



Jun 17th, 10:40 AM - 12:00 PM

Automating Data-Model Workflows at a Level-12 HUC Scale in a Distributed Computing Environment

Lorne Leonard

The Pennsylvania State University, lnl3@psu.edu

Christopher J. Duffy

The Pennsylvania State University, cxd11@psu.edu

Follow this and additional works at: <https://scholarsarchive.byu.edu/iemssconference>

 Part of the [Civil Engineering Commons](#), [Data Storage Systems Commons](#), [Environmental Engineering Commons](#), [Hydraulic Engineering Commons](#), and the [Other Civil and Environmental Engineering Commons](#)

Leonard, Lorne and Duffy, Christopher J., "Automating Data-Model Workflows at a Level-12 HUC Scale in a Distributed Computing Environment" (2014). *International Congress on Environmental Modelling and Software*. 13.
<https://scholarsarchive.byu.edu/iemssconference/2014/Stream-A/13>

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Automating Data-Model Workflows at a Level-12 HUC Scale in a Distributed Computing Environment

Lorne Leonard^a, Christopher J Duffy^a

(a) Department of Civil & Environmental Engineering, The Pennsylvania State University, United States of America {lnl3@psu.edu, cxd11@psu.edu}

Abstract: The HydroTerre web services provide the Essential Terrestrial Variable (ETV) datasets to create common hydrological models anywhere in the continental United States (CONUS). These services allow web users to download data for their own purposes in their own computing environment. The datasets are provided using standard Geographic Information System formats and the data transformation is dependent on the users' own needs, goals, and computing environment. In this article, we demonstrate the feasibility of automating data-transformation workflows for United States Geological Survey level-12 Hydrological Unit Codes (HUC-12) to be consumed in hydrological models. The Penn State Integrated Hydrological Model (PIHM) is demonstrated here, but the workflows serve as a template for other models to adapt and become new services. The focus of this article is the data transformation process, not the model results. We want to demonstrate that workflows empower modelers to create hydrological models rapidly anywhere in the CONUS, and to contribute to a dynamic resource that records provenance of HUC-12 models. To do this, an explanation is required of both the hardware and software architecture because the way in which they are coupled is critical for web service performance. A demonstration of the feasibility to automate data-model workflows for CONUS HUC-12 catchments is discussed with the emphasis on reproducibility by using data-model workflows and distributed computing resources.

Keywords: Data-model workflows; HydroTerre; Web services; HPC; PIHM.

1 INTRODUCTION

The HydroTerre web services (www.hydroterre.psu.edu) allows web users to download data and hydrological model results for their own purposes within their own computing environment. In this article, we demonstrate the feasibility of automating data-transformation workflows for United States Geological Survey HUC-12s (Seaber, Kapinos, & Knapp 1987) to be consumed in PIHM. The workflows can also serve as templates for other models to adapt and become new services. We want to demonstrate that workflows empower modelers to create hydrological models rapidly anywhere in the CONUS, and that our database can act as a data provenance resource for HUC-12 models. The way in which these services are coupled is critical for web service performance. An explanation of both hardware and software architecture is required to explain how the data-workflows operate. Section 1 introduces the reasons for using data-model workflows and what is automated. Section 2 provides an overview map of the architecture and Section 3 demonstrates the feasibility of automating data-model workflows for CONUS HUC-12 catchments.

1.1 Data-model workflows

The ETV services provide both spatial data at the HUC-12 scale, and time-series North American Land Data Assimilation System, (NLDAS 2011), climate forcing for a period of 30 years. Catchment data is available to the user by selecting a HUC-12. An email is then sent to the user with a link to download the selected data via a data workflow as shown by Leonard and Duffy (2013). Here, we demonstrate the next step of transforming the HUC-12 data into PIHM input datasets which requires numerous and time-consuming steps, via a web application. Transforming data is not simply about converting one file format to another. The data-model workflow is about preparing data to be consumed in a hydrological model. This involves, among other things, simplification of catchment boundaries, stream delineation, and generating meshes that represent the physical processes being

investigated by the modeler. At times, these steps produce erroneous results either due to the original data or due to the process itself. Thus, the modeler invests time manually changing individual parameters and input data to optimize the data inputs to generate quality hydrological models. Data and model provenance is then lost and frequently not easy to reproduce as there is no record for why the modeler made specific decisions. Therefore, the fundamental purpose of these workflows is to capture these altered steps and to enable reproducibility and provenance of data transformations.

To capture these steps requires documenting user interaction within the web application. The workflows described here do not restrict users from downloading the data transformation results and using the data offline. However, the emphasis of these workflows is reproducibility and rapid prototyping so users can retrieve a personal copy at the end of the process. Furthermore, the method to create the data inputs is easily re-created with the stored parameters of the user interaction to replicate the entire workflow processes. This may appear trivial for a few case studies. However, 100's of terabytes of data storage is necessary for the ETV web services and assuming 1 to 10 gigabytes of storage is required for 30 years of input data per HUC-12, 100's of terabytes of disk storage would be required to keep data-transformation steps for the CONUS. Data workflows eliminate the need to keep the data transformation results and only store the modelers' input.

1.2 Automation with data-model workflows

A user selecting a HUC-12 initiates the data-model workflow. There are three phases; the first is the ETV data workflow that is responsible for selecting, projecting, clipping, and extracting data within the HUC-12 catchment efficiently. The sources of the ETV data have been transformed from their original file formats to databases so that data can be consumed in multiple ways and are accessible in a distributed computing environment. What is being automated with this phase is the entire data selection process pertaining to the spatial context of the selected HUC-12. Meta-data is also included with the selection process, including the data sources, what data versions are used, and the software version used.

The second phase is the transformation of data generated in the first phase into PIHM input file formats. There are two categories to this phase. The first is the physical representation of the catchment, represented as a mesh and stream network. Each mesh cell has soil, geology, land cover, and climate-forcing variables assigned. Desktop PIHMgis (Bhatt et al. 2008) has been developed to assist in the creation of these files, but requires user intervention with the geometry creation of the catchment and stream delineation. The geometry steps have been automated as a high performance computing (HPC) web service and are controlled by user-defined variables that control the simplification process. Terrain, soil, geology, land cover, and forcing data from the ETV workflow are automatically assigned to mesh cells. The second category is model parameters that define the initial conditions and calibration values to control and calibrate the hydrological model. Default values are assigned to these parameters so that a beginner user can generate a hydrological model, but expert experience is required for model calibration. The third phase is the web-based user interface that captures the steps involved in phases one and two. All the workflow parameters are stored as database objects for fast retrieval, provenance, and reproducibility. This happens automatically when a user submits a task and is a critical procedure to sharing important parameters at a HUC-12 scale with other users.

1.3 Constraints

This article focuses on the data-model transformation workflow process and the use of a distributed computing environment to evaluate the input data. It is beyond the scope of this article to discuss hydrological model calibration and the validity of the model results. Due to administrative restrictions, executing model workflows discussed in the article may not be available to the public at times, but data-model workflows are available. In addition, the workflows presented here are restricted to one HUC-12 selected by the web user. The next phase of this prototype will deal with issues associated with scaling up to a network of HUC-12 watersheds, as discussed in Leonard & Duffy (2014), where flow direction between HUC-12s requires validation to verify the hierarchy and HUC-12 model calibration.

2 SYSTEM DESIGN

This section describes how computer hardware and software architecture contribute to automating data-model workflows. The hardware has been structured to support large volumes of data required to support data-model workflows anywhere in the CONUS. The data-model processing is distributed within the data-tier of the HydroTerre system, and the hydrological modeling is distributed to other HPC systems to compute PIHM models. Service-oriented architecture that is efficient and robust is critical to support rapid prototyping and distribution of the workflows.

2.1 Hardware and administration layers

The data-model workflows are implemented in a three-tier hardware layer system. The web interface tier hosts the web applications and services. ESRI's ArcGIS server software (ESRI, 2014) development kits (SDK) support GIS web applications and Microsoft SQL server (Microsoft, 2014a) is used to store, create, and query datasets. The web server disk is partitioned into three components for maximum performance; the windows server operating system, GIS datasets that perform select and query operations, and the third partition for workflow results that are temporarily stored for users to download via a web link.

Two 100-terabyte data support servers constitute the second tier with each server containing multiple processors. These servers contain thousands of databases that form the 30-year NLDAS-2 climate forcing dataset and queried by Microsoft SQL server. Components of the data workflows that retrieve the forcing data are implemented on this tier and form the first layer of the distributed computing system. All the components are executed in parallel for maximum performance and range from minutes to hours depending on the catchment size. The reader is referred to Leonard & Duffy (2013) for further details about compute times and data sizes of ETV datasets.

The web and data tiers are tightly united via a private fast network router to minimize performance lost when retrieving datasets between servers. The data-model workflows reside on these two tiers and are explained further in Section 2.2. Both ETV and data-model workflow results are compressed and zipped, reside on the web interface tier. Web users gain access to these results via a public network connection. The model support tier also gains access to the zipped data using the same public network. Using a specified PIHM account, a custom PIHM dispatcher application runs continuously to retrieve PIHM jobs via web services on the CyberSTAR cluster (CyberSTAR, 2014). The user does not need to login to the compute environment and the PIHM models are automatically dispatched to the compute nodes. Job management is achieved via the data tier, with all compute nodes accessing job tasks from the data tier, via user-project databases. Statistics about data and the model performance are returned to the web tier database. Computer names are recorded when a PIHM job is executed, so the exact workflow can be replicated on the exact distributed computer. Having the ability to replicate the entire workflow is extremely useful for debugging purposes when trying to determine if data is being lost or corrupted via network connections. Another issue for debugging is determining whether hardware and/or operating systems are causing problems with the model jobs. To the reader this may appear unnecessary, but when 100,000s of PIHM model jobs are running within different types of HPC environments, having this information is valuable to understand failure. The web tier is the interface to all aspects of the workflows, with the data and model tiers operating without any user intervention. Section 2.2 discusses how the three tiers work together when a web user starts the HydroTerre web application.

2.2 Overview map of service-oriented architecture

The workflows are service-oriented architecture (SOA) that comprise hundreds of discrete pieces of software which provide application functionality to other applications that constitute HydroTerre data-model workflows. Here we provide the reader with an overview map of the SOA and explain the main paths behind the web application. When a user visits, via a web browser, the web application, http://www.hydroterre.psu.edu/Development/HydroTerre_Leonard_Models, they are accessing

internet services hosted on the web interface tier. The HydroTerre user interface has been developed with Microsoft Silverlight (2014c) and ArcGIS server SDK (ESRI, 2014). The user interface is responsible for selecting, querying, creating, and retrieving Microsoft SQL Server datasets for display within the web application (Figure 1a). All data, displayed and used in controls, resides in databases on the data-tier; the user interface is data driven.

The main communication methods between the user interface and the data tier, and between the data tier and workflow service layer (Figure 1b) are Simple Object Access Protocol (World Wide Web Consortium, 2014a), Representational State Transfer (Fielding & Taylor, 2002), Web Services Description Language (World Wide Web Consortium, 2014b), and Microsoft Database Markup Language (2014b). The choice of communication technique depends on where the data resides, what tier layer, and system administration. However, the overriding choice depends on balancing performance with disk, memory, and number of central processing units. Due to differences in hardware configurations, just because one tool operates with high performance on server type 1, this does not equate that the same tool operates the same way on server type 2. Thus, some of the workflow tools called by the user interface have multiple versions simply due to where the tool resides.

The data tier (Figure 1c) has two categories. The first consists of ETV datasets and the reader is referred to Leonard & Duffy (2013) for further details about their function and computation complexity. A metadata repository is connected to the ETV datasets which stores information about the ETV datasets properties, version, and technical attributes. The metadata repository is queried by the web application and informs users, by populating interface controls, of available ETV datasets. The second category contains databases that store all the parameters chosen by users after they have chosen to execute the workflows. These databases are queried by the user interface to populate data controls so that users can interrogate the workflow results (success or failure), inspect provenance, create a clone of the user's parameters to tweak parameters, and re-submit the workflow.

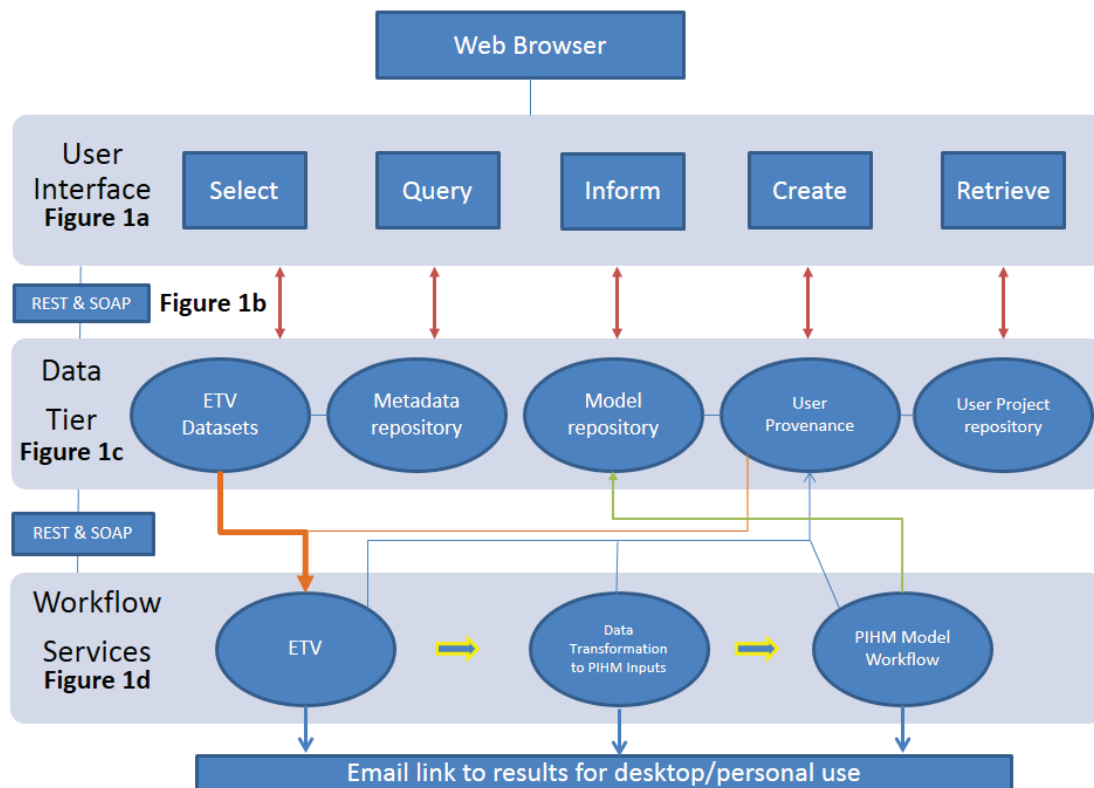


Figure 1. Service-oriented architecture for data-model workflows consists of three layers. The first layer is the web based user interface, supported by a data tier layer, and a workflow service layer.

When a user selects a HUC-12 and submits a job to execute the workflows, a new table row is created with a primary key. Each row contains the HUC identification key and the users' email address, and each workflow is stored as a separate Extensible Markup Language (XML) object. Thus, via the web interface, queries against HUC-12 names can be made to populate the data controls and replicate the workflow parameters. Recall that only the parameters for workflows are stored, not the model results. Therefore, a user cannot simply download the ETV or data-model results from a previous job, due to the large amount of disk storage required to store results. The workflow will need to be executed again, but at the HUC-12 scale, the time to re-create is minimal and requires slight effort from the user.

Assuming a user is creating a new, or re-executing, a HUC-12 job, the execution of workflow services are shown in Figure 1d, from ETV, data-model to PIHM model. There are potentially thousands of errors when executing the workflows to create or share data. To empower the web user to resolve an error returned during workflow execution, either due to administration or from parameter issues, a meaningful error object is returned to the user via the web interface. The error object has a unique key, the software tool name and version, operating system, machine name, timestamp, and reason for the error. Furthermore, the possible number of errors is increased, as tasks are distributed on various operating systems and developed with different computer languages that need to be aggregated into user-friendly messages.

To simplify the explanation of data automation workflows for HUC-12s in a distributed computing environment, the software versions are omitted. In addition, the computer language the software has been developed with is not included. Briefly, the software languages used in both Linux and Windows environments include C++, C, and Python. Windows environments additionally include C#, JavaScript, Silverlight, ASPX and SQL. These differences occur as Windows server is the operating system for the web and data tiers, while both Linux and Windows operating systems are used in the model tier. It should be noted that there are multiple workflow versions that have been designed and developed to be optimized for PIHM in HPC environments that are specific to PIHM in these environments. This is due to the emphasis on performance, which is constrained by the various computing environments and management practices. Other models will have the same difficulties and constraints. Here, we want to emphasize the importance of reproducibility, provenance, and rapid hypothesis testing by using data-driven automated workflows.

3 DEMONSTRATION AND EVALUATION OF AUTOMATED WORKFLOWS

At the website www.hydroterre.psu.edu, under the services tab, a standalone demonstration to execute the ETV workflow is available to the reader. Here, we present a prototype web application that does not treat the ETV workflow www.hydroterre.psu.edu/Development/HydroTerre_Leonard_Models as a standalone service. Instead, the service is coupled with both the data-model and model workflow services. Section 3 demonstrates the use of data-model workflows to automate hydrological modeling at the HUC-12 scale with PIHM. First, we show how a user defines and initiates the workflows. This is followed by a brief description of the initial results for evaluating PIHM models at every CONUS HUC-12 using the distributed compute environments.

3.1 Defining and initiating data-model and model workflows

Operating the web application requires user credentials to access remote HPC resources and a strategy to select catchments from a HUC-12 to CONUS state scale. After the user has created a selection list of HUC-12s to model, the data workflow is selected. At present, there is only one ETV workflow, so the user is not required to select a version to use within their model. However, the web user needs to select which data-model workflow they wish to use as highlighted in Figure 2a. In this prototype, the user can select to create stream networks using TauDEM (Tarboton, 2011), or they can use NHD streams. The user can define data-model workflow parameters by clicking on the interface button highlighted in Figure 2b to control parameters. Any changes will be applied to the selected HUC-12 selection list highlighted in Figure 2c. After defining the data-model workflow properties, the user can select which PIHM workflow version and HPC resource they wish to use (Figure 2d). The

user can define and control PIHM parameters, by clicking on the interface button highlighted in Figure 2e. Assuming the user has valid credentials and has defined the data-model workflows, the user initiates the process by clicking on the submit model button (Figure 2f) which adds the project objects to the workflow submission list (Figure 2g).

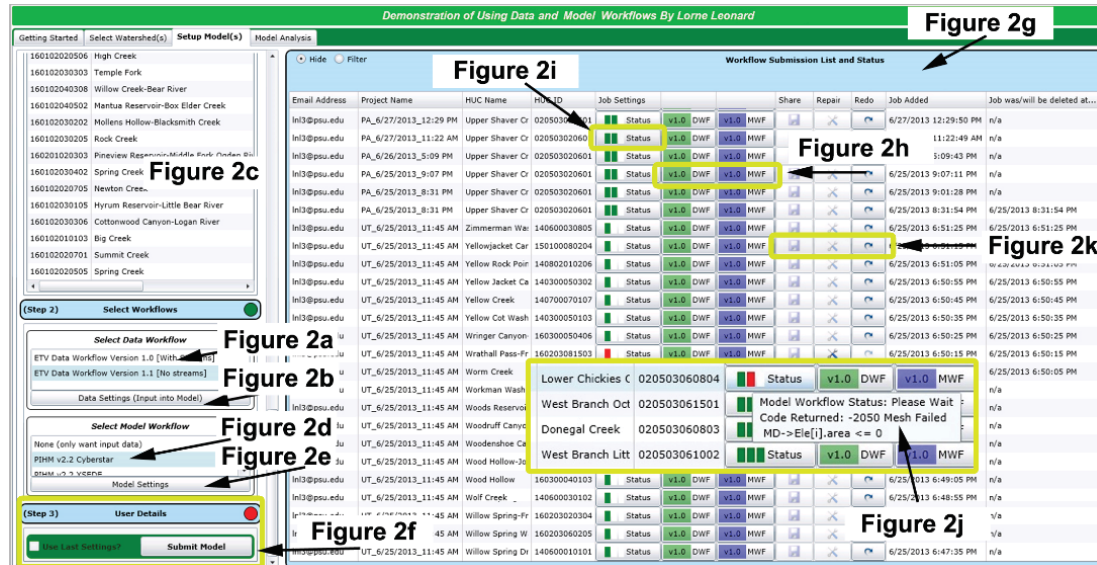


Figure 2. Demonstration of using data-model workflows to start a PIHM HUC-12 hydrological model using distributed computing environment.

The project object indicates the users' email, the project name, the HUC identification, and when the job was added to the submission list. The user can investigate all the workflow settings by clicking on the appropriate buttons highlighted in Figure 2h. The status of the workflows (Figure 2i) is indicated to the user with four colors; white indicates the workflow was not requested by the user, orange indicates the workflow has started, green indicates the workflow succeeded, and red indicates the workflow failed. When one workflow has succeeded, the next workflow starts on the next available computing environment. As described in section 2, the data-model workflows execute on the HydroTerre distributed computing system, while the PIHM models are distributed on the user's selected HPC resources. When the workflow has failed, the reason for failure is available to the user by clicking on the status button which reveals the dialog (Figure 2j). This provides the user with valuable information about ways to fix the problem. The main reason for failure using this prototype is poor meshes, which requires the user to modify catchment simplification parameters. However, this prototype web application simplifies the process of fixing these errors by the user selecting an existing project and cloning the project. Then, the user changes parameters to resubmit the workflow processes (Figure 2k).

The submission list embraces the provenance data associated with all the workflows. All the parameters chosen by users are kept in the user project databases, which allows users to query other modeler's choices involved with any CONUS HUC-12. Unfortunately, model results are not stored permanently, due to the large amount of disk resources that would be required. To store one PIHM model result for 30-years of forcing for all CONUS HUC-12 requires three petabytes of disk storage. However, by storing the provenance steps only, the need for large amounts of disk storage is reduced. Thus, this prototype can store thousands of workflow settings per HUC-12, providing a new resource for modelers using the system to gain insight to start a new model study and to download a refined model that is reproducible.

3.2 Evaluating CONUS HUC-12s using automated workflows

HPC resource CyberSTAR was used to evaluate all 90,762 CONUS HUC-12s using the automated workflow strategy. The ETV and data-model workflows took 80 hours to prepare the PIHM inputs for two months of forcing data. Approximately 25 million input files were generated and deleted during the

process, creating close to 450 gigabytes of input data. Each CONUS HUC-12 was appended to the submission list and six CyberSTAR nodes continuously downloaded and ran PIHM models with the input files. The model jobs took two weeks, generating 10 terabytes of results, which were immediately deleted once the outcomes (failure or success) were added to the project objects as discussed in section 3.1. The automated process has been repeated more than twelve times to improve the software and hardware infrastructure. More than a million ETV and data-model workflows have been generated to collect provenance data regarding each HUC-12 in the CONUS. Recall, these model results are not calibrated and the default settings may not be appropriate. However, the provenance data is useful for understanding where the workflows fail and with the error objects, where exactly within the workflows they failed. Table 1 summarizes the data-model and model workflow results. On average, across the CONUS states using default values, the data-model workflows succeeded 57.81% of the time. Of those that failed, stream network issues caused 26.91% of the errors, poor meshes caused 5.14% of the errors, and the remaining errors (10.14%) were due to hardware and operating system failures.

Table 1. Data-model workflow results.

	HUC12 Count	Data-Model Workflow				
		Success	Failed	Stream Network Failed	Mesh Failed	Remaining Error
State Average	1852.3	57.81	42.19	26.91	5.14	10.14

4 CONCLUSIONS AND RECOMMENDATIONS

In this article, we described automating data-model workflow services that transform CONUS Essential Terrestrial Variables (ETVs typically used in hydrological model studies) to model input datasets, and then execute the hydrological model in a distributed compute environment using a world wide web-based application that enables researchers and modelers to retrieve ETV and data-model data. By balancing hardware and software configurations, we demonstrated the feasibility of transforming data sources from several federal agencies that amounts to hundreds of terabytes of disk storage, by implementing a workflow prototype for HUC-12 catchments within the CONUS. We demonstrated the effectiveness of data-model workflows using automated strategies by distributing the data-model datasets on high performance computing environments with CONUS HUC-12s. This creates a provenance workflow dataset for reproducibility and is accessible to researchers and modelers in order to gain insight and develop PIHM hydrological models anywhere in the CONUS.

We have presented an important step towards eliminating hurdles involved with using physics based models such as PIHM in a HPC environment by seamless allocation of resources with minimal interaction from the user through web-based workflows, shared software, data and HPC resources. The next step is to improve the mesh generation techniques and scale from HUC-12 to major river basins, requiring adaptation of existing HUC-12 workflows to evolve the software to seamlessly allocate the larger data requirements within the existing prototype.

REFERENCES

- Bhatt, G., Kumar, M., & Duffy, C. J., 2008. Bridging the Gap between Geohydrologic Data and Distributed Hydrologic Modeling. In M. Sánchez-Marré, J. Béjar, J. Comas, A. Rizzoli, & G. Guariso (Eds.), Proceedings of the 2010 International Congress on Environmental Modelling and Software, vol. 2, 743-750.
- CyberSTAR, 2014. A Scalable Terascale Advanced Resource for Discovery through Computing. <http://www.ics.psu.edu/infrast> (last accessed 01.01.14.).
- ESRI, 2014. ArcGIS Server. <http://www.esri.com/> (last accessed 01.01.14.).
- Fielding, R., Taylor, R., 2002. Principled design of the modern Web architecture. ACM Transactions on Internet Technology. vol. 2, (2), 115-150.

- Leonard, L., Duffy, C.J., 2013. Essential Terrestrial Variable data workflows for distributed water resources modeling. *Environ. Modell. Softw.* 50, 85-96.
- Leonard, L., Duffy, C.J., 2014. HydroTerre: Selecting Up-stream level-12 HUCS using Depth-First Graphs anywhere in the Continental USA. In: 11th International Conference on Hydroinformatics, HIC 2014, New York City, USA.
- Microsoft, 2014a. SQL Server. <https://www.microsoft.com/en-us/sqlserver> (last accessed 01.01.14.).
- Microsoft, 2014b. Database Markup Language, DML. <http://msdn.microsoft.com/en-us/library/bb399400%28v=vs.110%29.aspx> (last accessed 01.01.14.).
- Microsoft, 2014c. Silverlight. <http://www.microsoft.com/silverlight> (last accessed 01.01.14.).
- NLDAS, 2011. North American Land Data Assimilation System, NLDAS. <http://ldas.gsfc.nasa.gov/nldas/NLDAS2forcing.php> (last accessed 08.01.11.).
- Seaber, P. R., Kapinos, F. P., & Knapp, G. L., 1987. Hydrologic unit maps. United States Geological Survey.
- Tarboton, D. G., 2011. TauDEM Hydrology Research Group. <http://hydrology.usu.edu/taudem/taudem5.0/index.html> (last accessed 02.03.11.).
- World Wide Web Consortium, 2014a. Simple Object Access Protocol, SOAP. <http://www.w3.org/TR/soap12-part1> (last accessed 01.01.14.).
- World Wide Web Consortium, 2014b. Web Services Description Language, WSDL. <http://www.w3.org/TR/wsdl> (last accessed 01.01.14.).