Advancing Distributed Data Management for the HydroShare hydrologic information system

Hong Yi*, Ray Idaszak, Michael Stealey, Chris Calloway

Renaissance Computing Institute, Chapel Hill, North Carolina, USA

Alva Couch

Tufts University, Medford, Massachusetts, USA

David G. Tarboton

Utah State University, Logan, Utah, USA

Highlights

- Uses iRODS as a back end in HydroShare to facilitate replication to the off site data store for disaster recovery.

- Employs iRODS federation to enable a partner institution to add storage into the HydroShare distributed data storage system.

- Provides iRODS user space to enable users to upload large files using iRODS clients and to add them into HydroShare resources.

- Uses iRODS rules and commands for on-demand resource bagging to take data operation close to data for enhanced performance.

- Enables interoperability with other iRODS-based systems, and provides fast transfer of big data to and from supercomputers

Advancing Distributed Data Management for the HydroShare Hydrologic Information System

Hong Yi*, Ray Idaszak, Michael Stealey, Chris Calloway

Renaissance Computing Institute (RENCI), University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, USA

Alva Couch

Tufts University, Medford, Massachusetts, USA

David Tarboton

Utah State University, Logan, Utah, USA

Abstract

HydroShare (https://www.hydroshare.org) is an online collaborative system to support the open sharing of hydrologic data, analytical tools, and computer models. Hydrologic data and models are often large, extending to multi-gigabyte or terabyte scale, and as a result, the scalability of centralized data management poses challenges for a system such as HydroShare. A distributed data management framework that enables distributed physical data storage and management in multiple locations thus becomes a necessity. We use the iRODS (Integrated Rule-Oriented Data System) data grid middleware as the distributed data storage and management back end in HydroShare. iRODS provides a unified virtual file system for distributed physical storages in multiple locations and enables data federation across geographically dispersed institutions around the world. In this paper, we describe the iRODS-based distributed data management approaches implemented in HydroShare to provide a practical demonstration of a production system for supporting big data in the environmental sciences.

Keywords: distributed data management, big data, data sharing, hydrologic

*Corresponding author
Email address:* `hongyi@renci.org` (Hong Yi)

information systems, collaborative environment, iRODS

1. Introduction

To enable more rapid scientific advances and discovery, it is critical to enable reproducible science and facilitate the ability for scientists to share their work and build on the work of others. Effective data discovery and reuse requires

a comprehensive cyberinfrastructure that enables diverse data with different types to be annotated, discovered, accessed, visualized, analyzed, published, and serves as a platform for collaboration. HydroShare (http://www.hydroshare.org) is one example of that cyberinfrastructure; it is a web-based collaboration system for cataloging and sharing hydrologic data, models, and tools to enable

more rapid advances in hydrologic understanding via collaborative data sharing, analysis, and modeling. Using HydroShare, scientists can easily discover, access, share, and collaboratively analyze hydrologic data and models, and hence accelerate hydrologic scientific discovery. Development of HydroShare was funded by the U.S. National Science Foundation (NSF) through its Software Infras-

tructure for Sustained Innovation program (awards 1148453 and 1148090, 2012-2017). The Consortium of Universities for the Advancement of Hydrologic Sciences, Inc. (CUAHSI) has assumed responsibility for operation of HydroShare as one of its water data services, and a new NSF collaborative award (1664061, 1664018, 1664119, 2017-2021) will further advance the modeling, collaboration,

storage and extensibility capabilities of HydroShare. This paper reports on the HydroShare distributed data storage and management system developed using an open source data grid middleware called iRODS (Integrated Rule-Oriented Data System) (Moore (2008); Russell et al. (2016)).

HydroShare is constructed from several coupled components, including a

Django front end (written in the programming language Python) that serves as a user interface, an iRODS distributed file system back end that stores files and data, a SOLR search engine (http://lucene.apache.org/solr/) that enables data discovery, and a variety of application servers that access iRODS and SOLR

via a Representative State Transfer (REST) application programming interface (API). This loose coupling of application servers allows extensibility as needs of hydrologic science evolve over time. (Refer to Section 2 and Figure 1 for a more detailed description of these coupled components in HydroShare.)

Hydrologic data and models can be large, often at multi-gigabyte or terabyte scales. To cope with large data sets, we use the iRODS data grid middleware as the distributed network data storage and management back end in HydroShare. iRODS provides a unified virtual file system for physical storage distributed across multiple locations and enables data federation across geographically dispersed institutions around the world. Using iRODS enables HydroShare to work with large files efficiently by leveraging iRODS parallel file transfer capability and addresses big data management challenges including system expansion, ability to deliver data to analysis applications efficiently, efficient data packaging, and off-site data replication for disaster recovery.

iRODS provides a distributed virtual framework for managing physical storages across multiple locations and enables storage federation across geographically dispersed institutions around the world. It is not a prepackaged solution, but rather, a middleware with pluggable architecture that supports developer-customizable policies at every point of the data management life cycle (Russell et al. (2016)), so that users are not limited by a pre-defined set of features. The flexibility and extensibility enabled by this pluggable architecture in iRODS allows us to move time-consuming data operations to iRODS, and to customize HydroShare quota management policies in iRODS. In addition, the storage federation capability in iRODS allows us to create a federated data storage system in HydroShare so that partner institutions can share the burden of storing large data sets.

In the following sections, we first provide background for our work, then describe how iRODS is used in HydroShare for managing big data in hydrological and environmental sciences. In addition, we discuss the strengths and weaknesses of our data management approach in the context of HydroShare. This is followed by discussion of the approach, use cases, and future work. Finally, we

3

2. Background

An overview of the functionality envisioned for HydroShare was given in Tarboton et al. (2014) and the initial software architecture for HydroShare was described in Heard et al. (2014). Idaszak et al. (2016) provided a case study of the application of modern software engineering to HydroShare. Horsburgh et al. (2015) described the data model and content packaging scheme for diverse hydrologic datasets and models used by HydroShare to enable storage, management, sharing, publication, and annotation of the diverse types of data and models used by hydrologic scientists. These diverse types of data and models are packaged into a resource bag using the BagIt File Packaging Format (Boyko et al. (2012)) for storing resources on disk and for serializing them to zipped files for transfer. We refer to this resource packaging operation as *resource bagging* hereafter for easier reference.

A *resource* is the granular unit used for data management and access control within HydroShare. Physically, a resource is a directory in a file system that adheres to the structure of the BagIt format. HydroShare uses a resource-centric approach in which resources are objects that can be created, stored, modified, versioned, shared, annotated, discovered, accessed, published, and acted upon by web apps independent of HydroShare. As shown in Figure 1, HydroShare's functionality and architecture can be organized into three categories: (1) iRODS-based distributed resource storage and management, (2) resource exploration, and (3) actions on resources. Each is implemented using system components that interact through APIs. The loose coupling between HydroShare and web apps enables extensibility in that anyone can develop a web app that interacts with resources stored in HydroShare. An example of a web app for visualizing spatial data in HydroShare is described in Crawley et al. (2017).
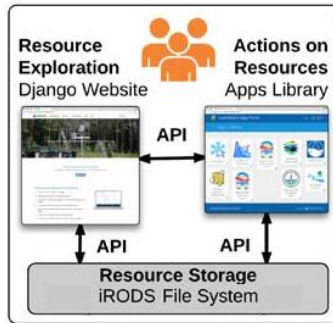
Figure 1: HydroShare component architecture

Vitolo et al. (2015) recently gave an overview of using web-based technologies
₉₀ for processing big data in the context of environmental sciences and highlighted
the fact that using web and cloud-based technologies for big data analysis is
increasingly acknowledged in the environmental community. There has been
increased development and availability of data repositories and archival systems
for the last several years, including FigShare (Hane (2013)), the NSF supported
₉₅ DataNet Federation Consortium (http://datafed.org/), DataONE (Cao et al.
(2016)), SEAD (Myers et al. (2015)), and CyVerse (Merchant et al. (2016);
Oliver et al. (2013)). These systems all have a similar basic structure: a data
store, the ability to "publish" data for public consumption, and ability to specify
metadata for published data so that it can be discovered by other researchers
₁₀₀ involved in similar tasks.

HydroShare enables interoperability with other digital repositories via a
standards-based approach to data storage and metadata. HydroShare's data
model (an adaptation of the Open Archive Initiative's Object Reuse and Ex-
change (OAI-ORE) standard Lagoze et al. (2008)), metadata structure (Dublin
₁₀₅ Core with extensions DCMI (2012)), and packaging scheme (the BagIt hier-
archical file packaging format) all use well-known standards within the digital
preservation and archival community. HydroShare's structured and comprehen-
sive resource types and metadata descriptions not only facilitate better interpre-

5

tation by users, but also enable users to write independent applications designed to operate on this structured content.

Unlike many other data publication systems, HydroShare enforces a strict compliance to metadata standards at all times, and strict synchronization between metadata and the presence of specific resource files. The reason for this requirement is that HydroShare allows authorized users to treat the resource as if published at any time, and download, use, and then re-upload the resource as part of the data life cycle, even before the resource is formally published. By contrast, many publication systems prohibit download until metadata is synchronized with object contents. That mode of operation would interfere with a core goal of HydroShare: to allow resource sharing *before* (or even without) formal publication.

HydroShare can also interoperate with other data repositories via its iRODS-based distributed data storage and management system. iRODS is used by large scientific research projects across the country and around the globe for managing petabytes of data in hundreds of millions of files on distributed storage resources. For example, Hedges et al. (2009) presented a rule-based data grid approach using iRODS for automatic data curation and preservation. Schnase et al. (2011) presented their iRODS-based approach for combining disparate data collections into a federated platform for implementing various data services in NASA Center for Climate Simulation. Sukhija and Datta (2013) presented a community grid web portal for storing, managing, and sharing large amounts of community health data in a data grid managed by iRODS. Using iRODS as the distributed data management back end in HydroShare enables HydroShare to inter-operate with other data repositories (*e.g.*, CyVerse) at the data layer via iRODS.

In the following sections, we describe the iRODS-based distributed data storage and management architecture and approach in HydroShare and discuss strengths and weaknesses of the approach in the context of HydroShare to provide reference for other similar systems. Some use cases enabled by the iRODS-based approach are also discussed.

6

3. iRODS-based big data management in HydroShare

HydroShare was implemented using the Django web framework with iRODS as back end storage for distributed data management. In HydroShare, a resource is stored in iRODS as a collection of data that is accessible as one unit. Physically, a HydroShare resource is stored as a directory of files adhering to the Bagit format specification. The name of the directory (a *collection* in iRODS terminology) for a HydroShare resource is a randomly generated universally unique identifier (UUID) that is used to uniquely identify a resource in HydroShare. Figure 2 shows an example of a HydroShare resource stored in iRODS and conforming to Bagit format.
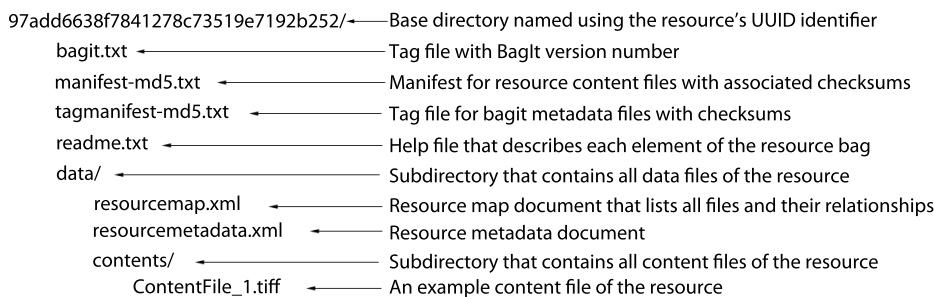
```
97add6638f7841278c73519e7192b252/ ◄── Base directory named using the resource's UUID identifier
             bagit.txt ◄─────────────── Tag file with BagIt version number
      manifest-md5.txt ◄─────────────── Manifest for resource content files with associated checksums
   tagmanifest-md5.txt ◄─────────────── Tag file for bagit metadata files with checksums
            readme.txt ◄─────────────── Help file that describes each element of the resource bag
                 data/ ◄─────────────── Subdirectory that contains all data files of the resource
           resourcemap.xml ◄─────────── Resource map document that lists all files and their relationships
      resourcemetadata.xml ◄─────────── Resource metadata document
             contents/ ◄─────────────── Subdirectory that contains all content files of the resource
            ContentFile_1.tiff ◄─────── An example content file of the resource
```

Figure 2: An example of a HydroShare resource collection stored in iRODS

A HydroShare resource differs from a regular directory of a computer file system in several ways. The resource directory in HydroShare is tightly linked with a data catalog maintained in Django, which records both the structure of and the metadata for that directory. HydroShare supports a number of *resource types* that constrain the contents of each HydroShare resource directory (and metadata) in a manner that makes the resulting HydroShare resource suitable for publication. Thus, one must modify the files and metadata for a resource directory in concert. Additionally, HydroShare allows a user to formally publish a resource as citable, which has the effect of prohibiting all further changes and providing a Digital Object Identifier (DOI) suitable for data citation.

For this reason, within an iRODS server files in HydroShare resources are

not owned by a specific HydroShare iRODS user, but by an independent single iRODS *proxy user* that the HydroShare software system controls. In this manner, the HydroShare system assures that the resource files and data catalog remain synchronized and immune from modification in iRODS outside of HydroShare. The *proxy user* assures that resources cannot be changed outside the HydroShare environment; if a resource was owned by a regular user in iRODS, an iRODS client outside of HydroShare could be used and thus changes could not be prevented. The decision to make HydroShare files owned by a dedicated iRODS *proxy user* gives rise to other practical issues, including how user disk usage and quotas are managed. HydroShare uses a hybrid approach that combines HydroShare custom software and native iRODS quota mechanisms for disk quota management as described in detail in section 3.3.

One important difference between HydroShare and other data publication systems is that it also supports pre-publication workflows involving many people and software applications. The access control system allows users to share their data only with specific users, form groups of users, and share data with those groups, both for the purpose of data use and collaborative data collection. Due to its inclusion of these dynamic and private pre-publication datasets, HydroShare uses more storage than a publication only system would use and supports all phases of the data creation and usage life cycle, including but not limited to data publication.

Thus another practical consideration in our design is that the project cannot afford to pay for storing all of the data that everyone might need to store. iRODS provides a straightforward solution in its mechanism for *federation*, which allows a set of geographically distributed disks to act like a single file storage device. Therefore, independent users or their institutions can add storage to HydroShare federated storage, where the collection of all available storage is managed as a single entity in HydroShare for the purpose of storing resources.

These basic building blocks—iRODS, the *proxy user*, storage federation, and a Django-based resource editor—are the elements from which big data management is built. The Django-based resource editor allows creation and manipula-

8

tion of resources, while the other components provide a storage medium that is scalable both in a technological and economic sense.

3.1. Working around http file transfer limits

Since uploading or downloading large (multi-gigabyte) files through the HTTP protocol from a web browser is problematic, most web applications impose some limit on the size of the file to be uploaded or downloaded from a web browser. In the HydroShare web interface, we support direct upload or download of files from local disk up to 1GB in size. For files larger than 1GB, we leverage iRODS file transfer capability to transfer files of arbitrary size. Hünich and Müller-Pfefferkorn (2010) gave a performance analysis of IRODS for managing large datasets with stress tests designed to test iRODS file transfer performance under the conditions of a large number of requests to transfer millions of files.

iRODS manages data objects and collections in independent administrative units called *zones* which are autonomous with their own security and users; these zones can be federated to form a distributed data management system with customized access control protocol. Each iRODS zone contains an iRODS catalog-enabled resource server which uses a relational database to organize the content of the zone and to maintain iRODS metadata. Files uploaded to the HydroShare Django server or selected from an iRODS zone are efficiently transferred to the main HydroShare iRODS zone via an iRODS client called iCommands. For large files (larger than 32MB by default), iRODS uses multiple threads to transfer data in parallel, which provides iRODS with a significant speed advantage over standard file transfer protocols (e.g., FTP).

3.2. Using iRODS Federation

In HydroShare, we employ a distributed data management approach based on iRODS zone federation. iRODS federation supports extensibility of the HydroShare storage and quota management system by providing partner organizations the option to store HydroShare resources on their own iRODS servers, to increase storage capacity for their users beyond the finite capacity provided by

9

HydroShare main storage. To support such an extensible distributed data storage ecosystem in HydroShare for better managing future growth, we are developing policies that describe how partner organizations can extend HydroShare main storage, and are also in the final process of developing and testing a distributed storage quota management system. This distributed storage quota management system will allow partner organizations to manage their own storage quota as needed once deployed in the near future.

The core of this federation-based approach is to store HydroShare resources in different iRODS zones as independent administrative units. These units can be located in geographically dispersed locations physically, yet all these zones are logically federated with the main zone for centralized data management in HydroShare.

The three main advantages of this federation approach are security, speed, and physically distributed storage. Each zone is a separate data administrative unit that has its own iRODS server and catalog database, which enables its own security boundary that is not exposed to other zones. On the other hand, all zones are federated via iRODS with authorization and access control that allows direct access to files across different zones via the iRODS *proxy user* in HydroShare. In addition, this distributed data storage and management approach allows us to execute *resource bagging*, a time-consuming operation especially for large resources, on the server where the data resides, boosting performance. This was done by creating *resource bagging* code in iRODS to avoid having to move data to the Django server.

Figure 3 shows a schematic representation of the iRODS zone federation-based distributed data management approach in HydroShare. The main HydroShare iRODS zone and HydroShare iRODS user zone are federated internally and grouped as internal HydroShare federated zones. On the file access level in the iRODS layer, a *proxy iRODS user* owns all HydroShare resources in iRODS. This prohibits users from making any changes to HydroShare resources except through the Django user interface and REST API, which act as the *proxy iRODS user*, and enforces an ownership and access control model defined and
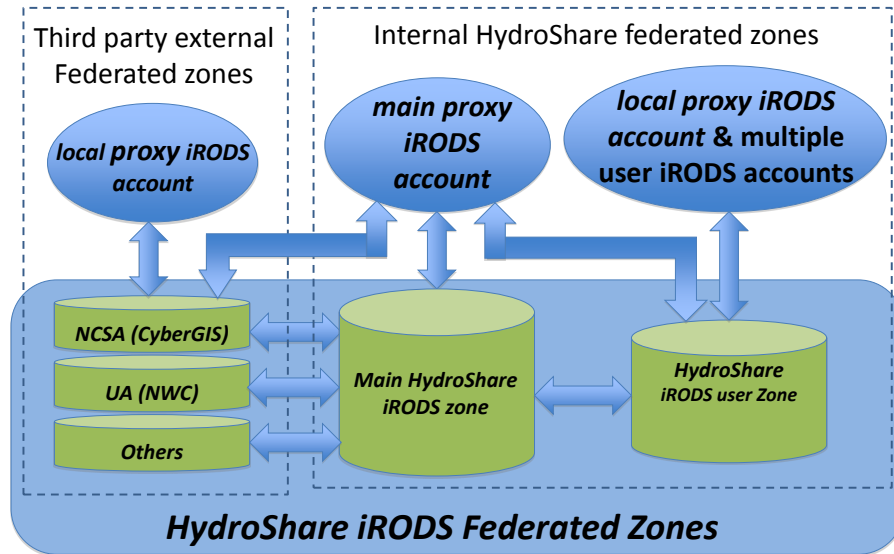
10

Figure 3: Schematic representation of the iRODS zone federation-based distributed data management approach in HydroShare

managed in Django (Couch et al. (2015)).

Federated iRODS servers are configured similarly—but not identically—to the main iRODS server for HydroShare. Each federated zone has its own dedicated *local proxy iRODS user* for the files it stores. By default, HydroShare stores new resources in the main HydroShare zone under the *main proxy iRODS user*. However, when a user asks HydroShare to create a resource from files stored in a federated iRODS zone, the HydroShare resource will be created in the same federated iRODS zone and owned by this *local proxy iRODS user*, which serves as the counterpart of the *main proxy iRODS user* in this local federated zone. A partner's federated iRODS zone must be configured to allow HydroShare read and write access to its *local proxy iRODS user* account; however, the partner's *local proxy iRODS user* account is prohibited from accessing the *main proxy iRODS user* account. Thus, the partner gives HydroShare the (limited) right to modify a local iRODS zone but does not gain the ability to directly modify the main HydroShare iRODS store.

11

Changing a file in a HydroShare zone without permission and without inter-action with the Django application can corrupt the resource from the perspective of the HydroShare system. This, in turn, would make future downloads of the resource invalid. Thus, no iRODS users other than the *proxy iRODS users* are allowed to modify HydroShare resources directly. A partner must agree not to modify its federated HydroShare zone except through the common HydroShare user interface and REST API that uses the *main proxy iRODS user* for file operations, thus preserving data integrity of the HydroShare resources.

As aforementioned, users can employ any iRODS zone as a data source to add files to HydroShare without being limited by file upload size from a web browser. The iRODS zone may be an independent iRODS data store such as CyVerse, or a zone federated with the main HydroShare zone. If users do not have an iRODS account in the HydroShare user zone, they may go to their profile page in HydroShare and create an iRODS account in the HydroShare user zone for creating "big data" resources.
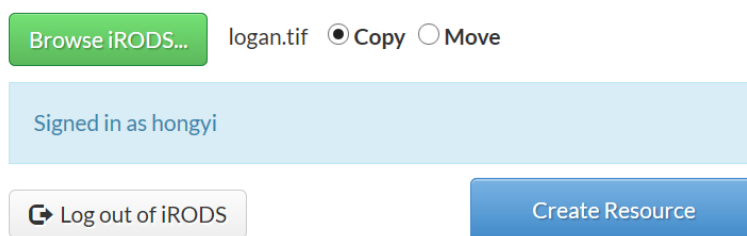

Figure 4: Assembled interface components in HydroShare for creating a resource from files in the federated iRODS user zone

Figure 4 shows a screen capture of assembled interface components in Hy-droShare for creating a resource using large files in the federated iRODS user zone. HydroShare detects whether or not the selected files come from an iRODS zone federated with the main HydroShare zone automatically in a manner that

is transparent to users, and handles federated files differently as needed. Specifically, if the selected files come from a federated iRODS zone (*e.g.*, the internal HydroShare user zone), a move-or-copy option allows the HydroShare user to choose to have the selected files either moved or copied into the resource. Moving the files in the same iRODS zone is a logical (and instant) operation since the physical file on the same iRODS server is not moved but just registered logically to a different location in the iRODS catalog, and thus the move option executes quickly on big data. By contrast, the copy operation copies the physical files efficiently via multiple threads on the same or a federated server, which necessarily takes more time than the move option on big data. The copy option applies to users who want to keep files in their personal IRODS user space for further use outside HydroShare.

3.3. Quota management

The strong requirement for data integrity and synchronization between Django and iRODS views of HydroShare resources requires the use of an iRODS *proxy user* that acts on behalf of Django rather than being regular iRODS users. Thus, quota management in HydroShare requires a hybrid approach involving both Django and iRODS. Use of regular iRODS user files in personal iRODS user space in a federated iRODS zone is monitored via iRODS methods, while ownership of HydroShare resources managed by the "proxy user" is determined by the Django server. Django's concepts of resource ownership are communicated to iRODS by setting iRODS "Attribute-Value-Units" (AVUs), which are key-value metadata stored directly in iRODS and associated with any iRODS file, folder, user, or storage resource. Specifically, the Django server records the quota user for each proxy-owned HydroShare resource folder in its AVUs to communicate this ownership to iRODS. Thus, much quota management can be done outside of HydroShare, in iRODS, using this extra metadata. Conversely, iRODS sets different AVUs on the iRODS *proxy user* to communicate total quota usage for all HydroShare users back to Django, for quota notification and enforcement.

Two quota related data models are managed on the HydroShare Django front-end server. The user quota model records quota allocation and usage information for each HydroShare user. The quota message model holds quota warning and enforcement messages and criteria variables which may be customized with different values as needed from the Django administrative web console, by a system administrator. These quota warning and enforcement criteria variables include quota soft and hard limit percentages and grace period days that are all configurable from the administrative console.

Table 1 shows actions taken in HydroShare under various conditions. These actions apply to both internal and external federated zones in HydroShare. When users exceed their quota, they may no longer add more files to HydroShare until they reduce their quota usage below the allocated quota or have been granted increased quota allocation.

Table 1: QUOTA ACTIONS TAKEN UNDER DIFFERENT CONDITIONS

Condition	Action
below soft limit	normal operation
above soft limit but within allocated quota	send warning emails and display warning message
above allocated quota within grace period	send warning emails and display warning message
above allocated quota beyond grace period	prohibit adding more files
above the hard limit	prohibit adding more files

Our hybrid approach allows distributed, flexible, and customizable quota management in HydroShare.

14

3.3.1. Encouraging data sharing via simplified quota policies

The quota scheme in HydroShare differs from other quota schemes prevalent for cloud data sharing. The *DropBox* policy of charging all users' quotas for an object they share is noticeably absent. In fact, the designers of HydroShare felt that this *DropBox* policy discourages data sharing by instituting a penalty for users who receive the data. Instead, in HydroShare, the user that creates a resource is considered the responsible party and becomes the initial owner and quota holder for a resource. Other users may be designated as resource owners and an owner may transfer the holding of quota to another owner of the resource, but at any time there is one responsible quota user for a resource. This is independent of how many times and with how many users the resource is shared. A resource shared with hundreds of users is still attributed to a single user for the purpose of quota management.

To understand the impact of this quota policy upon use of HydroShare, one must understand the basis of the sharing model in HydroShare. There are three levels of sharing: "view", "edit", and "owner". A "view" user can see and copy a resource; an "edit" user can make changes to data and/or metadata, and an "owner" can additionally make a resource public or publish it. The quota management system attributes the quota to one owner, even if there are multiple owners and hundreds of people have "edit" permission. This enables crowd-sourced data collection, instigated by an owner handing out hundreds of "edit" permissions, for which each editor is not penalized by an additional quota. Further, the owner can also specify that all editors can themselves grant edit permission to others, making the crowd potentially larger. Thus, large groups of people can be utilized in data collection and organization in previously unforeseen ways, *e.g.*, curating collections of data and collecting raw data from citizen science efforts.

3.3.2. Quotas and federation

The iRODS federation-based distributed data storage and management ecosystem facilitates and enables an extensible and flexible storage quota management

15

in that different quotas can be enforced for each partner's external federated storage. Although the quota policy in HydroShare includes working with users to accommodate their storage needs beyond their quota allocations, a partner institution may have terabytes of data that they want to put into HydroShare, beyond the system storage capacity provided by HydroShare's internal federated zones. Distributed quota management allows a partner institution to set up its own iRODS server to federate with HydroShare as part of external federated zones and manage quotas in this external zone. Figure 5 shows an example of administrative interface in HydroShare where user quotas can be configured by a system administrator for several federated zones.

	User	Allocated value	Used value	Unit	Zone
☐	test_user	20	0	GB	hydroshare_internal
☐	test_user	40	0	GB	hydroshare_external

Figure 5: An example of administrative interface in HydroShare for configuring user quota for federated zones

3.4. On-demand resource bag creation and update

Another benefit enabled by the iRODS-based distributed data storage and management is the performance gain obtained by leveraging iRODS commands and rules that operate directly on data where it resides. This data-centric approach takes operations to data to minimize data movement which is especially important for big data. As shown in Figure 2, a HydroShare resource collection conforms to the Bagit format, which includes Bagit metadata and manifest for each content file, with associated checksums for verification of resource content. Each HydroShare resource collection is zipped into a resource bag for data archive and easy transmission. In the early stages of HydroShare development, whenever the resource was created and updated, this zipped resource bag was created on the Django server, which was subsequently transferred to the iRODS back end storage. This introduced too much performance overhead due

16

to file zipping operations and transfer, and was not sustainable especially for big data. To eliminate unnecessary performance overhead, the Bagit workflow was changed to be invoked on-demand; updated bags are created only when the user requests to download the bag. To eliminate file transfer overhead, we delegate the resource Bagit operation to an iRODS Bagit rule that we customized for HydroShare and that directly operates on the resource collection wherever it is stored in iRODS.

In the following, we summarize our on-demand resource bagging workflow.

- A user creates a HydroShare resource A which in turn creates a resource collection directory hierarchy as shown in Figure 2 in a HydroShare iRODS zone. This includes all data files but excludes metadata files generated from Django metadata. Initially two iRODS flags, *bag_modified* and *resource_modified*, are created on the resource collection and set to *True*.

- A user updates the metadata for resource A. The metadata is updated in the database on the Django server, but the resource metadata file is not created or updated in iRODS. Instead, the *resource_modified* flag is set to *True* for on-demand file creation or update later.

- A user adds files to or deletes files from resource A. The *content* subdirectory is updated in iRODS to reflect the file updates, but the metadata files are not created or updated in iRODS. Instead the *resource_modified* flag is set to *True* for on-demand metadata file creation or update later.

- The resource metadata files are requested programmatically via a REST API method. If the *resource_modified* flag is *True*, the requested metadata files are created or updated on the Django server and transferred to iRODS. Then the files are transferred to the API requestor and this flag is set to *False*. Subsequent requests for the files check this flag; a *False* value does not trigger file update.

- A user requests to download the bag of resource A. If the *bag_modified* flag is *True*, the following actions take place in sequence:

17

- Resource metadata files are created on the Django server and transferred to iRODS. Then *resource_modified* is set to *False*.

- A custom iRODS Bagit rule is executed to generate Bagit metadata and manifest files with checksums conforming to Bagit specification in the resource collection.

- An iRODS command is executed to zip the whole resource collection into a zipped bag.

- The *bag_modified* flag is set to *False*.

- The zipped bag is transferred from iRODS to the HydroShare web client.

• Another user requests to download the bag of resource *A*. Now that the *bag_modified* flag is *False*, the zipped bag is directly transferred from iRODS to the HydroShare web client.

This on-demand resource bag creation and update workflow reduces computational overhead by ensuring the files are only created and transferred when needed. Leveraging iRODS rules and commands for resource bag creation to take operation directly to the data in iRODS gives us another layer of performance boost—especially for large resources—by eliminating the need for large file transfer in order to create each bag.

4. Discussion and future work

iRODS-based distributed data management not only facilitates data exchange and interaction between HydroShare and other iRODS-based systems such as CyVerse Discovery Environment, but also enables HydroShare to make data readily available to other systems for data analysis and visualization services. This section briefly discusses two use cases in HydroShare that illustrate possibilities enabled by the distributed data management approach. Other

18

points relevant to bringing partners into HydroShare ecosystem, potentially better leveraging iRODS ecosystem, and other ongoing and future work are also discussed.

4.1. National Water Model (NWM) data analysis in HydroShare

The National Oceanic and Atmospheric Administration (NOAA) National Water Center (NWC) launched operational production of its new high resolution National Water Model (NWM) in 2016 (NOAA (2016)). The NWM is a hydrological model that simulates observed and forecast streamflow over the entire continental United States. The model runs on supercomputers producing large volumes of data with hourly simulation of current conditions as well as medium-range and 30 day forecasts. To help make the NWM forecasts more accessible to scientists and the public, a rolling window of NWM data ranging from forty days to two years is registered and ingested into an iRODS zone. In turn, this data is made available to HydroShare apps such as NWM Forecast Explorer and Viewer, which allows browsing, exploration, and visualization of forecast data output from NWM. In the future, using distributed data management in HydroShare, the NWM data storage iRODS zone may be federated with the main HydroShare iRODS zone to bring NWM data into the HydroShare federated data ecosystem, which could enable easy creation of HydroShare resources using NWM data with structured metadata attached, to provide another platform for disseminating structured NWM forecast data in HydroShare.

4.2. Web apps for processing, analyzing, and visualizing big data in HydroShare

As stated in Section 2, HydroShare web apps are independent components that interact with HydroShare through the HydroShare REST API. On the one hand, this loose coupling between HydroShare and web apps enables extensibility in that anyone may develop a web app which interacts with resources stored in HydroShare. On the other hand, this loose coupling creates challenges for sharing data between two independent systems, especially for big data. Big data transfer via a native REST API is slow and problematic with transport file

size limited by the REST timeout. As described in Section 3.1, in HydroShare we impose a 1GB file size limit for direct file transfer via HTTP or REST and leverage iRODS parallel file transfer capability for transmission of files larger than 1GB.

We are currently investigating how to best leverage the iRODS federation and parallel file transfer capability in conjunction with the REST API to enable bidirectional file transfer between HydroShare and web apps. One potential approach under discussion is to use the HydroShare iRODS user zone as the temporary intermediate landing area for big files to be transferred. Other potential approaches such as installing iRODS server on the web apps to federate with the main HydroShare iRODS zone are also being evaluated. These potential approaches are aimed at leveraging the distributed iRODS federation capability to overcome the data transfer challenges inherent to any loosely coupled systems, and thus provide another use case that would be facilitated by our iRODS federation-based distributed data management ecosystem.

4.3. Bringing partners into HydroShare ecosystem

Although the iRODS federation-based distributed data grid in HydroShare enables resource storage at partner institutions, it imposes some commitments and technical requirements on them. First, a partnership with HydroShare involves commitments from both HydroShare and a partner. The partner must agree to provide persistent storage that is sufficiently protected from disastrous data loss, while HydroShare agrees to store the data for resources in the partner's storage. Since HydroShare is in essence guaranteeing that the DOIs that it issues when publishing data point to permanent resources, the partner must support that guarantee. The partnerships are not extended to individuals, and an institutional commitment (from, *e.g.*, a university or corporation) is required in order to form a partnership.

In addition, partnering imposes some technical requirements on partner institutions in that they need to install an iRODS server and federate it with the main HydroShare iRODS zone. However, the extra work required is justified by

20

the extended storage capacity at their disposal. HydroShare was built for managing hydrologic data and models at scale, and thus extensibility and scalability of the system is of paramount importance to the success of the system. In the near future, we plan to work with several partner institutions such as CyberGIS at NCSA (http://cybergis.illinois.edu) and the University of Alabama near the National Water Center (http://water.noaa.gov) to bring them into our external federated zones as part of HydroShare ecosystem.

4.4. Leveraging the iRODS ecosystem

HydroShare directly utilizes only limited capabilities of iRODS. Much more advanced iRODS capabilities are available outside HydroShare, including complex workflow management. Using iRODS independent of HydroShare, one can predefine data replication, processing, and copying steps that create future data products for HydroShare, and can even publish the results in HydroShare via iRODS workflows, using the HydroShare REST API. iRODS workflows can also involve data movement to and from supercomputing resources which may make it easier for researchers less well versed in supercomputing and cloud computing to take advantage of these resources.

iRODS federation enables several useful features, including fast and secure file transfer between federated zones and the ability to automatically process files using iRODS rules. iRODS is utilized in many supercomputing centers and organizations providing cloud computing, enabling iRODS users to transfer data to and from available compute resources quickly and easily. Thus, while the HydroShare Django website provides a front end for data preparation, iRODS as a component of HydroShare, provides a back end for moving data efficiently to compute resources in a supercomputer for processing such as running hydrologic models using data stored in HydroShare.

4.5. Other ongoing and future work

We are also working to make this distributed data management infrastructure in HydroShare more flexible. For example, we plan to allow users to customize the location where they want their resources to be stored. Currently,

21

HydroShare stores the resources in the main HydroShare zone or a federated zone depending on where the original resource content files came from. For more flexible quota management, it is desirable to give users an option to move their existing resources to a different federated zone within the HydroShare distributed data storage and management ecosystem.

Since HydroShare was built as a modular system based on loose coupling of dispersed components, and the hydrology-specific functionalities were encapsulated into software components as Django apps, we are working on generalizing HydroShare to make the software infrastructure applicable to other science domains beyond just hydrological and environmental sciences. Additionally, the loosely coupled iRODS-based network file system for scalable and extensible big data management approaches presented in this paper provide solid foundation and extensibility that enable and facilitate this generalization.

5. Conclusion

The iRODS-based approaches used for big data management in HydroShare provide a practical solution for storing and managing big hydrological data and models. Some examples in HydroShare that may leverage the big data management approaches have been presented to provide specific reference use cases enabled by our distributed data management approaches.

The highlights of the design of the iRODS-based distributed data management system in HydroShare are summarized below.

- Uses iRODS as a back end in HydroShare to facilitate replication to the off site data store for disaster recovery.

- Employs iRODS federation capability to enable a partner institution to add storage capacity into the distributed federated data storage ecosystem, extending HydroShare storage capacity.

- Provides iRODS user space to enable users to upload large files using iRODS clients and to add them into HydroShare resources.

22

- Uses iRODS rules and commands for on-demand resource bagging to take data operation close to data for enhanced performance.

- Enables more rapid resource file access by web apps via iRODS clients.

- Enables interoperability on the data level with other iRODS-based systems.

- Provides access to community data such as NWM data that is registered and ingested in a separate iRODS zone that may be federated with HydroShare data store.

- Provides fast transfer of big data to and from supercomputers and cloud computing providers.

As of this writing, the functionality reported here is deployed as part of HydroShare (https://www.hydroshare.org) with source code open (BSD 3-clause license) and available on github (https://github.com/hydroshare/hydroshare/). Active development continues to improve quota management in HydroShare and interconnectivity with HydroShare apps as well as many other enhancements.

Acknowledgments

References

Boyko A, Kunze J, Littman J, Madden L, Vargas B. The bagit file packaging format (v0.97), network working group internet draft. 2012. URL: `http://tools.ietf.org/html/draft-kunze-bagit-10`.

Cao Y, Jones C, Cuevas-Vicenttin V, Jones M, Ludäscher B, McPhillips T, et al. . Dataone: A data federation with provenance support. In: Proceedings of 6th International Provenance and Annotation Workshop. McLean, VA, USA; 2016. p. 230–4. doi:http://dx.doi.org/10.1007/978-3-319-40593-3_28.

Couch A, Tarboton D, Idaszak R, Horsburgh J, Yi H, Stealey M. A flexible file sharing mechanism for irods. In: Proc. 2015 iRODS User Group Meeting. Chapel Hill, NC; 2015. p. 61–8.

Crawley S, Ames D, Li Z, Tarboton D. Hydroshare gis: Visualizing spatial data in the cloud. Open Water Journal 2017;4(1):3–20. URL: http://scholarsarchive.byu.edu/openwater/vol4/iss1/2. doi:10.1111/1752-1688.12363.

DCMI . DCMI metadata terms, dublin core metadata initiative (DCMI). 2012. URL: http://dublincore.org/documents/dcmi-terms/.

Hane P. Sharing research data—new figshare for institutions. 2013. URL: http://www.against-the-grain.com/2013/09/sharing-research-data-new-figshare-for-institutions-2/.

Hünich D, Müller-Pfefferkorn R. Managing large datasets with irods—a performance analysis. In: Proceedings of the International Multiconference on Computer Science and Information Technology. Wisla, Poland; 2010. p. 647–54.

Heard J, Tarboton D, Idaszak R, Horsburgh J, Ames D, Bedig A, Castronova A, Couch A. An architectural overview of hydroshare, a next-generation hydrologic information system. In: Proceedings of the 11th International Conference on Hydroinformatics. New York City, USA; 2014. URL: http://academicworks.cuny.edu/cc_conf_hic/311/.

Hedges M, Blanke T, Hasan A. Rule-based curation and preservation of data: A data grid approach using irods. Future Generation Computer Systems 2009;25(4):446–52. doi:https://doi.org/10.1016/j.future.2008.10.003.

24

Horsburgh J, Morsy M, Castronova A, Goodall J, Gan T, Yi H, Stealey M, Tarboton D. Hydroshare: Sharing diverse environmental data types and models as social objects with application to the hydrology domain. JAWRA (Journal of the American Water Resources Association) 2015;52(4):873–89. doi:`10.1111/1752-1688.12363`.

Idaszak R, Tarboton DG, Yi H, Christopherson L, Stealey MJ, Miles B, Dash P, Couch A, Spealman C, Ames DP, Horsburgh JS. Hydroshare - a case study of the application of modern software engineering to a large distributed federally-funded scientific software development project. In: Carver J, Hong NPC, Thiruvathukal GK, editors. Software Engineering for Science. Taylor & Francis CRC Press; 2016. p. 219–36.

Lagoze C, de Sompel HV, Johnston P, Nelson M, Sanderson R, Warner S. Open archives initiative object reuse and exchange: Ore user guide primer. 2008. URL: `http://www.openarchives.org/ore/1.0/primer`.

Merchant N, Lyons E, Goff S, Vaughn M, Ware D, Micklos D, Antin P. The iplant collaborative: Cyberinfrastructure for enabling data to discovery for the life sciences. PLOS Biology 2016;14(1):1–9. doi:`10.1371/journal.pbio.1002342`.

Moore R. Toward a theory of digital preservation. International Journal of Digital Curation 2008;3(1):63–75.

Myers J, Hedstrom M, Akmon D, Payette S, Plale B, Kouper I, et al. . Towards sustainable curation and preservation: the sead project's data services approach. In: Proceedings of 2015 IEEE 11th International Conference on e-Science. Munich, Germany; 2015. p. 1–10. doi:`http://dx.doi.org/10.1109/eScience.2015.56`.

NOAA . National water model: Improving noaa's water prediction services. 2016. URL: `http://water.noaa.gov/documents/wrn-national-water-model.pdf`.

25

Oliver S, Lenards AJ, Barthelson R, Merchant N, McKay S. Using the iplant collaborative discovery environment. In: Current Protocols in Bioinformatics. John Wiley & Sons, Inc.; 2013. p. 1.22.1–1.22.26.

Russell T, Coposky J, Garde C. irods technical overview. In: Proc. 2016 iRODS User Group Meeting. Chapel Hill, NC; 2016. .

Schnase J, Duffy D, Tamkin G, Nadeau D, Thompson J, Grieg C, McInerney M, Webster W. The NASA center for climate simulation data management system - toward an irods-based approach to scientific data services. In: 2011 IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST). Denver, CO, USA; 2011. p. 1–6. doi:10.1109/MSST.2011.5937235.

Sukhija N, Datta A. C-grid: Enabling irods-based grid technology for community health research. In: Proceedings of 4th International Conference on Information Technology in Bio- and Medical Informatics (ITBAM). Prague, Czech Republic; 2013. p. 17–31. doi:10.1007/978-3-642-40093-3_2.

Tarboton D, Idaszak R, Horsburgh J, Heard J, Ames D, Goodall J, Band L, Merwade V. A resource centric approach for advancing collaboration through hydrologic data and model sharing. In: Proceedings of the 11th International Conference on Hydroinformatics. New York City, USA; 2014. URL: http://academicworks.cuny.edu/cc_conf_hic/314/.

Vitolo C, Elkhatib Y, Reusser D, Macleod C, Buytaert W. Web technologies for environmental big data. Environmental Modelling & Software 2015;63:185–98. doi:https://doi.org/10.1016/j.envsoft.2014.10.007.