Reproducible Hydrological Modeling with CyberGIS-Jupyter: A Case Study on SUMMA

Fangzheng Lyu
Department of Geography and Geographic Information Science
University of Illinois of Urbana-Champaign
flu8@illinois.edu

Dandong Yin
Department of Geography and Geographic Information Science
University of Illinois of Urbana-Champaign

Anand Padmanabhan
Department of Geography and Geographic Information Science
University of Illinois of Urbana-Champaign

Youngdon Choi
Department of Engineering Systems and Environment
University of Virginia
Charlottesville

Jonathan L. Goodall
Department of Engineering Systems and Environment
University of Virginia
Charlottesville

Anthony Castronova
Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI)

David Tarboton
Utah Water Research Laboratory
Civil and Environmental Engineering
Utah State University

Shaowen Wang†
Department of Geography and Geographic Information Science
University of Illinois of Urbana-Champaign shaowen@illinois.edu

ABSTRACT

CyberGIS-Jupyter is a cyberGIS framework for achieving data-intensive, reproducible, and scalable geospatial analytics using Jupyter Notebook based on advanced cyberinfrastructure. As a cutting-edge hydrological modeling framework, the Structure for Unifying Multiple Modeling Alternative (SUMMA) functions as a unified approach to process-based modeling. The purpose of this research is to investigate the feasibility of coupling CyberGIS-Jupyter with SUMMA to realize reproducible hydrological modeling. CyberGIS-Jupyter is employed to systematically integrate advanced cyberinfrastructure, including two high-performance computers – Virtual ROGER and XSEDE Comet, data management, and execution and visualization of SUMMA-based modeling. By taking advantage of CyberGIS-Jupyter, users can easily tune different parameters for a SUMMA model and submit computational jobs for executing the model on HPC resources without having to possess in-depth technical knowledge about cyberGIS or cyberinfrastructure. Computational experiments demonstrate that the integration of CyberGIS-Jupyter and SUMMA achieves a high-performance and easy-to-use implementation for reproducible SUMMA-based hydrological modeling.

KEYWORDS

CyberGIS-Jupyter, Geographic Information Science and Systems (GIS), High-Performance Computing, Hydrological Modeling, Science Gateway

1 Introduction

With rapidly advancing cyberinfrastructure and increasing computing power available, computationally intensive and sophisticated hydrological modeling can be conducted for a wide variety of applications ranging for example from flood mapping to understanding the complexity of water system dynamics across various spatial and temporal scales. However, as such modeling is becoming increasingly computation-, data-, and collaboration-intensive, significant challenges must be resolved to enable reproducible modeling workflows based on advanced cyberinfrastructure.

This research aims to demonstrate reproducible hydrological modeling through integrating CyberGIS-Jupyter and the Structure for Unifying Multiple Modeling Alternative (SUMMA) [3] based on advanced cyberinfrastructure. CyberGIS-Jupyter is an online geospatial computation platform for a large number of users to conduct and share scalable cyberGIS analytics via Jupyter Notebooks supported by advanced cyberinfrastructure resources [10] such as those provisioned by the Extreme Science and Engineering Discovery Environment (XSEDE). SUMMA is a hydrological modeling tool that is built on a common set of conservation equations and a common numerical solver, which together constitute the structural modeling core for enabling a controlled and systematic analysis of alternative modeling options.

To achieve flexible modeling options as a strength of SUMMA, it is important to support reproducible workflows that can be executed and shared using advanced cyberinfrastructure resources. CyberGIS-Jupyter allows seamless access to high-performance computing (HPC) resources while shielding the complexity of managing cyberinfrastructure access from hydrological modeling users. In this research, we focus on integrating CyberGIS-Jupyter and SUMMA to enable SUMMA models to harness the computational power of both Comet (an HPC resource on XSEDE) and Virtual ROGER (a cyberGIS supercomputer hosted by the CyberGIS Center for Advanced Digital and Spatial Studies at the University of Illinois) [4].

The integration is evaluated using a SUMMA modeling scenario where a large number of parameter settings need to be examined. A user environment is established within CyberGIS-Jupyter to orchestrate SUMMA model executions, which does not require SUMMA modeling users to possess technical knowledge about cyberGIS [9] or HPC. This user-friendly environment is conducive for repeating model runs by different users and supports collaborative validation of computationally intensive and sophisticated modeling workflows through Jupyter Notebooks [6] enabled by seamless access to advanced cyberinfrastructure and cyberGIS capabilities.

2 SUMMA

SUMMA is a model framework that allows for formal evaluation of multiple working hypotheses on model representations of physical processes and encourages hydrological modeling best practices [3]. However, the lack of capable and accessible cyberinfrastructure tailored for SUMMA has hampered its adoption and engagement in community hydrological modeling. A startup XSEDE allocation was used to investigate solutions for encoding and deploying the SUMMA methodology within CyberGIS-Jupyter to facilitate its broader adoption and use, and thereby elevate the state of practice of hydrological modeling science. A key focus of this work was to make hydrological modeling more reproducible and scalable by supporting modeling in cyberinfrastructure environments where modeling configurations can be managed, moving hydrological modeling from desktop computers, where software configuration for state-of-the-art hydrological models like SUMMA can be a barrier to entry to more scalable computational resources with provenance-aware workflow capabilities [2]. This extends our current capabilities and enables scientists to address hydrological hypothesis in more reproducible ways. To this end, we have made progress on deploying simulations on XSEDE Comet for specific research applications outlined below.

An important step in this research is to advance understanding on the role of both process parameterization and model parameter calibration on hydrological model simulations. One hypothesis is that varying both flux parameterization and model parameter calibration in combination will yield a hydrological model with better predictive capability than calibration using parameter values and expert selection of flux parameterization alone. We executed SUMMA on Comet with a small subset of the possible parameterizations. There are also variations within these parameterizations and further desirable computation is to exhaustively experiment with a myriad of possible combinations of flux parameterizations, in order to advance understanding and improve the predictive capability of this model. The integration with CyberGIS-Jupyter supports reproducible evaluation of these cases in SUMMA that would otherwise be computationally intractable.

3 Architecture

The architecture of the integrated system enables interactions among three key entities: users, CyberGIS-Jupyter, and HPC resources provided through XSEDE (e.g. Comet). In addition, there are four supporting components with which the key entities interact with: 1) external authentication system, 2) HydroShare [1] for hydrological data retrieval, 3) JupyterHub with appropriate cyberGIS and geospatial python libraries installed, and 4) Docker hub for SUMMA singularity image. HydroShare is a collaborative research platform for advancing hydrological data and model sharing [8].

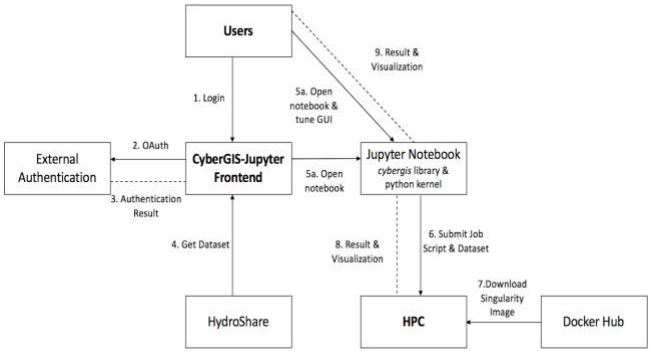

Figure 1: Architecture of CyberGIS-Jupyter for SUMMA computation management

Figure 1 highlights the nine major interactions that occur between the key entities and supporting components. In order to submit jobs to HPC through CyberGIS-Jupyter and fetch back the results of SUMMA-based modeling, firstly, a user needs to log into the CyberGIS-Jupyter frontend server. The authentication relies on an external provider, which in our current implementation is provided through Jupyter. Specifically, the user can successfully login into CyberGIS-Jupyter using her/his Github authentication that we connect to via OAuth. Authorization is done using a whitelist of valid GitHub usernames that are maintained by CyberGIS-Jupyter. Once successfully authenticated, the user has access to SUMMA datasets downloaded from HydroShare and cyberGIS library installed to serve as middleware to manage the interactions between CyberGIS-Jupyter and HPC resources. The

user is then able to open a Jupyter notebook within Python kernel and access the cyberGIS library that provides both a friendly interface and computation management support. The GUI widget provided by the cyberGIS library allows users to tune different modeling parameters. After finishing the parameter configuration, an HPC job script is generated automatically based on an existing template and the user's input. A SUMMA job can then be submitted to an HPC environment along with the job script, parameter configuration, and data from HydroShare.

In the HPC environment, a singularity image of SUMMA that we have tested is pulled from DockerHub. Combined with the input dataset and job script, the SUMMA job is executed. After successful execution the modeling results along with the associated visualization are transferred back to CyberGIS-Jupyter and displayed within the Jupyter notebook. Users can also download these modeling results to their local computing resources from CyberGIS-Jupyter.

4 Implementation

A cyberGIS package, implemented primarily in Python, was designed for users to download modeling cases, submit jobs to HPC, and retrieve back modeling results. There are five key classes involved in the implementation of the cyberGIS package: 1) SUMMA, 2) Auth, 3) SummaVis, 4) Editor, and 5) Floret, with two supporting actors: 1) HydroShare for the input datasets and 2) advanced cyberinfrastructure for high-performance computation. An example UML of cyberGIS package we implemented for SUMMA model was shown in Figure 2.

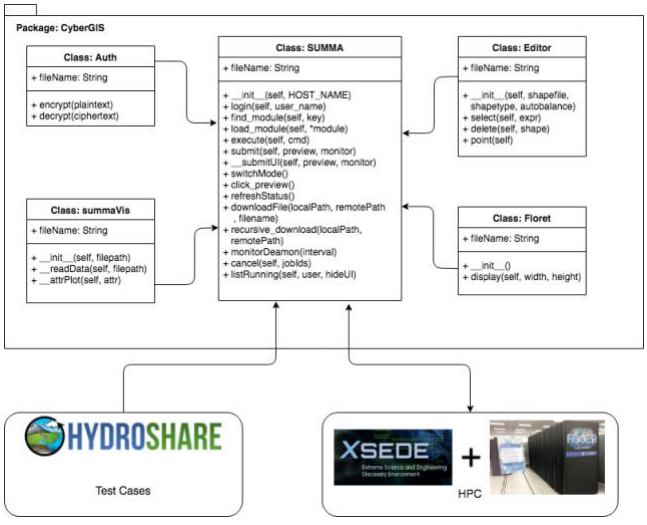

Figure 2: UML of cyberGIS package for SUMMA job submission

Firstly, the class Auth is for authenticating users with two methods *encrypt* and *decrypt* implemented to hide some important authentication information from users. Secondly, the class summaVis, which represents SUMMA visualization, is implemented to visualize SUMMA modeling result. The two methods *readData* and *attrplot* read and plot the results respectively. Furthermore, the Editor class provides read and write support for *ESRI* shapefile. It enables the SUMMA class to understand the datasets from HydroShare. The class Floret functions as a cyberGIS mapping tool and it is based on Leaflet. Most importantly, the class SUMMA integrated all the four classes above and is responsible for retrieving data from HydroShare, generating a GUI widget as well as many functionalities within the GUI widget for users to manipulate, submitting the SUMMA job script and input dataset to HPC, retrieving modeling results from HPC, and at last, visualizing and presenting the results. For example, the method *submitUI* implements a GUI widget on the Jupyter notebook while the *submit* method is responsible for submitting the job script as well as input datasets to HPC.

Apart from the five major classes introduced above, there are two important supporting actors: HydroShare for data downloading and advanced cyberinfrastructure like XSEDE Comet and Virtual ROGER for high-performance computation. HydroShare, as an online platform for sharing hydrological data and models, is adopted as the source of SUMMA input data. We have integrated five typical hydrological datasets into the integrated CyberGIS-Jupyter and SUMMA system. The singularity image of the SUMMA model is downloaded from DockerHub for Virtual ROGER and XSEDE Comet to run the model.

5 Experiment

There are 5 major steps in the experiment: 1) use the external authentication resources to log into the CyberGIS-Jupyter frontend server, 2) open the Jupyter notebook in the server and have parameters tuned in the GUI widget, 3) submit the job and orchestrate different stages of computation management, 4) retrieve the output of the SUMMA model and visualize the corresponding results, and 5) download the result and visualization of the SUMMA model to a user's local computing environment if needed.

In the first step, simply typing the URL for our server with any web browsers (https://hsjupyter.cigi.illinois.edu:8000/ is the server address used in this experiment) will introduce users the external authentication page. One thing worth noting is that the server URL used in this experiment can only be accessed once connecting to the University of Illinois of Urbana-Champaign VPN. However, we are actively setting up servers that will allow more users to log in without having to connect to the University VPN. Figure 3 shows two steps for accessing the CyberGIS-Jupyter frontend server. Once a user goes to the URL for the frontend server, they will be redirected to an external authentication page. After clicking the bottom "Sign in with GitHub", the user can use her/his GitHub username and password to continue to CyberGIS-Jupyter server as long as her/his GitHub account is in the whitelist. Using external authentication instead

of having users create an account will not only have users access the server more conveniently but also can avoid many potential security problems by simply keeping a whitelist for the users.

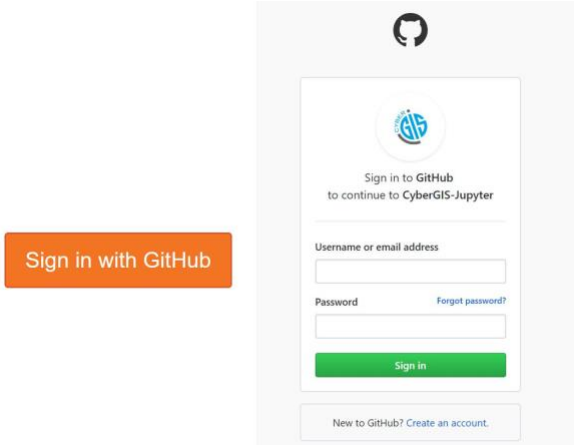

Figure 3: User authentication page

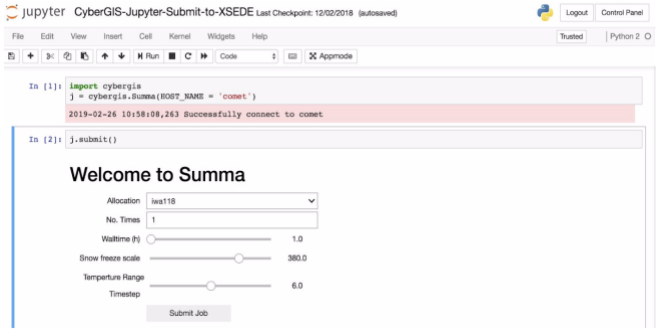

Figure 4: User interface

Secondly, the user interface is implemented in a simple and easy-to-use fashion. As it is shown in Figure 4, after successfully logging in to the server, a Jupyter notebook called CyberGIS-Jupyter-Submit-to-XSEDE will be available for job submission and computation management. In the first box of the notebook, users can import the cyberGIS library and create an object called SUMMA HPC access. While creating a *cybergis.summa* object, a connection between the Jupyter notebook and HPC will be created through a science gateway community account. Calling the submit function of the *cybergis.summa* object will invoke a GUI widget that allows users to configure SUMMA parameters and submit the HPC job. Within the GUI widget, there are five important tunable parameters implemented for users to generate different results from the SUMMA model. The first three parameters are specific to managing cyberinfrastructure access and HPC job execution. The remaining two parameters are for evaluating different SUMMA modeling configurations.

In the third step, we have implemented the functionality for submitting the job script and input dataset from the CyberGIS-Jupyter server to HPC resources. A precise and dynamic job status table is presented to users to make users aware of the current stage of computation progress (e.g., queuing, running, finished). Figure 5 provides one example of the job status table. Job id, the username of her/his supercomputer account, status, name of the job and other useful information can be made available in the table for users to understand the status of high-performance computation, which could also be important for achieving reproducible modeling workflows.

Figure 5: Job status table

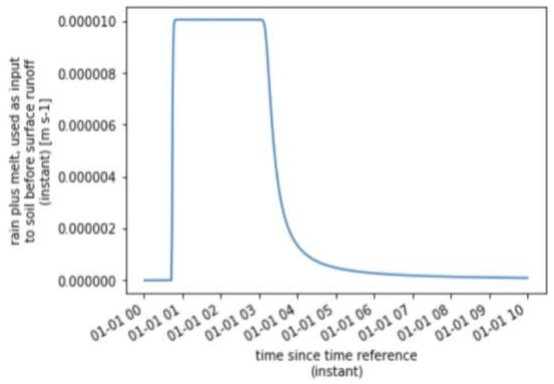

Figure 6: SUMMA job output

Figure 6 shows a simple visualization of one test case of SUMMA modeling conducted on XSEDE Comet. As part of the results we got from the SUMMA hydrological model, the graph represents the relationship between "rain plus melt, used as input to soil before surface runoff (instant) [m s-1]" and "time since time reference data (instant)". This test case took one input dataset "Colbeck1976" and two tunable parameters "snow freeze scale" equal to 380.0 and "temperature range timestamp" set as 6.0.

After submitting modeling jobs and fetching back results, a user may look at the output folder that includes all of her/his modeling results. As shown in Figure 7, there are the results of the SUMMA model for 5 different test datasets retrieved from HydroShare. The user can download those desirable results to her/his own local computer by using basic features of Jupyter Notebook.

Figure 7: SUMMA job output folder

6 Conclusion and Discussion

This study introduced the SUMMA model and the underlying architecture for conducting the computation of SUMMA modeling on advanced cyberinfrastructure through CyberGIS-Jupyter. The implementation of user interface, job script transfer, and computation management was described. We evaluated the modeling workflow based on multiple modeling scenarios to demonstrate computational reproducibility ranging from HPC management to sharing of modeling configurations, steps and outcome. Compared with the traditional methods of executing SUMMA jobs, our approach not only enables users to submit the SUMMA jobs without complicated setup and configuration processes, but also helps users to execute the jobs more effectively and efficiently by exploiting the power of heterogeneous cyberinfrastructure resources.

In the field of hydrology and geospatial sciences, there is a profusion of diverse simulation models, in part because they address a diversity of (unique) scientific problems and questions. However, this hampers the advancement of hydrological understanding because in comparing models that differ in multiple aspects it is difficult to unravel the causes for model differences and to use model inter-comparison as hypothesis testing to advance hydrological understanding. Modeling approaches such as SUMMA have been developed to directly address this fundamental problem and our research tackles this problem by integrating CyberGIS-Jupyter and SUMMA based on advanced cyberinfrastructure.

As research questions continue to grow in both scope and complexity, researchers must acquire and/or provision new advanced cyberinfrastructure resources. We argue that this is a considerable limitation and will be overcome by interfacing Jupyter-based science gateways (e.g. CyberGIS-Jupyter) directly with advanced cyberinfrastructure. Our overarching research goal beyond this study is to bring models, such as SUMMA, because of its flexible framework for supporting hypothesis testing in hydrological modeling, to a point where researchers and students can use these models within their learning work and research to advance understanding of hydrological processes in more systematic, transparent and reproducible ways. These are intended to serve as examples to enable related science communities to establish similar functionality for computation-, data- and collaboration-intensive models.

ACKNOWLEDGMENTS

This research is supported in part by the National Science Foundation (NSF) under grant numbers 1443080, 1664119, and 1743184. We greatly appreciate the XSEDE allocation support through two awards: SES170023 and EAR190007. Our computational work also used Virtual ROGER, which is a cyberGIS supercomputer supported by the CyberGIS Center for Advanced Digital and Spatial Studies and the School of Earth, Society and Environment at the University of Illinois. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

REFERENCES

[1] Castronova, A. M., L., & Seul, M (2017). Cloud-based Jupyter Notebooks for Water Data Analysis. Presented ata the AGU Fall Meeting Abstracts. Retrieved from https://ui.absabs.harvard.edu/abs/2017AGUFMIN11D..03C

[2] Choi, Y., Sadler, J.M. Castronova, A. M., Goodall, J. L., Bennett, A., Nijssen, B., … Tarboton, D. G. (2018). The Development of Shareable pySUMMA Simulation Environment using Singularity on HydroShare (Vol. 2018). Presented at the AGU Fall Meeting Abstracts. Retrieved from https://ui.adsabs.harvard.edu/abs/2018AGUFMED53C..16C

[3] Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., … Others. (2015). The structure for unifying multiple modeling alternatives (SUMMA), Version 1.0: Technical description. NCAR Tech. Note NCAR/TN-5141STR. Retrieved from http://opensky.ucar.edu/islandora/object/technotes%3A526/datastream/PDF/download/citation.pdf

[4] Connecting to Virtual ROGER. (2018, August 3). Retrieved April 10, 2019, from http://cybergis.illinois.edu/infrastructure/hpc-user-guide/connecting-to-roger/

[5] Home - XSEDE. (n.d.). Retrieved April 10, 2019, from https://www.xsede.org/

[6] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., … Others. (2016). Jupyter Notebooks-a publishing format for reproducible computational workflows. In ELPUB (pp. 87–90).

[7] Kouwen, N., Soulis, E. D., Pietroniro, A., Donald, J., & Harrington, R. A. (1993). Grouped response units for distributed hydrologic modeling. Journal of Water Resources Planning and Management, 119(3), 289–305.

[8] Tarboton, D. G., Idaszak, R., Horsburgh, J. S., Heard, J., Ames, D., Goodall, J. L., … Maidment, D. (2014). HydroShare: Advancing Collaboration through Hydrologic Data and Model Sharing. In International Congress on Environmental Modelling and Software. Retrieved from https://scholarsarchive.byu.edu/iemssconference/2014/Stream-A/7/

[9] Wang, S. (2010). A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis. Annals of the Association of American Geographers. Association of American Geographers, 100(3), 535–557.

[10] Yin, D., Liu, Y., Hu, H., Terstriep, J., Hong, X., Padmanabhan, A., & Wang, S. (2018). CyberGIS-Jupyter for reproducible and scalable geospatial analytics. Concurrency and Computation: Practice & Experience, 308, e5040.