 0, "failed_bytes_xs" : 0, "done_bytes_xs" : 92084927, "planned_files_xs" :
0, "to_date" : ISODate("2018-07-09T04:00:00Z") }
```

Transfer of files monitoring data are collected by the scripts fetchData_Transfer_UTA_Dest.py where SWT2 sites are the destination of transfer of files and fetchData_Transfer_UTA_Src.py where SWT2 sites are the source of transfer of files.

The source/destination clouds are :

- US
- IT
- CERN
- DE
- ES
- CA
- RU
- UK
- TW
- NL
- ND
- FR

SWT2 destinations/sources are :

- SWT2_CPB
- UTA_SWT2
- OU_OSCER_ATLAS

Snapshot of Transfer_Tables_UTA_Dest:


```
> db.Transfer_Tables_UTA_Dest.find()
{ "_id" : ObjectId("5add9284f95a8d7be8c16bcf"), "errors_xs" : 7, "files_xs" : 84, "bytes_xs" : NumberLong("11108521202"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T08:00:00Z"), "d
st_site" : "UTA_SWT2", "error_prcnt" : 7.692307692307693 }
{ "_id" : ObjectId("5adda092f95a8d03faf8cc68"), "errors_xs" : 19, "files_xs" : 78, "bytes_xs" : NumberLong("172749528465"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T09:00:00Z"),
"dst_site" : "UTA_SWT2", "error_prcnt" : 19.58762886597938 }
{ "_id" : ObjectId("5addaea2f95a8d08e8bb6918"), "errors_xs" : 7, "files_xs" : 117, "bytes_xs" : NumberLong("192250499688"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T10:00:00Z"),
"dst_site" : "UTA_SWT2", "error_prcnt" : 5.64516129032258 }
{ "_id" : ObjectId("5addbcb2f95a8d0e783243f9"), "errors_xs" : 2, "files_xs" : 192, "bytes_xs" : NumberLong("364547576272"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T11:00:00Z"),
"dst_site" : "UTA_SWT2", "error_prcnt" : 1.0309278350515463 }
{ "_id" : ObjectId("5addcac2f95a8d13ad2fe01c"), "errors_xs" : 4, "files_xs" : 116, "bytes_xs" : NumberLong("273510470766"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T12:00:00Z"),
"dst_site" : "UTA_SWT2", "error_prcnt" : 3.3333333333333335 }
{ "_id" : ObjectId("5addd8d6f95a8d190ecd7dff"), "errors_xs" : 39, "files_xs" : 61, "bytes_xs" : NumberLong("140676982550"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T13:00:00Z"),
"dst_site" : "UTA_SWT2", "error_prcnt" : 39 }
{ "_id" : ObjectId("5adde6e5f95a8d1ebcc8554c"), "errors_xs" : 53, "files_xs" : 103, "bytes_xs" : NumberLong("130344525849"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T14:00:00Z"),
 "dst_site" : "UTA_SWT2", "error_prcnt" : 33.97435897435898 }
{ "_id" : ObjectId("5addf4f5f95a8d23d93b73e2"), "errors_xs" : 45, "files_xs" : 128, "bytes_xs" : NumberLong("257518745066"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T15:00:00Z"),
 "dst_site" : "UTA_SWT2", "error_prcnt" : 26.011560693641616 }
{ "_id" : ObjectId("5ade0302f95a8d29e0235fe3"), "errors_xs" : 13, "files_xs" : 165, "bytes_xs" : NumberLong("265164701253"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T16:00:00Z"),
 "dst_site" : "UTA_SWT2", "error_prcnt" : 7.303370786516854 }
{ "_id" : ObjectId("5ade1115f95a8d2f913f4632"), "errors_xs" : 61, "files_xs" : 121, "bytes_xs" : NumberLong("218919837594"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T17:00:00Z"),
 "dst_site" : "UTA_SWT2", "error_prcnt" : 33.51648351648351 }
{ "_id" : ObjectId("5ade1f22f95a8d35a1a4b2d0"), "errors_xs" : 9, "files_xs" : 124, "bytes_xs" : NumberLong("291347213901"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T18:00:00Z"),
"dst_site" : "UTA_SWT2", "error_prcnt" : 6.7669172932330826 }
{ "_id" : ObjectId("5ade2d36f95a8d3bd5dddf96"), "errors_xs" : 28, "files_xs" : 199, "bytes_xs" : NumberLong("391249173807"), "src_cloud" : "US", "to_date" : ISODate("2018-04-23T19:00:00Z"),
 "dst_site" : "UTA_SWT2", "error_prcnt" : 12.334801762114537 }
{ "_id" : ObjectId("5ade2d36f95a8d3bd5dddf99"), "errors_xs" : 2, "files_xs" : 39, "bytes_xs" : NumberLong("12231080617"), "src_cloud" : "IT", "to_date" : ISODate("2018-04-23T19:00:00Z"), "d
st_site" : "OU_OSCER_ATLAS", "error_prcnt" : 4.878048780487805 }
{ "_id" : ObjectId("5ade2d36f95a8d3bd5dddf9c"), "errors_xs" : 5, "files_xs" : 0, "bytes_xs" : 0, "src_cloud" : "CERN", "to_date" : ISODate("2018-04-23T19:00:00Z"), "dst_site" : "UTA_SWT2",
"error_prcnt" : 4.878048780487805 }
{ "_id" : ObjectId("5ade2d39f95a8d3bd5dddf9f"), "errors_xs" : 19, "files_xs" : 55, "bytes_xs" : NumberLong("27480923176"), "src_cloud" : "DE", "to_date" : ISODate("2018-04-23T19:00:00Z"), "
dst_site" : "OU_OSCER_ATLAS", "error_prcnt" : 25.675675675675674 }
{ "_id" : ObjectId("5ade2d39f95a8d3bd5dddfa2"), "errors_xs" : 1, "files_xs" : 10, "bytes_xs" : NumberLong("15474787224"), "src_cloud" : "ES", "to_date" : ISODate("2018-04-23T19:00:00Z"), "d
st_site" : "UTA_SWT2", "error_prcnt" : 9.090909090909092 }
{ "_id" : ObjectId("5ade2d39f95a8d3bd5dddfa5"), "errors_xs" : 0, "files_xs" : 6, "bytes_xs" : 12180256, "src_cloud" : "CA", "to_date" : ISODate("2018-04-23T19:00:00Z"), "dst_site" : "SWT2_C
PB", "error_prcnt" : 9.090909090909092 }
{ "_id" : ObjectId("5ade2d39f95a8d3bd5dddfa8"), "errors_xs" : 0, "files_xs" : 2, "bytes_xs" : 19967534, "src_cloud" : "TW", "to_date" : ISODate("2018-04-23T19:00:00Z"), "dst_site" : "SWT2_C
PB", "error_prcnt" : 9.090909090909092 }
{ "_id" : ObjectId("5ade2d39f95a8d3bd5dddfab"), "errors_xs" : 4, "files_xs" : 228, "bytes_xs" : 1306991913, "src_cloud" : "RU", "to_date" : ISODate("2018-04-23T19:00:00Z"), "dst_site" : "OU
_OSCER_ATLAS", "error_prcnt" : 1.7241379310344827 }
{ "_id" : ObjectId("5ade2d39f95a8d3bd5dddfae"), "errors_xs" : 0, "files_xs" : 20, "bytes_xs" : NumberLong("7419088565"), "src_cloud" : "DE", "to_date" : ISODate("2018-04-23T19:00:00Z"), "ds
t_site" : "SWT2_CPB", "error_prcnt" : 1.7241379310344827 }
Type "it" for more
>
```

Table 9-3 Transfer_Tables_UTA_Dest Table:

_id	errors_xs	files_xs	bytes_xs	src_cloud	to_date	dst_site	error_prcnt
5b4137e2f95a8d691ac73fc4	1	2	1068480288	US	2018-07-07T22:00:00Z	UTA_SWT2	33.33333333
5b4137e2f95a8d691ac73fc7	0	3	3145728	IT	2018-07-07T22:00:00Z	OU_OSCER_ATLAS	0
5b4137e2f95a8d691ac73fca	0	1	1589145	CERN	2018-07-07T22:00:00Z	UTA_SWT2	0
5b4137e2f95a8d691ac73fcd	0	3	4194304	DE	2018-07-07T22:00:00Z	OU_OSCER_ATLAS	0
5b4137e2f95a8d691ac73fd3	0	5	80079104	ES	2018-07-07T22:00:00Z	UTA_SWT2	0
5b4137e2f95a8d691ac73fdc	6	92	10412733441	CA	2018-07-07T22:00:00Z	SWT2_CPB	6.12244898
5b4137e2f95a8d691ac73fdf	0	3	3145728	DE	2018-07-07T22:00:00Z	SWT2_CPB	0
5b4137e2f95a8d691ac73fe2	0	3	3145728	RU	2018-07-07T22:00:00Z	UTA_SWT2	0
5b4137e2f95a8d691ac73fe8	1	825	14485672052	DE	2018-07-07T22:00:00Z	UTA_SWT2	0.121065375
5b4137e2f95a8d691ac73feb	0	8	8388608	US	2018-07-07T22:00:00Z	SWT2_CPB	0
5b4137e2f95a8d691ac73fee	10	12	107595705779	TW	2018-07-07T22:00:00Z	OU_OSCER_ATLAS	45.45454545
5b4137e2f95a8d691ac73ff1	0	4	4194304	UK	2018-07-07T22:00:00Z	SWT2_CPB	0

Where _id = unique identifier for each document in the MongoDB

Snapshot of Transfer_Tables_UTA_Src :


```
> db.Transfer_Tables_UTA_Src.find()
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03c5"), "errors_xs" : 0, "files_xs" : 6, "bytes_xs" : 187739957, "src_site" : "OU_OSCER_ATLAS", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_clo
ud" : "CA", "error_prcnt" : 0 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03c8"), "errors_xs" : 0, "files_xs" : 1, "bytes_xs" : 364816, "src_site" : "OU_OSCER_ATLAS", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_cloud"
 : "TW", "error_prcnt" : 0 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03cb"), "errors_xs" : 9, "files_xs" : 13, "bytes_xs" : NumberLong("4176722819"), "src_site" : "OU_OSCER_ATLAS", "to_date" : ISODate("2018-06-05T22:00
:00Z"), "dst_cloud" : "DE", "error_prcnt" : 40.909090909090914 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03ce"), "errors_xs" : 0, "files_xs" : 2, "bytes_xs" : 3429714, "src_site" : "OU_OSCER_ATLAS", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_cloud
" : "RU", "error_prcnt" : 40.909090909090914 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03d1"), "errors_xs" : 0, "files_xs" : 1, "bytes_xs" : 1199559591, "src_site" : "OU_OSCER_ATLAS", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_cl
oud" : "ES", "error_prcnt" : 40.909090909090914 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03d4"), "errors_xs" : 0, "files_xs" : 2, "bytes_xs" : 85996014, "src_site" : "OU_OSCER_ATLAS", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_clou
d" : "NL", "error_prcnt" : 40.909090909090914 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03d7"), "errors_xs" : 0, "files_xs" : 10, "bytes_xs" : 182843175, "src_site" : "OU_OSCER_ATLAS", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_cl
oud" : "FR", "error_prcnt" : 40.909090909090914 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03da"), "errors_xs" : 0, "files_xs" : 3, "bytes_xs" : 270422133, "src_site" : "SWT2_CPB", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_cloud" :
"NL", "error_prcnt" : 40.909090909090914 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03dd"), "errors_xs" : 17, "files_xs" : 136, "bytes_xs" : NumberLong("63974881437"), "src_site" : "SWT2_CPB", "to_date" : ISODate("2018-06-05T22:00:00
Z"), "dst_cloud" : "DE", "error_prcnt" : 11.11111111111111 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03e0"), "errors_xs" : 0, "files_xs" : 14, "bytes_xs" : NumberLong("14033370942"), "src_site" : "SWT2_CPB", "to_date" : ISODate("2018-06-05T22:00:00Z"
), "dst_cloud" : "CA", "error_prcnt" : 11.11111111111111 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03e3"), "errors_xs" : 7, "files_xs" : 85, "bytes_xs" : NumberLong("94332043310"), "src_site" : "SWT2_CPB", "to_date" : ISODate("2018-06-05T22:00:00Z"
), "dst_cloud" : "FR", "error_prcnt" : 7.608695652173914 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03e6"), "errors_xs" : 1, "files_xs" : 8, "bytes_xs" : 93545501, "src_site" : "OU_OSCER_ATLAS", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_clou
d" : "UK", "error_prcnt" : 11.11111111111111 }
{ "_id" : ObjectId("5b1707e2f95a8d1f762b03e9"), "errors_xs" : 0, "files_xs" : 45, "bytes_xs" : NumberLong("28817185469"), "src_site" : "SWT2_CPB", "to_date" : ISODate("2018-06-05T22:00:00Z"
), "dst_cloud" : "ES", "error_prcnt" : 11.11111111111111 }
{ "_id" : ObjectId("5b1707e3f95a8d1f762b03ec"), "errors_xs" : 10, "files_xs" : 5, "bytes_xs" : 362879884, "src_site" : "SWT2_CPB", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_cloud" :
 "RU", "error_prcnt" : 66.66666666666666 }
{ "_id" : ObjectId("5b1707e3f95a8d1f762b03ef"), "errors_xs" : 0, "files_xs" : 10, "bytes_xs" : NumberLong("11716902413"), "src_site" : "UTA_SWT2", "to_date" : ISODate("2018-06-05T22:00:00Z"
), "dst_cloud" : "US", "error_prcnt" : 66.66666666666666 }
{ "_id" : ObjectId("5b1707e3f95a8d1f762b03f2"), "errors_xs" : 1, "files_xs" : 44, "bytes_xs" : NumberLong("60772400221"), "src_site" : "SWT2_CPB", "to_date" : ISODate("2018-06-05T22:00:00Z"
), "dst_cloud" : "US", "error_prcnt" : 2.2222222222222223 }
{ "_id" : ObjectId("5b1707e3f95a8d1f762b03f5"), "errors_xs" : 0, "files_xs" : 18, "bytes_xs" : NumberLong("56724044824"), "src_site" : "SWT2_CPB", "to_date" : ISODate("2018-06-05T22:00:00Z"
), "dst_cloud" : "CERN", "error_prcnt" : 2.2222222222222223 }
{ "_id" : ObjectId("5b1707e3f95a8d1f762b03f8"), "errors_xs" : 21, "files_xs" : 205, "bytes_xs" : NumberLong("47550288252"), "src_site" : "SWT2_CPB", "to_date" : ISODate("2018-06-05T22:00:00
Z"), "dst_cloud" : "UK", "error_prcnt" : 9.29203539823009 }
{ "_id" : ObjectId("5b1707e3f95a8d1f762b03fb"), "errors_xs" : 1, "files_xs" : 7, "bytes_xs" : NumberLong("6374771865"), "src_site" : "UTA_SWT2", "to_date" : ISODate("2018-06-05T22:00:00Z"),
 "dst_cloud" : "UK", "error_prcnt" : 12.5 }
{ "_id" : ObjectId("5b1707e3f95a8d1f762b03fe"), "errors_xs" : 0, "files_xs" : 1, "bytes_xs" : 1182008, "src_site" : "OU_OSCER_ATLAS", "to_date" : ISODate("2018-06-05T22:00:00Z"), "dst_cloud
" : "IT", "error_prcnt" : 12.5 }
Type "it" for more
>
```

Table 9-4 Transfer_Tables_UTA_Src Table:

_id	errors_xs	files_xs	bytes_xs	src_site	to_date	dst_cloud	error_prcnt
5b1707e2f95a8d1f762b03d7	0	10	182843175	OU_OSCER_ATLAS	2018-06-05T22:00:00Z	FR	0
5b1707e2f95a8d1f762b03da	0	3	270422133	SWT2_CPB	2018-06-05T22:00:00Z	NL	0
5b1707e2f95a8d1f762b03dd	17	136	63974881437	SWT2_CPB	2018-06-05T22:00:00Z	DE	11.11111111
5b1707e2f95a8d1f762b03e0	0	14	14033370942	SWT2_CPB	2018-06-05T22:00:00Z	CA	0
5b1707e2f95a8d1f762b03e3	7	85	94332043310	SWT2_CPB	2018-06-05T22:00:00Z	FR	7.608695652
5b1707e3f95a8d1f762b0401	0	2	2711586893	UTA_SWT2	2018-06-05T22:00:00Z	DE	0
5b1707e3f95a8d1f762b0404	0	1	1187030	OU_OSCER_ATLAS	2018-06-05T22:00:00Z	US	0
5b1707e3f95a8d1f762b0407	2	85	15738701831	SWT2_CPB	2018-06-05T22:00:00Z	ND	2.298850575
5b1707e3f95a8d1f762b040a	1	14	17137120336	SWT2_CPB	2018-06-05T22:00:00Z	IT	6.666666667
5b1715f3f95a8d23d3ff16af	0	3	2523955458	UTA_SWT2	2018-06-05T23:00:00Z	UK	0
5b1715f5f95a8d23d3ff16b2	5	2	125125874	OU_OSCER_ATLAS	2018-06-05T23:00:00Z	DE	71.42857143
5b1715f5f95a8d23d3ff16b5	2	155	125418253910	SWT2_CPB	2018-06-05T23:00:00Z	US	1.27388535

Where _id = unique identifier for each document in the MongoDB

2) Visualization of data:
   In order to visualize the data we extract the data from the database and use d3js charts to plot the data with respect to time . For each of the space tokens a separate series of charts are created. For Deletion there will be separate charts as well as for Transfers.

Storage Monitoring:
All the four storage monitoring pages for each of the space tokens will have a separate dashboard :
SWT2_CPB_DATADISK Storage Monitoring Dashboard
SWT2_CPB_SCRATCHDISK Storage Monitoring Dashboard
SWT2_CPB_LOCALGROUPDISK Storage Monitoring Dashboard
UTA_SWT2_DATADISK Storage Monitoring Dashboard

The dashboards will be made of one comparative chart and six charts for each of the parameters monitored : rucio ,expired ,gsiftp ,obsolete ,json and unavailable- showing the data recorded for last seven days
Chart 1:
The first chart gives us a comparative study of all the parameters with respect to time. Each parameter can be toggled to get a chart with the desired parameters. The time period of this chart can be changed .The time period can range from 7 to 365 days.

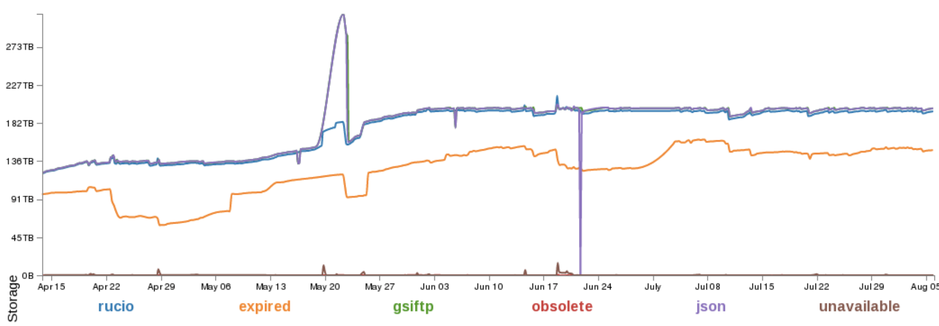


Figure 9-3 Comparison Chart

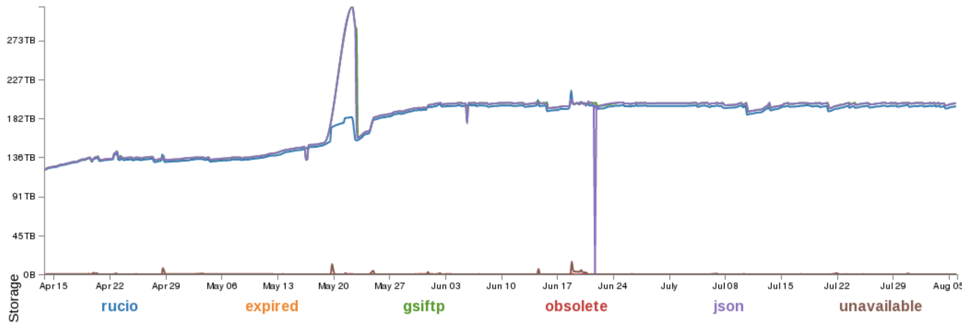


Figure 9-4 Comparison chart with the value line for “expired” toggled

The value for the range of days will be set from the settings page. Thus the data can be changes to reflect the behavior of the data over a different time period ranging from one day to last 365 days. Here it can be noted that the database will maintain the data for a year only.

Each of the parameters which are tracked will have a separate chart showing a snapshot of the behavior of the data over a period of seven days. However each parameter will have a further detail chart that would show the behavior of the parameter for a given time period(a year).

Following is an example of what a chart would look like. Here is an example for a 7-day chart for obsolete_SWT2_CPB_DATADISK

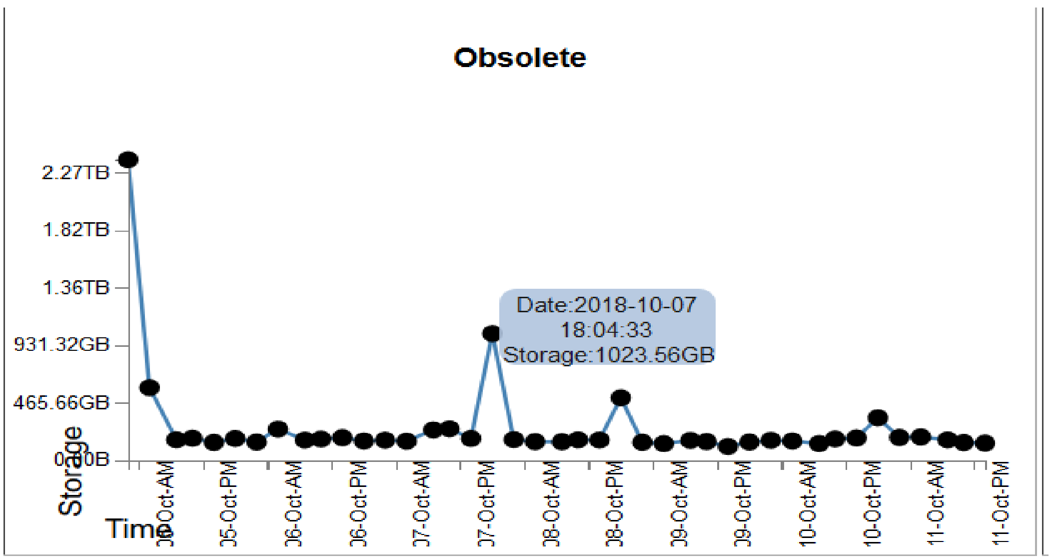


Figure 9-5 The obsolete chart showing 7 days of data

<u>As per system design</u>:

The x-axis values though show the same date but they are distinct data points based on their time capture. Due to lack of space in the x-axis the time has not been shown in the x-axis but every individual point in the chart would give the details about date, time and storage.

The y-axis auto scales and thus shows different number of values at different times.

The details chart can be reached by clicking on the name of the parameter. Here for example clicking on the name “Rucio” would land the user on the Rucio details chart for SWT2 CPB space token.

Below is a snapshot Rucio details chart :

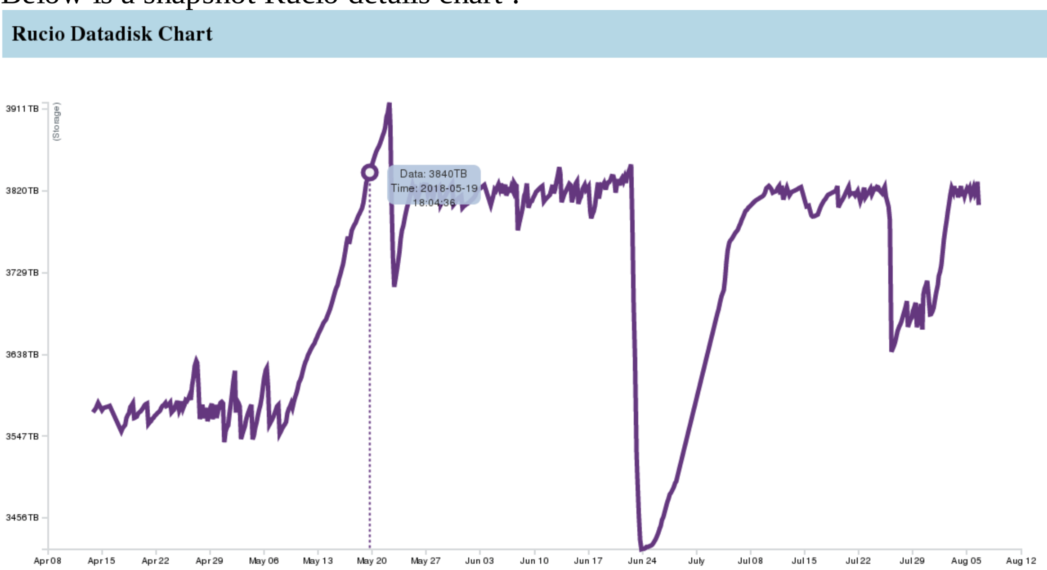


Figure 9-6 The rucio detail chart

Similarly charts are plotted for the rest of the parameters for the SWT2_CPB_DATADISK space token:

- json_SWT2_CPB_DATADISK
- expired_SWT2_CPB_DATADISK
- unavailable_SWT2_CPB_DATADISK
- gsiftp_SWT2_CPB_DATADISK
- obsolete_SWT2_CPB_DATADISK

and we will obtain something similar to this following chart

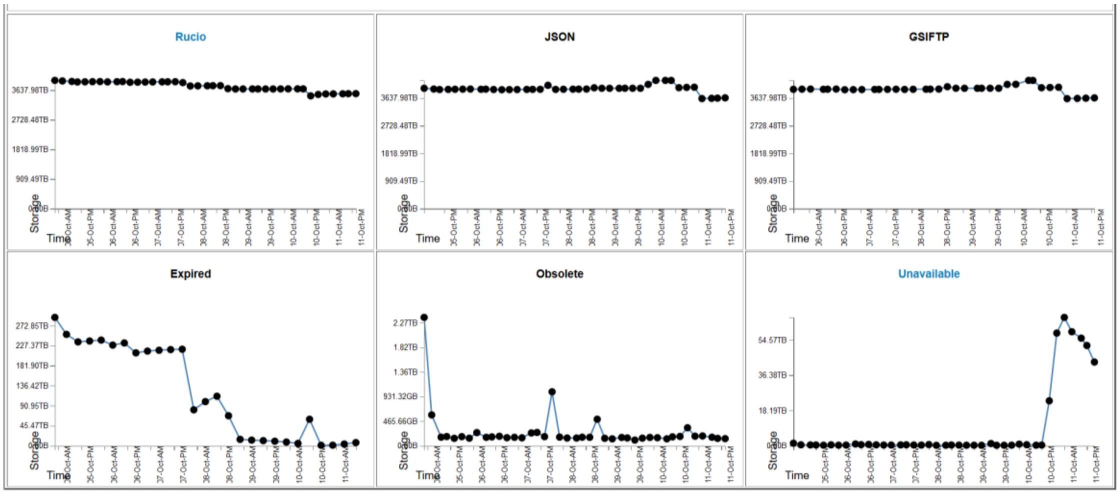


Figure 9-7 seven day chart

Deletion

Data from the storage units are marked for deletion from time to time when the files are no longer required or has already been moved to a different location out of the SWT2 system.

Deletion monitoring is another task done by Rucio. Rucio monitors all the data that is moved or deleted from the SWT2 system.

The data required to analyze the deletion of data from the clusters is obtained from the following Deletion JSON :

http://dashb-atlas-ddm.cern.ch/dashboard/request.py/matrix.json?tool=rucio&activity=Analysis+Input&activity=Data+Brokering&activity=Data+Consolidation&activity=Data+Export+Test&activity=Data+Rebalancing&activity=Debug&activity=Deletion&activity=Express&activity=Functional+Test&activity=Group+Subscriptions&activity=Production&activity=Production+Input&activity=Production+Output&activity=Recovery&activity=SFO+to+EOS+export&activity=Staging&activity=T0+Export&activity=T0+Tape&activity=User+Subscriptions&activity=default&activity=on&activity=on&activity=rucio-integration&activity=test&activity=test%3AT0_T1+export&activity=test%3AT1_T2+export&activity=testactivity10&activity=testactivity20&activity=testactivity70&activity_default_exclude=Upload%2FDownload+(Job)&activity_default_exclude=Upload%2FDownload+(User)&activity_default_exclude=Analysis+Download&activity_default_exclude=Analysis+Upload&activity_default_exclude=Production+Download&activity_default_exclude=Production+Upload&activity_default_exclude=CLI+Download&activity_default_exclude=CLI+Upload&activity_default_exclude=Analysis+Logs+OS+Upload&activity_default_exclude=Analysis+L

ogs+Upload&activity_default_exclude=Event+Service+Download&activity_default_exclude=Event+Service+Upload&activity_default_exclude=Production+Logs+OS+Upload&activity_default_exclude=Production+Logs+Upload&src_grouping=cloud&dst_cloud=%22US%22&dst_federation=US-SWT2&dst_grouping=cloud&dst_grouping=site&interval=60&prettyprint

the same data that we get from the json can be viewed in the following dashboard:

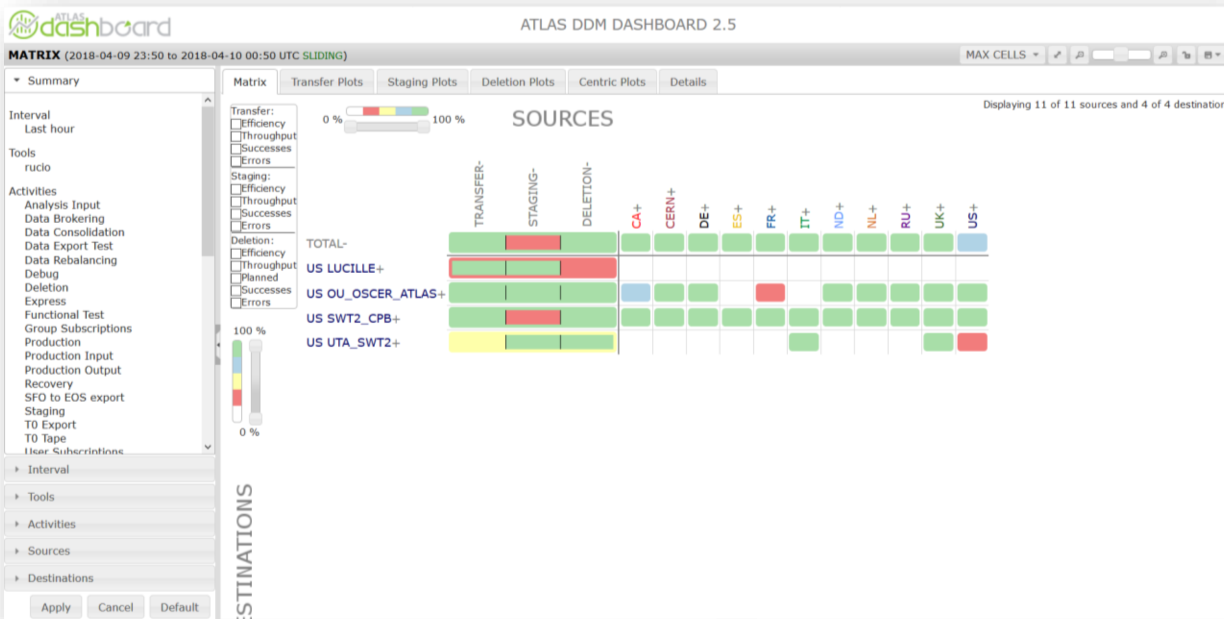


Figure 9-8 DDM TABLE view for our sites as a destination and Deletions

(Last Hour)

The data that we obtain from the json refreshes every hour. We can expect to get fresh set of data every hour. Thus a script will extract the data from the json every hour and store the data is the database. We are monitoring the parameters "dst_site","failed_bytes_xs", "planned_bytes_xs","done_bytes_xs", "failed_files_xs"," planned_files_xs" and "done_files_xs" which would essentially get interpreted to:

dst_site : The site from which data has been marked to be deleted.
failed_bytes_xs : The number of which were to be deleted from the cloud but failed to get deleted.

planned_bytes_xs : The number of bytes which are planned for deletion from a particular cloud.
done_files_xs: The number of bytes that were planned for deletion and were successfully deleted from a particular cloud.
failed_files_xs: The number of files that failed to get transferred
planned_files_xs: The number of files planned for deletion
done_files_xs: The number of files that successfully got transferred

Here site is nothing but the different Southwest Tier 2 locations:
US LUCILLE
US OU_OSCER_ATLAS
US SWT2_CPB
US UTA_SWT2

Here it can be noted that for deletions we will also track the behavior of the data deletion at the other two of the Southwest Tier 2 sites : US LUCILLE and US OU_OSCER_ATLAS besides the UTA sites US SWT2_CPB and US UTA_SWT2.

The data we require for analysis is extracted from the above mentioned json every hour. It is then stored in the table based on the dst_site

dst_site : SWT2_CPB target database: del_SWT2_CPB
dst_site : UTA_SWT2 target database: del_UTA_SWT2
dst_site : OU_OSCER_ATLAS target database: del_ OU_OSCER_ATLAS
dst_site : LUCILLE target database: del_LUCILLE

Which data is to be marked for deletion is decided by Rucio. Jobs are run on the RSEs for deletion of data. Sometimes there are issues due to which these jobs can not run till completion or due to other issues and the deletions of data fails.We are interested in monitoring how much of the data that was planned for deletion is successfully getting deleted and how much of it is failing to get deleted. If the amount of data that is failing to get deleted is more than the amount of data that is successfully getting deleted then that is a reason for concern.

The deletion landing page is a dashboard from which the user can choose the SWT2 site whose deletion health is to be monitored

Figure 9-9 Deletion Dashboard

Each of the options on the deletion dashboard takes user to the dashboard for the specific site's deletion monitoring dashboard.

Each dashboard will comprise of one comparative chart and six charts for each of the parameters: failed bytes, planned bytes, done bytes, failed files, planned files and done files.

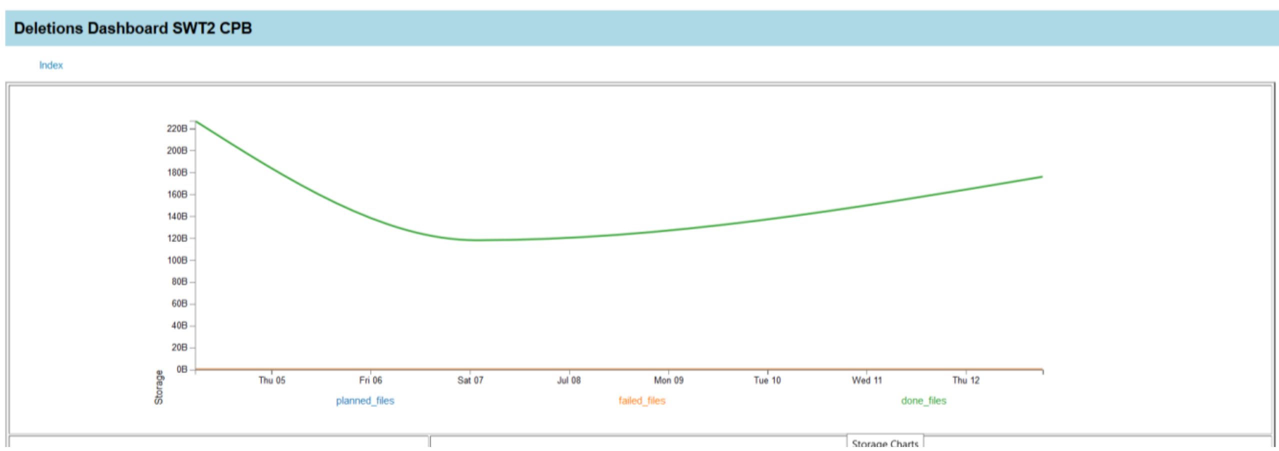


Figure 9-10 Snapshot of the comparative chart for SWT2_CPB

The chart above gives us the comparative chart based on the values of the number of planned files, failed files and done files over time. The user can choose to see the change in the numbers for the parameters over time by adjusting the range of the x-axis using the settings. The x-axis can range from 7 days to 365 days.

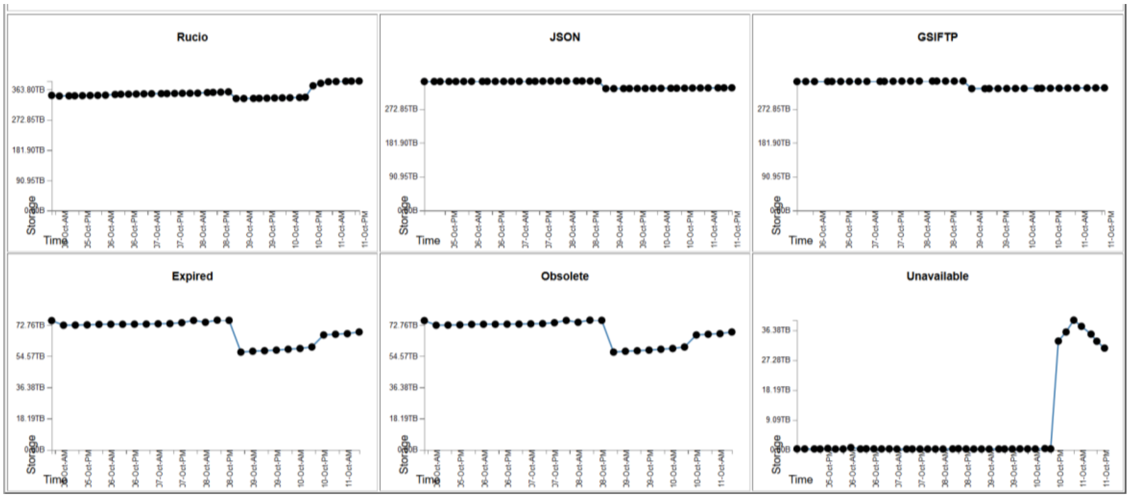


Figure 9-11 Snapshot of the weekly chart

The above chart gives a change of values of the monitored parameters over the last one week. Each parameter in turn has a details page which can be reached by clicking on the name of the parameter.

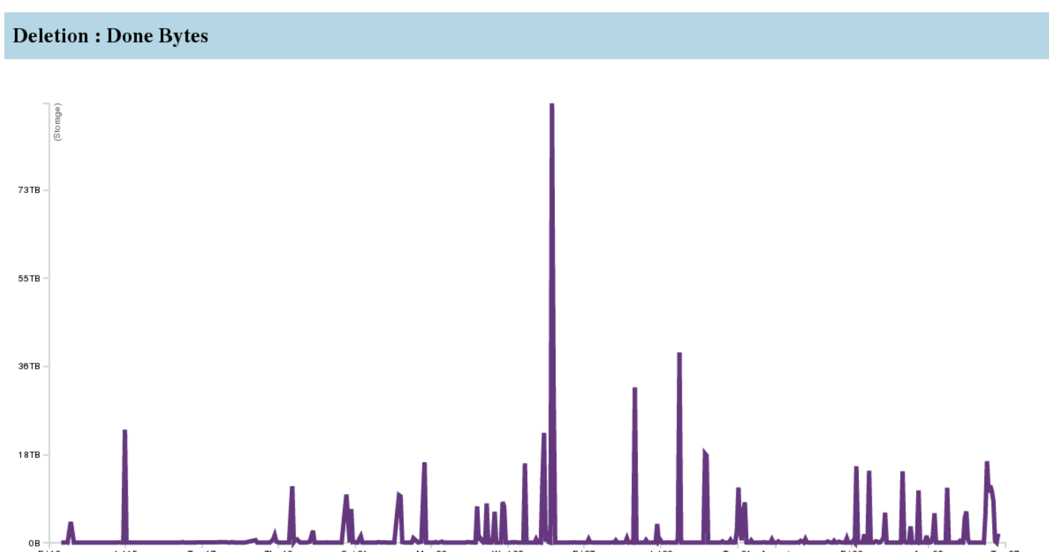


Figure 9-12 Snapshot of SWT2_CPB Done Bytes details page

The details page gives the change in behavior of the parameter over a time of a year. The range of days for the x-axis can be changed using the settings page.

Transfer

Transfer is a two way process. Data is transferred into the Southwest Tier 2 clusters from cloud storages from around the world and data is transferred out of the Southwest Tier 2 clusters into the various cloud storages around the world.

Transfer of data where Southwest Tier 2 clusters are the destinations

The snapshot below shows us the transfer of data between OU_OSCER_ATLAS, SWT2_CPB, UTA_SWT2 clusters as destinations and CA,CERN,DE,ES,FR,IT,ND,RU,TW,UK,US as (country based except for CERN) source clouds.

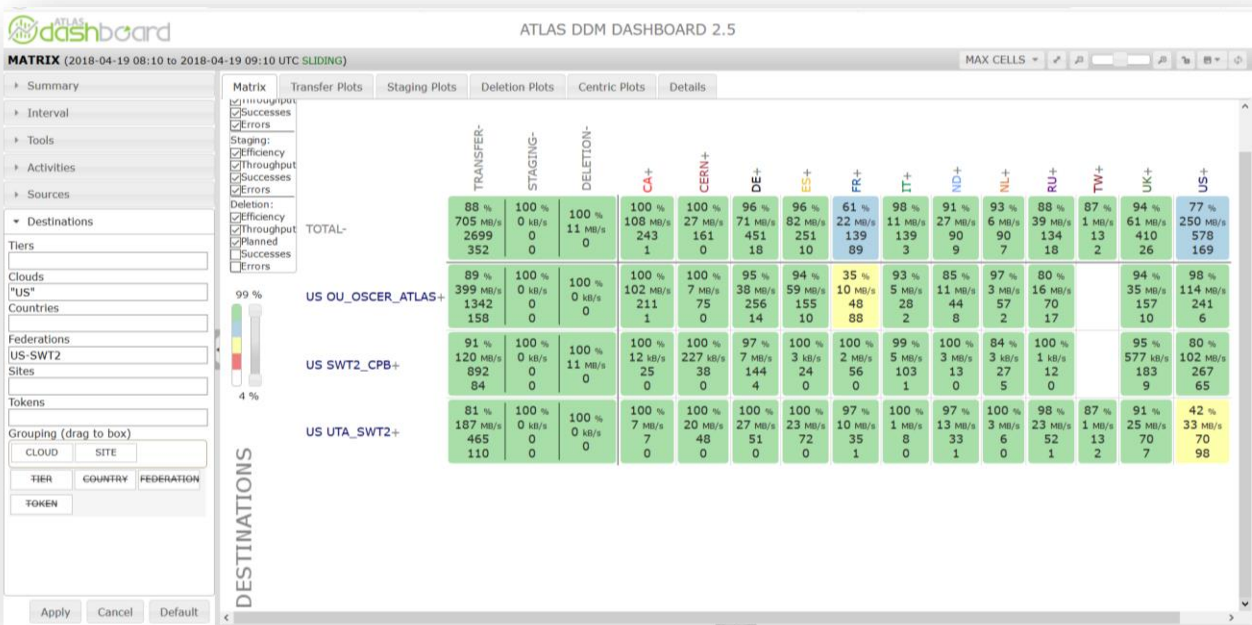


Figure 9-13 DDM Dashboard

Rucio monitors the transfer of files as well as failures in transfer. It calculates the percentage of failure compared to the number of files that were transferred successfully.If the percentage is alarmingly low (based on predetermined values) it alerts the concerned people at CERN.

The data that is transferred from clouds around the world to the SWT2 sites is monitored by Rucio by obtaining the data from the JSON:
http://dashb-atlas-ddm.cern.ch/dashboard/request.py/matrix.json?tool=rucio&activity=Analysis+Input&activity=Data+Brokering&activity=Data+Consolidation&activity=Data+Export+Test&activity=Data+Rebalancing&activity=Debug&activity=Deletion&activity=Express&activity=Functional+Test&activity=Group+Subscriptions&activity=Production&activity=Production+Input&activity=Production+Output&activity=Recovery&activity=SFO+to+EOS+export&activity=Staging&activity=T0+Export&activity=T0+Tape&activity=User+Subscriptions&activity=default&activity=on&activity=on&activity=rucio-integration&activity=test&activity=test%3AT0_T1+export&activity=test%3AT1_T2+export&activity=testactivity10&activity=testactivity20&activity=testactivity70&activity_default_exclude=Upload%2FDownload+(Job)&activity_default_exclude=Upload%2FDownload+(User)&activity_default_exclude=Analysis+Download&activity_default_exclude=Analysis+Upload&activity_default_exclude=Production+Download&activity_default_exclude=Production+Upload&activity_default_exclude=CLI+Download&activity_default_exclude=CLI+Upload&activity_default_exclude=Analysis+Logs+OS+Upload&activity_default_exclude=Analysis+Logs+Upload&activity_default_exclude=Event+Service+Download&activity_default_exclude=Event+Service+Upload&activity_default_exclude=Production+Logs+OS+Upload&activity_default_exclude=Production+Logs+Upload&src_grouping=cloud&dst_cloud=%22US%22&dst_federation=US-SWT2&dst_grouping=cloud&dst_grouping=site&interval=60&prettyprint

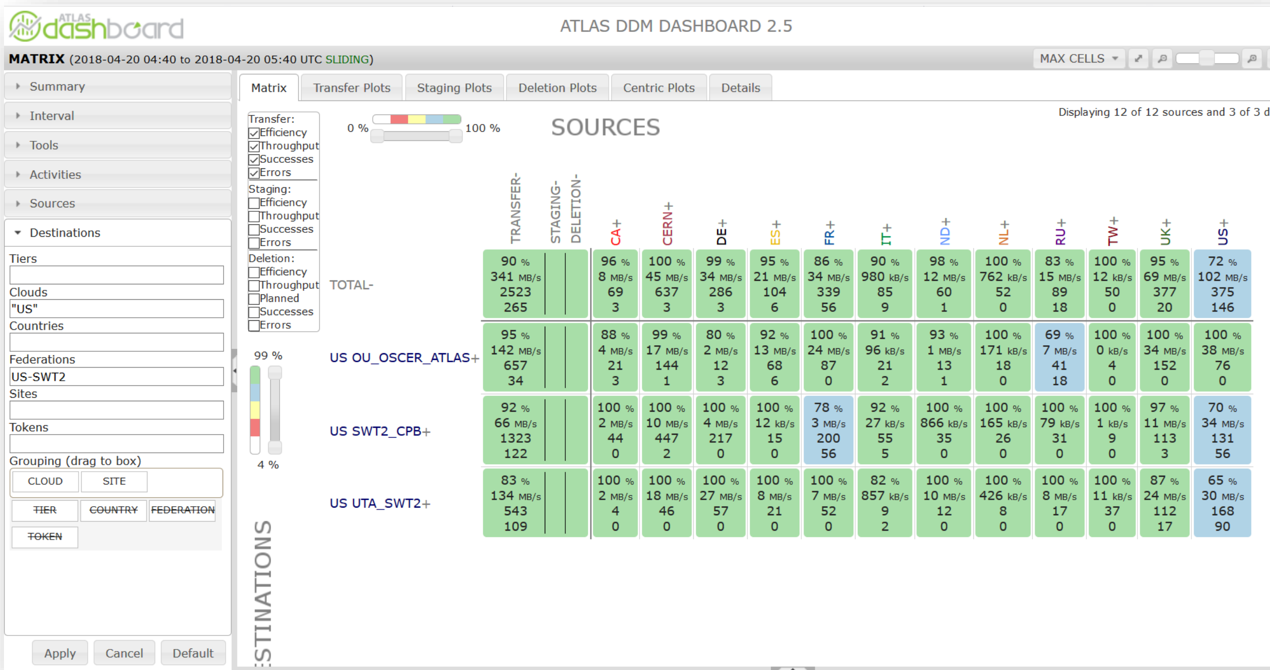


Figure 9-14 Pictorial representation of the data can be seen in the Rucio dashboard

The data that we obtain from the json refreshes every hour. Thus we can expect to get fresh set of data every hour. A script will extract the data from the json every hour and store the data is the database. We are particularly interested in the fields “errors_xs”,”files_xs” “bytes_xs”,“src_cloud”,“dst_site“ and “dst_cloud” which would essentially get interpreted to:

errors_xs : number of files that were scheduled to be transferred but failed to get transferred.

files_xs : number of files that were scheduled to be transferred and successfully got transferred.

bytes_xs : The number of bytes that were successfully transferred.

src_cloud : The source of data.

dst_site: The name of the destination RSE.

dst_cloud : The country where destination site is located.

For transfer other than the two SWT sites for UTA : SWT2_CPB and UTA_SWT2 , LUCILLE and OU_OSCER_ATLAS will be tracked.

The data we require for analysis is extracted from the above mentioned json every hour. It is then stored in the database table Transfer_Tables_UTA_Dest whose fields are:

errors_xs : number of files that failed to get transferred.

files_xs : number of files that successfully got transferred.

bytes_xs : The number of bytes that were successfully transferred.

src_cloud : The source cloud of data.

dst_site : The destination RSE.

error_prcnt : the percentage of files that failed to get transferred with respect to the total number of files scheduled for transfer between src_cloud and dst_site.

formula for calculation= [errors_xs / (errors_xs + files_xs) * 100] %

dst_cloud : The country where destination site is located.

As with deletions the files that are to be marked for transfer is decided by Rucio. Jobs are run on the RSEs for transfer of data from the source clouds to the destination RSE. Sometimes there are issues due to which these jobs cannot run till completion or due to other issues the transfer of data fails. We are interested in monitoring how much of the data that was planned for transferred is successfully getting transferred and how much of it is failing . If the amount of data that is failing to get transferred is more than the amount of data that is successfully getting transferred, then that is a reason for concern.

Based on a predetermined threshold value the following formula will monitor the performance of the system with respect to transfer of data.Here in the example the threshold value has been considered to be 10%. That will mean the system's performance will have to be investigated once the failure in file transfer exceeds 10% of total planned transfers

The formula to calculate the performance: [errors_xs / (errors_xs + files_xs) >10%]

To visualize the data the (number of sources)*(number of destinations) table is created. The chart will show the value of error_prcnt i.e. the percentage of failure in transfer of files

Transfer SWT2 Destination

	CA	CERN	DE	ES	FR	IT	ND	NL	RU	TW	UK	US
UTA_SWT2	0	0	100.0	100.0	100.0	80.0	100.0	100.0	100.0	100.0	38.46	89.47
OU_OSCER_ATLAS	50.0	0	100.0	0	85.71	85.71	100.0	100.0	100.0	100.0	0	100.0
SWT2_CPB	93.75	100.0	87.21	100.0	57.14	75.0	100.0	100.0	0	0	36.36	92.5

Figure 9-15 Transfer Dashboard : SWT2 Destination

Each of the cells in the chart gives us the percentage of successful file transfer calculated as per the data for the past hour as given by the equation:

(Number of files that successfully transferred / Total number of files that were scheduled to be transferred)*100

Each of these individual cell is clickable that will lead us to the details regarding the transfer of files between a pair of source cloud and destination site i.e say the source cloud 'CA' and destination site 'UTA_SWT2'. On click on a particular cell it will take us to the details page for the source cloud and destination site.

The Transfer Details page will give us four charts:

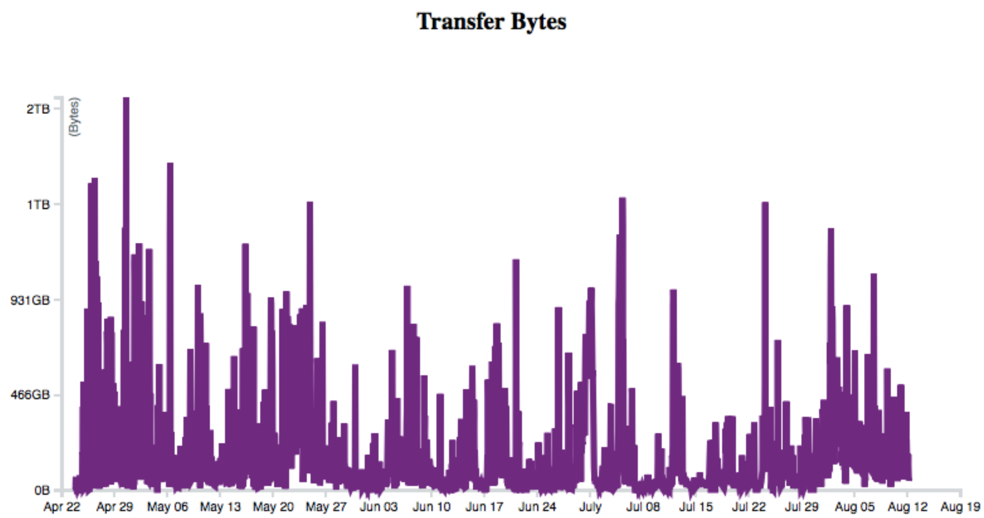


Figure 9-16 Transfer Bytes

Transfer bytes will track the number of bytes transferred over time. The time period can range from one day to last 365 days and can be adjusted from settings.

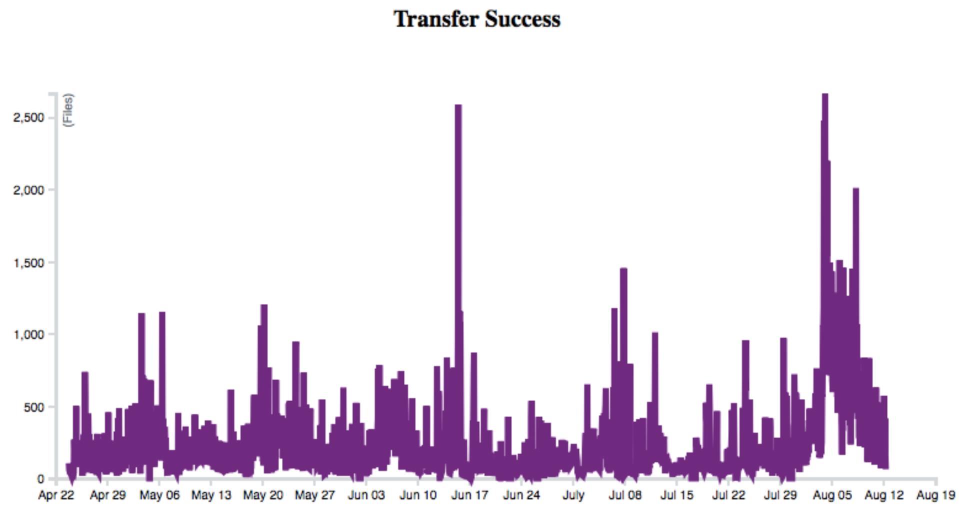


Figure 9-17 Transfer Success

Transfer Success chart tracks the successful transfer of files from source cloud to destination site over time.The y-axis gives the number of files whereas the x-axis tracks the time period which can range from one day to last 365 days.

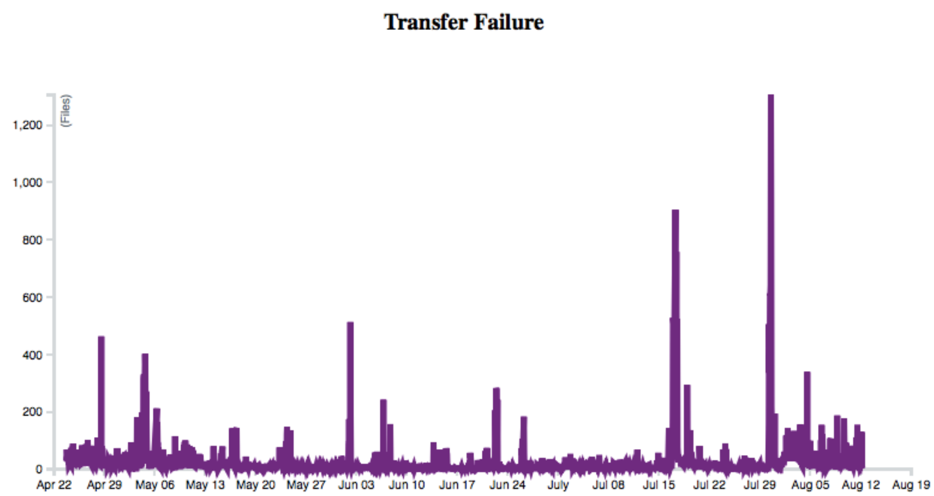


Figure 9-18 Transfer Failure

Transfer Failure chart tracks the failure of transfer of files from source cloud to destination site . The y-axis gives the number of files and the x-axis tracks the time period which can range from one day to last 365 days

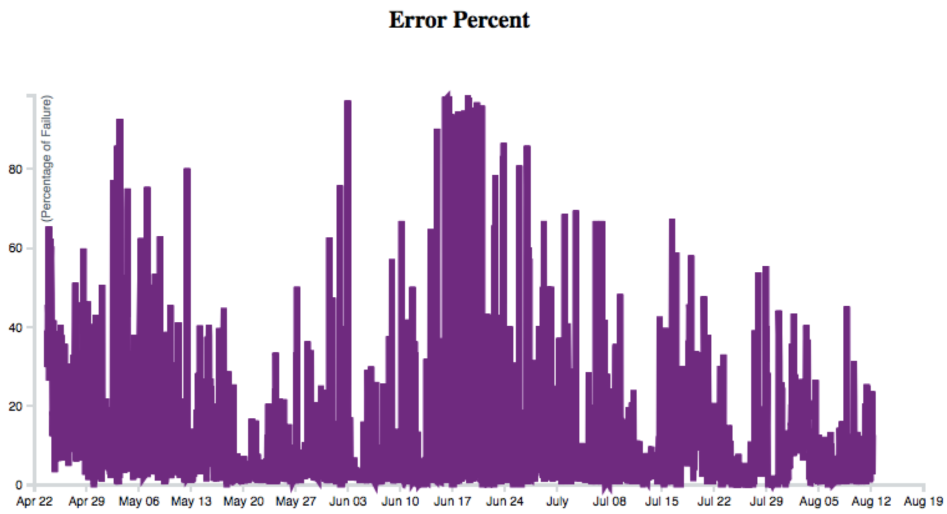


Figure 9-19 Error Percent

Error Percent tracks the percentage of failure of transfer of files from source cloud to destination site over time. The y-axis gives us the percentage of failure in transfer of files and x-axis tracks the time period which can range from one day to last 365 days.

Transfer of data where Southwest Tier 2 clusters are the source of data
The data to that is transferred to clouds around the world from the SWT2 sites is monitored by Rucio by obtaining the data from the JSON:

http://dashb-atlas-ddm.cern.ch/dashboard/request.py/matrix.json?tool=rucio&activity=Analysis+Input&activity=Data+Brokering&activity=Data+Consolidation&activity=Data+Export+Test&activity=Data+Rebalancing&activity=Debug&activity=Deletion&activity=Express&activity=Functional+Test&activity=Group+Subscriptions&activity=Production&activity=Production+Input&activity=Production+Output&activity=Recovery&activity=SFO+to+EOS+export&activity=Staging&activity=T0+Export&activity=T0+Tape&activity=User+Subscriptions&activity=default&activity=on&activity=on&activity=rucio-integration&activity=test&activity=test%3AT0_T1+export&activity=test%3AT1_T2+export&activity=testactivity10&activity=testactivity20&activity=testactivity70&activity_default_exclude=Upload%2FDownload+(Job)&activity_default_exclude=Upload%2FDownload+(User)&activity_default_exclude=Analysis+Download&activity_default_exclude=Analysis+Upload&activity_default_exclude=Produ

ction+Download&activity_default_exclude=Production+Upload&activity_default_exclude=CLI+Download&activity_default_exclude=CLI+Upload&activity_default_exclude=Analysis+Logs+OS+Upload&activity_default_exclude=Analysis+Logs+Upload&activity_default_exclude=Event+Service+Download&activity_default_exclude=Event+Service+Upload&activity_default_exclude=Production+Logs+OS+Upload&activity_default_exclude=Production+Logs+Upload&src_federation=US-SWT2&src_grouping=cloud&src_grouping=site&dst_grouping=cloud&interval=60&prettyprint

From the Transfer Dashboard if we choose the first option TransferSWT2Source we reach the Transfer SWT2 Source dashboard which shows up the percentage of transfer between the SWT2 source and destination cloud in a tabular format.

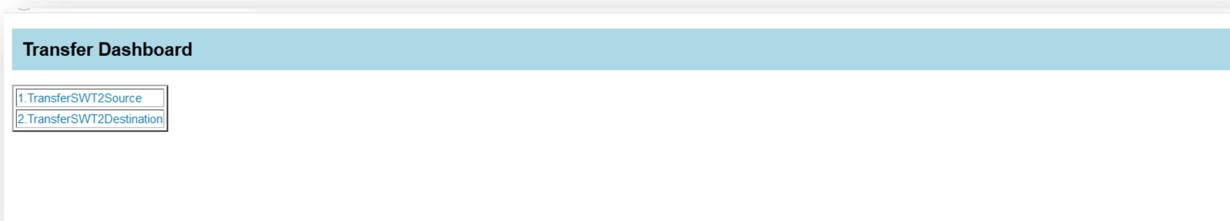
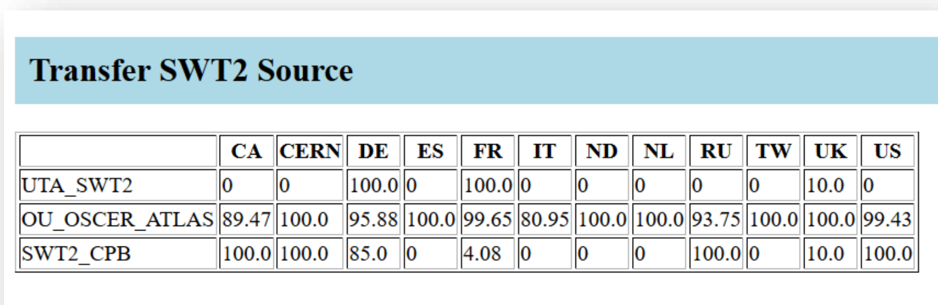


Figure 9-20 Transfer Dashboard


Transfer SWT2 Source

	CA	CERN	DE	ES	FR	IT	ND	NL	RU	TW	UK	US
UTA_SWT2	0	0	100.0	0	100.0	0	0	0	0	0	10.0	0
OU_OSCER_ATLAS	89.47	100.0	95.88	100.0	99.65	80.95	100.0	100.0	93.75	100.0	100.0	99.43
SWT2_CPB	100.0	100.0	85.0	0	4.08	0	0	0	100.0	0	10.0	100.0

Figure 9-21 Transfer Table SWT2 Source

The above table gives the percentage of successful transfer between a SWT2 source and destination clouds.
From the above table each cell can be chosen to view the transfer detail page.
The transfer detail page will contain four charts :

Transfer Bytes
Transfer Success
Transfer Failure
Transfer Percentage

Transfer Bytes

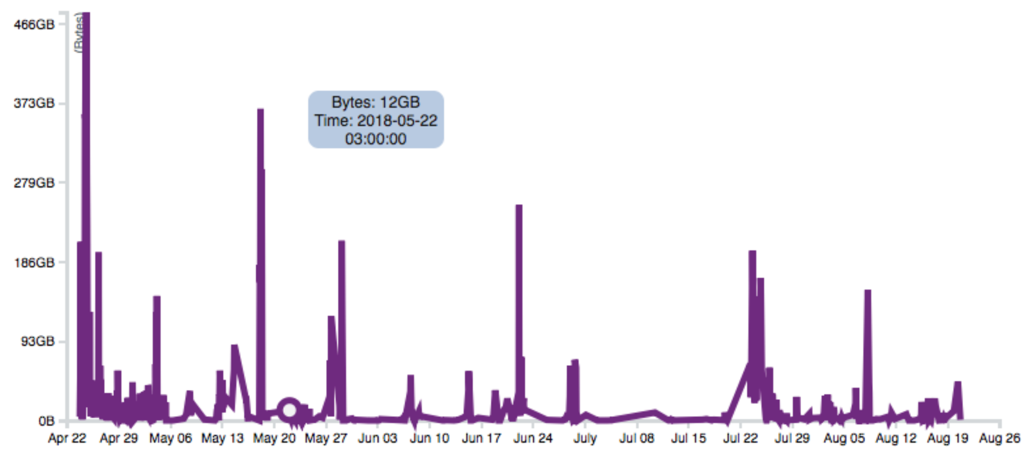


Figure 9-22 Transfer Bytes

Transfer Bytes will give us the amount of bytes transferred from source to destination cloud over time.

Transfer Success

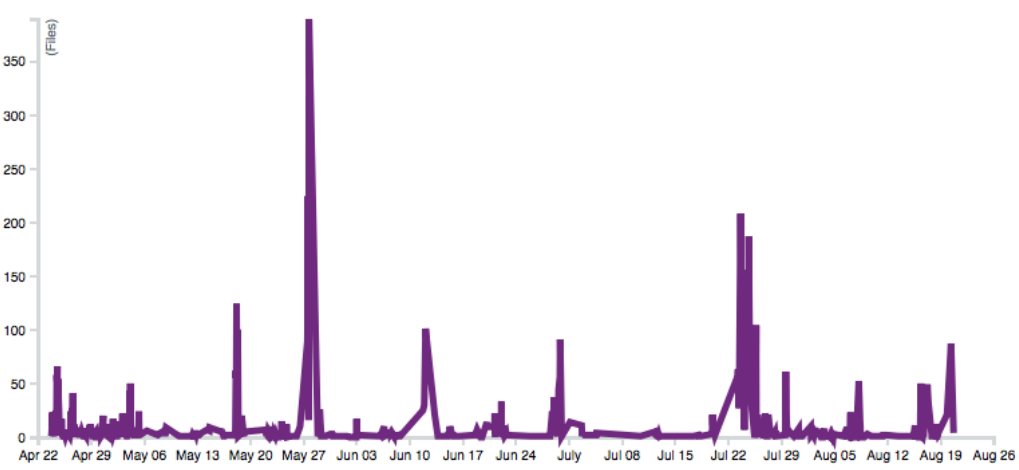


Figure 9-23 Transfer Success

Transfer Success will give us the number of files transferred successfully from source to destination.

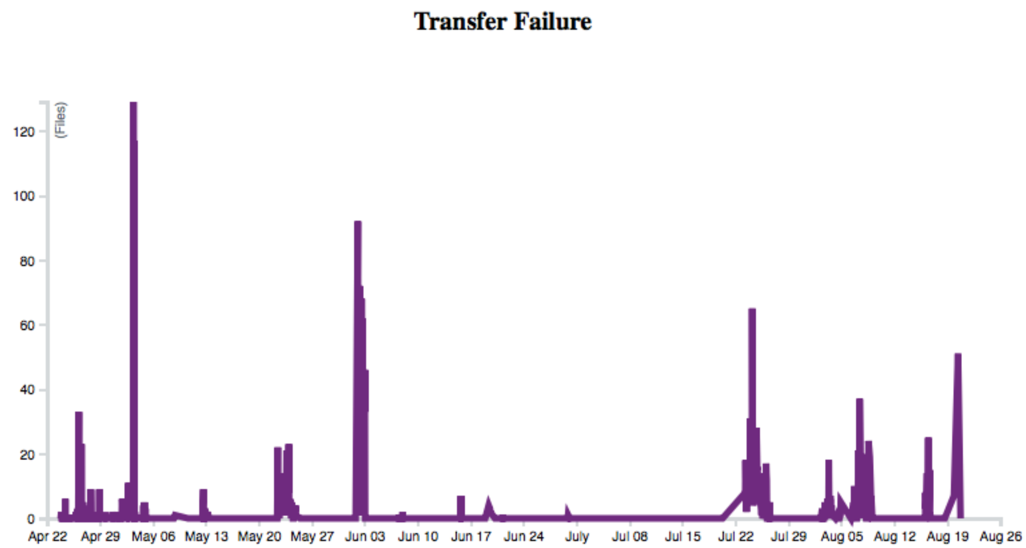


Figure 9-24 Transfer Failure

Transfer failure gives us the number of files that failed to get transferred from source to destination over a period of time.

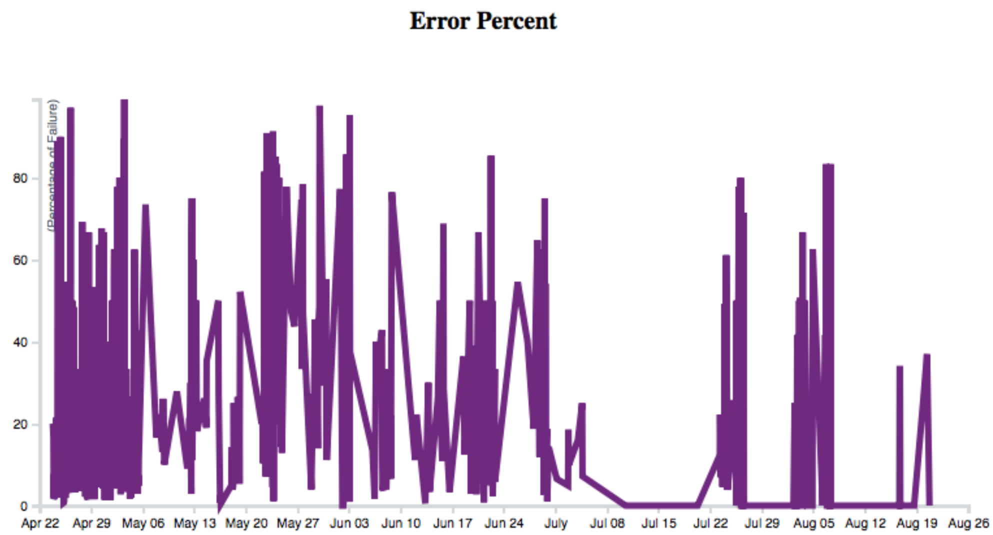


Figure 9-25 Error Percent

Error percentage chart gives us the percentage of files that failed to get transferred successfully though they were marked for transfer.

Chapter 10

Settings

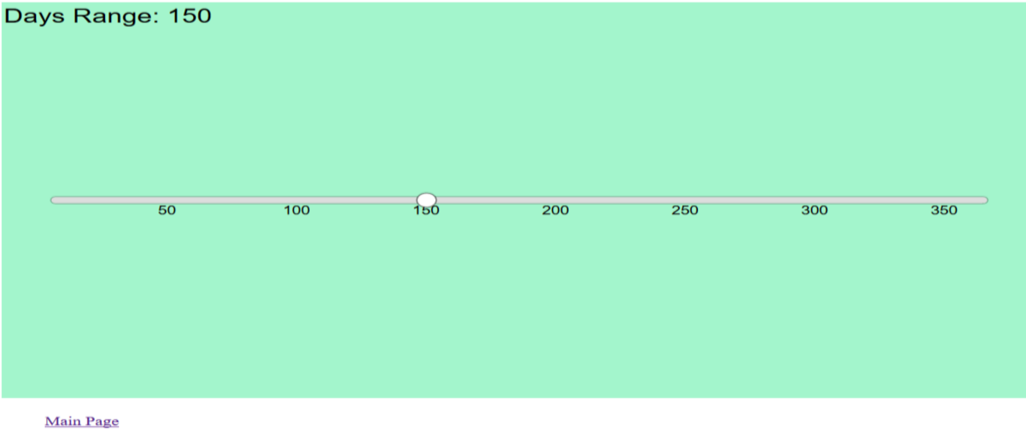


Figure 10-1 Settings

The settings page has been created to control range of days for which the user would like to see the comparative chart of all the storage parameters in the landing page of all monitoring dashboards as well as the details page of each parameter: rucio , json,expired,gsiftp, obsolete and unavailable.
Snapshot of SWT2_CPB Datadisk comparative chart. The slider can be dragged over the scale to change the number of days from 7 to 365.

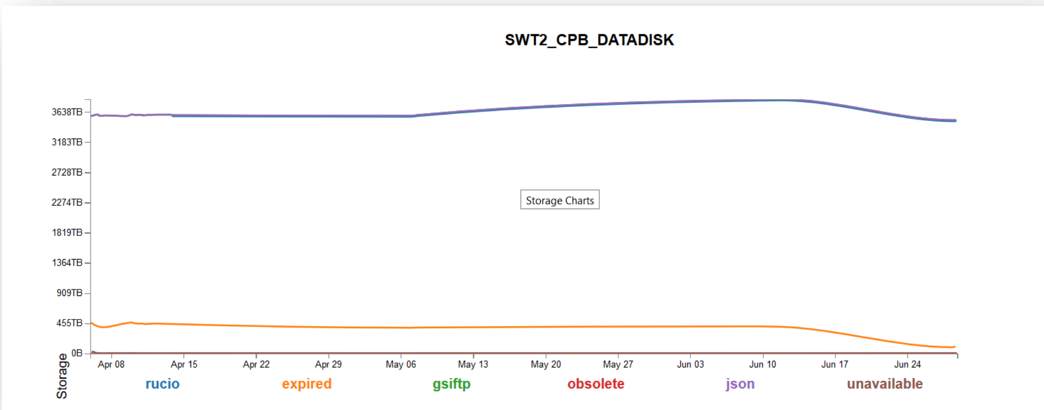


Figure 10-2 Comparison Chart

Similarly the change of range of days would affect all the details pages like Transfer details page and Deletion details page.

Chapter 11

Mail

The mailing system can support mailing to multiple people by just putting their name in the config file.

```
[mail]
toaddr: "antara.ray@mavs.uta.edu","mcguigan@uta.edu"
fromaddr: antara.ray@mavs.uta.edu
```

The code for the same can be found on page 78

Chapter 12

Deletion of data from the database

As we will maintain only a year worth of data in all the tables in the database the script deleteTableData.py will be scheduled to run on the first of every month to delete all the data from each of the tables that will be older than a year.

The code for the same can be found on page 79

Chapter 13

Conclusion

The system that has been built after my research on the monitoring needs of the clusters present in University of Texas at Arlington serving the Atlas experiment, will not only monitor the health of the data clusters but also help the people working on these clusters gain a deeper understanding about the system before any of the clusters start failing due to reasons varying from failure of deletion of old files from the system to overflow of data being stored in the system.

The system which will monitor the clusters as space tokens , the way in which the space at the clusters have been logically divided by CERN to store and process Atlas data, which will quickly point out the area that needs attention. Our approach towards acquiring this monitoring data was straight forward since early on we decided to use the data sources used by Rucio for monitoring. This is because these are secure clusters and thus anyone who does not have the required access cannot have any of the information regarding these clusters. Thus with the help of the people responsible for the maintenance of these clusters at UTA the data used by Rucio is being accessed for our monitoring system.

Next comes the storage of the data. We decided to use the date and time of the capture of each data point as our primary key against which all the data will be stored. We used the MongoDB for our database and created tables with respect to each space token which we are monitoring which are present in the UTA campus as well as tables are maintained for deletion and transfer of data at the two other data centers present at the university campuses of University of Oklahoma and Langston University.

Then we designed an interactive interface that will help in visualizing and analyzing the data. It will give a snapshot of the current condition of the clusters and will also give us a view of the cluster for last one year. It will also give the user the ability to drill down on the charts showing a year's data so as to have a deeper understanding about the cluster's performance over a past time frame. This will help in understanding past failures and equip the users to understand preventive measures to be taken before another failure happens in the future.

Lastly the scripts that will collect data every hour for the deletion/transfer jsons and every four hours for the storage monitoring json will also trigger mails informing the concerned authorities at UT Arlington if data levels at the clusters are beyond the safe threshold value or deletion and transfer of files are failing beyond acceptable percentages. On being aware of the situation of the cluster failure prevention measures can be taken more promptly much before any on the clusters go down completely. Also since the monitoring system will mainly serve the people here at UT Arlington it will give them the time to understand an issue in case an failure is waiting to happen instead of being informed of the issue once Rucio reports of it to CERN.

Thus though this monitoring system will run in parallel to Rucio it will give an advantage to the people here at UT Arlington to study the data collected by Rucio through various jobs being run on the local clusters and give a visual report about the clusters. Earlier monitoring of the clusters were only possible through the json files and snapshots of the clusters from Rucio. This interactive system will make the work on monitor the clusters comparatively easy and more understandable with minimum resources, reuse of existing resources and extensive use of powerful open source technologies.

Chapter 14

Future Work

Cluster monitoring can be a costly task requiring lots of resources and access to sensitive information and clusters. Anything done to make the task on monitoring easier even by a small amount can be of help. Our approach to reuse existing information and resources to come up with a scalable monitoring solution can help in monitoring data centers independently and not be completely dependent on remote monitoring.

The places where our monitoring solution would need more work is:

1. Complete independence from hard coded threshold values : Currently our solution has all the threshold values hard coded in the following scripts:

   fetchData_Deletion
   fetchData_Storage
   fetchData_Transfer_UTA_Src
   fetchData_Transfer_UTA_Dest

   The threshold values determine when a mail would be triggered. Thus if any change of these threshold values is required currently these above mentioned scripts have to be updated with the new threshold value. To make our solution completely data independent these values either need to be set in the config file or in the database from where it can be easily accessible to the code and can be easily updated as well.
2. The comparative study chart can be enhanced with a drill down feature :

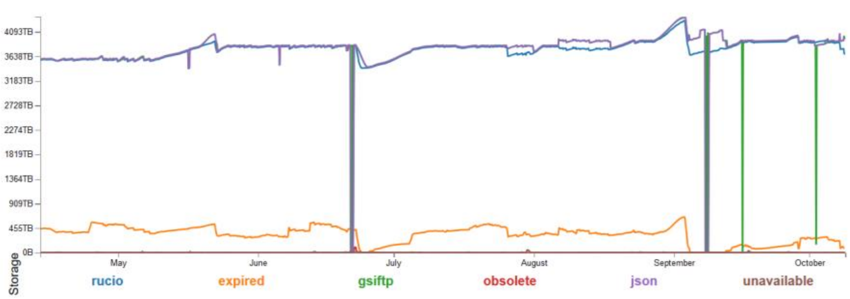


Figure 14-1 Comparison Chart

Currently the comparative chart can be used to see the difference in the values of the data points being monitored over a given period. This period of time can be altered using the Settings page : 7 days to 365 days. The individual lines can also be turned off by clicking on the name on the parameter (e.g. rucio, expired etc) to get a resulting chart with required parameters. It will be a good feature to allow user to drill down on this chart without having to set it in the Setting page. An auto drill down feature would enhance the usability of the chart.

Code for mailing system

sendMail.py

```
class sendMail:
 def mail(self,n):
    fromaddr =Config.get("mail", "fromaddr")
    toaddr=Config.get("mail", "toaddr")
    msg = MIMEMultipart()
    msg[ 'From' ] = fromaddr
    msg[ 'To' ] = toaddr
    msg[ 'Subject' ] = "Error"
    body = str(n)
    msg.attach(MIMEText(body, 'plain'))
    server=smtplib.SMTP('localhost')
    text = msg.as_string()
    server.sendmail(msg[ 'From' ],msg[ 'To' ].split(","), text)
    server.quit()
```

Code for table data deletion

```
client = MongoClient('mongodb://axr6305:secretPassword@localhost:27017/mydata')
db=client['mydata']
x = datetime.now() - relativedelta(years=1)


##############################Deletion#################################
db.del_SWT2_CPB.find({}).sort("to_date", pymongo.ASCENDING)
db.del_SWT2_CPB.delete_many({"to_date":{"$lte": x}})


db.del_UTA_SWT2.find({}).sort("to_date", pymongo.ASCENDING)
db.del_UTA_SWT2.delete_many({"to_date":{"$lte": x}})

db.del_OU_OSCER_ATLAS.find({}).sort("to_date", pymongo.ASCENDING)
db.del_OU_OSCER_ATLAS.delete_many({"to_date":{"$lte": x}})

db.del_LUCILLE.find({}).sort("to_date", pymongo.ASCENDING)
db.del_LUCILLE.delete_many({"to_date":{"$lte": x}})


####################################Storage########################

#rucio_SWT2_CPB_DATADISK


db.rucio_SWT2_CPB_DATADISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.rucio_SWT2_CPB_DATADISK.delete_many({"updated_at":{"$lte": x}})

#json_SWT2_CPB_DATADISK


db.json_SWT2_CPB_DATADISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.json_SWT2_CPB_DATADISK.delete_many({"updated_at":{"$lte": x}})

#expired_SWT2_CPB_DATADISK


db.expired_SWT2_CPB_DATADISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.expired_SWT2_CPB_DATADISK.delete_many({"updated_at":{"$lte": x}})

#unavailable_SWT2_CPB_DATADISK
```

```
db.unavailable_SWT2_CPB_DATADISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.unavailable_SWT2_CPB_DATADISK.delete_many({"updated_at":{"$lte": x}})

#gsiftp_SWT2_CPB_DATADISK


db.gsiftp_SWT2_CPB_DATADISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.gsiftp_SWT2_CPB_DATADISK.delete_many({"updated_at":{"$lte": x}})

#obsolete_SWT2_CPB_DATADISK


db.obsolete_SWT2_CPB_DATADISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.obsolete_SWT2_CPB_DATADISK.delete_many({"updated_at":{"$lte": x}})

#rucio_SWT2_CPB_LOCALGROUPDISK


db.rucio_SWT2_CPB_LOCALGROUPDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.rucio_SWT2_CPB_LOCALGROUPDISK.delete_many({"updated_at":{"$lte": x}})

#json_SWT2_CPB_LOCALGROUPDISK


db.json_SWT2_CPB_LOCALGROUPDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.json_SWT2_CPB_LOCALGROUPDISK.delete_many({"updated_at":{"$lte": x}})


#expired_SWT2_CPB_LOCALGROUPDISK


db.expired_SWT2_CPB_LOCALGROUPDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.expired_SWT2_CPB_LOCALGROUPDISK.delete_many({"updated_at":{"$lte": x}})

#unavailable_SWT2_CPB_LOCALGROUPDISK


db.unavailable_SWT2_CPB_LOCALGROUPDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.unavailable_SWT2_CPB_LOCALGROUPDISK.delete_many({"updated_at":{"$lte": x}})

#gsiftp_SWT2_CPB_LOCALGROUPDISK
```

```
db.gsiftp_SWT2_CPB_LOCALGROUPDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.gsiftp_SWT2_CPB_LOCALGROUPDISK.delete_many({"updated_at":{"$lte": x}})

#obsolete_SWT2_CPB_LOCALGROUPDISK


db.obsolete_SWT2_CPB_LOCALGROUPDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.obsolete_SWT2_CPB_LOCALGROUPDISK.delete_many({"updated_at":{"$lte": x}})

#rucio_SWT2_CPB_SCRATCHDISK


db.rucio_SWT2_CPB_SCRATCHDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.rucio_SWT2_CPB_SCRATCHDISK.delete_many({"updated_at":{"$lte": x}})

#json_SWT2_CPB_SCRATCHDISK


db.json_SWT2_CPB_SCRATCHDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.json_SWT2_CPB_SCRATCHDISK.delete_many({"updated_at":{"$lte": x}})

#expired_SWT2_CPB_SCRATCHDISK


db.expired_SWT2_CPB_SCRATCHDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.expired_SWT2_CPB_SCRATCHDISK.delete_many({"updated_at":{"$lte": x}})

#unavailable_SWT2_CPB_SCRATCHDISK


db.unavailable_SWT2_CPB_SCRATCHDISK.find({}).sort("to_date", pymongo.ASCENDING)
db.unavailable_SWT2_CPB_SCRATCHDISK.delete_many({"to_date":{"$lte": x}})

#gsiftp_SWT2_CPB_SCRATCHDISK


db.gsiftp_SWT2_CPB_SCRATCHDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.gsiftp_SWT2_CPB_SCRATCHDISK.delete_many({"updated_at":{"$lte": x}})

#obsolete_SWT2_CPB_SCRATCHDISK


db.obsolete_SWT2_CPB_SCRATCHDISK.find({}).sort("updated_at", pymongo.ASCENDING)
db.obsolete_SWT2_CPB_SCRATCHDISK.delete_many({"updated_at":{"$lte": x}})
```

```
#rucio_UTA_SWT2


db.rucio_UTA_SWT2.find({}).sort("updated_at", pymongo.ASCENDING)
db.rucio_UTA_SWT2.delete_many({"updated_at":{"$lte": x}})

#json_UTA_SWT2


db.json_UTA_SWT2.find({}).sort("updated_at", pymongo.ASCENDING)
db.json_UTA_SWT2.delete_many({"updated_at":{"$lte": x}})

#expired_UTA_SWT2


db.expired_UTA_SWT2.find({}).sort("updated_at", pymongo.ASCENDING)
db.expired_UTA_SWT2.delete_many({"updated_at":{"$lte": x}})

#unavailable_UTA_SWT2


db.unavailable_UTA_SWT2.find({}).sort("to_date", pymongo.ASCENDING)
db.unavailable_UTA_SWT2.delete_many({"to_date":{"$lte": x}})

#gsiftp_UTA_SWT2


db.gsiftp_UTA_SWT2.find({}).sort("to_date", pymongo.ASCENDING)
db.gsiftp_UTA_SWT2.delete_many({"to_date":{"$lte": x}})

#obsolete_UTA_SWT2


db.obsolete_UTA_SWT2.find({}).sort("updated_at", pymongo.ASCENDING)
db.obsolete_UTA_SWT2.delete_many({"updated_at":{"$lte": x}})


########################################Transfer#############################################

db.Transfer_Tables_UTA_Dest.find({}).sort("to_date", pymongo.ASCENDING)
db.Transfer_Tables_UTA_Dest.delete_many({"to_date":{"$lte": x}})


db.Transfer_Tables_UTA_Src.find({}).sort("to_date", pymongo.ASCENDING)
db.Transfer_Tables_UTA_Src.delete_many({"to_date":{"$lte": x}})
```

References

[1] CERN https://home.cern/about

[2] ATLAS SWT2 http://www.atlas-swt2.org/

[3] what is atlas? http://atlasexperiment.org/what_is_atlas.html

[4] ATLAS experiment https://en.wikipedia.org/wiki/ATLAS_experiment

[5] The Worldwide LHC Computing Grid
https://home.cern/about/computing/worldwide-lhc-computing-grid

[6] Rucio, the next-generation Data Management system in ATLAS
https://www.sciencedirect.com/science/article/pii/S2405601415006409

[7] USATLASSWT2SITE REPORT APRIL 2016
https://indico.cern.ch/event/466991/contributions/1143586/attachments/1259191/1860158/SWT2-hepix.pdf

[8] What is the ATLAS Experiment?
http://www.singularitysymposium.com/atlas-experiment.html

[9] MongoDB Inc. https://www.mongodb.com/

[10] Python Software Foundation https://www.python.org/

[11] D3JS https://d3js.org/

[12] LHCOPN lhcopn.cern.ch

[13] The Large Hadron Collider https://home.cern/topics/large-hadron-collider

[14] DDM TABLE view for our sites as a destination and Deletions (Last Hour)
http://dashb-atlas-ddm.cern.ch/ddm2/#activity=(Analysis+Input,Data+Brokering,Data+Consolidation,Data+Export+Test,Data+Rebalancing,Debug,Deletion,Express,Functional+Test,Group+Subscriptions,Production,Production+Input,Production+Output,Recovery,SFO+to+EOS+export,Staging,T0+Export,T0+Tape,User+Subscriptions,default,on,on,rucio-integration,test,test%3AT0_T1+export,test%3AT1_T2+export,testactivity10,testactivity20,testactivity70)&date.interval=60&dst.cloud=(%22US%22)&dst.federation=(US-SWT2)&grouping.dst=(cloud,site)&m.content=(d_dof,d_dot,d_eff,d_faf,d_plf,s_eff,s_thr,t_eff,t_err,t_suc,t_thr)

[15] DDM JSON URL for our sites as a destination AND deletions (LAST HOUR)
http://dashb-atlas-
ddm.cern.ch/dashboard/request.py/matrix.json?tool=rucio&activity=Analysis+Input&acti
vity=Data+Brokering&activity=Data+Consolidation&activity=Data+Export+Test&activi
ty=Data+Rebalancing&activity=Debug&activity=Deletion&activity=Express&activity=F
unctional+Test&activity=Group+Subscriptions&activity=Production&activity=Productio
n+Input&activity=Production+Output&activity=Recovery&activity=SFO+to+EOS+expo
rt&activity=Staging&activity=T0+Export&activity=T0+Tape&activity=User+Subscripti
ons&activity=default&activity=on&activity=on&activity=rucio-
integration&activity=test&activity=test%3AT0_T1+export&activity=test%3AT1_T2+ex
port&activity=testactivity10&activity=testactivity20&activity=testactivity70&activity_d
efault_exclude=Upload%2FDownload+(Job)&activity_default_exclude=Upload%2FDo
wnload+(User)&activity_default_exclude=Analysis+Download&activity_default_exclud
e=Analysis+Upload&activity_default_exclude=Production+Download&activity_default_
exclude=Production+Upload&activity_default_exclude=CLI+Download&activity_defaul
t_exclude=CLI+Upload&activity_default_exclude=Analysis+Logs+OS+Upload&activity
_default_exclude=Analysis+Logs+Upload&activity_default_exclude=Event+Service+Do
wnload&activity_default_exclude=Event+Service+Upload&activity_default_exclude=Pr
oduction+Logs+OS+Upload&activity_default_exclude=Production+Logs+Upload&src_
grouping=cloud&dst_cloud=%22US%22&dst_federation=US-
SWT2&dst_grouping=cloud&dst_grouping=site&interval=60&prettyprint

[16] DDM TABLE view for our sites as a source of transfers (LAST HOUR)
http://dashb-atlas-
ddm.cern.ch/ddm2/#activity=(Analysis+Input,Data+Brokering,Data+Consolidation,Data
+Export+Test,Data+Rebalancing,Debug,Deletion,Express,Functional+Test,Group+Subsc
riptions,Production,Production+Input,Production+Output,Recovery,SFO+to+EOS+expor
t,Staging,T0+Export,T0+Tape,User+Subscriptions,default,on,on,rucio-
integration,test,test%3AT0_T1+export,test%3AT1_T2+export,testactivity10,testactivity2
0,testactivity70)&date.interval=60&grouping.src=(cloud,site)&m.content=(d_dot,d_eff,s
_eff,s_thr,t_eff,t_err,t_suc,t_thr)&src.federation=(US-SWT2)

[17]DDM JSON URL for our sites as a source of transfers (LAST HOUR)
http://dashb-atlas-
ddm.cern.ch/dashboard/request.py/matrix.json?tool=rucio&activity=Analysis+Input&acti
vity=Data+Brokering&activity=Data+Consolidation&activity=Data+Export+Test&activi
ty=Data+Rebalancing&activity=Debug&activity=Deletion&activity=Express&activity=F
unctional+Test&activity=Group+Subscriptions&activity=Production&activity=Productio
n+Input&activity=Production+Output&activity=Recovery&activity=SFO+to+EOS+expo
rt&activity=Staging&activity=T0+Export&activity=T0+Tape&activity=User+Subscripti

ons&activity=default&activity=on&activity=on&activity=rucio-integration&activity=test&activity=test%3AT0_T1+export&activity=test%3AT1_T2+export&activity=testactivity10&activity=testactivity20&activity=testactivity70&activity_default_exclude=Upload%2FDownload+(Job)&activity_default_exclude=Upload%2FDownload+(User)&activity_default_exclude=Analysis+Download&activity_default_exclude=Analysis+Upload&activity_default_exclude=Production+Download&activity_default_exclude=Production+Upload&activity_default_exclude=CLI+Download&activity_default_exclude=CLI+Upload&activity_default_exclude=Analysis+Logs+OS+Upload&activity_default_exclude=Analysis+Logs+Upload&activity_default_exclude=Event+Service+Download&activity_default_exclude=Event+Service+Upload&activity_default_exclude=Production+Logs+OS+Upload&activity_default_exclude=Production+Logs+Upload&src_federation=US-SWT2&src_grouping=cloud&src_grouping=site&dst_grouping=cloud&interval=60&prettyprint

[18] MongoDB https://en.wikipedia.org/wiki/MongoDB

[19] USATLASSWT2SITE REPORT OCTOBER 2017
https://indico.cern.ch/event/637013/contributions/2739297/attachments/1541617/2417789/SWT2-hepix-2017.pdf

Biographical Information

Antara Ray joined the University of Texas at Arlington in fall 2016. She completed her B.Tech in Information Technology from Heritage Institute of Technology, Kolkata in 2006.

She worked as a software engineer for HCL Technologies in India for 4 years, Cognizant Technology Solutions in India for 1 year and Tata Consultancy Services in India for 3 years.

Her research interests are in areas of data analysis and software development.