Authority Aggregation and Indexing

Overview

The University of Iowa primarily worked on the design and implementation of a support infrastructure layer for an eventual ecosystem of Linked Open Data servers and systems. Given the somewhat immature nature of currently deployed LOD resources (e.g., offline SPARQL endpoints), the project decided that it was advisable to deploy our own services for the various LOD resources. This allowed us to

- make reasonable assumptions regarding resource availability,
- control performance characteristics, since we controlled the hardware, and
- control the nature of the data returned to queries, since we controlled the software.

The work done regarding this last point includes tuning the rank order of results, the specificity of what comprised a match to a user query, and what data were returned. In particular, we were able to inject an additional triple indicating a particular entity's rank in the results - something not present in the underlying triplestore.

For examples of integration of these services into other elements of the project, please see Architecture for Authority Lookup.

Authorities

Our deployment process became regularized to the extent that a number of authority sources were included:

- Agrovoc (agricultural concepts)
- DBpedia (general knowledge)
- FAST (general subject headers from OCLC, derived from LoC subject headers)
- GeoNames (places in the real world)
- Getty (content relating to artistic works)
    - AAT (concepts)
    - ULAN (persons and organizations)
    - TGN (places)
- Library of Congress
    - Genre
    - Name
    - Subject
- MeSH (NLM medical subject headings)
- NALT (National Agricultural Library Thesaurus)
- VIAF (authority cross-walks)

All of these services, including versions supporting human interaction with the results, are available at http://services.ld4l.org/ld4l_services/index.jsp. Direct human exploration of the various triplestores using SPARQL is available at http://services.ld4l.org/fuseki/.

Request Parameterization

To simplify both the creation of new services and the understanding by developers of applications consuming these services, we standardized the parameters accepted by the various services as much as possible:

- query - the string containing the term(s) on which to search. *Note that the Lucene tag library supports 'or'ing discrete terms (the default) or 'and'ing them (where each term in the query must appear), as well as explicit and and or boolean operators* (required)
- maxRecords - the maximum number of entity URIs to return (optional)
- startRecord - the position in the result list to begin returning records (this used with maxRecords allows for result pagination) (optional)
- entity - where relevant for a given authority source (e.g., DBpedia), the class of the entity URIs to be returned (optional)

Hence the following query - *http://services.ld4l.org/ld4l_services/getty_batch.jsp?query=Picasso&maxRecords=10&entity=Person* - will return the triples relevant to 10 entities of class Person (i.e., from the Getty ULAN authority) where the word Picasso appears. Note that the actual number of triples return can vary widely due to differences in coverage between entities, even within a single authority source.

Technology Stack

The overall architecture was implemented entirely with open source tools:

- Apache HTTPD - the standard v. 2.4 web server deployed with macOS
- ld4l_services - this is a Java Server Pages (JSP) application (available at https://github.com/eichmann/ld4l_services) heavily reliant on two JSP tag libraries:
    - LuceneTagLib - a wrapper for executing Lucene full text searches and accessing the results of those searches (available at https://github.com/eichmann/LuceneTagLib)
    - SPARQLTagLob - a wrapper supporting SPARQL queries from a JSP page in the same manner used to access relational databases using the SQL standard tag library (available at https://github.com/eichmann/SPARQLTagLib)
- Apache Tomcat application container - we specifically are using version 9.0.0.M9, although pretty much any version of Tomcat would work, as we're not using an particular features of this version.
- Apache Jena Fuseki - the SPARQL endpoint, version 2.4.0
- Java SE Runtime Environment - version 1.8.0

Processing Flow

- a request arriving at services.ld4l.org is routed to one of two redundant application servers (see the server configuration discussion below)
- the relevant JSP page runs a Lucene query, receiving back a set of entity URIs specific to the particular authority
- for each entity URI, the JSP page constructs a SPARQL query and submits it to Fuseki (using the virtual host name to allow load balancing)
- Fuseki executes the SPARQL query and returns a set of RDF triples
- the JSP page returns the triples to the requesting site, injecting synthesized triples representing rank into the result corresponding to the entity URI's position in the Lucene search results

Server Configuration

- Mac Pro (late 2013), 3 GHz, 8 cores, 64 GB memory, macOS High Sierra (v. 10.13.6)
- Promise Pegasus2 disk array, 8x4tb RAID5, Thunderbolt2 connection to the Mac Pro

Two equivalent configurations were deployed, each with full copies of the LOD on the disk array. An Apache virtual host configuration was used to both manage the services.ld4l.org domain configuration and to configure the two machines using Apache's balancer feature to identify the first machine as the primary service provider with the second machine as a "hot spare." Each instance of the ld4l_services application access the data using the virtual host name, providing redundancy both in the application and in query processing. Adding additional BalanceMembers is trivial and provides a significant ability to scale overall capacity.

```
<VirtualHost *:80>
    ServerName services.ld4l.org
    ServerAdmin david-eichmann@uiowa.edu
    DocumentRoot "/Library/WebServer/LD4L-Documents"
    <Proxy "balancer://fuseki">
        BalancerMember "http://localhost:3030"
        BalancerMember "http://deep-thought.slis.uiowa.edu:3030" status=+H
        ProxySet lbmethod=byrequests
    </Proxy>
    <Proxy "balancer://tomcat">
        BalancerMember "http://localhost:8080"
        BalancerMember "http://deep-thought.slis.uiowa.edu:8080" status=+H
        ProxySet lbmethod=byrequests
    </Proxy>
    RewriteEngine On
    RewriteRule ^/fuseki$ fuseki/ [R]
    ProxyPass "/fuseki" "balancer://fuseki" stickysession=JSESSIONID
    ProxyPassReverse "/fuseki" "balancer://fuseki"
    ProxyPassMatch "^/.*" "balancer://tomcat" stickysession=JSESSIONID
    ProxyPassReverse "/" "balancer://tomcat"
    <Directory "/Library/WebServer/LD4L-Documents">
        Options FollowSymLinks Multiviews
        MultiviewsMatch Any
        AllowOverride None
        Require all granted
    </Directory>
    ErrorLog "/private/var/log/apache2/ld4l-error_log"
    CustomLog "/private/var/log/apache2/ld4l-access_log" combined
</VirtualHost>
```

A Complete List of GitHub Repositories Related to the Project

- ld4l_services (https://github.com/eichmann/ld4l_services) - A web app providing name lookup and triple extraction from a number of cached sources, including DBpedia, FAST, GeoNames, GRID, and VIAF.

- biblio (https://github.com/eichmann/biblio) - Prototype interface for semantic library catalog data
- BIBFRAMETagLib (https://github.com/eichmann/BIBFRAMETagLib) - JSP Tag library providing access to a local cache of BIBFRAME data
- fast (https://github.com/eichmann/fast) - Application scaffold for FAST data
- FASTTagLib (https://github.com/eichmann/FASTTagLib) - JSP Tag library providing access to a local cache of FAST data
- geonames (https://github.com/eichmann/geonames) - Application scaffold for GeoNames data
- GeoNamesTagLib (https://github.com/eichmann/GeoNamesTagLib) - JSP Tag library providing access to a local cache of GeoNames data
- viaf (https://github.com/eichmann/viaf) - Application scaffold for VIAF data
- VIAFTagLib (https://github.com/eichmann/VIAFTagLib) - JSP Tag library providing access to a local triplestore with VIAF data
- LuceneTagLib (https://github.com/eichmann/LuceneTagLib) - A JSP tag library supporting access to Lucene indices from JSP.
- SPARQLTagLib (https://github.com/eichmann/SPARQLTagLib) - A JSP tag library providing functionality roughly equivalent to the JSTL SQL tag set, just for a triple store.