Science and Technology Facilities Council

ICAT and SciCat Evaluation

Testing of performance for future data volumes

Kevin Phipps
May 2023 (ICAT F2F meeting in Berlin)

Introduction

- This work was done in 2019 as part of the Diamond Datastore Project looking at how to take the Diamond Archive into the future

- Disclaimer: both ICAT and SciCat have evolved since then so some findings may no longer be true

- The slides on ICAT performance were presented at the ICAT F2F in 2020
  A cut down version of them will be presented here.

Science and Technology Facilities Council

Objectives

- To analyse the responsiveness of the data catalogues to metadata growth
- Simulate the metadata growth 6 years in the future by duplicating the data from 2018 a year a time
- Measure the time taken to respond to queries after each ingestion step
- Consider the queries as required by the frontend in the MyData and Browse views

Test procedure

Starting with a copy of the current database (2.5 billion files)

- Run 2 tests collecting query timing data
- Run a duplicator script to copy the data for 2018 as if it was being inserted for a new year (0.5 billion files)
- Repeat this 6 times – simulating 6 years into the future (5.5 billion files)

Test types

2 tests

- A “simple test” recording results using a high resolution counter

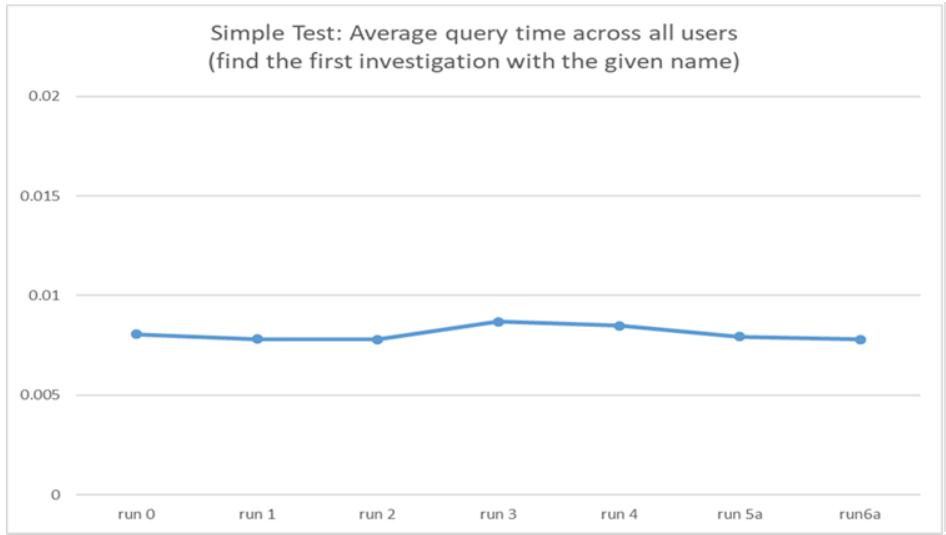
  Contains 4 tests:
  - find the first investigation with a given name
  - find the first 50 datasets for the given investigation ID
  - find the first 50 datafiles for the given dataset ID
  - find a single datafile with a given ID

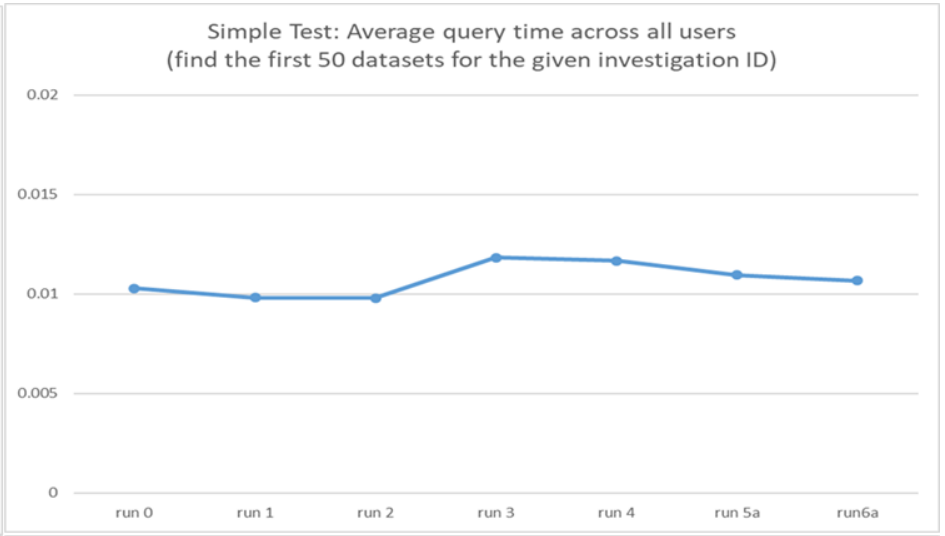
- A “long query” ordering the datafiles in a dataset by creation time and returning the first 50

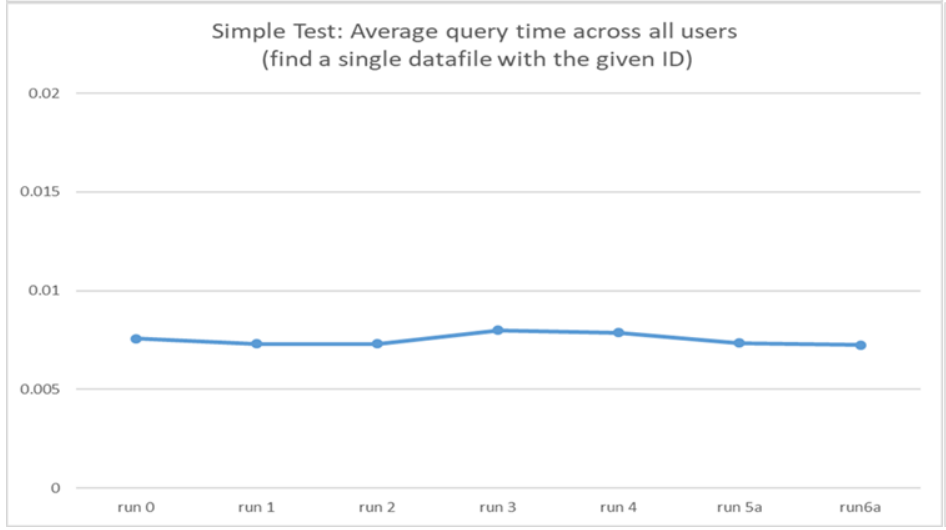
Science and Technology Facilities Council

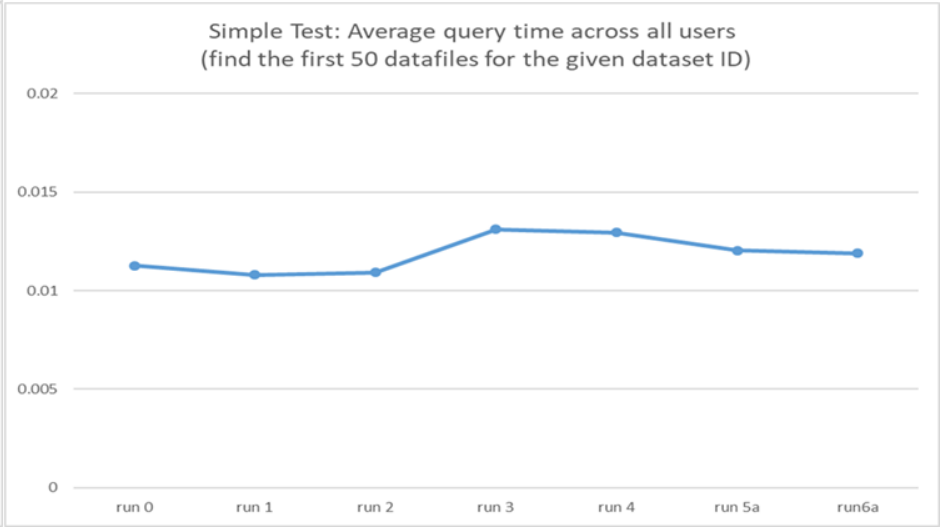
Terminology:
Investigation (ICAT) = Proposal (SciCat) = Visit (Diamond)

Averaged timings across all 90 users. 100 timings done for each user with the 10 furthest from the mean being removed.









Science and Technology Facilities Council

ICAT Simple test: results

Simple test: results analysis

- Response time falls on runs 1 and 2 then rises noticeably before falling gradually from runs 3 to 6
- The rise is most likely due to the database moving nodes between runs 2 and 3
- Ignoring the rise, the average fall in response time was around 2.5% per year
- Overall, response times got faster!

Science and Technology Facilities Council

Science and Technology Facilities Council

ICAT Simple test: results by user type

Simple test: user type results analysis

- Queries run faster for read-all users
  - ~10-15% faster when a single result is returned
  - ~30-40% faster when 50 records are returned

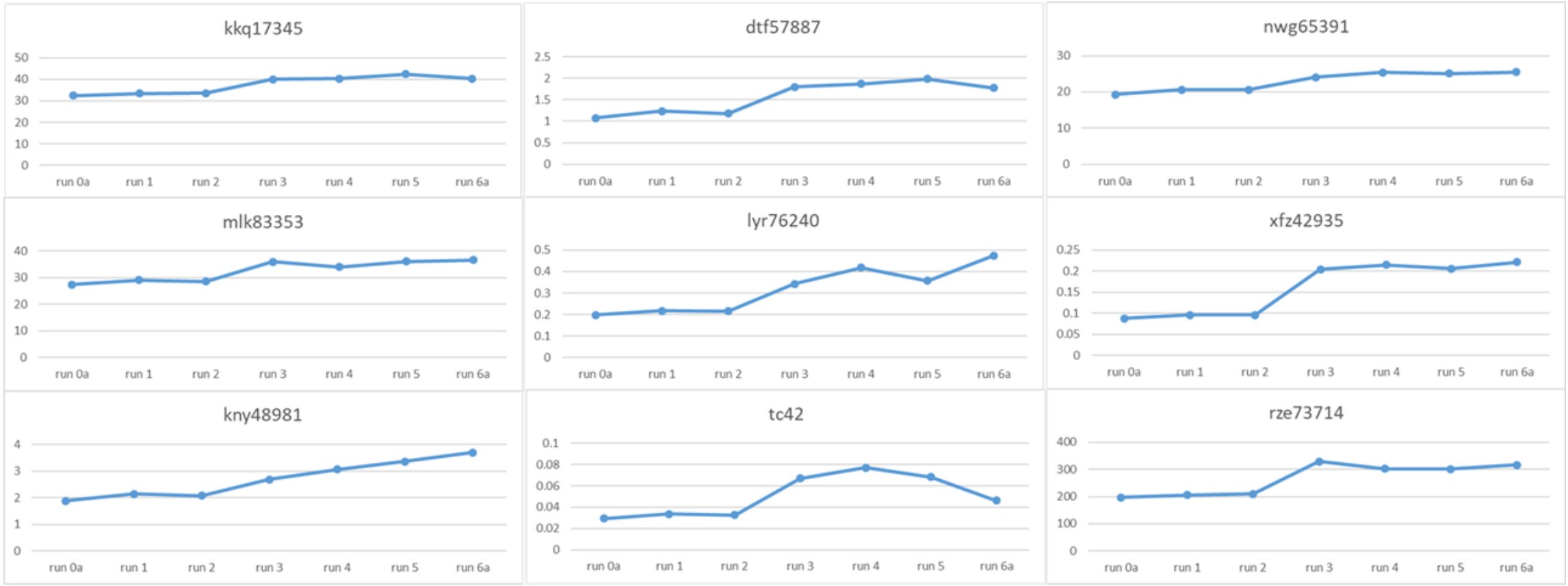
- This is due to a simpler rule being used for “read-all” than for InstrumentScientists and InvestigationUsers which require multiple table joins

- Queries run slightly faster for Instrument Scientists than for Investigation Users

“Long query” test

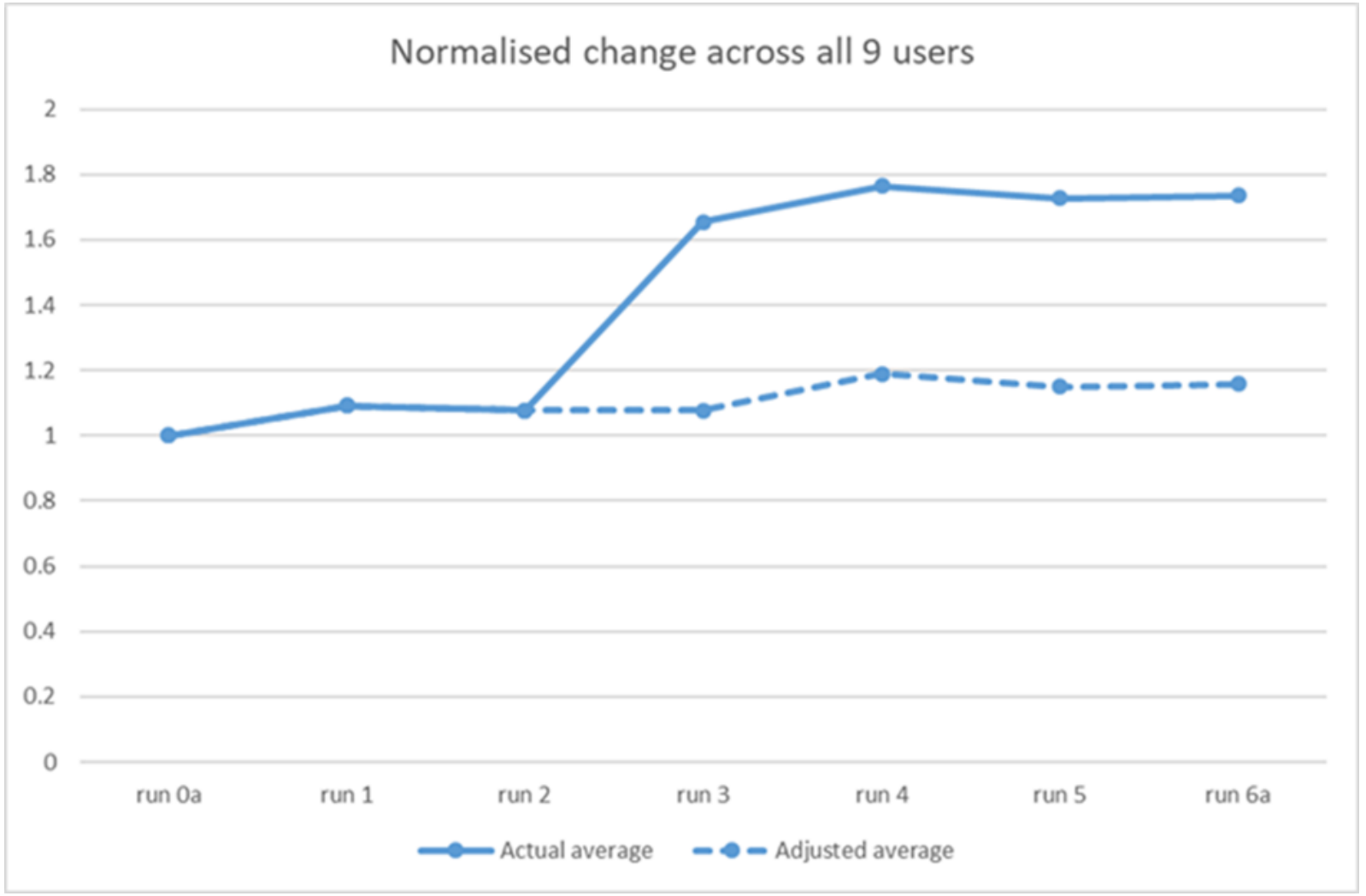
(Order the datafiles in a dataset by creation time and return the first 50)

- Took much longer to run than expected
- A full set of results was only collected for the first 9 users



Science and Technology Facilities Council

Long query: 9 users results



Science and Technology Facilities Council

Long query: normalized and averaged results

Long query results analysis

- The rise between runs 2 and 3 was more pronounced than in the simple tests: ~50% rise compared to 10-20%
- Adjusting for the rise between runs 2 and 3, the overall rise over the other runs was 16%, equating to ~3% per year
- The rise was not consistent. There were 2 runs where there was actually a fall in the time taken to run the queries.

Overall conclusions (ICAT)

- For “simple” queries returning small amounts of unordered data, performance improves slightly as more data is added. This is not fully understood but may be due to Oracle caching or improving execution plans.

- For longer running queries requiring ordering of a large number of rows, performance does not degrade significantly as more data is added.

- In both cases the change was only a few percent per year of data added, which over the next 5-10 years should not be a concern.

SciCat Performance Testing

Aim:

- To try to repeat the same performance testing that was done for ICAT and compare the results.

- During initial testing of SciCat it became clear that this was not going to be entirely possible, so:
  - the simple test to retrieve a single proposal was repeated
  - the “long” test was not possible so all the files for a Dataset were retrieved (it is not possible to sort, filter or paginate Datafiles in SciCat)

SciCat Limitations

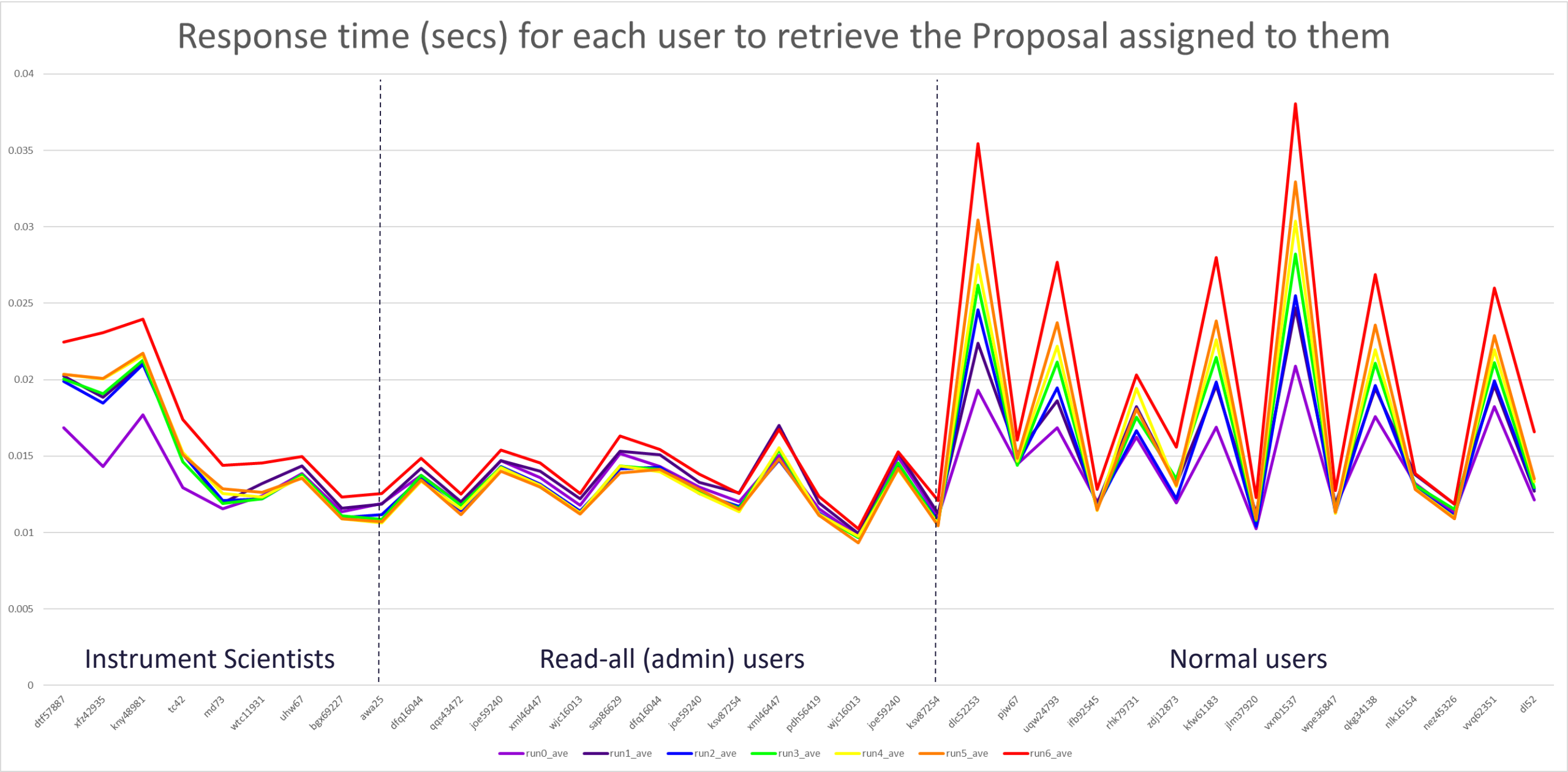
A couple of limitations were found:

- A limit of around 3000 Proposals per user (dependent on the length of Proposal names).
  For Diamond at the time, the top 3 users had just under 2000 Proposals.

- A limit of around 100,000 Datafiles per Dataset above which the request times out and no data is returned.
  In the Diamond ICAT there many Datasets with more than 100,000 files – and yes, we know this is not good!

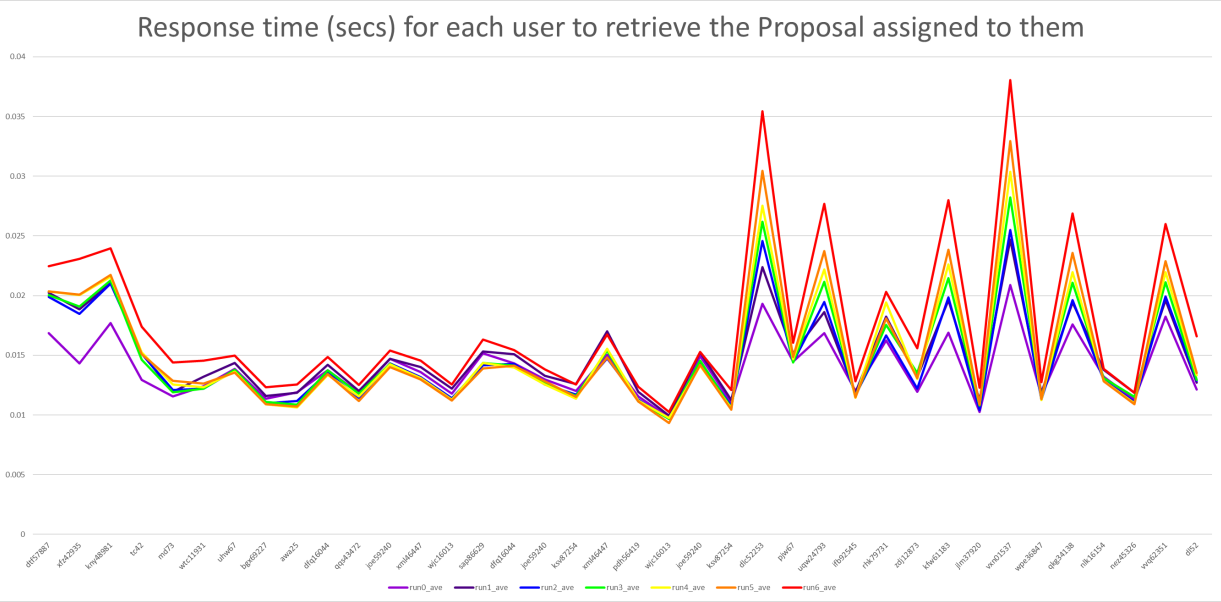
Proposal testing results

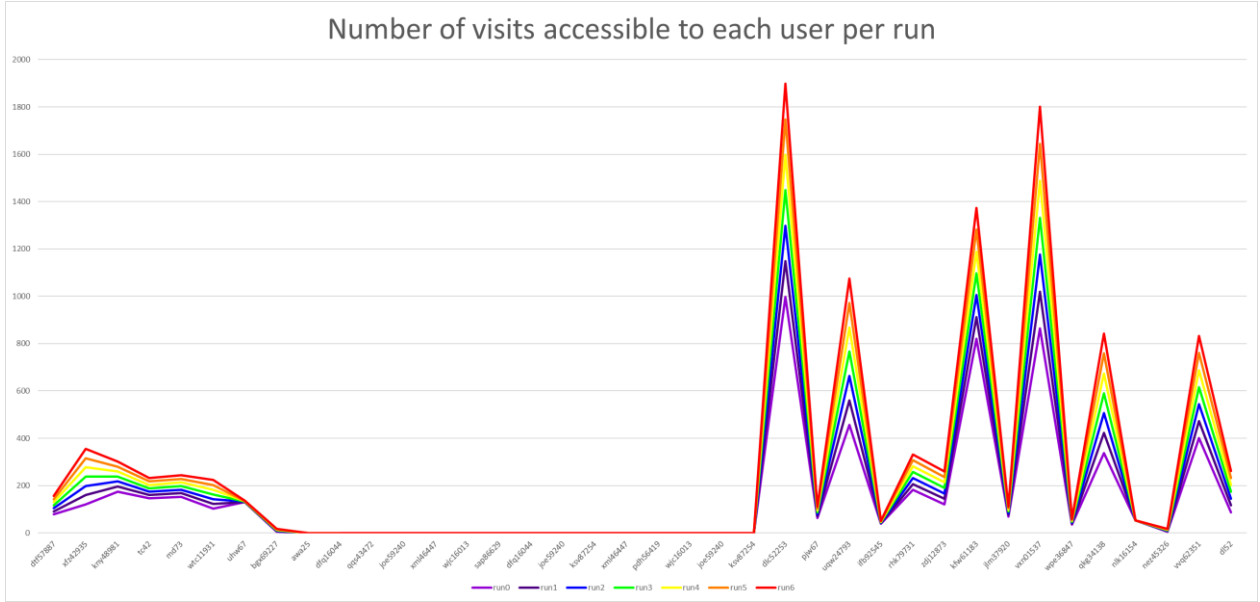
(Lightweight query retrieving a small amount of metadata about the Proposal)



Proposal testing results

- Response time correlation with the number of visits accessible to the user

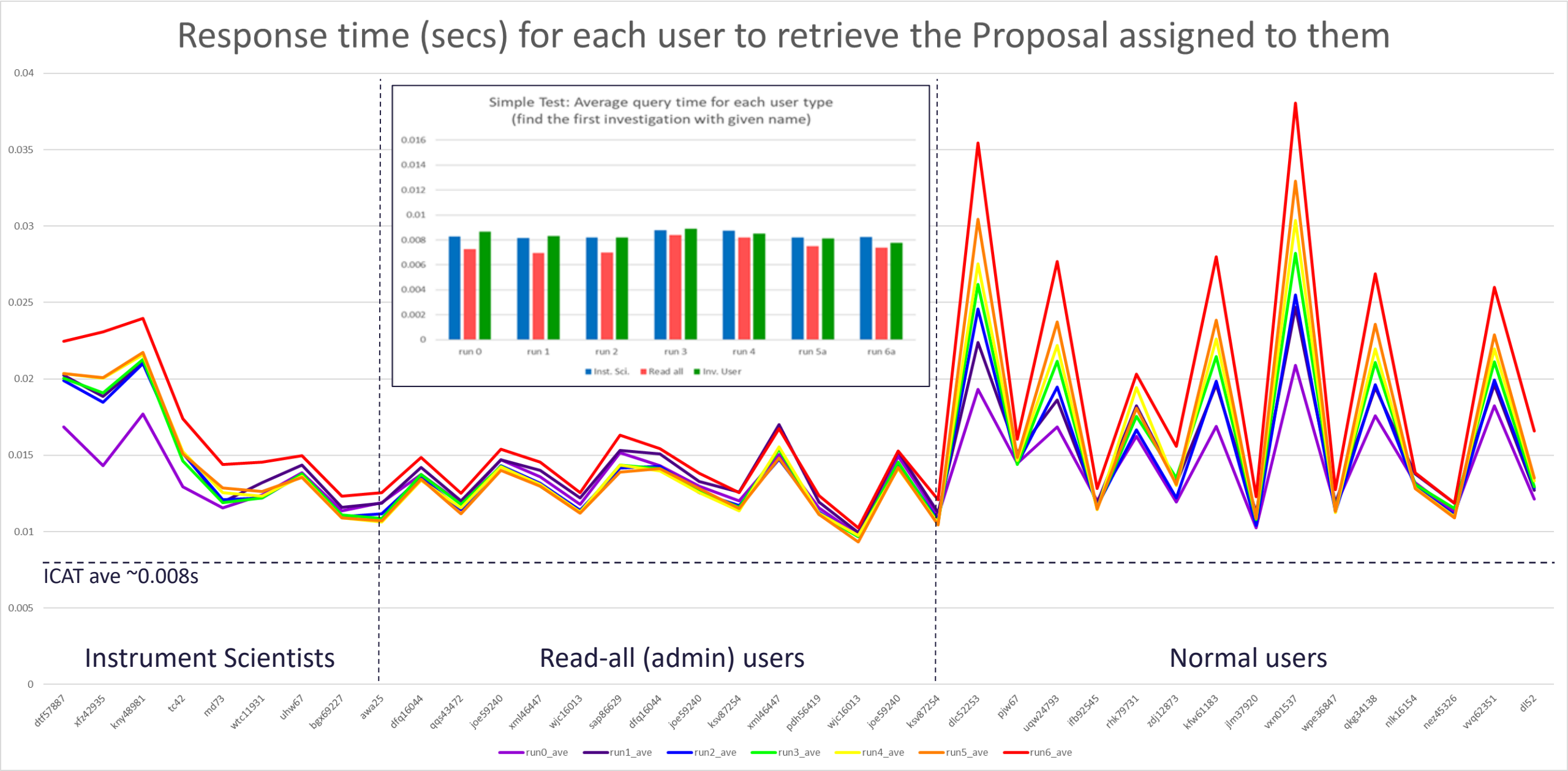




Science and Technology Facilities Council

Proposal testing results

(Same graph but with ICAT data added for comparison)



Proposal test results

- There seems to be a direct correlation between the number of visits a user is on and response times.
  - Users on more visits experienced correspondingly increased response times compared to other users accessing the same data.
  - A doubling of the number of visits for a user resulted in roughly double the response time to access the same data.

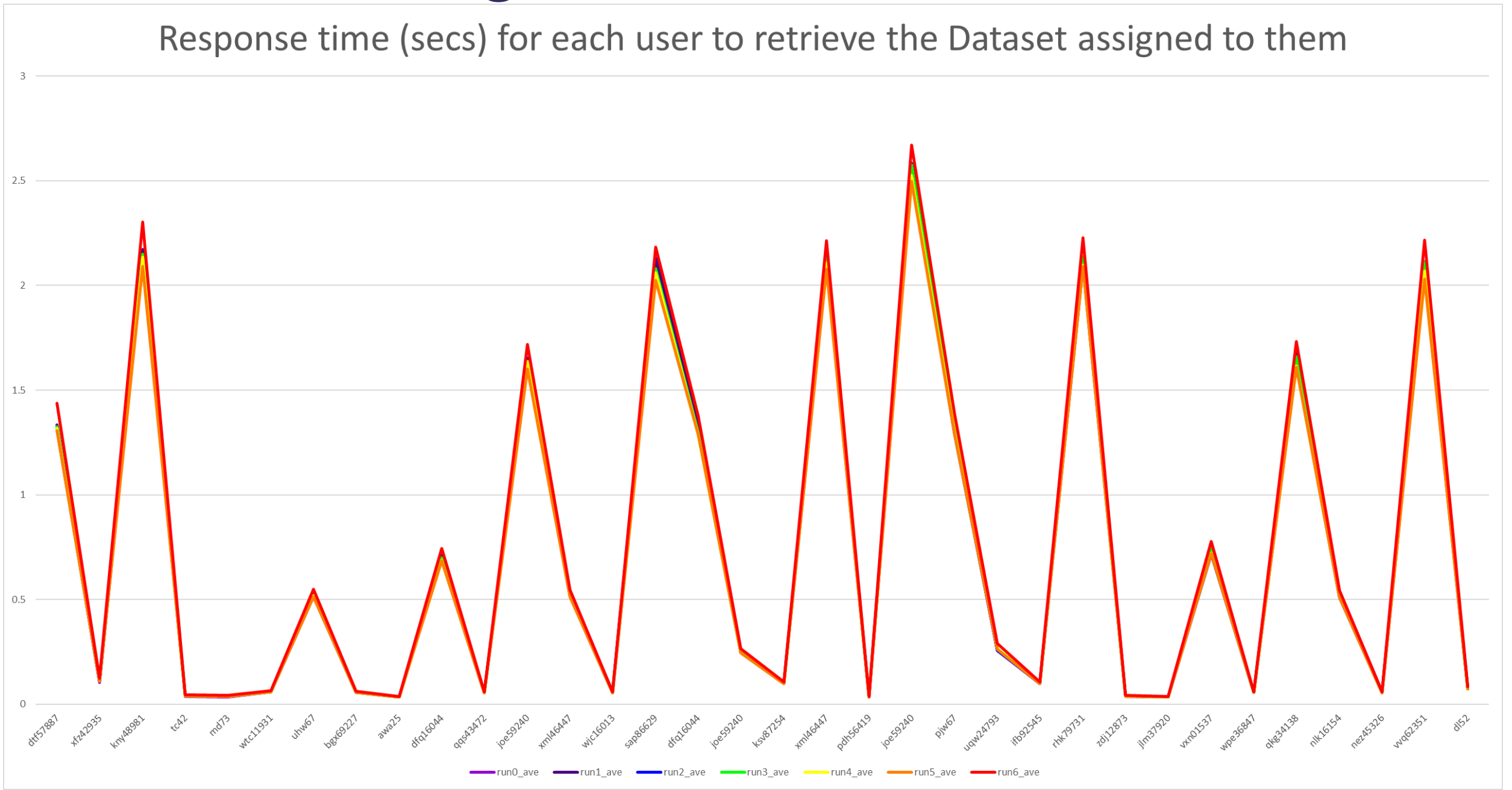
- Response times in the equivalent ICAT test were quicker with an average around 8ms whilst most SciCat response times started between 10 and 20ms and rose significantly for some users.

- Underlying response time increase due to data growth was low at around 5% over the 6 year period. For ICAT the response time *fell* by around 15%.

- The authorisation mechanism based on visit groups does not scale very well (for small queries), adding an (additional) average increase in response time of 35% for “normal” users.

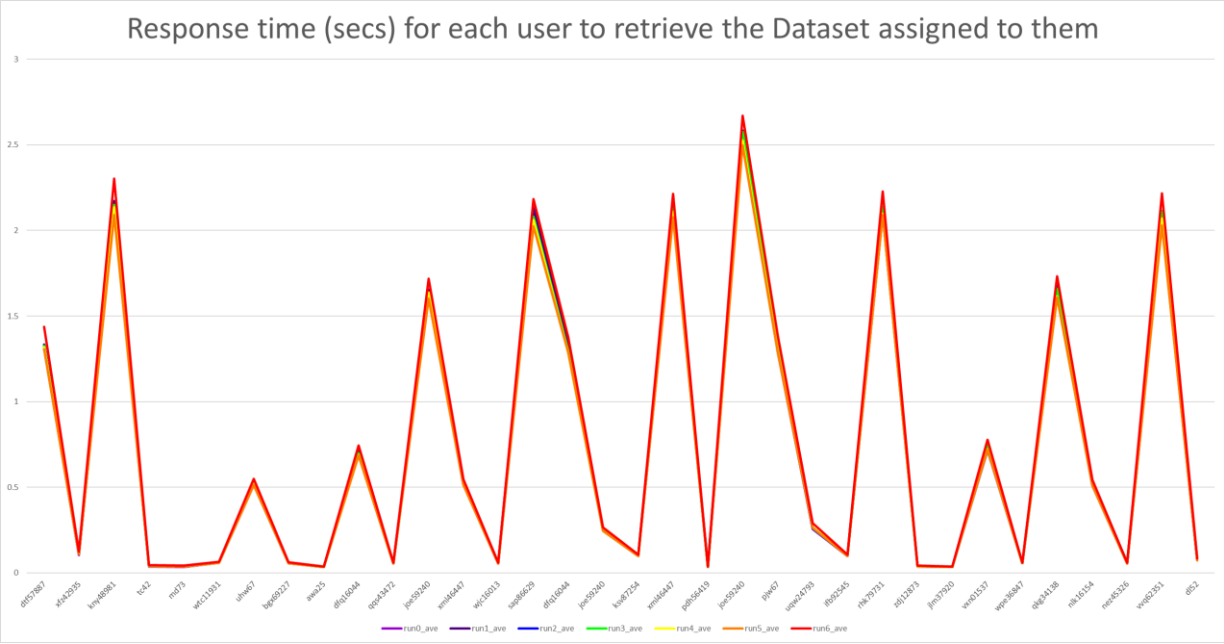
Dataset testing results

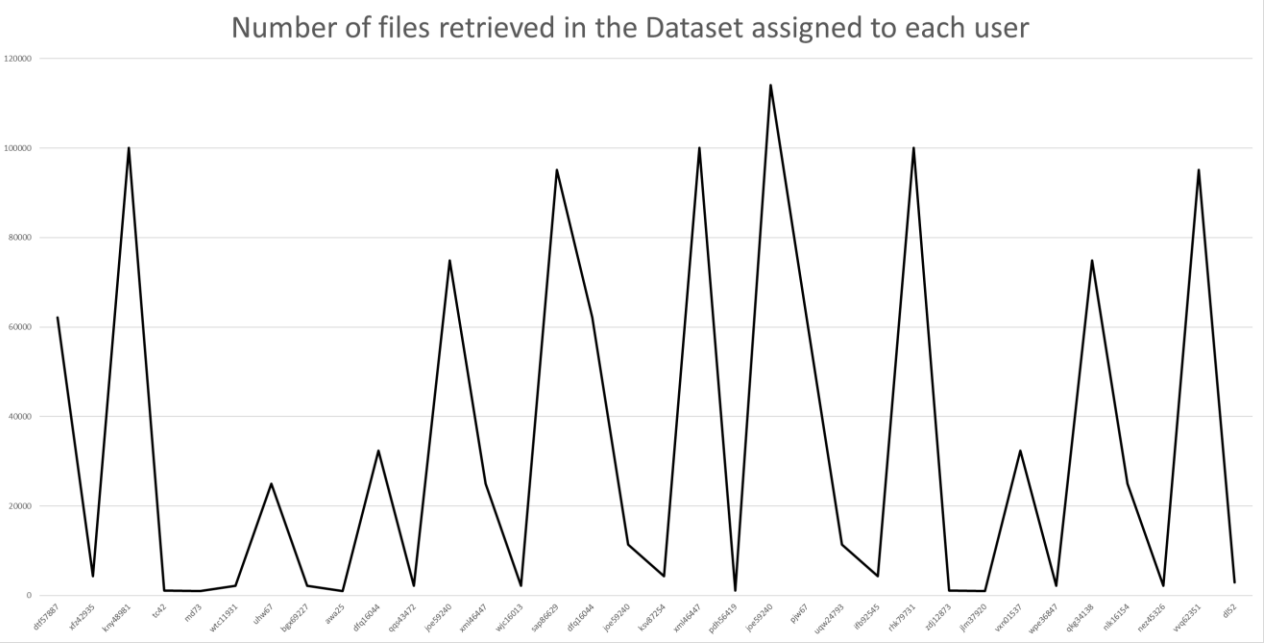
(Heavy query retrieving the Dataset metadata and a list of all the files – as the front end does)



Dataset testing results

- Response time correlation with the number of files in the Dataset retrieved





Science and Technology Facilities Council

Dataset test results

- The response times hardly changed for each user throughout the runs.
- The increase due to the number of visits was still measurable but with the longer response times, became negligible in the results.
- The SciCat backend (Catamel) and database appear to handle the increasing data size well with around a 5% increase in response times seen over the 6 year period (in line with increase for Proposal test).
- The backend also appears to be able to return lists of files from a Dataset very quickly.
  A very repeatable rate of nearly 50,000 files per second was measured.
  From ICAT/Oracle the rate is likely to be more like 10,000/sec.
  Note that for most purposes, particularly a web GUI, returning this many results would be impractical.

Conclusions from the performance testing

- Overall the SciCat results were more repeatable and predictable than the results of the equivalent ICAT testing.
- There was more variation in the ICAT results, some of it unexplainable.
- In the Proposal test, where it is easiest to compare ICAT and SciCat results, ICAT recorded faster response times, but differences in testing setups could account for this.
- The testing revealed some examples of the overhead that both catalogues add to perform authorisation.
- SciCat's inability to sort, filter and paginate Datafiles was a concern for Diamond where we have so many historic Datasets containing thousands of files where this would be essential.

Main reasons for keeping ICAT for Diamond

- Our Database Services team did not provide MongoDB databases, nor did we have any MongoDB experience in the group
- We still need to support ICATs for ISIS and the Central Laser Facility at RAL so a move to SciCat would mean the group would need to either support both, or work towards migrating those facilities as well.
- The DataGateway project was already underway to become our new frontend for ICAT
- SciCat was not “production ready” at the time. We found a number of bugs whilst testing.
- A requirements analysis revealed that ICAT met about 50% and SciCat about 40%
- The Diamond archive system has a lot of tools in place and working for 10+ years.
  Many of these would need re-writing to work with SciCat.
  In particular, there was not the effort available in Diamond to support this.

UKRI Science and Technology Facilities Council

Questions?

UKRI Science and Technology Facilities Council

Thank you

Science and Technology Facilities Council @STFC_matters Science and Technology Facilities Council