Centre de Calcul de l'Institut National de Physique Nucléaire et de Physique des Particules

TREQS-2, The IN2P3 prestaging tool

Pierre-Emmanuel Brinette,
Bernard Chambon

HPSS User Forum 2017

Agenda

- Why a prestaging tool ?
- Brief history
- TREQS-2
  - Design
  - Monitoring
- Future plan

Why a prestaging tool ?

- HEP computing model implies periodical reprocessing campaign
  - Ie. Large subsets of raw data acquired by detectors are read an reprocessed on the computing cluster.
  - Usually many hundreds of TB over few days.

- This kind of activity generates a huge load on HPSS
  - Data were initially written months or years ago and spread over many tapes.
  - Recall operations may implies many mounts/dismounts of the same tapes.

- Idea :
  - Increase staging performances by grouping files per tape

- Prestaging principles :
  - Trap the users file read request,
  - Create a queue for each file requested on the same tape,
  - Order this queue according to the (logical) position of the file on the tape,
  - Read the tape according to this order.

Brief History

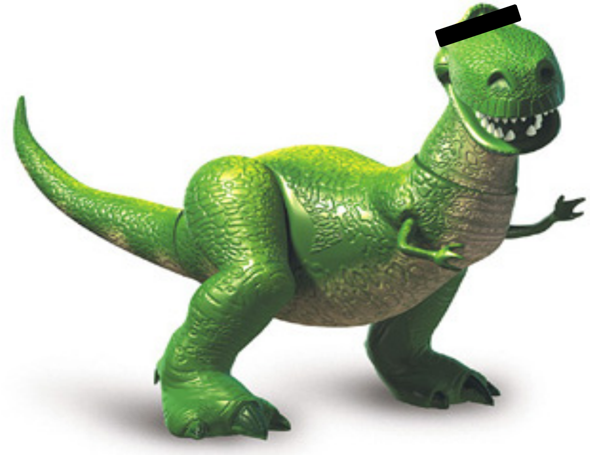
- Many different implementation of this principles
  - Ie: ERADAT @ BNL
  - ATOS

- At IN2P3 : TREQS (Tape REQuest Scheduler)
  - Client server/model

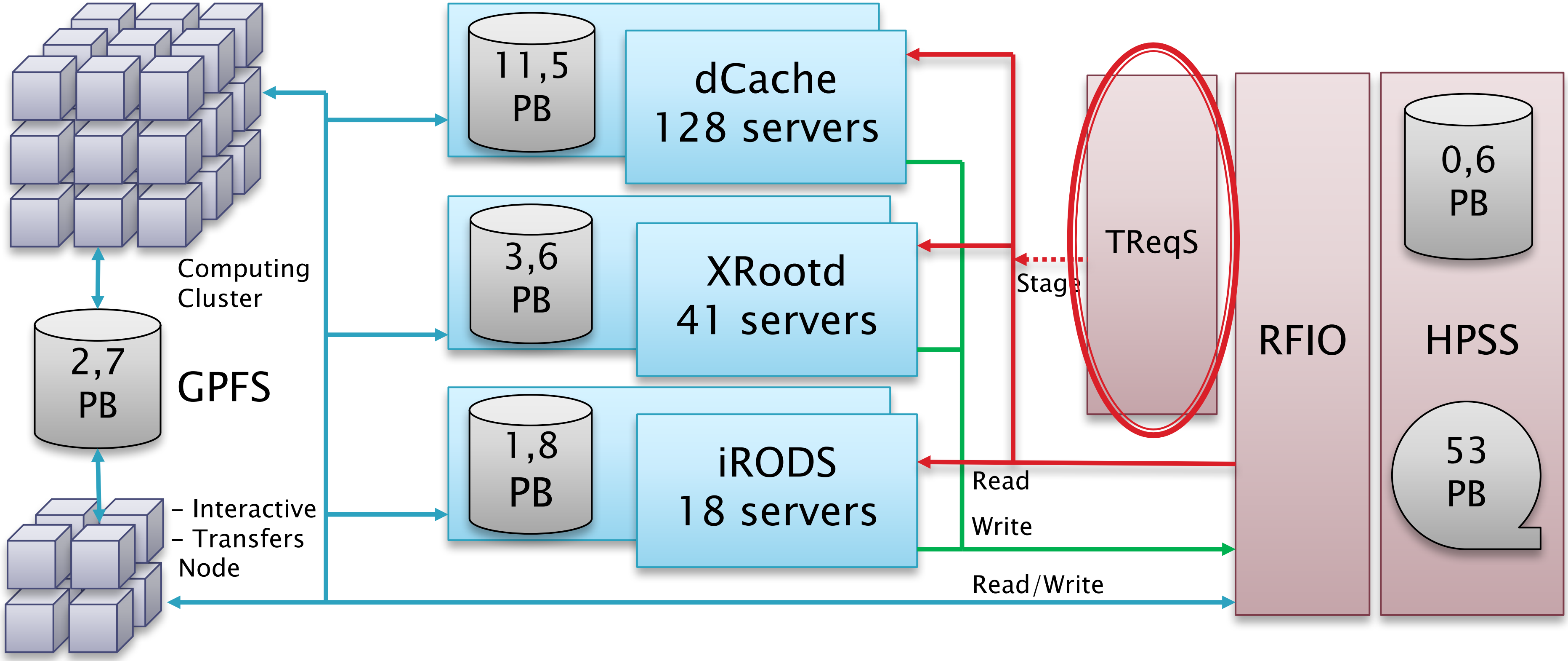
- Positioning
  - Between storage middleware and HPSS
  - For HPSS staging only (tape → disk)

- Previous version (used from 2009 to 2016)
  - Use a mysql database to store requests,
  - Requests directly inserted in the database by clients
  - Tape scheduling is done by the server on the DB

- Some limitations:
  - Scalability, performances,
  - Many statics parameters
  - Lack of functionality
    - Requests can’t be easily canceled



TREQS positioning



- 85 % of HPSS accesses are performed through storage middleware
  - **dCache** (LCG/egee),
  - **Xrootd** and **iRods**
- Still some direct accesses to HPSS but decreasing

- ALL **Read** operations from storage middleware are handled by **TREQS**

TREQS 2 Design

TREQS 2

- New development started in early 2016
  - Lead by senior developers
    - Bernard Chambon and Lionel Schwarz
  - Still in JAVA
  - Project managed by Maven
  - Stored in IN2P3 gitlab
  - Jenkins for continuous integration process
  - Sonar for code audit

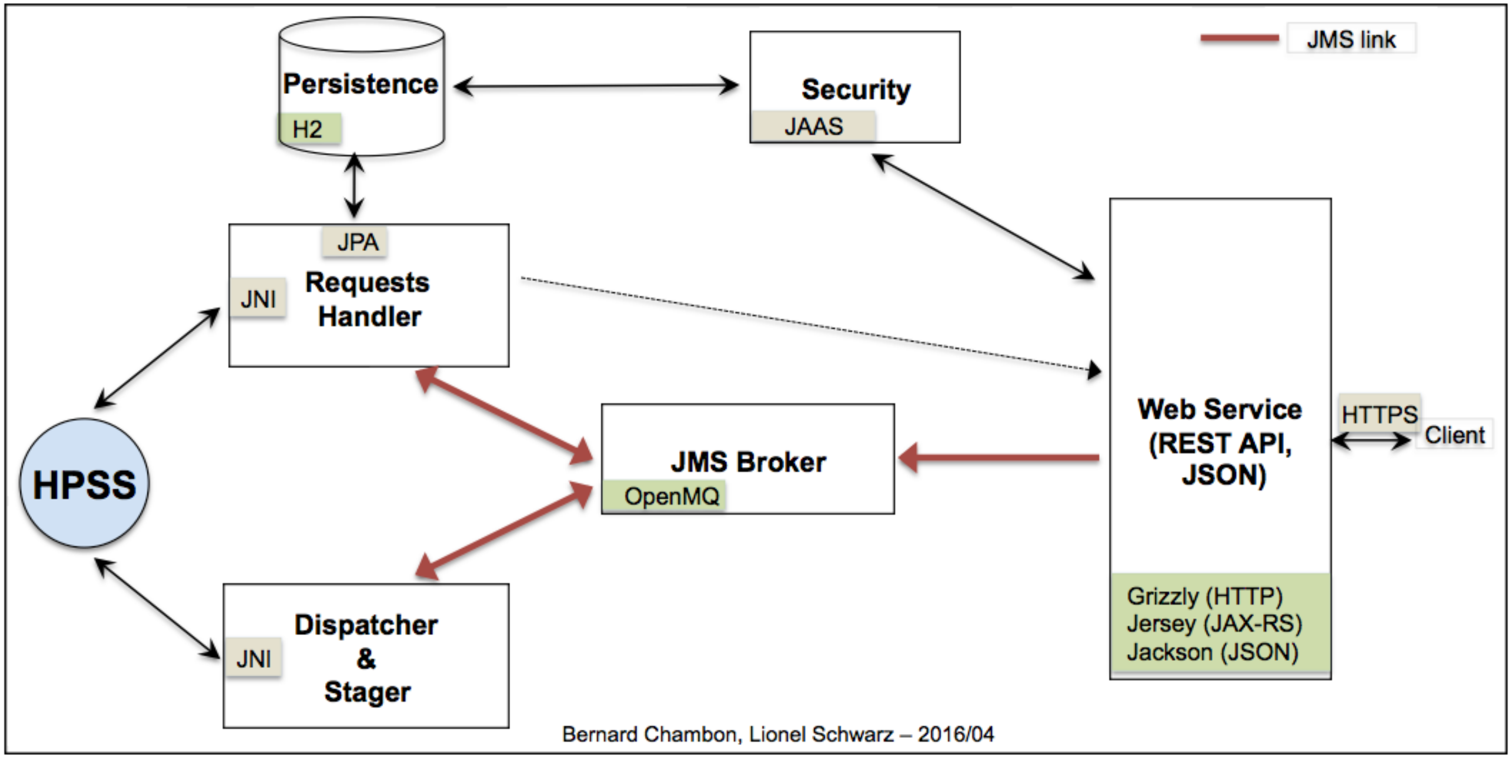
- First work presented at Hepix Spring 2016
  - https://indico.cern.ch/event/466991/contributions/1143626/

TREQS 2 Features

- TREQS 2 features
  - **Aggregate** requests over time per **tape**, **sorting** files according to logical file position on tape: → **queue**
  - **Stage queues** according creation time,
  - **Limit** the number of simultaneous running queues, per tape model
    - (ie: 10 drives allocated for T10K-D)
  - Provide **role management** (user's role = ADMIN,USER)
  - Provide **control** (on/off)
    - on tape,
    - on tape-model (T10K-C,T10K-D)
    - on HPSS access,
    - on queues processing,
    - on submission of client requests
  - Provide **cancelation** of client requests
  - Provide **persistence** for requests (useful for server stop & start)
  - Provide **archiving** for ended requests (built-in CSV archiver)

TREQS 2 Design

- Client / server model
- Server architecture
  - REST API with JSON, over HTTPS
  - JMS for internal exchanges, components with well delimited scope, less shared data structures
  - H2 DB as persistence: Fast, embedded (or server), 100% java
  - JNI to address HPSS API in C
  - JAAS for authentication & authorization
  - Mustache+DataTables for out-of-the-box monitoring web pages



Bernard Chambon, Lionel Schwarz – 2016/04

Client

REST API :

Staging :

```
curl -X POST -H 'Content-Type':'application/json' \
>     -d '{"file" : {"filename": "/hpss/in2p3.fr/group/ccin2p3/treqs/RUN01/ccwl9159.2773_000100Mb_0001.dat"}}' \
>     http://treqs:changeit@localhost:8080/treqs2/staging/request
{"id":"6f248806-f347-483a-adf4-a4f3cb8a38d1","status":"SUBMITTED","submitted_date":"2017-10-13T13:02:12Z","state":"SUBMITTED","owner":{"username":"treqs","role":"ADMIN"},"file":{"filename":"/hpss/in2p3.fr/group/ccin2p3/treqs/RUN01/ccwl9159.2773_000100Mb_0001.dat","state":null},"remote_address":"127.0.0.1"}
```

Status :

```
curl http://treqs:changeit@localhost:8080/treqs2/staging/request/6f248806-f347-483a-adf4-a4f3cb8a38d1
{
    "file": {
        "dispatched_date": "2017-10-13T13:02:12Z",
        "filename": "/hpss/in2p3.fr/group/ccin2p3/treqs/RUN01/ccwl9159.2773_000100Mb_0001.dat",
        "filesize": 104857600,
        "offset_position_on_tape": 838860800,
        "position_on_tape": 330,
        "state": "DISPATCHED",
        "status": "DISPATCHED",
        "tape": {
            "model": {
                "max_parallel_staging": 28,
                "name": "T10K-D",
                "reading_rate": 240,
                "status": "ENABLED"
            },
            "name": "KT757300",
            "status": "ENABLED"
[...]
}
```

Client

- Client : ‘treqs.py’
  - Written in python
  - Main usage : wrap the transfer command (‘treqs copy’)
    - At IN2P3 : RFIO (rfcp)
    - May works with any other command, even ‘cp’ over HPSS-FUSE
  - Bulk mode to stage a list of file in HPSS
    - Like quaid in HPSS 7.5.1
  - Monitor user activity (queue status, requests, etc)
    - Tabular output

- Admin client : ‘treqsadmin.py’
  - Written in python
  - Used to control server behaviors
  - Enable/disable {tape|user|submission|hpss…}

TREQS 2 Improvements

- Faster Metadata queries
  - HPSS metadata queries triggers at each file request,
  - Files that are already on disk cache are immediately in final state,
  - Others files are immediately scheduled on queues

- Increase the numbers of parallel recalls
  - dCache (main storage per LCG)
    - 100-200 recall per pool
    - Tens of pool per group/user
    - Max: more than 4000 // recalls
  - Xrootd / iRods
    - 50 // connections per server
    - 20 servers accessing HPSS
    - Up to 1000 // recalls

- TREQS handles thousands of simultaneous file requests
  - Only few of hpss_stage() handle by HPSS core server
  - Depend of the number of drives

- Staging rate has been improved up to 50% on large dataset.
  - Compared to TREQS 1
  - Benefit of increased number of // connections

Monitoring

Monitoring

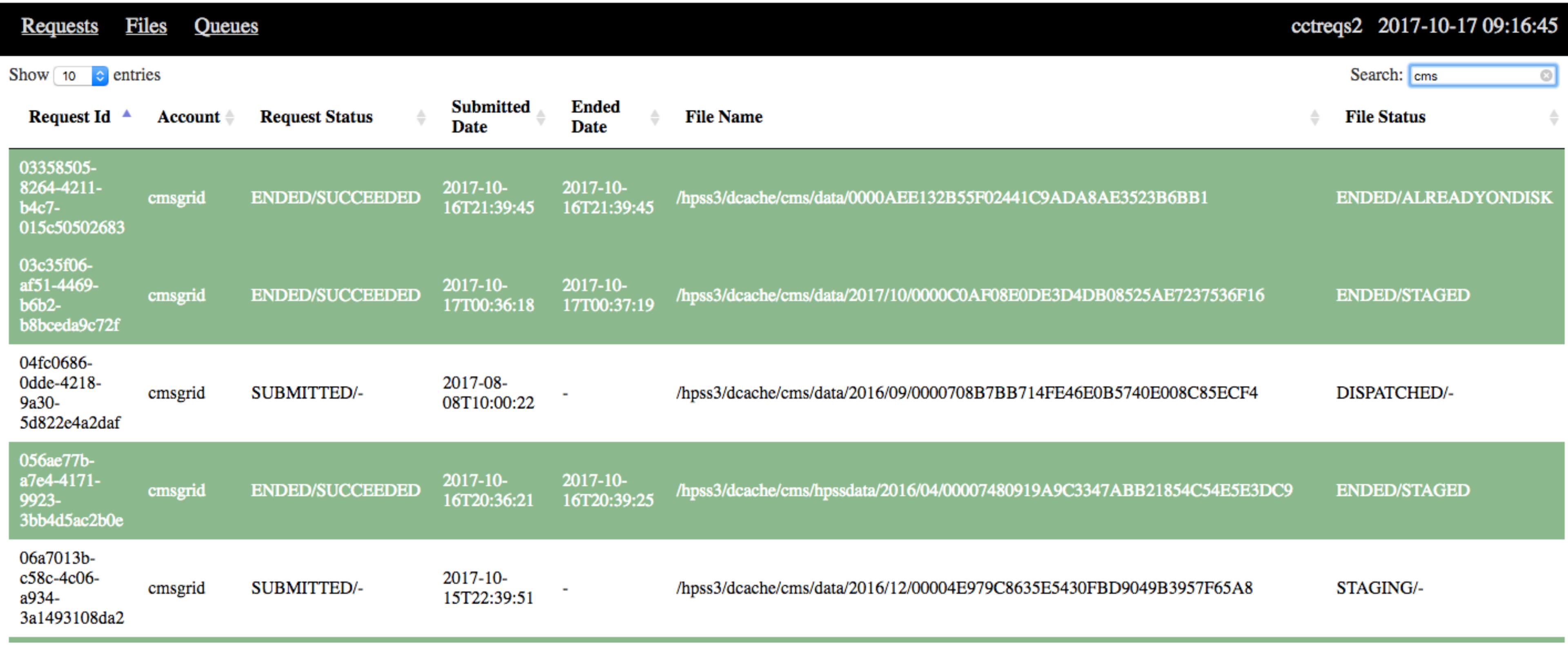
- Different level of monitoring

- Real time monitoring
  - Out of the box monitoring (Mustache+Datatable)
  - Web dashboard
    - By querying the web service

- Log based monitoring
  - End of processing logs sent to ElasticSearch cluster in JSON
  - Automatically indexed
  - Many possibilities offered by Kibana

Real Time monitoring

- Mustache + Datatable (embeded in treqs2)



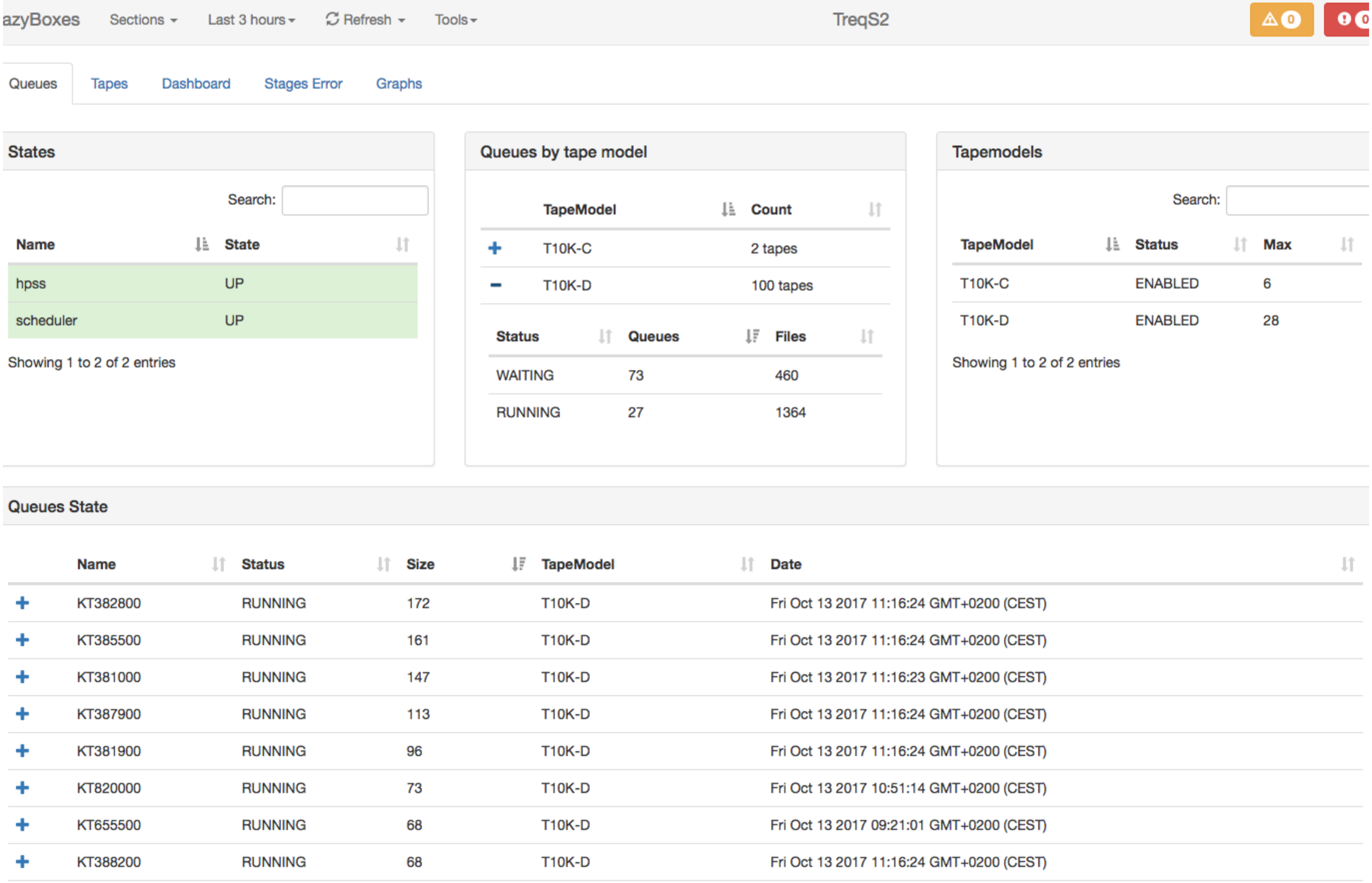
cctreqs2 2017-10-17 09:16:45

Show 10 entries

Search: cms

Request Id	Account	Request Status	Submitted Date	Ended Date	File Name	File Status
03358505-8264-4211-b4c7-015c50502683	cmsgrid	ENDED/SUCCEEDED	2017-10-16T21:39:45	2017-10-16T21:39:45	/hpss3/dcache/cms/data/0000AEE132B55F02441C9ADA8AE3523B6BB1	ENDED/ALREADYONDISK
03c35f06-af51-4469-b6b2-b8bceda9c72f	cmsgrid	ENDED/SUCCEEDED	2017-10-17T00:36:18	2017-10-17T00:37:19	/hpss3/dcache/cms/data/2017/10/0000C0AF08E0DE3D4DB08525AE7237536F16	ENDED/STAGED
04fc0686-0dde-4218-9a30-5d822e4a2daf	cmsgrid	SUBMITTED/-	2017-08-08T10:00:22	-	/hpss3/dcache/cms/data/2016/09/0000708B7BB714FE46E0B5740E008C85ECF4	DISPATCHED/-
056ae77b-a7e4-4171-9923-3bb4d5ac2b0e	cmsgrid	ENDED/SUCCEEDED	2017-10-16T20:36:21	2017-10-16T20:39:25	/hpss3/dcache/cms/hpssdata/2016/04/00007480919A9C3347ABB21854C54E5E3DC9	ENDED/STAGED
06a7013b-c58c-4c06-a934-3a1493108da2	cmsgrid	SUBMITTED/-	2017-10-15T22:39:51	-	/hpss3/dcache/cms/data/2016/12/00004E979C8635E5430FBD9049B3957F65A8	STAGING/-

Real Time monitoring



States

Search:

Name	State
hpss	UP
scheduler	UP

Showing 1 to 2 of 2 entries

Queues by tape model

	TapeModel	Count
+	T10K-C	2 tapes
−	T10K-D	100 tapes

Status	Queues	Files
WAITING	73	460
RUNNING	27	1364

Tapemodels

Search:

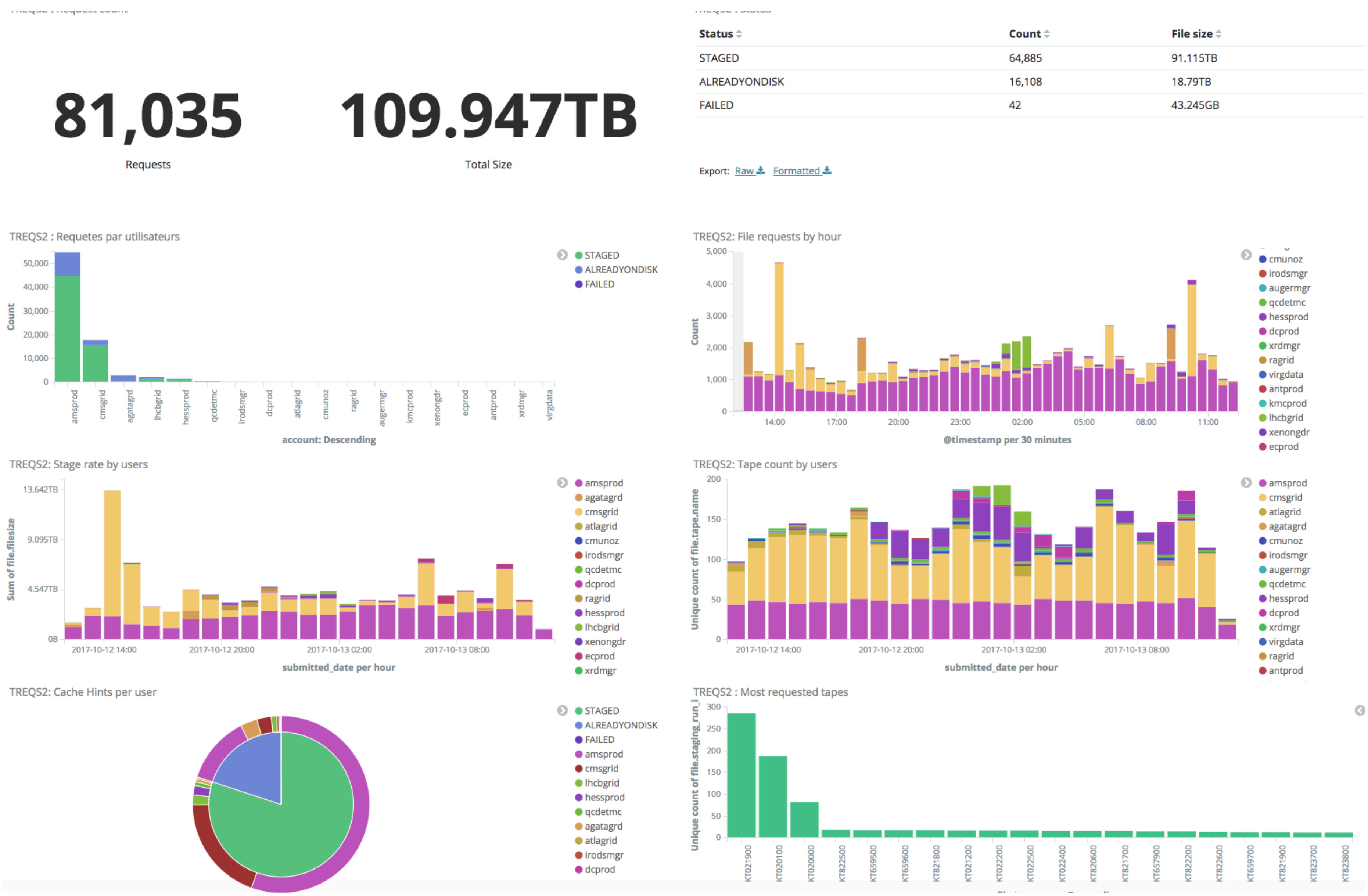
TapeModel	Status	Max
T10K-C	ENABLED	6
T10K-D	ENABLED	28

Showing 1 to 2 of 2 entries

Queues State

	Name	Status	Size	TapeModel	Date
+	KT382800	RUNNING	172	T10K-D	Fri Oct 13 2017 11:16:24 GMT+0200 (CEST)
+	KT385500	RUNNING	161	T10K-D	Fri Oct 13 2017 11:16:24 GMT+0200 (CEST)
+	KT381000	RUNNING	147	T10K-D	Fri Oct 13 2017 11:16:23 GMT+0200 (CEST)
+	KT387900	RUNNING	113	T10K-D	Fri Oct 13 2017 11:16:24 GMT+0200 (CEST)
+	KT381900	RUNNING	96	T10K-D	Fri Oct 13 2017 11:16:24 GMT+0200 (CEST)
+	KT820000	RUNNING	73	T10K-D	Fri Oct 13 2017 10:51:14 GMT+0200 (CEST)
+	KT655500	RUNNING	68	T10K-D	Fri Oct 13 2017 09:21:01 GMT+0200 (CEST)
+	KT388200	RUNNING	68	T10K-D	Fri Oct 13 2017 11:16:24 GMT+0200 (CEST)

Kibana based dashboard :



TREQS2 : Request count

81,035

Requests

109.947TB

Total Size

TREQS2 : Status

Status ⇕	Count ⇕	File size ⇕
STAGED	64,885	91.115TB
ALREADYONDISK	16,108	18.79TB
FAILED	42	43.245GB

Export: Raw ⬇ Formatted ⬇

TREQS2 : Requetes par utilisateurs

TREQS2: File requests by hour

TREQS2: Stage rate by users

TREQS2: Tape count by users

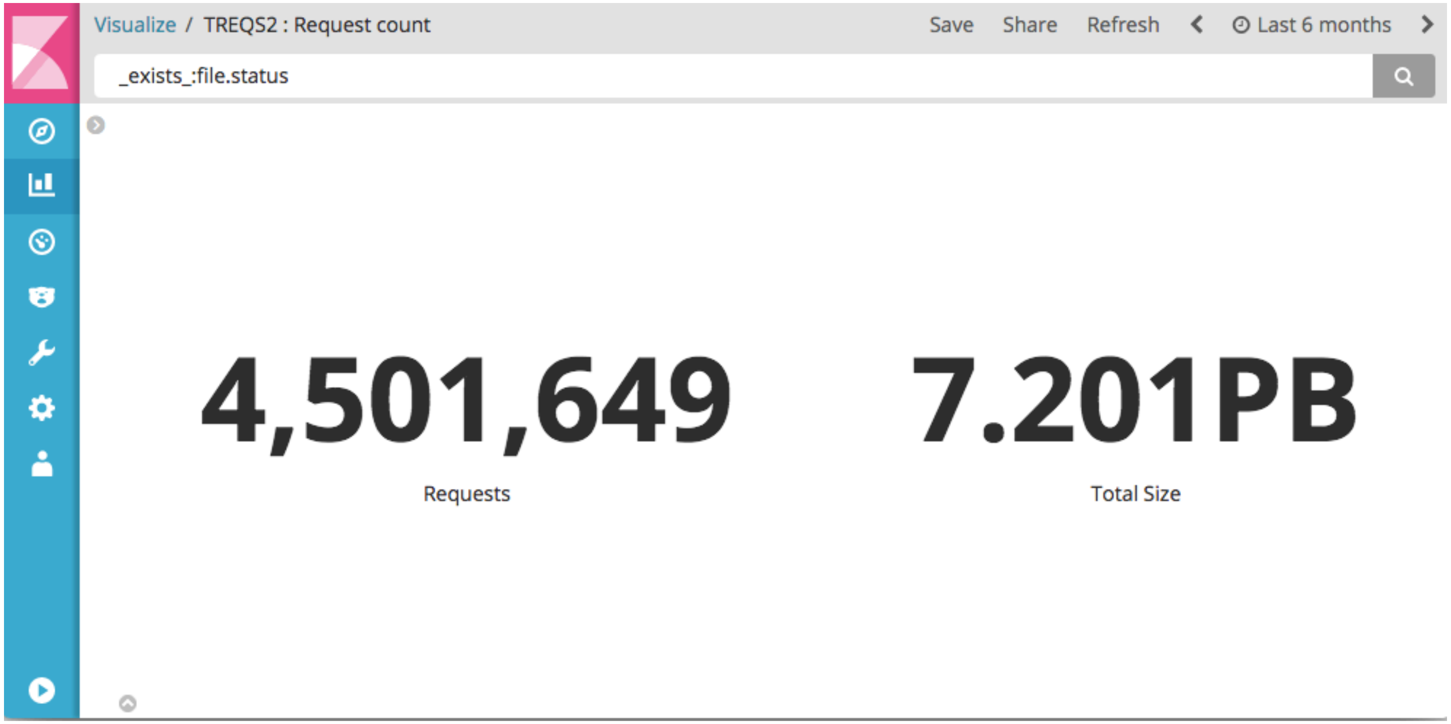
TREQS2: Cache Hints per user

TREQS2 : Most requested tapes

As a conclusion

Feedback

- In production since May 2017



exists:file.status

4,501,649
Requests

7.201PB
Total Size

- Very stable
  - 1 issue due to an H2 bug, quickly fixed
  - Service only restarted during the scheduled maintenance

Future

- Still relevant even with HPSS 7.5 new features
  - Useful to control / throttle user activity,
  - Limit the number of drives for recall operations,
  - Still store requests even the core server is down,
  - Would benefit of HPSS Tape Ordered Recall
    - But may need some changes in the code (background staging)

- Code available for the HPSS community
  - https://gitlab.in2p3.fr/cc-in2p3-dev/treqs2
  - License : GPLv3
  - Account opened on request

- Next release : Log enhancement
  - Extract more files metadata from HPSS
    - File creation date, access counts, etc …
  - Usefull to collect access stats with Elasticsearch / Kibana
    - Ie: How 'old' are the recalled data, ...

Conclusion

Thank you

Visit us at SC17

Denver, CO | hpc connects.

Booth #743