

Data Avenue: Remote Storage Resource Management in WS-PGRADE/gUSE

Ákos Hajnal[†], Zoltán Farkas[†], Péter Kacsuk^{†‡}

[†]Laboratory of Parallel and Distributed Systems
Institute for Computer Science and Control
Hungarian Academy of Sciences
Budapest, Hungary

{hajnal.akos, farkas.zoltan, kacsuk.peter}@sztaki.mta.hu

[‡]Computer Science and Software Engineering Department
University of Westminster
115 New Cavendish Street
London W1W 6UW, United Kingdom
kacsukp@westminster.ac.uk

Abstract—State-of-the-art gateways are connected to several distributed computing infrastructures (DCIs) and are able to run jobs and workflows simultaneously in all those different DCIs. However, the flexibility of accessing data storages belonging to different DCIs is a missing feature of current gateways. SZTAKI (Institute for Computer Science and Control) has developed a Data Avenue Blacktop service and a Liferay-based Data Avenue portlet that open the door for integrating such features into science gateways. The paper explains the design considerations of the Data Avenue Blacktop service and its usage scenarios in science gateways through the Data Avenue portlet.

Keywords—data storage systems, data handling, data transfer, grid computing

I. INTRODUCTION

The EU FP7 SCI-BUS project develops science gateways for more than 25 different European user communities based on the WS-PGRADE/gUSE gateway framework [1]. This framework enables the development and execution of scientific workflows where different nodes of the workflow can run on different distributed computing infrastructures (DCIs) including clusters, supercomputers, grids and clouds. Many of these communities exploit this feature of gUSE and run their workflows on several DCIs. These workflows typically process large data files that are sometimes stored in storages belonging to different DCI types. Therefore, the SCI-BUS communities required a solution to access data files stored in different DCI storage resources to be processed by gUSE workflows.

A task force was formed inside SCI-BUS to survey the requirements of the different user communities and to develop an initial specification of a service that can transfer data among different DCI storages. The initial specification had been completed by 2013 summer, and was published in IWSG'2013 [2]. Then a significant rethinking and implementation work has been started that resulted in the first version of the Data Avenue Blacktop service that was released together with a WS-PGRADE/gUSE Data Avenue portlet in September 2013. Based on the experiences gained with this version of the Data Avenue service, we have further developed the service, and its second, improved version was released in February 2014. In the current paper we report on this improved version, and show its application possibilities.

The primary goal of Data Avenue concept was twofold: on one hand, to simplify the use of advanced storage resources for different user communities, possibly having little IT competence, and on the other hand, to make it possible to access various storage resources from workflows executed in science gateways. As a result of the software engineering effort presented in this paper a high level facade to access diverse computational resources is obtained that combines grid abstraction libraries and cloud abstraction libraries. This solution also proved to be useful in bridging between different data sources.

The remainder of the paper is organized as follows. Section II describes the main concepts as well as design considerations of Data Avenue services through its typical use-cases. Section III describes security considerations. Section IV discusses related work; finally, section V concludes the paper.

II. DATA AVENUE USE-CASES

The architecture of Data Avenue was designed to cover as many use-cases as possible. The separation of the user interface and its core functionality opened a variety of ways of how Data Avenue can be exploited in different usage scenarios, and also made it possible to use Data Avenue services programmatically, besides the provided graphical user interface.

In the following subsections the typical use-cases of Data Avenue are discussed in detail.

A. Browsing Storage Resources

Data Avenue offers a uniform view, browsing capabilities, and handling of data residing on a wide range of storage resources through a portable, intuitive, web-based graphical user interface, which avoids software installation requirements, learning different tools for different storage resources, and hides technical details of using storage related protocols as much as possible.

The user interface of Data Avenue, hereafter abbreviated as UI, has been implemented as a portlet providing a web-based interface for the users. The Data Avenue UI is part of the WS-PGRADE/gUSE base portlet set, thus once a WS-PGRADE/gUSE science gateway has been set up, Data Avenue UI is usable via a simple web-browser for all portal users. In addition, as UI is freely available and downloadable as an individual portlet (open source), it can be deployed in

any portlet container independently of the WS-PGRADE/gUSE framework, which makes possible to exploit Data Avenue service in other science gateways as well. Also, a web page is available called Data Avenue @ SZTAKI, which is a public deployment of Data Avenue UI hosted at SZTAKI. On this web page, users can freely try out and use Data Avenue services without any software installation.

Data Avenue UI offers a convenient way for the users to browse and manage data on various storage resources. At the time of writing this paper, six storage resource types are accessible: HTTP(s), Secure FTP (SFTP), GridFTP [3], Storage Resource Management (SRM) [4], Amazon Simple Storage Service (S3) [5], integrated Rule-Oriented Data System (iRODS) [6]. Regardless of the particular technology, Data Avenue provides a uniform rendering and handling of data and directory structure for the users.

The graphical user interface of Data Avenue reminds us to the classical, two-panel, Norton Commander-like layout, as it is shown in Fig. 1. Files and folders are represented by icons shown as table rows along with details such as file size and last modification date; function buttons for the available operations are shown below the panels.

To connect to a storage resource the user is only required to specify host URL, type of the storage (protocol), and give authentication data needed to access the given storage resource. The URL may potentially include an initial path following hostname, called the *working path*. The required credentials depend on the particular storage resource to be connected. Examples of authentication data are: username-password (e.g., HTTP, SFTP), x509 proxy (GridFTP, SRM), access key-secret keys (S3). Once Data Avenue successfully authenticates to the server, it lists file and folder names in the

actual working path found on the connected storage resource. Navigation across folders can be performed in the usual way by double clicking on a subfolder or on the parent directory icon (“..”); single click allows of selecting a file or a folder.

To organize data on remote storage resources new subdirectories can be created (*Mkdir* button), and any file or directory can be renamed (*Rename* button), or deleted (*Delete* button), respectively.

Files can be uploaded from the local disk to the remote storage resource by clicking on the *Upload* button and selecting a local file to be uploaded; or downloaded, respectively, by selecting a remote file and clicking on the *Download* button.

Most frequently visited storage resources can be added as *favorites*, which make later connections to the same server easier and quicker. For security reasons, security-sensitive data (such as passwords, proxies) are not saved, only data such as host address, protocol, and initial working path.

As it might be seen, Data Avenue offers an intuitive user interface designed to provide a layout and controls that the users are already familiar with. This avoids the necessity of learning how to use dedicated tools, and due to uniform rendering and handling of data sets residing on various types of storage resources, Data Avenue hides technical details of accessing storage resources.

B. Data Transfer between Different Storage Resources

Copy and move operations between storage resources of different types can also be performed by Data Avenue. In the UI, we connect to the *source* storage resource in one panel (*source panel*), and connect to the target storage resource in the other one (*target panel*). In the target panel, we navigate to the target location where the data should be copied, select

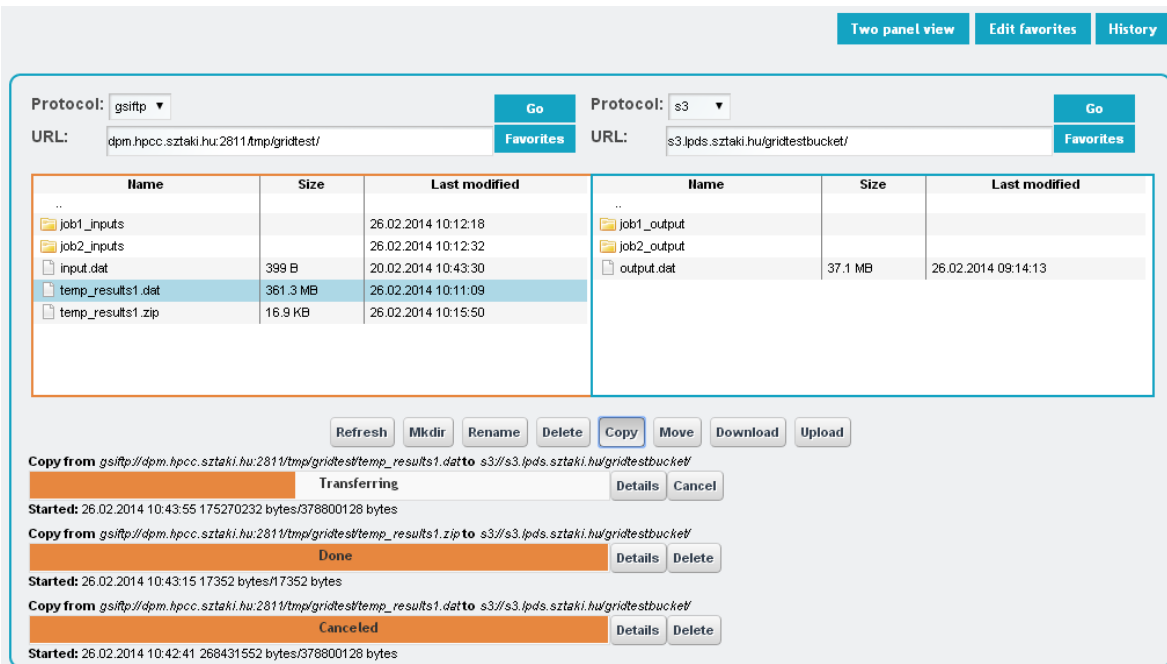


Figure 1. Data Avenue graphical user interface

a file or a folder to be transferred in the source panel, and click on the *Copy* or *Move* button as appropriate. Transfer of the selected file/folder starts immediately, as reflected by the related progress bar below the panels along with status information about the task (see Fig. 1). Active transfers can be aborted (*Cancel* button), furthermore, details about the transfer can be obtained by clicking on the *Details* button. The status of the task turns from *Transferring* to *Done* as soon as the copy operation is complete – or *Failed*, respectively, if any error occurred during transferring (*failure cause* indicates the cause of failure).

Such tasks are performed by the Data Avenue service in the background, which keep running even if the user leaves the web page. Status of previously initiated transfers can be viewed at any time later, revisiting the UI (and clicking on the *History* button). History entries not yet of interest can be removed from the list (*Delete* button).

Such *asynchronous* transfers are ensured by a central component of the Data Avenue architecture, called the Blacktop. The Blacktop is a web application providing web service interface for Data Avenue clients – among them, the UI itself, which basically serves as a lightweight graphical user interface layer for Data Avenue.

C. Data Avenue Web Services

As mentioned earlier, the core functionality of Data Avenue services is realized by the Blacktop component. Blacktop services are publicly available through a standard web service interface over HTTP (via SOAP messages or REST), hosted at SZTAKI. The decomposition of core services from the UI opens further possibilities of exploiting Data Avenue services other than through the UI: it makes it possible use Data Avenue from Java applications based on the publicly available Java library provided, or from applications written in other programming languages via standard web service calls, using the appropriate web service API available in that environment. These use-cases are illustrated in Fig. 2.

Web service operations – whose functionalities are reflected in the user interface presented in the previous section – include directory contents listing (*list*), directory creation (*mkdir*), directory deletion (*rmdir*), file deletion (*delete*), and file- or directory renaming (*rename*). In addition, meta-information retrieval operations are available to query what storage resource types are supported by the Blacktop, what credentials are needed to authenticate to a given storage resource, and what sort of operations are available on the selected storage (read, write, directory creation, etc.).

The Blacktop was designed as a plug-in architecture, where different plugins, called the adaptors implement “low-level access” to storage resources of specific type(s). This architecture is shown in Fig. 3.

Web service operation requests sent to the Blacktop use Uniform Resource Identifiers (URIs) to refer to storage resources (files or folders), which are of the form: `protocol://host/path`, where *protocol* specifies the protocol used to communicate with the storage resource, *host*

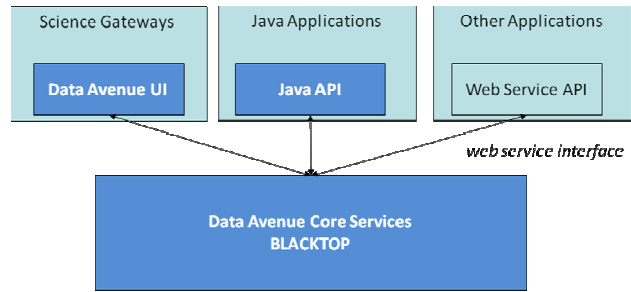


Figure 2. Use-cases of Data Avenue web services

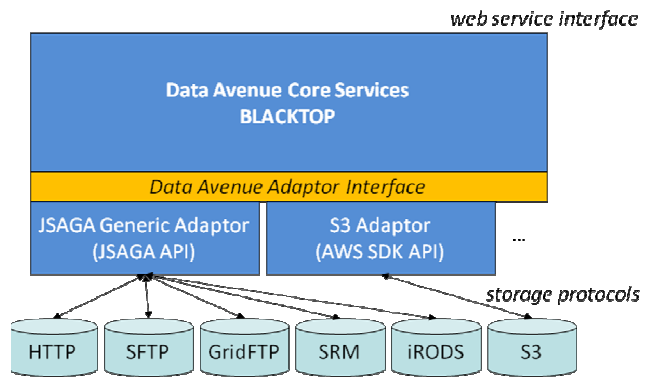


Figure 3. Blacktop adaptor architecture

specifies the internet location of the storage, and *path* specifies the location where the file or a folder resides within the given storage resource. Based on the protocol (scheme) part of the URI, the Blacktop can choose the appropriate adaptor (based on Blacktop’s internal *Adaptor Registry*), and dispatch the operation request.

JSAGA Generic Adaptor is based on JSAGA API [7], which is a Java implementation of the Simple API for Grid Applications (SAGA) specification from the Open Grid Forum. Using JSAGA API, storage resources HTTP(s), SFTP, GridFTP, SRM, and iRODS can be accessed and managed. In addition to *synchronous* operations such as directory listing, creation, renaming or deletion, JSAGA API itself makes it possible to copy or move files or folders *asynchronously* between storage resources that it supports (which is done via monitorable *tasks* that run in individual threads). Blacktop ensures persistence for such tasks, and delegates task status and progress information for the users on query, and provides task abortion possibility, respectively.

Transferring data between storage resources handled by different adaptors of the Blacktop (e.g., from GridFTP to S3) is done via streaming. The copy operation is performed by opening and reading the input stream of the source file (provided by the source adaptor), and creating and writing the output stream of the target file (provided by the target adaptor). This process is entirely managed by the Blacktop, tracking size, bytes transferred and progress information as long as the operation completes. On copying directories, target directory structure reflecting the source structure is

also created. Copy and move operations performed by Data Avenue are maintained and persisted by the *Transfer Manager* module of the Blacktop, through which status, progress information as well as the history of previous transfers can be obtained.

Efficiency measurement of data transfers done through Data Avenue and the comparison of I/O rates with tools capable of performing the same task are out of the scope in this paper. Web service calls used to launch copy tasks is an overhead; moreover, Blacktop load and different network configurations may also have influence on the efficiency of these transfers. We note that Blacktop is capable of using third-party transfer whenever possible to lower CPU and network load of the Blacktop. Such transfers are typically more efficient, as these are done directly between source and target storage resources. For example, transfers between GridFTP servers or between different S3 regions are performed using third-party transfers.

III. SECURITY CONSIDERATIONS

As with using any mediation service, a number of security concerns may arise. This section describes the measures designed and implemented to preserve confidentiality of users' security-sensitive data.

Data Avenue services are available over HTTPS connection, which ensures confidentiality of data passed through this channel to the Blacktop. The Blacktop is hosted in a secured infrastructure; in addition, to increase security within the Blacktop itself, user credentials are stored in the system memory only, kept for the duration of the client session (erased after a specific period of time of user inactivity).

The security of the communication between the Blacktop and the different storage resources is ensured by the corresponding protocol used to access the storage resource. For example, communication with GridFTP, SFTP, SRM, or S3 servers accessible over HTTPS protocol is considered to be secure due to the underlying communication layer. Transferring data from HTTP servers is unsecure, but HTTP servers are "read-only", which prevents exposing confidential data accidentally, i.e. to copy data to unsecure HTTP location.

When using third-party transfer, use of Data Avenue does not affect security, and as long as the Blacktop is not compromised, transfer of data from one storage resource to another is as safe as the protocols used to access the individual storage resources.

IV. RELATED WORK

Numerous tools exist capable of managing data residing on storage resources such as GridFTP GUI [8], DragonDisk [9], Cyberduck [10], Transmit [11]. These tools are typically able to access storage resources of a certain type, or a few, accessible over closely related protocols. Each tool has its own graphical user interface, specific usage, furthermore, to access a particular storage resource the appropriate client software needs to be downloaded, installed and configured first. Data Avenue addresses these issues by offering a uniform rendering and handling of data residing on different

storage resources through an intuitive, web-based graphical user interface, which avoids software installation and configuration requirements as well as learning of usage as much as possible.

Most tools available to manage storage resources offer the possibility to download data from the remote storage to the local machine, or upload data from local disk, respectively. Transferring data from one storage to another, however, might be very difficult using these tools. Downloading data from a *source* storage to the local disk, then uploading them to the *target* storage may be inefficient, or even impossible, e.g., when data set size exceed our local disk capacity, or file sizes on the source storage resource exceed file size limit of our personal computer. Data Avenue can manage transferring data between storages of arbitrary size due to copying via streaming (though limitations may be implied by the storage resources themselves).

Globus Online [12, 13] provides high performance, secure, third party data movement and synchronization between Globus "endpoints" accessible over GridFTP protocol. Globus Online is available on the web and as a web service (REST API) too. In addition to data transfers, it offers group management and authorization services. Authentication to access endpoints is based on MyProxy. Data Avenue provides similar services, and though MyProxy is not yet among the supported authentication mechanisms, it offers access to a wider set of storage resource types.

JSAGA API [7] used by Data Avenue's JSAGA Generic Adaptor has also an extensible plug-in architecture, and in addition to storage access, it is capable of launching and managing jobs on grid infrastructures. Data Avenue exploits a subset of JSAGA capabilities, namely, data adaptors of protocols related to physical storage resources. JSAGA proved to be a reliable API for performing Data Avenue tasks, however, serving multiple, concurrent users, and providing a web based graphical interface had to be implemented by Data Avenue. In addition, using the S3 Adaptor, Data Avenue can access cloud storages as well.

Parrot [14] is a transparent virtual file system that allows any ordinary program to be attached to many different remote storage systems, including HDFS, iRODS, Chirp, and FTP. It can be applied to almost any program without re-writing, re-linking, or re-installing. Parrot is useful for running batch jobs in large scale distributed systems. Parrot could also be used as a data access layer in Data Avenue for accessing further storage resources; in the case of massive, concurrent use, however, mounting/unmounting operations required by Parrot might be expensive.

Data Avenue uses Amazon's AWS SDK [15] to access S3 storages. BlobStore API of Apache jclouds [16] is also a promising alternative to access cloud storages of a wider range.

V. CONCLUSIONS

Data Avenue service opens a new horizon for workflows accessing several DCIs. WS-PGRADE/gUSE workflow nodes were able to run on many different kinds of distributed computing infrastructures (clusters, supercomputers, grids and clouds), but transfer of data among these DCIs requires

very special solutions. From now on by using the Data Avenue services WS-PGRADE/gUSE workflows will get full flexibility concerning the DCIs both from the point of view of code execution and data management. The integration of the WS-PGRADE/gUSE workflow manager with the Data Avenue service has been started, and the new WS-PGRADE/gUSE release that contains this feature is expected at the end of May 2014. This integration work will be reported in a forthcoming paper afterwards.

Of course, Data Avenue services can be used by other workflow systems and gateways as well. The Data Avenue service is integrated into a Liferay portlet, and this portlet is stored in the SCI-BUS Portlet Repository, which is publicly available for every gateway developer. As a consequence, any gateway that is based on Liferay can also use this service. Further on, Workflow developers can follow the idea of how to integrate the Data Avenue service with a workflow manager by downloading the code of the WS-PGRADE portal from sourceforge once the integration work has been finished.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 283481 (SCI-BUS), 312579 (ER-Flow), 607380 (VIALACTEA), and 608886 (CloudSME).

REFERENCES

- [1] P. Kacsuk, Z. Farkas, M. Kozlovsky, G. Herman, A. Balasko, K. Karoczkai, and I. Marton, "WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities", *Journal of Grid Computing*, Volume 10, Issue 4, pp. 601–630.
- [2] Z. Farkas, P. Kacsuk, A. Balasko, K. Karoczkai, M. Santcross, S. Olabariaga, "Data Bridge: solving diverse data access in scientific applications.", in proceedings of the 5th International Workshop on Science Gateways. Zurich, Switzerland, 3–5 June, 2013 Online proceedings.
- [3] W. Allcock, "GridFTP: Protocol Extensions to FTP for the Grid", *Global Grid ForumGFD-R-P.020*, 2003.
- [4] A. Shoshani, "Storage Resource Management", GGF-4, 2002. <https://sdm.lbl.gov/srm-wg/doc/02.02.srm.joint.design/index.htm> [19 March 2014]
- [5] Amazon Simple Storage Service. <http://aws.amazon.com/s3> [19 March 2014].
- [6] Integrated Rule-Oriented Data System. <https://www.irods.org> [19 March 2014].
- [7] Java implementation of the Simple API for Grid Applications. <http://software.in2p3.fr/jsaga/> [25 February 2014]
- [8] W. Liu, R. Kettimuthu, B. Tieman, R. Madduri, B. Li, and I. Foster, "GridFTP GUI: An easy and efficient way to transfer data in grid", in proceedings of the Third International ICST Conference on Networks for Grid Applications (GridNets 2009), Athens, Greece, Sep. 2009, pp. 57–66.
- [9] DragonDisk. <http://www.dragondisk.com> [19 March 2014]
- [10] Cyberduck. <http://cyberduck.io> [19 March 2014]
- [11] Transmit. <https://panic.com/transmit> [19 March 2014]
- [12] A. William, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, and I. Foster, "The Globus striped GridFTP framework and server", in proceedings of the 2005 ACM/IEEE conference on Supercomputing, IEEE Computer Society, 2005, p. 54.
- [13] <https://www.globus.org> [19 March 2014]
- [14] D. Thain and M. Livny, "Parrot: Transparent user-level middleware for data intensive computing", in proceedings of Workshop on Adaptive Grid Middleware at PACT, January, 2003.
- [15] AWS SDK for Java. <http://aws.amazon.com/sdkforjava> [19 March 2014]
- [16] Apache jclouds. <http://jclouds.apache.org> [19 March 2014]