Hybrid Online-Offline Digital Collections

Hussein Suleman
Department of Computer Science, University of Cape Town
Private Bag X3
Rondesbosch, 7701, South Africa
hussein@cs.uct.ac.za

Marc Bowes
Department of Computer Science, University of Cape Town
Private Bag X3
Rondesbosch, 7701, South Africa
mbowes@cs.uct.ac.za

Matthew Hirst
Department of Computer Science, University of Cape Town
Private Bag X3
Rondesbosch, 7701, South Africa
mhirst@cs.uct.ac.za

Suraj Subrun
Department of Computer Science, University of Cape Town
Private Bag X3
Rondesbosch, 7701, South Africa
ssubrun@cs.uct.ac.za

ABSTRACT

Online digital repositories are rapidly becoming the norm to store various different forms of content, including academic documents and heritage collections. There are many advantages to online systems, including general accessibility and the increasing use of Web browsers as a platform. However, it can argued that, for the preservation of heritage, offline systems may offer other advantages, such as replication, simplicity and cost-effectiveness. This paper describes an attempt to exploit advantages of both approaches in the form of a combined online-offline digital repository, where some content is online on a Web server and other content is offline on a DVD-ROM. This new approach to designing systems for preservation and access is especially useful in areas with limited Internet bandwidth, such as in most African countries. Experimental results confirm that this system is effective and efficient.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*distributed systems*; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*dissemination*

General Terms

Design, Performance

Keywords

Digital repositories, ICT for development, standalone

1. INTRODUCTION

Traditional digital repository systems tend to either be developed for a specific collection, such as the system created to manage the Bleek and Lloyd collection [3], or tend to be heavyweight. They are heavyweight in that they require a slew of software to be installed, usually including a database management system and a Web server. This leads to digital collections traditionally being accessed as online/Web-based applications. In places where Web connectivity is not available, or limited, access is difficult. To enable access without Web connectivity, it is necessary to revert to distributable repositories.

A distributable repository is a collection of information, along with a system to access it, that can be distributed on a removable medium such as a CD or a DVD. The information is therefore static if one uses a read only medium.

Greenstone [5] is the only popular digital library system that supports distribution on CD or DVD. However, Greenstone is based on an older Web paradigm and, once a collection has been created on removable media, updates are not possible. Greenstone has been popular in developing countries for the distribution of content such as electronic books on farming and human rights, as this information can be distributed without the need for any Internet connection. This works well where the information is fixed and no additional digital objects are added to the collection over time.

However, many digital collections (such as the Bleek and Lloyd Collection) are characterised by a large initial digitisation effort followed by a trickle of additions over time. In countries without widespread broadband Internet connectivity, transferring the initial collection over the Internet is impractical but transferring the updates is an option. This is the premise of the CALJAX system presented in this paper: that a hybrid digital library system can provide current updates integrated into a large offline collection, thus giving users in developing countries full access to information using a combination of offline dissemination and limited Internet bandwidth.

The rest of this paper provides an overview of some repository technology and AJAX, then how the CALJAX system was designed. Various experimental results are then discussed to illustrate how CALJAX is effective and efficient. Finally, conclusions and future work are presented.

2. BACKGROUND

As this project is associated with the usually distinct topics of digital repositories and Web technology, each is discussed separately in the following sections.

2.1 Digital repositories

Ideally, an AJAX-based digital collection would be run directly off the storage medium with a standard browser as the only software requirement. To date, the system closest to providing such functionality is the Greenstone Digital Library software. Greenstone currently allows users to browse digital repository content directly from a CD-ROM but requires prior installation of several packages, including a Web server, before this is possible [5]. Greenstone is described as providing a way of organising information and publishing it on the Internet in the form of a fully-searchable, metadata-driven digital library. It ingests metadata (supporting various standards such as Dublin Core, RFC 1807, etc) and various types of digital resources, using different plug-ins for different document formats, to produce its own set of XML data files. A Java applet interface called the Librarian handles the management of the repository. This includes creating collections, defining and editing metadata sets for collections, reviewing and editing metadata and configuring the system (e.g., configuring the plug-ins).

On the other extreme is the Fedora repository toolkit[2], an object repository with a carefully-defined set of APIs that allow applications to interoperate with the repository. It is based on metadata and data stores, using XML files and databases as necessary. Fedora has several prerequisite software package requirements, such as a Java runtime environment and a database system. It does not allow the creation of distributable copies of its collections and its entire operation is virtually real-time and online. It provides only the base framework, without a full workflow management system.

Of all the current repository tools, only Greenstone supports redistributable collections, but this requires prior installation of some software, and, most importantly, does not provide users with current content. This was the motivation for the development of the CALJAX system: to create collections that would work directly off the storage medium used and seamlessly integrate current content into offline collections, with the only requirement being a standard Web browser.

2.2 AJAX

Web applications for a long time have been less rich and less responsive than their desktop counterparts. However, with the introduction of Web 2.0 and technologies such as Asynchronous JavaScript and XML (AJAX) [1], this gap is closing. AJAX represents a shift in what is considered possible on the Web. AJAX incorporates [1]: XHTML and CSS for basic syntax and layout of information; DOM to manipulate documents within the browser; XML and XSLT for informatiomn interchange and transformation; XMLHttpRequest to asynchronously load data from remote sites; and Javascript as the glue holding everything together.

In the classic Web interaction model the communication is one way, with the user performing an action that is then POSTed back to the server, which does some processing and then returns an HTML page to the client. This limits the interactivity and feedback of a website. AJAX allows two way communication to occur between the browser and the Web server without having to install additional software or reload the page each time. It introduces an AJAX engine that acts as an intermediary between the user and the server. Instead of loading a Web page, when a session is started the browser loads an AJAX engine written in JavaScript. In cases where AJAX is used, this AJAX engine is then responsible for rendering the user interface as well as communicating with the server or other sources.

The design of CALJAX is presented as an alternative to the existing repository systems, based on a strong AJAX foundation.

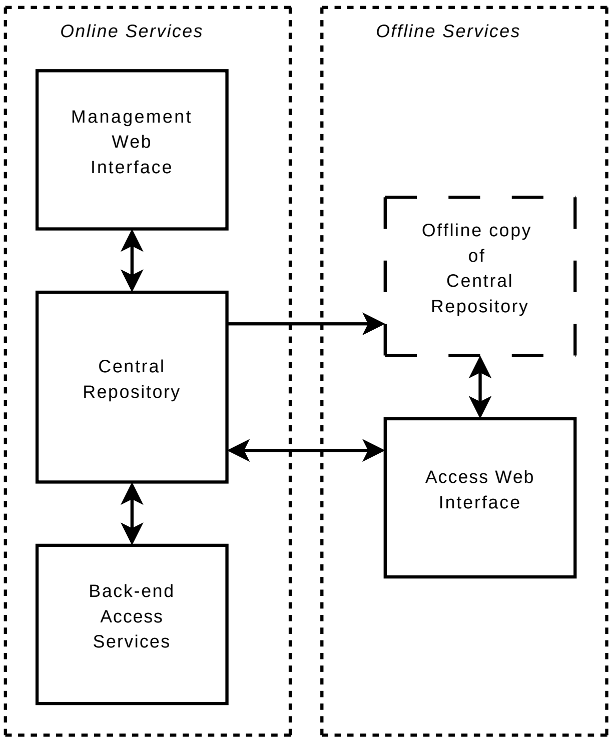

Figure 1: Overview of CALJAX system

3. DESIGN OF CALJAX

CALJAX is made up of independent components to provide search and browse services and to manage the online repository. These are depicted in Figure 1. CALJAX was developed as a prototype to demonstrate the feasibility of using recent advances in Web technology to improve global access to information - the features implemented were therefore the most common ones, to serve as proof-of-concept. It is not intended that this system will replace production systems - but instead that it will inform their future development.

AJAX was therefore chosen as a base technology to build CALJAX, a generic repository management and access system. CALJAX is arguably both lightweight and distributable, with a minimal software footprint, but without sacrificing usability or functionality. The system was designed to support lightweight access mechanisms (browse, search) for the end-user while also including a lightweight management interface for the administrator - the rest of this paper focuses on the search and browse services only.

One often-quoted limitation of Web browsers is that they force AJAX applications to execute within a sandbox that prevents integration of resources from multiple locations - commonly known as cross-site scripting. Due to the potential usefulness of allowing this feature, the W3C has begun developing a security framework within which this will be allowed in newer and future Web browsers [4]. This feature (supported in Firefox 3.5+ already) was exploited in CALJAX to enable integration of content from a local source with content from a remote source, thus allowing the user full access to the most current content.

3.1 Repository

The central repository contains a collection of digital objects and metadata. These are stored as files in hierarchical directories, with each file having an associated metadata file. The advantage of this approach, especially for offline access, is that the files can be accessed directly, without a mediating layer.

When the repository is distributed, the central repository's contents are simply copied to some form of removable media (CDR, DVDR, flash drive, etc.). Thus the offline version of the site has a complete copy of the central repository's contents at the time the copy is made.

3.2 Search/Browse Services

The back-end access services are pre-processors that generate indices for searching and browsing of the repository's contents. These indices are stored alongside the contents of the central repository and copied to removable media as well. In order to enable access via a Web browser, all indices need to be stored as XML files that can be parsed from within a Javascript environment. Pre-processing is necessary to speed up access to result sets, especially since offline use may be slower than server-based systems.

Pre-processors for browsing generate lists of objects in sorted order of various fields, such as title - these are necessary because there is no way for AJAX to list the contents of a directory. For example, the index for "title" will contain a list of all file names sorted in order of title. Pre-processors for searching generate inverted files that can be used with a typical extended boolean search implementation in the access Web interface. For example, the index for the word "water" will contain a list of all names of files that contain the word "water".

For greater efficiency, browse indices also are split into shards, where each shard contains only as many entries as can be displayed on a single screen. For example, shard1 could contain identifiers for entries 1-10, shard2 could contain identifiers for entries 11-20, etc. In order to list 10 identifiers starting at an arbitrary position, a maximum of 2 shards will need to be read from disk. Thus, the speed of browse operations will be constant irrespective of the size of the collection.

The access Web interface is a typical Web-based repository interface offering the user search and browse features. When a user invokes one of these services, a Javascript application retrieves the relevant indices and presents the user with a set of potentially relevant results. A user is able to navigate through multiple pages of results and view digital objects of interest.

3.3 Online-offline Integration

The Javascript application also makes a request to the central repository for updates to the collection since the time the copy was created. For simplicity, the central repository sends back metadata for all new and updated entries, assuming that updates are not frequent. The central repository creates this list by comparing timestamps on all items with the timestamp of creation sent by the Javascript application. This timestamp of creation is added to each DVD that is created specifically to support updates. It is quite possible that different versions of the collection will exist as snapshots burnt to DVD at different times - but the online-offline integration will ensure that all users will get access to the same complete collection of digital objects!

The updates from the server are then interspersed into the result set at the client end, giving the user the illusion of an up-to-date collection on removable media. For a search operation, the client application adds the updated items when accessing the inverted file indices. For a browse operation, the client application inserts items into their correct positions in the shards before displaying results to the user.

If there is no Internet connection, the online integration feature will silently fail. When the users choose to view digital objects, the local files will be presented whenever possible and the online versions otherwise.

4. EXPERIMENTAL RESULTS

Experiments were conducted on both the standalone/offline performance and the integration of online and offline collections. These are described in the following sections.

4.1 Offline Performance

Performance tests were conducted by indexing and browsing data collections of various sizes and recording the time taken to do so using the Linux time command and instrumentation of the software.

The data collections used were simple text files filled with test data. This allowed for strict control over the number of files, as the exact number of files could be generated for each test.

The pre-processor performance test was conducted by running the pre-processor on collections containing various numbers of files (10, 50, 100, 500, 1000, 5000, 10000) and recording the time it takes to pre-process each batch.

The browsing test was conducted by loading 3 pages from the first 3 browsable fields and averaging the time taken to load the pages. The reason more than one browsable field was used is because the time taken to open a page can be influenced by the number of items in the quick navigation toolbar and this changes between the browsable fields.

DVD testing was then performed using the Bleek and Lloyd collection (35 950 files). The collection was pre-processed, written to and run off a DVD, where it was browsed in the same manner as the local testing, with the load times recorded and averaged. This test was only conducted with one collection size as it is shown that collection size has little effect on browsing time.

All performance testing was done on a Mac Book Pro 5.2. The test was done with 10 items per page. The online updates were disabled for this test.

Figure 2 shows the time taken to generate a browse view listing a subset of digital objects off a hard drive. This time is linear but does not increase much with increases in collection size. The sharp drop in time when the number of files is less than 11 occurs because no paging of results is necessary. The non-constant time is due to other operations, such as the generation of quick navigation links in

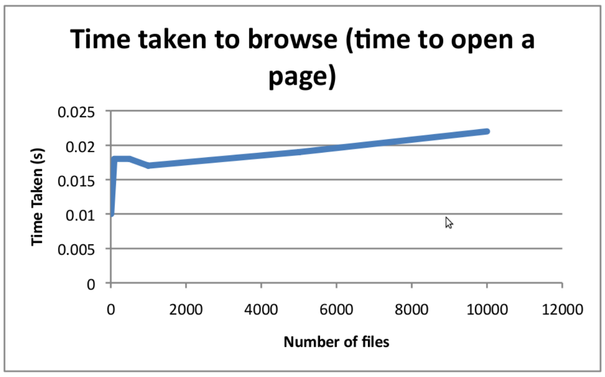

Figure 2: Time taken to browse a collection

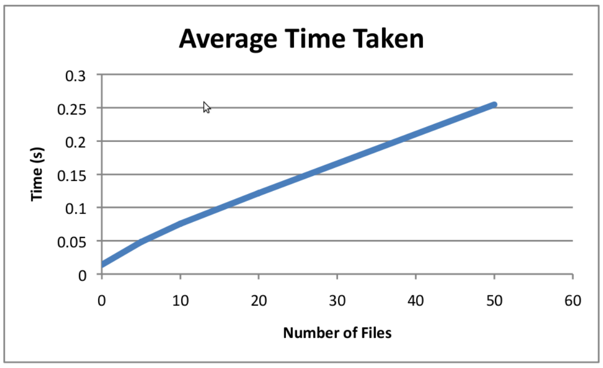

Figure 3: Time taken to integrate online files into a result set

the interface. Without these links, the time to generate a browse view is appoximately constant, which is to be expected with a reasonable indexing implementation.

When the collection is browsed off a DVD-ROM, the times taken to browse a single page vary from 0.296 seconds on Chrome to 0.999 seconds on Internet Explorer. This is still acceptable for most applications.

Similar results were obtained for the search function.

4.2 Online-Offline Integration Performance

In order to determine the efficiency of the online updates, the load times of browsing with varying numbers of online updates was measured. A local stub file was used for the updates. This does not play a role in the results as the only difference between the local stub file and an update from an external machine is the loading time of the website (the update is downloaded when the site is first opened and saved to memory). It thus has no effect on the speed of the browsing operations. The load time was calculated by browsing through 6 pages in the first sortable category and averaging the time taken to load the pages. As only the difference in browse time due to the number of files in an update was being investigated, the category being browsed was kept constant.

Figure 3 shows the time taken to integrate updates into a result set before displaying it to the user.

Due to the time taken to browse a page increasing linearly with the number of files in an update, for large updates the responsiveness and performance of the Web page would be negatively affected and would thus have a negative impact on the usability of the system. If the updates are small, the online update feature will not have a great impact on the browsing experience. However, if the updates become too large, other approaches to minimise the integration of data will need to be considered.

5. CONCLUSIONS AND FUTURE WORK

CALJAX was developed as a proof-of-concept system to illustrate that a digital repository can include the best of both online and offline worlds - using offline collections for speed and online collections for currency.

Experiments have confirmed that the design is scalable and efficient for preprocessing and online operations. Also, the online-offline integration works reasonably well for small collections but suffers from linear scalability. A future version of the system could address this by returning only select results and not all updates on request.

Ultimately, this prototype has demonstrated that it is possible to leverage the latest Web technology to support simpler dissemination of information in developing countries with poor Internet connectivity. As a consequence of this form of dissemination, the data must be stored in a much simpler format, lending itself to preservation as well. While the services tested were limited to search and browse, this hybrid approach is applicable to annotations, recommendations and a host of similar services where the primary digital objects are substantially larger than the auxiliary information - thus, poor Internet connectivity need not be a reason to limit the quality of services to users in developing countries!

6. ACKNOWLEDGEMENTS

This project is made possible by funding from University of Cape Town, NRF and the Telkom-NSN-Telesciences Centre of Excellence at UCT.

7. REFERENCES

[1] J. J. Garrett. Ajax: A new approach to Web applications. Technical report, Adaptive Path, 2005. Available: http://www.adaptivepath.com /ideas/essays/archives/000385.php.

[2] T. Staples, R. Wayland, and S. Payette. The fedora project: An open-source digital object repository management system. *D-Lib Magazine*, 9(4), April 2003. Available: http://www.dlib.org/dlib/april03/staples/04staples.html.

[3] H. Suleman. in-browser digital library services. In L. Kovacs, N. Fuhr, and C. Meghini, editors, *Proceedings of Research and Advanced Technology for Digital Libraries, 11th European Conference (ECDL 2007)*, pages 462–465. ECDL, September 2007. Available: http://pubs.cs.uct.ac.za /archive/00000434/01/ ecdl_2007_ajax.pdf.

[4] A. van Kestern. Access control for cross-site requests. W3c working draft, World Wide Web Consortium, September 2008. Available: http://www.w3.org/TR /2008/WD-access-control-20080912/.

[5] I. Witten, S.-J. Cunningham, B. Rogers, R. McNab, and S. Boddie. Distributing digital libraries on the web, cd-roms, and intranets: Same information, same look-and-feel, different media. In J. Yen and C. C. Yang, editors, *Proceedings of First Asia Digital Library Workshop: East Meets West*, pages 98–105, 1998. Available: http://nzdl.sadl.uleth.ca/gsdl/collect/publicat/ index/assoc/HASH0199/95cc7f7f.dir/doc.pdf.