

Search Engines, Applications & Metadata Harvesting

Patrick Dowler







A Search Engine?

- search engine for data services (SIA, SSA, etc)
- a value-added service or application that sits on top of DAL services
- intent: harvest metadata only
- delegate data delivery back to original DAL service



A Search Engine?

- employs custom indexing to enhance performance of certain classes of queries
 - non-standard conditions
 - non-standard functions
 - non-standard or very expensive join semantics
- value-added results
 - consistent scoring valued by users (i.e. google rank)
 - non-standard attributes, metrics, etc.
 - intregration with other datasets, derived products, special tools



Use Cases

- Multi-Wavelength Analysis
 - search for sets of overlapping images with members in subgroups
 - subgroups typically defined using spectral coverage
 - used to study energy-dependent features
 - example: find sets of UBVRI data for photometric redshifts
- requires: spatial and spectral coverage
- status: prototype developed
 - custom index creation is very time consuming
 - queries are quite fast but not yet fully optimised



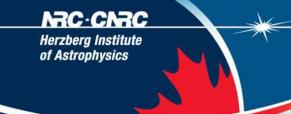
Use Cases

- Time-Dependent Analysis
 - search for sets of overlapping images with a specified timesampling pattern
 - used to find/analyse variable and moving objects
 - > example: find sets of V-band data with 5 images taken 1 day apart
- requires: spatial and time coverage, spectral useful
- status: protype developed
 - custom indexing was quick and dirty
 - queries hard to formulate
 - query performance is seriously inadequate



Use Cases

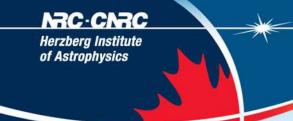
- Trajectory Search
 - search for images that lie along a trajectory
 - used to recover moving solar system objects and refine orbit
 - example: candidate has two positions with errors and times, find more images that might contain the moving object
 - example: candidate is a trail on a single image, find more images that might contain the moving object
- requires: spatial and time coverage, spectral and SNR useful



Summary of Use Cases

- the use cases generally rely on having good Coverage. Bounds in all three axes
 - for SSA spec (7.5.5.1) the Coverage.Bounds.Spatial is optional and one may have to derive it from the Coverage.Location.Spatial and the Aperture
- proposal: make Coverage.Bounds.Spatial mandatory

Note: In SIA 1.0, the WCS params are optional and hardly any services actually supply them...



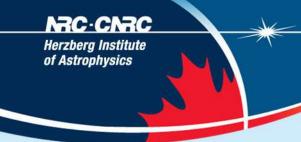
Harvesting Issues

- initial loading of metadata
 - want to minimise load on services, handle large query results
- incremental update of metadata
 - * this will happen often (polling), must really minimise the load
- completeness
 - want harvester to be confident of completeness
 - avoid regular full scan (never, or at least not very often)
 - avoid multiple redundant queries to ensure completeness



Harvesting

- initial loading & large query results
- possible solutions
 - batch query results: INDEX
 - not generally robust for very long queries, especially if the source is also changing
 - > requires service to deliver results in a consistent order
 - query by timestamp, order by timestamp, top: SINCE and TOP
 - requires sorting by service, requires another param or overload TOP
 - query by timestamp range, tune bounds to manage result size
 - puts onus on client to tune queries and be nice



Harvesting

- incremental updates
 - this would be run regularly (daily? weekly?)
 - need to find new metadata/data and modified metadata
 - does not seem necessary for harvester to know data changed

• solution:

- query by timestamp: SINCE sufficient unless you fall behind
- query by timestamp range
 - same as SINCE if omit upper bound
 - could tune bounds to avoid truncation if you fall behind



Harvesting

- completeness
 - need to be able harvest all metadata and terminate
 - ultimately, a harvesting query needs to return no (new) results

• issues:

- truncation of query results needs to be explicit in result
- client needs to make progress in the imperfect internet
 - client needs to be able to process some entries and resume after a failure with minimal redundancy



Harvesting Summary

- current SSA query params:
 - INDEX allows one to step through large result, but cannot ensure completeness without excessive burden on service
 - SINCE is sufficient for frequent incremental updates, but may suffer from truncation and is inadequate for bootstrapping
- proposal: range query on modification time (Curation.Date?)
 - METATIME=[t1]/[t2]
 - service must explicitly say when result is truncated
 - make the query param and result value mandatory



A Small Detail...

- search engine will harvest Curation.CreatorID and will want to delegate data retrieval back to original service, which requires a query on one of these to get the AccessRef that is correct
 - SSA spec (7.4.2) says CREATORID query param is optional
 - SSA spec (7.5.3) says Curation.CreatorID is required output
- proposal: make CREATORID a required query param



Summary

- creation of search engines and other value-added applications
 - generally requires complete Coverage. Bounds metadata
- harvesting metadata requires handling or avoiding large query results, incremental updates, and a way to ensure completeness
 - range query on metadata modification time is simplest solution
 - modification time must be mandatory query output