A CyberGIS-Jupyter Framework for Geospatial Analytics at Scale

Dandong Yin
Geography and Geographic Information Science
CyberGIS Center for Advanced Digital and Spatial Studies
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
dyin4@illinois.edu

Yan Liu
Geography and Geographic Information Science
CyberGIS Center for Advanced Digital and Spatial Studies
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
yanliu@illinois.edu

Anand Padmanabhan
Geography and Geographic Information Science
CyberGIS Center for Advanced Digital and Spatial Studies
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
apadmana@illinois.edu

Jeff Terstriep
CyberGIS Center for Advanced Digital and Spatial Studies
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
jefft@illinois.edu

Johnathan Rush
CyberGIS Center for Advanced Digital and Spatial Studies
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
jfr@illinois.edu

Shaowen Wang
Geography and Geographic Information Science
CyberGIS Center for Advanced Digital and Spatial Studies
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
shaowen@illinois.edu

ABSTRACT

The interdisciplinary field of cyberGIS (geographic information science and systems (GIS) based on advanced cyberinfrastructure) has a major focus on data- and computation-intensive geospatial analytics. The rapidly growing needs across many application and science domains for such analytics based on disparate geospatial big data poses significant challenges to conventional GIS approaches. This paper describes CyberGIS-Jupyter, an innovative cyberGIS framework for achieving data-intensive, reproducible, and scalable geospatial analytics using the Jupyter Notebook based on ROGER - the first cyberGIS supercomputer. The framework adapts the Notebook with built-in cyberGIS capabilities to accelerate gateway application development and sharing while associated data, analytics and workflow runtime environments are encapsulated into application packages that can be elastically reproduced through cloud computing approaches. As a desirable outcome, data-intensive and scalable geospatial analytics can be efficiently developed and improved, and seamlessly reproduced among multidisciplinary users in a novel cyberGIS science gateway environment.

PEARC17, July 09-13, 2017, New Orleans, LA, USA

ACM ISBN 978-1-4503-5272-7/17/07…$15.00
https://doi.org/10.1145/3093338.3093378

CCS CONCEPTS

• **Information systems** → **Geographic information systems**;
• **Computer systems organization** → *Distributed architectures*;
• **World Wide Web** → Web services;

KEYWORDS

CyberGIS, computational reproducibility, geospatial big data, flood mapping, science gateway

ACM Reference format:
Dandong Yin, Yan Liu, Anand Padmanabhan, Jeff Terstriep, Johnathan Rush, and Shaowen Wang. 2017. A CyberGIS-Jupyter Framework for Geospatial Analytics at Scale. In *Proceedings of PEARC17, New Orleans, LA, USA, July 09-13, 2017,* 8 pages.
https://doi.org/10.1145/3093338.3093378

1 INTRODUCTION

From the late 1990s, cyberinfrastructure has been playing increasingly important roles in mainstream scientific discoveries [5]. Dedicated to reduce its barriers to broad scientific communities, science gateways have achieved significant impacts on numerous scientific domains including particle physics [3], molecular chemistry [10], public health [1] and many others.

In geospatial domains, cyberGIS - geospatial information science and systems (GIS) based on advanced computing and cyberinfrastructure (CI) [22] - has enabled computation- and data-intensive knowledge discovery by gaining unprecedented insights into the complex and geospatially connected world from both natural and social sciences perspectives [12, 23, 28]. Pushing the frontiers of

science gateway, a suite of cyberGIS gateway applications [6, 7, 11] have been developed to simplify access to advanced cyberGIS and cyberinfrastructure by providing interactive, online interfaces to scalable geospatial analytics. However, as the diversity of cyberGIS data and applications keeps growing, cyberGIS-enabled geospatial analytics poses challenges against traditional web-based gateway approaches.

Due to the high variety and complexity of cyberGIS analytics, it is difficult to provide a comprehensive solution in a single gateway application mimicking traditional desktop GIS. Therefore, it is typical for each gateway application to focus on a specific type of analytics (e.g. CyberGIS-BioScope [7] for biomass-to-biofuel supply chain system optimization; FluMapper [16] for mapping the spread of influenza-like illnesses from Twitter data; and TopoLens [6] for accessing high-resolution national topographic datasets).

Given the enormous application space of cyberGIS, agile development for new gateway applications is urgently needed. In most desirable cases, domain researchers should be able to implement and customize their unique needs for gateway applications, instead of depending on dedicated developers. However, most traditional cyberGIS gateways were developed with web GIS, i.e. GIS system that adopts browser/server architecture, typically with interactive graphic user interfaces as front end and a set of dedicated services in the back end. In traditional web GIS development cycles, it is difficult to achieve agile development, especially for domain researchers. There are three main reasons are as follows:

(1) Developing a complete web application with interactive graphical user interfaces (GUIs) from scratch requires professional skills that most of geospatial researchers do not possess;
(2) Handling large-scale computation with middleware orchestration behind front-end interfaces requires substantial knowledge of cyberGIS and HPC
(3) Operational maintenance, including deployment, user and data managements, etc. pose significant overheads for common researchers.

As a result, the intensive development requirements of web GIS becomes a major bottleneck to meet the proliferating needs of cyberGIS capabilities from domain researchers. Therefore, to fully leverage the power of cyberGIS, it is necessary not only to reduce the barrier of accessing cyberGIS via gateway applications, but also to reduce the barrier of developing gateway applications so that common researchers could efficiently construct their own applications from basic building blocks.

One approach to resolving aforementioned issues for gateway application development is to couple the development and use of a gateway application as an integrated platform that engages researchers, developers, and users and couples the computing, storage, and software resources as project resources for easy access from them. It has been recognized by academic computing communities that "the distinction between *users* and *developers* is actively harmful"[20]. An integrated gateway application platform can effectively fill this gap, and most recent advances in Jupyter (http://jupyter.org/) provides a promising solution. The increasing support of interactive extensions enables Jupyter to serve as a user-friendly application interface as well as an agile development environment. Meanwhile, the emergence of JupyterHub (https://jupyterhub.readthedocs.io) makes it possible to deploy Jupyter on distributed infrastructure.

This paper describes CyberGIS-Jupyter, an innovative framework that integrates cloud-based Jupyter notebooks (highly interactive read-eval-print loop (REPL) environment) [9] with HPC resources as part of a hybrid computing environment [14, 18]. The framework addresses the development challenge as follows: 1) adopts Jupyter notebooks instead of web GIS as the front end interface to provide a consistent and agile playground for both developers and users; 2) encapsulates advanced cyberGIS capabilities within a pre-configured and containerized environment; 3) achieves on-demand provisioning through cloud computing to elastically deploy and manage multiple instances of gateway applications. Furthermore, the reproducible deployment enables researchers to share and build on each other's work to innovate large-scale geospatial analytics cumulatively in a collaborative fashion. With this framework, community-driven gateway development and deployment becomes feasible. To the best of our knowledge, our work provides the first general framework to modularize gateway development and deployment for domain researchers.

The remainder of this paper is organized as follows. Section 2 examines the related work of CyberGIS-Jupyter; Section 3 presents the design and architecture of CyberGIS-Jupyter; Section 4 demonstrates the framework's agility in transforming complex cyberGIS computation into interactive gateway applications with a case study of computing height above nearest drainage (HAND) computation at 10m resolution for conterminous US (CONUS); and Section 5 concludes the paper and discusses future work.

2 RELATED WORK

To lower the barrier of entry to CI, a series of science gateways have been developed to enable computation- and data-intensive research and education [11, 25, 26]. In order to provide easy access, most science gateways adopt the Software as a Service (SaaS) [21] approach, i.e. providing applications through interactive web services with Web 2.0 technology and Service-Oriented Architecture (SOA) [13]. A similar architecture was adopted for cyberGIS gateways [11]. On top of the generic science gateway architecture, a typical geospatial gateway usually adopts web-based GIS capabilities such as OpenLayers (http://openlayers.org) as frontend interfaces; uses geospatial middleware (e.g. GISolve [24]) to access HPC capabilities; and imports/exports standard geospatial web services via OGC standards (http: //opengeospatial.org/). To achieve desirable friendliness for end users, the development of web GIS based gateways requires specialized web development and design skills and is time-consuming. As the need for customized cyberGIS capabilities proliferates, the web GIS gateways become a bottleneck to meeting the vast and dynamic requirements of cyberGIS applications.

Jupyter, formerly known as IPython [17], is a language agnostic platform designed for interactive data-driven discovery based on scripting languages. A Jupyter notebook is an interactive extension of the read-eval-print loop (REPL) environment, where the evaluation result could be a fully functional interactive HTML elements including plain text, tables, figures, animations and webmaps. As an open-source project, Jupyter is highly configurable

and extensible, which attracts many community empowered enhancements, such as generating interactive HTML widgets (https://github.com/ipython/ipywidgets) and embedding cartographic maps (https://github.com/pbugnion/gmaps). With Jupyter notebooks, an interactive, lightweight online interface for geospatial analytics and visualization becomes feasible. In CyberGIS-Jupyter, we developed a set of interactive tools for cyberGIS capabilities, including HPC job management, geospatial visualization and integration with other cyberGIS components. It is worth noting that the HPC community has conducted exploratory work on integrating Jupyter, e.g. the remote spawner initially developed at the San Diego Supercomputer Center (SDSC) (https://github.com/zonca/remotespawner) launches Jupyter servers as standalone HPC jobs; and Wrangler [8] uses Jupyter as part of the visualization portal. However, these efforts merely connect Jupyter notebooks with HPC environments in straight-forward fashions, leaving substantial space for innovation. Meanwhile CyberGIS-Jupyter is designed to serve as a complete solution for developing and sharing cyberGIS gateway applications, including front-end interaction, geospatial visualization, hybrid HPC integration and an framework for application development and deployment.

As the on-demand resource-provisioning feature of cloud computing becomes increasingly available, the traditional HPC with high-end computing capability is beneficial to be integrated with cloud computing resources. Consequently, hybrid HPC, which leverages advanced cyberinfrastructure as a synergistic stack consisting of traditional HPC, scalable databases, cloud environments as well as big data computation frameworks like Hadoop and Spark, represents a new frontier of advanced CI [14, 18]. Similar approaches can also be found in XSEDE resources such as Bridges (http://psc.edu/bridges) and JetStream (http://jetstream-cloud.org/). In geospatial domains, exploration of geospatial big data management and analysis in cloud computing environments has also been actively discussed [27] and deployed (e.g. ESRI ArcGIS Online as a complete cloud-based platform for spatial analysis and map sharing). However, given the tremendous heterogeneity of various hybrid HPC components, it is non-trivial to integrate these capabilities into science gateways without increasing development complexity significantly. In CyberGIS-Jupyter, we deploy the framework directly on top of a cloud platform to leverage on-demand provisioning capabilities. With service containerization [15], the framework is able to encapsulate various hybrid HPC services in a pre-configured container image, so that domain researchers could exploit hybrid HPC services via a simple interface to meet their customized needs. The framework is currently deployed on ROGER, the first cyberGIS supercomputer dedicated for geospatial problem solving (http://go.illinois.edu/ROGER). Leveraging the hybrid HPC design, ROGER provides a desirable environment for developing technical solutions for geospatial analytics at scale.

3 FRAMEWORK

3.1 Key Characteristics

3.1.1 Agility. Due to the data disparity and computational intensity [22] of cyberGIS-enabled geospatial analytics, it is difficult to provide a comprehensive graphical user interface (GUI) serving all analytical components. The current practice of developing customized and web-based GUI interfaces for webGIS-based gateways is often costly. As an alternative to the two approaches, we propose to use Jupyter notebook as an agile GUI development platform for cyberGIS.

Using ipywidgets, Jupyter's Interactive HTML Widgets library, we developed a set of widgets-based utilities to cover common cyberGIS operations. These include a HPC job management interface (Figure 2) and a web-based geospatial visualization solution supporting the widely-adopted standards of tile mapping services (TMS) and web mapping services (WMS) (Figure 4).

Similar kinds of interfaces could be adopted for customized cyberGIS functions, which would enable agile GUI development. Whenever a new cyberGIS analytics is built, the developers could come up with a simple widget interface to support parameter tuning of the related computational model, followed by invoking builtin cyberGIS widgets for HPC and geospatial visualization.

3.1.2 Reproducibility. The Jupyter notebook is designed as a publishing format for reproducible computational workflows [9]. The primary purpose of a notebook file is to maintain a record of workflow execution. Therefore, Jupyter also supports saving and loading the status of widgets along with the notebook file. Additionally, in CyberGIS-Jupyter, we also serialize corresponding status and parameters of cyberGIS functionalities embedded in the notebooks (e.g. the configuration file of HPC jobs).

However, notebooks are merely a record of computation programs. Computational reproducibility requires computational environments, including the exact versions of all external libraries, to be recorded and reproducible [4, 19]. To further record and reproduce computational environments, CyberGIS-Jupyter adopts the container virtualization technology provided by Docker [15]. A container is a lightweight virtualized environment executing on a host system, which could be further serialized/snapshot with its entire computation environments recorded as an image and reproduced elsewhere. A container is created from an immutable image file. An image is created from a configuration file that explicitly lists the steps required to build the image including specific versions of all dependent software. These features have attracted considerable attention from the computational reproducibility research community [2].

In CyberGIS-Jupyter, we deploy Docker inside cloud infrastructure, and for each user instance, their Jupyter notebooks are hosted inside one dedicated container. With the combination of Docker and Jupyter notebook, CyberGIS-Jupyter aims to achieve computationally reproducible geospatial analytics.

3.2 Application

Instead of serving a single dedicated application, CyberGIS-Jupyter provides a platform where cyberGIS applications can be agilely developed and deployed. As shown in Figure 1, each cyberGIS application is standardized as an application package, which can be deployed on the cloud as application instances to serve a large number of users.

Each application consists of a core cyberGIS program, a docker image and a stack of Jupyter notebooks. The core cyberGIS program can take input from any available geospatial data services including

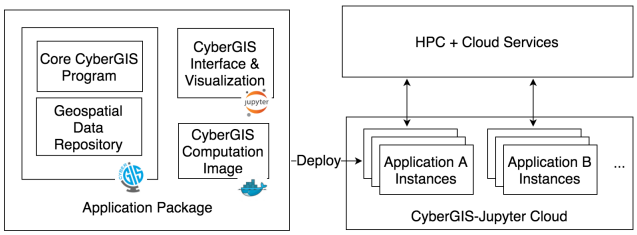

Figure 1: CyberGIS-Jupyter application and deployment

parallel file systems or scalable databases; the program itself can be any type of HPC computation including MPI/OpenMP, GPU or hybrid, and its output address can be files or web mapping services including TMS and WMS. A template application package encapsulating core cyberGIS functionalities are developed to form a basis for customized applications. The major functionalities in the template application package are as follows.

3.2.1 CyberGIS Computation Image. The template image contains various geospatial software (i.e., GDAL, Esri FileGDB extension, PROJ4, GEOS, NetCDF4/HDF4, SpatialLite/SQLite) and parallel computing libraries (i.e., MPI and GNU Parallel), as well as a functional Jupyter notebook server with fundamental interfaces to exploit cyberGIS capabilities, and a pre-configured system for accessing geospatial services (e.g. web mapping service for visualization). Containers could be launched from the template image for users to perform computation and customization, including installing/uninstalling libraries, manipulating local data sets and execute lightweight computation. A customized container can further be committed as a flexible image, forming the basis for developing new cyberGIS applications.

3.2.2 CyberGIS Interface. The cyberGIS user interfaces contain a set of widgets that are pre-configured to provide user-friendly access to HPC resources. These widgets are easy to invoke inside notebooks, providing automatic job submission/management with HPC computation in the backend. Results from HPC computation are made available in near real-time within notebooks using a Network File System (NFS) protocol from HPC file systems to the container. At the end of HPC computation, additional visualization and data analysis could be further performed using the notebooks.

3.2.3 Geospatial Visualization. Geospatial visualization services such as tile mapping services (TMS) and web mapping services (WMS) are compatible within Jupyter notebooks. Mapping services hosted on external servers can be accessed remotely by the notebook using IFrame (Figure 4); Intermediate results generated inside the notebooks can also be visualized with the computational resources of the container using our innovative Python library *Floret*, which provides a user-friendly API to fundamental web-mapping services for mixed vector/raster geospatial datasets (Figure 5). Its raster function is seamlessly integrated with the GDAL tiling tools to enables on-spot web-mapping data processing and service hosting. Compared with the existing Python geospatial visualization library *Folium* (https://github.com/python-visualization/folium) which mostly focuses only on vector data, *Floret* bridges the gap of in-notebook raster data visualization along with vector

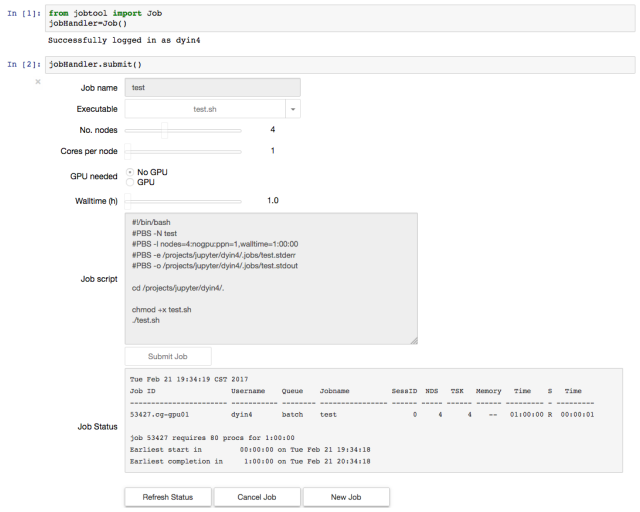

Figure 2: Embedded interface for job submission inside Jupyter notebook.

data, hence enables more comprehensive geospatial analytics to be done in notebooks.

3.3 Architecture

As is depicted in Figure 3, the system has four layers of structures ranging from users to HPC clusters. The major services between end users and HPC computation environments are as follows.

3.3.1 Proxy. The proxy serves as the entry point for all users. Users are directed to the entry address where their requests get handled by the proxy. As the first step, the proxy directs users to the JupyterHub server with authentication and authorization properly configured. Later in the process, the proxy forwards requests directly to Jupyter notebooks running in distributed containers.

3.3.2 JupyterHub. JupyterHub is a central management unit for handling authentication and scheduling standalone Jupyter servers. In CyberGIS-Jupyter, we customize the authentication module to provide an identity verification solution in the HPC context. As users are granted access to actual supercomputer resources, proper identity management is critical for system security and accounting purposes. In the current deployment, authentication is delegated to NCSA's LDAP directory service. After authentication, authorized users are associated with a ROGER account with a quota for data storage and computation resources. In the meantime, a request for launching users' dedicated containers are sent to the Docker swarm.

3.3.3 Docker Swarm. Docker swarm is responsible for spawning and managing all Docker containers across a specific group of virtual machines (VMs) (the swarm), which is a group of instances on Openstack cloud services. The cloud environment with containerization provides fine-grain on-demand provisioning of infrastructure as a service. With the two-level (Openstack and Docker) virtualization, the system achieves:

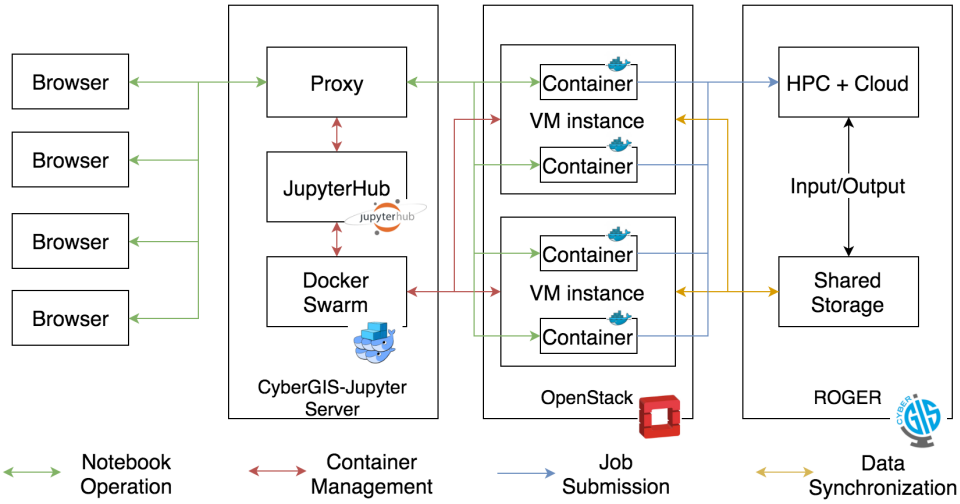

Figure 3: CyberGIS-Jupyter service architecture on ROGER

- **Dynamical load-balancing:** Docker Swarm manages the number of containers active on a VM. When a user launches a notebook, the most idle VM will be chosen to spawn a Docker container with the user's notebooks.
- **On-demand provisioning:** The capacity of the system can be controlled by adjusting the number and size of VMs controlled by Docker Swarm. VMs can be created and added to the swarm to meet increased demand.

3.3.4 Batch Computation. Batch job system is deployed on ROGER HPC to handle computationally intensive tasks sent from the containers. Batch jobs are able to leverage parallel computing resources, large high-performance storage systems, and a variety of geospatial and scientific software to greatly expand the capabilities of a typical Jupyter notebook environment.

3.3.5 Shared Storage. Instead of possessing local storage space on the cloud, users are assigned a dedicated storage space on ROGER's clustered file system, which is exposed as NFS to the OpenStack VM that the user's Docker container is hosted on. From the VM, the storage space is mapped into the notebook container using a Docker host volume. The storage space on ROGER's clustered file system is also accessible by JupyterHub to be loaded as the home directory for Jupyter users. In this way, we achieved:

- **Volume expansion:** as the volume of storage space is not limited by the cloud resources, users are able to access and operate on geospatial big data without losing the agility of virtual computational environment.
- **Real-time access:** data produced by the batch job is immediately available within notebooks. Likewise, data produced in notebooks is available from the batch HPC system.
- **Fault tolerence:** user data are not be affected by potential failures of notebook, container or VM.
- **Collaborative sharing:** by configuring group access permissions on cloud directories, collaborators are able to view, edit, upload and download team resources including codes, notebooks and data in the same project directory.

4 CASE STUDY

The effectiveness of CyberGIS-Jupyter has been studied in enabling and enhancing the collaboration of the National Inundation Mapping Experiment (NFIE) [12] by engaging distributed research teams, sharing data and software resources, and conducting reproducible flood inundation mapping methods and their educational use.

4.1 NFIE HAND collaboration

NFIE is a multidisciplinary collaboration for innovating national-scale flood forecasting and mapping capabilities, conducted by the U.S. National Water Center in partnership with pertinent research communities. A key ongoing effort of NFIE is to produce the 10m resolution Height Above Nearest Drainage (HAND) geospatial raster dataset that serves as a basis for mapping river segment scale flood inundation. The pursuit of HAND requires close collaboration in methodology development, software development, computation, validation, and education and outreach. The team members are from 7 institutions and funded by multiple agencies and industry. The HAND experiment in year 2016 is a representative example of transforming hydrological research from studying local watersheds by a single research group to continental hydrology research involving multiple research teams in cyberGIS, hydrology, hydraulics, and government agencies. The computation of HAND needs to handle the 10m USGS 3DEP national elevation dataset (about 180 billion raster cells) and the National Hydrography Dataset (NHDPlus; http://www.horizon-systems.com/nhdplus/) with approximately 2.67 million stream reaches.

The current collaboration in the HAND experiment has already leveraged the advantage of the HPC+Cloud hybrid architecture on the ROGER supercomputer to eliminate the dramatic cost and intolerable turnaround time in data transfer, result sharing, and the management of heterogeneity of multiple computing environments (e.g. local and clusters). The development of the HAND methodology and workflow software took five months. The first 10m HAND dataset

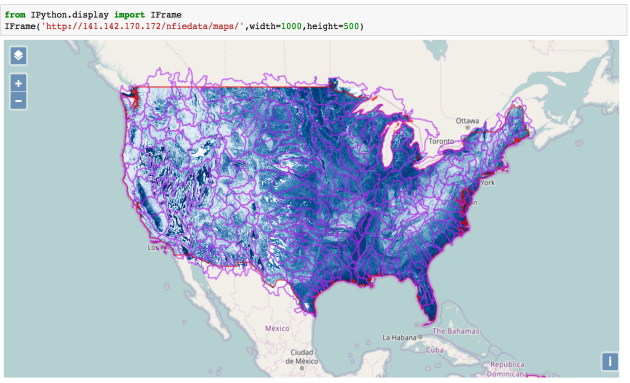

Figure 4: HAND result embedded in Jupyter notebook. The result is calculated at 10m resolution for CONUS from the 10m USGS 3DEP national elevation dataset (NED) and the national hydrography dataset (NHDPlus). The map includes 331 HUC6 units.

for the conterminous U.S. (CONUS) was computed on ROGER in 1.5 days, as shown in Figure 4. Further development and outreach of HAND methodology will incorporate various methods contributed by a broader range of groups in order to solve existing issues and refine the HAND workflow. HAND will also be used heavily in various education and training activities. Retrospecting the current loosely coupled collaboration components, we require a more effective and efficient collaboration solution to meet the next-phase development and community use intensity. CyberGIS-Jupyter is chosen as an enhanced platform for the next-level collaboration. Table 1 lists the comparison of these requirements between the current and the forthcoming way.

The main source of the enhancement using CyberGIS-Jupyter is the ability to share extensive project resources in notebooks so that collaboration team members can contribute simultaneously, including reviewing data, performing interactive investigations, creating visualizations, etc. In the current way, there exists a de facto sequential workflow to develop new components: researchers propose a method and check with each other using documents in emails or meetings; developers start coding on GitHub; new model runs are submitted to HPC resources, model results reviewed internally, results are shared with researchers to validate; and discussion on results are based on local or limited online visualization. Furthermore, the development of a new component often needs multiple rounds of such process, leading to long turnaround time. Using CyberGIS-Jupyter, we aim to complete the development of a new workflow component in a single Jupyter notebook, which can significantly accelerate the collaboration with desirable flexibility, reproducibility, and interactivity required by the team work.

4.2 HAND as a CyberGIS-Jupyter application

The HAND CyberGIS-Jupyter application is developed by containerizing HAND software environment, data, and computation tools. The CyberGIS-Jupyter HAND Dockerfile builds from the template cyberGIS computation image, which includes a common set of open source geospatial software installations. HAND notebooks are then

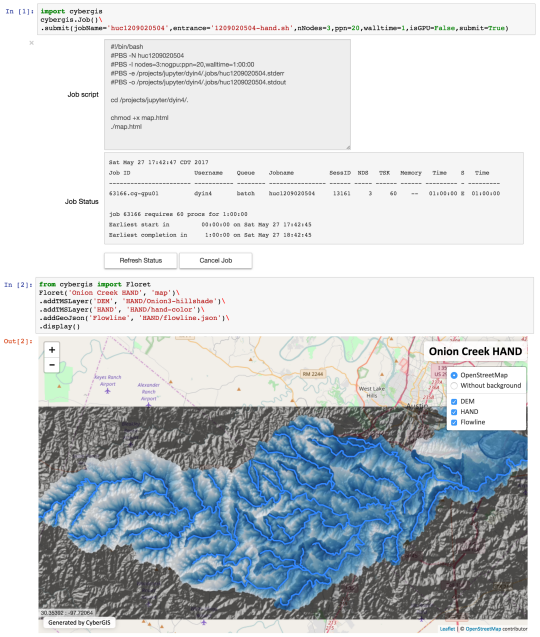

Figure 5: HAND computation and visualization for the Onion Creek watershed in Texas. The map cell view includes three layers: DEM (hillshade view), HAND, and flowlines.

developed and made available within the CyberGIS-Jupyter framework. Specifically, the CyberGIS-Jupyter HAND Dockerfile includes the following features:

- Geocomputation software environment. In addition to the prebuilt softwares in the template cyberGIS computation image, HAND software such as TauDEM and HAND workflow, are configured in Dockerfile to pull a specific version of the source code from GitHub and compile at Docker instance initialization
- HAND input data includes sample data for development purposes and national-level large geospatial datasets for production runs to update HAND. Sample data are stored in the volume storage allocated by OpenStack in Docker host VMs. National datasets such as USGS 3DEP elevation dataset, water boundary dataset, NHDPlus are stored on ROGER GPFS and mounted via NFS, specified in Docker host VM configuration. In addition, external data access APIs are configured to retrieve geospatial and hydrological data from data source portals such as HydroShare (http://hydroshare.org).
- Small-scale computations for development, testing, and validation are done within the notebook container. Computation on large hydrological units uses the built-in job submission and visualization tools in the template cyberGIS computation image, which then sends the computation to ROGER HPC environment.
- HAND input/output data visualization uses the innovative *Floret* library (section 3.2.3). Raster layers generated on ROGER HPC as TMS are displayed as *Floret* cells, together with auxiliary DEM and flowlines (Figure 5). The pyramiding of output

Table 1: NFIE HAND collaboration: current vs. CyberGIS-Jupyter enhanced

Collaboration Requirements	*Current*	*Enhanced*
Methodology development	• Writeups as shared document • Communicated via emails, teleconf	• Methodology notebook – Math formula – Code snippets – Sample results
Software development	• Source codes only; computing environment needs to be maintained and synchronized manually	• Function notebooks with both source codes and computing environments effectively synchronized between researchers
Computation	• Conducted by a dedicated person, a bottleneck	• Notebook interface to workflow computation on ROGER – Everyone can launch
Result validation	• Data: direct download or via iRODS • Validation results: shared document in Google Drive	• Integrated validation notebooks with reproducible input, statistics, and output
Visualization	• Local: download and use desktop GIS. Almost impossible for large outputs • Online: Tile Map Service (TMS); webGIS. Only available for major output data	• Integrated data, code and visualization notebooks – Traditional visualization libraries – Jupyter IFrame cells – CyberGIS Floret library

rasters is computed within the batch job on ROGER HPC to improve performance and make HAND notebooks responsive.

Figure 5 demonstrates an abridged notebook interface to HAND calculation for an example hydrological unit. The computation job is submitted via the cyberGIS interface. It is worth noting that in this example, the computational parameters are provided through the function call and the job is automatically submitted after executing the cell. In other words, GUI representation of parameters is offered as an option to meet adaptive preferences between user-friendliness and reproducibility. After the computation job completes, the results are automatically retrieved for visualization within the notebook.

4.3 Usage in HAND development and outreach

There are two typical usage scenarios for the HAND application: modeler-developer-operator and instructor-student.

A proposed change to a HAND methodology component often engages hydrology and cyberGIS researchers, HAND developers and operators in an iterative refinement process. In this process, a refinement notebook is setup to illustrate a current issue and proposed improvement. For example, in order to study the discrepancy between the flowlines generated by TauDEM and those compiled in NHDPlus, a researcher presents a comparison within the notebook. Methods are proposed by researchers as math formula and document cells. These methods are evaluated in the same notebook to validate their quality and then taken by developers to update HAND workflow code. Next, with the updated HAND software, the national scale HAND is updated in the same notebook using the job submission tool. Upon job completion, researchers review the updated HAND map using the map cell which plots a TMS layer in the notebook.

In the instructor-student scenario, the HAND notebook includes all steps of the workflow, instructions on how each step works, and sample output. This notebook can be used by instructors to create homework assignments that are distributed to students. Students launch a copy of notebooks inside their own container, learn by executing/customizing example steps and answer homework questions inside the notebook. A complete view of the HAND workflow notebook is accessible at http://cybergis-jupyter.tk/HAND.html.

A series of CyberGIS-Jupyter notebooks including the HAND notebook, along with the CyberGIS-Jupyter platform itself, were heavily used and widely acknowledged for the hands-on training and the open challenges during the 2017 UCGIS Summer School (http://www.ucgis.org/summer-school-2017). We conducted a user survey to collect feedback from 35 summer school participants on the CyberGIS-Jupyter system for GIS analytics. Seventeen of them responded, in which:

- over **76%** strongly (4+/5) agree that CyberGIS-Jupyter reduces the turnaround time of geospatial research collaboration and prefer CyberGIS-Jupyter over web GIS app in terms of research collaboration;
- over **88%** strongly (4+/5) agree that CyberGIS-Jupyter increases the reproducibility of geospatial research, and are willing to adopt Jupyter in their own research/education;
- over **94%** are very likely (4+/5) to adopt CyberGIS-Jupyter for developing geospatial methodology with collaborators

when materials, including equations, code snippets, data and experimental results need to be shared.

5 CONCLUDING DISCUSSION

CyberGIS-Jupyter is established to achieve reproducible geospatial analytics at scale in broader scientific domains by reducing the barrier to accessing advanced CI and cyberGIS capabilities via novel science gateway approaches. Exploiting recent advances in Jupyter, cloud computing, and hybrid HPC architecture, the framework provides a holistic solution for developing and sharing cyberGIS applications. Aiming to achieve desirable agility and reproducibility, the framework adopts a comprehensive architecture where a standard Jupyter notebook is enhanced to include user-friendly interfaces to foundational cyberGIS capabilities such as computation management, raster data tiling and web mapping services. Applications built on CyberGIS-Jupyter can further customize notebooks with specific geospatial analytical tools as well as big data. Researchers can choose to run analytics either within a container or by accessing external HPC resources. With the HAND case study, we demonstrated how the framework enhances domain-specific cyberGIS collaboration into an interactive application for team-based development and education practices at scale. In sum, CyberGIS-Jupyter significantly lowers the barrier of entry to cyberGIS and enables various opportunities for data- and computation-intensive geospatial research and education with desirable computational reproducibility support.

To further enhance collaborative and reproducible cyberGIS analytics, we plan to provide a dedicated service to manage community-driven cyberGIS application repositories. Geospatial researchers would be equipped with a public platform to create and publish their own CyberGIS-Jupyter applications for others to reproduce and adapt.

The CyberGIS-Jupyter interface could also serve as a light-weight geospatial frontend to manage different middleware or micro services. Furthermore, as Jupyter notebook provides better support for general web rendering, existing web-based cyberGIS science gateways may be integrated with CyberGIS-Jupyter for better computational reproducibility support.

6 ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation (NSF) under grant numbers 1047916 and 1443080. The computational work used the NSF-supported ROGER supercomputer (1429699). This work is also supported in part by the ECSS program of XSEDE, which is supported by NSF grant number 1053575.

REFERENCES

[1] V Ardizzone, R Barbera, A Calanducci, M Fargetta, E Ingrà, I Porro, G La Rocca, S Monforte, R Ricceri, R Rotondo, D Scardaci, and A Schenone. 2012. The DECIDE Science Gateway. *J Grid Computing* 10, 4 (27 Oct. 2012), 689–707.

[2] Carl Boettiger. 2015. An Introduction to Docker for Reproducible Research. *Oper. Syst. Rev.* 49, 1 (Jan. 2015), 71–79.

[3] John W Cobb, Al Geist, James A Kohl, Stephen D Miller, Peter F Peterson, Gregory G Pike, Michael A Reuter, Tom Swain, Sudharshan S Vazhkudai, and Nithya N Vijayakumar. 2007. The Neutron Science TeraGrid Gateway: a TeraGrid science gateway to support the Spallation Neutron Source. *Concurr. Comput.* 19, 6 (25 April 2007), 809–826.

[4] Robert Gentleman and Duncan Temple Lang. 2007. Statistical analyses and reproducible research. *J. Comput. Graph. Stat.* (2007).

[5] Tony Hey and Anne E Trefethen. 2005. Cyberinfrastructure for e-Science. *Science* 308, 5723 (6 May 2005), 817–821.

[6] Hao Hu, Xingchen Hong, Jeff Terstriep, Yan Y Liu, Michael P Finn, Johnathan Rush, Jeffrey Wendel, and Shaowen Wang. 2016. TopoLens: Building a CyberGIS Community Data Service for Enhancing the Usability of High-resolution National Topographic Datasets. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale (XSEDE16)*. ACM, New York, NY, USA, 39:1–39:8.

[7] Hao Hu, Tao Lin, Yan Y Liu, Shaowen Wang, and Luis F Rodríguez. 2015. CyberGIS-BioScope: a cyberinfrastructure-based spatial decision-making environment for biomass-to-biofuel supply chain optimization. *Concurr. Comput.* 27, 16 (2015), 4437–4450.

[8] C Jordan, D Walling, W Xu, S A Mock, N Gaffney, and D Stanzione. 2015. Wrangler's user environment: A software framework for management of data-intensive computing system. In *2015 IEEE International Conference on Big Data (Big Data)*. 2479–2486.

[9] T Kluyver, B Ragan-Kelley, F Pérez, and others. 2016. Jupyter Notebooks—a publishing format for reproducible computational workflows. *Proceedings of the 20th International Conference on Electronic Publishing* (2016).

[10] Jens Krüger, Richard Grunzke, Sandra Gesing, Sebastian Breuers, André Brinkmann, Luis de la Garza, Oliver Kohlbacher, Martin Kruse, Wolfgang E Nagel, Lars Packschies, and Others. 2014. The MoSGrid science gateway–a complete solution for molecular simulations. *J. Chem. Theory Comput.* 10, 6 (2014), 2232–2245.

[11] Yan Liu, Anand Padmanabhan, and Shaowen Wang. 2015. CyberGIS Gateway for enabling data-rich geospatial research and education. *Concurr. Comput.* 27, 2 (2015), 395–407.

[12] Yan Y Liu, David R Maidment, David G Tarboton, Xing Zheng, Ahmet Yildirim, Nazmus S Sazib, and Shaowen Wang. 2016. A CyberGIS Approach to Generating High-resolution Height Above Nearest Drainage (HAND) Raster for National Flood Mapping. The Third International Conference on CyberGIS and Geospatial Data Science.

[13] Eric A Marks and Michael Bell. 2008. *Service Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology.* John Wiley & Sons.

[14] Gabriel Mateescu, Wolfgang Gentzsch, and Calvin J Ribbens. 2011. Hybrid Computing—Where HPC meets grid and Cloud Computing. *Future Gener. Comput. Syst.* 27, 5 (2011), 440–453.

[15] Dirk Merkel. 2014. Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux J.* 2014, 239 (March 2014).

[16] Anand Padmanabhan, Shaowen Wang, Guofeng Cao, Myunghwa Hwang, Zhenhua Zhang, Yizhao Gao, Kiumars Soltani, and Yan Liu. 2014. FluMapper: A cyberGIS application for interactive analysis of massive location-based social media. *Concurr. Comput.* 26, 13 (10 Sept. 2014), 2253–2265.

[17] Fernando Perez and Brian E Granger. 2007. IPython: A System for Interactive Scientific Computing. *Computing in Science and Engg.* 9, 3 (May 2007), 21–29.

[18] Judy Qiu, Jaliya Ekanayake, Thilina Gunarathne, Jong Youl Choi, Seung-Hee Bae, Hui Li, Bingjing Zhang, Tak-Lon Wu, Yang Ruan, Saliya Ekanayake, Adam Hughes, and Geoffrey Fox. 2010. Hybrid cloud and cluster computing paradigms for life science applications. *BMC Bioinformatics* 11, 12 (2010), S3.

[19] Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, and Eivind Hovig. 2013. Ten simple rules for reproducible computational research. *PLoS Comput. Biol.* 9, 10 (Oct. 2013), e1003285.

[20] Matthew J Turk. 2013. How to Scale a Code in the Human Dimension. *arXiv preprint arXiv:1301.7064* (29 Jan. 2013). arXiv:1301.7064

[21] M Turner, D Budgen, and P Brereton. 2003. Turning software into a service. *Computer* 36, 10 (Oct. 2003), 38–44.

[22] Shaowen Wang. 2010. A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis. *Ann. Assoc. Am. Geogr.* 100, 3 (2010), 535–557.

[23] Shaowen Wang. 2016. CyberGIS and spatial data science. *GeoJournal* 81, 6 (9 Aug. 2016), 965–968.

[24] Shaowen Wang, M P Armstrong, Jun Ni, and Yan Liu. 2005. GISolve: a grid-based problem solving environment for computationally intensive geographic information analysis. In *CLADE 2005. Proceedings Challenges of Large Applications in Distributed Environments, 2005.* IEEE, 3–12.

[25] Nancy Wilkins-Diehr. 2007. Special issue: science gateways—common community interfaces to grid resources. *Concurr. Comput.* 19, 6 (2007), 743–749.

[26] N Wilkins-Diehr, D Gannon, G Klimeck, S Oster, and S Pamidighantam. 2008. TeraGrid Science Gateways and Their Impact on Science. *Computer* 41, 11 (Nov. 2008), 32–41.

[27] Chaowei Yang, Manzhu Yu, Fei Hu, Yongyao Jiang, and Yun Li. 2017. Utilizing Cloud Computing to address big geospatial data challenges. *Comput. Environ. Urban Syst.* 61, Part B (2017), 120–128.

[28] Junjun Yin, Aiman Soliman, Dandong Yin, and Shaowen Wang. 2017. Depicting urban boundaries from a mobility network of spatial interactions: A case study of Great Britain with geo-located Twitter data. *International Journal of Geographical Information Science* (Jan. 2017).