Jun 17th, 2:00 PM - 3:20 PM

Tethys: A Software Framework for Web-Based Modeling and Decision Support Applications

Norm Jones
Brigham Young University, njones@byu.edu

Jim Nelson
Brigham Young University, jimn@byu.edu

Nathan Swain
Brigham Young University, nathan.swain@byu.net

Scott Christensen
Brigham Young University, sdc50@byu.net

David Tarboton
Utah State University, dtarb@usu.edu

See next page for additional authors

Presenter/Author Information

Norm Jones, Jim Nelson, Nathan Swain, Scott Christensen, David Tarboton, and Pabitra Dash

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

Tethys: A Software Framework for Web-Based Modeling and Decision Support Applications

Norm Jones[1], Jim Nelson[1], Nathan Swain[1], Scott Christensen[1], David Tarboton[2], Pabitra Dash[2]
[1]*Brigham Young University, Provo UT 84602, njones@byu.edu, jimn@byu.edu, nathan.swain@byu.net, sdc50@byu.net*
[2]*Utah State University, Logan UT 84322, dtarb@usu.edu, pabitra.dash@usu.edu*

Abstract: We have developed a software framework called Tethys to aid in the creation of web-based water resource modeling applications. This suite is a Python-based scripting environment that leverages open source tools for geoprocessing of spatial data, map rendering and visualization, distributed computing, and database management. The system makes it possible to deploy a calibrated, high-resolution watershed or surface water model as a web-based application for decision support. The framework provides data management, access to computing resources, and pluggable components (e.g. plots, maps and user controls) that enable rapid development of modeling applications We have used the system to develop prototype applications for land use change impacts such as burned area analysis and urban master plan development, impact of snowmelt on spring runoff in mountainous regions, and flood early warning.

***Keywords**:* Cloud computing; decision support; watershed modeling; distributed computing; geoprocessing

1 INTRODUCTION

CI-WATER is a cooperative research grant funded by the National Science Foundation involving researchers from two states and four universities: Brigham Young University, the University of Utah, Utah State University, and the University of Wyoming. CI-WATER is being used to establish a robust and distributed cyberinfrastructure (CI) consisting of data services, visualization tools, and a comprehensive education and outreach program that will support this integration and improve the manner in which computer models are used to support long-term planning and water resource management in the U.S. Intermountain West.

One of the objectives of the CI-WATER project is to enhance access to data and computationally intensive modeling. Water resources stakeholders and decision makers often use hydrologic simulations to estimate and analyze watershed responses to specific scenarios. Such decisions are formed largely on the basis of reports generated from hydrologic simulation analysis and forecasts. One problem is that at the time a report is generated from the simulation model it is impossible to analyze all current or even conceive of future scenarios for which the model may be applicable. This limits the utility of the model in making good decisions. Our objective is to enable an environment where data-rich, spatially distributed simulations can be hosted as living models that can be used to look at a wide range of existing and possible future scenarios that might be critical in making important water resources decisions.

Recent advances in cloud computing offer the opportunity to facilitate better use of water resource models as decision-making tools. In a cloud environment, modeling software can be hosted on a remote server and utilized by multiple remote users via a web interface. This eliminates the need for the model user to procure and maintain the high performance hardware required by the models, deal with issues related to software installation and platform incompatibilities (Mac vs. PC vs. Linux, etc.), or monitor and install software updates. All that is needed to use the system is an internet connection and a web browser. However, the software development required to deploy an effective cloud-based modeling system can be substantial. As part of the CI-WATER project, we have developed a software framework which we call “Tethys” that can simplify this process. The Tethys framework consists of a stack of software tools for web-content management, user accounts, spatial databases, geoprocessing, mapping and visualization, distributed computing and model scripting. These tools can be used to develop cloud-based applications or “web apps” ranging from tools for building and running basic models from scratch, all the way to highly customized workflows designed for specific decision-making scenarios such as flood mapping, land-use change analysis, and climate-change impact modeling. In this paper, we describe the primary components of the Tethys framework and we present use cases illustrating prototype applications built using the framework.

2 TETHYS SOFTWARE FRAMEWORK

The Tethys software framework synthesizes several free and open source software (FOSS) projects that are needed for the development of cloud-based hydrologic modeling applications. Tethys includes software that offers typical web development tools in the form of a web framework as well as web-based geographic information systems (GIS), high performance computing management utilities, and interactive scientific visualization libraries. Applications are developed using a Python software development kit (SDK) that includes modules for working with each of the software components.

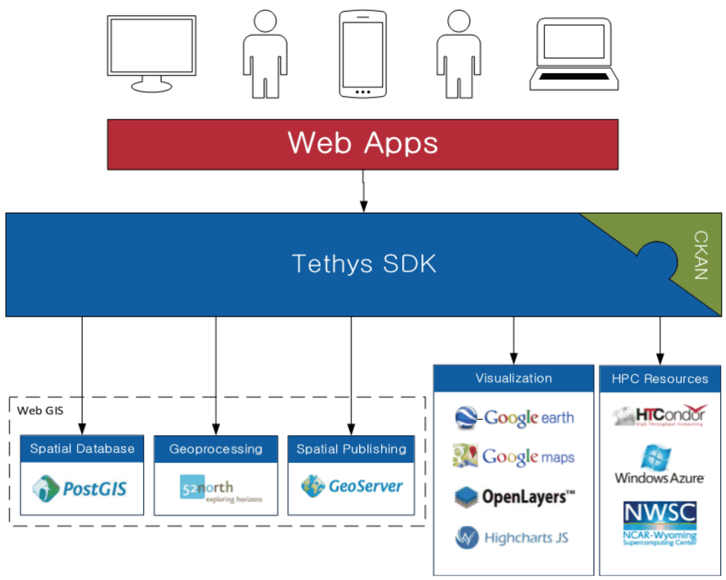

Figure 1. The Tethys software framework diagram.

A summary of the primary software included in the Tethys software framework is shown in Figure 1. The Web GIS capabilities are provided by three software projects: GeoServer, PostGIS, and 52°North, which provide spatial data publishing, storage, and geoprocessing capabilities, respectively. Scientific visualizations are accomplished using Google Earth, Google Maps, and OpenLayers for mapping and Highcharts JS for plotting and tabular data visualization. The framework also includes HTCondor for managing distributed computing and tools for interacting other high performance computing (HPC) resources such as Microsoft Windows Azure cloud computing and the NCAR Wyoming supercomputing facility (NWSF). The Tethys SDK for developing web apps is implemented as a plugin for the CKAN data management system. With the exception of the Google and Microsoft components, all of the software included in Tethys are FOSS. A brief description of each component is provided.

2.1 Web-Based GIS

Some of the more specialized needs for hydrologic app development arise from the spatial data components of the models that are used in the apps. Distributed hydrologic models, for example, often require several raster or vector layers to provide the spatially varying parameters such as land use maps, digital elevation models, and rainfall intensity. GIS software is required to acquire, modify, store, visualize, and analyze spatial data. However, incorporating FOSS GIS can be difficult for app developers, because of the large number of FOSS GIS software that are available [Steiniger & Hunter, 2012].

We have chosen software that implement Open Geospatial Consortium (OGC) standards to ensure interoperability [OGC, 2014]. Although there are several different types of GIS software that offer a wide range of functionality, for simplicity we divide the GIS software needs of hydrologic app development into three broad categories: spatial data management, map rendering, and geoprocessing.

2.1.1 Spatial Database

Spatial database management systems (DBMS) store geographical data in a file system that is suitable for large datasets with thousands of geographical features and provide an efficient mechanism to store, query, analyze, and update these data [Steiniger & Hunter, 2012]. Many Spatial DBMSs use existing structured query language (SQL) databases as the underlying structure with SQL as the API.

The Tethys SDK incorporates PostGIS, a spatial database extension for the PostgreSQL FOSS database, to provide spatial data storage capabilities for app developers [Holl & Plum, 2009; Nguyen, 2009]. It adds extra column types including raster, geometry, and geography. It is the most extensive implementation of the applicable OGC standard [Steiniger & Hunter, 2012]. The extension also provides hundreds of database functions for basic analysis of GIS objects and coordinate transformation. It also supports spatial indexing schemes that allow for quick retrieval of records from large spatial tables during query [Nguyen, 2009].

2.1.2 Map Rendering

There are two capabilities needed to visualize spatial data in a web application: a map server and a mapping library or plugin for the browser. The role of a map server is to publish the data as standardized web services. Mapping libraries and plugins access the data that is published on map servers via the web service application programming interface (API) and render the data as interactive maps.

The Tethys SDK provides GeoServer for publishing spatial data as web services. GeoServer is a Java-based program, which is implemented with OGC web service standards [Iacovella & Youngblood, 2013] meaning the resources served by GeoServer can be accessed as OGC web services including KML for use with Google viewers. Spatial data layers are styled in GeoServer using the Styled Layer Descriptor (SLD) OGC standard. GeoServer is capable of serving many common spatial files types including Shapefiles, ArcGRID, GeoTIFF and others. It can also be used to publish spatial database tables from PostGIS [Iacovella & Youngblood, 2013].

Tethys provides several alternatives for creating visualizations of spatial datasets in apps including: OpenLayers and Google Maps™ libraries and the Google Earth™ mapping plugin. OpenLayers is a pure JavaScript web-mapping client library [Steiniger & Hunter, 2012] for rendering interactive maps on a web page [Hazzard, 2011]. It is capable of displaying 2D maps of OGC web services (including KML). OpenLayers did not support a 3D globe at the time of writing this paper. Google Earth™ can be used to visualize spatial data in a 3D globe environment using KML. However, Google Earth™ requires the user to install a browser plugin that is not supported on all systems. Google Maps™ is provided as plugin-free alternative to Google Earth™, though it does not include 3D globe capabilities. It provides the ability to render spatial data in a 2D mapping environment similar to OpenLayers. Section 2.2 supplies a more in-depth discussion of the visualization capabilities specific to Tethys.

2.1.3 Geoprocessing

The Tethys software framework includes 52°North WPS as one means for supporting geoprocessing needs in hydrologic app development. 52°North WPS is a full open-source implementation of the OGC WPS standard [52°North, 2014; Schut, 2007]. It provides an extensible, pluggable framework for publishing geoprocessing algorithms as web services and it can be linked with geoprocessing libraries such as GRASS, Sextante, and ArcGIS® Server for out-of-the-box geoprocessing capabilities [Steiniger & Hunter, 2012]. 52°North WPS also allows developers to publish custom Python [Python Software Foundation, 2013] and R [Chambers, 2013] scripts as web services.

For data that is stored in a PostGIS-enabled database, geoprocessing can also be accomplished using the spatial database SQL functions that are included with the PostGIS extension. PostGIS includes functions for splicing, dicing, morphing, reclassifying, and collecting/unioning raster and vector types. It also includes functions for extracting vectorizing rasters, clipping rasters with vectors, and running stats on rasters by geometric region [Holl & Plum, 2009].

2.2 Visualization Tools

Visualizations of spatial data can be built into apps using the Tethys SDK using the OpenLayers, Google Earth™, or Google Maps™. To provide the functionality necessary in a GIS modeling environment a custom visualization interface was built around Google Earth. This interface gives the end user increased flexibility in visualizing data sets by providing a legend, and controls for adding and removing layers, reordering layers, changing the symbology, as well as animation controls for temporally varying data sets. These added features are essential to visualizing and understanding model results. This custom interface is configurable, so that a developer can enable only the features desired for a particular application.

Highcharts is a JavaScript library created by Highsoft AS. The Tethys SDK incorporates the Highcharts library for easily displaying and visualizing tabular data. The charts are interactive with hovering effects, pan and zoom capabilities, and the ability to export the plots as images. Supported plots include line, spline, area, area spline, column, bar, pie, scatter, angular gauges, area range, area spline range, column range, bubble, box plot, error bars, funnel, waterfall and polar chart types [Highsoft AS, 2014].

2.3 Distributed Computing

Many hydrologic modeling applications require a significant amount of computing resources. One of the goals of the Tethys software framework is to provide an environment that facilitates access to the necessary resources for various modeling applications. Running stochastic realizations of a model is a common example of an application that needs significant computing resources. This is a case of “embarrassingly parallel” computing and it is easily implemented in a distributed computing environment. To achieve this environment Tethys relies on HTCondor, a job management system developed at the University of Wisconsin–Madison that enables pooling of local resources (PCs on a local network), commercial cloud computing resources, and supercomputers [Thain, 2005].

2.4 CKAN Data Management System

The CKAN Data Management System provides some of the core web capabilities to the Tethys software framework. CKAN is a specialized content management system that is tailored to hosting scientific dataset. Some of the functionalities it provides out-of-the-box include: user authentication and authorization management, data management, data visualization, a REST API for accessing data programmatically, and social media integration for sharing data. Data can be uploaded to CKAN through the a web browser and it can be stored with a rich set of metadata. In addition, CKAN provides a set of interfaces for building extensions [Open Knowledge Foundation, 2014].

2.5 Python Software Development Kit

The Python SDK ties all of the components of the Tethys software framework together and provides tools to aid in rapid development of apps. The SDK is implemented as a plugin to the CKAN data management system, which in turn is powered by the Pylons Python web framework. As such, apps are developed using the foundation provided by CKAN and Pylons via the SDK, which provides functionality such as user access management, data management, and a Model View Controller (MVC) framework for building the dynamic web pages that make up the web app.

The software components of Tethys are incorporated into apps through various Python modules that come installed with the SDK. The SDK also takes advantage of the abilities of templating by providing a method for inserting common user interface elements (maps, plots, form elements) using only a few lines of code. Furthermore, all apps are able to draw on the existing scientific modules available to the Python language.

3 USE CASES

Using the Tethys software framework, we have developed several prototype web apps for cloud-based modeling, including a snowmelt simulator and an app to analyzing changes in runoff due to land use changes.

3.1 Utah Energy Balance Model

The Utah Energy Balance (UEB) snowmelt model [Tarboton et al., 1995; Mahat & Tarboton, 2012] is a distributed model for the simulation of snowmelt surface water inputs used to address water availability in snowmelt driven environments. UEB evaluates in parallel the snowmelt at each grid cell over a watershed [Sen Gupta & Tarboton, 2013]. The UEB App built using the Tethys software framework provides the functionality to generate a data input package that contains all necessary input files to run UEB. The UEB model parameters are configured using a web interface comprising forms specifying data sources for inputs and to optionally change parameters from their defaults. When a user submits a request to generate a model input data package the UEB App sends that request to our hosted Application Server (App Server) through a HTTP POST web service call. Information is transmitted as a JSON string along with user selected configuration input files contained in a zip file. The App Server runs a series of python scripts that prepare input files by extracting required data from the input sources and formatting input files as required by UEB. This results in a zip file that contains all input files necessary to run UEB for the user specified domain. When the model input data package is received from the App Server, it is saved as new resource inside a new dataset of type ‘model-package’ in the CKAN data publishing system. Figure 2 shows the dependency of UEB App on various components.

The App Server currently hosts the following data sources necessary to generate UEB input data:

- DEM (Greater Salt Lake area)
- Daymet time series data (for Utah and for year 2011)
- NLCD (USA)

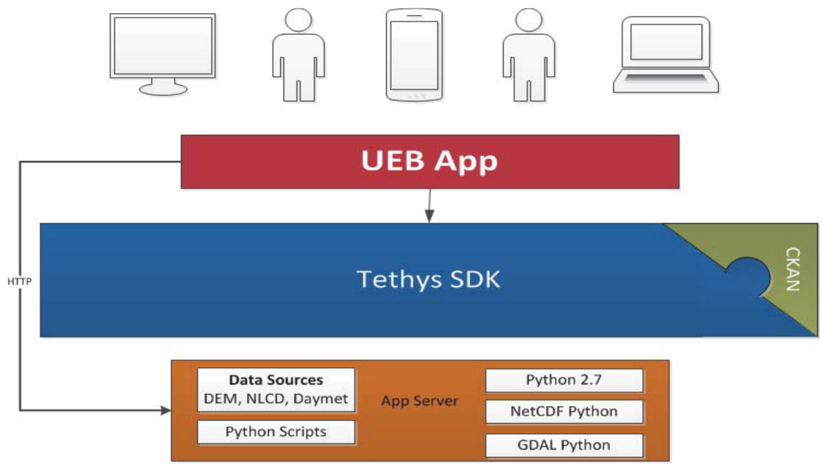

Figure 2. UEB App dependency diagram.

3.2 GSSHA Land Use Change

Gridded Surface Subsurface Hydrologic Analysis (GSSHA) is a distributed hydrologic model developed and maintained by the United States Army Corp of Engineers [Downer & Ogden, 2004]. Spatially varying watershed characteristics can be represented in GSSHA using GRASS ASCII rasters. A common analysis that can be performed using GSSHA is a land use change analysis that may be prompted by a proposal for developing an undeveloped parcel.

We developed a web app that automates the land use analysis (see Figure 3) using the Tethys SDK. Upon launching the app, the user is prompted to select a previously uploaded GSSHA model package or upload their own. The user is prompted to select the map they wish to edit, which is displayed using an editable Google Maps™ map. The editable map allows the user to draw the areas affected by the change or upload a shapefile. After editing the map the user is walked through a series of steps for assigning hydrologic characteristics for the changed areas. Finally, the user can save the scenario, run it, and view the results.

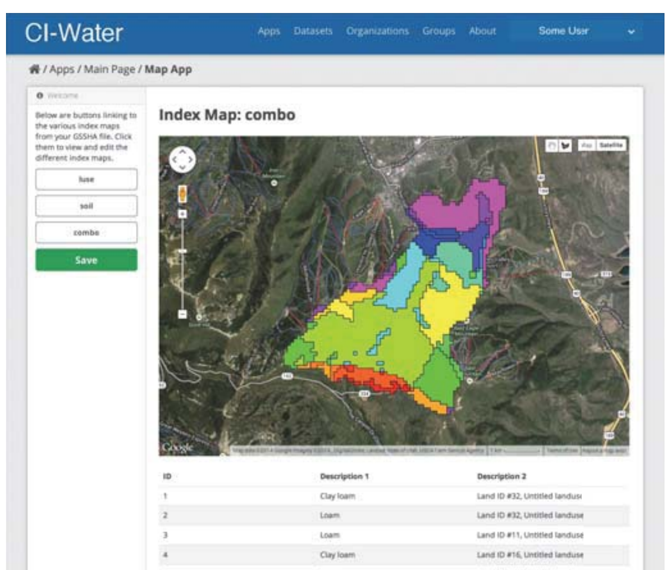

Figure 3. Screenshot of land use change app developed with Tethys SDK.

Behind the scenes, the land use change app uses PostGIS to store and geoprocess the land use map, GeoServer to publish the KML visualizations of the rasters, Google Maps™ and Highcharts JS snippets for visualizing the data, and a custom 52°North service for executing the GSSHA model. In addition, the app uses GsshaPy, which is a custom SQLAlchemy object relational model that is capable of reading GSSHA file packages into an SQL database and vica versa. When the user selects a GSSHA input package, it is read into the PostGIS database using GsshaPy. Modifications to the model are recorded in the database. The model is then written back to file from the database prior to execution.

4 CONCLUSIONS

The Tethys software framework is a scripting environment that facilitates the development of cloud-based modeling tools for water resource modeling applications. These tools can be used to take an existing model and host it in a format that facilitates use of the model as a decision support tool in the context of water resource management or emergency response. In some cases, it can be used to provide a complete modeling environment where a user can leverage online date resources to build models from scratch. In addition to ease of use, cloud-based modeling systems avoid many of the difficulties associated with traditional software development and distribution and they provide access to potentially unlimited computational resources via distributed computing. Further information about this project can be found at www.ci-water.org.

5 ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1135483.

6 REFERENCES

52°North. 2014. Home - 52°North Initiative for Geospatial Open Source Software GmbH. http://52north.org/ (last accessed 12 March 2013)
Chambers, J. 2013. The R Project for Statistical Computing. http://www.r-project.org/ (last accessed 12 March 2013)
Downer, C.W. and F.L. Ogden, 2004. GSSHA: A model for simulating diverse streamflow generating processes. J. Hydrol. Engrg. 9(3), 161-174.
Douglas Thain, Todd Tannenbaum, and Miron Livny, 2005. "Distributed Computing in Practice: The Condor Experience" Concurrency and Computation: Practice and Experience, 17(2-4), 323-356
Hazzard, E. 2011. OpenLayers 2.10 : beginner's guide. Birmingham: Packt Publishing.
Highsoft AS. 2014. Highcharts - Interactive JavaScript charts for your web projects. 2014, http://www.highcharts.com/ (last accessed 12 March 2013).
Holl, S., & Plum, H. 2009. PostGIS. GeoInformatics, 03/2009, 34-36. doi: citeulike-article-id:4463470
Iacovella, S., & Youngblood, B. 2013. GeoServer Beginner's Guide: Packt Publishing.
Mahat, V., Tarboton, D.G., 2012. Canopy radiation transmission for an energy balance snowmelt model. Water Resour. Res. 48 W01534.
Nguyen, T. T. 2009. Indexing PostGIS Databases and Spatial Query Performance Evaluations. International Journal of Geoinformatics, 5(3), 1-9.
OGC. 2014. Open Geospatial Consortium | OGC(R). 2014, http://www.opengeospatial.org/ (last accessed 12 March 2013).
Open Knowledge Foundation. 2014. ckan - The open source data portal software. 2014, http://ckan.org/ (last accessed 12 March 2013)
Python Software Foundation. 2013. Python. http://python.org/about/ (last accessed 12 March 2013)
Schut, P. 2007. OpenGIS Web Processing Service. (pp. 87): Open Geospatial Consortium Inc., Wayland, MA, USA.
Sen Gupta, A., Tarboton, D.G., 2013. Using The Utah Energy Balance Snow Melt Model To Quantify Snow And Glacier Melt In The Himalayan Region, Proceedings of the Western Snow Conference 81st Annual Meeting. Adaptive Water Management in a Changing Climate: Jackson Hole, Wyoming, pp. 103-114.

Steiniger, S., & Hunter, A. J. S. 2012. Review: The 2012 free and open source GIS software map – A guide to facilitate research, development, and adoption. Computers, Environment and Urban Systems. doi: 10.1016/j.compenvurbsys.2012.10.003

Tarboton, D.G., Chowdhury, T.G., Jackson, T.H., 1995. A Spatially Distributed Energy Balance Snowmelt Model, In: Tonnessen, K.A., Williams, M.W., Tranter, M. (Eds.), Biogeochemistry of Seasonally Snow-Covered Catchments (Proceedings of a Boulder Symposium, July 1995). IAHS Publ. no. 228: Wallingford, pp. 141-155.

Thain, D., Tannenbaum, T., & Livny, M. 2005. Distributed computing in practice: the Condor experience. *Concurrency & Computation: Practice & Experience, 17*(2-4), 323.