Jun 16th, 10:40 AM - 12:00 PM

Reproducible Research within the DataNet Federation Consortium

Reagan W. Moore
University of North Carolina at Chapel Hill, rwmoore@renci.org

Arcot Rajasekar
University of North Carolina at Chapel Hill, sekar@renci.org

International Environmental Modelling and Software Society (iEMSs)
7th Intl. Congress on Env. Modelling and Software, San Diego, CA, USA,
Daniel P. Ames, Nigel W.T. Quinn, and Andrea E. Rizzoli (Eds.)
http://www.iemss.org/society/index.php/iemss-2014-proceedings

Reproducible Research within the DataNet Federation Consortium

Reagan W. Moore, Arcot Rajasekar

University of North Carolina at Chapel Hill, rwmoore@renci.org, sekar@renci.org

Abstract: The DataNet Federation Consortium (DFC) is an NSF funded project that provides cyberinfrastructure for federating data management systems into a collaboration environment. Researchers are able to build a shared collection, apply analysis workflows, and manage the analysis results. The shared collections may span storage resources at multiple institutions, and multiple types of data management systems. The analysis workflows may include staging of files to remote compute platforms, and in-place analysis at the remote data storage location. The workflows can be captured as shareable objects, with automatic capture of input files, input parameters, and output files. A key feature is support for interoperability mechanisms for accessing data from remote repositories, using an appropriate external protocol. It is possible to automate acquisition of data from external resources, transform the data sets into required formats, and save results within the collaboration environment. The interoperability mechanisms can be controlled by policies that automate data acquisition steps, automate validation of collection properties such as integrity, and automate execution of administrative tasks such as data migration. Researchers can share a workflow, modify input parameters, re-execute the workflow, and share the output results. This enables reproducible data-driven research. The DFC builds upon the integrated Rule Oriented Data System, which is open source software middleware available from http://irods.org. The DFC web site is http://datafed.org.

***Keywords**:* Data grids, reproducible research, federation.

1 REPRODUCIBLE RESEARCH

A principal goal of data-driven research is the ability to reproduce analyses, enabling a second researcher to derive the same result. A standard approach has been the creation of workflows that encapsulate the knowledge that is needed to do the analysis. Billah et al. (2012) describe workflows used in hydrologic modelling using the Variable infiltration Capacity Macroscale Hydrologic Model, VIC (2012). Miles and Band (2012) describe the use of workflows to automate execution of the Regional Hydro-Ecologic Simulation System, RHESSys (2000). Guru et al. (2009) describes challenges in using scientific workflow tools in the hydrology domain. In each case, if the workflows can be shared, then a second researcher should be able to use the workflow to re-generate the original research result.

Reproducible data-driven research has been selected as a goal of the DataNet Federation Consortium (DFC), an NSF funded project that promotes the creation of national scale collaboration environments through federation of existing data management systems. Researchers who are collaborating on a project should be able to share not only derived data products, but also the workflows that were used to create the products. The infrastructure that is needed to build a federated collaboration environment and support reproducible data-driven research is described in this paper.

1.1 Introduction

The DataNet Federation Consortium uses data grid technology to federate data management systems for six national initiatives in Hydrology (University of Virginia Department of Civil and

Environmental Engineering, University of North Carolina at Chapel Hill Institute for the Environment), Oceanography (Ocean Observatories Initiative), Engineering (CIBER-U: Cyber-Infrastructure-Based Engineering Repositories for Undergraduates Project), plant Biology (the iPlant Collaborative), Cognitive Science (Temporal Dynamics of Learning Center), and Social Science (Odum Institute and the Dataverse network). Each project relies on collaboration between researchers from multiple institutions to conduct the research. The data sets that are shared may be real-time sensor data (oceanography), or surveys (social science), or genomics data (plant biology), or CAD-CAM diagrams (engineering), or digital elevation maps (hydrology), or videos (cognitive science). The researchers may access data holdings in federal repositories (NASA, NOAA, USGS, EPA, Department of Agriculture, Department of Transportation), may apply processing algorithms to transform the data to suitable environment variables, may generate research data products that are published for use by the broader discipline, and may build a reference collection for use by future researchers. Thus the data management infrastructure needs to support and automate multiple types of activities:

- Data retrieval
- Data processing
- Data sharing
- Data indexing and annotation
- Data publication
- Data preservation

Policy-based data management systems enable the separation of the control semantics from the set of user interfaces and the data manipulation operations (Moore and Rajasekar, 2007). Effectively, a researcher should be able to select a preferred user interface, define a set of workflows that will be executed on data sets, and define a set of control policies for managing the processing pipeline. The control of the environment is implemented as computer actionable policies that are enforced for each type of action invoked by a client access mechanism. To maintain consistency, the environment must also manage state variables that track the outcome of each operation. These capabilities can be used to automate administrative tasks, such as data staging, data migration, and data archiving. The same approach can also be used to automate validation of assessment criteria to verify that the system is working correctly. The management of data and metadata then reduces to the enforcement of management procedures that are implemented as computer executable workflows.

1.2 iRODS Data Grid

The integrated Rule Oriented Data Systems (iRODS) was developed by Rajasekar et al. (2006) and implements policy-based data management. The iRODS software is open source middleware (iRODS 2006), that sits between the user clients and the remote resource where either computation or data storage is done. The middleware manages interactions with each repository, manages name spaces for file names, and enforces the management policies. Figure 1 describes the peer-to-peer client server architecture. The software middleware is installed at each site where data may be stored. An iRES server manages interactions with the storage systems at the site, by converting actions requested by a client to the protocol used by the storage system. The iRES server also includes a rule engine which reads a local rule base to select policies for enforcement. Operations performed by the iRES server, such as generation of a checksum, or replication of a file, or setting of an access control, will generate state information that is registered in a central metadata catalog. The iCAT server maps from the catalog operation request to the protocol required by a relational database, and updates the state information after each operation. Descriptive and provenance metadata are also stored in the iCAT relational database using schema indirection. For each metadata item, the attribute name, attribute value, and an attribute comment are stored. Attributes can be added to each of the name spaces managed by the data grid, including user-defined attributes about users, collections, files, and storage systems. Users can assign a logical name to a file, and organize the files in logical collections, independently of the actual storage location. The data grid “owns” the registered data in that the files are stored under a data grid account that is established at each storage location. Users authenticate to the data grid, which in turn authenticates itself to the remote storage location. The data grid manages access controls, authenticating every access and authorizing every operation.

A peer-to-peer architecture connects the servers, since there may be hundreds of iRES servers. A client can connect to any server, and operation requests are moved between iRES servers as

needed. Thus a user can connect to an iRES server and request a file by its logical path name. The iRODS data grid forwards the request to the iCAT server to resolve the physical location of the file, and then sends a request to the server where the file resides for delivery to the user. This is a form of third party data transfer (a request to the first server initiates traffic between a second and third server). The peer-to-peer architecture makes it possible for a user to find and retrieve files without knowing the file's physical location, the physical path name of the file, or the protocol required by the storage system, and without having a personal account at the remote storage location.

Two additional servers are used to manage delayed or periodic rule execution and message exchange over a message bus. The iSEC server maintains a queue of outstanding rules that will be executed at a specified time. A typical example is to defer the execution of a replication request to a tape archive such as the IBM High Performance Storage System. The initial request to write returns immediately, with the data grid at a later time making the copy. This provides a way to turn nominally synchronous operations into asynchronous operations. The iXMS server is used to manage messages needed to support debugging of distributed operations and the tracking of transfer progress when moving large files. A user can control the execution of a distributed workflow, stepping through the individual operations, through messages sent between servers. A client can track the execution status of a long-running task, through messages that are periodically sent back from a server.

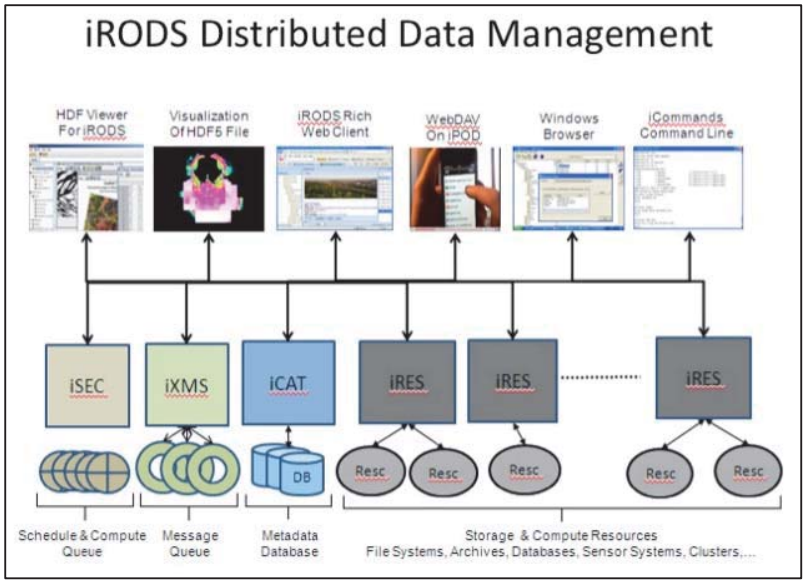

Figure 1. Data grid software middleware servers

Within the DataNet Federation Consortium, a separate data grid was established for each discipline. For the Hydrology collaboration, the data grid included resources at the University of Virginia and the University of North Carolina at Chapel Hill. Hydrologists using the data grid saw a shared name space for organizing files into shared collections. Thus files present on an iRES server in Virginia could be organized into the same logical collection (directory structure) with files stored on a server in Chapel Hill. The access controls on the shared files were maintained in the iCAT catalog along with the information needed to authenticate a user. On each access, the identity of the researcher was authenticated, and the desired operation was authorized independently of the actual storage location. Access controls were set for each file, enabling the creation of both private and public collections. Researchers could keep data private while the research was being done, and then choose to publish the data once the results were verified. Policies can be implemented that version files when they are changed, that enforce strict access controls such that a person can only see their own files, that manage time-dependent access controls, and that automate distribution of files across storage systems.

A second notable feature of the peer-to-peer architecture is that the choice of clients was not limited by the choice of storage systems. The software/middleware responds to action requests from web

browsers, Unix command line tools, I/O libraries such as C and Fortran, web services, workflow systems such as Kepler and Taverna, scripting languages such as Python and Java, and digital libraries such as the Fedora Commons digital repository software. The middleware translates from the protocol used by the client to the protocol required by the storage system. In this sense, a data grid is a broker that enables researchers to federate heterogeneous storage environments into a single logical resource. In actual practice, a data grid provides the interoperability mechanisms needed to interact with an external system's protocol, and the control mechanisms needed to enforce management policies across the distributed environment, as shown in Rajasekar et al. (2010). The data grid also acts as a virtualization interface not only hiding the idiosyncrasies of the back-end repositories through uniform interfaces, but also mediating authentication and authorization by providing a virtual user space and a single sign-on facility.

A third notable feature is that policy enforcement is done at the level of operations performed by the middleware software. Effectively, policy enforcement is decoupled from the choice of client or storage system. This makes the system highly extensible. New storage systems can be added and new clients can be supported without impacting the management policies. The types of policies that are typically implemented include:

- Selection of preferred storage location for files in a collection
- Selection of desired physical path name for a file
- Setting of time dependent access controls
- Setting of a retention period
- Parsing of descriptive metadata from the file contents
- Generation of a checksum or digital signature
- Aggregation of files into bundles such as a tar file
- Replication
- Creation of derived data products (thumbnails for images)
- Logging of all update operations in an audit trail
- Periodic verification of integrity
- Periodic generation of status reports

The choice of policy is based on a consensus across the collaborating researchers. Each researcher can choose to keep data private, and specify their own management policies for their own collections. At the same time, collaborating researchers can jointly specify the policies that will be applied for their shared collections.

2 AUTOMATING RESEARCH WORKFLOWS

Policies within iRODS control the execution of procedures that are composed by chaining together data management operations (called micro-services) into a workflow. The micro-services include basic functions for data manipulation, data access, environment property setting, invocation of remote protocols, etc. The workflow is executed under the control of the distributed rule engine. Thus the workflow could have operations that are applied at multiple storage locations, or could invoke micro-services that access external web services. The workflow can include the staging of data to a remote server, the invocation of an external application at the remote server, and the registration of output data sets back into the data grid. Using these mechanisms, the distributed rule engine can implement a data processing pipeline.

As shown by Miles and Band (2012) researchers can use the data grid to automate analyses:

- Automate data retrieval from a federal repository. A micro-service is written that encapsulates the knowledge needed to interact with the protocol of the repository, request a data set, and retrieve the data.
- Automate transformation operations on a data set. A micro-service is written that applies the appropriate data sub-setting libraries. Examples include execution of netCDF and HDF5 library calls to extract data subsets.
- Automate execution of a research application. A micro-service is available (remoteExec) for supplying input parameters to an application at a remote site, tracking the execution status, and retrieving standard out and standard error messages.

A workflow can be created that chains together the individual micro-services. The iRODS data grid can define a policy that controls when and where the workflow is executed. Thus the data management system could detect the addition of new data files, select the processing steps to apply, execute the workflow, and save the results. This has the potential to change the way descriptive metadata is generated for collections. Usually, descriptive metadata represents the context for the contents of a file. In a policy-based system, descriptive metadata can be created that represents features within the data set. The processing steps needed to identify the features are captured in micro-services, and policies are defined for the automated execution of the feature detection algorithm each time a file is added to the system. Each found feature is registered as searchable descriptive metadata for the file. A researcher can then define the set of features they desire for selecting a relevant data set, and have the system automatically index data sets by features.

2.1 Sharing Workflows

Reproducible data-driven research becomes possible when the workflows that automate an analysis can be shared, along with the input and output files associated with each execution of the workflow. In the iRODS data grid, the concept of workflow-structured-objects was implemented. A workflow can be written using a workflow language and saved as a text file. The text file can then be registered into the data grid. When the text file that contains the workflow language is accessed, the workflow is executed. Since the workflow is mapped to the same logical name space used for files, the same access controls are enforced. For instance, the workflow can be shared or can be held private.

To automate the tracking of workflow executions, additional constructs represented in Figure 2 are needed.

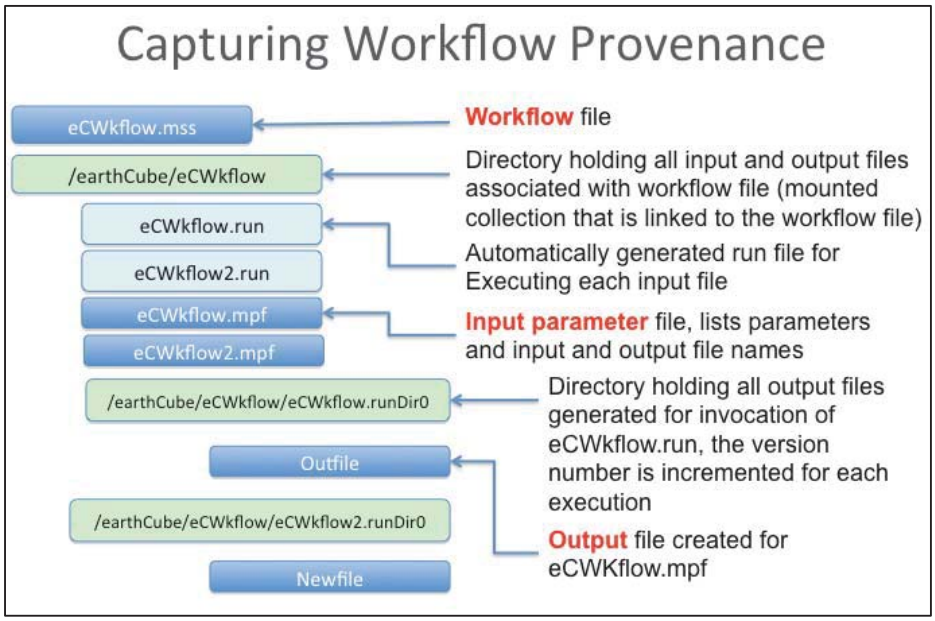

Figure 2. Managing workflow input and output files on each workflow execution

A mechanism was required to differentiate between retrieval of the text file that represented the workflow, and the invocation of the workflow execution. This was done by representing each workflow as a ".mss" file. The ".mss" file contains the workflow that is actually executed under the control of the iRODS rule engine, and must be written in the iRODS rule language.

To track provenance of workflow execution, a command was added to enable association of a workflow collection with the ".mss" file. The workflow collection is then used by the data grid to automatically associate output files created by running the workflow with the corresponding input files. When an input parameter file (".mpf" file) for the workflow is put into the workflow collection, a ".run"

file is automatically created. Accessing the ".run" file causes the execution of the workflow defined in the linked ".mss" file. For each invocation of the workflow, an output sub-directory is created. Directives in the ".mpf" file specify which input, intermediate, and output files will be automatically stored in the output sub-directory. The name of the output sub-directory is created by replacing the extension of the workflow input file name with ".rundir0".

Each time a new input parameter file is put into the workflow directory, a new ".run" file is created to control execution. Similarly, each time a ".run" file is accessed, a new output sub-directory is created with the name defined by incrementing the number in the extension. Thus the second run of a workflow will have the output saved in a directory with the extension ".rundir1". Since the workflow file, ".mss", the input file, ".mpf", and the input and output files can be shared, it is possible for a researcher to enable reproducible research:

- A second researcher can re-execute exactly the same workflow and verify they get the same result (reproduce).
- A workflow can be executed with new input parameters, and the original and new result sets can be compared (reuse).
- The workflow can be modified, and the output between the different versions of the workflow can be compared (recompose).

3 FUTURE WORK AND CONCLUSION

The management of reproducible data-driven research is an example of the use of policy-based systems to automate interactions with data. In future planned work, policy-based systems are also being applied to manage interactions with Software Defined Networks and to manage interactions with storage systems (Data Direct Networks Web Object Scalar storage). Software Defined Networks implement policies within an OpenFlow router that controls the routing of network packets. The policies can be updated dynamically, changing how the network responds to a data transmission request. Through the iRODS data grid, it is possible to interact with an OpenFlow router to manage the selection of disjoint network paths for parallel I/O streams when moving very large files. The iRODS data grid manages the information about the location of the files, and the OpenFlow router manages the data transfer. By using policies to control end-to-end data movement, the time to transfer multi-terabyte files can be minimized.

Similarly, when policies are integrated with storage systems, standard administrative tasks such as data distribution and data replication can be automated. Policies can be implemented that move data to the site where the data will be primarily accessed. Replicas can be created and stored at geographically distinct sites. The integrity of replicas can be verified by evaluating a checksum and comparing the current checksum value with a stored value. The required number of replicas can be verified and missing replicas can be added as needed. The policies in the storage system manage the properties of the storage environment, while the policies in the data grid manage the properties of the data collection.

Reproducible data–driven research will require additional development to define the execution context associated with each workflow. Within the iRODS data grid, the micro-services constitute well-defined operations that have a specific context in terms of the state information they modify. The micro-services are operating system agnostic, in that the I/O calls they execute are mapped to the protocol of each storage environment by the iRODS middleware. Thus a processing pipeline written in the iRODS rule language is operating system independent, and will be executable in the future on technologies to which the iRODS middleware has been ported.

However, most processing pipelines also invoke application codes that require specific choices for operating system and computer hardware. An upgrade to the iRODS data grid that is being developed is the concept of a compute resource. The storage hierarchy is being extended to support naming and management of computation nodes. An executable can be assigned to a computation node, and a workflow can be distributed across both computation and storage nodes. This will give tight control over the execution of workflows.

A more loosely coupled integration between external workflows and data management is already supported through the integration of the NCSA Cyberintegrator with the iRODS data grid. Micro-

services have been written that manage the submission of a workflow to the NCSA Cyberintegrator, the monitoring of the progress of the workflow, and the retrieval of the workflow output products back to the data grid. The NCSA Cyberintegrator manages the execution of the workflow on an HPC platform, while the iRODS data grid manages the data products that are created. This style of integration of processing pipelines with collection-based data management is expected to become the basis for data-driven research.

Another area of research is the development of linked-data vocabularies to describe the operations performed in the data processing pipeline. This is needed to characterize operations performed within both the iRODS data grid and external workflow systems such as Kepler and Taverna. The long term goal is to be able to express the operations that are needed by a workflow, and migrate a processing pipeline to a new workflow environment by requesting the necessary operations. This is also needed for true interoperability between workflow systems. The iRODS data grid provides a strong basis for characterizing operations performed by data management tasks, and can thus serve as a starting point for defining basic data management operations. In the long term, reproducible data-driven research requires the ability to describe the operations that were performed using a standard vocabulary, and the ability to re-execute an analysis based upon that description.

ACKNOWLEDGMENTS

The research reported in this paper was funded through NSF grant OCI-0940841, the DataNet Federation Consortium.

REFERENCES

Billah, M.M., Goodall, J. L., Rajasekar, A., Moore, R. W. 2012. Application of the integrated Rule Oriented Data System (iRODS) to support regional-scale hydrologic modeling. Proceedings of American Geophysical Union Conference.

Guru, S. M, Kearney, M., Fitch, P., Peters, C., 2009. Challenges in using scientific workflow tools in the hydrology domain, 18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009.

iRODS, 2006 integrated Rule Oriented Data System, iRODS. , https://www.irods.org (last accessed 13.03.14.).

Miles, B., Band, L., 2012. Toward a geo-informatics framework for understanding the social and biophysical influences on urban nutrient pollution due to residential impervious service connectivity. AGU Fall Meeting.

Moore R, Rajasekar, A., 2007. Rule-Based Distributed Data Management, Grid 2007: IEEE/ACM International Conference on Grid Computing.

Rajasekar A, Wan, M., Moore, R., Schroeder, W., 2006. A Prototype Rule-based Distributed Data Management System. HPDC workshop on Next Generation Distributed Data Management, Paris, France, 2006.

Rajasekar, A., Wan, M., Moore, R., Schroeder, W., Chen, S.-Y., Gilbert, L., Hou, C.-Y., Lee, C., Marciano, R., Tooby, P., de Torcy, A., Zhu, B., 2010. iRODS Primer: Integrated Rule-Oriented Data System. Morgan & Claypool.

RHESSys, 2000. Regional Hydro-Ecologic Simulation System, RHESSys. http://fiesta.bren.ucsb.edu/~rhessys/index.html (last accessed 13.03.14).

VIC, 2012. Variable Infiltration Capacity Macroscale Hydrologic Model, VIC. http://www.hydro.washington.edu/Lettenmaier/Models/VIC/ (last accessed 13.03.14).