

Using the Google Search Appliance for Federated Searching: A Case Study

Abstract

This article discusses an experiment by the University of Nevada, Reno to do federated searching of digital collections and vendor databases using version 4.1 of the Google Search Appliance (GSA). The digital content included in this test came from the university's locally held CONTENTdm and geospatial data collections and a sample of records from EBSCO's Academic Search Premier database. The latter set of records revealed many of the limitations that the GSA has in being able to successfully index and retrieve content that is dynamically generated and that requires authentication by a third party. Finally, the article briefly touches on conversations that UNR has had with the members of the New England Law Libraries Association (NELLCO), which is doing a test of version 4.2 of the GSA on one of its member institutions digital collections.

Keywords

Academic Search Premier, dynamically generated content, EBSCO Academic Search Premier, Google Search Appliance, XML.

Acknowledgements

The author wishes to thank the following individuals for their assistance with the project discussed in this article:

Steven D. Zink, University of Nevada Reno
Araby Greene, University of Nevada Reno
Gregg Steffanelli, University of Nevada Reno
Rick Anderson, University of Nevada Reno
Tracy L. Thompson, NELLCO
Roberta Woods, Franklin Pierce Law Center Library
Tamara Scott, Google Enterprise Search
John Gregory, Google Enterprise Search
Michael Gorrell, EBSCO Publishing

Author

Mary Taylor (taylormk@unr.edu) is the Metadata Services Coordinator for the University of Nevada, Reno.

Introduction

As the Metadata Services Coordinator for the University of Nevada, Reno, two of my areas of interest are metadata harvesting and federated search. During my first months in this position I served on a committee to review and evaluate federated search products. After numerous meetings with vendors and product trials, the committee came away with the conclusion that the functionality offered by these products did not merit the price quotes we were receiving. When I first started in this position and inquired about the five most important tasks that I should be working on, the Dean of University Libraries, Dr. Steven D. Zink, mentioned that finding out the costs and technical requirements for implementing the Google Search Appliance (GSA) should be at the top of the list. The GSA is

Using the Google Search Appliance for Federated Searching: A Case Study

an off the shelf combination of a 1U standard rack-mountable server hardware (frequently known as “The Google Box”)ⁱ and an administrative software module that can be configured to perform searches on a defined network (such as an intranet or website) of over 220 different types of file formats.

Dr. Zink, who is also the university’s Vice President of Information Technology, had made inquiries with Google several years earlier about the cost of implementing the GSA as a federated search product. At that time, the cost of GSA was too expensive to merit an in-depth investigation. As part of my participation in the federated search committee, I decided to make a new inquiry for the sake of due diligence. I also wanted to learn more about how libraries could use it for searching their digital collections and also what the price was for the non-profit and education sector. The Dean supported this idea, especially because Google offers potential customers a 60-day trial period to install and test the GSA. Our attitude was that even if the cost of the GSA was still too expensive, Google’s policy of offering a free trial made it worthwhile to at least bring in the GSA for a test. The model at that time, version 4.1 and we were most interested in learning the requirements for using it to do federated searching of our digital collections and resources.

Crosswalking the Google Search Appliance With Academic Search Premier

During the summer and fall of 2004, we held several conference calls with sales and systems engineering staff from Google’s Enterprise Search group. The main question that we posed to them was how the Search Appliance would work with the types of content that libraries manage. We gave them an overview of our digital collections, including the online library catalog (Innovative Millennium), GIS and map data, CONTENTdm collections, electronic journals, and third party vendor databases. Ideally, we wanted to figure out how to use the Search Appliance as a single point of access, especially to vendor databases and electronic journals. An initial test by Google’s demonstration servers that emulate the GSA was able to successfully index and retrieve from our geospatial/map and CONTENTdm collections.

Thanks to our conference calls with Google, we had gained a basic understanding of how the GSA works. It follows the document model by crawling and caching content based on an item’s URL or filename. The cache index uses this information as an item’s unique identifier in order to link it to subsequent search results:

“The Google Search Appliance crawls your content and creates a master index of documents that’s ready for instant retrieval using Google’s search technology whenever a customer or employee types in a search queryⁱⁱ.”

During the first conference call, we gave an overview to our colleagues at Google about vendor databases like Academic Search Premier and how articles

Using the Google Search Appliance for Federated Searching: A Case Study

in these databases are dynamically generated and have session generated URLs. As an example, the Persistent URL (PURL) for an article about granting “most favored nation” status to China, contains information about the port that authenticates UNR access to EBSCOhost (innopac.library.unr.edu:80) and also the identifiers for the article’s storage database (db=f5h) and unique number (an=9609131521):

<http://0-search.epnet.com.innopac.library.unr.edu:80/login.aspx?direct=true&db=f5h&an=9609131521>

A session generated URL for the same article is much longer and in addition to information about the article’s storage database (db=f5h), it also includes the session identifier (sessionmgr5) and search query (2DChina++%22most++favored++nation:

http://0-web18.epnet.com.innopac.library.unr.edu/citation.asp?tb=1&_ug=sid+D7C7A13A%2D8CB9%2D443D%2D8E41%2DC47B312BF7BA%40sessionmgr5+db+f5h+cp+1+50E5&_us=frn+1+hd+True+hs+True+cst+0%3B1+or+Date+ss+SO+sm+KS+sl+0+dstb+KS+mh+1+ri+KAAACB1B00094089+DF6E&_uso=tg%5B0+%2D+db%5B0+%2Df5h+hd+False+clv%5B1+%2DY+clv%5B0+%2DY+op%5B0+%2D+cli%5B1+%2DRV+cli%5B0+%2DFT+st%5B0+%2DChina++%22most++favored++nation%22+mdb%5B0+%2Dimh+0817&cf=1&fn=1&rn=1

Because we wanted to use the GSA as a point of entrance for vendor databases and electronic journal collections, we needed to find a specific database to crosswalk with the GSA. We were specifically interested in EBSCO’s Academic Search Premier database because of its wide subject coverage and appeal to undergraduate and novice library users. Having the GSA as its point of access would ideally make it an easy and attractive research tool for these users and hopefully increase its overall usage. After speaking with the staff at Google’s Enterprise Search group, we then contacted EBSCO’s Chief Information Officer, Michael Gorrell for two conference calls. The first call was to give him an overview about the Google Search Appliance and to see if EBSCO would be willing to let us use the Academic Search Premier database for our test. For the second conference call, we brought in the Systems Engineer for Google’s Enterprise Search Group, John Gregory, to discuss with Gorrell how their products would work together. EBSCO agreed to participate in a small test, in which a demonstration servers from Google’s Enterprise Search group would index and caches content from Academic Search Premiere. EBSCO did not want to have the demonstration servers directly access and crawl Academic Search Premier. The primary reason for this request was to strain its network, but also technical limitations in version 4.1 of the GSA.

The bulk of digital collections in libraries are vendor databases contain

Using the Google Search Appliance for Federated Searching: A Case Study

session generated URLs, which makes the question of how the GSA would cache this content hard to answer. If an article's URL changes from session to session, whatever the GSA stores in the cache would direct subsequent search results to an Error 404 page because that URL no longer exists. It was not clear to us how the GSA could link the session generated URLs stored in the cache back to the originating articles. While explaining this potential barrier to Google, we asked if they had customers who were used the GSA to search dynamically generated content. We were told that some of their customers were using the GSA to search Customer Relationship Management systems that have dynamically generated content. However we were not given details about how these customers had specifically resolved the GSA's issues with dynamically generated content. Sometime after the test, we found a review of the latest release of the GSA that states that mentions one of its drawbacks as being an inability to integrate with content or document management systemsⁱⁱⁱ

Gregory suggested three possible solutions for making it possible for the GSA to work with dynamically generated URLs. The library's authentication process is a proxy rewrite, so we could investigate how to rewrite the session-generated URL back to its persistent URL (PURL) before caching. Another solution would be to see if the proxy rewrite could strip out the session generated portions of a URL, such as the search query, and then pass the remaining information to the GSA as the URL. This solution was problematic in part because the session generated information included in the URL is located at different sections, not the beginning or end of the URL string. More important was that the Systems Office had already time and effort into implementing a proxy rewrite for the library's vendor database and electronic journal collections. Trying to pursue either of these options was not a realistic solution. Gregory also suggested that EBSCOhost generate a list of all of Academic Search Premier's persistent URLs (about 18 million) for the demo servers to crawl. Again, this suggestion was not realistic given the strain it would put both on EBSCOhost's network and staff.

In the end, Gorrell offered an alternative that proved to be the approach that we took for the test. EBSCO has a database interface that employs the Simple Object Access Protocol (SOAP), an XML-based format for exchanging information in a decentralized environment. Customers can access the SOAP interface through EBSCOhost or can locally host records in this format. EBSCO provided a sample set of XML records from Academic Search Premier that were articles about the 1989 Tiananmen Square Protests or granting "Most Favored Nation" status to China. The Systems Office loaded these records on to a server and then generated a URL for Google's demo servers to crawl and index. It became clear right away that GSA could not return meaningful search results for these files because it could not differentiate between the file's markup tags and article content. It treated both equally as text. There needed to be a way to transform the records into HTML in order for the demo server to be able to crawl and index only the text from the article. We asked the library's webmaster, Araby Greene, for her opinion about the best way to transform these records. In order

Using the Google Search Appliance for Federated Searching: A Case Study

to come up with an answer she experimented with two different approaches to make the files “interpretable” to the demo servers.

She first created an index page and corresponding XSLT style sheet to generate a HTML file from the persistent URL embedded in the XML file. Gregory reviewed these files and replied that even with the style sheet to transform the persistent URLs, unlike a web browser, the demo server cannot interpret tags from within a file. Given this feedback, Greene returned to the question of how to transform the XML files to HTML in manner that would work with the GSA. Her solution was to write and run a script that generated a HTML file for each of the XML files. This transformation was successful, except for an error message for two files, which she had to manually transform. Subsequent attempts to use this script for transforming XML files have been successful with none of the files requiring manual intervention. She also determined that if the GSA could crawl the files starting from the home page, it would be more efficient to change the folder holding the sample records into a subweb on the library’s network. This approach was successful and keyword searches using the terms “Most favored nation,” “China,” and “Tiananmen Square “ generated meaningful search results.

Challenges In Implementing the GSA Version 4.1

Although the outcome of the test was successful, the final decision was to hold off bringing in the GSA for an onsite trial until the next release. We also decided to look into a short-term solution of getting institutional access n Google Scholar, especially because it is free. The test we had done with Academic Search Premier brought up several technical issues and financial considerations. It was still not clear to us if the GSA could pass through a third party site’s authentication process and be able to cache dynamically generated content. Hosting the database records on our network meant investing time and effort to create scripts and style sheets that could transform the XML files into HTML. Greene estimated that she spent at least ten hours working on this portion of the project. It also remained unclear exactly how the Search Appliance would fit into the authentication process required to restrict access to members of the UNR community. Not having clear answers to these questions left us wondering if it could work as a federated search tool for our collections.

In reviewing the case studies^{iv} on the Enterprise Search section of Google’s website, we noted that many of the clients listed were use the GSA for publicly accessible websites or intranets. Given that the GSA was developed and marketed for these types of digital environments, one theory that we developed was that it might only be able to search either completely inside or outside of a firewall. Another factor to consider is that the content in these environments generally consists of discrete documents and stable URLs, making it easier for the GSA to create a stable unique key to store in it cache. One evaluation we found for version 4.1 backed up this conclusion.

Using the Google Search Appliance for Federated Searching: A Case Study

“[The GSA is] best used in tactical external or internal intranet installations where content need not be indexed directly from dynamic repositories^v.”

We also found a brief article about Oxford University’s trial of the GSA that also included issues about how to index both public and restricted sections of its network^{vi}. Their conclusion was that the most straightforward (though expensive) solution would be to implement two models, a GSA for publicly available content and another for restricted content:

“... one for outside the firewall and one for inside. The most likely involved routing all searches through a proxy server maintained separately, which would check all accesses to see if they would work from outside the firewall, and annotating the database according. It is worth noting that if all Oxford Web sites had put their restricted material on a separate Web server (e.g. oucs-oxford.ox.ac.uk) or used a naming convention (e.g. oucs.ox.ac.uk/oxonly/), it would be easy to configure the box to provide the external search as needed^{vii}.”

While we knew that the GSA would be able to cache and index CONTENTdm and GIS collections, doing a full test of it on Academic Search Premier would have required switching to the SOAP based interface and taking an additional charge to our existing contract with EBSCO. It made no sense to make a significant financial investment in a database interface that we would potentially have no other use for besides the test. Even more important was that our original reason for looking into a test of the GSA was Google’s policy of offering a free trial period. Having to pay an additional charge for the SOAP interface to EBSCOhost meant that the test would no longer be free. The other option of hosting a local version of Academic Search Premier on our network would also add an additional charge to our contract with EBSCO and also require additional support from the Systems Office.

The final and most problematic financial issue for implementing the GSA is that it not only uses the document model for content indexing and caching, but also in determining the price. At present there are four different versions of the GSA. The recently released “mini” GSA costs around \$3000, can search up to 100,000 documents and is marketed to “small and medium-sized businesses^{viii}”. The model that would most likely fit within the budget and collections requirements of a library, the GB-1001, costs around \$30,000 and can search up to 500,000 documents^{ix}. Given the massive amount of data stored in an average vendor database, such as the 18 million unique items in Academic Search Premier, using a discrete document or URL as the basic pricing unit would increase the GSA’s cost far above \$28,000. Even the highest end model of the GSA (the GB-8008 which costs around \$450,000) would not be able to cache and index the full scope of Academic Search Premiere.

Follow up with NELLCO and Search Appliance Release 4.2

Using the Google Search Appliance for Federated Searching: A Case Study

During our conference calls with Google, we had been told that the next release of the Search Appliance, scheduled for April 2005, would have some added functionality that could resolve the issues with dynamically generated content and authentication. Several months after our decision to hold off on bringing in the GSA for a trial, one of our colleagues noticed a summary of presentations at the International Coalition of Library Consortia's April 2005 meeting in Boston. The New England Law Library Consortium (NELLCO)'s presentation about federated searching and the GSA at Their presentation, "NELLCO's Blue Sky Thinking - a possible alternative to federated searching^x," discussed many of the same issues that we had faced. Like us they had started looking at the GSA and talking to Google after an unsatisfactory review of federated search product. They were also planning on coordinating a test of the GSA on library and vendor content as a proof-of-concept.

I contacted NELLCO's executive director, Tracy L. Thompson, who delivered the presentation, which led to several telephone and e-mail conversations about the GSA with her and Roberta Woods from the test site, the Franklin Pierce Law Library. I shared with them our notes from the conference calls and test and they in turn discussed the research that they had been doing for an upcoming meeting with Google and an interested vendor about testing the GSA at one of its member libraries. During our conversation, Thompson made the very good point that while Google Scholar might provide a good short term solution for applying Google's PageRank technology to searching scholarly content, its disadvantage is that the collection development is determined by which publishing companies and institutions happen to be contributing content, instead of local needs and policies. At present, Google Print (the division that oversees the Google Scholar and the "Google for Libraries" initiatives) has declined to name all of the participating publishers. Furthermore, organizations such as NELLCO's member institutions have found that niche publications, especially for professions such as theirs, are not yet being making it into Google Scholar'.

XML Feeds and Authentication API in Release 4.2

Speaking with Thompson and Woods was extremely helpful in further refining our technical knowledge about the GSA. Woods located two key pieces of information that we had requested during our conference calls - how to cache and authenticate 3rd party content. This information is not found on the Search Appliance section of Google's website, but rather the its Code section of Google's website^{xi}. According to the documentation, Release 4.2 of the Search Appliance has a "3rd Party Content Feed API" that makes it possible to handle dynamically generated content by converting search results into XML. We were able to find a case study on the Search Appliance section of Google's website, about Sur La Table's e-commerce website, discusses the process of converting the search results for dynamically generated content from a 3rd party database into an XML feed:

Using the Google Search Appliance for Federated Searching: A Case Study

“Because the Google Search Appliance gives administrators access to Google results as an XML data feed, Grant was able to integrate the results easily into a Cold Fusion environment. Pointing the Google Search Appliance at an offline product database, Grant then mapped the search results to live URLs using a few simple scripts^{xii}.”

Woods was also able to locate the documentation about the Authentication Application Protocol Interface (API), for release 4.2, the “Secure Content API.” although after the answer given to us during the chat session, it still is not clear how authentication scales to the level of working with electronic journals and databases, where access is for a large and distributed user base and requires more than a single user id and password. The latest developer’s guide to the Application Feeds Protocol, released on June 2nd 2005, does describe a process similar to the one we used for our test:

“To create a feed, you will convert your data to XML. You will then upload the XML to the appliance using a web form or a script. A script that creates the XML data and pushes it to the appliance is known as a custom connector. The XML data that you push to the appliance is the feed^{xiii}.”

Conclusions

Despite the “plug and play” statements in their promotional materials, doing a full implementation of the Google Search Appliance in the library environment not only requires the cost of purchasing and annual licensing fees, but also additional manpower and unforeseen costs like having a SOAP database interface. At present, our institution has decided to defer implementation until it becomes clearer how the GSA can best authenticate and index/retrieve dynamically generated content from 3rd party databases. Our short-term solution has been to work with implementing Google Scholar. This decision does not mean that we have abandoned the idea of using the Search Appliance, but instead that we want to gain a better understanding of the technologies underneath it and to also advocate for development that can address these needs. Our impression is that working with this type of content, especially when it is dynamically generated, discrete documents, is new to Google. It will likely take more work by Google and customer input in order to make the Search Appliance be a “plug and play” solution for 3rd party dynamically generated content. We continue to stay in touch with our contacts at Google and EBSCO and are following the work that NELLCO is doing with their experiment to test the Search Appliance on Franklin Pierce’s collections.

Using the Google Search Appliance for Federated Searching: A Case Study

Notes

ⁱ *Review: Implementing the Google Search Appliance in an Intranet environment* (<http://www.macosex.com/articles/review-implementing-the-google-search-appliance-in-an-intranet-enviro.html>)

ⁱⁱ <http://www.google.co.il/enterprise/gsa/>

ⁱⁱⁱ *ibid*, Implementing the Google Search Appliance

^{iv} <http://www.google.com/enterprise/customers.html>

^v "Google upgrades search appliance", by Ann Bednarz, Network World (6/7/04) p. 29;

^{vi} Rahtze, S. (January 2005). Looking for a Google Box? Ariadne, Retrieved January 10, 2005, from <http://www.ariadne.ac.uk/issue42/rahtz/>

^{vii} Rahtze, S. (January 2005). Looking for a Google Box? Ariadne, Retrieved January 10, 2005, from <http://www.ariadne.ac.uk/issue42/rahtz/>

^{viii} Google Mini

^{ix} http://www.google.com/enterprise/gsa/product_models.html

^x Tracy Thompson Conference Presentation

^{xi} <http://code.google.com>

^{xii} Sur La Table Case Study

^{xiii} http://code.google.com/gsa_apis/feedsguide.html#system