Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

Best practices: SAM

Robert Illingworth

FIFE Workshop

21 June 2016

FABRIC FOR FRONTIER EXPERIMENTS

SAM data management

- This is not intended as an introduction or overview
  - See, for example, last year's workshop if you want one of those
- Instead I shall mention some of the issues that we've encountered
- This is therefore a somewhat random collection of topics

Performance

- Overall SAM performance depends on the number of files, to some extent on the complexity of the metadata, and on the rate of client calls to the server
  - Actual data volume is irrelevant here (although obviously important when actually accessing it)

- You're likely to need 10s of millions of files catalogued before database performance becomes an issue

- We've made a number of internal performance improvements based on experience
  - Faster for free!
- But you can help by structuring and querying metadata the right way

Metadata query performance

- Bigger files are better – means there are fewer of them
- Built in metadata fields (run number, application, datastream, etc) usually better than defining your own
- For numeric generic parameters, make sure the type is set correctly
  - Treating them as strings is slower and greater/less than queries don't work properly
- More selective metadata queries are better; worse case is queries on multiple mostly overlapping terms each returning a large number of results
- The parent/child operators make it easy to build up extremely complex queries – these can be slow

Metadata query performance

- The `defname:` operator re-evaluates the target definition every time you use it
- In some cases `dataset_def_name_newest_snapshot` may be preferable – it substitutes the most recent snapshot, which is a static list.
- The command

  ```
  samweb list-files --dump-query <dimensions>
  ```

  displays the query after all defname terms are expanded (as well as the actual SQL query); in some cases you can rewrite it to be simpler by factoring out common terms or eliminating redundant ones

Bulk query/modification operations

- When doing large scale operations, use methods which operate on multiple files whenever possible
  - Less overhead, plus some can be implemented more efficiently this way
- This is mainly aimed at anyone using the API
- Currently available declare files; query metadata; query locations by name; bulk declare files
- On their way: modify metadata; list locations for an entire dataset (implemented in the server but not yet in the samweb client)
- Likely to come eventually: bulk update locations

Offsite file storage

- The SAM catalogue is storage agnostic
  - File locations don't have to be at Fermilab
- Offsite storage does have to support standard storage protocols so that we can transfer and access files
  - Gridftp, xrootd, etc

- So far no-one is really using this
  - We have used files in Amazon S3 as input to cloud jobs

SAM Station/projects

- Recently updated SAM stations to newer version with built in HTTP support
  - I think very few people actually noticed
  - New version should be more scalable

- One ongoing problem is that preemption isn't handled very well
  - May see multiple processes with the same batch ID
  - Files seen by a preempted job will not be resent to other jobs
  - It also tends to mess up the station monitor plots
  - Need to look at final process state or existence of child file to determine if a file really has been processed

Other storage protocols

- We set the default access protocol to gridftp for most cases
- But both projects and the get-file-access-url command support various other protocols
  - For example xroot, http

```
samweb start-process --schemas=xroot,http,gsiftp …
```

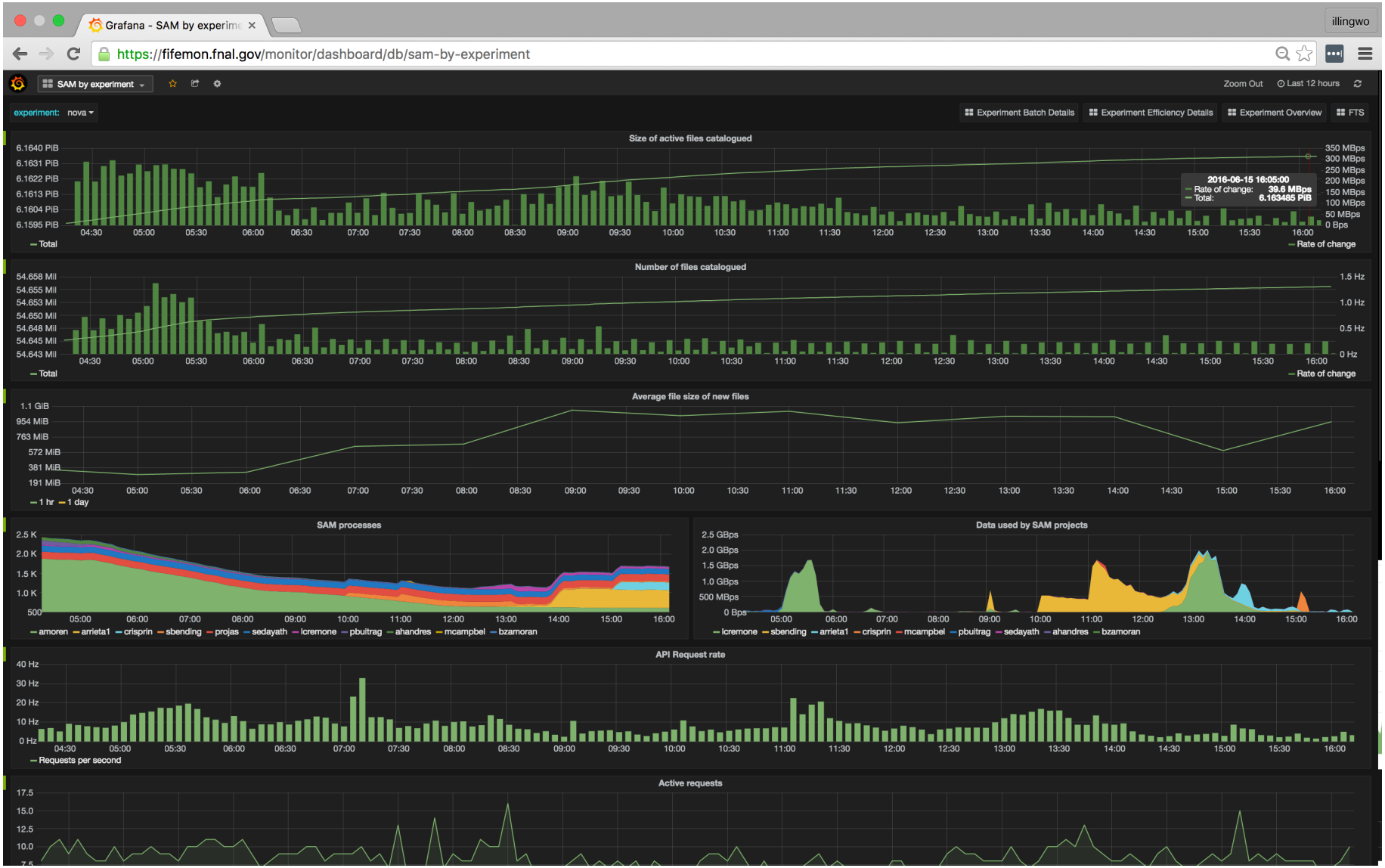
(or equivalent ifdh command)

This returns the first access protocol in the list that is supported by the storage system holding the file

Authentication/Authorization

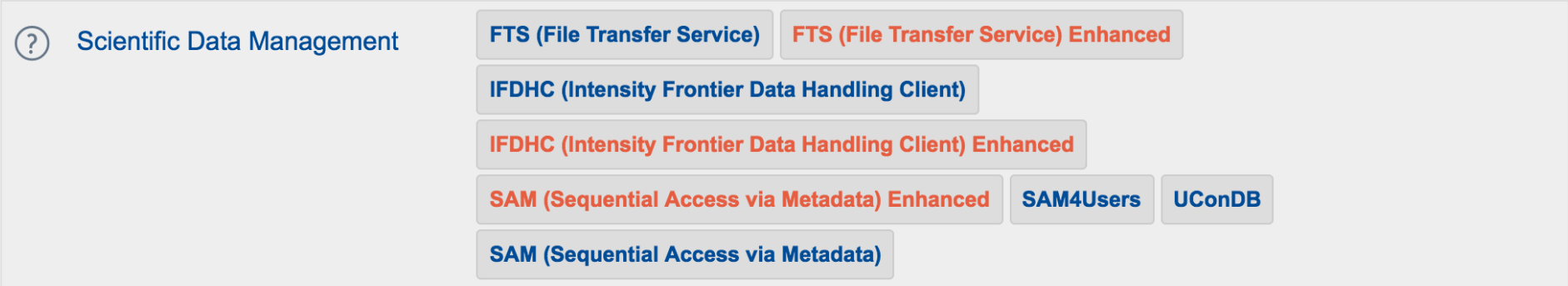
- Authentication for SAM is done by x509 certificates/proxies
  - (Black magic as far as most people are concerned…)
- This uses the same proxies as job submission and ifdh, so most users don't need to do anything
- Cases which cause problems are automated and production tasks
- Automated tasks need a proxy that is independent of a login session
  - For individuals this means kcron/kx509
  - For production tasks you can use a proxy from a service principal
- Note that SAM does not (currently) use the VOMS authentication that jobsub and dCache use
  - If you have a certificate you want to have access you need to request that it is added to the SAM database

Monitoring with FIFEMon



Support

- Support is available through the servicedesk – now in the Scientific Computing Services catalogue:



- Using the right link sends the ticket directly to the correct group (especially important outside normal business hours).

Final thoughts

- If you have requirements that seem out of the ordinary (large number of files, complex queries, etc) please ask first
  - There may be a better way of doing it
  - Or we may even be able to add new functionality

- It's much better to try and address issues before there are a few million new files with incorrect metadata