



**US Army Corps
of Engineers®**
Engineer Research and
Development Center



Engineered Resilient Systems

Analysis of ERS Use Cases for iRODS

Kevin D. Winters, Mark A. Cowan, Glover E. George,
Megan E. Gonzalez, Brian Priest, Omar Morris,
and Jonathan Landrum

August 2020



The U.S. Army Engineer Research and Development Center (ERDC) solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at www.erdcd.usace.army.mil.

To search for other technical reports published by ERDC, visit the ERDC online library at <http://acwc.sdp.sirsi.net/client/default>.

Analysis of ERS Use Cases for iRODs

Kevin D. Winters, Mark A. Cowan, Glover E. George, and Megan E. Gonzalez

*U.S. Army Engineer Research and Development Center (ERDC)
Information Technology Laboratory (ITL)
3909 Halls Ferry Road
Vicksburg, MS 39180-6199*

Brian Priest, Omar Morris, and Jonathan Landrum

*Geocent Jackson Mississippi Office
1000 Highland Colony Parkway, Suite 5203
Ridgeland, MS 39157*

Final Report

Approved for public release; distribution is unlimited.

Prepared for Headquarters, U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Work unit 92L5D8

Abstract

The purpose of this paper is to discuss the challenges inherent with High Performance Computing (HPC) data storage access and management, the capabilities of iRODS, and the analysis of several Engineered Resilient Systems (ERS) use cases relating iRODS capabilities to the teams' stated needs. Specifically, these teams are the ERS Data Analytics group (specifically their research on rotorcraft maintenance in conjunction with the U. S. Army Aviation and Missile Research Development and Engineering Center [AMRDEC]), the ERS Environmental Simulation research team, the ERS Sensor Systems research team, and the HPC/Scientific computing group representing the "General HPC User."

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

Contents

Abstract	ii
Preface.....	v
1 Introduction	1
1.1 Background	1
1.2 Purpose	2
1.3 Objectives.....	2
1.4 Approach.....	3
2 HPC Data and Management Challenges.....	4
2.1 Data challenges	4
2.2 Process challenges	5
2.3 Management challenges	5
3 iRODS Core Competencies and Capabilities	6
4 ERS iRods Use Cases	8
4.1 Data analytics ERS rotorcraft maintenance project	8
4.2 Data analytics specific challenges.....	9
4.3 iRODS related solutions.....	9
4.4 Data analytics iRODS use case.....	11
4.4.1 Use case: Ingest new data.....	11
4.4.2 Use case: Perform data analysis.....	12
4.4.3 Use case: Review audit data.....	13
4.5 Environmental simulation	14
4.6 ES specific challenges.....	14
4.7 iRODS related solutions.....	15
4.8 ES iRODS use case.....	17
4.9 Sensor systems	19
4.10 Sensor systems specific challenges.....	19
4.11 iRODS related solutions.....	20
4.12 Sensor systems iRODS use case.....	20
4.13 General HPC user.....	22
4.13.1 General HPC user challenges.....	22
4.13.2 iRODS related solutions.....	23
4.14 General HPC iRODS use case	23
5 Conclusion.....	26
References	28
Acronyms and Abbreviations.....	29

Figures and Tables

Figures

Figure 1. Organizational data challenges (Sivarajah et al. 2017).	4
Figure 2. Data analytics directory structure.....	10
Figure 3. Data analytics iRODS use case Unified Modeling Language (UML) diagram.	11
Figure 4. ES directory structure.	16
Figure 5. ES iRODS use case UML diagram.	17
Figure 6. Sensor systems iRODS use case UML diagram.....	21

Tables

Table 1. iRODS core competencies.	6
Table 2. iRODS main capabilities.....	7

Preface

This is a deliverable product under the Engineered Resilient Systems (ERS) Program, Data Analytics Work Package, Collaborative Tradespace Analytics Work Unit 92L5D8. Dr. Owen J. Eslinger was the program manager, and Dr. Robert M. Wallace was the Technical Director of the ERS program.

The work was performed by the Computational Analysis Branch (CAB) of the Computational Science and Engineering Division (CSED), Engineer Research and Development Center (ERDC), Information Technology Laboratory (ITL), Vicksburg, MS. At the time of publication Dr. Jeffrey L. Hensley was Chief, CAB; Dr. Jerrell R. Ballard, Jr. was the Chief, CSED; Ms. Patti S. Duett was the Deputy Director of ITL and the Director was Dr. David A. Horner.

COL Teresa A. Schlosser was the Commander of ERDC, and Dr. David W. Pittman was the Director.

1 Introduction

1.1 Background

High performance computing (HPC) has become an indispensable tool to users across widely divergent sectors. Researchers, engineers, and scientists utilize HPC resources and networks to develop advanced products, perform research, and solve global-scale problems. This work demands mathematically intense computation and generates enormous amounts of data. Managing the complex simulations, models, software applications, sensor networks, and their ever-growing data outputs represents a complex challenge in today's environment of large and often distributed data sets, rapidly evolving technologies, increased reliance on outsourced resources, and stringent regulations regarding data access and use. These challenges are difficult for traditional file systems to handle. Modern data management applications that can address these critical and changing needs are a necessity. One such system is the Integrated Rule-Oriented Data System (iRODS). iRODS was developed in a supercomputing environment, thus, it scales to very large use and integrates seamlessly into HPC systems. It was specifically designed to meet the complicated and varied needs of data management in large scale computational environments spread over geologically dispersed areas.

iRODS is an enterprise-scale, open source data management suite of tools used by research organizations and government agencies worldwide in mission critical environments. iRODS logically organizes data and collections within storage resources, which are accessed by users via a unified namespace. iRODS allows data to be stored on disparate, heterogeneous storage devices while users access the data via a virtual interface, regardless of the physical storage location. iRODS allows for the creation of metadata, both user-generated and system-generated, that can aid in the discovery of relevant data and be used by the workflow engine to perform complex data operations, such as replication, access control, or any other condition defined by policy. The comprehensive metadata system in iRODS, in contrast to traditional file systems, allows for the creation of rich, descriptive, customizable metadata, which can be applied to individual files or to collections. This metadata then allows users to easily query iRODS to retrieve and utilize data in a simple and efficient manner.

iRODS is a proven technology that has been deployed in large-scale scientific research projects such as the National Atmospheric and Space Administration Center for Climate Simulation (Schnase et al. 2011); the Wellcome Trust Sanger Institute (Chiang et al. 2011; Clapham 2018); and the Texas Advanced Computing Center (TACC) at the University of Texas at Austin (Texas 2018), among many others. These projects are using iRODS for managing petabytes of data in hundreds of millions of files on distributed storage resources. The Wellcome Trust Sanger Institute's deployment is one of the world's largest, integrating an iRODS environment of 129 servers into a Lustre based HPC files system that houses more than 18 petabytes of data serving hundreds of users (Chiang et al. 2011; Clapham 2018).

1.2 Purpose

The purpose of this paper is to discuss the challenges inherent with HPC data storage access and management, the capabilities of iRODS, and the analysis of several Engineered Resilient Systems (ERS) use cases relating iRODS capabilities to the teams stated needs. Specifically, these teams are the ERS Data Analytics group (specifically their research on rotorcraft maintenance in conjunction with the U.S. Army Aviation and Missile Research Development and Engineering Center [AMRDEC]), the ERS Environmental Simulation research team, the ERS Sensor Systems research team, and the HPC/Scientific computing group representing the "General HPC User." These teams were chosen because of the established maturity of their research and their ability to be broadly representative of many other ERS groups while simultaneously highlighting unique differences and needs that exist among the different research groups of ERS.

1.3 Objectives

The use cases in this paper will demonstrate the strength and flexibility of iRODS in solving the multi-faceted needs that arise in diverse, large-scale HPC environments.

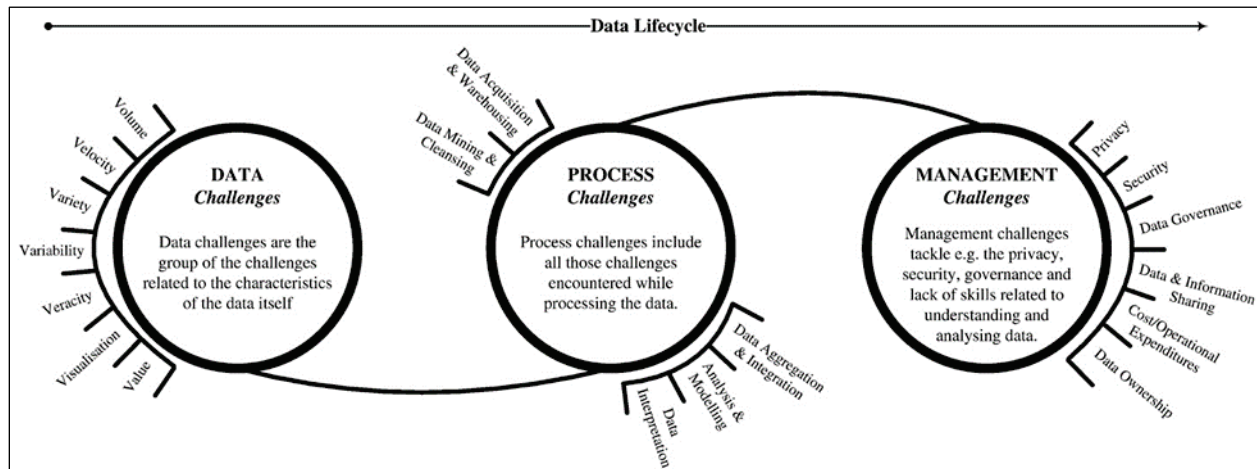
1.4 Approach

This document will discuss the challenges with HPC data storage access and management and will demonstrate how iRODS can address these challenges through analysis of several ERS use cases. It will also discuss how iRODS can address the challenges of HPC data storage access and management.

2 HPC Data and Management Challenges

The challenges facing public and private institutions that utilize HPC environments are numerous. Large-scale, computationally intensive applications are expected to generate exabytes of data over the next five to ten years (Berkeley Lab 2018). With data being generated and stored at unprecedented volumes and rates, the success of large-scale research and development efforts will require new hardware, software, and management strategies, especially for efforts that require information sharing and collaboration between parties that are geographically separated. In Figure 1, the challenges are grouped into the following three specific areas: data challenges, process challenges, and management challenges.

Figure 1. Organizational data challenges (Sivarajah et al. 2017).



2.1 Data challenges

Data challenges are specific to the data itself; HPC environments typically work with many different types of datasets with different data types, both structured and unstructured, requiring a variety of different storage devices. The variety of data and the sheer volume can present challenges to users and stakeholders when it comes to understanding the big picture of what is stored within the environment. A comprehensive metadata strategy is required to accurately describe the data so users and stakeholders alike can make more informed decisions about the vast amount of data stored within the environment.

2.2 Process challenges

Process challenges are challenges users face when working with the data. Often, HPC users are aggregating and integrating different datasets to perform higher level problem solving. Being able to locate and make use of the right data becomes more challenging as the volume and variety increases. Also, as data begins to age and is accessed less frequently, it is typically moved to lower priority storage (e.g., archive) and this movement of data can make it difficult for users to find what they are looking for.

2.3 Management challenges

Management challenges are comprised of the overarching concerns of the organization from a data management and access perspective, such as data governance, security, ownership, secure collaboration, and administrative visibility to name a few. Without data governance and established organizational data management policies that are enforceable within the system, these challenges quickly turn into problems that drive reactionary rather than proactive decision-making. It is imperative that organizational policies not only be developed and expressed to users, but also enforceable within the system.

3 iRODS Core Competencies and Capabilities

Organizations that engage in large-scale, cutting-edge research and development require a powerful, scalable, flexible, efficient, easy-to-use, and easy-to-maintain enterprise-level system for data storage and retrieval. The system should also automate workflow; allow the creation and utilization of customized, descriptive, and rich metadata along with storage and access policies; and include admin tools for the auditing and monitoring of system resources, users, groups, and data access. The application must work with stringent security mandates and allow scientists and engineers to intuitively use the system so it becomes a seamless tool that aids their work instead of an intrusive impediment that requires excess time to learn and integrate into their routine.

iRODS has a multitude of capabilities and strengths that can be tailored to meet the unique and specific needs of different institutions and companies. The following is a discussion of iRODS capabilities as they apply specifically to use cases determined by several different Engineer Research and Development Center (ERDC) work groups. Table 1 shows the iRODS core competencies, the four major strengths of iRODS, and the details of each competency.

Table 1. iRODS core competencies.

Competency	Details
Data Virtualization	Allows access to distributed storage assets under a unified namespace, freeing organizations from getting locked into single-vendor storage solutions
Data Discovery	Utilizes a metadata catalog that describes every file, directory, and storage resource in an iRODS Zone
Automated Workflow	Permits any action to be initiated by any trigger on any server or client in an iRODS Zone
Secure Collaboration	Users only need to log into their home iRODS Zone to access data hosted on remote Zones

Within these core competencies are the eight main capabilities of iRODS (Table 2).

Table 2. iRODS main capabilities.

Capability	Details
Automated Ingest	Automatically onboard new or updated data to iRODS when placed in a specified “landing zone” directory
Storage Tiering	Allows for incoming data to be handled differently depending on what type of data it is, or for timing of how long data should remain in a given location before moving
Compliance	Enforceable user agreements and data access policies as established by the owner of the data
Publishing	Selectively make data sets available that accompany written articles so that research replication can be more easily performed
Provenance	Track the chronology of ownership of data with automatically activated Policy Enforcement Points
Data Integrity	Ensure validity of data with the underlying filesystem or optionally with the SHA256 or MD5 checksum of the data object
Auditing	All actions in the system pass through Policy Enforcement Points, which allows for a highly detailed view of the system as it operates
Indexing	A catalog of both the data and the metadata is kept in a relational database in order to make searching faster and easier

Real world examples of iRODS ability to handle large volumes of data are explained by Chris Smith of Distributed Bio: “A pet interest of mine is to figure out how to make a scheduler more intelligent—where it places jobs with respect to data? You need a metadata service for the scheduler to use. Where is this file located? Where are the replicas?” Smith sees two key attributes to iRODS. First is the ability “to annotate data with metadata over and above the usual Unix time-based stuff. This is the part that will be useful to end users” (Davies 2011).

Second, the rules engine capability makes data management very powerful. “You can execute a number of rules on how you manage data over time. For example, expiry—I have data living on very expensive tier 1 storage. At a certain time, I migrate to tier 2. You can do very complex things—automatic replication to disaster recovery sites, check summing, combat bit rot a little” (Davies 2011).

4 ERS iRods Use Cases

Specific challenges for the ERDC include a need to manage data across multiple geographically-separated users while ensuring enforcement of Department of Defense (DoD) security and authentication standards; group-specific data-related access and usage policies; automation of workflow; data storage, retrieval, replication, back-up, and archiving; quality control; and metadata generation, capture, and cataloging. There are several areas of concern that were important to all groups. Those include the following:

- Data security, access, and permissions policies
- Data storage policies
- Metadata
- Workflow automation

The next sections will detail specific use cases as indicated by different research groups and the solutions that address their unique needs. The iRODS-based solutions will be presented by describing iRODS workflows tailored to each group. Workflows in this sense are structured and repeatable patterns of actions and behaviors that manage and process data and provide actionable services.

4.1 Data analytics ERS rotorcraft maintenance project

The Data Analytics team at ERDC's Information Technology Laboratory (ITL) provides visualization and analysis tools and support services to enable the analysis of an ever-increasing volume of data. For this research effort, the ERS Data Analytics team has been tasked with building a data lake architecture for the ingestion, aggregation, tracking, and analysis of numerous large datasets generated from a diverse group of sensor arrays, flight logs, and repair and maintenance procedures. These files are supplied by AMRDEC. The largest file is a terabyte sized binary data file that when extracted is comprised of 1.3 million individual SQLite databases. The file contains flight information and sensor data generated by the various sensors mounted on rotorcraft platforms. The other files are single SQLite database files that contain flight logbook data, component tracking, and maintenance data. This information is sensitive and must adhere to strict access policies.

Currently, there is a single user who accesses and makes use of the data, but this will increase in the future. This large sensor data file is static and will not be written to. It will be read frequently, so data transfer will be primarily concerned with file retrieval for analysis.

4.2 Data analytics specific challenges

Data analytics require a secure, federated, shared repository that the offsite collaborators from AMRDEC can place data into, monitor, audit, and provide controlled access for necessary ERDC users. Currently, sensor data is aggregated and placed on a server. The binary file is large, preventing transfer over the wire. The server is physically transported to ERDC and the data is downloaded directly from the server. This situation, while workable, is not optimal, nor is it an efficient or productive use of time and energy.

Searching and retrieval of specific files is an issue that is of great concern. Searching for data among the million plus files without metadata proves cumbersome and is a problem that must be addressed to improve workflow efficiency. If not addressed, the problem will only grow as more files are added and as additional users begin to access and perform data analysis on the files. Also, of great importance is the ability to audit file permissions and user access and the creation and strict enforcement of user agreements created by the data owners.

Expressed Needs

- Secure shared repository that allows access to credentialed external collaborators
- File and directory level enforceable user agreements
- Metadata policies
- Strict and granular access permissions
- Customizable auditing tools with data logging and visualization capabilities.

4.3 iRODS related solutions

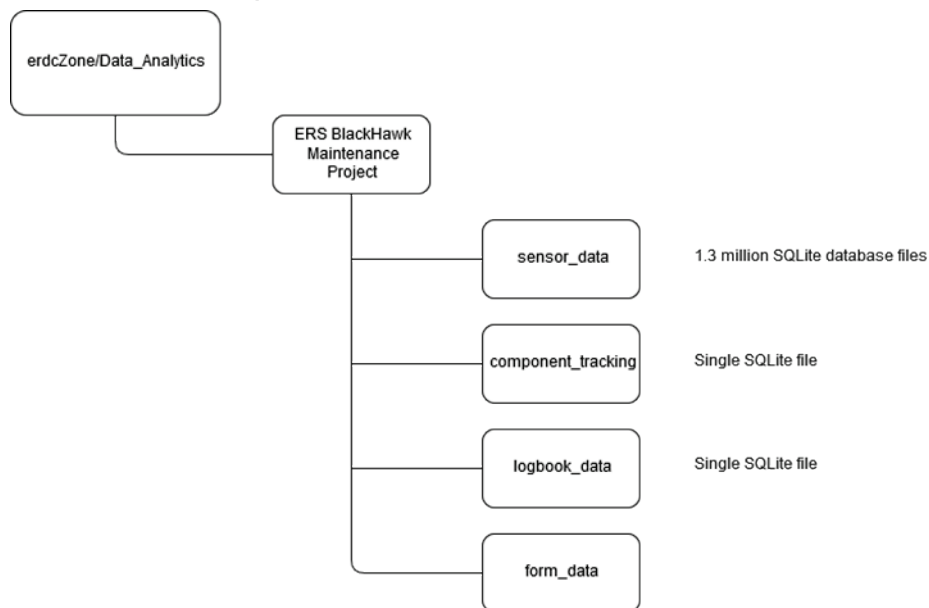
iRODS will provide a secure data repository that meets the DREN security and authentication mandates with limited access and strict enforcement of AMRDEC's access rules. This repository would act as a landing zone (LZ). Flight sensor data could be placed into the LZ periodically instead of

waiting years and allowing the data to grow into a multi-terabyte file. Upon being placed in the folder, iRODS will automatically ingest the data, tag it with metadata, and route it to its correct directory folder via workflow automation polices crafted by the data owners and administrators. Thus, providing a convenient, secure storage location complete with rules, which automate workflow, data retrieval, analysis, and storage.

Tagging the data with metadata will allow slices or portions of the data that the researcher is interested in instead of loading the entire dataset into memory to find the desired information.

Figure 2 illustrates the proposed iRODS directory structure. This is amenable to change and will grow to include other aircraft research projects in the coming months.

Figure 2. Data analytics directory structure.



iRODS further addresses the stated needs of the group by providing auditing tools that will enable the data owners and the administrator to have detailed information about a file or folder over the entire life of the information; from its inception to deletion. This auditing information is available via the iRODS auditing plugin. The plugin audits all user activity and sends audit events to the Elastic Search, Logstash, Kibana (ELK) stack to create visually aesthetic and customizable dashboard graphs. While all groups expressed a deep interest in the data auditing and visualization

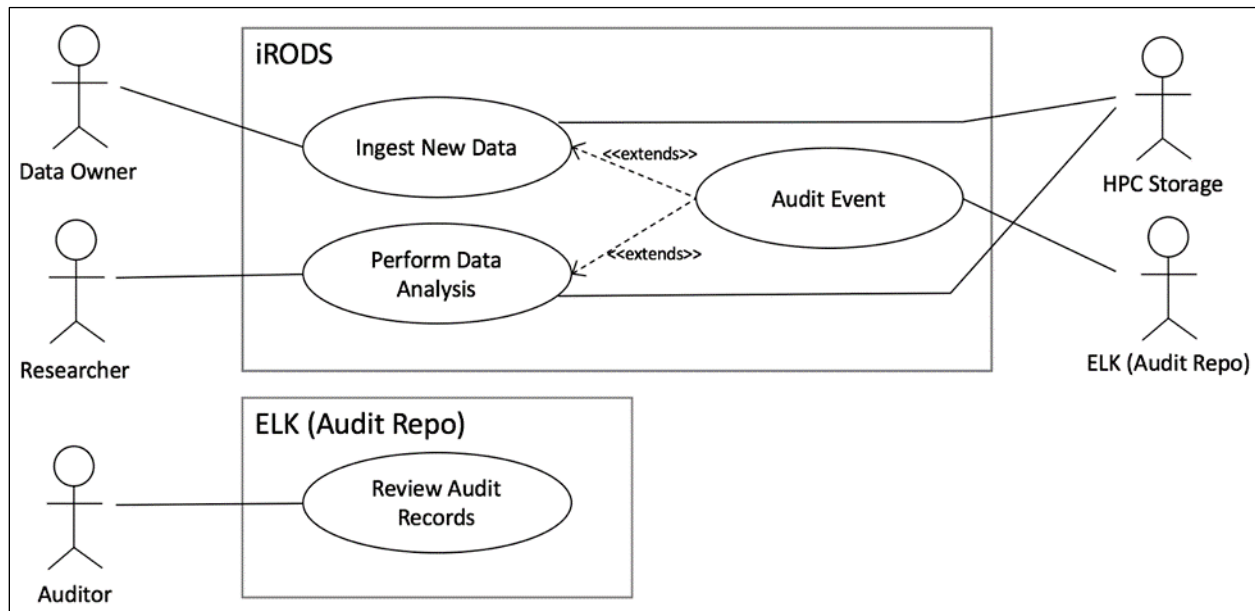
tools, this feature is of special interest to this work group due to the sensitive nature of the aircraft data and the data owner's express demand that data access be tightly controlled, and its use regularly tracked and audited. The auditing tools are customizable to suit the specific areas each group admin chooses to target and monitor.

4.4 Data analytics iRODS use case

4.4.1 Use case: Ingest new data

Description - The AMRDEC data owner shares new data with the Data Analytics research team by accessing the iRODS environment within the DoD Supercomputing Resource Center (DSRC) and placing the data into a LZ (i.e., iRODS data ingestion point) (Figure 3).

Figure 3. Data analytics iRODS use case Unified Modeling Language (UML) diagram.



Actors

- Data Owner (AMRDEC data owner)
- HPC Storage

Process Steps

1. Logon - The data owner logs into the HPC environment using their existing HPC credentials.

2. Upload Data - The data owner navigates to the folder designated as an LZ for the platform the data relates to and uploads the files.
3. Workflow - This action triggers an iRODS rule: automated ingest. The automated ingest rule scans the LZ and detects the new files. The rule tags them with system generated metadata. The rule removes them from the LZ and according to file type moves them to their correct destination.

4.4.2 Use case: Perform data analysis

Description - The ERDC user is going to take sensor information from a set of flights over a certain month. The user joins this sensor information with component tracking and maintenance data from the same time period. This data will be entered as inputs into a machine learning program to determine how different flights affect the craft in terms of required maintenance and part replacement.

Actors

- Researcher (research team Hawk Project Data Manager)
- HPC Storage (Physical storage devices within the DSRC)
- ELK Audit Repo (Elasticsearch, Logstash, and Kibana)

Process Steps

1. Logon - The ERDC user logs into iRODS with a Python script run from the command line. Their account is authenticated and, according to their permissions, the files and folders they have access to are made available to them.
2. Search - The user uses metadata for a quick and efficient search for sensor data from a beginning and ending date range and from a specific set of sensors. This query is joined with a search for maintenance and component tracking data over the same time period
3. Retrieval - The files are found and pushed to the machine learning algorithm.
4. Post Analysis - Depending on the analysis performed, additional metadata will be added if applicable to the results. The inputs will also be updated and tagged with new metadata. Any new material derived from the simulations or analysis will also be tagged with metadata indicating its parent simulation and the creator of the analysis. Other system generated metadata is also applied such as file size, date created, etc.

5. Workflow - iRODS, according to file type, routes the resultant new derived material and output files to their correct destination folder.
6. End - Close iRODS session or repeat steps with new analysis or simulation run.

4.4.3 Use case: Review audit data

Description - The AMRDEC data owner adds new data to the shared repository and performs an audit on the repository to ensure that access protocols are being held to and to track data usage and integrity.

Actors

- Auditor (AMRDEC data owner)
- ELK Audit Repo (Elasticsearch, Logstash, and Kibana)

Process Steps

1. Logon - The data owner (AMRDEC user) has several new sensor files ready to be added to the data currently in the sensor data folder of the ERS Rotorcraft Maintenance Project iRODS directory. They place the files in the LZ.
2. Workflow - This action triggers an iRODS rule: automated ingest. The automated ingest rule scans the LZ and detects the new files. The rule tags them with system generated metadata. The rule removes them from the LZ and according to file type moves them to their correct destination, which in this case is the sensor data folder.
3. Auditing - The admin, via a secure portal, uses the ELK stack to monitor the shared data repository. The admin uses Kibana for visualization of the log data they wish to see. Their dashboard has been configured to display group members, file size, time of access, and data transfer amounts. This information is logged in order to provide a descriptive and detailed set of auditing data in order to maintain a thorough record of how the sensitive aircraft information is being used. Thus, the data owner is aware of who is accessing the data and if the data has been altered.

The following key needs are addressed by iRODS:

- CLI based iRODS access
- Workflow automation via PEPs
- Rich descriptive metadata

- Secure, shared repository that allows access to credentialed external collaborators
- Customizable auditing tools with data logging and visualization capabilities
- Data provenance.

4.5 Environmental simulation

Environmental simulation (ES) conducts research in the areas of hydrodynamic engineering to provide situational information to support other ERS efforts. ES utilizes the expertise of the ERDC Coastal and Hydraulics Laboratory (CHL) Rivers and Estuaries Engineering Branch (REEB) and Hydrologic Systems Branch (HSB) for geomorphic, hydraulic, sedimentation, and hydraulic engineering in the entire river system, from the headwaters to the ocean boundary. REEB and HSB make extensive use of Adaptive Hydraulics (AdH). Per ERDC-CHL (2012) “AdH is a modular, parallel, adaptive finite-element model for one-, two- and three-dimensional flow and transport. AdH is a module of the Department of Defense (DoD) Surface-Water Modeling System and Ground-Water Modeling System. AdH simulates groundwater flow, internal flow and open channel flow. The AdH module was developed in the Engineer Research and Development Center’s Coastal and Hydraulics Laboratory and is a product of the System-Wide Water Resources Program.”

4.6 ES specific challenges

ES’s current AdH workflow process does not include the creation or use of descriptive and searchable metadata. This is one the most requested features common to all workgroups. Once a model that has been developed and validated is moved to long term archival storage it can be difficult to locate or recreate without intimate knowledge of the process behind its creation. This is due to lack of descriptive metadata that would aid a researcher or engineer unfamiliar with the data, but that may need it for a new project or as a continuation to a past project.

ES also desires the creation of rules that would manage storage and routing of files through their file structure and the required application of metadata to model runs that produce a result that a user wishes to save. The need for detailed policies that control storage of critical data is needed to replace the current 30-day automatic purge rule in place on the HPC network.

With a current group structure of 5 – 10 researchers producing and utilizing datasets in the 10 – 100s of gigabytes, there will be a large amount of data transferred daily. This includes both reads and writes.

ES's data policies will need to be flexible in the future as more users are added to the group and as collaborators from other groups and departments will require access to their data.

ES needs to have a clearly defined directory structure that houses the different models and simulations, as well as their required input and resultant output files.

The following are expressed needs for ES:

- Metadata polices
- Storage and retrieval policies for current and newly generated data
- Python client interface
- Secure shared repository that allows access to credentialed external collaborators.

4.7 iRODS related solutions

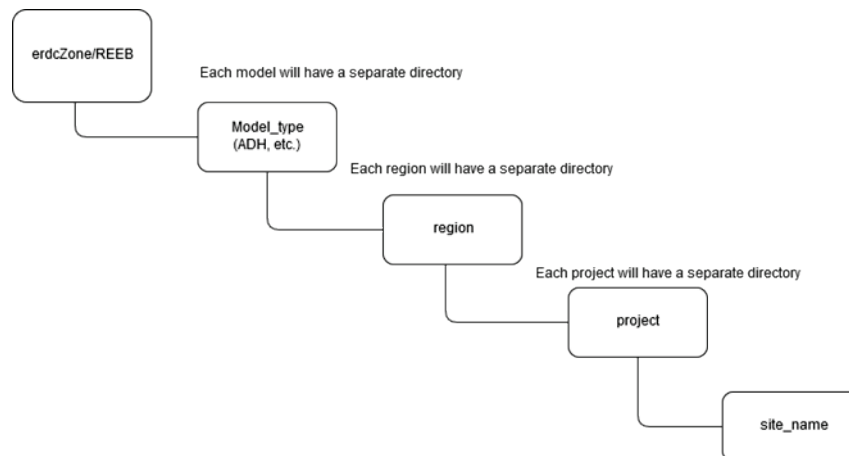
As metadata generation and indexing is one of iRODS core strengths, it is perfectly suited to address this requirement. After several conversations with users, metadata templates that are a combination of automatic system generated metadata, and metadata tags supplied from user input are being developed. The following labels are some of the fields the ES AdH users have decided to use as the basis of their metadata template:

- Project Name
- Author
- Mesh bounding box
- Region label
- Project label
- Model [ADH, ADRIC, etc.]
- Model type [Dam break, Hurricane etc.]
- Model Version
- Mesh projection
- Creation time
- List of output types
- List of project files

These labels, and several others, have been incorporated into a metadata template applied by a rule written in Python that is triggered automatically by the Python Rule Engine when the input files and their outputs are placed in a folder designated as an LZ. The Python Rule Engine is an extensible plugin for iRODS. The Rule Engine Plugin Framework (REPF), which keeps track of system states and interprets both system-defined rules and user-defined rules, is a critical component of the iRODS system. Rules are definitions of actions that are to be performed by the server. These actions are defined in multiple ways, depending on the language that is used to define the actions. In the native iRODS Rule Language, the actions are defined with microservices and other actions. The REPF determines which defined rules are loaded and active and then delegates to the plugins to execute any relevant action. For the Python Rule Engine Plugin, it loads the Python interpreter and executes rules and policies written in Python. By placing a file in this folder, the user indicates that it is data they wish to retain and thus, should be placed in iRODS. This allows admins and users to write and develop rules and policies in a language with which they are familiar.

This rule, written to aid in ES workflow automation, is activated when data is moved into the LZ. It scans the LZ, and according to file type/extension, applies file specific metadata, and moves the files into their designated locations in the group's file structure. From there, additional rules will determine the length of time they are held before being moved to long term storage and/or archive. Figure 4 illustrates the REEB's anticipated ES-affiliated iRODS directory structure.

Figure 4. ES directory structure.

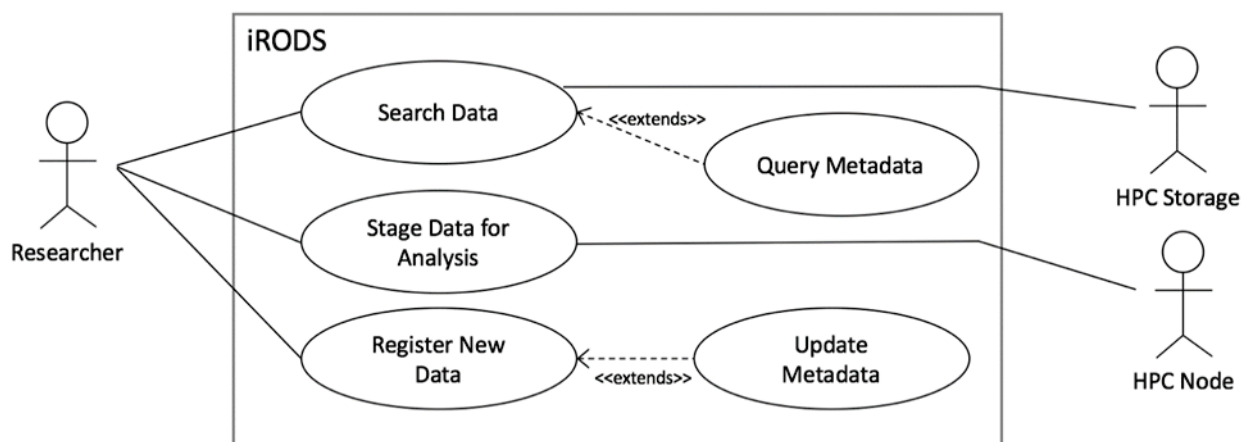


The metadata makes retrieval and reuse of the data a simple and easy matter, and the rules governing storage-tiering automate the process of data storage and prevent unwanted purges of relevant information and files.

This also addresses another key requirement stated by users: the expressed desire to use a Python based API to access iRODS and to create and implement rules and policies with Python scripts.

4.8 ES iRODS use case

Figure 5. ES iRODS use case UML diagram.



Description - The ES user collects the necessary input files to validate a new hydrology model for a river basin. The user logs on to iRODS with the Metalnx graphic user interface (GUI). The required input files are searched for using metadata queries. The model is validated, saved, and the completed model and its updated input files are tagged with metadata and moved to their correct folders. The several runs and their results generated during the validation process are no longer needed and are deleted.

Actors

- Researcher (Environmental Simulation team)
- HPC Storage
- HPC Node

Process Steps

1. Logon - Metalnx is a web application designed to work alongside iRODS. It provides a GUI based alternative to command line interaction with iRODS. For users that prefer this method of access, it simplifies most administration, data management, and metadata creation and querying tasks, removing the need to memorize the long list of iRODS command line arguments. The user clicks their Metalnx icon on their desktop and then enters their username and password into the appropriate fields. The Metalnx and iRODS login credentials are the same.
2. Retrieval - After their directory structure is visible, the user searches for each of the input files needed for the simulation validation runs. The files required for input have an input metadata attribute. Each file can be queried with this tag joined by an inclusive search on the type of model and the date range need plus any other metadata attributes the user chooses.
3. Staging - After finding the required files, the user places them together in a single staging folder, which is then pushed to either Onyx or Topaz.
4. Validation - After completing the number of runs required until the desired results are achieved, the user is ready to move the outputs into iRODS. After a model is completely validated, the completed model and its updated inputs moved to the REEB LZ. Previous runs that the user no longer needs to keep are discarded.
5. Workflow - This transfer of files into this folder triggers the Python Rule Engine. This action on this folder represents a dynamic policy enforcement point (PEP). A PEP is a logical location in the iRODS Server code where policy can be inserted by an organization. Policy takes the form of a set of rules that are executed by the rule engine. The user created rule runs a scan of the folder, and new or changed files are then tagged with system generated and user supplied metadata. The rule enforces a policy that for the data to be retained and moved to its correct folder, it must have metadata appended to it. After application of metadata, the files are routed to their predefined folders contained in the project directory according to file type. This allows the files to be accessible to authorized users.

The following key needs are addressed by iRODS:

- GUI for interacting with iRODS
- Workflow automation via rules and PEPs
- Rich descriptive metadata policies with selective enforcement.

4.9 Sensor systems

The ERS Sensor Systems group conducts applied research in geospatial and related sciences. The group conducts research, development, and evaluation of current and emerging geospatial technologies that will help characterize and measure phenomena within the physical (terrain) and social (cultural) environments encountered by the Army. Their research areas and capabilities include terrain analysis and reasoning, passive and active spectral signature analysis, fluorescence, photogrammetry, terrain visualization, precision surveying and mapping, image analysis, data management, geographic information systems, data/image fusion, and spatiotemporal reasoning and numerical analysis (ERDC-GRL 2016).

4.10 Sensor systems specific challenges

The Sensor Systems group, due to the broad scope of their research and their work with utilizing data from remote external sensors over a large geographical area, has many unique challenges to address.

With a current group structure of 10+ researchers producing and utilizing datasets in the 10- 100s of gigabytes, there will be a large amount of data transferred daily. This includes both reads and writes.

Currently, the group uses the Air Force Research Laboratory (AFRL) Girder data management system. When combining files for model simulation runs, a user will refer to the large files to prevent replicating and transferring several multi-gigabyte files. These references are universally unique identifiers (UUID) generated by Girder and are not easily human readable, being comprised of a long string of alphanumeric text, and thus, are susceptible to user error when copying them. The sensor group would like to replace this with a system that will provide a more efficient and streamlined automated workflow that includes the use of a command line interface combined with a Python API. The group requires short and long-term storage policies, including descriptive metadata, and will make extensive use of this metadata when making queries to retrieve data for research purposes or simulation and model runs. The sensor group would also like to make use of the auditing and editing tools to track permissions, file utilization, and resource allocation.

The following are expressed needs:

- Metadata polices
- Extensive use of database querying and indexing capabilities
- Storage and retrieval policies for current and newly generated data
- Python client interface for access and rule/policy creation and implementation
- Secure, shared repository that allows access to credentialed external collaborators
- Customizable auditing tools with data logging and visualization capabilities.

4.11 iRODS related solutions

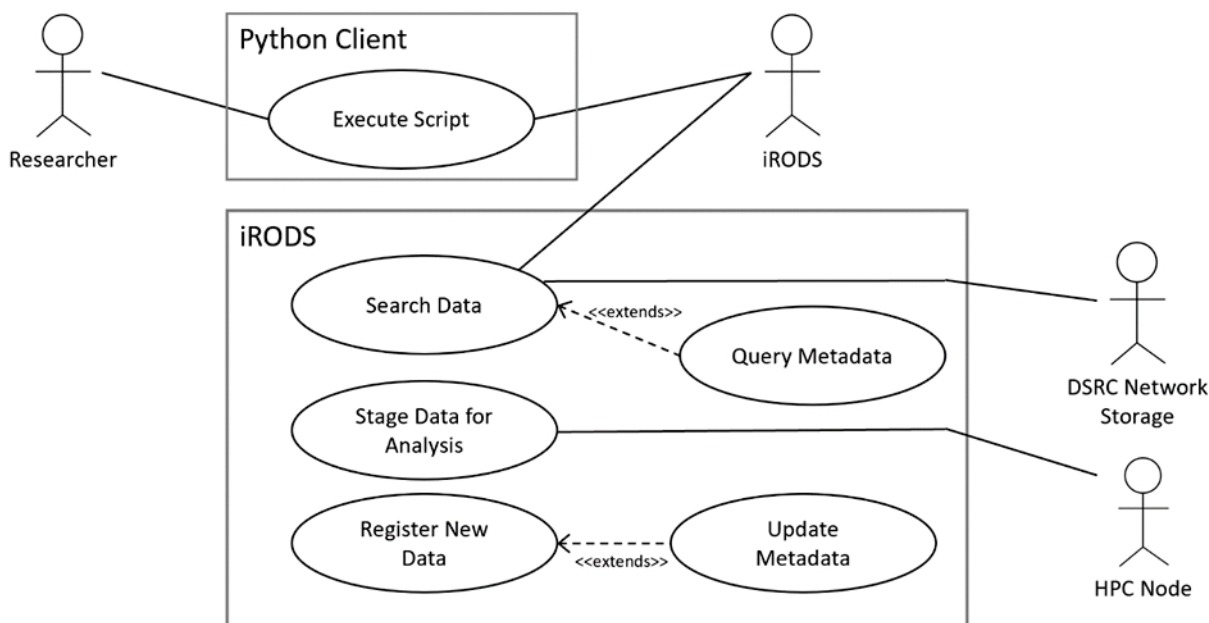
The switch to iRODS will eliminate the current system of UUIDs that is currently in place. Workflow procedures can be developed that automatically retrieve and stage data from various resources and collections without the need for UUID references pointing to required meshes and input files. These policies, which will be written in Python with the sensor system collection admin and data owner, will streamline data retrieval and analysis, model and simulation runs, and incorporating and ingesting data from external sources.

The sensor group will make heavy use of iRODS core capability of metadata creation, indexing, and querying the metadata database to retrieve and link files.

4.12 Sensor systems iRODS use case

Figure 6 depicts the sensor system iRODS use case: The sensor system user collects the necessary input files to validate a new sensor model for soil testing in a tropical region. The user logs on to iRODS with the iRODS-Python Client. The required input files are searched for using metadata queries. The model is validated, saved, and the completed model and its updated input files are tagged with metadata and moved to their correct folders. The several runs and their results generated during the validation process are no longer needed and are deleted. The user repeats this process many times to create several validated models. The completed models and their inputs are tarred together and sent to a special repository created by the user to share with a select group. When the files reach their destination folder, they are automatically untarred by iRODS.

Figure 6. Sensor systems iRODS use case UML diagram.



Actors

- Researcher (Sensor Systems team)
- iRODS
- HPC Storage
- HPC Node

Process Steps

1. Logon - The user logs into the network and begins an iRODS session via a command line Python script. The script contacts the iRODS server, authenticates the user, and instantiates a client session.
2. Retrieval - The user performs metadata-based searches to locate the files needed for simulation input.
3. Staging - After locating the required files with metadata queries, the file paths are placed in a Python script to pull the files for staging and transfer to Onyx or Topaz.
4. Validation - The user runs a Python script from the command line that performs several actions. The script transfers the files to the HPC network for simulation runs. After completing the number of runs required until the desired results are achieved, the model is validated and ready to be moved to its correct folder for later reuse.

5. Workflow Automation - The completed model, its outputs, and its updated inputs are tagged with user and system generated metadata and moved to their correct folders. Metadata on the collection folders is also updated. Previous runs that the user no longer needs to keep are discarded. These steps are repeated for several different simulation runs. The derived data of all models, when ready to transfer to long term storage or retrieval for analysis, is aggregated and tarred into large files that, after transfer, are automatically untarred into their constituent files when placed in their destination locations. These files are routed to their respective storage repositories by iRODS based on metadata.

The following are key needs addressed by iRODS:

- Python API for interacting with iRODS
- Workflow automation via rules and PEPs
- Rich descriptive metadata
- Granular permissions.

4.13 General HPC user

ERDC operates a variety of large HPC systems to serve the HPC needs of engineers and scientists throughout the DoD. It provides a complete HPC environment, including hardware, software, data storage, archiving, visualization, training, and expertise in the Computational Fluid Dynamics (CFD), Computational Structural Mechanics (CSM), and Environmental Quality Modeling (EQM) Computational Technology Areas as well as Education Outreach and Training Coordination (EOTC) (DoD 2016).

4.13.1 General HPC user challenges

This user has stated they would like to have many of the same features as the other groups, such as access policies, storage and retrieval rules, and metadata generation, but with certain differences. They would like some of their data to remain on the HPC network and have other portions of the data stored separately. ERDC has several Git repositories specific to a scientific computation project or application. The user would like their code to remain on the HPC network in the current ERDC Git repository specific to the project they are working on. The output files from the model and simulation runs would also be stored on the HPC network on both Onyx and Topaz. Other information related to the project such as research, scholarly articles, CSV files, AVI files, etc. can be stored separately outside

of the HPC network. The user requires descriptive metadata on all of their files and folders, both internal and external to iRODS, and detailed storage policies to replace the current 30-day purge rule that exists on the HPC network.

The following are expressed needs:

- Metadata polices
- Storage and retrieval policies for current and newly generated data
- Lustre file system scan and in place registration of certain files and directories.

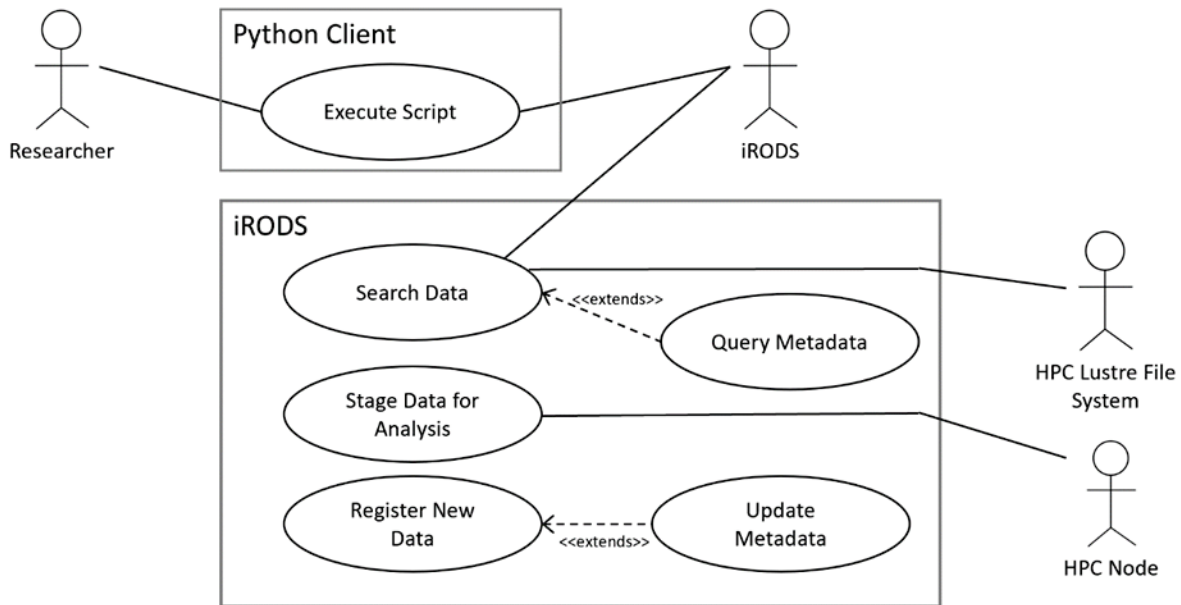
4.13.2 iRODS related solutions

The user has stated they would keep their code base external to iRODS, but they would need all file locations indexed and stored in iRODS with metadata to enable quick searching and retrieval of required code through iRODS. One of iRODS' strengths is its ability to work with multiple file systems. iRODS will scan the file system and register the location of the external data and store its location with metadata. iRODS can also implement a storage policy for the user that scans the file system and pulls files older than a certain time period, moves them into iRODS for storage and then, when necessary, transfers them to long term archival. These movements would be noted in iRODS' database which will update the file path registrations and metadata attributes. iRODS will internally house important descriptive information about the various code bases and relevant research and descriptive material about ongoing projects.

4.14 General HPC iRODS use case

Description - The HPC logs onto iRODS to search for code housed in the HPC Lustre File system. After retrieving the code and pushing it to Onyx, the outputs, updated files, and any newly generated code are placed in their correct locations. These locations are updated in iRODS and tagged with metadata. The user then implements a Python rule, which will move data into iRODS for storage and archival.

HPC iRODS use case UML diagram.



Actors

- ERDC HPC User

Process Steps

1. Logon - The user logs into iRODS with a Python script run from the command line.

The script contacts the iRODS server, authenticates the user, and instantiates a client session.

2. Retrieval - Using metadata queries from the command line, the user locates the code they wish to move to Onyx. The code is stored external to iRODS, but iRODS has an updated database of file and folder locations with metadata of code stored in the Lustre file system. With iRODS metadata queries, retrieval of the requested code, although it is housed externally to the iRODS storage vaults, is quick and efficient.
3. Staging - After locating the desired code, the user moves the code to Onyx. HPC - Computation is performed on the HPC network.
4. Lustre File System Scan - Outputs are moved to their location on the HPC Lustre file system and the new files and their locations are registered in place in iRODS. iRODS updates its database of the Lustre file system.

5. Storage - To prevent unwanted deletion, the user has a Python rule in iRODS that scans the Lustre File System. Any files that user wishes to keep are moved into iRODS before they are purged for being in violation of the HPC 30-day rule. These files are held in iRODS and then moved to long term storage according to parameters the user has defined in their rule.

The following are key needs addressed by iRODS:

- Rich descriptive metadata
- Python API for interacting with iRODS
- Indexing and iRODS in-place registration of code on Lustre File System
- Custom storage policies.

5 Conclusion

ERDC is engaged in cutting-edge, next-generation research that produces massive volumes of digital information. Climate simulation models, computational fluid dynamics calculations, and geospatial research are only a few of the projects that ERDC conducts. These programs involve diverse systems, sensors, technologies, interfaces, remote locations, and collaborators. Data intensive modeling analyses require large, varied sets of input data and produce equally, or more complex, outputs with ownership of the data many times held by external stakeholders. iRODS offers powerful solutions to common needs and problems faced by all groups as well as flexible and unique answers to the requirements of individual units.

Any new system must be intuitive and have a low barrier of operation for new users. iRODS offers a scalable, flexible interface to the user and abstracts away the underlying physical architecture, which allows the scientist or engineer to focus on their job without the impediment of a complicated, difficult to intuit system of data management and access.

iRODS provides ERS researchers the ability to create, store, index, monitor, reproduce, and retrieve their data in automated workflows accessed by the method of their choice. This can be done either via actionable scripts written in Python, activated through the command line, or by using a GUI. iRODS empowers researchers by allowing them to harness the abilities of their HPC environments while providing federated, secure, work repositories accessible by credentialed external collaborators and stake holders. This allows interoperability across a variety of internal and external parties and infrastructure systems.

As data volumes at ERDC grow and as the services utilized and simulations run at ERDC become more complex, iRODS can become a powerful manager of this generated data. Virtualization will allow users to store, search, retrieve, and control how their data is shared and accessed regardless of where and on what device their data is housed. Data discovery via customized metadata will allow ease of retrieval and when combined with specifically tailored rules and policies will enable automation and simplification of a user's workflow. Secure collaboration will allow external users to access and deposit data into iRODS and allow for granular file and directory access policies that can be developed at the

discretion of the owner of the data. iRODS also gives division chiefs and project managers the ability to monitor and audit the use of critical resources under their supervision through customizable dashboards and auditing infographics. This is key for several reasons; performing automatic, periodic analysis of data integrity, relevancy and storage prevents the retention of unnecessary and outdated data. Efforts and resources can be allocated and focused where most needed and, when necessary, manpower and hardware can be shifted to where it can be put to the best use.

The advantages of iRODS include the increased ability to share, reproduce, and ensure the integrity of data as well as enabling collaboration on projects with user-authored, automated workflows. By illustrating several approaches using iRODS for addressing the challenges faced by ERS in creating secure, reproducible scientific workflows on large data sets, specifically tailored solutions have been provided to a wide range of challenges faced by different ERS research teams.

References

- Berkeley Lab / U.S. Department of Energy. 2018. Scientific Data Management. Berkeley, CA: University of California Berkeley Lab. (Accessed 2 January 2019). <http://crd.lbl.gov/departments/data-science-and-technology/SDM/>.
- Chiang, G.-T., P. Clapham, G. Qi, K. Sale and G. Coates. 2011. Implementing a genomic data management system using iRODS in the Wellcome Trust Sanger Institute. *BMC Bioinformatics* 12(1): 361. <https://doi.org/10.1186/1471-2105-12-361>.
- Clapham, P., J. Constable, and D. Holland. 2018. Informatics Support Group | High Performance Computing. Wellcome Sanger Institute, Genome Research Limited. (Accessed 2 January 2019) <https://www.sanger.ac.uk/science/groups/informatics-support-group>.
- Davies, K. 2011. iRODS: The metadata solution to data management. *Bio-IT World*. (Accessed 1 February 2011). http://www.bioworld.com/BioIT_Article.aspx?id=104220#Wellcome_Relief.
- Department of Defense - Supercomputing Resource Center (DoD-SRC). "About the ERDC DSRC. 2016. About the Department of Defense Supercomputing Resource Center. (Accessed 24 May 2016). <https://www.erdchpc.mil/about/index.html>.
- Schnase, J. L., D. Q. Duffy, G. S. Tamkin, S. Strong, D. Ripley, R. Gill, S. S. Sinno, Yingshuo, L. E. Carriere, L. M. Brieger, R. W. Moore, A. Rajasekar, W. Schroeder and M. Wan. 2011. iRODS-based climate data services and virtualization-as-a-service in the NASA center for climate simulation. American Geophysical Union, Fall Meeting 2011, abstract id. IN41A-1397.
- Sivarajah, U., M. M. Kamal, Z. Irani, and V. Weerakkody. 2017. Critical analysis of big data challenges and analytical methods. *Journal of Business Research* 70:263–286.
- Texas Advanced Computing Center, Office of the Vice President for Research. 2018. Corral User Guide. Austin, TX: University of Texas at Austin Advanced Computing Center. (Accessed 7 January 2019). <https://portal.tacc.utexas.edu/user-guides/corral>.
- U.S. Army Engineer Research and Development Center - Coastal and Hydraulics Laboratory (ERDC-CHL). 2012. Adaptive Hydraulics Model System. (Accessed 2 January 2019). <https://www.erdch.usace.army.mil/Media/Fact-Sheets/Fact-Sheet-Article-View/Article/476708/adaptive-hydraulics-model-system/>.
- U.S. Army Engineer Research and Development Center - Geospatial Research Laboratory (ERDC-GRL). 2016. GRL Missions. (Accessed 2 January 2019). <https://www.erdch.usace.army.mil/Locations/GRL/Missions.aspx>.

Acronyms and Abbreviations

Acronym	Meaning
ACL	Access Control Lists
AdH	Adaptive Hydraulics
AFRL	Air Force Research Laboratory
AMERDEC	U. S. Army Aviation and Missile Research Development and Engineering Center
CFD	Computational Fluid Dynamics
CHL	Coastal and Hydraulics Laboratory
CLI	Command Line Interface
CSM	Computational Structural Mechanics
DoD	Department of Defense
ELK	Elastic Search, Logstash, Kibana
DSRC	DoD Supercomputing Resource Center
EOTC	Education Outreach and Training Coordination
ERDC	Engineer Research and Development Center
ERS	Engineered Resilient Systems
ES	Environmental Simulation
EQM	Environmental Quality Modeling and Simulation
GUI	Graphic User Interface
HPC	High Performance Computing
HSB	Hydrologic Systems Branch
ITL	Information Technology Laboratory
iRODS	Integrated Rule-Oriented Data System
LZ	Landing Zone
PAM	Pluggable Authentication Modules
PEP	Policy Enforcement Point
REST	Representational State Transfer
REEB	Rivers and Estuaries Engineering Branch
REPF	Rule Engine Plugin Framework
UML	Unified Modeling Language
USACE	U.S. Army Corps of Engineers
UUID	Universally Unique Identifiers
WAR	Web Application Resource

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) August 2020		2. REPORT TYPE Final report		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Analysis of ERS Use Cases for iRODS				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Kevin D. Winters, Mark A. Cowan, Glover E. George, Megan E. Gonzalez, Brian Priest, Omar Morris, and Jonathan Landrum				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER 92L5D8	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center Information Technology Laboratory 3909 Halls Ferry Road, Vicksburg, MS 39180-6199				8. PERFORMING ORGANIZATION REPORT NUMBER ERDC/ITL SR-20-8	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Headquarters, U.S. Army Corps of Engineers Washington, DC 20314-1000				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The purpose of this paper is to discuss the challenges inherent with High Performance Computing (HPC) data storage access and management, the capabilities of iRODS, and the analysis of several Engineered Resilient Systems (ERS) use cases relating iRODS capabilities to the teams stated needs. Specifically, these teams are the ERS Data Analytics group (specifically their research on rotorcraft maintenance in conjunction with the U. S. Army Aviation and Missile Research Development and Engineering Center [AMRDEC]), the ERS Environmental Simulation research team, the ERS Sensor Systems research team, and the HPC/Scientific computing group representing the "General HPC User."					
15. SUBJECT TERMS				iRODS (Computer system) High performance computing Computer architecture Computer science Web-based user interfaces	
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT UNCLASSIFIED	b. ABSTRACT UNCLASSIFIED	c. THIS PAGE UNCLASSIFIED			19b. TELEPHONE NUMBER (include area code)