Server-Side Workflow Execution using Data Grid Technology for Reproducible Analyses of Data-Intensive Hydrologic Systems

Bakinam T. Essawy (orcid.org/0000-0003-2295-7981)

Department of Civil and Environmental Engineering,

University of Virginia, Charlottesville, Virginia

Jonathan L. Goodall* (orcid.org/0000-0002-1112-4522)

Department of Civil and Environmental Engineering,

University of Virginia, Charlottesville, Virginia

Hao Xu (orcid.org/0000-0001-6659-6511)

Data Intensive Cyber Environments (DICE) Center,

University of North Carolina, Chapel Hill, NC

Arcot Rajasekar (orcid.org/0000-0003-2280-386X)

School of Library and Information Science,

University of North Carolina, Chapel Hill, NC

James D. Myers (orcid.org/0000-0001-8462-650X)

Inter-university Consortium for Political and Social Research,

University of Michigan, Ann Arbor, MI

Tracy Kugler (orcid.org/0000-0002-3427-9789)

Minnesota Population Center,

University of Minnesota, Minneapolis, MN

Mirza M. Billah (orcid.org/0000-0002-9716-4102)

Department of Biological Systems Engineering,

Virginia Tech, Blacksburg, Virginia

Mary C. Whitton (orcid.org/0000-0003-2880-2550**)**

Renaissance Computing Institute (RENCI)

University of North Carolina, Chapel Hill, NC

Reagan W. Moore (orcid.org/0000-0003-2363-413X**)**

School of Library and Information Science,

University of North Carolina, Chapel Hill, NC

* To whom correspondence should be addressed (E-mail: *goodall@virginia.edu*; Address: University of Virginia, Department of Civil and Environmental Engineering, PO Box 400742, Charlottesville, Virginia 22904; Tel: (434) 243-5019)

Running title: Workflow Execution using Data Grids

Keywords: Hydrologic modeling, Workflows, Reproducibility, Federation, iRODS

Key points (less than 100 characters each):

- Reproducibility of data-intensive analyses remains a significant challenge
- Data grids are necessary for reproducibility of workflows using large, distributed data sets
- Data and computations should be co-located on servers to create executable Web-resources

Index terms: Informatics/Cyberinfrastructure [1908]; Hydrology/Modeling [1847]; Informatics/Workflow [1998]; Informatics/Software tools and services [1976]; Informatics/ Software re-use [1978]

Abstract (< 250 words)

Many geoscience disciplines utilize complex computational models for advancing understanding and sustainable management of Earth systems. Executing such models and their associated data pre- and post-processing routines can be challenging for a number of reasons including (1) accessing and pre-processing the large volume and variety of data required by the model, (2) post-processing large data collections generated by the model, and (3) orchestrating data processing tools, each with unique software dependencies, into workflows that can be easily reproduced and reused. To address these challenges, the work reported in this paper leverages the Workflow Structured Object (WSO) functionality of the Integrated Rule-Oriented Data System (iRODS) and demonstrates how it can be used to access distributed data, encapsulate hydrologic data processing as workflows, and federate with other community-driven cyberinfrastructure systems. The approach is demonstrated for a study investigating the impact of drought on populations in the Carolinas region of the United States. The analysis leverages computational modeling along with data from the Terra Populus (TerraPop) project and data management and publication services provided by the Sustainable Environment-Actionable Data (SEAD) project. The work is part of a larger effort under the DataNet Federation Consortium (DFC) project that aims to demonstrate data and computational interoperability across cyberinfrastructure developed independently by scientific communities.

Summary (< 75 words)

Executing computational workflows in the geosciences can be challenging, especially when dealing with large, distributed, and heterogeneous data sets and computational tools. We present a methodology for addressing this challenge using the Integrated Rule-Oriented Data System (iRODS) Workflow Structured Object (WSO). We demonstrate the approach through an end-to-end application of data access, processing, and publication of digital assets for a scientific study analyzing drought in the Carolinas region of the United States.

1. Introduction

There is an exponential growth in data available to geoscientists. The quantity of satellite data are growing rapidly (Acharya et al., 1998) and data from sensor networks are being widely used, in observatories such as the Critical Zone Observatory (CZO) (Anderson et al., 2008), the National Ecological Observatory Network (NEON) (Cowles et al., 2010), and the Ocean Observing Initiative (OOI) (Keller et al., 2008). Various groups are making available large collections of model-derived data including climate projections and reanalysis products for use by scientists. Public data repositories are used in many scientific disciplines as a means for sharing data collected by the so called “long-tail” of the scientific community (Dunlap et al., 2008). The number of public repositories will likely increase as funding agencies enforce requirements that scientists submit data products resulting from their funded research to these public repositories.

This exponential growth in data will impact modeling and data analysis approaches used in many geoscience disciplines. As datasets grow in complexity and resolution, there is a need for improved tools to derive information from raw data sources in support of a particular research objective. These challenges arise not only because processing large, semantically-unstructured datasets can be complex and time consuming, but also because capturing the computational workflows scientists complete for a particular study can be challenging. New strategies are needed so that these scientist-authored computational workflows can make use of the latest available data and be reproduced and reused by other scientists.

One strategy for dealing with the growing volume of available data has focused on creating standards for accessing remote data collections using Web service Application Programming Interfaces (APIs). The Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic Information System (HIS) has created standards for both an API called Water One Flow (WOF) and a data exchange language called Water Markup Language (WaterML) to facilitate transmission of hydrologic time-series data on large repositories using Web services (Maidment, 2008). The Open Data Access Protocol (OpenDAP) is another widely used protocol for accessing and subsetting scientific data using Web services (Cornillon et al., 2003). OpenDAP focuses in particular on gridded data and includes the concept of server-side data subsetting and format conversion that are essential for operating on large, remote files.

While the Web service approach for data access has significant benefits, it also has limitations in that the network protocol for performing the data transfers using Web services operates over HTTP. For large files, this approach is not optimal and potentially not feasible. Data grid technology provides an alternative approach for managing distributed data and computational resources. Data grids typically include features such as authentication, replication, authorization, auditing, and metadata support that are needed to manage large, distributed data collections (Foster, 2011; Rajasekar et al., 2010). These tools are better suitable for handling large files compared to Web services because they allow for parallel data transfers and provide automated fault tolerance and restarts when connectivity is lost during a transfer. Data grid technology has been used in the atmospheric and climate sciences, notably in the Earth System Grid and Earth System Grid Federation projects (Williams et al., 2011, 2008), but it has not been widely adopted in other geosciences disciplines to date. In particular, research is needed to determine best practices and approaches for leveraging the technology to address specific needs in the hydrologic modeling community, which is the focus of this research.

The objective of this research is to explore approaches for leveraging data grid technology in hydrologic modeling to support reproducible workflows using large datasets. This is some of the first research applying data grid technology for hydrologic modeling. Its primary contribution is a general methodology for analyzing large, distributed data collections, by moving processing to data and using data grids to automate data transfers and staging, in combination with automated formal publication of generated data assets. This will be important as hydrologists seek to scale up watershed models to larger river basins where data sizes and computational processing make reproducibility more challenging.

The work is focused on a use case where a scientist wishes to create a workflow automating the data processing steps required to create a publication-ready figure from a large collection of model output files, greater than 2GB for a single run, produced using a Variable Infiltration Capacity (VIC) (Liang and Lettenmaier, 1994) hydrologic model. The use case, which is more fully explained in Section 3, demonstrates server-side data processing on large data collections, using data grid technology for data transfers, and federation with public data repositories for reproducibility of the analysis workflow. It represents one of the first applications of the newly developed Workflow Structured Object (WSO) functionality in the iRODs, which has general applicability to other scientific domains with significant data management challenges. While systems like MyExperiment (De Roure et al., 2009) also focus on server-side execution of scientist-authored workflows and provide advanced features for workflow sharing and publication, they focus on using Web services for data transfer rather than grid technology.

This research also addresses the challenge of federation across different cyberinfrastructure systems. It is likely that data-intensive studies will need to access many cyberinfrastructure systems for data gathering, processing, modeling, and publication. This paper demonstrates this concept for a use case that involves three cyberinfrastructure systems: the DataNet Federation Consortium (DFC) for data storage and compute resources, the Sustainable Environment-Actionable Data (SEAD) for data publication, and Terra Populus (TerraPop) for data access. Federation across these systems requires agreed upon standards and protocols that allow for interoperability. Different types of federation are demonstrated in our solution in order to address the transfer and management of both large and small data collections.

This paper is part of a special issue on the Geoscience Paper of the Future (GPF). GPF is envisioned as a paper where all digital assets used in the study are published as open, online resource published with unique identifiers and key metadata including titles, abstracts, licenses, authors, and contacts. In this paper, the key digital assets are published through SEAD with Digital Object Identifiers (DOIs) and key metadata attributes. The research itself is also aimed at the vision and goals of GPF focusing in particular on the use case where computation is needed on distributed data resources. It seeks to define methods for moving data from distributed servers within a data grid automatically using federation approaches and defining workflows that aid in capturing the provenance of how data were moved and processed to create publication-ready visualizations generated using multiple reference data collections. As data volumes continue to grow, such techniques will be critical to achieve the GPF goals.

The remainder of the paper is organized as follows. In Section 2 we provide background on data grid technology to orient the reader. In Section 3 we present the use case in further detail, followed by the design and implementation of a prototype system for solving the use case in Section 4. Finally, we provide a discussion of key aspects of our approach in Section 5 before offering concluding remarks in Section 6.

2. Data Grid Technology

Data grids are systems that enable access and sharing of large data sets that are physically distributed across the Internet, but appear to the user as a single file management system. The Integrated Rule-Oriented Data System (iRODS) is a data management system that includes the capability to federate data grids (Rajasekar et al., 2010). Federation allows for the creation of virtual data collections by logically arranging data from distributed resources under a virtual collection hierarchy. Globus is another data grid technology and is used within scientific communities and includes GridFTP for fast data transfer of large files (Foster, 2011). While iRODS and Globus are commonly used within some specific scientific domains (Allcock et al., 2002; Kyriazis et al., 2008), their use is not widespread within the hydrology community.

Data grids are particularly useful for scientific communities such as hydrology that rely on multiple data and computational resource providers. The iRODS-powered Data Federation Consortium (DFC) grid, which is used for this research, was developed as part of a National

Science Foundation (NSF) funded project and provides support for federation of both resources and services. The work reported here is part of the DFC project and uses a DFC data grid for storage and long-term access to datasets stored across heterogeneous resources. The core iRODS software is developed and maintained by the iRODS Consortium at the Renaissance Computing Institute (RENCI), which is a partnership between the University of North Carolina at Chapel Hill (UNC-CH) and the Data Intensive Cyber Environments (DICE) Center at UNC-CH. iRODS currently runs in Linux/Unix environments.

iRODS has a client-server architecture. The iRODS client software can be installed and run on any computer. Each iRODS grid installation has two types of servers: exactly one iRODS Metadata Catalogue (iCAT) server and one or more iRODS resource servers, most frequently storage resource servers, e.g., data disks. Our system was developed on iRODS release 4.0, which includes software for the iRODS client, the resource server, and the iCAT server. iRODS uses the term zone as an abstraction for the physical components of an iRODS grid installation, i.e., the iCAT server and one or more resource servers that are part of the grid.

This work uses the recent development of iRODS Workflow Structured Objects (WSO), which enable workflows to be executed directly with iRODS commands. While iRODS is a mature, widely used software tool, this is some of the first work using the WSO functionality of iRODS. Therefore, this research was completed as a close collaboration between hydrologists defining the scientific workflows and the iRODS and WSO developers made possible through the DFC project. One goal of this work was to provide an example use case of applying WSO that could be beneficial for other iRODS users with interests in utilizing WSO in the future.

Figure 1a provides an overview of the file structure for a WSO. A WSO requires two primary files: a workflow file (*.mss) and a parameter file (*.mpf). The workflow file defines the sequence of operations to be performed by the workflow and the parameter file lists the input arguments used when executing the WSO. The parameter file also specifies any files in iRODS that should be staged-in (transferred to the physical directory on the iRODS resource server where the WSO is executed) or staged-out (put into an iRODS collection) prior to and following the execution of the workflow (Rajasekar, 2014). Examples of workflow and parameter files are provided in iRODS documentation, specifically from https://wiki.irods.org/index.php/Workflow_Objects_(WSO)#Files_in_WSO.

When the user creates and uploads a parameter file, iRODS automatically generates a run file (*.run), which is then used by the client to execute the workflow. One workflow file can be used to create many instances of a WSO with each instance having a unique parameter file (see the wso, wso0, and wso1 collections illustrated in Figure 1). The data files used by the workflow are stored in runDir collections. Within each WSO, there could be multiple runDir collections, one for each execution of the workflow. Workflows can include scripts and other scientist-authored code installed on the server in the iRODS/server/bin/cmd directory (Figure 1b).

A WSO is executed by performing the following steps. (1) The user issues the *iput* command, which is part of the iRODS icommands client library, to transfer a workflow file (*.mss) from a client machine into an iRODS collection. (2) The user issues the *imkdir* command to make a new collection within the collection containing the workflow file (see the wso collection shown in Figure 1). (3) The user issues the *imcoll* command to mount this newly created collection. (4) The user issues the *iput* command to transfer a parameter file (*.mpf) into the mounted collection. This operation results in the system creating a run file (*.run) in the mounted collection. (5) The user issues the *iget* command on the run file to execute the workflow. The system then creates a new collection in the mounted directory (see the runDir collection shown in Figure 1) and the staged-in and workflow generated output files are stored in this new collection. The same workflow can be executed for different parameter files by repeating steps 4 and 5 for a new parameter file, with each new parameter file resulting in an additional WSO collection (see wso0, wso1, …. shown in Figure 1) ("Workflow Objects (WSO)," 2013).

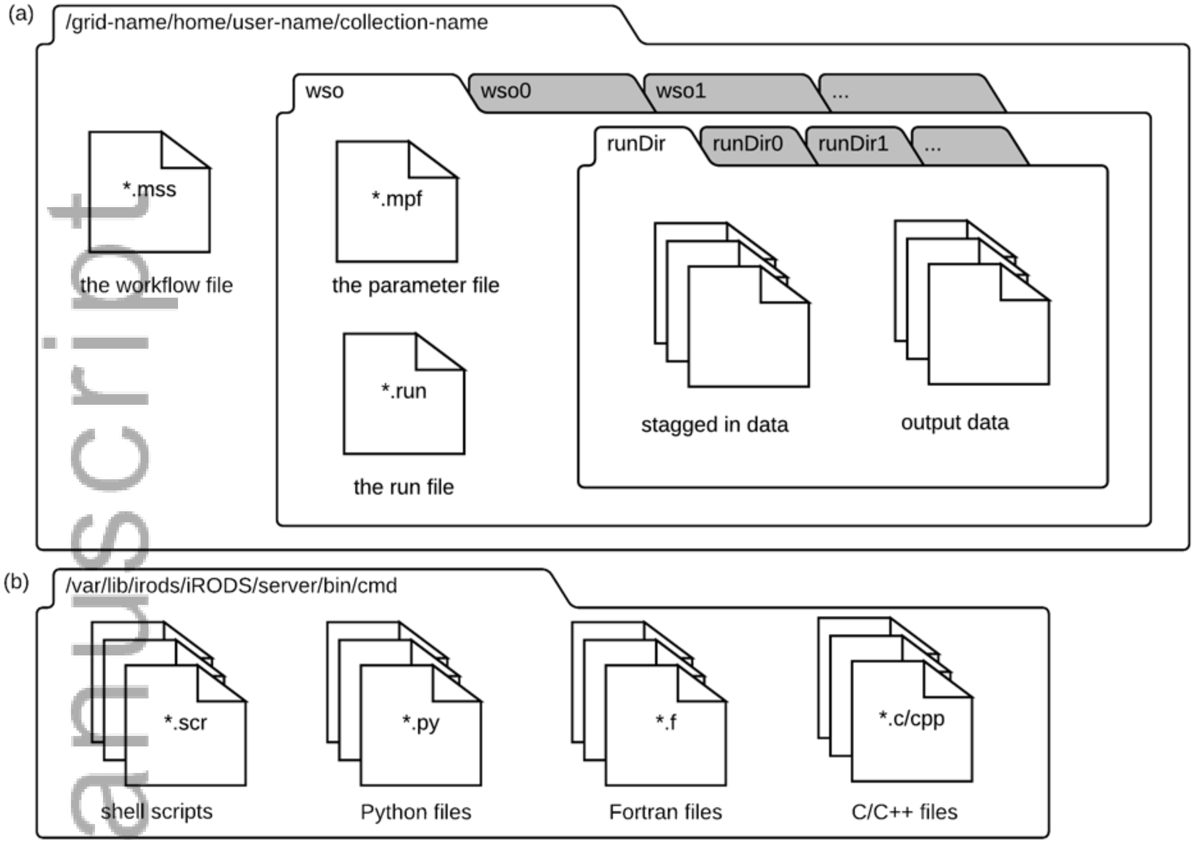


Figure 1. (a) The structure of an iRODS Workflow Structured Object (WSO). (b) The WSO may utilize scripts installed in the iRODS/server bin/cmd directory for server-side data processing.

There are a number of workflow environments available to geoscientists, e.g., Kepler (Altintas et al., 2004), Taverna (Oinn et al., 2004), Triana (Harrison, Andrew, 2008), and Pegasus (Deelman et al., 2005). Like iRODS WSO, these workflow systems make trade-offs between power and flexibility. Many enable large-scale, parallel workflow execution on distributed resources, providing users real-time status information on the workflow execution (Vahi et al., 2013). While workflow systems share many similarities, there are also key differences, which can often be subtle, that determine their suitability for addressing particular use cases. We used iRODS WSO in this analysis because our use case required a data processing pipeline consisting of a set of scientist authored scripts that operate on data collections already within iRODS. Future work comparing and contrasting iRODS WSO with other workflow environments for completing this or other use cases relevant to hydrologic modeling would be a useful extension to this research.

3. Use Case Description

The prototype software described in this paper is designed to address a use case where a scientist has created a simulation using the Variable Infiltration Capacity (VIC) model for the Carolinas region of the United States. The model has been calibrated and validated for this region as part of a prior study (Billah et al., 2015) and can be used to address other hydrologic research questions as well. The scientist that created the model has published the model's input and output files on the Web for use by other scientists. A second scientist learns about the model and wishes to use the model's output files to test her own research question about drought impacts on counties within a study region. The scientist is interested in how soil moisture deficit predicted by the model varied for different populated communities within the study region. While this application is analyzing historical events, it would be relatively straight-forward to set up the calibrated model to analyze current conditions and to identify populated regions vulnerable to drought conditions within the region. Such information would be valuable to resource managers in better understanding the severity of the drought and its impact on population centers within the region.

The second scientist downloads the model output files published online by the first scientist and creates the visualization by writing her own Python scripts. The scientist downloads the population data for the study counties to a local working directory. The VIC soil moisture outputs are organized in a set of “flux files,” one for each node in the modeling domain. The Python scripts sort through these data extracting relevant information and summarizing the soil moisture time series. Geospatial processing tools are used to relate the coordinates of the model nodes to counties in the study region. The result of this data processing is a comma separated values (CSV) file with the soil moisture deficit and population for each of the five counties. Finally, the scientist programs the Python script to use this CSV file to produce a publication-ready figure for visualizing the drought impacts.

In addition to publishing the scripts and data files from this analysis on a public data repository, which is now a relatively straight-forward exercise given the proliferation of online data repositories, the scientist also wishes to publish the workflow used to perform the analysis as a Web executable resource. The scientist wishes to take this approach for the following reasons.

- Having the overall workflow be executable server-side means the scripts and model output data can be co-located, removing the need to download the large model output file to the scientist’s machine prior to the workflow execution.

- By keeping datasets server-side, it is easier to ensure the data has not been modified after making a local copy (its provenance can be proven). With the ability to publish the model and reference data once, and to keep them on the server, only the visualization results need to be retrieved and published for subsequent runs.

- Having server-side execution of the workflow controls for potential variability across different hardware and software configurations on a client machine. Even with this relatively simple use case of creating a figure, there is potential for different operating systems and versions of analysis software to result in differences in the end product. These software dependencies could result in additional time for scientists to trouble shoot errors. More critically, these dependencies could result in an end product without errors or warnings, but with inconsistencies due to non-breaking differences between dependent software versions.

Simply put, having data and processing co-located on a server as a Web executable resource results in a more controlled environment, which is critical for reproducibility.

The scientist uses iRODS WSO to create the Web executable resource. As part of the WSO, the scientist defines the steps to automatically stage-in the required VIC output and population data that are stored in iRODS collections. The population data comes from TerraPop, which provides global-scale data sets that focus on human population characteristics, land use, land cover, and climate change (Minnesota Population Center, 2013). The Terra Populus data access system was used to create customized data extracts, combining variables from multiple sources into a single package. Users can browse the TerraPop collection and select the required variables; the variable required in this paper was the total population for each county in the United States. After submitting our data request, the system generated a data package that included a shapefile for all the counties in the United States, with unique GEOID identifiers, and a CSV file that includes the GEOID and name of each county (Figure 2). This data package was then automatically uploaded onto the TerraPop grid as an iRODS collection. By federating the DFC-hydrology and TerraPop zones and configuring authorizations, we are able to have the population data remain on the TerraPop server and be automatically staged-in for use by the WSO.

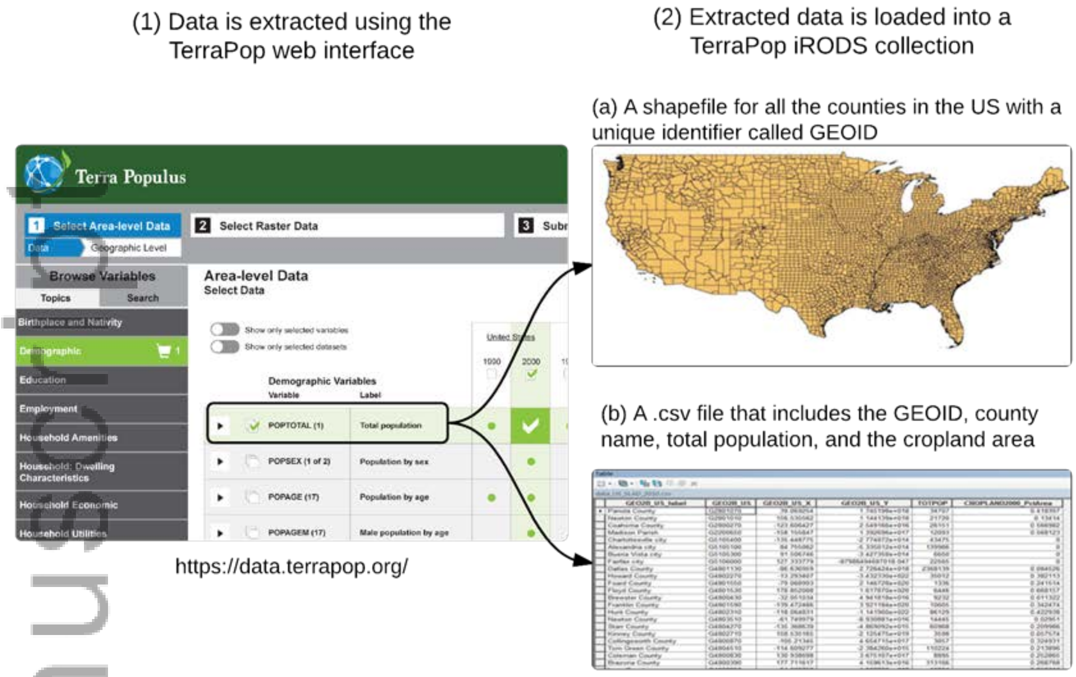

(1) Data is extracted using the TerraPop web interface
(2) Extracted data is loaded into a TerraPop iRODS collection
(a) A shapefile for all the counties in the US with a unique identifier called GEOID
(b) A .csv file that includes the GEOID, county name, total population, and the cropland area
Terra Populus
1 Select Area-level Data
2 Select Raster Data
3 Subr
Browse Variables
Topics
Search
Birthplace and Nativity
Demographic
Education
Employment
Household Amenities
Household: Dwelling Characteristics
Household Economic
Household Utilities
Area-level Data
Select Data
Show only selected variables
Show only selected datasets
Demographic Variables
Variable
Label
POPTOTAL (1)
Total population
POPSEX (1 of 2)
Population by sex
POPAGE (17)
Population by age
POPAGEM (17)
Male population by age
United States
1990
2000
https://data.terrapop.org/

Figure 2. Details on how the county-level population data is requested and extracted using the TerraPop web interface into an iRODS data collection. From this collection, iRODS stages-in the required files prior to the workflow execution.

Finally, the data (including code) resulting from the analysis are published using products provided by the Sustainable Environment-Actionable Data (SEAD) project (Myers et al., 2015). The SEAD project supports publication, preservation, and sharing of data generated by scientists including data generated by running models. Using SEAD, teams of researchers can upload, share, annotate, and review input datasets and model outputs within an access-controlled Project Space, and then formally publish collections of data with associated metadata and provenance for long-term preservation (generating a Digital Object Identifier (DOI) and standards-based archival package, and registering the data with the DataONE catalog for discovery). Our use of SEAD included manual entry of data and metadata via a web interface and bulk uploads of files and programmatic submission of the output figure with metadata to SEAD, which leveraged SEAD's RESTful Web API.

4. Prototype Software Design and Implementation

We present the prototype software aimed at addressing the use case by first describing the steps taken to configure the server-side software and data, next describing the steps required to configure the WSO, then describing the steps required to execute the WSO from the client machine, and concluding with a summary of the results from executing the workflow.

4.1. Server-Side Configuration

To perform the server-side configuration, we first installed iRODS resource server version 4.0 software on an Elastic Cloud Computing (EC2) instance in the Amazon Web Services (AWS) cloud. We chose AWS because it provides on-demand computing resources and services that can be easily scaled to meet demands. The EC2 service provided through AWS allows users to rent virtual machines (instances) with different capabilities and pay by the CPU hour. For prototyping purposes, we used a Linux-based medium sized machine (m3) with 3.75 GB of memory, 4 vCPU, 15 GB of SSD-based local instance storage, and 64-bit platform for the iRODS resource server ("Amazon EC2 Instances," 2015). Next this new iRODS resource server was configured to be part of the DFC-hydrology zone that has its iRODS Metadata Catalog (iCAT) server on a machine running at RENCI. We had to configure the AWS EC2 instance to be associated with an elastic IP address to avoid having to update the EC2 instance's IP addresses in the iCAT server following each restart of the EC2 instance.

We then developed a WSO on the iRODS resource server to implement the data visualization workflow described in the use case. This required that the user have an account on the server itself with read/write access to the cmd directory (Figure 1b). It was also necessary to set read/execute rights on the files associated with the WSO so that they could be executed by the iRODS user account. We uploaded to the iRODS resource server the VIC model output files from SEAD (where the original scientist had published them for use by the community), the Python scripts created by the scientist to generate the visualization, and the shell script, also created by the scientist, used to sequence the execution of the Python scripts on the iRODS resource server. The VIC source code is not included in SEAD because the source code is available from the developer's GitHub page instead (see https://github.com/UW-Hydro/VIC).

4.2. Client-Side Configuration

The client machine can be any computer with the iRODS client software installed. In this prototyping work, we used a second EC2 instance as the client machine simply to avoid moving data into and out of the AWS cloud. We installed the icommands iRODS client software library on the client machine. The icommands software includes a set of commands that perform operations such as make a new directory (*imkdir*) or put a file into an iRODS collection (*iput*) (Weise et al., 2008). The icommands client library includes an environment configuration file that is used to point to a particular iRODS zone and set default user credentials for accessing the iRODS zone. In our case, we configured the icommands environment to operate on the DFC-hydrology zone and entered user credentials representing the scientist accessing the system.

The general file structure required for creating a WSO was described in Section 2 and in Figure 1a. For our particular application, we first created a workflow file (PopVsSm.mss) that specifies the steps required to execute the workflow. The workflow file simply specified that the workflow should execute the scientist-authored shell script installed on the iRODS server cmd directory. We put the PopVsSm.mss file into an iRODS collection and then made a new collection named “vic_soilmositure.” We mounted this new collection, effectively making it a WSO.

4.3. Executing the Workflow

Once the WSO is mounted, it is then possible to execute the workflow. This process is described in general in Section 2. Here we provide specifics of the WSO execution for the use case. The general flow of data and sequence of commands for executing the WSO execution for the use case is described in Figure 3.

(1) The user initiates execution of the workflow by issuing an *iget* command on the PopVsSm.run file that is in the mounted WSO collection. The PopVsSm.mpf parameter file defines the data required by the workflow and stages these files from different iRODS collections into the directory on the iRODS resource server where the WSO is executed. In our case, we staged-in the VIC model output data stored in the DFC-hydrology grid and county-level

population data from the TerraPop grid. While these two datasets are stored within different grids, it is possible to gain access to the data directly using iRODS authentication because the grids are federated.

(2) Once all required data is staged into the iRODS resource server directory where the workflow is executed, the workflow file specifies that the scientist-authored shell script stored on the iRODS server should be executed. This shell script then calls a series of scientist-authored Python scripts that process the staged-in data to create the output figure.

(3) A final step in the shell script is publishing the figure resulting from the workflow automatically to a SEAD project space for sharing with colleagues and subsequent publication. The SEAD API is used for this purpose and allows for the submission of the file along with associated metadata to a SEAD project space.

(4) Upon completion of the workflow, key output data are staged-out into iRODS collections according to specifications in the parameter file. This allows the files to be accessible to authorized users in the grid.

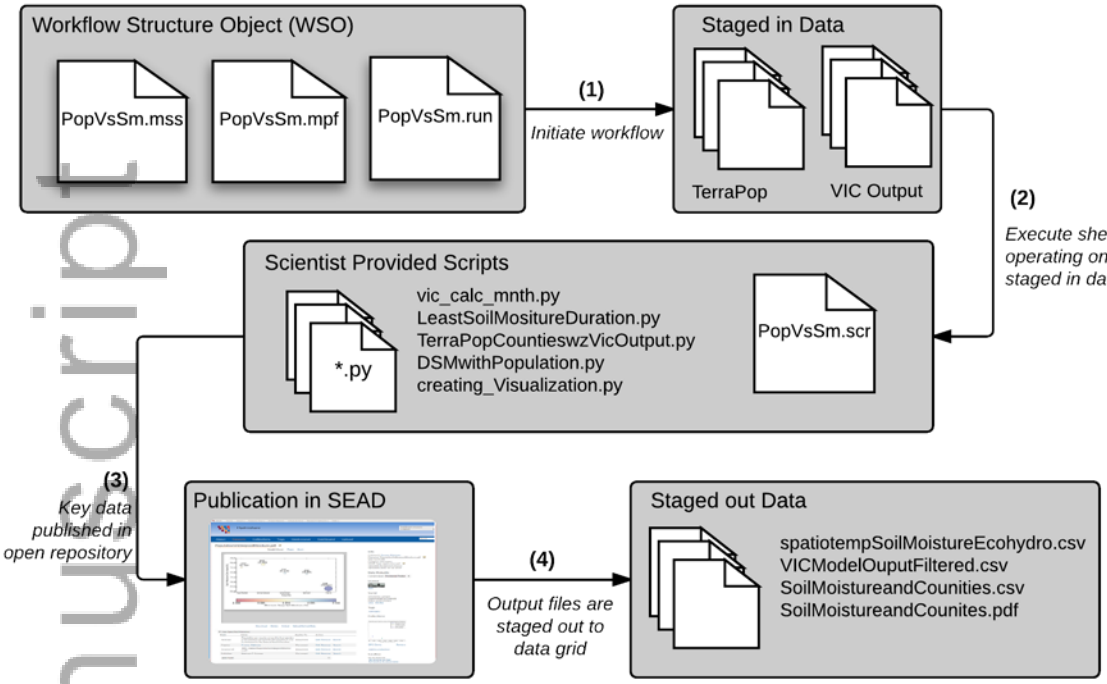
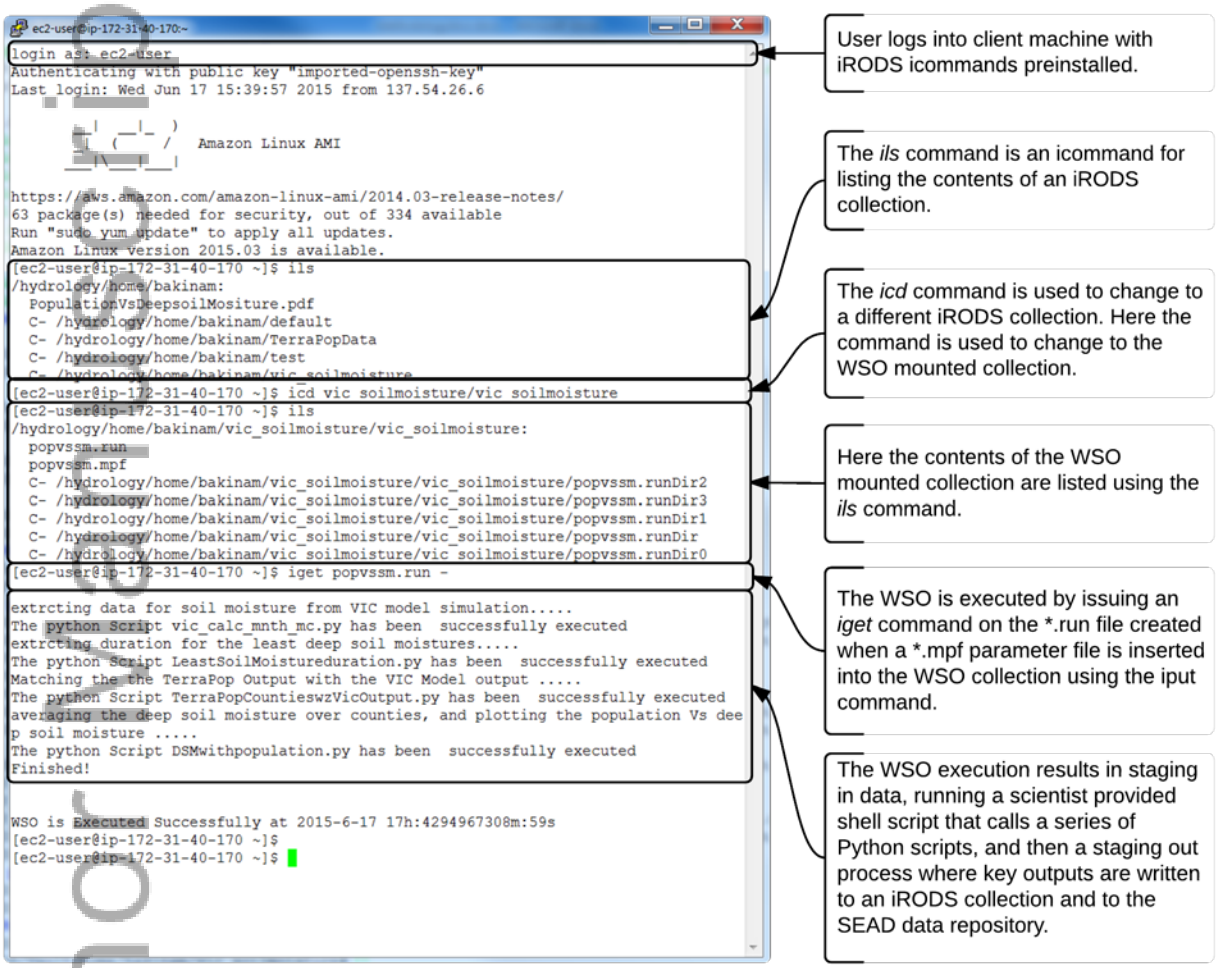

Workflow Structure Object (WSO)
PopVsSm.mss
PopVsSm.mpf
PopVsSm.run
(1)
Initiate workflow
Staged in Data
TerraPop
VIC Output
(2)
Execute shell script operating on staged in data
Scientist Provided Scripts
*.py
vic_calc_mnth.py
LeastSoilMositureDuration.py
TerraPopCountieswzVicOutput.py
DSMwithPopulation.py
creating_Visualization.py
PopVsSm.scr
(3)
Key data published in open repository
Publication in SEAD
(4)
Output files are staged out to data grid
Staged out Data
spatiotempSoilMoistureEcohydro.csv
VICModelOuputFiltered.csv
SoilMoistureandCounities.csv
SoilMoistureandCounites.pdf

Figure 3. The steps that occur on the server-side when a user executes the WSO. Data is staged-in from iRODS collections, scientist-authored scripts are run to create the figure, data is published through a SEAD project space using the SEAD API, and key output data is staged-out back into iRODS collections.

Figure 4 shows the steps for executing a WSO from a user's perspective when working with the icommands client library. The user must know which iRODS collection contains the script files required for executing the WSO to be able to execute it. Once the user has logged into the client machine, the user changes the working directory to the iRODS logical path where the WSO has been mounted. In this case, the WSO was mounted as the "vic_soilmoisture" collection. The user next issues an *iput* command to put the parameter file (PopVsSm.mpf) into the mounted WSO. This step is not illustrated in Figure 3 for brevity, but results in the generation of a run file (popvssm.run) in the collection. Finally, the client executes the workflow by issuing an *iget* command on the popvssm.run file.



```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Wed Jun 17 15:39:57 2015 from 137.54.26.6

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2014.03-release-notes/
63 package(s) needed for security, out of 334 available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2015.03 is available.
[ec2-user@ip-172-31-40-170 ~]$ ils
/hydrology/home/bakinam:
  PopulationVsDeepsoilMositure.pdf
  C- /hydrology/home/bakinam/default
  C- /hydrology/home/bakinam/TerraPopData
  C- /hydrology/home/bakinam/test
  C- /hydrology/home/bakinam/vic_soilmoisture
[ec2-user@ip-172-31-40-170 ~]$ icd vic soilmoisture/vic soilmoisture
[ec2-user@ip-172-31-40-170 ~]$ ils
/hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture:
  popvssm.run
  popvssm.mpf
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir2
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir3
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir1
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir0
[ec2-user@ip-172-31-40-170 ~]$ iget popvssm.run -

extrcting data for soil moisture from VIC model simulation.....
The python Script vic_calc_mnth_mc.py has been  successfully executed
extrcting duration for the least deep soil moistures.....
The python Script LeastSoilMoistureduration.py has been  successfully executed
Matching the the TerraPop Output with the VIC Model output .....
The python Script TerraPopCountieswzVicOutput.py has been  successfully executed
averaging the deep soil moisture over counties, and plotting the population Vs dee
p soil moisture .....
The python Script DSMwithpopulation.py has been  successfully executed
Finished!


WSO is Executed Successfully at 2015-6-17 17h:4294967308m:59s
[ec2-user@ip-172-31-40-170 ~]$
[ec2-user@ip-172-31-40-170 ~]$
```

User logs into client machine with iRODS icommands preinstalled.

The *ils* command is an icommand for listing the contents of an iRODS collection.

The *icd* command is used to change to a different iRODS collection. Here the command is used to change to the WSO mounted collection.

Here the contents of the WSO mounted collection are listed using the *ils* command.

The WSO is executed by issuing an *iget* command on the *.run file created when a *.mpf parameter file is inserted into the WSO collection using the iput command.

The WSO execution results in staging in data, running a scientist provided shell script that calls a series of Python scripts, and then a staging out process where key outputs are written to an iRODS collection and to the SEAD data repository.

Figure 4. The steps required from a client machine in order to execute the WSO using the icommands client library.

4.4. Results from the Workflow Execution

When the workflow is executed, the output messages are written to the console, although all computation is performed on the server-side and no data (other than the output messages) are transferred to the client machine. Once the workflow execution has completed, the user can access the output collection called runDir resulting from the workflow execution. The runDir file contains by default the stdout from the execution of the workflow along with any staged-in and derived data from the workflow ("Workflow Objects (WSO)," 2013).

The workflow also results in publication of the workflow results to a SEAD project space. Figure 5 shows the data collections as they appear through the SEAD project space website. Most data were uploaded using the SEAD web interface. Figure 6 shows the figure resulting from the WSO execution that was automatically written to the SEAD project space using the SEAD API as a final step in the WSO execution.

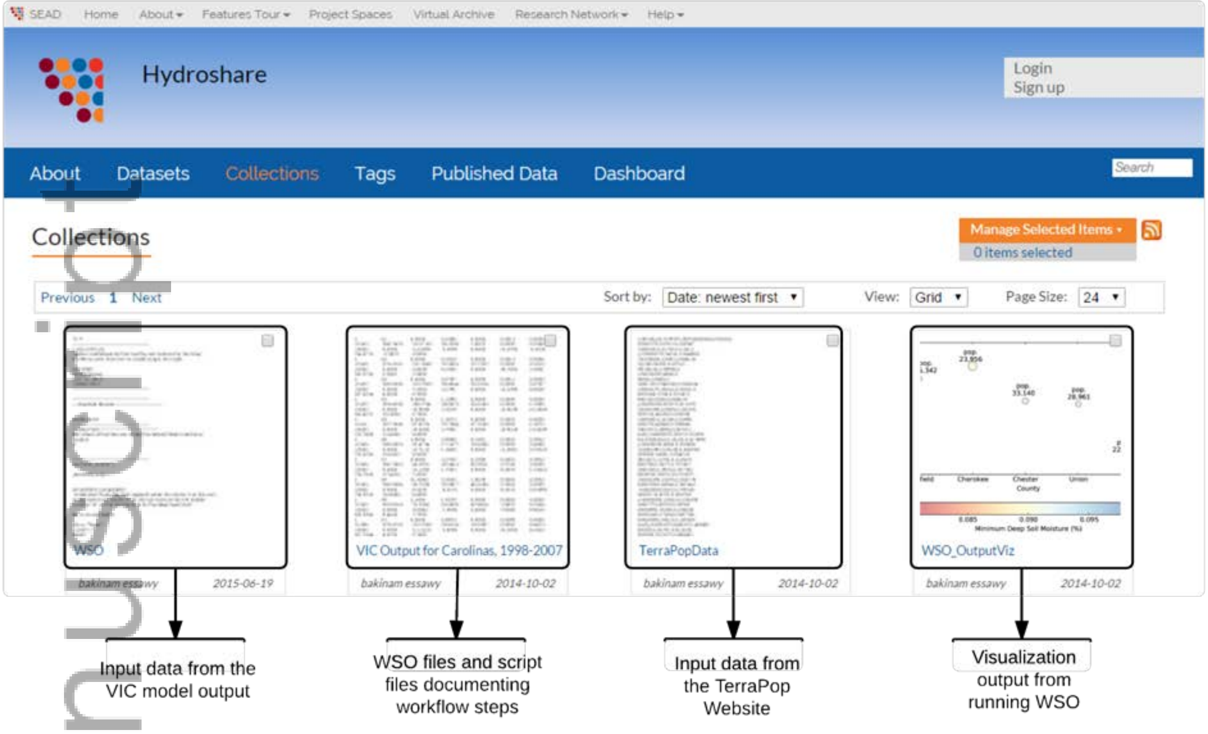
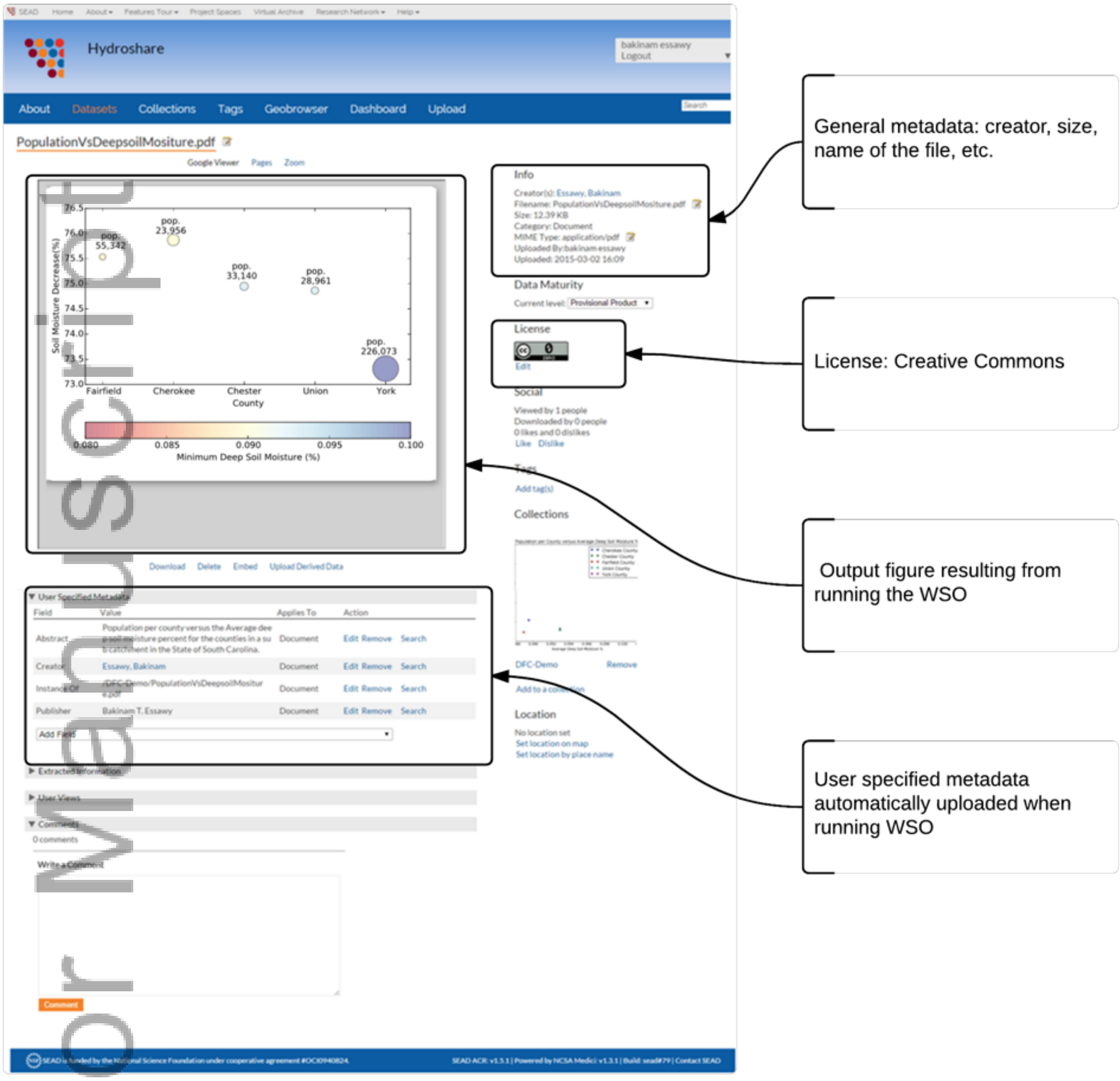

SEAD
Home
About
Features Tour
Project Spaces
Virtual Archive
Research Network
Help
Hydroshare
Login
Sign up
About
Datasets
Collections
Tags
Published Data
Dashboard
Search
Collections
Manage Selected Items
0 items selected
Previous 1 Next
Sort by: Date: newest first
View: Grid
Page Size: 24
WSO
bakinam essawy 2015-06-19
VIC Output for Carolinas, 1998-2007
bakinam essawy 2014-10-02
TerraPopData
bakinam essawy 2014-10-02
WSO_OutputViz
bakinam essawy 2014-10-02
Input data from the VIC model output
WSO files and script files documenting workflow steps
Input data from the TerraPop Website
Visualization output from running WSO

Figure 5: Contents of the Sustainable Environment Actionable Data (SEAD) project space used for storing and accessing data used in the workflow.



PopulationVsDeepsoilMositure.pdf

Info

Creator(s): Essawy, Bakinam
Filename: PopulationVsDeepsoilMositure.pdf
Size: 12.39 KB
Category: Document
MIME Type: application/pdf
Uploaded By:bakinam essawy
Uploaded: 2015-03-02 16:09

Viewed by 1 people
Downloaded by 0 people
0 likes and 0 dislikes

No location set

User Specified Metadata

Field	Value	Applies To	Action
Abstract	Population per county versus the Average deep soil moisture percent for the counties in a sub catchment in the State of South Carolina.	Document	Edit Remove Search
Creator	Essawy, Bakinam	Document	Edit Remove Search
Instance Of	/DFC-Demo/PopulationVsDeepsoilMositure.pdf	Document	Edit Remove Search
Publisher	Bakinam T. Essawy	Document	Edit Remove Search

Add Field

0 comments

SEAD is funded by the National Science Foundation under cooperative agreement #OCI0940824.

SEAD ACR: v1.5.1 | Powered by NCSA Medici: v1.3.1 | Build: sead#79 | Contact SEAD

General metadata: creator, size, name of the file, etc.

License: Creative Commons

Output figure resulting from running the WSO

User specified metadata automatically uploaded when running WSO

Figure 6: View of figure, produced by executing the WSO, within the SEAD project space. The workflow uses the SEAD API to upload this resource along with metadata to the SEAD project space.

5. Discussion

5.1. Reproducibility

To support transparency and reproducibility of this work as envisioned by the Geoscience Paper of the Future (GPF) project, the data collections in the use case (e.g., the VIC output files, the TerraPop data, the WSO files, and the output figure) were published in SEAD. As part of this publication process, each collection was given metadata including a brief abstract, creators, the publisher, and then published to generate a Digital Object Identifier (DOI) (Table 1). The output figure resulting from the WSO execution was first written to a SEAD project space along with basic metadata as a final step in the WSO execution using the SEAD API. From there, the scientist logged into the SEAD web interface and set additional metadata fields to publish the resource with an assigned DOI. Any combination of automated and manual entry is supported and researchers can choose which data to publish. In our case, we automatically captured outputs from multiple test runs before manually selecting, annotating, and publishing (including creating a DOI for) only the final run.

Use of an open, metadata-aware repository makes it simple to capture additional derived data and provenance information as research continues. By publishing the reference data, scripts, and output data separately in SEAD, we also demonstrate the ability for larger reference data to be published once, and then referenced via provenance links from the derived output files that could be generated by many researchers over time. For example, the VIC output files used in this workflow may be used in other research studies. If each publication using these VIC output files references its DOI, it will be possible to track the impact of the model output files through citation counts similar to what is done now for tracking citation counts of research papers.

Other end-points could be used for publishing key digital assets from the WSO workflows. For example, the Consortium of Universities for the Advancement of Hydrologic Sciences, Inc. (CUAHSI) HydroShare system is in development and could serve as an alternative or secondary end-point for publishing results with more discipline-specific metadata (Horsburgh et al., 2015; Morsy et al., 2014; Tarboton et al., 2014), as could systems such as FigShare or Zenodo. We anticipate a growing number of such repositories and for federation between them (e.g., SEAD is already a member node in DataOne (Michener et al., 2012), advertising our WSO publications through DataONE’s catalog). This research shows how iRODS WSO could play an important role in moving data resources within such data repositories to and from computational resources to support data computation use cases.

Table 1: Key digital assets used in the study that are published through SEAD with basic metadata.

Title	**DOI**	**Author**	**Contact**	**Abstract**	**License**
TerraPop Data Extract	10.5967/M08P5XH5	Essawy, Bakinam	Goodall, Jonathan	Population data extracted from TerraPop (https://data.terrapop.org) for the study region.	Creative Commons (CC)
VIC Output for Carolina, 1998-2007	10.5967/M0DF6P6F	Essawy, Bakinam	Goodall, Jonathan	Output from a VIC model for the Carolinas, USA calibrated for the period 1998-2007 to study drought impacts.	Creative Commons (CC)
WSO	10.5967/M0J6	Essawy,	Goodall,	The scripts and related files	Creative

	7DXR	Bakinam	Jonathan	used to create the iRODS Workflow Structured Object (WSO).	Commons (CC)
WSO_OutputViz	10.5967/M0513W51	Essawy, Bakinam	Goodall, Jonathan	Impact of 2007 drought on five counties in the study region.	Creative Commons (CC)

Using a public cloud offers further opportunities for reproducibility. It is possible to quickly set up virtual machines (VMs) with a variety of operating systems to reproduce computational analyses. It is also possible to capture images of VM instances that can be stored for future reproducibility. Exploring the use of virtual containers (e.g., the Docker project) rather than VMs would be a useful extension to this work. Virtual containers can reduce set up time and storage costs compared to VMs for software, like what was used in this work, that run in a Linux operating system.

5.2. Federation

Federation across cyberinfrastructure systems is a key aspect of this work. Federation describes how distinct and formally disconnected systems interoperate. There is a growing set of cyberinfrastructure systems available to scientists, and many studies will benefit from the use of more than one of these systems. Effective ways for federating across these systems will result in powerful tools that save scientists' time and encourage reproducibility through automatic data transfers handled directly by systems. This concept was illustrated in our study by showing how distinct cyberinfrastructure systems can be federated and used collectively within a single workflow execution.

Figure 7 provides a depiction of the workflow that emphasizes different data collections and approaches for federating between DFC, TerraPop, and SEAD. The use case in this study represents two levels of federation that we believe are relevant for most scientific studies. The federation between the AWS machine where the workflow was executed and the TerraPop reference data is what we term a strong federation, while the federation between the AWS machine and SEAD is what we term a weak federation. A strong federation is based on a strong trust model where one data grid administrator can add credentials of users of other data grid, and grant access to resources based on authentication through other data grids. One primary benefit of this level of federation is that data grid technology can be used to transfer files between the two systems. For large files, this level of federation will be important because of the functionality provided by data grids like iRODS that are designed specifically to ensure rapid and successful transfer of large files over a network. Weak federation, based on federation through Web service APIs, allows for greater flexibility and less required trust between systems, because all operations are through services. Transferring large data through Web services, however, it not ideal for the reasons we outlined in Section 2.

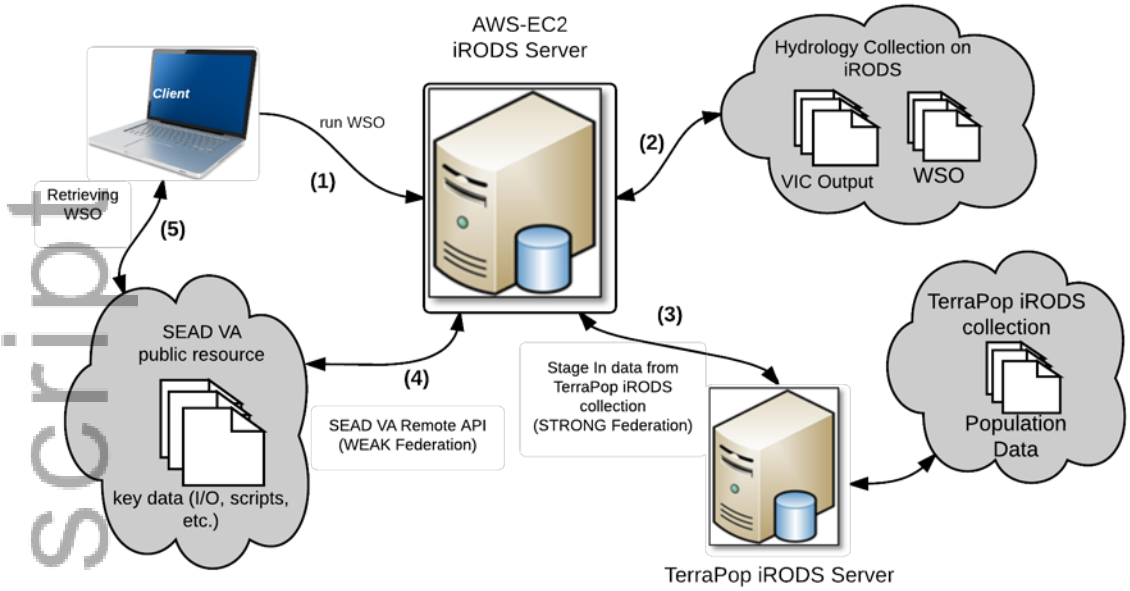

AWS-EC2
iRODS Server
Client
run WSO
(1)
Retrieving WSO
(5)
Hydrology Collection on iRODS
VIC Output
WSO
(2)
(3)
TerraPop iRODS collection
Population Data
SEAD VA public resource
key data (I/O, scripts, etc.)
(4)
SEAD VA Remote API (WEAK Federation)
Stage In data from TerraPop iRODS collection (STRONG Federation)
TerraPop iRODS Server

Figure 7. Main components and data flow in the workflow emphasizing data collections and federation approaches

5.3. Adoption

While there are many advantages to the approach described in this paper, there are also important barriers to adoption, especially in terms of the current prototype system. Currently, users of the system need to be familiar with an iRODS client (e.g., the icommands client library used in this study). They must also be aware of steps for executing a WSO. Developers need an understanding of how to structure new WSOs and will need access to the server running the iRODS resource server software for installation and configuration of the WSO.

There are opportunities for abstracting the complexity of directly interfacing with iRODS WSO for end users in order to encourage broader adoption of the technology. One way to do this would be to have someone familiar with iRODS WSO take input from the scientist including the scripts needed to execute the workflow and the location (iRODS logical path name) of the input data for the scripts. The administrator would then mount a WSO with an example parameter file and make it available through the system to end-users. The user could then execute the workflow either using the icommands client library, as described in the paper, or through other tailored client applications able to operate on iRODS collections including executing WSOs stored within iRODS collections. We believe this would be a fairly straightforward process for moving scientist-authored codes into a form that is Web-executable.

5.4. Data Size and Heterogeneity Challenges

This work only begins to illustrate the potential benefit of using data grid technology for executing workflows that require heterogeneous data from distributed data sources. We showed how WSOs allow for automatically staging-in of required data distributed across a data grid. We also showed how data produced from the workflow can be staged-out, meaning written to collections in the data grid where it can be accessible to other users. While it was not demonstrated in this use case, one can execute a distributed workflow across the network on multiple iRODS resource server using WSO.

This approach allows the location of the input and output files for a computational tool to be independent of the location where the processing is done. However, unlike approaches that rely only on Web service APIs for data staging prior to workflow execution, iRODS provides a more robust data staging approach that leverages grid technology. While the use case demonstrated the concept using fairly small file sizes, the solution we used can be applied to larger terabyte scale data as well. Given that modeling in many geoscience disciplines requires access to large, distributed data, data grid technology provides a powerful way for data staging associated with workflow execution.

6. Conclusions

The focus of this paper is on creating scientist-authored workflows as Web-executable resources in data grids. The iRODS WSO provides researchers with the ability to publish their research methods for computational studies as workflows that specify the tools, data, and sequence of steps taken to complete the study. All of these digital objects (data, software, model outputs, etc.) can be made accessible to other users of the data grid as well as to non-grid users through publication in SEAD.

There are many challenges in reaching the ultimate goal of reproducibility, especially when dealing with data-intensive modeling analyses that require a large, diverse set of input data and generate a large, diverse set of output data. Through this work, we argue that reproducibility will require more server-side data processing, where reference data is managed along with the model itself, than what is common now. This is due to the large and increasing size of datasets used by geoscientists, and the growing complexity of software and software dependencies that require constrained environments to ensure reproducibility.

We also argue for multiple federation approaches as means for providing interoperability across the variety of cyberinfrastructure systems needed for data access, analysis, modeling, and publication services. Federation approaches most often used in geoscience disciplines emphasize Web service APIs, however to support large datasets, the community should have broader adoption of data grid federation approaches as well. The use of both approaches was demonstrated for a use case that leveraged four federated but heterogeneous cyberinfrastructure systems: DFC, TerraPop, and SEAD, and via an existing connection with SEAD, DataONE.

Any approach for making scientific computations into Web-executable resources must have a low barrier to entry for users. We have proposed an approach that allows scientists to write scripts as is typically done now for data analysis using languages familiar to scientists, and then making these scripts available as Web-executable resources to scientists using iRODS WSO technology. Future work should explore embedding of iRODS WSOs into systems that include tailored interfaces for scientific communities. Then, rather than the steps described in the paper for executing WSO that include the use of the icommand client library, the end user could have a more tailored interface for viewing and executing workflows that abstracts technical details from the end user.

There are encouraging trends toward increased publication of data (including code) used in scientific studies. It is important that the momentum behind these trends result in scripts and workflows as Web-executable resources to capture their full potential in advancing reproducibility goals. The advantages of Web executable resources include the increased ability to share, reproduce, and collaborate on scientists-authored workflows. While the potential of scientific scripts and workflows as Web executable resources is clear, important issues remain related to managing large data and computation collections. We have demonstrated here an approach using data grids for addressing this challenge, and have argued for moving processing to reference data stored within data grids as a method for creating reproducible scientific workflows on large datasets.

Acknowledgements

This work was supported by the National Science Foundation (NSF) under award numbers ACI-0940841, ACI-0940824, and ACI-0940818, and by Amazon Web Services (AWS) through an Education Research Grant award. This research would not have been possible without assistance from the larger iRODS, DFC, SEAD, and TerraPop teams. The data used are listed in Table 1 and can be found in the SEAD repository at the DOIs provided in Table 1.

References

Acharya, A., Uysal, M., Saltz, J., 1998. Active disks: programming model, algorithms and evaluation. SIGPLAN Not. 33, 81–91. doi:10.1145/291006.291026

Allcock, B., Bester, J., Bresnahan, J., Chervenak, A.L., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnel, D., Tuecke, S., 2002. Data management and transfer in high-performance computational grid environments. Parallel Comput. 28, 749–771. doi:10.1016/S0167-8191(02)00094-7

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludäscher, B., Mock, S., 2004. Kepler □: An Extensible System for Design and Execution of Scientific Workflows, in: 16th International Conference On. IEEE, 2004. pp. pp. 423–424. doi:10.1109/SSDM.2004.1311241

Amazon EC2 Instances [WWW Document], 2015. URL http://aws.amazon.com/ec2/instance-types/ (accessed 6.7.15).

Anderson, S.P., Bales, R.C., Duffy, C.J., 2008. Critical Zone Observatories: Building a network to advance interdisciplinary study of Earth surface processes. Mineral. Mag. 72, 7–10. doi:10.1180/minmag.2008.072.1.7

Billah, M.M., Goodall, J.L., Narayan, U., Reager, J.T., Lakshmi, V., Famiglietti, J.S., 2015. A methodology for evaluating evapotranspiration estimates at the watershed-scale using GRACE. J. Hydrol. 523, 574–586. doi:10.1016/j.jhydrol.2015.01.066

Cornillon, P., Gallagher, J., Sgouros, T., 2003. OPeNDAP: Accessing data in a distributed, heterogeneous environment. Data Sci. J. 2, 164–174. doi:10.2481/dsj.2.164

Cowles, T., Delaney, J., Orcutt, J., Weller, R., 2010. The Ocean Observatories Initiative: Sustained Ocean Observing Across a Range of Spatial Scales. Mar. Technol. Soc. 44(6), 54–64. doi:http://dx.doi.org/10.4031/MTSJ.44.6.21

De Roure, D., Goble, C., Stevens, R., 2009. The design and realisation of the Virtual Research Environment for social sharing of workflows. Futur. Gener. Comput. Syst. 25, 561–567. doi:10.1016/j.future.2008.06.010

Deelman, E., Singh, G., Su, M., Blythe, J., Gil, Y., Kesselman, C., 2005. Pegasus □: A framework for mapping complex scientific workflows onto distributed systems. Sci. Program. 13, 219–237.

Dunlap, R., Mark, L., Rugaber, S., Balaji, V., Chastang, J., Cinquini, L., DeLuca, C., Middleton, D., Murphy, S., 2008. Earth system curator: metadata infrastructure for climate modeling. Earth Sci. Informatics 1, 131–149. doi:10.1007/s12145-008-0016-1

Foster, I., 2011. Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. IEEE Comput. Soc. 15, 70–73. doi:doi:10.1109/MIC.2011.64

Harrison, Andrew, et al, 2008. WS-RF workflow in Triana. Int. J. High Perform. Comput. Appl. 22, 268–283. doi:10.1177/1094342007086226

Horsburgh, J.S., Morsy, M.M., Castronova, A.M., Goodall, J.L., Gan, T., Yi, H., Stealey, M.J., Tarboton, D.G., 2015. Hydroshare: Sharing Diverse Environmental Data Types and Models as Social Objects with Application to the Hydrology Domain. JAWRA J. Am. Water Resour. Assoc. n/a–n/a. doi:10.1111/1752-1688.12363

Introduction to Workflow as Objects [WWW Document], 2012. URL https://wiki.irods.org/index.php/Introduction_to_Workflow_as_Objects (accessed 6.7.2015).

Keller, M., Schimel, D.S., Hargrove, W.W., Hoffman, F.M., 2008. A continental strategy for the National Ecological Observatory Network. Front. Ecol. Environ. 6, 282–284. doi:10.1890/1540-9295(2008)6[282:ACSFTN]2.0.CO;2

Kyriazis, D., Tserpes, K., Kousiouris, G., Menychtas, A., Varvarigou, T., 2008. Data Aggregation and Analysis -based A Grid approach for Medicine and Biology. Int. Symp. on. IEEE 841–848.

Liang, X., Lettenmaier, D.P., 1994. A simple hydrologically based model of land surface water and energy fluxes for general circulation models. J. Geophys. Res. 99, 14,415–14,428. doi:10.1029/94JD00483

Maidment, D.R., 2008. Bringing Water Data Together. J. Water Resour. Plan. Manag. 134, 95–96.

Michener, W.K., Allard, S., Budden, A., Cook, R.B., Douglass, K., Frame, M., Kelling, S., Koskela, R., Tenopir, C., Vieglais, D.A., 2012. Participatory design of DataONE—Enabling cyberinfrastructure for the biological and environmental sciences. Ecol. Inform. 11, 5–15. doi:10.1016/j.ecoinf.2011.08.007

Minnesota Population Center, 2013. Terra Populus: Beta Version [Machine-readable database]. Minneapolis: University of Minnesota.

Morsy, M.M., Goodall, J.L., Bandaragoda, C., Castronova, A.M., Greenberg, J., 2014. Metadata for Describing Water Models, in: International Environmental Modelling and Software Society (iEMSs) 7th International Congress on Environmental Modelling and Software. doi:10.13140/2.1.1314.6561

Myers, J., Hedstrom, M., Akmon, D., Payette, S., Plale, B.A., Kouper, I., McCaulay, S., McDonald, R., Suriarachchi, I., Varadharaju, A., Kumar, P., Elag, M., Lee, J., Kooper, R., Marini, L., 2015. Towards Sustainable Curation and Preservation: The SEAD Project's Data Services Approach. Proc. IEEE 11th Int. e-Science Conf. Munich, Ger. doi:10.1109/eScience.2015.56

Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P., 2004. Taverna : a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 20, 3045–3054. doi:10.1093/bioinformatics/bth361

Rajasekar, A., 2014. Workflows [WWW Document]. 6th Annu. iRODS User Gr. Meet. June 2014 Inst. Quant. Soc. Sci. MA. URL http://irods.org/wp-content/uploads/2014/06/Workflows-iRUGM-2014.pdf (accessed 8.12.15).

Rajasekar, A., Moore, R., Hou, C.-Y., Lee, C. a., Marciano, R., de Torcy, A., Wan, M., Schroeder, W., Chen, S.-Y., Gilbert, L., Tooby, P., Zhu, B., 2010. iRODS Primer: Integrated Rule-Oriented Data System, Synthesis Lectures on Information Concepts, Retrieval, and Services. doi:10.2200/S00233ED1V01Y200912ICR012

Tarboton, D.G., Idaszak, R., Horsburgh, J.S., Heard, J., Ames, D., Goodball, J.L., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., Maidment, D.R., 2014. HydroShare: Advancing Collaboration through Hydrologic Data and Model Sharing, in: Ames, D.P., Quinn, N.W.T., Rizzoli, A.E. (Eds.), International Environmental Modelling and Software Society (iEMSs) 7th International Congress on Environmental Modelling and Software. doi:978-88-9035-744-2

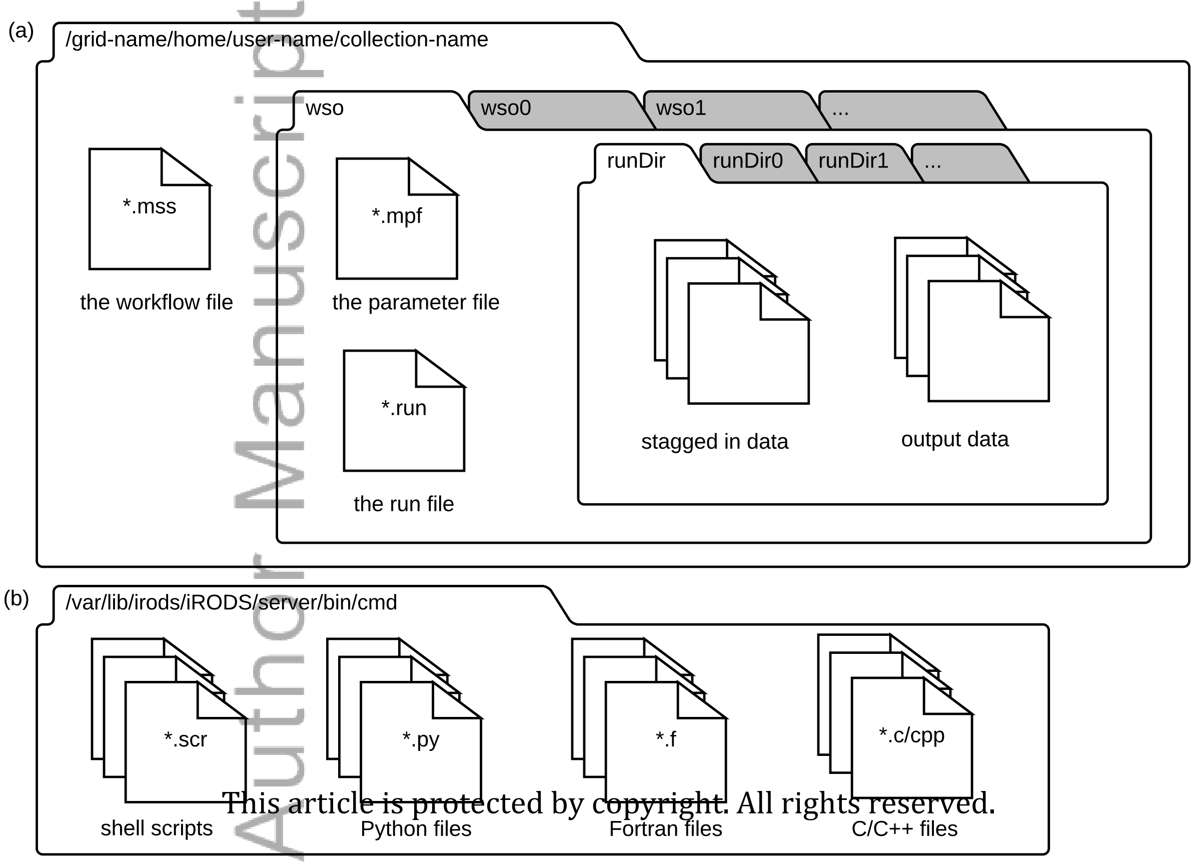
Vahi, K., Harvey, I., Samak, T., Gunter, D., Evans, K., Rogers, D., Taylor, I., Goode, M., Silva, F., Al-shakarchi, E., Mehta, G., Jones, A., Deelman, E., 2013. A General Approach to Real-time Workflow Monitoring, in: A General Approach to Real-Time Workflow Monitoring. In High Performance Computing, Networking, Storage and Analysis (SCC). pp. 108–118. doi:10.1109/SC.Companion.2012.26

Weise, A., Wan, M., Schroeder, W., Hasan, A., 2008. Managing Groups of Files in a Rule Oriented Data Management System (iRODS). Comput. Sci. 321–330. doi:10.1007/978-3-540-69389-5_37

Williams, D.N., Ananthakrishnan, R., Bernholdt, D.E., Bharathi, S., Brown, D., Chen, M., Chervenak, A.L., Cinquini, L., Drach, R., Foster, I.T., Fox, P., Hankin, S., Henson, V.E., Jones, P., Middleton, D.E., Schwidder, J., Schweitzer, R., Schuler, R., Shoshani, A., Siebenlist, F., Sim, A., Strand, W.G., Wilhelmi, N., Su, M., 2008. Data management and analysis for the Earth System Grid. J. Phys. Conf. Ser. 125, 012072. doi:10.1088/1742-6596/125/1/012072

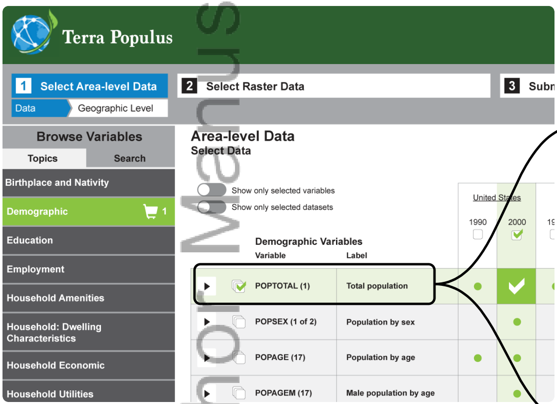
Williams, D.N., Lawrence, B. N. Lautenschlager, M. Middleton, D., Balaji, V., 2011. The Earth System Grid Federation: Delivering globally accessible petascale data for CMIP5, in: Proceedings of the 32nd Asia-Pacific Advanced Network Meeting. pp. 121–130. doi:10.7125/APAN.32.15

Workflow Objects (WSO) [WWW Document], 2013. URL https://wiki.irods.org/index.php/Workflow_Objects_(WSO) (accessed 6.7.15).


(a)
/grid-name/home/user-name/collection-name
wso
wso0
wso1
...
*.mss
the workflow file
*.mpf
the parameter file
*.run
the run file
runDir
runDir0
runDir1
...
stagged in data
output data
(b)
/var/lib/irods/iRODS/server/bin/cmd
*.scr
shell scripts
*.py
Python files
*.f
Fortran files
*.c/cpp
C/C++ files

(1) Data is extracted using the TerraPop web interface

(2) Extracted data is loaded into a TerraPop iRODS collection



1 Select Area-level Data | 2 Select Raster Data | 3 Subn

Data | Geographic Level

Browse Variables

Topics | Search

- Birthplace and Nativity
- Demographic 1
- Education
- Employment
- Household Amenities
- Household: Dwelling Characteristics
- Household Economic
- Household Utilities

Area-level Data
Select Data

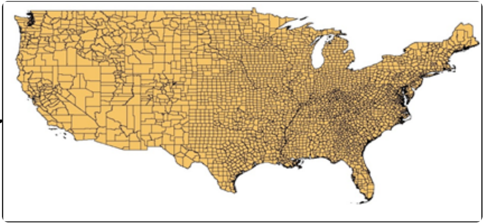
○ Show only selected variables
○ Show only selected datasets

Demographic Variables

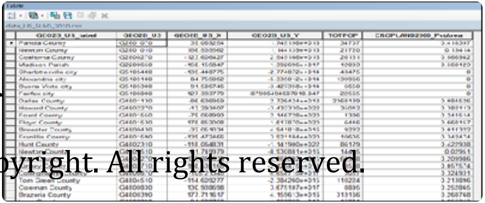
	Variable	Label	United States 1990 □	United States 2000 ☑	19 □
▶	☑ POPTOTAL (1)	Total population	●	✓	●
▶	□ POPSEX (1 of 2)	Population by sex		●	
▶	□ POPAGE (17)	Population by age	●	●	
▶	□ POPAGEM (17)	Male population by age		●	

https://data.terrapop.org/

(a) A shapefile for all the counties in the US with a unique identifier called GEOID

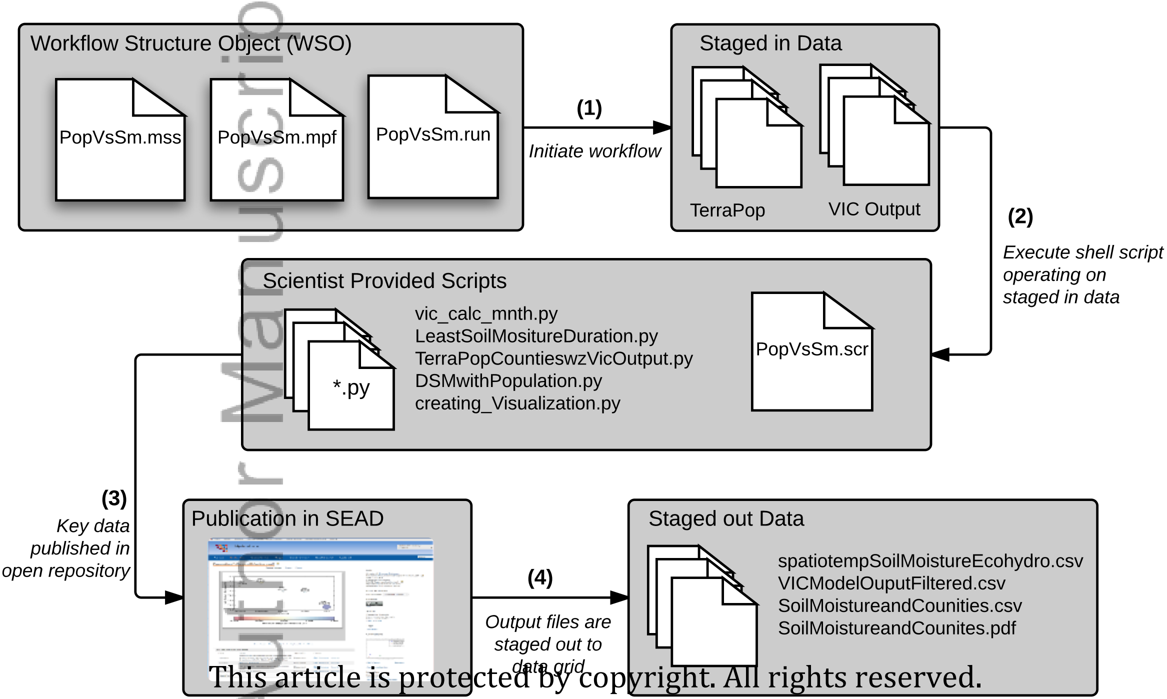


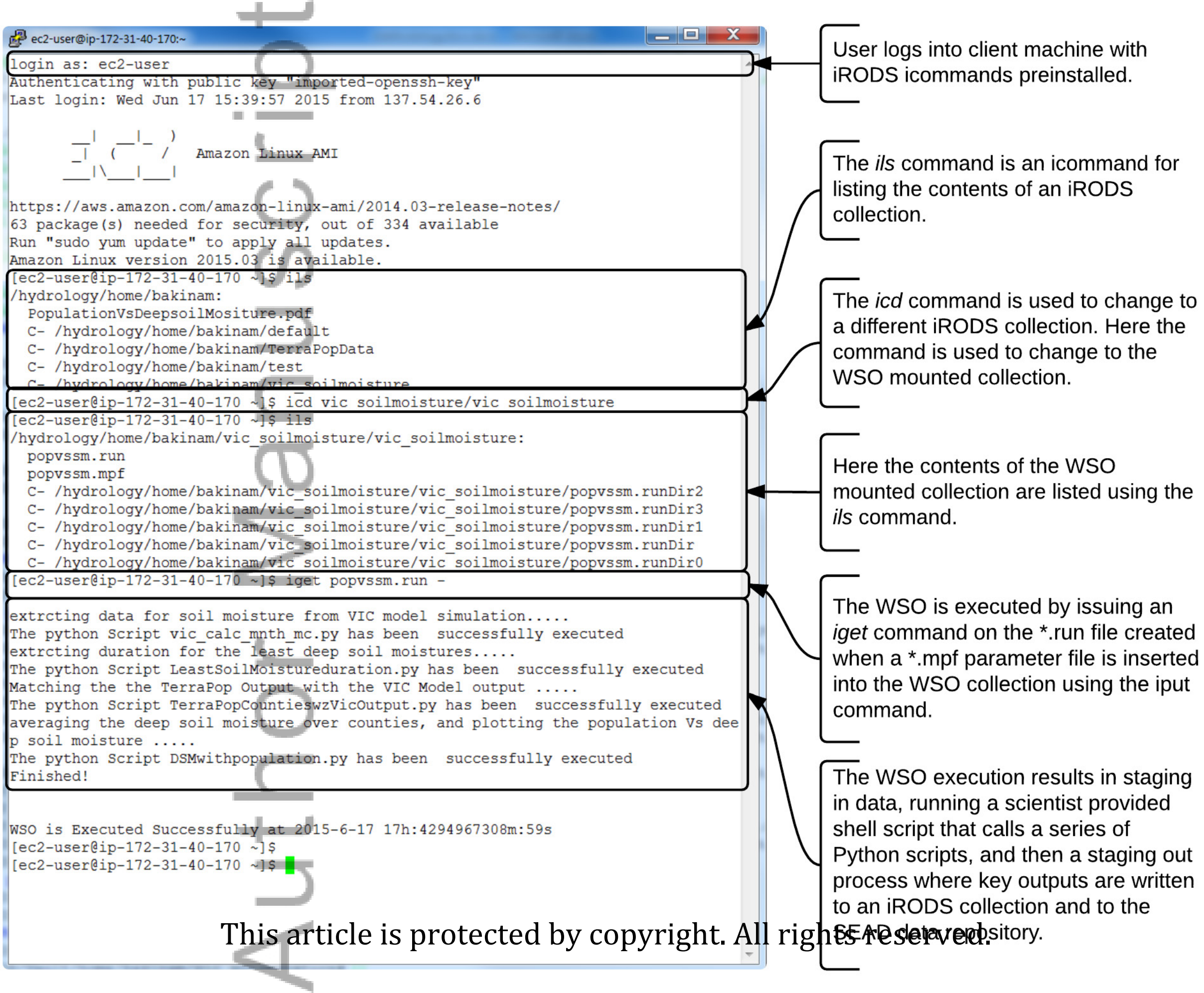
(b) A .csv file that includes the GEOID, county name, total population, and the cropland area



data_US_SLAD_2010.csv

	GEOID_US_label	GEOID_US	GEOID_US_X	GEOID_US_Y	TOTPOP	CROPLAND2000_PctArea
▸	Panola County	G2801070	31.093254	1.745198e+215	34737	0.416397
	Newton County	G2801010	896.535562	1.944139e+215	21720	0.13416
	Coahoma County	G2800270	-123.626427	2.541066e+215	26151	0.966982
	Madison Parish	G2200650	-958.155847	1.396696e+217	12093	0.564123
	Charlottesville city	G5105400	-935.448775	-2.774872e+216	43475	0
	Alexandria city	G5105100	84.755562	-5.335012e+216	139966	0
	Buena Vista city	G5105300	91.536746	-3.427358e+216	6650	0
	Fairfax city	G5106000	927.303779	8798649468704 8.347	22565	0
	Dallas County	G4801130	86.636959	2.226434e+313	2368139	0.484536
	Howard County	G4802270	41.293607	-3.432330e+322	35012	0.387413
	Foard County	G4801550	75.058993	2.446778e+320	1336	0.341514
	Floyd County	G4801530	978.853008	1.443870e+320	6446	0.468157
	Brewster County	G4800430	35.059034	4.544818e+315	9232	0.841332
	Franklin County	G4801590	495.472465	3.521584e+320	10605	0.343474
	Hunt County	G4802310	-918.054831	-1.941980e+322	86129	0.422938
	Newton County	G4803510	1.745979	-8.530881e+315	14445	0.07961
	[illegible]	G4804270	[illegible]	[illegible]	[illegible]	0.209986
	Kinney County	G4802710	[illegible]	[illegible]	[illegible]	0.857574
	Collingsworth County	G4800870	[illegible]	4.654715e+317	3057	0.324931
	Tom Green County	G4804510	-114.626277	-2.384260e+315	110224	0.213896
	Coleman County	G4800830	130.908598	3.675187e+317	8895	0.252865
	Brazoria County	G4800390	177.711617	4.155613e+315	313156	0.268768


Workflow Structure Object (WSO)
PopVsSm.mss
PopVsSm.mpf
PopVsSm.run
(1)
Initiate workflow
Staged in Data
TerraPop
VIC Output
(2)
Execute shell script operating on staged in data
Scientist Provided Scripts
*.py
vic_calc_mnth.py
LeastSoilMositureDuration.py
TerraPopCountieswzVicOutput.py
DSMwithPopulation.py
creating_Visualization.py
PopVsSm.scr
(3)
Key data published in open repository
Publication in SEAD
(4)
Output files are staged out to data grid
Staged out Data
spatiotempSoilMoistureEcohydro.csv
VICModelOuputFiltered.csv
SoilMoistureandCounities.csv
SoilMoistureandCounites.pdf



```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Wed Jun 17 15:39:57 2015 from 137.54.26.6

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2014.03-release-notes/
63 package(s) needed for security, out of 334 available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2015.03 is available.
[ec2-user@ip-172-31-40-170 ~]$ ils
/hydrology/home/bakinam:
  PopulationVsDeepsoilMositure.pdf
  C- /hydrology/home/bakinam/default
  C- /hydrology/home/bakinam/TerraPopData
  C- /hydrology/home/bakinam/test
  C- /hydrology/home/bakinam/vic_soilmoisture
[ec2-user@ip-172-31-40-170 ~]$ icd vic_soilmoisture/vic_soilmoisture
[ec2-user@ip-172-31-40-170 ~]$ ils
/hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture:
  popvssm.run
  popvssm.mpf
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir2
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir3
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir1
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir
  C- /hydrology/home/bakinam/vic_soilmoisture/vic_soilmoisture/popvssm.runDir0
[ec2-user@ip-172-31-40-170 ~]$ iget popvssm.run -

extrcting data for soil moisture from VIC model simulation.....
The python Script vic_calc_mnth_mc.py has been  successfully executed
extrcting duration for the least deep soil moistures.....
The python Script LeastSoilMoistureduration.py has been  successfully executed
Matching the the TerraPop Output with the VIC Model output .....
The python Script TerraPopCountieswzVicOutput.py has been  successfully executed
averaging the deep soil moisture over counties, and plotting the population Vs dee
p soil moisture .....
The python Script DSMwithpopulation.py has been  successfully executed
Finished!


WSO is Executed Successfully at 2015-6-17 17h:4294967308m:59s
[ec2-user@ip-172-31-40-170 ~]$
[ec2-user@ip-172-31-40-170 ~]$
```

User logs into client machine with iRODS icommands preinstalled.

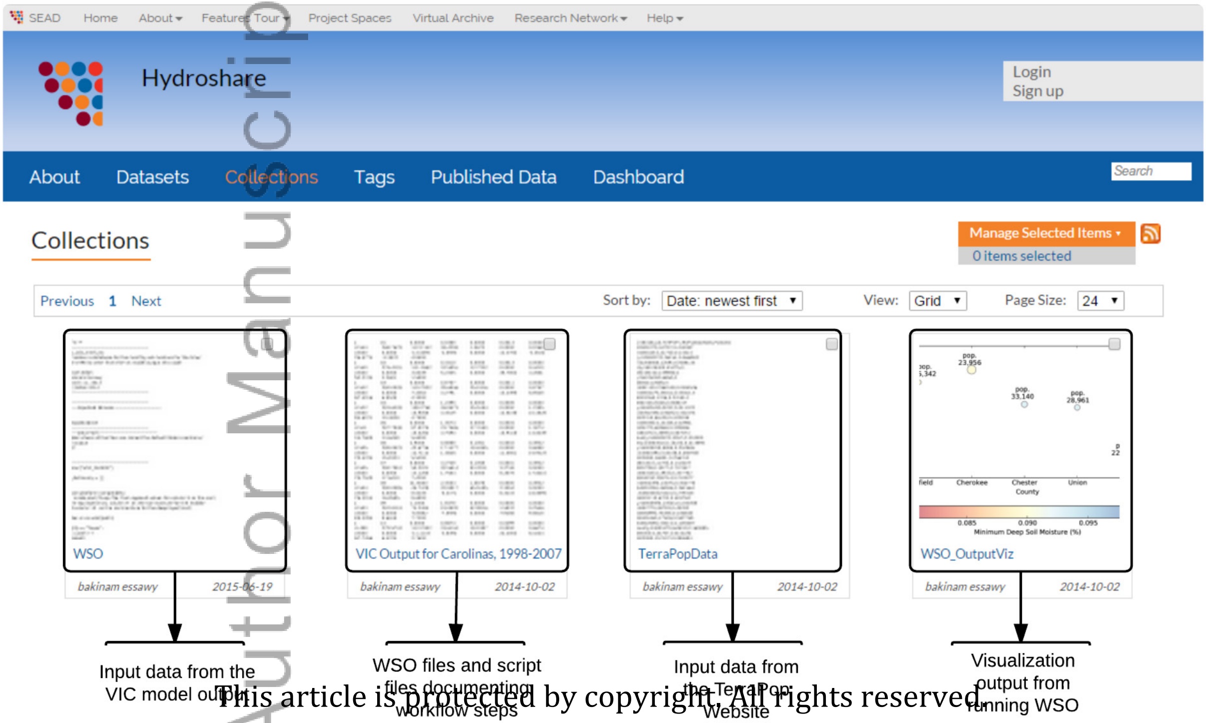
The *ils* command is an icommand for listing the contents of an iRODS collection.

The *icd* command is used to change to a different iRODS collection. Here the command is used to change to the WSO mounted collection.

Here the contents of the WSO mounted collection are listed using the *ils* command.

The WSO is executed by issuing an *iget* command on the *.run file created when a *.mpf parameter file is inserted into the WSO collection using the iput command.

The WSO execution results in staging in data, running a scientist provided shell script that calls a series of Python scripts, and then a staging out process where key outputs are written to an iRODS collection and to the SEAD data repository.

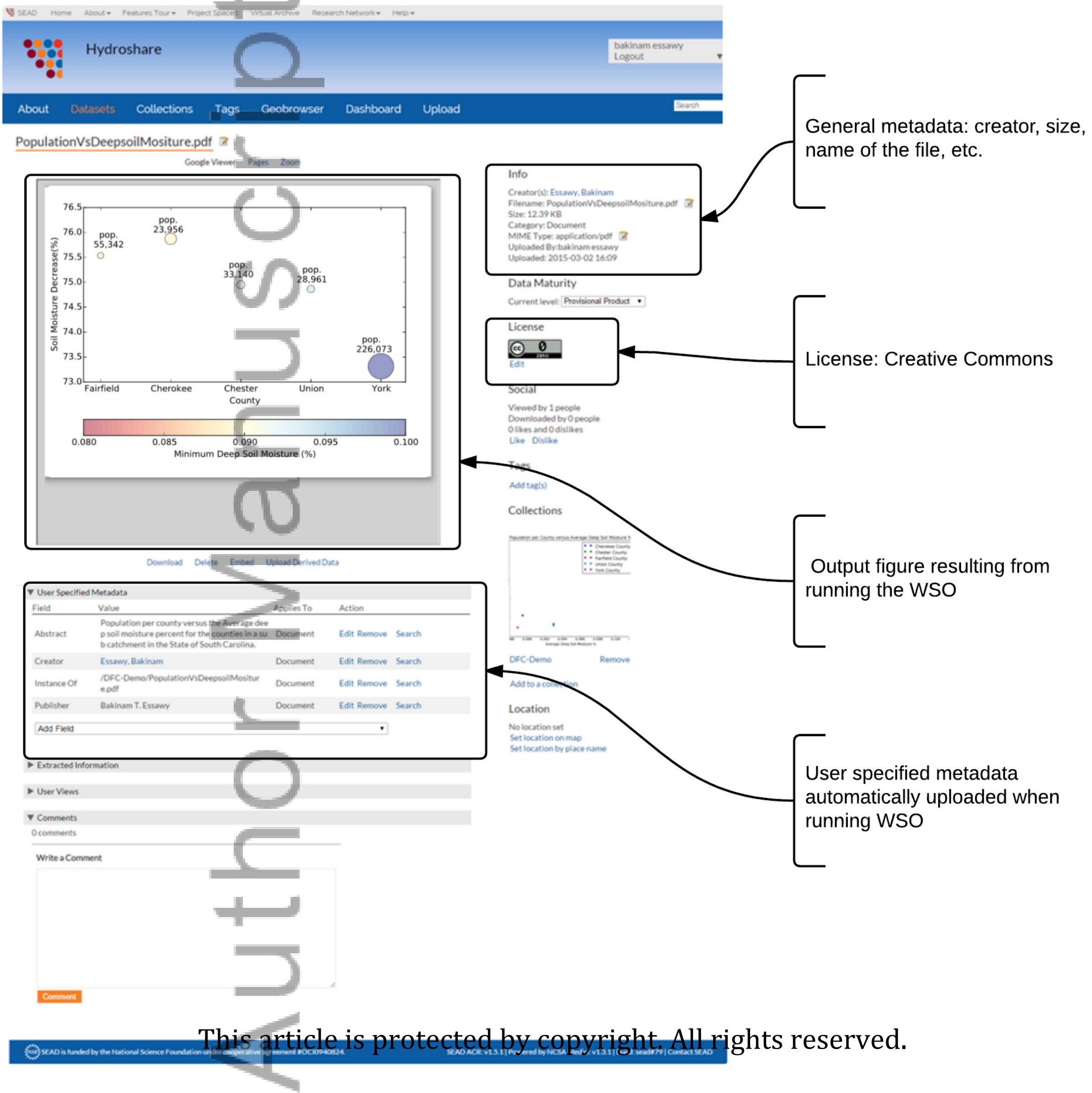


Input data from the VIC model output

WSO files and script files documenting workflow steps

Input data from the TerraPop Website

Visualization output from running WSO



PopulationVsDeepsoilMositure.pdf

Info

Creator(s): Essawy, Bakinam
Filename: PopulationVsDeepsoilMositure.pdf
Size: 12.39 KB
Category: Document
MIME Type: application/pdf
Uploaded By:bakinam essawy
Uploaded: 2015-03-02 16:09

Data Maturity

Current level: Provisional Product

License

Edit

Social

Viewed by 1 people
Downloaded by 0 people
0 likes and 0 dislikes
Like Dislike

Tags

Add tag(s)

Collections

Location

No location set
Set location on map
Set location by place name

▼ User Specified Metadata

Field	Value	Applies To	Action
Abstract	Population per county versus the Average deep soil moisture percent for the counties in a sub catchment in the State of South Carolina.	Document	Edit Remove Search
Creator	Essawy, Bakinam	Document	Edit Remove Search
Instance Of	/DFC-Demo/PopulationVsDeepsoilMositure.pdf	Document	Edit Remove Search
Publisher	Bakinam T. Essawy	Document	Edit Remove Search

Add Field

▶ Extracted Information

▶ User Views

▼ Comments

0 comments

Write a Comment

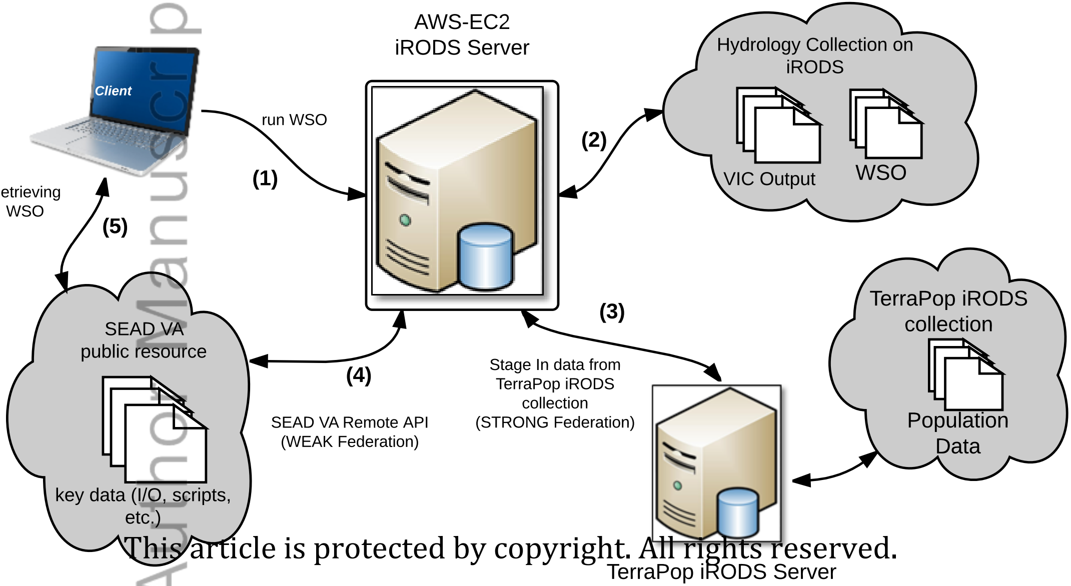
SEAD is funded by the National Science Foundation under cooperative agreement #OCI0940824.   SEAD ACR: v1.5.1 | Powered by NCSA Medici v1.3.1 | [illegible] sead#79 | Contact SEAD

General metadata: creator, size, name of the file, etc.

License: Creative Commons

Output figure resulting from running the WSO

User specified metadata automatically uploaded when running WSO


AWS-EC2
iRODS Server
Client
run WSO
(1)
(2)
Hydrology Collection on iRODS
VIC Output
WSO
Retrieving WSO
(5)
SEAD VA public resource
key data (I/O, scripts, etc.)
(4)
SEAD VA Remote API (WEAK Federation)
(3)
Stage In data from TerraPop iRODS collection (STRONG Federation)
TerraPop iRODS collection
Population Data
TerraPop iRODS Server