EOS Monitoring and Analytics Tools

August 2014

Author:
Hana Wurzelova

Supervisor:
Luca Mascetti

CERN openlab Summer Student Report 2014

Project Specification

The IT DSS group at CERN runs and evaluates innovative cloud storage technologies for their application to big data problems in high-energy physics research. One of the main storage systems is EOS, a multi-petabyte disk storage built from commodity hardware heavily used by LHC and non-LHC experiments, primarily for physics data.

In the scope of a common monitoring framework for our cloud storage services and in order to improve the analytics, the manageability and the user interface, this project's aim was to:

- implement a probing system to check storage health and main KPI
- implement features for the web visualization of storage KPIs
- investigate the integration of other monitoring within the framework
- compare different solution of storage monitoring and analytic tools

Abstract

The aim of this openlab project was to improve user experience of EOS, a disk based data storage system used by LHC and non-LHC users. The online monitoring systems for EOS are various and this project main focus was on improving EOS Cockpit structure and functionalities and simplifying SLS probing system. The EOS Cockpit structure was changes according to users' requirements, and new features were implemented. The SLS probing script was significantly simplified and adapted to the next generation tools used for displaying service levels.

Thank You Note

I would like to express my gratitude to my supervisor Luca Mascetti, who supported me and always found time for me in his busy schedule. I also want to thank to Jan Iven who helped me when Luca had other commitments. And I would like to thank you all, my great colleagues and friends I met at CERN during my openlab summer. I am grateful for how much I learned from all of you.

Table of Contents

1 Introduction

The IT DSS group at CERN runs and evaluates innovative cloud storage technologies for their application to big data problems in high-energy physics research. One of the main storage systems is EOS, a multi-petabyte disk storage built from commodity hardware heavily used by LHC and non-LHC experiments, primarily for physics data.

This project’s main aim is to improve the analytics, the manageability and the user interface of the current monitoring framework for our cloud storage services.

1.1 EOS

EOS is a storage software solution to manage multi-PB data storage. The system is running on commodity hardware with disks and is heavily used by LHC and non-LHC experiments, primarily for physics data. For more information, go to http://eos.cern.ch.

2 EOS Cockpit

EOS Cockpit is an online web monitoring tool for EOS. It is used mainly by physicists from the LHC experiments. The information displayed by EOS Cockpit includes quotas set for specific users and groups within the experiments and also how much space is actually used or the number of files stored.

This way, the users or groups responsible can easily see visualized data on how much space and files they have used in EOS. The quota structure is occasionally more complicated due to the different kind of quotas present (users and groups) and for some nested structure inside the namespace tree.

2.1 EOS Cockpit Structure

EOS Cockpit displays data in three different ways: in a tree map, organizational chart (basically a tree structure), and a table. The old website structure included one html file for each of these charts for each experiment.

The tree map and table web pages then relied on javascript for displaying the information inside the tree. This was inconvenient for the users who needed to access directly their own quota and storage usage information on a daily basis. They had to click through the sometimes complicated tree structure before being able to access their data.

One of my tasks was to rewrite the original python script generating the website. The inputs for this script are raw files with quotas for all the users and groups using EOS. The script needs to: parse the quota information, process the data, create corresponding html files, insert the data in there, and display the data in a specific way.

I reorganized the templates to simplify the process of changing the scripts, which I also reorganized, edited and added new ones. I then changed the python script to generate new folders and separate html files for each quota node in the tree structure. This way, we can provide users with a simple link to the data they want to see.

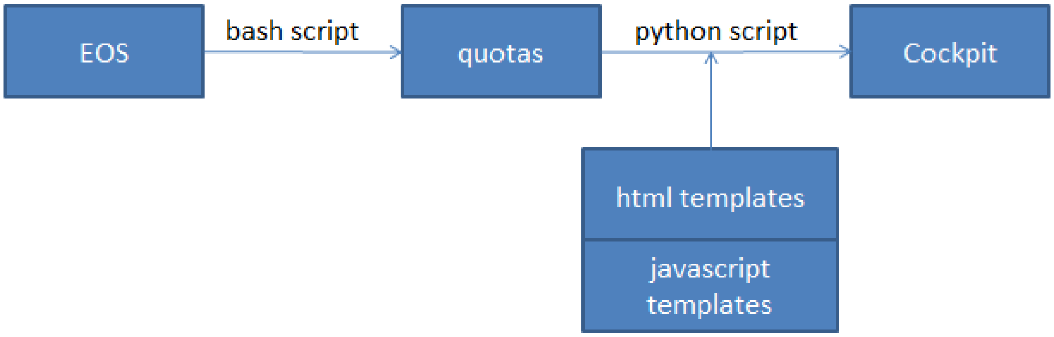
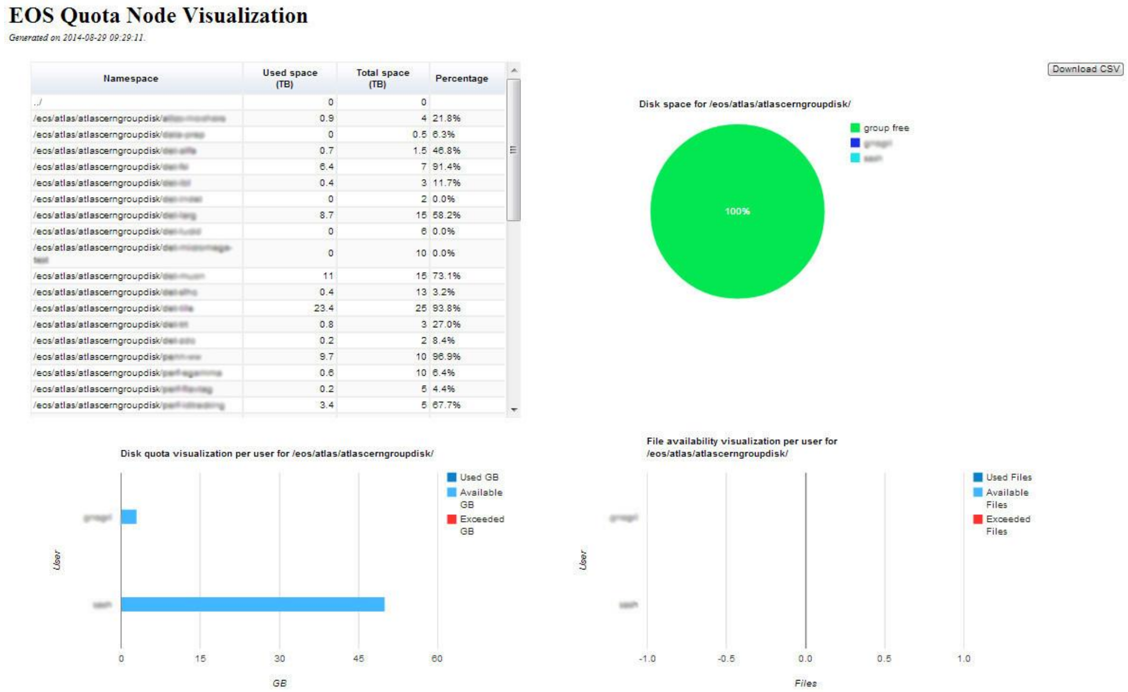


Figure 1: Quotas processing from EOS to EOS Cockpit



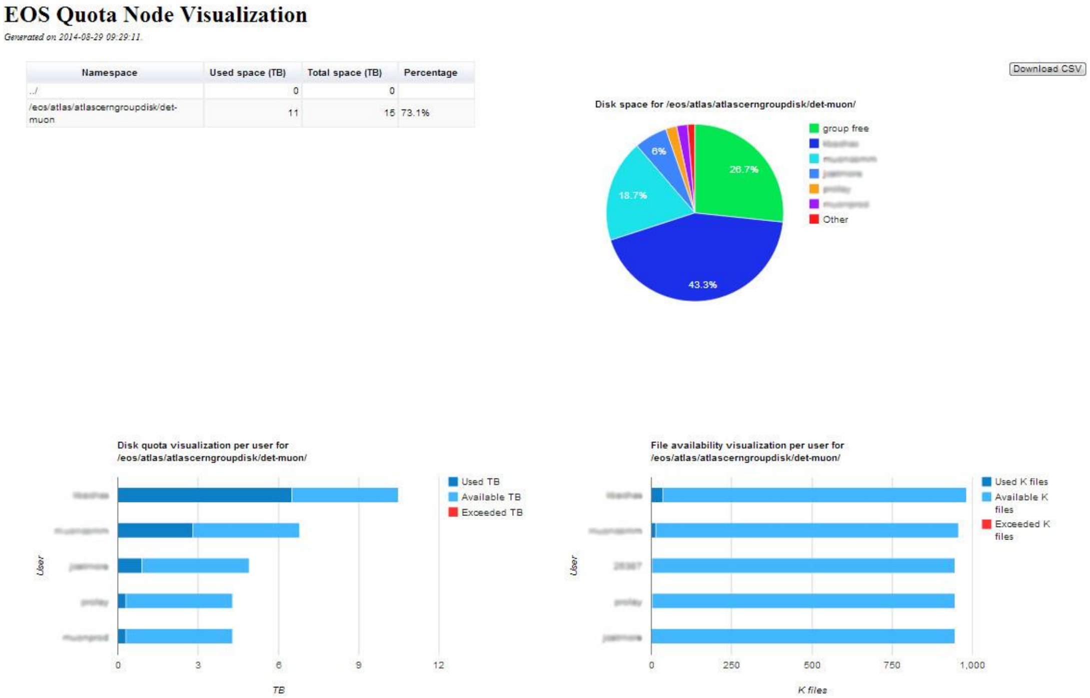
EOS Quota Node Visualization

Generated on 2014-08-29 09:29:11.

Namespace	Used space (TB)	Total space (TB)	Percentage
../	0	0	
/eos/atlas/atlascerngroupdisk/[redacted]	0.9	4	21.8%
/eos/atlas/atlascerngroupdisk/[redacted]	0	0.5	6.3%
/eos/atlas/atlascerngroupdisk/[redacted]	0.7	1.5	46.8%
/eos/atlas/atlascerngroupdisk/[redacted]	6.4	7	91.4%
/eos/atlas/atlascerngroupdisk/[redacted]	0.4	3	11.7%
/eos/atlas/atlascerngroupdisk/[redacted]	0	2	0.0%
/eos/atlas/atlascerngroupdisk/[redacted]	8.7	15	58.2%
/eos/atlas/atlascerngroupdisk/[redacted]	0	6	0.0%
/eos/atlas/atlascerngroupdisk/[redacted]	0	10	0.0%
/eos/atlas/atlascerngroupdisk/[redacted]	11	15	73.1%
/eos/atlas/atlascerngroupdisk/[redacted]	0.4	13	3.2%
/eos/atlas/atlascerngroupdisk/[redacted]	23.4	25	93.8%
/eos/atlas/atlascerngroupdisk/[redacted]	0.8	3	27.0%
/eos/atlas/atlascerngroupdisk/[redacted]	0.2	2	8.4%
/eos/atlas/atlascerngroupdisk/[redacted]	9.7	10	96.9%
/eos/atlas/atlascerngroupdisk/[redacted]	0.6	10	6.4%
/eos/atlas/atlascerngroupdisk/[redacted]	0.2	5	4.4%
/eos/atlas/atlascerngroupdisk/[redacted]	3.4	5	67.7%

Download CSV
Disk space for /eos/atlas/atlascerngroupdisk/
group free
100%
Disk quota visualization per user for /eos/atlas/atlascerngroupdisk/
Used GB
Available GB
Exceeded GB
User
0
15
30
45
60
GB
File availability visualization per user for /eos/atlas/atlascerngroupdisk/
Used Files
Available Files
Exceeded Files
User
-1.0
-0.5
0.0
0.5
1.0
Files

Figure 2: EOS Cockpit table view for node: /eosatlas/eos/atlas.html



EOS Quota Node Visualization

Generated on 2014-08-29 09:29:11.

Namespace	Used space (TB)	Total space (TB)	Percentage
../	0	0	
/eos/atlas/atlascerngroupdisk/det-muon	11	15	73.1%

Download CSV
Disk space for /eos/atlas/atlascerngroupdisk/det-muon/
group free
Other
26.7%
43.3%
18.7%
6%
Disk quota visualization per user for /eos/atlas/atlascerngroupdisk/det-muon/
Used TB
Available TB
Exceeded TB
User
0
3
6
9
12
TB
File availability visualization per user for /eos/atlas/atlascerngroupdisk/det-muon/
Used K files
Available K files
Exceeded K files
User
0
250
500
750
1,000
K files

Figure 3: EOS Cockpit table view for node: /eosatlas/eos/atlas/atlascerngroupdisk/det-muon.html

2.2 Historical Data Display

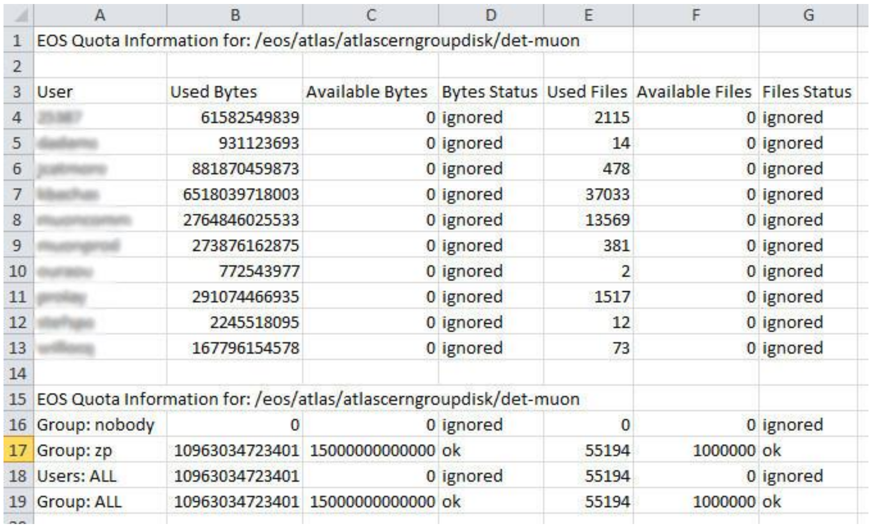
My advisor and I originally thought we would implement a historical data display into EOS Cockpit. After I created a demo and we discussed the idea with users from the LHC experiments and we concluded that we would rather implement a data export feature instead. This is described in the following section.

2.3 Standardized Data Export

The EOS users access the monitoring interface for two reasons. The first one is to actually check the storage used by the group or user at the moment. The second is to provide overview information about EOS and data storage for management and planning purposes. For this reason, our team was asked to enable easy export of the quota information provided by EOS Cockpit.

The data are now exported into csv format by a simple button click on the Cockpit website, using javascript. The table includes quotas and used space and number of files information for the specific node the user is viewing. There is also aggregated information on the whole group.

The structure of the export has changed a few times and the final format is as shown in Figure 4 below.



	A	B	C	D	E	F	G
1	EOS Quota Information for: /eos/atlas/atlascerngroupdisk/det-muon						
2							
3	User	Used Bytes	Available Bytes	Bytes Status	Used Files	Available Files	Files Status
4	[redacted]	61582549839	0	ignored	2115	0	ignored
5	[redacted]	931123693	0	ignored	14	0	ignored
6	[redacted]	881870459873	0	ignored	478	0	ignored
7	[redacted]	6518039718003	0	ignored	37033	0	ignored
8	[redacted]	2764846025533	0	ignored	13569	0	ignored
9	[redacted]	273876162875	0	ignored	381	0	ignored
10	[redacted]	772543977	0	ignored	2	0	ignored
11	[redacted]	291074466935	0	ignored	1517	0	ignored
12	[redacted]	2245518095	0	ignored	12	0	ignored
13	[redacted]	167796154578	0	ignored	73	0	ignored
14							
15	EOS Quota Information for: /eos/atlas/atlascerngroupdisk/det-muon						
16	Group: nobody	0	0	ignored	0	0	ignored
17	Group: zp	10963034723401	15000000000000	ok	55194	1000000	ok
18	Users: ALL	10963034723401	0	ignored	55194	0	ignored
19	Group: ALL	10963034723401	15000000000000	ok	55194	1000000	ok

Figure 4: Data export to csv format

3 EOS SLS Probing Script Simplification

SLS is an abbreviation that stands for Service Level Status. The system provides online monitoring information for numerous IT services provided at CERN. One of these services is also EOS. My task was to simplify old probing script that was used by SLS for high level monitoring of data storage availability in EOS.

3.1 Original Script

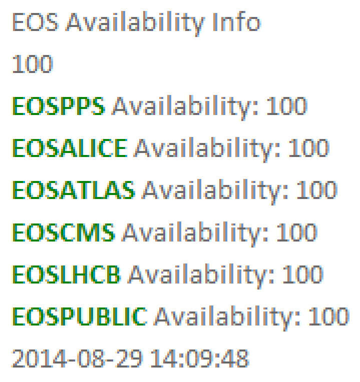
The old probing bash script was roughly 1000 lines long, this was unnecessary for the next generation tool supposed to display service information. The old system was not only able to track the availability of the service, but also able to track all the main KPI, while the new system is suppose to only provide RYG graph. The previous script was replaced by a new shorter version providing a different output.

3.2 New Solution

I implemented a new shortened script. The scripts as well as the original one parses standardized input xml file with information about availability of EOS systems. In addition to this, it also calculates an EOS aggregated availability indicator and creates an xml file, which includes short formatted html information about the availability of EOS subparts.

The new script is implemented in python and has about 100 lines, which makes it significantly more easily editable and understandable for the future use.

In addition, the script is easily reusable for different services. In fact it has been adopted also by the CASTOR service.



```
EOS Availability Info
100
EOSPPS Availability: 100
EOSALICE Availability: 100
EOSATLAS Availability: 100
EOSCMS Availability: 100
EOSLHCB Availability: 100
EOSPUBLIC Availability: 100
2014-08-29 14:09:48
```

Figure 5: Formatted display of the output of the SLS script

4 Conclusion

The main objectives were achieved. The user experience for EOS Cockpit was improved by reorganizing the online interface and including the possibility to download a csv file. The script for EOS monitoring in SLS was simplified and updated to fit the new SLS requirements.

In the future, the EOS Cockpit could be extended and also display historical data for analytical purposes for which a demo is already available. In general, the monitoring systems for EOS are many and a big scale simplification might be a useful step forward.