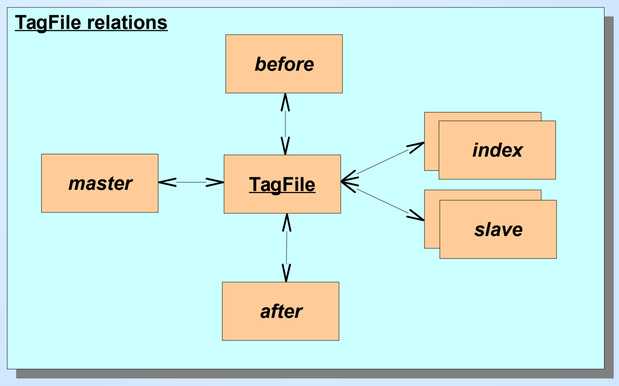
QuerySpaces on Hadoop for the ATLAS EventIndex

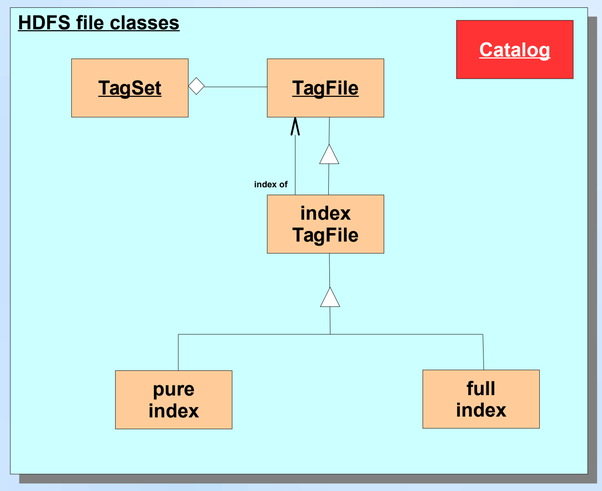
CHEP2015 – Computing in High Energy and Nuclear Physics 2015, 13-17 April 2015, Okinawa, Japan

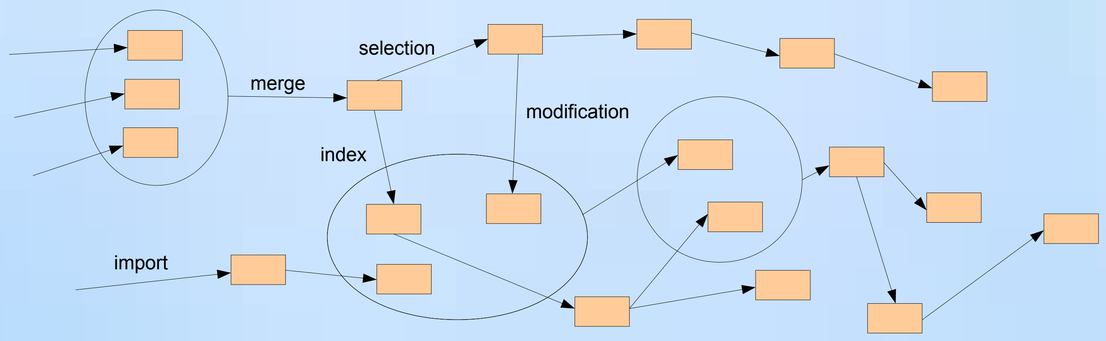
J. Hřivnáč[4*], R. Yuan[4], J. Cranshaw[2], A. Favareto[1], F.Prokoshin[6], C. Glasman[3], R. Többicke[5]
on behalf of the ATLAS Collaboration
[1]Università di Genova and INFN, Genova, Italy, [2]Argonne National Laboratory, Argonne, IL, United States, [3]Universidad Autonoma de Madrid, Madrid, Spain, [4]LAL, Université Paris-Sud and CNRS/IN2P3, Orsay, France, [5]CERN, Geneva, Switzerland, [6]Universidad Technica Federico Santa Maria, Chile
*Corresponding author

A Hadoop-based implementation of the adaptive query engine serving as the back-end for the ATLAS EventIndex. The QuerySpaces implementation handles both original data and search results providing fast and efficient mechanisms for new user queries using already accumulated knowledge for optimisation. Detailed descriptions and statistics about user requests are collected in HBase tables and HDFS files. Requests are associated to their results and a graph of relations between them is created to be used to find the most efficient way of providing answers to new requests The environment is completely transparent to users and is accessible over several command-line interfaces, a Web Service and a programming API.

- Each search task creates a new TagFile (in the cache), either as a full-content file or as an index. This TagFile is registered in the Catalog and can be reused as a basis for future searches. All TagFiles can be represented in a connected graph of dependencies, captured in the Catalog. All TagFiles can be accessed with the same interface.
- Searching can use several techniques:
  - Key-based search: Very fast.
  - Full Map/Reduce: Reasonably fast and very flexible. Query clause can contain any legal (Java) code to perform the selection and extraction.
  - Scan search: Slow, but transparent. Usefull for debugging.
- Search result can represent several operations:
  - Subset selection.
  - Merge.
  - Content modification: Any data field can be changed/added/removed.
  - Re-arrangement: In most cases a creation of new index to allow fast key-based search.
- TagFiles are stored in the form of Hadoop Map files with indexes kept in memory. Usage of more advanced storage formats is under investigation.



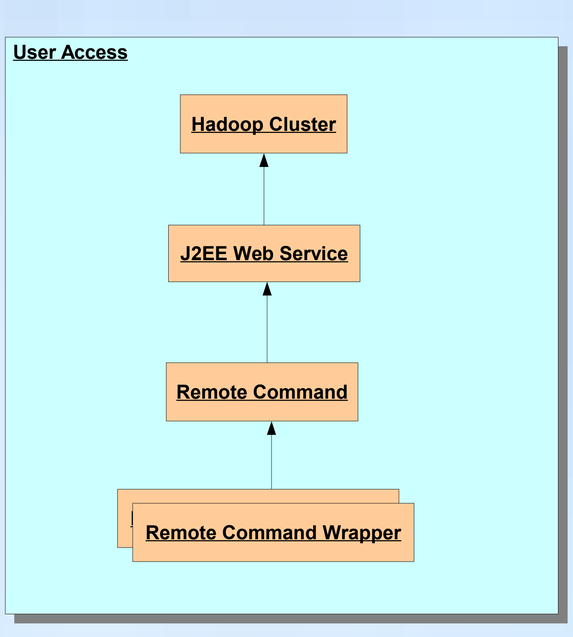




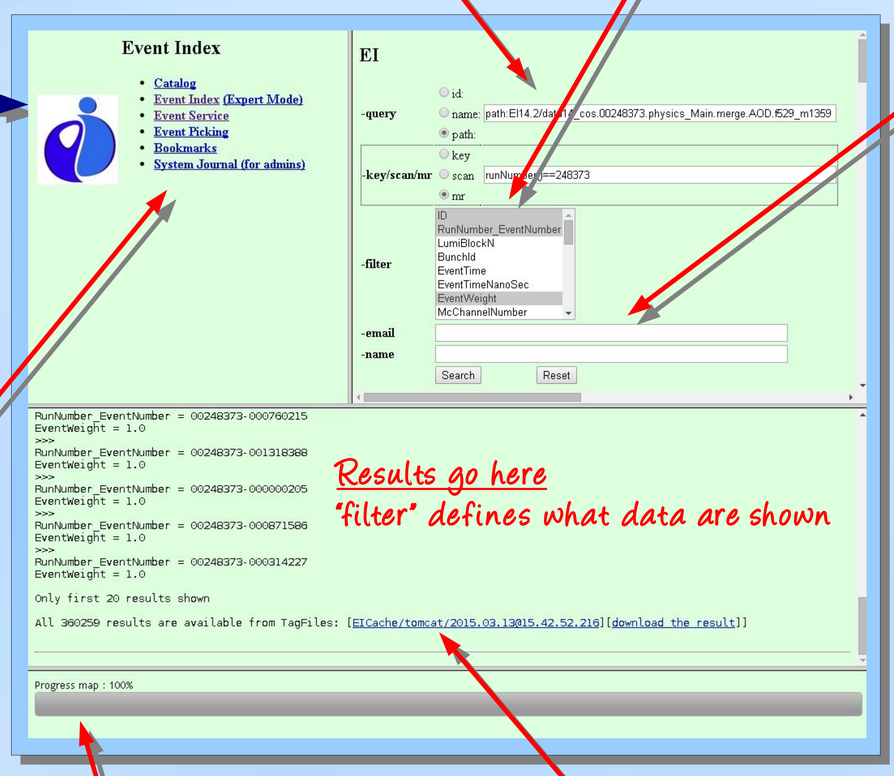
Command examples

```
catalog -query  'path:EI14.2/data14_cos.00248373.physics_Main.merge.AOD.f529'
catalog -count  'path:EI14.2/data14_cos.00248373.physics_Main.merge.AOD.f529'
catalog -query  'exact EI14.2/data14_cos.00248373.physics_Main.merge.AOD.f529_m1359'
catalog -query  'pandauserid:30278306.1 transid=dc92d22685eb45dfa0271d7fcc9715f6'
catalog -add    'path:EICache/Test name:ATest'
catalog -query  'path:EICache/Test' -modify 'mytag:tag1'
catalog -delete 'path:EICache/Test'
ei -query $qrx -show 2 -key '00248373-000983650..00248373-000983656'
ei -query $qry -key '00248373-000983650..00248373-000983656' -scan 'hasGuid("EC2886B0-519E-704B-8F3A-92CF124E3D5E")'
ei -query $qry -key '00248373-000983650..00248373-000983656' -scan 'hasGuid("EC2886B0-519E-704B-8F3A-92CF124E3D5E")' -count '1'
ei -query $qry -key '00248373-000983650..00248373-000983656' -scan 'hasGuid("EC2886B0-519E-704B-8F3A-92CF124E3D5E")' -count 'EventWeight*EventWeight'
ei -query $qry -key '00248373-000983650' -output index -index 'id=String.valueOf(BunchId)'
ei -query $qry -key '00248373-000983650..00248373-000983656' -output index -index 'id=String.valueOf(BunchId)' -scan 'hasGuid("EC2886B0-519E-704B-8F3A-92CF124E3D5E")'
ei -query $qry -mr 'BunchId==2390 && LumiBlockN==70'
ei -query $qry -mr 'runNumber()==248373' -filter 'BunchId RunNumber_EventNumber'
ei -query $qry -mr 'true' -count 'EventWeight'
ei -query $qry -mr 'BunchId==2390' -output index -index 'id=String.valueOf(BunchId)'
ei -query $qry -mr 'hasGuid("EC2886B0-519E-704B-8F3A-92CF124E3D5E")' -filter 'String token()' -index 'oid0'
ei -query $qry -eventlist eventlist.txt -filter 'clid0'
ei -query $qry -show 2 -mr 'true' -extent 'NewField=String.valueOf(BunchId*BunchId)'
ei -query $qry -show 2 -mr 'true' -update 'NewField=String.valueOf(2*NewField)'
```

- There are several ways to access EventIndex functionality:
  - Using simple command line interface, available from ATLAS CVMFS or downloadable as a standalone package usable on Linux, MS and MacOSX.
  - Using Graphical WebService.
  - Accessing simple WebService from command line.
  - From end-user applications with interfaces to specific clients (Event Server, Event Lookup,...) accessing WebService behind the scene.



Choose Service here
- Expert Mode has the same arguments as interactive command
- EventService and EventPicking are special cases of EventIndex
- Bookmarks gives you your previous searches (and results)



Formulate your query here

Default choices give sensible answers

Fields have context-sensitive help

Results go here
"filter" defines what data are shown

Here you see the job progress

You can re-start the same query (with some modified parameters) here

http://cern.ch/hrivnac/Activities/Packages/TagConvertor