DATA MINING SERVICE FOR OAISTER DIGITAL LIBRARY

Petr Doležel, Tomáš Dulík
Tomas Bata University in Zlin
Faculty of Applied Informatics
Nad Stráněmi 4511
760 05 Zlín
Czech Republic

ABSTRACT

OAIster is a digital library with more than 10 million resources - one of the largest repositories in the world. It supports two search methods: ordinary web-based search and the SRU search-retrieve protocol which returns metadata in BibClass format. Unfortunately with SRU, the OAIster database search is quite time-consuming and only single keyword queries are supported. The goal of this paper is to present new web service for searching in OAIster repository. The service is based on the SQI web service and allows multi keyword search. It is currently implemented in the ObjectSpot search engine but it might be used in any other service which supports SQI and RSS 2.0 data formats. The results and comparison with SRU client are evaluated in this paper.

KEYWORDS

SQI, RSS, SOAP, digital library, OAIster, ObjectSpot

1. INTRODUCTION

Digital libraries might be defined as large, organized collections of information objects. Standard libraries store treasures of information which are accessible through the gateway – mostly digitalized version of catalogue. On the other hand the digital library is the treasure itself. While standard libraries have mainly conservative function the digital libraries lay stress on access. (Witten, Ian H., 2005)

These libraries are very important part of academic life because of scientific papers that are usually stored in these libraries. Search engines are the gateway to users. Usually digital libraries have their own search engine and additionally provide their metadata to standard search portals such is Google or Yahoo. According to the study among students in England was discovered that many of them use standard web search engines such is Google for searching educational resources (Griffiths, J., Brophy, P., 2005). Within the project iCamp new search portal was developed [1]. ObjectSpot comes with the idea to join several search web services into one search engine. It is a web-based federated search client plus middle-ware mediator for distributing queries and collecting results to digital libraries and learning object repositories that implement the Simple Query Interface [2].

Searching in such libraries is commonly provided via web-based interface. In some cases additional solution is also available. These solutions differ in various techniques that are represented by different web services or protocols such is SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration), WSDL (Web Services Description Language), SQI (Simple Query Interface), SRU/SRW (Search and Retrieve via URL/Search and Retrieve Web Service), OpenSearch, CQL (Contextual Query Language) or XQuery (Coyle, F. P., 2002).

This paper is focused on OAIster digital library. It was developed by group "Digital Library Production" on the University of Michigan. The OAIster project was designed to establish a broad, generic information retrieval resource pointing to publicly available digital resource representations, mostly provided by the research library community (Hagedorn, K., 2003). This project is based on utilization of OAI protocol that allows the accessibility of database records. The project started in 2002 and the library contains currently more than 10 million records [3].

The title "OAIster" was chosen according the OAI protocol. This protocol can be easily implemented to various digital libraries thanks many open access tools. It was designed in Santa Fe and the platform was named Open Archives Initiative (OAI). It was "developed as a means to federate access to diverse e-print archives through metadata harvesting and aggregation" (Shreeves, S. L., et al., 2005). The logic of OAI is divided into two parts: data providers or repositories and service providers or harvesters. The first group makes the metadata available via OAI protocol and the second one harvest metadata from them via the same protocol (Lagoze, C. and Van de Sompel, H., 2001). The whole process is based on four features: unique identifier, common metadata standard (Dublin Core base BibClass), communication via HTTP and strict usage of XML.

2. IMPLEMENTATION

In this paper we deal with two protocols/services: SRU and SQI. SRU/SRW is a protocol that uses Internet to carry the message between user and target. This protocol usually transfers the query via HTTP GET method (SRU); however, it might be implemented also via SOAP (SRW). The service is based on XML and the result of searching is an XML document (McCallum, S., 2006). OAIster SRU is available online at the following http address:

http://www.hti.umich.edu/cgi/s/sru/sru?operation=searchRetrieve&query=QUERY&
x-collid=oaister&startRecord=START&maximumRecords=COUNT

Where QUERY is query statement (it is the keyword in this case), START represents the first record in record set and COUNT means the number of returned records. The character set of returned document is ISO-8859-1 and the XML document specification is BibClass.

The SQI service was developed as a universal interoperability layer for educational networks. SQI is similar to SRW; however, there are some differences. SRW does not provide mechanisms for authentication and access control. In addition for reporting errors SQI uses faults, SQI is based on session management concept and uses several methods for synchronous or asynchronous communication while SRW lacks these features. (Simon, B., et al., 2005)

2.1 HTML Grabbing

The solution described in this paper is based on HTML grabbing. The web-based OAIster search engine was used as the data source. The process consists of several steps. First of all, the query has to be transformed according to GET method used for searching on OAIster website. Simple keyword list has to be changed into part of URL:

keyword1 keyword2 keyword3 → &q1=**keyword1**&op2=And&q2=**keyword2**&op3=And&q3=**keyword3**

After that the HTTP request is sent. The HTML file returned is loaded and prepared for data grabbing. Start and end of result part is found according to unique marks represented by HTML comment string. The HTML code of this part is not valid according to W3C standard so the code has to be cleaned and prepared for parsing. Each unnecessary tag (e.g. bold, italic, span, div, etc.) or attribute (e.g. nowrap) is eliminated and the result part of the page is transformed to XML. Now it is prepared for XML parser. Required result format is generated when the document is parsed. In this case the result format is RSS 2.0.

2.2 SQI Target

SQI target is implemented as a SOAP service and is created in PHP with use of Nusoap library. SQI methods implemented in this target with possible parameters can be seen in Table 1.

Table 1. SQI methods implemented in OAIster target

Method	Parameter
setQueryLanguage	VSQI (simply keywords separated by spaces)
setMaxQueryResults	Might vary from 1 to 50
setResultsSetSize	Greater than 0
setMaxDuration	Greater than 0
setResultsFormat	RSS20 (RSS version 2.0 – UTF-8 encoding)
synchronousQeury	Query statement in query language, start result greater than 0
getTotalResultsCount	Query statement in query language

This new SQI target was included into the iCamp project ObjectSpot search engine [1]. ObjectSpot provides more SQI targets from several different digital libraries.

3. RESULTS

3.1 Response Times

The response time of SRU method was compared with SQI target mentioned above. Because of OAIster SRU limitation, only single keyword search was used in the comparison. The fastest times for both methods are shown in Table 2. Response times for multiple keyword searches are provided only for SQI service.

Table 2. Fastest times for both services

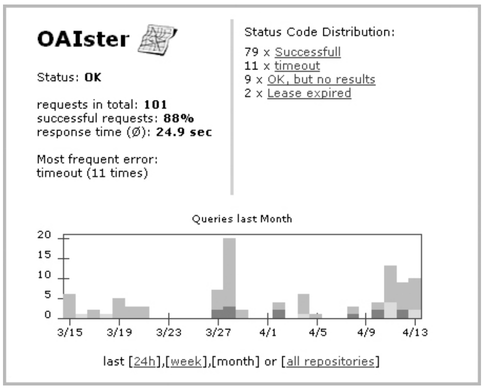
Keyword(s)	Total results	SRU	SQI
“the”	10 132 994	server error	~58 s
“data”	1 603 964	~53 s	~23 s
“education”	309 982	~34 s	~21 s
“astronomy”	41 930	~9 s	~13 s
“pluto”	886	~7 s	~13 s
“quaoar”	16	~5 s	~7 s
“the AND data”	1 226 504	N/A	~35 s
“education AND astronomy”	1513	N/A	~10 s
“education AND astronomy AND pluto”	15	N/A	~7 s

As can be seen in Table 2, the time needed for successful response varies according to total amount of records. SQI target was successful in all tests. The SRU service failed with common keyword “the” that is actually in every English article. More than 10 millions of records were found and with such huge amount of data the SRU method threw server error. In contrast to SRU, the SQI target was successful even though the response time was quite long. The SRU method was faster for results with record set containing approximately thousands of items. It is one of a few advantages of this service.

The multi-keyword search was not available for SRU thereby the time was measured only for SQI target. The speed is still suitable since the multi-keyword solution for SRU method would be incomparable.

3.2 Reliability

This SQI target is used in ObjectSpot for several months. The results shown in Figure 1 represent last month statistic (year 2008). As can be seen, the average response time is about 24 seconds. That means the average search results contain thousands of records and therefore – according to Table 2, utilization of the SRU for low record counts would not bring any improvement into the average response times.



OAIster

Status: **OK**

requests in total: **101**
successful requests: **88%**
response time (Ø): **24.9 sec**

Most frequent error:
timeout (11 times)

Status Code Distribution:
79 x Successfull
11 x timeout
9 x OK, but no results
2 x Lease expired

last [24h],[week],[month] or [all repositories]

Figure 1. Month statistics for OAIster in ObjectSpot

As it is shown in Figure 1 the most frequent error was timeout. This is caused by slow search engine on the OAIster site; however, the share of successful requests is 88%.

4. CONCLUSION

Both SQI and SRU service have their advantages and disadvantages. As can be seen in Table 3, our SQI target offers more advantages:

Table 3. Advantages and disadvantages of the SQI target

Advantages	Disadvantages
Faster than SRU for average and large record sets	Slower than SRU for small record sets
Multiple keyword search	Depends on non standard HTML coding
Character set is UTF-8	
SQI service implementation	
RSS 2.0 output	

HTML coding is not standardized in case of data logic. That means HTML page could be changed without affecting the record set content. But such change might completely change the logic of the HTML page. This is the largest disadvantage; however, it seems that there is a sign of some data logic in case of OAIster. In addition the HTML cleaning process makes the SQI target robust to small changes and if the SQI target is kept updated this disadvantage could be eliminated. The code of the target is written with respect to possible changes in the page logic so the eventual change is very easy and quick. Following advantages might override this disadvantage.

The large advantage is the speed of the HTML grabber that is mostly faster than SRU. The processing time and conversion to RSS format is insignificant considering the loading time. Next advantage is the multiple keyword search. This option is not provided in SRU solution. Data in OAIster library are saved in ISO-8859-1 character set because of indexing and better searching; however, for XML output the UTF-8 encoding is more suitable. New SQI target provides the result in this character set. Also the RSS format is more common than OAIster BibClass format.

Future work on this target might be focused on greater robustness and stability and on implementation of new SQI methods especially for asynchronous communication.

ACKNOWLEDGEMENT

This work was supported by research project iCamp (FP6, IST-027168). This support is gratefully acknowledged.

NOTES

[1] ObjectSpot: iCamp SQI service search box. Developed by R. Koblischke, S. Sobernig, S. Sigurdarson and F. Wild. [Accessed April, 2008: http://www.objectspot.org/]

[2] iCamp web: Inside the ObjectSpot [Accessed April, 2008: http://www.icamp.eu/]

[3] Press release: *OAIster Reaches 10 Million Records*. 2007 [Accessed April, 2008: http://www.oaister.org/docs/press_release.pdf]

REFERENCES

Coyle, F. P., 2002. *XML, Web Services, and the Data Revolution*. Addison Wesley.

Griffiths, J. and Brophy, P., 2005. *Student Searching Behavior and the Web: Use of Academic Resources and Google*. Library Trends, v. 53, no. 4, pp. 539-554.

Hagedorn, K., 2003. OAIster: a "no dead ends" OAI service provider. *Library Hi Tech*, Vol. 21, Part 2, pp. 170-181

Lagoze, C. and Van de Sompel, H., 2001. The Open Archives Initiative: Building a low-barrier interoperability framework. *Proceedings of First ACM/IEEE-CS Joint Conference on Digital Libraries*: New York, USA, pp. 54–62.

McCallum, S., 2006. A look at new information retrieval protocols: SRU, OpenSearch/A9, CQL, and XQuery. *72nd IFLA General Conference and Council*.

Shreeves, S. L., et al., 2005. *Current Developments and Future Trends for the OAI Protocol for Metadata Harvesting*. Library Trends 53, no. 4, pp. 576-589

Simon, B., et al., 2005. A Simple Query Interface for Interoperable Learning Repositories. *Proceedings of the WWW 2005 Workshop on Interoperability of Web-Based Educational Systems*: Chiba, Japan, pp. 11-18.

Witten, Ian H., 2005. *Digital Libraries and Society: New Perspectives on Information Dissemination*. Design and Usability of Digital Libraries: Case Studies in the Asia Pacific, edited by Y.L. Theng, et al. Information Science Publishing, 2005. pp. 191-215.