Davis: A Generic Interface for iRODS and SRB

Shunde Zhang [#1], Paul Coddington [#2], Andrew Wendelborn [*3]

[#] *ARCS, eResearch SA, and the School of Computer Science, University of Adelaide*
Adelaide, SA, Australia 5005

[1] `shunde.zhang@arcs.org.au`

[2] `paul.coddington@adelaide.edu.au`

[*] *The School of Computer Science, University of Adelaide*
Adelaide, SA, Australia 5005

[3] `andrew@cs.adelaide.edu.au`

***Abstract*—The Storage Resource Broker (SRB), and its successor, the Integrated Rule-Oriented Data System (iRODS), are typical data grid software systems, which are widely exploited by researchers to store large data collections with associated metadata. On the other hand, the proprietary iRODS/SRB protocol results in limited support of high-quality and generic user interfaces. Although a few Graphical User Interfaces (GUIs) and web interfaces have been developed for SRB and iRODS, they either have restricted functionality, or are customized for the needs of particular user groups. Moreover, they are not easy or straightforward to use, especially for basic users, which sometimes stops potential users from adapting iRODS/SRB as their storage solution. In order to tackle these shortcomings, Davis, a generic gateway to iRODS/SRB, has been developed, aiming to expand the use of iRODS/SRB to users with any level of computer skills, as well as making it easier to interface iRODS/SRB with other applications. Davis conforms to the open standard and broadly accepted WebDAV protocol, with additional features to facilitate HTTP access. This paper first investigates existing iRODS/SRB client tools, and then describes the approach of implementing Davis and its features, followed by some use cases to demonstrate the usefulness of a WebDAV interface to iRODS/SRB.**

I. Introduction

The SRB project, developed by San Diego Supercomputer Center (SDSC), started in 1996 with a goal of reading from different data sources via a single data access interface [1]. After two years, a complete software architecture was finished and released as the first version, and the architecture has not been changed since then. It has two major parts: a uniform data access API and a metadata catalogue [2]. The uniform data access API is used to read data from heterogeneous distributed storage resources, such as file systems, database systems, and archival storage systems. On the other hand, the metadata catalogue uses a relational database to store metadata of remote data items. It also provides a virtual view of these items with user-defined collections (like directories of a file system). From a user's point of view, SRB is a virtual file system, transparent to actual data locations, and it acts as a single access interface to mass data repositories.

In 2006, iRODS [3] was developed as a successor of SRB. iRODS inherits SRB's architecture, but with an enhanced design; it also provides additional functions, the most important of which are a rule engine and micro-services. With the rule engine, customized strategies and user-defined workflows are possible [4]. Note that iRODS is open source under a BSD license, while SRB is not.

Over the past few years, SRB has been used extensively to provide an interface to massive data for many research communities around the world and has become an essential component of many data grids. Recently, iRODS has gained more interest as a replacement of SRB. For example, a few UK projects use SRB [5]: the Atlas Data Store keeps petabytes of data in tape and provides a SRB front-end with disk caches for fast data access; the Diamond Light Source project registers data on the beamlines into SRB so that this data can be synchronized to a central storage periodically and stored properly; Daresbury SRS project uses SRB for users to query metadata. Also the Arts and Humanities Data Service stores and manages large distributed data sets in SRB and is now investigating a migration from SRB to iRODS [4]. In the US, SDSC manages a number of shared collections of up to 126 million files and 830 TB for several projects, including data grids, digital libraries and persistent archives [6].

For using iRODS/SRB, SDSC provides a few basic client tools, such as Scommands/iCommands, InQ/iRODS Explorer and MySRB. Moreover, many big SRB user groups developed their own portals for SRB. However, the iRODS/SRB protocol is a proprietary protocol, which makes it hard to develop a customized client tool or use general tools to access iRODS/SRB. Therefore, it is not easy for general users to use or for developers to integrate with other software. Furthermore, these tools either have limited functionality, or are developed for a particular group. Therefore, in order to enable more people to use iRODS/SRB, it is necessary to explore other solutions, including other popular data transfer protocols and best practice of user interface. Among the available open standards, WebDAV is such an open standard based on HTTP protocol, and extensively supported by many operating systems and software. The use of WebDAV is as simple as using a file system, even to a user with very basic knowledge of computer systems. Additionally, with a broad range of software support [7] and the standard HTTP protocol, the integration with other software is effortless. Thus, the combination of WebDAV and iRODS/SRB benefits not only the end users but also the developers.

This paper introduces the system structure of iRODS/SRB and investigates their client tools and portals in Section 2. Section 3 gives details of a few popular file transfer protocols for comparison. After a description of the state of the art, reasons are given for the choice of WebDAV in this solution. Next, the detail of our implementation of a WebDAV gateway to iRODS/SRB, called Davis, is illustrated in Section 4 and some use cases are described in Section 5. Lastly, future work is mentioned after a conclusion of the current work.

II. An Overview of iRODS/SRB

In principle, SRB and iRODS are designed as a Virtual File System, with a similar structure: a metadata catalogue and a file access API [2]. The metadata catalogue (MCAT of SRB or ICAT of iRODS) stores name-value pairs (for SRB) or name-value-unit triples (for iRODS) in a relational database as metadata associated with an object in iRODS/SRB, and presents mechanisms to save, delete or query metadata. The file access API offers a virtual file system for users to access remote data in a similar way to using local file systems, and it is configurable to get access to data in various storage media, such as a remote NFS-mounted file system, a tape device, a disk array or even a zip file. On top of that, iRODS provides a rule engine, which accepts pre-defined sequences of actions to be executed in particular events like a workflow system [3]. Note that, iRODS/SRB uses a proprietary protocol requiring a specific port, which needs a dedicated iRODS/SRB client for communicating with it.

By default, iRODS/SRB only supports username/password and GSI authentication. Recently, Shibboleth [8] has evolved to become a major authentication method, with supports from many open source software tools, such as Sakai [9, 10]. The benefit of using Shibboleth is that users can use only one username and password to access different systems. However, iRODS/SRB itself does not support Shibboleth directly. The common practice is to use a Short Lived Credential Service (SLCS) server [11], which is also used to provide Shibboleth authentication to Globus services [12]. The SLCS server generates a complete certificate (with private key and public key) for valid Shibboleth Identity Provider (IdP) users. This certificate lasts only ten days by default. After that, a new certificate has to be requested again. In the login process, the user requests a SLCS certificate with an IdP account, and the SLCS server issues a certificate to the user. Then the SLCS certificate is submitted to iRODS/SRB for GSI authentication, as shown in the following diagram.

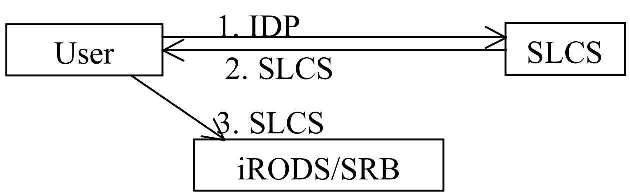

Fig. 1. GSI login with SLCS certificate

III. A Survey on Existing Client Tools of iRODS/SRB

There are a number of existing iRODS/SRB tools. Some of them are developed by SDSC and others are developed by the user community. This survey looks at the major functions of each tool from a user's point of view. In particular, the authentication methods, supported operating systems, user interface and the underlying technology are emphasized. Generally, these tools can be categorized into the following types.

A. Command-Line Interface

iCommands [13] are a group of command-line utilities written in C and developed by SDSC. They provide client tools to normal users, including functions of data transfer and metadata management, with versions for Windows, Mac and Linux. iCommands are Unix-like commands with names starting with 'i' to access iRODS objects and metadata. For example, icd acts like cd of a Unix system to change current directory. It accepts password or GSI authentication. iCommands are also handy to system administrators, providing system level functions for management of accounts, zones, resources, and domains, as well as backup and synchronization for objects. Similarly, **Scommands** [14] are a set of command-line utilities for SRB with the same purpose as iCommands, providing similar functions.

Java CoG Kit [15] is a set of command line tools written in Java for Globus. Particularly, its globus-url-copy command supports SRB protocol, which means a user can use srb://server.name/path/to/file as the source or destination of the command to copy files from or to SRB. This command allows password and GSI authentication, but it can only transfer files and has no metadata functions.

B. Graphical User Interface

InQ [14] is a native Windows GUI written in C++ offering a browser tool for data transfer (drag and drop) and metadata query on SRB, while iRODS Explorer [13] is the counterpart for iRODS. They both only support password login and can only run on Windows. Moreover, **iRODS Explorer** does not allow drag-and-drop from/to desktop at the current version.

Hermes [16] is a Java GUI jointly developed by the ARCHER project [17] and ARCS (the Australian Research Collaboration Service). It provides a rich client for data transfer and metadata management, with versions for Windows, Mac and Linux. It also supports different authentication methods, such as SRB account, GSI, MyProxy and Shibboleth. Hermes is based on Apache's Common-VFS [18], which provides a common API for accessing various file systems. The support for iRODS is under development.

JUX [19] is a Java GUI developed by CCIN2P3. It is based on JSAGA [20] and provides a browser view to SRB. It also supports preview of images and audio files. As a Java program, it can be run on Windows, Linux and Mac. JUX doesn't provide functions for metadata manipulation. iRODS will be supported in a future version.

vbrowser [21] is another Java GUI developed by VL-e (Virtual Laboratory for e-Science) mainly for the research of functional Magnetic Resonance Imaging (fMRI). It is intended to be a single access point to the grid, and provides mechanisms to access SRB and GridFTP servers. It is based on VLET, which is an abstraction layer to various middleware components, including data resources [22]. It is also extensible, with which users can develop plugins to view and process files. For example, a workflow plugin was implemented to allow users to invoke workflows in vbrowser interface [22]. However, vbrowser doesn't have any features for metadata management, and it does not support SRB 3.5 or iRODS.

YourSRB [23] is another Java GUI aiming to offer SRB data access as well as fine-grained metadata control to users who require advanced functions, such as modifying metadata of multiple files in a batch. Users can create a metadata schema and apply that to all SRB objects. It also offers a function for users to create a SRB federation. It does not support iRODS. The development of YourSRB is guided by the use cases of materials science and maritime archaeology research groups in James Cook University.

C. Web Interface/Portal

MySRB [14] is a web interface to SRB developed by SDSC. It also gives users a tool for data transfer and metadata management from a web browser, so it does not support drag-and-drop files from/to desktop. iRODS web browser gives similar functionalities to iRODS users.

SRB Plugin for Plone [24] was developed by the ARCHER project [17] and enables a SRB view in the Plone web content management system for file browsing/downloading and metadata management. This integration makes it easier to use SRB for other Plone components.

D. Other Interfaces

GridFTP DSI [25] provides a GridFTP interface for SRB. It only allows file transfer without the ability of metadata management. The GridFTP interface makes it possible to integrate SRB with Globus components. There is no equivalent GridFTP DSI for iRODS yet.

Jargon [13] is a Java API for developers and is capable of doing a variety of file operations and metadata manipulation. It does not have any user interface. Jargon is the base of all Java GUIs described above and of Davis. Jargon supports SRB and iRODS. **Prods** [13] provides a similar PHP API to iRODS.

E. Limitations of existing interfaces

To summarize, the above iRODS/SRB tools and other popular tools have limitations. Firstly, they are mostly custom clients specific to iRODS/SRB, and this requires users to learn their usage. Secondly, some iRODS/SRB client tools, especially web portals, are designed for a particular user group's needs. Thirdly, some only work for a particular operating system, and those implemented in Java can work across multiple operating systems but do not provide the look-and-feel of native programs on the OS. Furthermore, they don't follow open standards, because the iRODS/SRB protocol is not open standard. Additionally, iRODS and SRB use special ports, 1247 and 5544 respectively, which are not commonly opened on firewalls, so that iRODS/SRB tools need access to these ports. Lastly, it is not easy to integrate iRODS/SRB with other software, thus development is needed in many cases.

Therefore, it is necessary to look for another open-source-based and easy-to-use solution to overcome these disadvantages. According to the functionality presented by the current iRODS/SRB tools, as a guideline, the new tool must provide the following functions: essential file operations, such as creating, moving, renaming deleting files and directories; file permissions, such as granting/revoking a permission to/from an iRODS/SRB user on a file; metadata management, such as adding, deleting, modifying iRODS/SRB metadata; authentication, at least password authentication.

Without any modification of iRODS/SRB, it is appropriate to implement a gateway that accepts an open standard protocol and sits in front of iRODS/SRB. The next section explores a few popular file transfer protocols and justifies our choice of WebDAV as a standard interface to iRODS/SRB.

IV. Standard protocols and the choice of WebDAV

This section investigates a few popular and widely used file transfer protocols and tools, in the commercial sector or open source community. After comparing their features and network properties, WebDAV is chosen and shown to meet the above guidelines.

A. Popular file transfer protocols

File Transfer Protocol (FTP) is a basic and popular protocol for data transfer and distribution but with not support metadata. **Common Internet File System** (CIFS) [26] is an extension of the Server Message Block (SMB) protocol, and designed to request files and print services over a local network for all systems, while **Samba** (http://www.samba.org) is a typical SMB server. Most desktop operating systems, such as Linux, Mac and Windows, have built-in SMB/CIFS client. However, SMB/CIFS doesn't have metadata features and the port it uses (445) is typically blocked by firewalls.

Amazon Simple Storage Service (Amazon S3) (http://aws.amazon.com/s3) is a service that offers storage to the Internet users. It has proprietary REST and SOAP interfaces for users to access from anywhere. It is also open so users can write their own clients. Objects in Amazon S3 are identified with a unique key, and can be shared to any user. HTTP and BitTorrent (http://www.bittorrent.org) are the two protocols for data download.

WebDAV [7] stands for "Web-based Distributed Authoring and Versioning". It adds a set of extensions to the HTTP protocol so that users can edit and manage files on remote web servers collaboratively. WebDAV is defined in an IETF standard, RFC 2518 [27]. It uses the term resource for a Uniform Resource Identifier (URI), and the term collection for a set of URIs. WebDAV employs the following basic HTTP methods:

- GET – get contents and details, e.g. modification time, of a resource (file)
- PUT – upload a resource (file)
- DELETE – delete a resource (file) or collection (directory)

- HEAD – retrieve details of a resource, e.g. modification time

It also extends HTTP by adding the following methods:

- PROPFIND — retrieve properties, stored as XML, from a resource. It is also overloaded to retrieve the collection structure (directory hierarchy) of a remote system.
- PROPPATCH — change and delete multiple properties on a resource.
- MKCOL — create a collection (directory).
- COPY — copy a resource (file) or collection (directory) from one URI to another.
- MOVE — move a resource (file) or collection (directory) from one URI to another.
- LOCK — put a lock on a resource. Locks can be shared and exclusive.

The locking mechanism of WebDAV provides a way of serializing accesses to a resource, and prevents another user from writing to a file while it is being edited. The use of locking is decided by the WebDAV client. Moreover, there is a number of software products with built-in WebDAV client [7], such as Microsoft Office and PhotoShop.

B. The reasons for choosing WebDAV

Among existing open standards, WebDAV is the best option to meet our requirements. Firstly, WebDAV is an open standard, and most operating systems, such as Windows, Linux and MacOS have built-in WebDAV clients, which provide a generic interface that has the same look-and-feel as the OS itself, thus no extra software needs to be installed. Secondly, WebDAV uses HTTP(S) for any connection, wihch is commonly opened in most firewalls (if not, a HTTP proxy can be used). Moreover, it is easy and straightforward to maintain only one service for providing access to files via the web or desktop folders in a standard way. Furthermore, HTTP(S) interface is extendable, which makes it possible to implement functions not supported by standard WebDAV protocol, such as permission setting and metadata management. Lastly, it is easy to integrate HTTP/WebDAV with other software, as many of them support HTTP natively. Also WebDAV resources can be mounted to any OS and appear as a normal file system. Software can read and write data in it without modification.

V. The implementation – Davis

Davis is a WebDAV gateway to any iRODS/SRB server regardless of its location. It has two major functions: the WebDAV service, and browser mode. WebDAV service provides basic WebDAV functions to WebDAV clients that support WebDAV version 1 and 2, such as the built-in clients of Windows, Linux and MacOS, as well as other third-party software. Browser mode enables users to use a web browser, such as Internet Explorer or Firefox, for using extra functions that are not provided by WebDAV clients. Basically, browser mode shows on a browser the details of an iRODS/SRB collection, such as creation time of files and file size. On that page, users can download a file, change file permissions and modify metadata. A screenshot of browser mode is shown in Figure 2, which shows a list of files in the user's home directory on the background and permission dialog in the front. The permission dialog shows a list of permissions of a file and provides a form on the right where the user can change the permissions of this file.

Fig. 2. A screenshot of browser mode

The architecture of Davis can be described as three modules, as shown in Figure 3. The first one, WebDAV Handler handles all requests on file operations from WebDAV clients communicating with WebDAV protocol. Table 1 lists the implemented methods in Davis, as well as the corresponding iCommands, and iRODS/SRB operations.

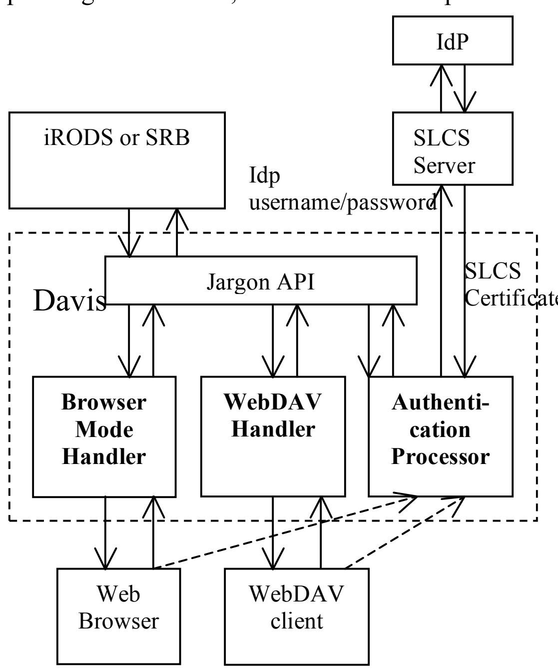

Fig. 3. Structure of Davis

The Browser Mode Handler provides web pages for metadata management and permissions setting. To simplify the structure, the asynchronous JavaScript and XML (AJAX) technology is exploited to convey requests from the web pages to Servlets. In addition, dojo toolkit (http://dojotoolkit.org), an open source modular JavaScript library, is chosen to implement the front-end web page as it provides extensible JavaScript widgets, including AJAX widgets, for web development. POST method is used by the AJAX widget to submit data from the web page to the handler, while GET method with a URL of a collection returns a web page with the details of this collection.

The Authentication Processor accepts a username and password from the client and authenticates against the appropriate method, either iRODS/SRB user system or Shibboleth. When a request comes in, the Authentication Processor will validate the credential attached in the request. If no credential is found, an HTTP status code 401 is returned for users to enter username/password. If a credential presents, different authentication methods will be used according to the username field. If the username is of the form 'IdP\account', the account is treated as an IdP account name and sent to a SLCS server to get back a SLCS certificate. Then the certificate is used for GSI authentication on iRODS/SRB. If the username doesn't have a back-slash, it will be used for normal iRODS/SRB password authentication. After the authentication is successful, requests are directed to the appropriate handler according to the request's HTTP method.

Data transfer using Davis benefits from the HTTP protocol. For instance, as an HTTP session is connection-less, there is no live connection between the client and the server, which saves server resources. As the HTTP header *Ranges* allows the client to specify the starting point (offset of a file) of a file download, it enables resumable download and segmented download. Resumable file downloads make use of server resources more efficient by allowing a user to temporarily suspend the download of a large file and then resume the download later. If the network access is temporarily interrupted or if the computer is put to sleep, the user doesn't have to resume the download from the beginning when access is restored. Segmented downloads divide a big file into segments and start off multiple threads to download each segment. When one segment ends, the client starts a new segment automatically. Downloading multiple segments concurrently can significantly improve data download speed.

VI. Performance Evaluation

The aim of this experiment is to compare the data transferring performance of Davis with iCommands.

The test environment was set up on a server with two quad-core Intel Xeon 3.16GHz CPUs and 16GB memory, running CentOS 5. iRODS 2.0.1 was used in the test with Davis on the same machine, which runs on Jetty 6.1.12 with 1.5GB heap size JVM and accepts HTTP and HTTPS requests for this test. Using HTTPS is recommended in a production service, as it is the only practical mechanism for protecting user credentials. As a consequence, it also protects the data. The client is a VM with 512MB memory and two Intel Xeon 3.20GHz CPUs. Both the server and the client are in the same local network (1Gbps Ethernet). Six tests have been performed to transfer files of 4MB, 8MB, 16MB, 32MB, 64MB, 128MB, 256MB, 512MB, 1024MB, and 2048MB. The six tests are, file upload with iput, file download with iget, file upload via Davis (HTTP and HTTPS) with curl, file download via Davis (HTTP and HTTPS) with curl. iRODS is configured to use 4MB window size and maximum 16 threads as these values are recommended by Yoshimi Iida according to her tests (https://www.irods.org/index.php/Lyon-KEK). The real number of threads is chosen by iget and iput according to the actual file size. If the file size is less than 32MB, one thread will be used. Other than that, 16 threads will be used. Each test is done 20 times and the average is shown on Figure 4.

The results show that Davis (HTTP) performs similarly to iCommands. Transfer via HTTPS is about 2-3 times slower because the overhead of SSL encryption and decryption. However, the transfer rate of HTTPS transfers is steady, mostly between 10 to 20 MBs. Similar HTTPS transfer tests have been done on a 100Mbps Ethernet network, which is a

TABLE I
DAVIS OPERATIONS

WebDAV method	Equivalent iCommand	Davis operation
COPY	icp	Copy a file or directory; if directory, it recursively copies all child directories
DELETE	irm	Delete a file or directory; if directory, it recursively deletes all child directories
GET	iget	Download a file. If the target is a directory, it is assumed to be accessed by a browser so that a HTML page is returned
MKCOL	imkdir	Create a new directory
MOVE	imv	Move a file or directory to another place
PRODFIND	ils	Get details of a directory, such as sub directories and files
PUT	iput	Upload a file
HEAD		Return basic information about the specified resource, e.g. last modified time
OPTIONS		Get a list of supported methods for the target resource
LOCK		Lock a resource before editing
UNLOCK		Unlock a locked resource
POST		Not used by WebDAV; used by browser mode to communicate with AJAX.

common network connection to a user's desktop, and the transfer rate we got from those tests ranges from 9MB/s to 11MB/s, which is the maximum transfer rate one can achieve from a 100Mbps Ethernet network. Thus, if a user is connected at 100Mbps they will see no difference in performance in using Davis compared to iCommands.

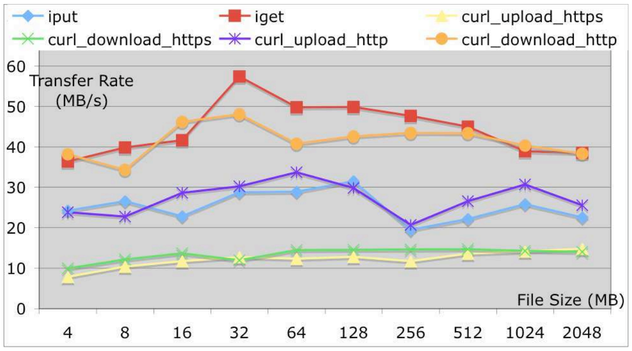

Fig. 4. Test Results of iCommands and cURL

VII. USE CASES

The integration of WebDAV and iRODS/SRB opens a window to the use of iRODS/SRB by more and more researchers. The following are a few examples.

A. *Easy access*

The biggest benefit of using WebDAV as the gateway is that iRODS/SRB collections can be mounted to the client system as a folder; thus, native clients, such as Windows Explorer, Mac Finder, and Gnome Nautilus, can be used to access data in iRODS/SRB, without requiring iRODS/SRB clients to be installed on the file system. Accessing iRODS/SRB is as easy as accessing a local hard disk, and the use of iRODS/SRB is the same as using a native program, with features like drag-and-drop. The user group will be expanded to more people, especially Windows or Mac users or those without much IT expertise.

There are WebDAV clients for hand-held devices, such as Mobiles and iPhone. A test has demonstrated that browsing and getting files from SRB is achievable on iPhone. A demo is available on YouTube to show how to access SRB from iPhone: http://www.youtube.com/watch?v=o2yu8ZJBOck.

Accessing from a PDA-like handheld device is very useful, especially for monitoring experiments remotely. For example, an experiment generates data continuously and saves it into iRODS/SRB, the researcher can monitor the process at home, or while travelling, by checking the data or metadata generated in iRODS/SRB via Davis, without staying in the laboratory, and this does not require any development or help from a system administrator.

B. *Integration with other applications*

It is easy to integrate iRODS with other applications via Davis. The following are some examples where this has been demonstrated.

Some software, such as Microsoft Office and PhotoShop, allows users to open a file directly from WebDAV, without getting it to the local drive. After editing, the file can be directly saved to iRODS/SRB via WebDAV immediately. Furthermore, as a WebDAV resource can be mounted to the system as a normal file system, legacy software can be used to read and write files on WebDAV without the need to implement support for iRODS, which requires development with iRODS APIs. This is very helpful for users who want to run a HPC job but with data stored in iRODS. They can easily mount the iRODS collection via Davis and keep using the existing software.

Another example is, via the HTTP interface, a resource URL of Davis can be set in a Handle.NET system [28] to be mapped to a unique HDL Identifier, which is a HTTP URL. If a user puts it into a web browser, Handle.NET system resolves the unique HDL Id and redirects the user to Davis, and the requested file will be downloaded automatically.

THREDDS (Thematic Realtime Environmental Distributed Data Services) Data Server [29], a popular system for sharing environmental science and remote sensing data, supports reading data via WebDAV. A test has been done to supply data stored in SRB to THREDDS Data Server via Davis. As a result, this data is available to users via the OPeNDAP protocol [30], which is supported by THREDDS. Moreover, as THREDDS Data Server can be configured to work with Web Mapping Service (WMS), it can generate a WMS overlay with its data; Google Earth can then display this overlay [31].

VIII. 7. CONCLUSION AND FUTURE WORK

This work has evaluated a number of existing iRODS/SRB client tools, which shows that the current tools are either of limited functionality, have potential firewall restrictions to the user's desktop, support SRB but not iRODS, or are customized for special needs. Normally, a new user has to spend much effort for using iRODS/SRB. Thus, investigating a new client tool has been conducted with reference to standard protocols and existing tools. Based on the investigation, a WebDAV gateway for iRODS/SRB has been implemented, for the purpose of being a generic and one-stop solution for users, developers and systems. Source code, binaries and documents of the software described in this paper (Davis) are available at http://projects.arcs.org.au/trac/davis.

Davis is deployed as a production service by ARCS, as part of the ARCS Data Fabric. Future work will address its performance and scalability. Additionally, future research will be conducted on storing and querying various metadata for iRODS via Davis.

REFERENCES

[1] R. Moor, R. Marciano, M. Wan, et al., "Towards the interoperability of web, database, and mass storage technologies for petabyte databases," in *In Proceedings of the Fifth NASA GSFC Conference on Mass Storage Systems and Technologies (College Park)*, 1996.

[2] C. Baru, R. Moore, A. Rajasekar, et al., "The SDSC Storage Resource Broker," in *In Proceedings of CASCON*, 1998.

[3] A. Rajasekar, M. Wan, R. Moore, et al., "A Prototype Rule-based Distributed Data Management System," in *HPDC*, Paris, France, 2006.

[4] H. Mark, H. Adil, and B. Tobias, "Curation and Preservation of Research Data in an iRODS Data Grid," in *Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*, 2007.

[5] T. Mortimer-Jones. (2008). SRB and iRODS work at the STFC and National Grid Service.[Online]. Available. http://www.nesc.ac.uk/action/esi/download.cfm?index=3731.

[6] R. W. Moore, "Managing Large Distributed Data Sets Using the Storage Resource Broker," *ITEA Journal of Test and Evaluation,* 2007.

[7] J. Whitehead. (2008). WebDAV Resources.[Online]. Available. http://www.webdav.org/.

[8] Internet2. (2007). Shibboleth Project - Internet 2 Middleware.[Online]. Available. http://shibboleth.internet2.edu/.

[9] The_Sakai_Foundation. (2008). Sakai Project.[Online]. Available. http://www.sakaiproject.org/portal.

[10] A. Young. (2008). Guanxi Project.[Online]. Available. http://www.guanxi.uhi.ac.uk/index.php/Guanxi.

[11] SWITCH. (2008). Short Lived Credential Service (SLCS).[Online]. Available. http://www.switch.ch/grid/slcs/.

[12] R. Groeper, C. Grimm, S. Piger, et al., "An Architecture for Authorization in Grids using Shibboleth and VOMS," in *Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications*, 2007.

[13] SDSC. (2008). iRODS website.[Online]. Available. http://www.irods.org.

[14] SDSC. (2008). SRB Website.[Online]. Available. http://www.sdsc.edu/srb/index.php.

[15] G. v. Laszewski. (2006). Java CoG Kit SRB Integration.[Online]. Available. http://wiki.cogkit.org/index.php/Java_CoG_Kit_SRB_Integration.

[16] ARCS and JCU. (2008). Hermes.[Online]. Available. http://projects.arcs.org.au/trac/commons-vfs-grid/.

[17] (2009). ARCHER project.[Online]. Available. http://archer.edu.au.

[18] Apache. (2008). Commons Virtual File System.[Online]. Available. http://commons.apache.org/vfs/.

[19] CCIN2P3. (2008). JUX (Java Universal eXplorer).[Online]. Available. http://cc.in2p3.fr/docenligne/821.

[20] IN2P3_Computer_Center. (2008). JSAGA.[Online]. Available. http://grid.in2p3.fr/jsaga/.

[21] S. D. Olabarriaga, P. T. d. Boer, K. Maheshwari, et al., "Virtual Lab for fMRI: Bridging the Usability Gap," in *Second IEEE International Conference on e-Science and Grid Computing (e-Science'06)*, 2006.

[22] T. Glatard, K. Boulebiar, and S. D. Olabarriaga, "Workflow Integration in VL-e Medical," in *21st IEEE International Symposium on Computer-Based Medical Systems*, 2008.

[23] M. J. Wyatt, N. G. D. Sim, D. L. Hardy, et al., "YourSRB: A cross platform interface for SRB and Digital Libraries," in *In Proc. Fifth Australasian Symposium on Grid Computing and e-Research*, Ballarat, Australia, 2007.

[24] JCU. (2008). SRB Plone Product.[Online]. Available. https://eresearch.jcu.edu.au/wiki/wiki/SrbPloneProduct.

[25] Globus. (2008). GT 4.0 GridFTP: Storage Resource Broker (SRB).[Online]. Available. http://www.globus.org/toolkit/docs/4.0/data/gridftp/GridFTP_SRB.html.

[26] Microsoft. (2008). CIFS Protocol.[Online]. Available. http://msdn.microsoft.com/en-us/library/aa302240.aspx.

[27] The_Internet_Society. (1999). HTTP Extensions for Distributed Authoring -- WEBDAV.[Online]. Available. http://www.ietf.org/rfc/rfc2518.txt.

[28] (2009). HANDLE.NET - The Handle System.[Online]. Available. http://www.handle.net.

[29] R. Pandya, B. Domenico, and M. Marlino, "Finding and using data in educational digital libraries," in *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, 2003.

[30] A. Woolf, K. Haines, and C. Liu, "A Web Service Model for Climate Data Access on the Grid," *International Journal of High Performance Computing Applications,* vol. 17, pp. 281-295, 2003.

[31] Google. (2008). Using Image Overlays and 3D Models [Online]. Available. http://earth.google.com/userguide/v4/ug_imageoverlays.html.