MAKING TEMPORAL SEARCH MORE CENTRAL IN SPATIAL DATA INFRASTRUCTURES

P. Corti[a], B. Lewis[a*]

[a] Center for Geographic Analysis, Harvard University, Cambridge MA USA

KEY WORDS: data discovery, geoportal, metadata, search engine, Spatial Data Infrastructure, SDI, spatio-temporal, WorldMap

ABSTRACT:

A temporally enabled Spatial Data Infrastructure (SDI) is a framework of geospatial data, metadata, users, and tools intended to provide an efficient and flexible way to use spatial information which includes the historical dimension. One of the key software components of an SDI is the catalogue service which is needed to discover, query, and manage the metadata. A search engine is a software system capable of supporting fast and reliable search, which may use any means necessary to get users to the resources they need quickly and efficiently. These techniques may include features such as full text search, natural language processing, weighted results, temporal search based on enrichment, visualization of patterns in distributions of results in time and space using temporal and spatial faceting, and many others. In this paper we will focus on the temporal aspects of search which include temporal enrichment using a time miner - a software engine able to search for date components within a larger block of text, the storage of time ranges in the search engine, handling historical dates, and the use of temporal histograms in the user interface to display the temporal distribution of search results.

1. INTRODUCTION

A Spatial Data Infrastructure (SDI) (Groot and McLaughlin, 2000, Infrastructures, 2004, Masó et al., 2012, Rajabifard et al., 2009) is a framework of geospatial data, metadata, users and tools intended to provide an efficient and flexible way to use spatial information. An open and interoperable SDI exposes its search capabilities to other systems using open standards for any client to access (Kralidis, 2009). In an Open Geospatial Consortium (OGC) compliant SDI, users can query metadata using the Catalogue Service for the Web standard (CSW) to search for layers (Consortium, 2017a), and the Web Feature Service standard (WFS) to search for features (Consortium, 2017b).

SDI search can be dramatically strengthened by adding a search engine to the stack. A search engine is a software system capable of supporting fast and reliable search which may use any means necessary to get users to the resources they need quickly and efficiently. Techniques may include features such as full text search, natural language processing, weighted results, fuzzy tolerance results, faceting, hit highlighting, recommendations and many others. Harvard WorldMap, a public SDI containing thousands of geospatial layers and maps (Guan et al., 2012), has added a search engine called HHypermap (Paolo Corti, 2017) based on Solr/Lucene to its GeoNode-based software stack (Benthall and Gill, 2010). This provides a restful API on top of the OGC standards, enabling enhanced features provided by the search engine (Smiley and Pugh, 2009).

Given a large audience of historians within the academic community, WorldMap needs to provide a way for users to search layers and features on dates and date ranges. Dates can be provided as a specific date field when compiling metadata or can be detected and extracted by parsing the text contained in the layers metadata and feature string fields by using regular expressions. For this purpose, the search engine index is synced with the SDI geodatabase using tasks in a task queue. The RESTful search engine API can be then used to do powerful searches based on dates and date ranges. It is also possible to support temporal histograms in the user interface of the system using the date faceting feature of the search engine.

*Corresponding author

2. FEATURES OVERVIEW

WorldMap (http://worldmap.harvard.edu/) is a geospatial content management system and public SDI which makes it possible for registered users to publish geospatial content on the web. Thanks to its open source software stack based on GeoNode, users can upload geospatial vector and raster data, and combine it with existing geospatial datasets to create map collections. Data and related metadata can be managed, styled, edited, and queried using a user interface based mostly on OGC standards such as WFS, WMS, SLD, CSW and others. Furthermore users can restrict access to data to specific users or group of users.

The information contained in WorldMap is constantly growing, and at the time of writing the system is being fed by more than 20,000 registered users, who have uploaded more than 25,000 layers and created more than 7,000 maps. Furthermore, the HHypermap Registry component of WorldMap provides access to more than 13,000 map web services based on OGC and Esri REST protocols, for a total of almost 200,000 remote layers being made accessible. HHypermap is still being developed, with the goal being to comprise a comprehensive registry of public map service layers.

With such a volume of information, WorldMap needed an efficient, fast and reliable way to perform search on the published geospatial information. To provide this, a search engine based on Solr/Lucene has been added on top of the stack. All of the layers metadata in Postgres RDBMS are kept in sync with the Solr search engine index using a task queue asynchronous approach based on Celery/RabbitMQ (Ellis et al., n.d.).

The API and the search client developed for it allows users to search layer metadata using depict-dates and date ranges. To this purpose any date contained in datefield metadata attributes

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-4-W2-93-2017 | © Authors 2017. CC BY 4.0 License.

are synced to the search engine. Furthermore, as text metadata fields contain dates embedded in text, the system tries to detect those dates using an approach based on regular expressions, a date miner, formats them and loads them to the search engine. This provides the basis for search using depict-date ranges to discover map layers (Figure 1).

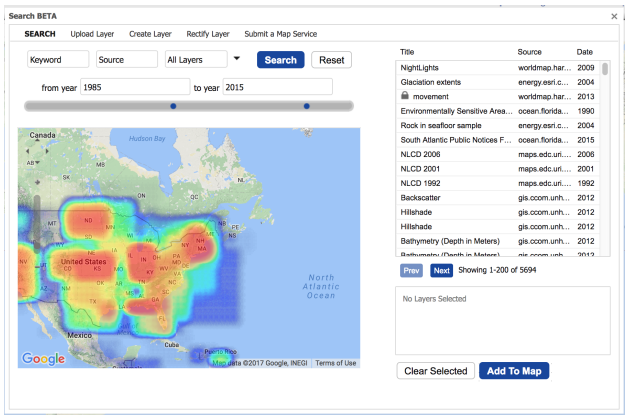

Figure 1. Search using depict-date ranges to discover map layers in an SDI

Solr is a very powerful platform for performing date algebra and date range search. It can also be used on date fields to return faceted results very quickly, in order to provide a list of layers (and features) which have a depict date falling in a given interval of time. Temporal faceting will be eventually used to provide a temporal histogram of the returned results, as has been implemented in the Billion Object Platform (BOP), another geospatial search platform developed by the CGA (Figure 2).

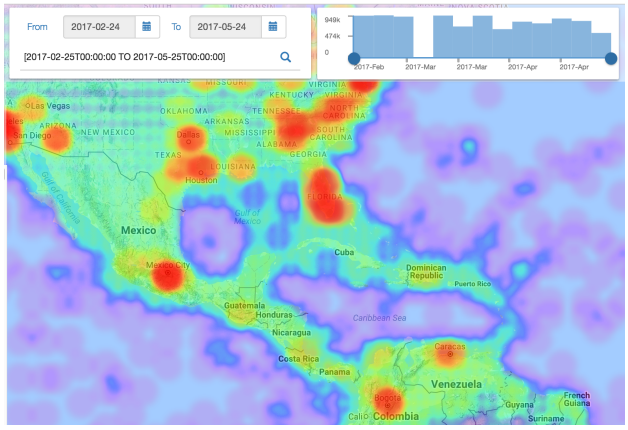

Figure 2. Using temporal faceting to provide temporal histograms

3. DATE INDEXING APPROACH

WorldMap users typically provide details about the depict date and the temporal extent of a given geodataset in one of the following ways:

- Using dates in metadata date fields, such as temporal extent start and end date
- Using dates in the geodataset title, abstract and other text fields

A similar approach could be used (in future) to support feature level search for features in a vector geodataset:

- Date field values for each feature of the dataset
- Using dates in the text fields of the dataset

The HHypermap Registry component of WorldMap scans all of the dates in the metadata date fields and adds them to the date database for a given dataset. Registry also parses metadata text fields in order to mine dates from text. This mining is performed using an approach based on regular expressions.

Registry detects changes in metadata fields and send a task for each service/layer to the task queue. For each service/layer a task in the task queue is responsible for metadata parsing and another task for keeping in sync the Solr database with the relational database, based on PostgreSQL.

4. ARCHITECTURE OF THE SYSTEM

Both Worldmap and HHypermap Registry are based on a open source stack. WorldMap is based on GeoNode, which is developed on top of Django, PostgreSQL/PostGIS, GeoServer, GeoNetwork/pycsw, GeoExplorer.

HHypermap Registry is developed on top of Django, PostgreSQL, Celery/RabbitMQ, Solr, pycsw, MapProxy and memcached. HHypermap provides at the same time a user interface, a task queue, a remote services map cache, an OGC CSW endpoint, and an application API based on a search engine (Solr) which the WorldMap search interface uses (Figure 3).

4.1 User Interface

The user interface for HHypermap provides a view of the health of the harvested web services and layers (uptime statistics), as well as tools for administrators for adding and removing web services. The health-check-harvested uptime statistics are used to provide weighted results to the end users.

4.2 Task Queue

The task queue processes a large number of tasks needed to check the health of the many services/layers, sync the index of the search engine, and collect the dates from metadata and text. Tasks may be scheduled or run on demand by the administrators.

4.3 Map Cache

All of the layers for harvested remote services are cached by MapProxy, and WorldMap uses this cache to provide instant previews for remote layers in the search tool and to provide tiles to the maps.

4.4 OGC Catalogue

The OGC CSW endpoint, based on pycsw, provides a way for other systems/clients to interact with the HHypermap database using the CSW standard.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-4-W2-93-2017 | © Authors 2017. CC BY 4.0 License.

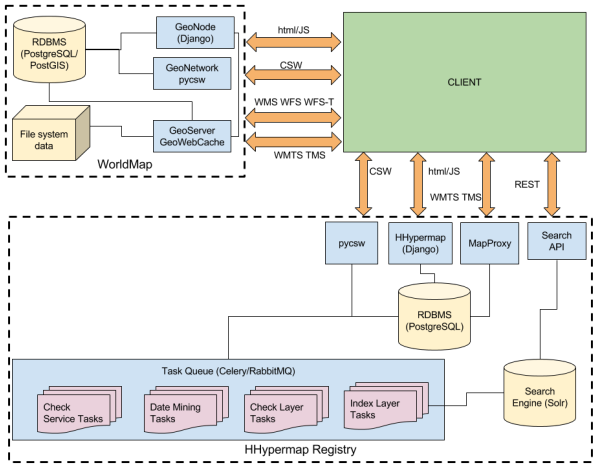

Figure 3. Architecture of the System

4.5 Application API

The registry API, which is built on top of Solr using Swagger, provides advanced search functionalities to WorldMap: users are able to locate spatio/temporal information using search engine features such as spatial and temporal facets, full text search, weighted results and much more.

5. FUTURE WORKS

Date mining from text is not trivial. The current approach used in HHypermap Registry is to run a Python script which uses regular expression to extract dates from text. Rather than using a custom approach the authors are looking to possibly integrate in the solution a specific open source library which takes care of date finding in text. A natural candidate for HHypermap, which is based on Python and uses an approach based on regular expressions, is the Python library named datefinder (Alec Koumjian, 2017). A more complex approach could be to use libraries based on Natural Language Processing.The authors are looking with particular interest at Stanford NLP (Manning et al., 2014).

REFERENCES

Alec Koumjian, e. a., 2017. datefinder. `https://github.com/akoumjian/datefinder`.

Benthall, B. and Gill, S., 2010. Sdi best practices with geonode. *Retrieved June* 1, pp. 2010.

Consortium, O. G., 2017a. Catalogue Service. `http://www.opengeospatial.org/standards/cat/`. [Online; accessed 13-April-2017].

Consortium, O. G., 2017b. Web Feature Service. `http://www.opengeospatial.org/standards/wfs`. [Online; accessed 13-April-2017].

Ellis, L., Grimes, J., Goel, A. and Scheiber, M., n.d. Bruce: A distributed code execution service.

Groot, R. and McLaughlin, J. D., 2000. *Geospatial data infrastructure: concepts, cases, and good practice*. Oxford university press Oxford.

Guan, W. W., Bol, P. K., Lewis, B. G., Bertrand, M., Berman, M. L. and Blossom, J. C., 2012. Worldmap–a geospatial framework for collaborative research. *Annals of GIS* 18(2), pp. 121–134.

Infrastructures, D. S. D., 2004. the sdi cookbook. *GSDI/Nebert*.

Kralidis, A. T., 2009. Geospatial web services: The evolution of geospatial data infrastructure. In: *The Geospatial Web*, Springer, pp. 223–228.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S. and McClosky, D., 2014. The stanford corenlp natural language processing toolkit. In: *ACL (System Demonstrations)*, pp. 55–60.

Masó, J., Pons, X. and Zabala, A., 2012. Tuning the second-generation sdi: theoretical aspects and real use cases. *International Journal of Geographical Information Science* 26(6), pp. 983–1014.

Paolo Corti, e. a., 2017. Hhypermap. `https://github.com/cga-harvard/HHypermap`.

Rajabifard, A., Kalantari, M. and Binns, A., 2009. Sdi and metadata entry and updating tools. *SDI convergence*.

Smiley, D. and Pugh, E., 2009. *Solr 1.4 enterprise search server*. Packt Publishing Ltd.

Revised July 2017