

Data Store and Query System



- Import
 - Procedures
 - Status
- Event Lookup
 - Fast Client
 - Web GUI
- Web Service
 - News, Problems and Solution
 - Secondary Server
- **ToDo**



Import - Procedures

- VALIDATED datasets arrive from Consumer in groups of 100-300 conf files
- Those are usually imported within a day (so no backlog)
 - data imported within hours, mc takes longer
- Problems can happen (e.g. AFS unavailability last week)
- Consistency and correctness of imported files are checked nightly
 - Problematic datasets are re-imported
- Some TagFiles imported in summer have invalid keys
 - should be fixed in Consumer
- It would be useful to verify not only internal consistency, but also consistency with AMI, Rucio,...
 - and put results into Catalog
 - there are certainly errors coming from re-imports, corrected imports, summer failures,...
- Files can be imported from HAR file (almost automatic procedure put in place)
 - Slower
- If needed, import can be optimised (to make it up to factor 10x faster)
 - as discussed in December
 - still problem with large datasets (millions of files) - too slow 
- The storage is **heterogeneous** by design: the TagFiles are described by *.conf files, which are included in the Catalog during import
 - => TagFile characteristics (schema,...) are valid for concrete TagFile
 - => different TagFiles can have different characteristics, even different format



help would be welcomed



Import - Status

IMPORTED

=====

EI 2009 2	6568
EI 2009 1	526
EI 2010 2	347527
EI 2010 1	4519
EI 2011 2	514183
EI 2011 1	3944
EI 2012 2	291066
EI 2012 1	4400
EI 2013 2	184593
EI 2013 1	525
EI 2014 2	32180
EI 2014 1	492
EI 2015 2	1491907
EI 2015 1	21470
MC 2009 2	0
MC 2009 1	0
MC 2010 2	0
MC 2010 1	0
MC 2011 2	0
MC 2011 1	0
MC 2012 2	60
MC 2012 1	3
MC 2013 2	0
MC 2013 1	0
MC 2014 2	0
MC 2014 1	0
MC 2015 2	2005896
MC 2015 1	13645

WAITING (VALIDATED PARTIAL IMPORTED MERGING FAILED)

=====

EI 2009 0	0	0	0	0
EI 2010 0	3	0	0	0
EI 2011 0	91	0	0	0
EI 2012 0	4	0	0	0
EI 2013 0	88	0	0	0
EI 2014 0	4	0	0	0
EI 2015 0	1111	0	0	1
MC 2009 0	0	0	0	0
MC 2010 0	0	0	0	0
MC 2011 0	0	0	0	0
MC 2012 0	0	0	0	0
MC 2013 0	0	0	0	0
MC 2014 0	0	0	0	0
MC 2015 0	1697	0	0	0

status from 25Jan morning (nightly statistics)



Event Lookup - Fast Client

- It is a special program, much simpler than the general "ei" command, so it's much faster and less flexible. In other words, it's a single-purpose program.
- It is multithreaded (fork_and_join), it uses all cores.
- The same functionality (with additional options) is available from the "ei" command
 - but slower
- There is a space for optimisation (if needed):
 - Don't search already found GUIDs.
 - Optimise multithreading (currently one thread per core).
 - Search all GUIDs in one TagFile together (currently each search is new).
 - Send compressed result (useful for big results).
 - Use dedicated server.
- Similar special programs can be provided for other user requirements. 

```
java -jar EventLookup.exe.jar <arguments>   
<arguments>: [-h|--help] [-api [simple|rich]] [-e <runs-events>] [-f <runs-events filename>]  
[-s <stream name>] [-p <amitag>] [-d [AOD|<data type>]] [-t <format type>]
```





Event Lookup - Fast Client

```
% el -f ev1.txt -api simple
```

```
 0 INFO  (Util.Init           : 62) : Initialised, version: 1.20.1+ [25/Jan/2016 at 13:37:58 CET by hrivnac]
95 INFO  (Apps.ELCLI          : 151) : CLI arguments: [-f, ev1.txt]
1959 INFO (Apps.ELCLI          : 313) : Searching TagFile for id:EI15.1. runNumber:00278880 dataType:AOD
3090 INFO (Apps.ELCLI          : 240) : Found 9 TagFiles for 1 runs
3093 INFO (Util.SimpleGuider       : 119) : Using Simple API
...
4414 INFO (Apps.ELCLI          : 251) : Found guids:
D3D51279-40A3-1E40-B4DA-E0EA4D6BE21F StreamESD
E6CC54DB-32FB-054E-AC05-C3CEDA07550C StreamAOD
A2FE0F1F-9364-7545-AF90-D2DC26621B37 StreamESD
...
4415 INFO (Apps.ELCLI          : 252) : 8 guids found in 9 TagFiles for 1 runs, 1 events
4484 WARN (Journal.Record      : 144) : Journal not updated due to missing authorisation
4484 INFO (Apps.ELCLI          : 117) : Task took 4s
```

```
% el -f ev1.txt -api rich
```

```
 0 INFO  (Util.Init           : 62) : Initialised, version: 1.20.1+ [25/Jan/2016 at 13:37:58 CET by hrivnac]
99 INFO  (Apps.ELCLI          : 151) : CLI arguments: [-f, ev1.txt, -api, rich]
2008 INFO (Apps.ELCLI          : 313) : Searching TagFile for id:EI15.1. runNumber:00278880 dataType:AOD
3090 INFO (Apps.ELCLI          : 240) : Found 9 TagFiles for 1 runs
3093 INFO (Util.RichGuider       : 119) : Using Rich API
...
4476 INFO (Apps.ELCLI          : 251) : Found guids:
>>>
00278880-00558085589
D4A7979D-F229-D248-BBDC-40CCC6B504E1 StreamAOD
6182B042-3643-984A-8781-0B6F0E6FE5B2 StreamESD
...
4477 INFO (Apps.ELCLI          : 252) : 9 guids found in 9 TagFiles for 1 runs, 1 events
4549 WARN (Journal.Record      : 144) : Journal not updated due to missing authorisation
4549 INFO (Apps.ELCLI          : 117) : Task took 4s
```



Event Lookup - Fast Client

order of magnitude

n_of_events	found_guids	searched_runs	searched_tagFiles	t_local[s]	t_lxplus[s]
1	3	1	6	7	2
10	39	1	6	6	2
50	185	1	6	11	3
100	381	2	12	11	4
500	1803	9	54	35	11
1000	3808	22	146	52	31
5000	9957	50	319	201	120
8879	24179	66	426	534	193

- Database query for smallest request is < 1s, there is an access program overhead
- Remote (lxplus) requests were running about 10 min after local requests and Tomcat was restarted. This means, that we profit from the Hadoop cache (but not from the Tomcat one).
- Using -s/-p arguments will speed up the search by an order of magnitude.
- The search is executed on the same machine as the import etc. - so performance is clearly much worse than in case of a dedicated server.
- Should do more comprehensive benchmarks (random selection of searched tags, simultaneous access from several clients,...) - so we can tune the cache,...





Event Lookup - Web GUI



Event Index

- [Globo Help](#)
- [Catalog](#)
- [Event Index \(Expert Mode\)](#)
- [Event Lookup](#)
- [Bookmarks](#)
- [System Journal \(for admins\)](#)



Event Lookup

List of 'runnumber evtnumber' (-e) 00278880 558085589, 00278880 210318172

Stream name (-s)

physics_Main ▼

AMI tag (-p)

Data type (-d)

*** all data types *** ▼

GUID type (-t)

AOD ▼

-email

-api

☐ simple ☒ rich

Search

Reset

```
>>>
00278880-00210318172
ABA9C248-E945-F84A-BD2C-9A312A03C6FD StreamAOD
AAB98A83-7374-694C-A44D-38FD1F285832 StreamAOD
>>>
00278880-00558085589
D4A7979D-F229-D248-BBDC-40CCC6B504E1 StreamAOD
E6CC54DB-32FB-054E-AC05-C3CEDA07550C StreamAOD
```



Web Service - News, Problems and Solution

- Sometimes (about once a month) server get stuck for various reasons
 - availability monitored and server restarted if needed
- Tomcat logfile filled-up root filesystem
 - restart failed because root filesystem was full
 - added additional check for logfile size in monitor
- Two ways of access (with identical functionality):
 - <http://atlas-event-index.cern.ch> visible in CERN, without authentication
 - used by Panda
 - <https://atlas-event-index.cern.ch> visible worldwide, Shiboleth authentication
- J2EE proxy server not used any more
- Very long queries (like scanning of 200GB TagFile) fail on timeout
 - will change Apache/Tomcat configuration



Web Service - Secondary Server

- New (secondary) server (almost) running @aiatlas054
- It will serve several purposes:
 - fallback if aiatlas016 unavailable
 - development
- Finalising configuration
 - aiatlas054 should have been an exact clone of aiatlas016, but it isn't (discussing with IT)
 - firewalls, url aliases, kerberos,...



To Do

- Clean the code and procedures
 - There is a lot of dead code from canceled requirements
- Improve Web Service
 - need input, requests
- Improve logistics
 - cache cleaning
- Fix TagFiles with wrong keys
 - problem in Consumer files, some can be solved
- Find a way to import very big datasets
 - merging millions of files
- Cross-check consistency with AML, Rucio
- Update Catalog with meta-metadata
 - number of events per TagFile, archiving status,...
- Use accumulated statistics & journal
- Backup *.1 TagFiles into HAR archives
- Use better Catalog backup
 - currently just nightly dump
- Develop meaningful benchmarks
- Try approach with HBase table with pointers (Rainer)
 - has been tried for ES, looked promising
- Remove *.2 files (from HDFS & Catalog)
- Consolidate import logistics:
 - Consumer writing mapfiles + duplicate detection + mapfiles enhancements (trigger + ...) + mapfiles merge
 - will also make import much faster