Journal of Physics: Conference Series **396** (2012) 062017 doi:10.1088/1742-6596/396/6/062017

Improving PHENIX search with Solr, Nutch and Drupal.

Dave Morrison, Irina Sourikova

Brookhaven National Laboratory, Upton, NY 11973, USA

E-mail: dave@bnl.gov, irina@bnl.gov

Abstract. During its 20 years of R&D, construction and operation the PHENIX experiment at the Relativistic Heavy Ion Collider (RHIC) has accumulated large amounts of proprietary collaboration data that is hosted on many servers around the world and is not open for commercial search engines for indexing and searching. The legacy search infrastructure did not scale well with the fast growing PHENIX document base and produced results inadequate in both precision and recall. After considering the possible alternatives that would provide an aggregated, fast, full text search of a variety of data sources and file formats we decided to use Nutch [1] as a web crawler and Solr [2] as a search engine. To present XML-based Solr search results in a user-friendly format we use Drupal [3] as a web interface to Solr. We describe the experience of building a federated search for a heterogeneous collection of 10 million PHENIX documents with Nutch, Solr and Drupal.

1. The PHENIX document collection

The PHENIX experiment at RHIC is a large collaboration of 70 institutions from 14 countries that has been working on building and operating the PHENIX detector over the last two decades. During that time PHENIX has accumulated more than a Terabyte of proprietary documents that are hosted on many web servers around the world. Only a minority of these documents have a well defined format and are stored in the document databases, the rest is a heterogeneous collection that is spread over many servers at the participating institutions. For many years PHENIX used ht://Dig [4] to provide a collaboration-wide web search, but the large volume of documents (more than 10 million currently) started to overwhelm the old search tool that produced increasingly unsatisfactory results and needed a replacement.

2. Search options

To provide fast and relevant search results many big web sites implement custom search and hundreds of search solutions are now available [5]. It is not an easy task to choose the right one. The Google Search Appliance [6] is a popular choice, but it would be costly to support several PHENIX web servers and there is no built-in support for using internal knowledge about the data. Open source tools offer more flexibility (at the expense of requiring a programmer to install and configure them) and we were interested to find search tools that would be able to leverage the fact that our users know how the data is structured fairly well. Most PHENIX users can limit the search to a specific sub domain right away, for example searching only in the mail archives or published papers. After considering several possibilities, we decided to use Apache

Nutch and Solr which provide a good mix of functionality and flexibility and support sub domain searching.

3. Web crawling with Apache Nutch

Nutch [1] is a multi-protocol, multi-threaded, distributed crawler and full-text indexer that works well for collections of 1 - 200 million documents. It uses Tika [16] as a parser and Solr as a search front-end. Nutch configuration is time-consuming but renders flexibility in return. We customize url filtering, crawl scheduling and use the subcollections plug-in to group urls into logical collections. To aggregate collaboration data from many participating institutions we initialize the crawling with the url list of PHENIX collaboration servers and then control the crawling frontier by filtering the external links with the regex-urlfilter plug-in. To skip binary, image and other unwanted file formats we extend the Nutch list of suffixes to ignore in the suffix-urlfilter plug-in.

We use the subcollections plug-in to group urls into logical collections. This plug-in is activated in the nutch-site.xml and is defined in the subcollections.xml. The snippet of subcollections.xml looks like this:

```
<subcollection>
  <name>Published</name>
  <id>publish</id>
  <whitelist>
    www.phenix.bnl.gov/WWW/publish/
    www.phenix.bnl.gov/WWW/talk/pub_papers.php
  </whitelist>
</subcollection>
```

The introduction of subcollections noticeably improved the PHENIX search precision.

The ability of Nutch to parse large variety of file sources (file system, databases, Wikis) and formats (text, pdf, ppt, etc) substantially increased the search domain while the Nutch plug-in architecture allows to easily incorporate new formats when the parsers for them become available.

4. Searching with Apache Solr

Solr [2] is a standalone full-text search server that accepts the input via XML, JSON or binary over HTTP, query via HTTP GET and produces XML, JSON or binary results. It supports rich document handling, faceted search out of the box, hit highlighting and query spelling suggestions. It also provides many ways to customize search result rankings, sorting and displaying. Solr has an extendable data schema that supports adding new fields to the documents for searching, sorting and faceting. We use this feature to add PHENIX subcollections created in Nutch to the Solr data schema. Solr comes with the built-in administrative web interface that provides a comprehensive search statistics.

5. Web interface with Drupal

Since Solr results in XML, JSON or binary formats are not very user-friendly, we added Drupal [3] to serve as the web interface and to display search results in a familiar (Google-like) fashion. Drupal is an open source Content Management System (CMS) that supports functionality via thousands of add-on modules. It has a vibrant community of users and developers and is used by such sites as The White House [7], The Economist [8] and Examiner.com [9]. Drupal provides Apache Solr Search Integration and Nutch modules for integration with Solr and Nutch. To protect private web domains we integrate Drupal with Shibboleth [15].

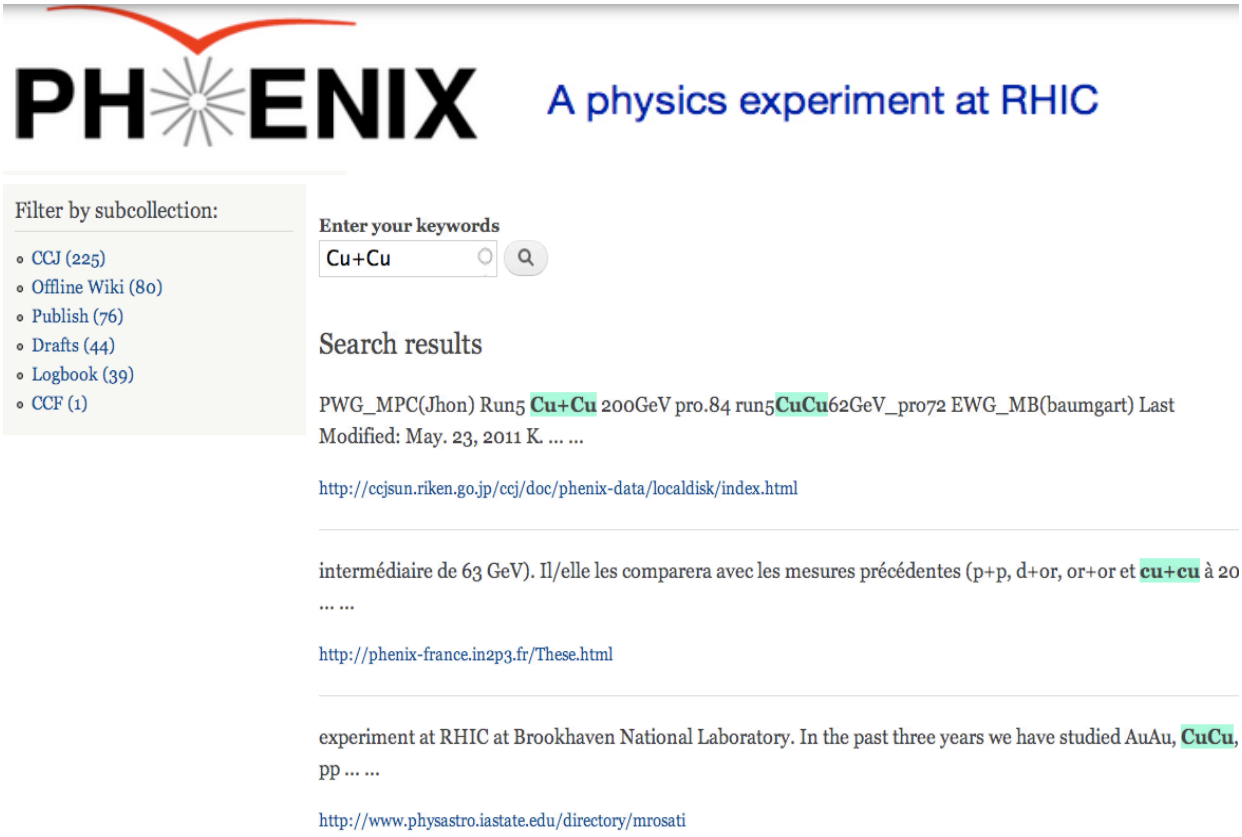

Figure 1. Screenshot of PHENIX search showing the first three results of searching for 'Cu+Cu'. The links in the upper left-hand corner show logical document collections with the count of search hits.

6. Putting it all together

Out search configuration includes Nutch 1.4.dev, Solr 3.4.0 and Drupal 7.12, all running on a single server. Solr, Nutch and Drupal work together by sharing an extendable data schema defined in schema.xml and present in all three configuration areas. Solr and Nutch resolve schema name differences via solrindex-mapping.xml in the Nutch configuration area.

Nutch and Solr integration in Drupal is a rather recent module and out of the box configuration of the versions that we used did not work. Nutch 1.4 schema was lacking Drupal version 7 modifications and this issue was resolved by writing a custom Drupal module and getting help from developers [10].

After we configured Nutch, Solr and Drupal to work together, Nutch crawled about half a million documents in two weeks before slowing down dramatically. The problem stemmed from the old private directories with broken links and restrictive permissions that caused http errors. Nutch effectively stopped on urls with certain types of errors by going to an infinite loop of

retries [11]. A few problem urls could have been filtered out by regex-url filter but filtering too many of them slows down the crawler significantly. We found that starting with seed urls from the middle of the directory tree and not from the top mitigates the above mentioned problem. Tweaking several other configuration options made the crawl steady and Solr was getting about half a million documents per week.

While we are still waiting for the initial crawl to finish, we opened the new search for the collaboration to try out and comment and finalize the design of the most useful document collections. The user experience was extremely positive. Although the initial crawl of all PHENIX domains takes several weeks on a single-node Nutch setup, each subsequent recrawl will become shorter since most PHENIX documents remain static and Nutch automatically discovers unchanged pages via AdaptiveFetchSchedule crawling. Also since Nutch updates Solr index incrementally, all documents that were indexed during the first crawl will remain searchable during the recrawl. For more dynamic web collections Nutch can be run on a Hadoop cluster [14].

7. Summary

Nutch, Solr and Drupal provide a complete set of open source tools to build a federated search for a heterogeneous collection of 10 million PHENIX documents. High-precision search is achieved by defining logical document collections for the most frequently searched web areas. Solr aggregation of multiple index sources allowed us to make further improvements by writing a custom parser for the email archives and sending the index to Solr.

PHENIX is not alone in trying to provide a unified search to the collaboration. Some of the other methods employed by HENP experiments include uploading some of the documents into the database [13] or the Content Management System [12]. The main advantages of the PHENIX approach are:

- A comprehensive search is achieved by including many document sources and formats
- No code writing and not relying on users to upload the documents
- Adding new sources is as easy as adding a new line to the source url file
- Solr and Nutch provide excellent search performance and scalability
- Common search interface for public and private web domains

References

[1] http://nutch.apache.org/
[2] http://lucene.apache.org/solr/
[3] http://drupal.org/
[4] http://www.htdig.org/
[5] http://www.searchtools.com/
[6] http://www.google.com/enterprise/search/
[7] http://www.whitehouse.gov/
[8] http://www.economist.com/
[9] http://www.examiner.com/
[10] http://drupal.org/node/708886comment-5083502
[11] https://issues.apache.org/jira/browse/NUTCH-1245
[12] http://indico.cern.ch/contributionDisplay.py?contribId=114&sessionId=7&confId=149557
[13] http://indico.cern.ch/contributionDisplay.py?contribId=531&sessionId=7&confId=149557
[14] http://hadoop.apache.org/
[15] http://shibboleth.net/
[16] http://tika.apache.org/