Scientific Computing

Developing a new search index

Alan Kyffin
Data and Software Engineering Group
STFC Scientific Computing

Current situation

The index

- icat.lucene - free-text search component

- Supports:
  - Synonyms
  - Faceting
  - Unit conversion

- Single-threaded and cannot be scaled

Indexing

- Done by icat.server in background threads
- Index ‘schema’ defined in icat.server code
- ‘Schema’ is structured, leading to nested indexes
- Full re-indexing takes weeks

Searching

- Searches go via icat.server (/icat/search/documents)
- Gets documents from icat.lucene
- Authorization: documents IDs then checked against rules

The future

The index

- ElasticSearch

Indexing

- New component separate from icat.server
- Flat ‘schema’
- (Hopefully) full re-indexing possible in days not weeks

Indexing

- Performance difference between icat.server JPQL and database SQL?

- What should the ‘schema’ be?
  - What needs indexing?
  - or rather, what do users search for?

- How to know what needs indexing?
  - Periodic query by mod_time
  - Message on message queue

Searching (authorization)

- Another new component (search-api?)
- Need to re-implement ICAT rules in the search-api
- Fortunately, they are straightforward:
  - is the user an InvestigationUser for the investigation
  - or are they an InstrumentScientist for the instrument used
  - or is this public data
- Get all the user's investigation and instrument IDs beforehand and add a filter term to the search

Searching (authorization)

- Get all the user's investigation and instrument IDs beforehand and add a filter term to the search
- User already requires a JWT from DataGateway
- Do a query when they log into DataGateway and add them to their JWT

Science and Technology Facilities Council

Scientific Computing

Questions?

Science and Technology Facilities Council

Scientific Computing

Thank you

sc.stfc.ac.uk