A CYBERGIS INTEGRATION AND COMPUTATION FRAMEWORK FOR HIGH-RESOLUTION CONTINENTAL-SCALE FLOOD INUNDATION MAPPING

Yan Y. Liu, David R. Maidment, David G. Tarboton, Xing Zheng, and Shaowen Wang

ABSTRACT: We describe a Digital Elevation Model (DEM)-based hydrologic analysis methodology for continental-scale flood inundation mapping, implemented as a cyberGIS scientific workflow in which a 1/3rd arc-second Height Above Nearest Drainage (HAND) raster data for the conterminous U.S. (CONUS) was computed and employed for subsequent inundation mapping. A cyberGIS framework was developed to enable spatiotemporal integration and scalable computing of the entire inundation mapping process on a hybrid supercomputing architecture. The first 1/3rd arc-second CONUS HAND raster dataset was computed in 1.5 days on the CyberGIS ROGER supercomputer. The inundation mapping process developed in our exploratory study couples HAND with National Water Model (NWM) forecast data to enable near real-time inundation forecasts for CONUS. The computational performance of HAND and the inundation mapping process was profiled to gain insights into the computational characteristics in high-performance parallel computing scenarios. The availability of HAND has broad and significant research implications that may lead to further development and improvement of continental-scale inundation mapping methodologies.

(KEY TERMS: computational methods, cyberGIS, data management, geospatial analysis, height above nearest drainage (HAND), inundation mapping, streamflow)

Senior Research Programmer (Liu), CyberGIS Center for Advanced Digital and Spatial Studies, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801; Professor (Maidment), Center for Water and Environment, University of Texas, Austin, Texas 78712; Professor (Tarboton), Department of Civil and Environmental Engineering, Utah State University, Logan, UT 84322-4110; Graduate Student (Zheng), Department of Civil, Architectural and Environmental Engineering, University of Texas, Austin, Texas 78712; and Professor (Wang), CyberGIS Center for Advanced Digital and Spatial Studies, University of Illinois at Urbana-Champaign, Champaign, Illinois 61820 (E-Mail/Liu: yanliu@illinois.edu)

Submitted to the Journal of the American Water Resources Association (JAWRA), April 16, 2017.

INTRODUCTION

In August 2016, the National Weather Service brought into operation a National Water Model (NWM) (U.S. National Oceanic and Atmospheric Administration (NOAA), National Water Model. Accessed March 10, 2017, http://water.noaa.gov/about/nwm), which forecasts the flow on approximately 2.69 million stream reaches covering about 5.2 million kilometers of rivers and streams of the continental United States (CONUS). The academic community is collaborating with the National Weather Service to enhance NWM in a project called the National Flood Interoperability Experiment (NFIE) (Maidment, 2016a). A key component of that project is to extend the forecasting of flood discharge into forecasting of water depth and inundation extent at the continental scale. The companion papers by Maidment *et al.* (2017) and Zheng *et al.* (2017) explain the context and the methodology of this task for watersheds in Texas. This paper demonstrates how the methodology for forecasting flood depth and inundation can be extended to the continental scale.

Central to this methodology is a technique called Height Above Nearest Drainage (HAND) (Nobre *et al.*, 2011; Nobre *et al.*, 2016; Rennó *et al.*, 2008), which uses a Digital Elevation Model (DEM) to define the height of each cell in the land surface above the cell in the nearest stream to which the drainage from that land surface cell flows. The HAND method is applied to the stream reaches used in NWM, which themselves are derived from the medium resolution NHDPlus dataset (US Geological Survey (USGS) and Environmental Protection Agency (EPA), NHDPlus. Accessed March 10, 2017, http://www.horizon-systems.com/nhdplus). By combining the NHDPlus with the USGS 3D Elevation Program (3DEP) dataset (USGS, 3DEP. Accessed March 10, 2017, https://nationalmap.gov/3DEP) at 1/3rd arc-second (about 10-meter) cell resolution, this paper shows how the HAND raster can be determined for the continental United States, and also how hydraulic geometry relationships and synthetic rating curves can be determined for each stream reach so that the forecast discharge can be converted to forecast depth for each stream reach, and then to flood inundation extent using the HAND approach. The combination of all these techniques is here referred to as Continental Flood Inundation Mapping (CFIM).

Given the significant computational challenges for conducting the inundation mapping process at CONUS scale for these massive geospatial datasets (i.e., 3DEP DEM and NHDPlus), we develop a computational model based on cyberGIS (also known as geographic information science and systems based on advanced cyberinfrastructure) (Wang, 2010) to provide a scalable integration and computation framework that is able to create HAND maps using cyberinfrastructure resources.

Our cyberGIS framework addresses the data, software, and computation challenges through collaboration among NFIE, the National Science Foundation (NSF) CyberGIS software project (Wang *et al.*, 2013), NSF HydroShare (Tarboton *et al.*, 2014; Horsburgh *et al.*, 2016), USGS, the NSF CyberGIS Facility (that houses the Resourcing Open Geospatial Education and

Research (ROGER) supercomputer) (Wang, 2017), and the Extreme Science and Engineering Discovery Environment XSEDE (Towns *et al.*, 2014). Our open source software solution constructs a cyberGIS workflow that couples the scalable and high-performance TauDEM software (Tesfa *et al.*, 2011) for DEM-based hydrologic analysis and a collection of open source geospatial software for pre- and post-processing of geospatial data. ROGER, which has a hybrid supercomputing architecture, provides an integrated high-performance data handling, analysis, modeling, and visualization platform for CFIM by coupling high-throughput computing (HTC), high-performance computing (HPC) and cloud computing. Although the first experiment for producing the 1/3rd arc-second HAND raster for CONUS required 8 days of compute time on ROGER, after refining the workflow efficiency and model calibration, the second run finished in 1.5 days. The inundation mapping process is then computed using HAND and NWM forecast data.

The availability of HAND has broad and significant research implications because it opens the door for geospatial scientists and hydrologists to improve inundation mapping methodologies at an unprecedented continental scale. The development of the CFIM computational model demonstrates the advantages of cyberGIS-enabled hydrologic analysis and modeling and the feasibility of employing continental hydrologic computations on a national cyberinfrastructure. The following sections describe computational challenges in the CFIM methodology. We demonstrate a cyberGIS integration model and a two-level parallel computing model as a holistic computational model for tackling these challenges. The effectiveness and performance of the computational model are illustrated through the computational experience and results obtained from the HAND and inundation mapping generation workflow. We conclude with discussion of the advantages and limitations of our approach, and ideas for future work.

DATA AND COMPUTATIONAL CHALLENGES

The development of CFIM methodology is a multidisciplinary collaboration of science communities in hydrology, hydraulics, geographic information science (GIScience), and meteorology. Computation cuts across all these disciplinaries and plays a central role that not only provides significant computing power on national cyberinfrastructure resources for the computation of NFIE experiments, but also develops an integrated solution that addresses the data, software, computation, visualization, and community collaboration challenges. The main research question raised in our work is: *Is it feasible to compute inundation maps for CONUS at 1/3rd arc-second or higher resolution and automate the computation on 3DEP DEM and NHDPlus?* From a computational perspective, the following issues need to be resolved.

- Terabytes and gigabytes of high-resolution national-scale terrain, water, and weather data that are distributed by multiple data sources and vary greatly in spatiotemporal scales and resolutions need to be integrated, processed, and analyzed.
- In developing the inundation mapping workflow, computational bottlenecks at the processing, modeling, and analysis steps of the proposed inundation mapping

methodology need to be identified and resolved for scalable CONUS-level computation on advanced cyberinfrastructure.

- The entire inundation mapping process needs to be automated such that the resulting software can be used to produce near real-time inundation maps from continuous NWM forecast data. The automated software needs to be componentized in order to incorporate the refinement and enhancement of various methodological logics for inundation mapping.
- High-performance and scalable computation is important to produce the output within a reasonable turnaround time of, say, a few hours, to match the working pace of the iterative research collaborations and the pace of NWS forecast data publishing. Taking weeks would seriously hinder the team progress. For achieving near real-time flood forecast, taking more than one day to compute the inundation map would be impractical. The current NWM, which runs in production on NOAA's Luna and Surge supercomputers (Top500 supercomputer ranking. Accessed March 10, 2017, https://www.top500.org/list/2016/11/?page=1), has a turnaround time of about 2 hours. The inundation mapping computation should not introduce additional significant delays.

Achieving these goals is highly dependent on the ability of the underlying computational model to provide a platform where the methods, software, data, and results could be deployed and shared in a responsive way that fosters iterative collaboration for methodological development and validation. Given the responsiveness requirements for research collaboration and national inundation mapping computation, we pursue a cyberinfrastructure-based computational model to address these challenges with two key foci. First, we configured an integrated computational platform on a hybrid supercomputing architecture that allows for the automation and integration of the inundation mapping workflow as an open software solution and provides a prototype solution for collaboration, data sharing, visualization, and high-performance computing. Second, we evaluated the scalability of the computational solution to both the data size and the number of computer processors so that the turnaround time for computation can be reduced by simply adding more computing power. This model achieves a prototype environment where compute, data storage, and network resources are integrated on demand on a centralized platform for building the required online geospatial and hydrologic services. This online problem-solving environment, in turn, serves as a community platform for broader engagement and outreach of continental inundation mapping research.

The continental scale inundation mapping methodology described in the companion paper (Zheng *et al.*, 2017) takes as input a DEM (here 1/3 arc second USGS 3DEP elevation DEM) and hydrography, comprised of geospatial vector data of flow lines, catchments and water bodies (here the US NHDPlus dataset). Our computational approach uses the generalized terrain analysis concepts of TauDEM (Tarboton, 1997; Tarboton *et al.*, 2008; 2009; Tesfa *et al.*, 2011). DEM derived streams are initiated at the sources of NHDPlus streams to produce a stream raster consistent with the DEM. A general method for calculating distance to stream in the vertical

direction was used to produce HAND from this stream raster. Three output datasets for inundation mapping are produced:

1. HAND raster of the same resolution as the input DEM. The HAND value of each raster cell represents the height of each raster cell above the nearest stream along the flow path from that cell to the stream. The HAND raster is a reference dataset for inundation mapping, that is produced once and only needs updating when input DEM or NHDPlus data source is updated. The size of this raster is the same as the input DEM.
2. Hydraulic property table with a record for each catchment defined in NHDPlus. This table is derived from the HAND raster and a defined input list of stage values to compute. Each row in the table represents the hydraulic properties for a river reach in NHDPlus for each designated stage height. Attributes calculated from HAND for each reach and stage include surface area, bed area, volume, top width, wetted perimeter, cross sectional area, hydraulic radius, and uniform flow discharge. This table serves as a lookup table to interpolate the water depth given a specific river discharge forecast in NOAA NWM and an assumption of uniform flow. This table is updated when HAND or NHDPlus is updated or an improved rating curve method or different roughness (Manning's ***n***) is to be applied. The number of rows in this table is (*number_of_river_reaches* x *number_of_stages*).
3. Inundation maps based on discharge forecast data from NOAA NWM for each NHDPlus reach. Each inundation map is a raster of the same size as HAND. The value for each raster cell is determined by comparing the HAND value with the water depth interpolated from the hydraulic property table for the catchment of the associated NHDPlus reach. A set of inundation maps are generated for each NWM forecast initialization timestamp, which includes a series of forecast timestamps. Each forecast timestamp needs an inundation map layer.

The foremost challenge of national scale inundation mapping is rooted in the data, including terrain data and open water data in space and time (Maidment, 2016b). Table 1 lists the properties of the national-scale geospatial input datasets used for producing the aforementioned outputs. The geospatial data involved in this work represent typical scientific big data in volume, variety, and velocity. The volume of geospatial data is determined by its spatial extent and resolution. For example, the USGS 3DEP elevation dataset covers the entire U.S. Its size increases by a factor of 100 as the resolution moves from 1/3rd arc-second (10-meter) to 1/27th arc-second (1-meter). The entire 3DEP dataset, when completed, is estimated to be 5PB. Geospatial data come in different representations (e.g. raster and vector), formats, projections, visualization forms, etc. The varied data sources require software and computing capabilities that are able to transform, convert, and harmonize different datasets within a coherent spatial extent for inundation analysis. At the same time, the speed at which the input data changes is influenced by two sources. First, USGS 3DEP DEM and NHDPlus are constantly updated, which means the HAND raster and the hydraulic property table need to be updated. Second, NOAA NWM forecast is a spatiotemporal dataset that is updated hourly (for short range) or

daily (for medium range). Simply integrating these input datasets for accessing, processing, and analyzing is beyond the capabilities of traditional desktop-based geospatial or hydrologic computing platforms. Even basic geospatial data processing tasks, such as subsetting/clipping and re-projection, are formidable for these datasets on desktop computers. Unfortunately, most geographic information systems (GIS) are built as desktop software and, hence, ill-suited to meet our needs. Often in research computing, a methodology is developed on desktop computers, tested using sample datasets, and it is assumed to run on larger machines and datasets without a problem. This approach ignores the scaling issue as problem size increases. For the development of the CFIM methodology, this approach is problematic for a number of reasons. First, sample datasets cannot reveal the insights hidden in the national datasets, which in turn, affects the choice and implementation of the methodology itself. Second, the computational intensity, which depends on both the computing complexity of the underlying method and the actual computing cost when applying the method to the input data, may increase non-linearly (e.g., polynomial increase) with data size. Finally, a single computer does not have enough memory to hold the input data listed in Table 1 as well as the intermediate data that is generated at runtime.

TABLE 1. Properties of the Input Data Sources.

Data Source	*Resolution & Coverage*	*Size*	*Format*	*Update Frequency*
USGS 3DEP Elevation Dataset	1/3rd arc-second (10m); Entire U.S.	635939 x 282122 (180b cells); 718GB uncompressed	ArcGrid, GeoTIFF	Partial update every 3 months
	1/9th arc-second (3m); Partial U.S.	~10 times larger than 1/3rd arc-second	ArcGrid, GeoTIFF	N/A
	1/27th arc-second (1m); Partial U.S.	~100 times larger than 1/9th arc-second	ArcGrid, GeoTIFF	N/A
NHDPlus from EPA and USGS	1:100,000; Entire U.S.	~2.69 million reaches; 12 layers;18GB	Esri FileGDB	Version 2.1
	1:24,000 Partial U.S.	~30 million reaches; ~77 layers in pre-release versions	Esri FileGDB	Not released yet
NOAA NWM channel_rt forecast	1:100,000; Entire U.S.	2,699,225 reaches; short range: 15 hourly forecasts medium range: 80 forecasts (3-hr; 10 days)	NetCDF	short range: hourly; medium range: daily

COMPUTATIONAL MODEL

In order to support the computational needs for methodology co-development and

production-level computation for real-time inundation mapping, a high-performance and scalable cyberGIS integration and computation framework was built on the ROGER supercomputer to provide a holistic computational model for collaboration and computation. A two-level parallelization approach was developed as a scalable computing strategy to exploit parallel computing resources for the implementation of the inundation mapping methodology and the execution of the corresponding computational workflow on ROGER.

CyberGIS Integration Model

We exploited massive computing power enabled by advanced cyberinfrastructure (e.g., XSEDE) to address the integration and computational challenges presented in the NFIE CFIM project. Allocable resources on cyberinfrastructure include not only hardware (compute, memory, storage, and network) resources, but also software environment, parallel computing libraries, and higher-level services such as performance profiling and acceleration, community application development, collaborative science gateways (Lawrence *et al.*, 2015) through user support programs such as XSEDE ECSS (Wilkins-Diehr *et al.*, 2015). The cyberinfrastructure approach integrates a powerful computational platform via supercomputers that enables the development of HPC solutions for domain applications.

In a traditional cyberinfrastructure approach, computational resources are allocated to user accounts on a set of supercomputers that are distributed at major U.S. supercomputing centers. Users then interact with supercomputers through direct system-level access: secure connection for login and data transfer; configuration of software and library environment in a Linux/Unix operating system for code development; and computing using job schedulers that send user computing jobs into a shared job queue and wait for available computing nodes as they are freed from other jobs. In this computing centric approach, while system-level resources are integrated, the user is faced with daunting HPC complexity. This approach, moreover, involves frequent data transfer, software update and re-deployment, and re-computation of results between a local computer and the remote supercomputer, which is especially costly when graphic user interfaces (GUI) are needed in geographic information systems (GIS) to verify results. The issues for our project are obvious since computation for just a small watershed may result in several GB of output data, hindering methodology development even when the supercomputer provides adequate computation.

A promising advance in supercomputing, hybrid supercomputing architecture, has been recognized as a potential solution for providing a highly integrated computational platform from application to system level for the support of large research computing projects such as the NFIE CFIM. A hybrid supercomputing architecture typically couples HPC, cloud, and data-intensive computing resources together on a single physical supercomputer. Data are stored on a central storage system that establishes access protocols to the cloud and HPC through parallel file systems. Since the cloud component focuses on the usage model, computing resources can be configured according to application needs while remaining highly integrated with other

components. Typically, the cloud allocation for a research project includes a set of services that HPC, alone, is unable to support, such as data servers, databases, web servers, mapping and visualization servers.

In CFIM, most of the data processing and analysis employs and outputs geospatial data, which presents a typical case of the aforementioned integration and computation challenges that arise from cyberGIS problems. CyberGIS is cyberinfrastructure-enabled high-performance, integrated, and collaborative GIS (Wang, 2010). As a major approach in our framework, we employ ROGER, a dedicated cyberGIS supercomputer, as a hybrid supercomputer to provide a centralized sandbox solution for the integration and computation of the entire process of methodology development and inundation mapping data computing. The hybrid supercomputing architecture on ROGER has three components: HPC (32 computing nodes with 12 equipped with GPU), cloud (13 nodes), and data-intensive computing (a Hadoop cluster of 11 nodes). The three components share 5PB of usable storage via the GPFS parallel file system. This architecture makes it possible to eliminate the cumbersome intermediate steps in data transfer and software management in the HPC-only supercomputer usage model and presents an online research computing environment with direct access to and processing of input and output data by end user.

Figure 1 shows the cyberGIS integration model on ROGER, which integrates data, software, computation, and an online problem-solving environment (PSE) on ROGER's cloud, HPC, and storage systems (currently, the data-intensive computing cluster is not used) in order to automate or streamline the methodological components of CFIM.

The three major national data sources for CFIM are deployed and updated on the 5PB GPFS storage system on ROGER. An HTTP-based download tool was developed to deploy the 3DEP DEM and the NHD data from USGS (USGS, The National Map. Accessed March 10, 2017, ftp://rockyftp.cr.usgs.gov/vdelivery/Datasets/Staged/). The full coverage 1/3rd arc-second 3DEP DEM is organized as 936 1º x 1º tiles in its original form and projection (EPSG:4269), but is converted to GeoTIFF format. The downloading and conversion used multiple computing nodes on ROGER to process these tiles in parallel using GNU Parallel (GNU Parallel. Accessed March 10, 2017, https://www.gnu.org/software/parallel/), an embarrassingly parallel computing tool. The entire DEM is read as the Open Geospatial Consortium (OGC) virtual raster (VRT) format. This dataset can be updated by checking the timestamp on the USGS FTP server when new tiles are updated. ROGER also hosts a 3DEP elevation data service - TopoLens (Hu *et al.*, 2016) for the customization of DEM data delivery. 3-meter and 1-meter DEMs do not have the full coverage for CONUS. They are downloaded from the USGS FTP server or generated from 3DEP LiDAR datasets that are transferred to ROGER using the high-performance GridFTP protocol. NHDPlus medium-resolution (1:100,000) and high-resolution (1:24,000) data were downloaded and deployed in its original Esri FileGDB format. The Linux 64-bit FileGDB library is installed to access them within GDAL (Geospatial Data Abstraction Library (GDAL). Accessed March 10, 2017, http://gdal.org). The NOAA NWM forecast data is in NetCDF4

format and is downloaded using a streaming script that is able to pull the *channel_rt* short or medium range forecast for each forecast initialization time stamp when it becomes available on NWM (NOAA, NWM forecast data site. Accessed March 10, 2017, http://nomads.ncep.noaa.gov/pub/data/nccf/com/nwm/prod/) or HydroShare (HydroShare THREDDS server. Accessed March 10, 2017, http://thredds.hydroshare.org/thredds/catalog/nwm/catalog.html). ROGER HPC alone (8TB memory in total), is able to provide sufficient compute, memory, storage, and network resources for the most demanding geospatial and hydrologic functions to be applied on the entire 3DEP elevation dataset. For example, the maximum memory requirement of TauDEM functions is four times of the input DEM, i.e., 3.2TB for processing the entire 1/3rd arc-second 3DEP DEM.

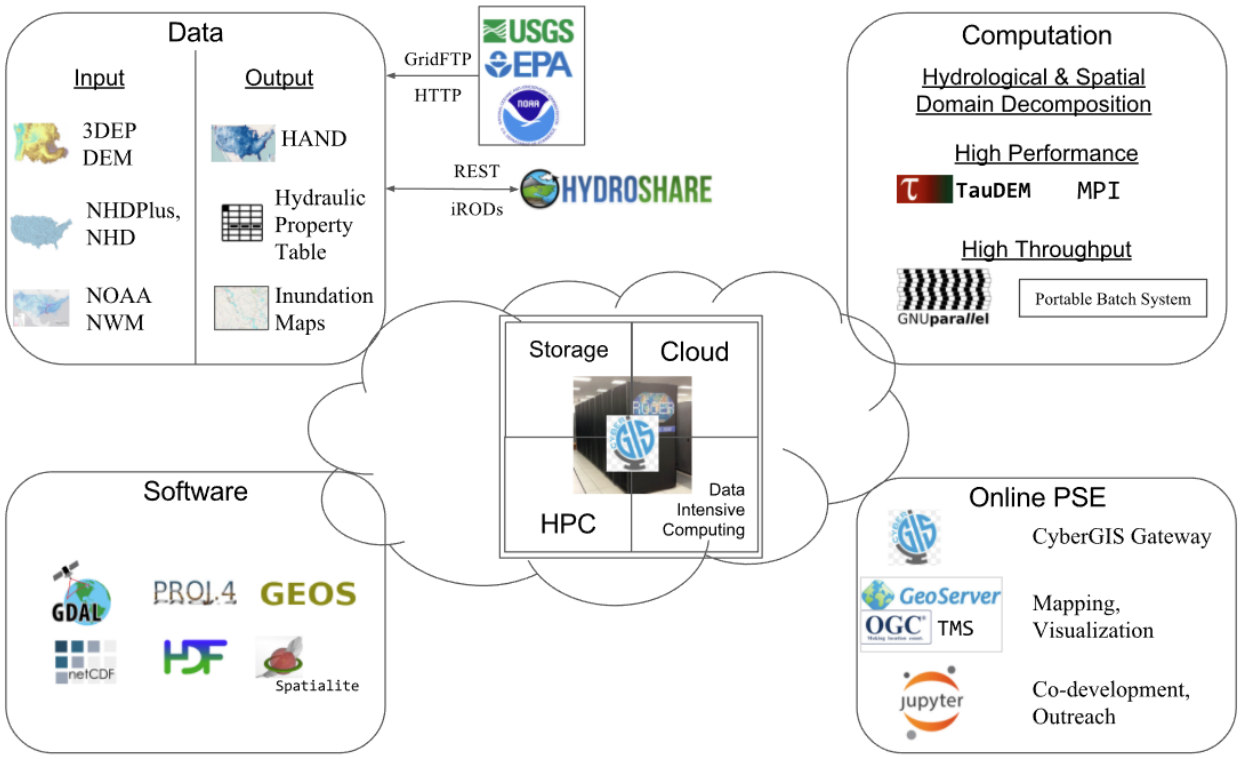

FIGURE 1. CyberGIS Integration Model

All of the hydrologic and geospatial software needed to handle these input datasets are open source or in free libraries. They are customized and built into the ROGER geocomputation software environment. Different versions and customizations are managed by the Environment Modules (Environment Modules. Accessed March 10, 2017, http://modules.sourceforge.net/) system to allow for a maximum degree of compilation and runtime flexibility. For example, the GDAL library is built with Proj4 (for reprojection), GEOS (for spatial and geometrical operations), NetCDF4/HDF5 (for weather data handling), SpatialLite (for vector storage over shapefile's 2GB limitation), and Esri FileGDB extension.

Because external geospatial software libraries are generally incapable of exploiting computing power across multiple computing nodes for processing a single dataset, in-house efficient and parallel computing software must be developed when employing external libraries becomes too costly. A computational intensity study was conducted to examine the performance of these libraries with large datasets. In general, the processing of the input and intermediate data in CFIM can be categorized into raster, vector, and raster-vector processing. Raster processing functions include clipping and hydrologic information extraction. Vector functions include inlet/outlet identification, flowline analysis, multi-layer operations such as river reach-catchment join and georeferencing with non-spatial NWM data. Raster-vector processing includes vectorization, rasterization, and dynamic inundation mapping by coupling HAND raster, catchment raster, and NWM vector data. Most computationally intensive processing in CFIM is on raster data processing. Map algebra (Tomlin, 2012) is an abstract way to look at operations on raster data in one of the four categories: local, focal, zonal, and global. Basic raster operations such as clipping, local or focal operations can be handled by GDAL efficiently even on large datasets. Most of the DEM-based hydrologic information analysis functions, such as pit filling and flow direction detection, are global functions that might require, in a worst case scenario, visiting all of the input grid cells for calculating the value of an output cell. For these functions, the high-performance TauDEM (Tesfa *et al.*, 2011) is deployed on ROGER HPC to exploit parallel computing power for large-scale analysis (see section *Scalable Computing* for details).

Our computational profiling exercises showed that language choice and efficient implementation of commonly used general functions are critical to performance. For example, a raster function that needs to visit each grid cell is much more efficient when implemented using the GDAL C/C++ API, instead of the Python API because the Python Numpy package requires an intermediate data model conversion. Take an example of efficient implementation of general functions, one can implement a SQL query to compile a flowline shape file for a hydrologic unit from the NHDPlus high-resolution dataset for HUC4 unit 1209 with inlet-outlet relationship encoded. The SQL query can be executed by GDAL's *ogr2ogr* tool to join two layers:

```
ogr2ogr -mapFieldType Real=String -sql "SELECT NHDPlusBurnLineEvent.NHDPlusID AS
NHDPlusID, NHDPlusBurnLineEvent.ReachCode AS ReachCode, NHDPlusFlowlineVAA.FromNode
AS FromNode, NHDPlusFlowlineVAA.ToNode AS ToNode FROM NHDPlusBurnLineEvent LEFT JOIN
NHDPlusFlowlineVAA ON NHDPlusBurnLineEvent.NHDPlusID=NHDPlusFlowlineVAA.NHDPlusID
WHERE NHDPlusBurnLineEvent.ReachCode LIKE '120902%'" 120902-flows.shp
HRNHDPlus1209.gdb
```

However, since no optimization is built on this specific query in the file geodatabase, executing the *left join* between two layers with 162,233 and 162,840 records, respectively, takes 1 hour and 44 minutes on Intel Xeon E5-2660 v3 2.6GHz processor using the native Esri FileGDB. The same function, implemented using the GDAL Python API, by building a static hash table on the key `NHDPlusID` for joining attributes from the two layers, takes only 5.8 seconds. A number of specialized tools have been developed to eliminate such performance

bottlenecks in the CFIM computational workflow.

Intermediate results and output for each step in the CFIM computational workflow provide data content for investigating and validating our methodological development. We also seek community feedback by publishing our output datasets as open data. The data sharing and visualization requirements demand an efficient collaboration solution that makes best efforts to avoid the transfer of hundreds of gigabytes of data. Accordingly, an online problem-solving environment (PSE) is built on the ROGER OpenStack cloud. A set of services is built as on-demand virtual machine images (VM) and container (e.g., Docker (Docker containerization. Accessed March 10, 2017, https://www.docker.com)) instances. These cloud services have read-only access to the ROGER storage system and so directly consume data generated by ROGER HPC.

As shown in Figure 1, data sharing is enabled through web-based downloading and the underlying iRODS data federation between ROGER and HydroShare's iRODS data storage at RENCI (RENCI, HydroShare project. Accessed March 10, 2017, http://renci.org/research/hydroshare/). The iRODS data management system is used to provide cross-domain data integration between ROGER and HydroShare in order to share HAND results through HydroShare data repositories. Online visualization of national-scale raster results at 1/3rd arc-second resolution or finer is beyond the capabilities of a single mapping server. Our strategy to build the visualization tiles for multiple zoom levels, i.e., raster pyramids, is two-fold. First, the computation needed to generate the visualization data occurs on the same computing nodes that produce CFIM analysis results. Second, the need for a powerful mapping server to host massive visualization data is eliminated by publishing a raster layer as an OGC standard Tile Map Service (TMS), which only requires a web server to hold the data and supports the tile rendering using the straightforward [*tile.x, tile.y, zoom_level*] mapping to tile image file path. Visualization data is then rendered within a browser using a CyberGIS WebGIS module. Common geospatial datasets, such as the NHD water boundary dataset (WBD), are hosted on GeoServer, an open source mapping server based on Java and Tomcat. A Jupyter (Jupyter interactive computing. Accessed March 10, 2017, http://jupyter.org) interactive analysis environment has recently been prototyped for co-development purposes in order to exchange methodology ideas, code snippets, and results. In this environment, the same data and software environment is built as a Docker template and instantiated in the cloud for each user. For example, the HAND workflow can be viewed in a Jupyter notebook (e.g., on https://goo.gl/0D35bj) with instructions, scripts, and visualization results.

In the literature, Snow *et al.* (2016) developed a computational forecast framework and a web-based visualization application to tackle similar NFIE questions. High-density ensemble national-scale stream forecasts were produced by downscaling runoff forecasts generated by ECMWF and routing the runoff using the RAPID model (David *et al.*, 2011). Streamflow forecasts are displayed using the Tethys Platform (Jones *et al.*, 2014). In addition to obvious difference in inundation mapping methodology and focus, our computational model differs in the

following aspects. First, our model is designed to achieve 1/3rd arc-second or finer inundation mapping for CONUS and covers all 2.67 million reaches in NHDPlus from the beginning. Second, high-throughput computing is the only parallel computing model used in Snow *et al.* (2016). HTCondor (Bockelman *et al.*, 2015) is used to employ multiple processors to compute the downscaling of ECMWF runoff forecast on multiple watersheds. We provide a more comprehensive parallelization to achieve both high throughput (via job scheduler and GNU Parallel) and high performance (via TauDEM). Last, the cyberGIS approach based on ROGER as a sandbox for integrating massive data and computing resources and building online problem-solving environment provides an efficient and scalable environment for tackling computational challenges in CFIM. A similar approach could also be used to enhance the ECMWF-RAPID computational forecast framework.

Scalable Computing

The cyberGIS integration model makes it possible to streamline and automate the hydrologic and geospatial functions needed for the CFIM process into a computational workflow. The performance of this workflow depends on identifying and accelerating the functions that are computationally expensive. The computational performance of an individual CFIM function is mainly dependent on the resolution and scale of the input datasets. A scalable computing solution to CFIM must be able to utilize massive computing power on cyberinfrastructure via parallel computing in order to finish a computation in a reasonable amount of time. Our parallel computing framework leverages two levels of parallelism to accelerate CFIM functions and the entire workflow. These two levels of parallelisms are exploited by two types of parallel computing models: high-throughput and high-performance computing, respectively.

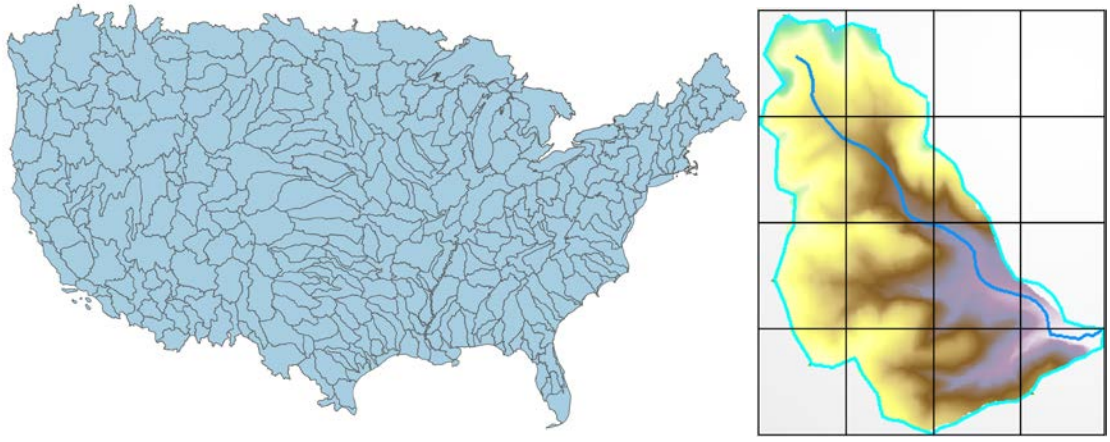

FIGURE 2. Two-level parallelization based on hydrologic and spatial domain decomposition. On the left, the top level parallelization decomposes the U.S. continent based on the well-developed HUC system, resulting in 336 HUC6 units. On the right, each HUC unit is computed in parallel

using regular spatial domain decomposition in a single computing job.

The first level of parallelism, leveraging the Hydrologic Unit Code (HUC) system (USGS, HUC system. Accessed March 10, 2017, http://water.usgs.gov/GIS/huc.html), implements a divide-and-conquer strategy to provide a hydrologic and spatial data decomposition (Wang *et al.*, 2009) mechanism for the parallel computing of all the HUC units for CONUS, as shown on the left in Figure 2. In hydrology, watersheds are organized as a hierarchical hydrologic unit network, known as HUC. Four levels of HUCs (2, 4, 6, and 8) are fully supported by NHD and NHDPlus. At each level, the boundary of each HUC unit has been carefully drawn so that each unit can be studied independently with minimal interference from neighboring units. The HUC system provides a natural spatial domain decomposition framework to divide CONUS terrain. It provides an explicit spatial granularity from which we can match and allocate the runtime computing power to the computational requirements of each HUC unit, making the batch processing of all HUC units in CONUS possible. We chose HUC6 as the basic decomposition HUC level. There are 336 HUC6 units on CONUS, but the five Great Lakes units are not considered. For the 331 units in consideration, each of them is computed independently on the clipped DEM and NHDPlus input for that unit. In parallel computing, this type of parallelization belongs to the high-throughput computing model in which embarrassingly parallel computing jobs are sent to a batch job scheduler to saturate the computing bandwidth of a computing cluster.

The second level of parallelism is for the parallel computing of each individual HUC6 unit, shown on the right in Figure 2. Regular spatial domain decomposition (e.g., row-, column-, or block-wise decomposition) is applied to distribute input and output data domains to a set of processors. These domains form a network topology that represents their adjacency relationship. Ghost zones are established to store the boundary data belonging to neighboring domains for runtime data exchange via the broadly used Message Passing Interface (MPI) among participating processors. TauDEM, the well-known high-performance hydrologic information analysis software built on MPI, is employed to process individual HUC6 units.

Our two-level parallelization strategy provides a comprehensive scalable computing framework that is adaptive to data size, resolution, and the number of allocated processors. HUC6 is chosen because TauDEM exhibits the best computing efficiency at this level when experiments were conducted. As TauDEM's performance is accelerated, we can apply it to higher level HUC, which results in fewer computing jobs, but each job requires higher performance from more processors. Given sufficient computing power and TauDEM numerical performance, it is possible to compute the entire CONUS as a single computing job at 1/3rd arc-second resolution. On the other hand, when 3-meter, 1-meter and sub-1-meter DEM become available for CONUS, this framework can be applied with an appropriate HUC level decomposition that is determined by TauDEM's capability to handle a single DEM.

Pseudo algorithms, shown in Figure 3, illustrate how the two-level parallelization was

applied to obtain the three major CFIM output datasets. All of the *for* loops on HUC units and the *for* loop on NWM forecast dataset in Algorithm 3 employ the high-throughput computing model to execute each iteration in parallel on ROGER HPC. Inside of the outer *for* loop, the costly hydrologic functions are computed using TauDEM on multiple processors. At any time, the number of processors involved in computation is the product of the number of HUC units being computed and the number of processors allocated to run TauDEM. In Algorithm 3, the number of processors adds an additional multiplicative factor to the number of forecasts in the NWM forecast dataset.

Algorithm 1: HAND generation

```
foreach HUC unit h:
    clip DEM from USGS 3DEP and flowline from NHDPlus;
    compute HAND for h;
merge HAND at the HUC level into CONUS.
```

Algorithm 2: Hydraulic property table generation

```
foreach HUC unit h:
    prepare HAND input, reach polygons for h;
    read stage table;
    foreach COMID and stage height:
        compute hydraulic properties including channel geometry and rating
curve;
merge hydro property table at the HUC level into CONUS.
```

Algorithm 3: Inundation map generation

```
foreach NOAA NWM forecast:
    foreach COMID:
        Interpolate inundation depth from hydraulic property table and HAND;
    generate forecast table;
    foreach HUC unit h:
        generate inundation map raster from the forecast table;
        generate TMS visualization layer;
    merge inundation map raster and visualization layer for CONUS.
```

FIGURE 3. Pseudo algorithms for generating the major three output datasets in CFIM.

The HUC level parallelization and the temporal parallelization in Figure 3 generate batch computing jobs, which are then submitted to ROGER HPC directly to the PBS job scheduler or through explicit resource reservation. When GNU Parallel is used, intra-node parallelism can be exploited by running 20 jobs on the 20 CPU cores on each computing node. In order to efficiently manage the computation, we can conduct computational intensity analysis to understand the distribution of computing cost among all the HUC level jobs. Figure 4 illustrates this analysis on the number of cells and data cells on each HUC's DEM raster, the main

determinants of run time for a function that takes DEM as input. The results from this analysis can be used to guide the allocation and scheduling of computing nodes on ROGER HPC in order to reduce turnaround time (the time between when the first job starts and the last job ends).

FIGURE 4. Data density map for 331 HUC6 units on CONUS, excluding the five Great Lakes units. The coloring denotes the number of grid cells in the DEM of each unit The map is visualized using equal-count (quantile) symbology. The unit is in millions.

The computation of individual HUC6 units is more challenging and requires high-performance hydrologic analysis software such as TauDEM. We accelerate TauDEM to scale to thousands of processors and DEMs of tens of gigabytes through the XSEDE ECSS program. Through the work in Fan *et al.* (2014), Survila *et al.* (2016), and Yildirim *et al.* (2016), we have identified a set of computational bottlenecks of older TauDEM versions and improved the numerical performance and the parallel algorithm for the two flow direction functions in TauDEM by eliminating bottlenecks in file IO and runtime communication and developing more performant parallel algorithms. In HAND computation, the accelerated TauDEM reduced the HUC6 level HAND computation from 4.42 CPU years to 1.34 CPU years with the adoption of the new flow direction parallel algorithm. In the next section, detailed performance gain is presented.

HEIGHT ABOVE NEAREST DRAINAGE (HAND) COMPUTATION

The HAND methodology has been previously proposed and applied at local watershed scale. The computation of HAND for CONUS focuses on establishing an efficient and scalable scientific workflow that is able to derive HAND and associated hydrologic information from big terrain and hydrography data as an automated process. The automated process can then be easily adapted to terrain and hydrography data of varied resolution and scale. In this section, we describe the data and information flow in HAND computation and discuss insights gleaned from generating the HAND at CONUS scale.

HAND Workflow and Computational Analysis

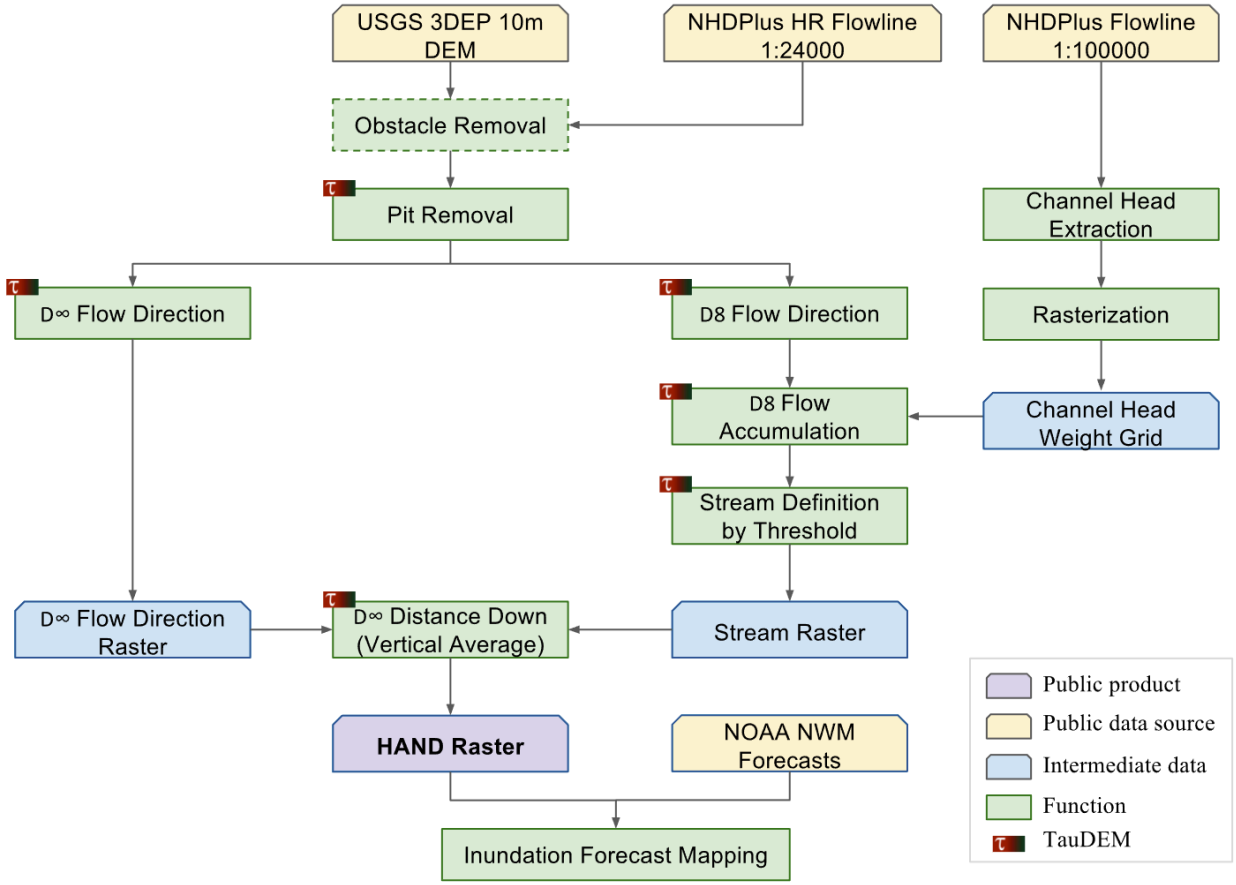

FIGURE 5. HAND computational workflow for a HUC unit.

Figure 5 illustrates the HAND computational workflow. HAND, by its definition, is a geospatial raster dataset in which the value of each cell is the height above its nearest drainage. To identify the nearest drainage, we construct a flow direction grid of the same resolution and spatial extent using the *D*∞ flow direction representation (Tarboton, 1997). The *D*∞ grid is derived by TauDEM from the hydrologically conditioned DEM of the studied HUC unit. The hydrologic conditioning consists of two steps: obstacle removal and pit removal. The obstacle removal function adjusts the elevation of DEM cells that are on identified river flowlines but blocked by topographic structures such as roads and dams. The flowlines used for obstacle removal come from the high resolution NHDPlus dataset, which has more than 30 million reaches, in order to improve the coverage and accuracy. The methodology of obstacle removal is being prototyped and, thus, is currently integrated into the HAND workflow as an abstract interface. The pit removal function calls TauDEM's *pitremove* function (Tarboton *et al.*, 2008) which takes the raw DEM as input. The input DEM for the specified HUC unit is generated by clipping 3DEP DEM using GDAL. The clipping function uses the USGS NHD Water Boundary

Dataset (WBD) to retrieve the boundary shape of a HUC unit and creates a 10-kilometer DEM clipping buffer to avoid edge effects along the HUC boundary. Since the 3DEP DEM is organized as a virtual raster in VRT format, there is no need to create a single DEM with CONUS coverage for clipping purposes.

With the *D*∞ method, a flow routing network is constructed on a grid by analyzing the topographic data only. The next step in HAND is to compute a stream network in raster format, where streams are rasterized cells on a grid of the same resolution and spatial extent. The stream grid is also derived using the DEM model, with guidance from NHDPlus. This process includes a series of vector and raster processing functions. The vector processing step takes the Flowline layer of the medium resolution NHDPlus to identify channel heads. Each flowline feature has two attributes: FromNode and ToNode. Channel heads are identified if the FromNode of the corresponding flowline is not a ToNode (downstream) of any other flowlines. The output of this step is a point dataset, which is then rasterized to create a channel head weight grid. TauDEM's *D8* flow direction function generates a *D8* flow direction grid from the same hydrologically conditioned DEM used in calculating the *D*∞ grid. The *D8* grid is then used by TauDEM's flow accumulation function (*aread8*) to generate weighted accumulated areas using the channel head weight grid. The *threshold* function in TauDEM is called with threshold value 1 to generate the stream grid from the *aread8* output. The result of this is a stream grid aligned with the DEM but initiated at the source of each NHDPlus stream. Taking the *D*∞ flow direction grid and the stream grid as input, TauDEM's Distance to Streams function (*DistanceDown*) produces the HAND value of each cell using the vertical distance measure. We can also use the horizontal distance or the combination of horizontal and vertical distance in this function if these additional distance grids are of interest.

The HAND workflow does not assume a particular resolution of input DEM and the NHDPlus dataset. For the handling of NHDPlus, however, the vector processing varies by resolution because of the difference in data layer structure at those resolutions. For example, in the medium resolution dataset, all the flowlines of a HUC unit can be queried by matching the *REACHCODE* attribute of the Flowline layer with the HUC code. In the high-resolution NHDPlus, two layers (*NHDPlusBurnLineEvent* and *NHDPlusFlowlineVAA*) that have spatial object information and *REACHCODE* information, respectively, need to be joined to retrieve FromNode-ToNode relationship for a flowline.

All of the input, intermediate output, and HAND data for a HUC unit are packaged as a data bag, a similar data model used in HydroShare (Horsburgh *et al*., 2016). All the input and intermediate output raster data must have the same spatial extent, resolution, and cell size, which may be a limitation for incorporating input data with different size of these attributes. In that case, additional geospatial processing is needed to align the raster data. There is no need for reprojecting DEM since TauDEM is able to handle both projected and geographic coordinate systems. The spatial projection in NHDPlus is automatically converted to DEM projection when the channel head weight grid is generated.

Computational Experience

We conducted the first HAND computation for 331 out of the 336 HUC6 accounting units. Each unit comprised a computing job that was submitted to ROGER HPC. Each job used 60 to 180 processor cores based on a coarse estimation of computational intensity, described in the section, "Scalable Computing." The first run was completed on April 16, 2016 and consumed a total of 4.42 CPU years. On average, each unit used 65.6 cores and took 1.78 hours to compute. The first run took about 8 days to finish on the shared ROGER HPC job queue. Figure 6(a) depicts the computing time of all 331 jobs. The large variation shows a heterogeneous computing profile for the 331 HUC6 units of different sizes, topographical, and hydrologic characteristics (e.g., the number of pits, flat regions and their sizes). Among the TauDEM functions called in the workflow, the two flow direction algorithms (*D8* and *D∞*) took, on average, 72.65% of the workflow computing time. The first run was conducted as a stress test to calibrate a more accurate computational intensity estimation for the units. Using the computation profile obtained from the first run, the workflow was adjusted for better configuration of edge contamination, DEM buffer size, and inlet identification from rivers passing through a watershed unit. This information also helped us capture CPU and memory requirements.

Following the first run, the second run was completed on May 29, 2016, using the calibrated workflow and newly accelerated *D8* and *D∞* algorithms (Survila *et al.*, 2016). Figure 6(b) depicts the computing time for all of the 331 units. The second run finished in 36 hours and consumed 1.34 CPU years in total. On average, each job used 65.26 cores and took 0.54 hours to compute. The two-flow direction algorithms took only 12.65% of the workflow computing time, on average. The majority, 70.57% time, was spent on GDAL commands for pre- and post-processing. The total input and output of HAND for CONUS takes about 5TB disk space.

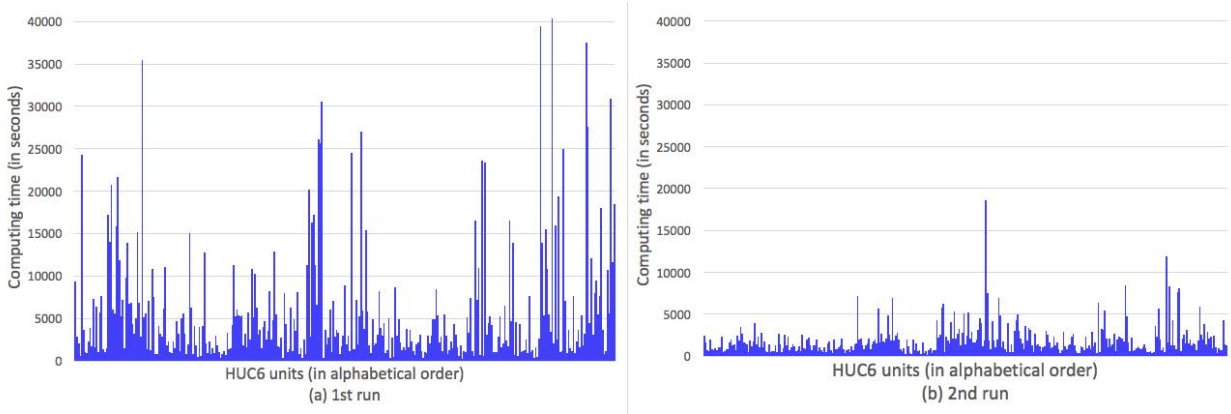

FIGURE 6. Computing time distribution: comparison between the first (a) and second (b) run.

The main source of acceleration in TauDEM's flow direction algorithms (Tarboton, 1997) is illustrated in Figures 7 and 8. A computational strategy was applied to allow multiple

processors to efficiently compute the flats resolving function, the most costly function in the two flow direction algorithms. A flat is a set of contiguous cells on DEM with same elevation or zero slope value. Determining the flow direction on flat cells requires an iterative algorithm that is computationally costly. Figure 7 shows the distribution of 16,560,871 flats on a hydro-conditioned DEM of a HUC6 unit. The original TauDEM uses an implicit communication mechanism to exchange ghost zone data on the boundary of the decomposed data domains in each MPI process. This mechanism has the benefit of hiding the inter-process communication complexity with automatic ghost zone data exchange after each iteration of the flats resolving function. However, this mechanism introduces significant communication cost as more processors are used to analyze larger DEM, as demonstrated by the performance difference between the first and second HAND run. A strategy to reduce the communication cost by locating and localizing flats resolving was developed to process local flats that are fully contained in spatial domains on a MPI process without any communication. Flats whose boundary shape cross multiple processes are shared flats and processed via MPI communication functions. The identification of local and shared flats was efficiently implemented with $O(n)$ computing complexity.

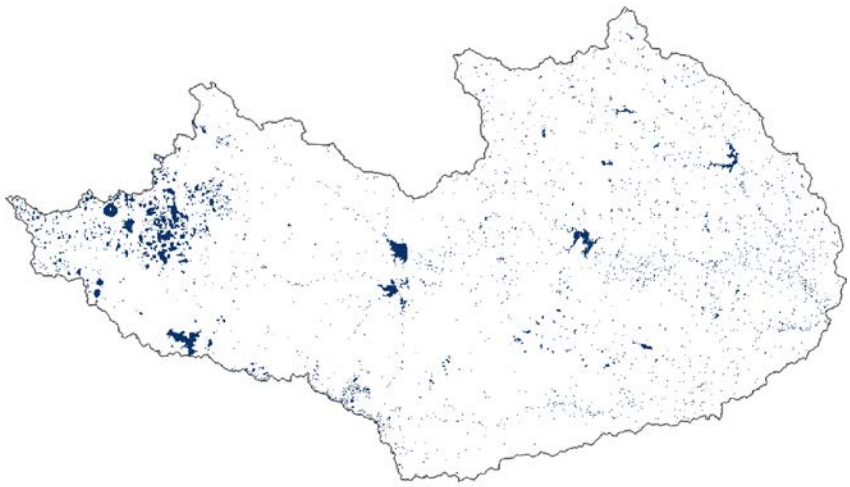

FIGURE 7. Illustration of the map of flats for HUC6 unit 120901, Middle Colorado-Concho. Flats in hydrologically conditioned DEM may include natural flats, flat surfaces in DEM (such as water surface where elevation information on the water channel beneath the water is not available), and filled pits.

With this strategy, an experiment on the *D8* algorithm was conducted on 4 computing nodes, using 1 to 32 processors, to evaluate the performance of flats resolving, the major bottleneck in both *D8* and $D\infty$. TauDEM version 5.3.7, which has not yet incorporated the acceleration code, is used for comparison. For the parallel runs, using 4, 8, 16, and 32 processors, respectively, 90.81%, 89.98%, 76.03%, and 66.89% local flats are identified and processed without inter-processor communication. The performance gain, measured as the time taken to

finish the *D8* function, is shown in Figure 8. In TauDEM version 5.3.7, the flats resolving function takes the majority of the computing time in all cases, although both the flow direction function and the flats resolving function scale well as the number of processors increases. With the flats resolving acceleration, this function is no longer a bottleneck. The execution time of *D8* algorithm decreased from 3.4 hours to 6 minutes and 11 seconds on one processor and 1227.22 seconds to 77 seconds on 32 processors. Using 32 processors, the flats resolving function requires only 2.36 seconds, compared to 1152.72 seconds on TauDEM version 5.3.7. In the accelerated version, the slightly worse performance of the *D8* function using 32 processors, compared to using 16 processors, indicates that the parallel IO costs outweigh the benefits from employing more than 16 processors on the 2.18GB DEM.

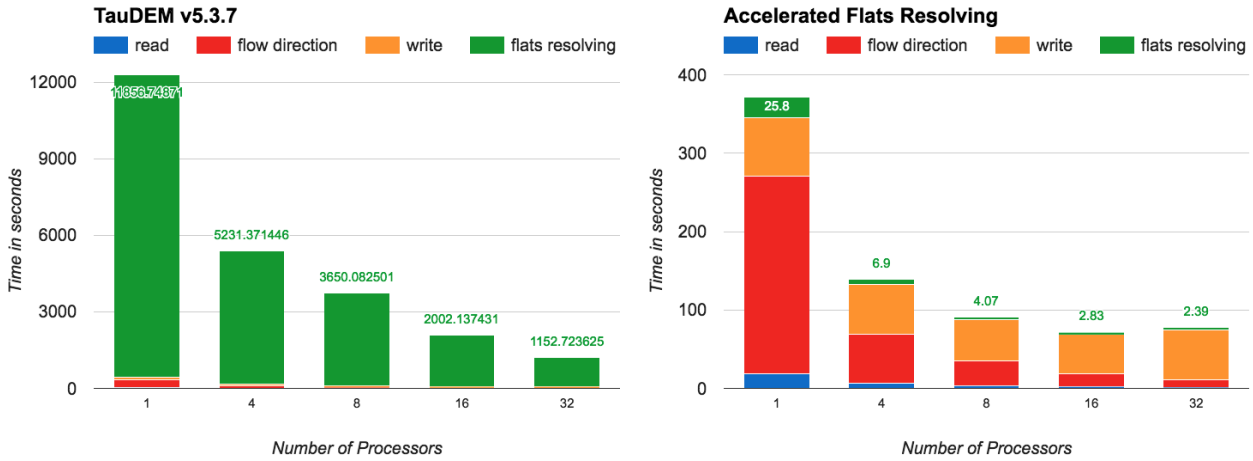

FIGURE 8. Performance of the *D8* flow direction algorithm before and after the acceleration on flats resolving. HUC6 unit: 120901. DEM size: 42877 x 21711, 2.18GB. For visual purpose, the range on Y-axis is plotted based on actual maximal values obtained in the two tests, respectively.

We also note that as the number of processors increases, the row-wise decomposition in TauDEM dramatically increases the number of shared flats—the decomposed stripes become narrower in vertical dimension and so more likely to split flats onto different MPI processes. We are exploring better decomposition strategies, e.g., block-wise and/or adaptive decomposition, for resolving this issue.

Results and Evaluation

Figure 9 shows the HAND map for CONUS, generated from the second run. Each HAND raster of an HUC unit is published as an OGC TMS map layer on the ROGER cloud. A CONUS layer is created by merging overlapping tiles. The availability of this HAND dataset piqued our interest in evaluating the results and identifying methodological improvements, which is the described in related work (Zheng *et al.*, 2017). All of the data and visualization layers are published online (NFIE CFIM data site. Accessed March 10, 2017, http://nfie.roger.ncsa.illinois.edu/nfiedata/) to engage further community evaluation.

The HAND experiment successfully demonstrated the feasibility of automating the calculation of HAND at 1/3rd arc-second resolution with CONUS coverage and the effectiveness of the proposed computational model. All 331 HUC6 units are computed using the high-throughput computing model. Because the ROGER HPC is a shared resource, accurate estimation of the turnaround time is difficult. Based on the established computational profile in the second run, we are developing fine-grained methods to schedule HAND jobs based on resource reservation. The computation performance of a single HUC unit is driven by the continuous acceleration of TauDEM. At the time of this paper's writing, we are able to finish the *pitremove* function on the entire 1/3rd arc-second DEM for CONUS in 2 hours (Yildirim *et al.*, 2016).

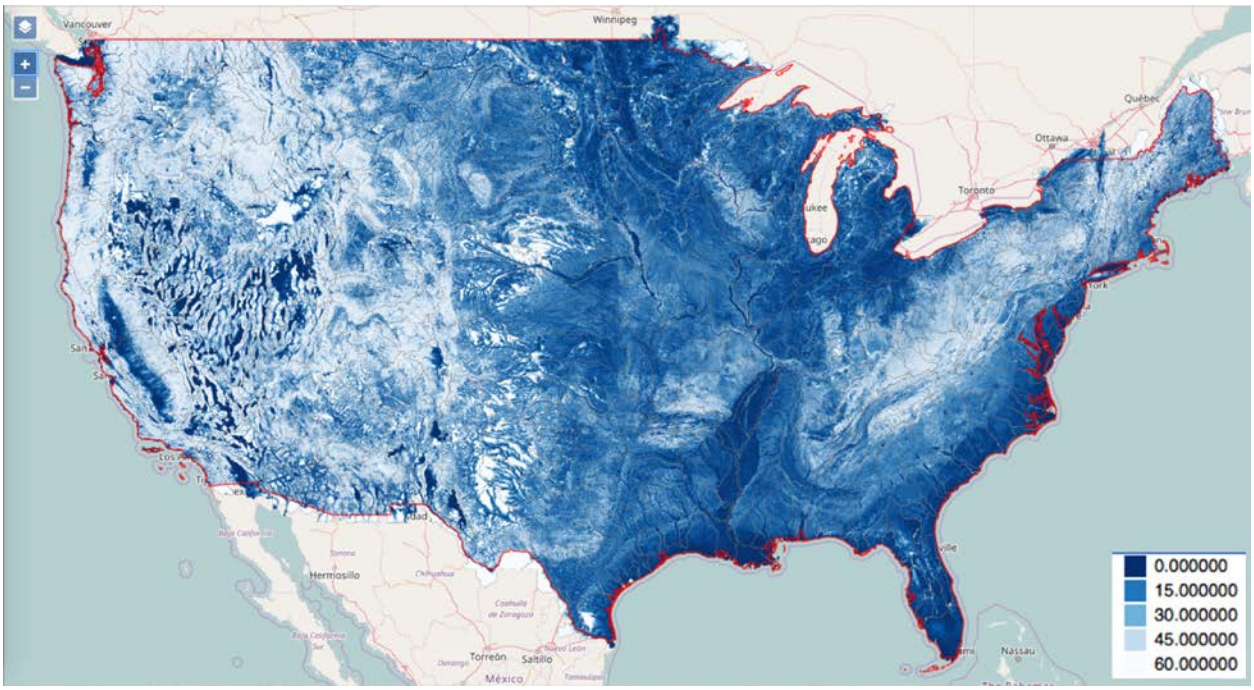

FIGURE 9. CONUS view of the 1/3rd arc-second HAND with U.S. 5km boundary, as of May 29, 2016. Projection: Web Mercator (EPSG:3857).

The HAND workflow is scalable to higher resolution DEM and NHDPlus. An experiment on a HUC6 unit was conducted to analyze the scalability at each step of the HAND workflow, shown in Figure 10. DEM resolution is the main determinant to the execution time of HAND, shown by 3m columns (5.4GB, 80160 x 48058 cells) and 10m columns (595MB, 26730 x 16025 cells). The most expensive functions in 3m DEM computation are the sequential DEM clipping and post-processing (i.e., removing the 10-kilometer buffer when creating HAND) by GDAL. In contrast, TauDEM performance scales well in proportion to DEM size. The major impact of using NHDPlus high-resolution dataset is on the performance of flowline retrieval because the join function operates on 30 million vectors, instead of 2.69 million in the medium resolution dataset.

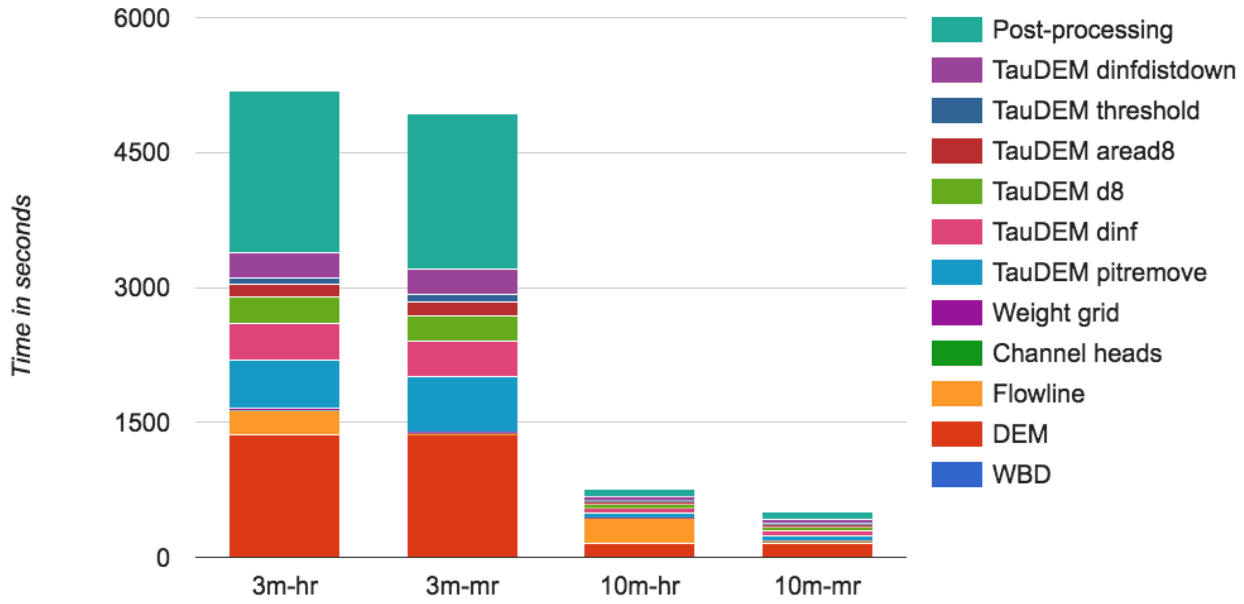

FIGURE 10. HAND scalability to 3DEP DEM and NHDPlus resolution. Study watershed: HUC6 unit 120402, Galveston Bay-Sabine Lake. DEM resolutions: 1/9th arc-second (3m) and 1/3rd arc-second (10m). NHDPlus resolutions: 1:100,000 (mr) and 1:24,000 (hr). 6 computing nodes (120 processors) on ROGER were used in each case.

HYDRAULIC PROPERTY TABLE AND INUNDATION MAPPING

HAND is a reference dataset from which hydraulic properties for each river reach defined in NHDPlus can be derived as another reference dataset for real-time flood inundation forecast—the hydraulic property table. For a list of specified stage heights, the hydraulic property table establishes the rating curve of each river reach and needs to be computed only once. Two major properties recorded for inundation mapping are stage height and the corresponding discharge calculated from HAND. In real-time forecast scenarios, the streaming forecast data from NOAA NWM provides the estimated discharge for each river reach at each forecast timestamp. Since NHDPlus and NWM forecast share the common index of river reach or catchment (i.e., COMID in NHDPlus and the station ID in NWM), the inundation information for each grid cell on HAND can then be calculated in three steps: 1) indexing grid cells with COMID; 2) calculating the water depth of each river segment at the specified forecast timestamp by looking up the hydraulic property table with the estimated discharge; and 3) comparing the estimated water depth with the HAND value at each grid cell to determine the inundation depth.

Figure 11 shows the computation workflow of inundation mapping at CONUS scale. It has three major components. First, the hydraulic property table is calculated by a series of raster and vector computing that takes the HAND data package, a pre-defined stage height list, and NHDPlus as input. This computation is decomposed at the HUC6 level into 331 computing jobs. The output tables for each unit are then merged as the CONUS-level table. The hydraulic

property table, in CSV or NetCDF4 format, is published as a public dataset. Second, an inundation forecast table is computed for each NWM forecast timestamp and stored as either CSV or NetCDF4 files. This table is computed at CONUS scale directly since it does not introduce significant computing cost. Third, the inundation mapping visualization process is invoked at the HUC6 level to generate map layers for each forecast table. A CONUS view of the inundation map is generated by merging HUC6-level map tiles. Both the forecast table and map layers are published online.

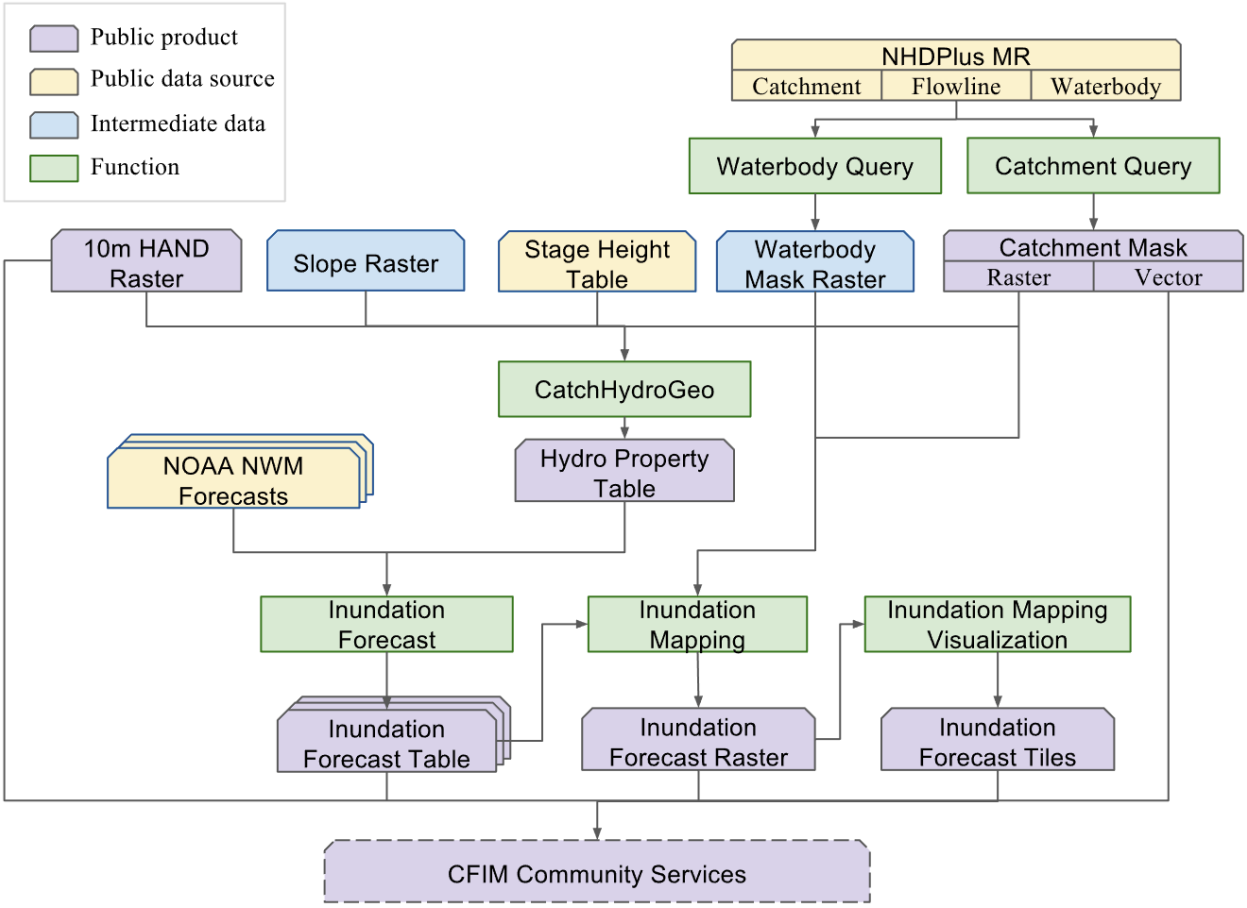

FIGURE 11. Preliminary inundation mapping workflow in CFIM.

Computing the Hydraulic Property Table

A list of hydraulic properties can be computed for each catchment along with corresponding stage height from HAND and NHDPlus, including surface area, bed area, volume, top width, wetted perimeter, wet area, hydraulic radius, and discharge. In the current configuration, the stage height table includes 82 heights, increasing above zero at one-foot interval.

A number of auxiliary attributes needed for the computation of these properties, such as

catchment ID, river segment slope, and length, are available in the NHDPlus file geodatabase. In order to carry the catchment ID (COMID) information embedded in the vector format of NHDPlus into the raster computing of HAND, a catchment grid of the same spatial extent, projection, and resolution as HAND is generated first. Catchment polygons in an HUC6 unit is retrieved from the *Catchment* layer in NHDPlus and then rasterized. A high-performance raster processing function, *CatchHydroGeo*, is developed using the TauDEM parallel computing framework to derive hydraulic properties from HAND, the catchment grid, the stage height table, and the slope grid which is the output of the TauDEM $D\infty$ flow direction function, as shown in Figure 11.

In this computational process, the query for the catchment polygon of a COMID can be implemented by directly querying the NHDPlus file geodatabase using GDAL's *ogr2ogr* tool. However, as we noted when we computed HAND, these direct queries perform poorly because the query involves an inner join of two layers: Flowline and Catchment, which have 2,691,344 and 2,647,454 records, respectively. If there are *n* river reaches in an HUC6 unit, in a non-optimized database implementation, the inner join may take (*n* x 2,691,344) lookup operations to obtain the catchment polygon information. To address this computational bottleneck, the catchment polygon query function is implemented using a pre-sized hash table as a Python script. This script requires only one scan of the Flowline layer to set up the hash table, and takes (*n* x *log(*2,691,344*)*) lookup operations to finish.

The output NetCDF4 table for each HUC6 unit is merged to the CONUS NetCDF4 table using a sequential Python script. Since there is no COMID overlapping between any two HUC6 units, there is no need to handle the intricacy of merging catchments that cross more than one HUC6 unit. This, however, may be an issue if non-HUC-based spatial domain decomposition strategies are used.

Computing for Real-time Inundation Forecast

The availability of the hydraulic property table allows straightforward translation of water depth information from NOAA NWM streamflow forecast information. The NWM discharge forecast of a river reach (station) is linearly interpolated to the water depth by looking up the hydraulic property table, which has the (*water depth* : *discharge*) mapping for each stage height defined by the stage height table. In addition, other inundation criteria specific to certain communities, e.g., anomaly map for emergency management, can be incorporated in this process. The computation of the inundation forecast table engages the processing of two relational tables, i.e., the hydraulic property table and the NWM forecast, and does not involve geospatial processing.

Computational Requirements for real-time CONUS inundation mapping visualization

The inundation forecast table can be used by hydrologists and others who understand the NHDPlus and NWM. For users who need to make decisions based on inundation mapping visualization, a preliminary process was developed to understand the computational requirements for continental-scale inundation mapping visualization and explore geospatial streaming computing solutions that support high-performance and scalable real-time inundation visualization.

In our current prototype, the generation of inundation maps for each NWM forecast includes two steps, as shown in Figure 11. First, an inundation map raster is generated by comparing the HAND value of a grid cell and the forecast water depth, using the catchment grid in order to identify the relevant cell. If a cell's forecast water depth is larger than the HAND value, it is marked as inundated. A masking option is available to show or hide cells covered by the masking layer. For example, all of the water body areas are masked because the CFIM model is not suitable for inundation analysis on water body objects. The inundation map is generated at the HUC6 level so that all the units can be computed in parallel. Second, the visualization function takes the GeoTIFF file of a unit's inundation map and a coloring scheme and generates a tile pyramid that covers multiple zoom levels. The tiles are published as OGC TMS map layers for visualization. Aggregating HUC6-level tiles into a single CONUS map layer is straightforward using image overlay techniques.

The computation using the current CONUS inundation mapping visualization solution is extremely intensive, though. The inundation map raster generation step needs to read and write multiple rasters of HAND size. In addition, the visualization step requires an enormous number of file I/O operations to create small tile images.

We conducted a CONUS scale experiment on the entire inundation forecast process, which includes the generation of inundation forecast table, maps, and visualization tiles, to study the performance of our current solution on 15 computing nodes of ROGER HPC. GNU Parallel is used to manage the computation on the 15 reserved computing nodes. Since the purpose of this experiment is to understand the computational footprint, we did not use advanced scheduling strategies to optimize the turnaround time. Instead, random scheduling was used to keep the 300 processor cores on the 15 nodes busy.

In the experiment, the short-range NWM forecast at forecast initialization time 12:00:00am, March 23, 2017 UTC was used. Table 2 shows the time distribution of each step. The first three steps needed to generate the inundation forecast maps took 44 minutes 31 seconds on 15 computing nodes, which means using 15 or more computing nodes to generate inundation maps is sufficient to match the hourly pace of the short-range NWM forecast data streaming. Figure 12 illustrates the web GIS interface which shows one of the 15 inundation forecast maps.

TABLE 2. Execution time of the inundation map generation process, in seconds.

	NWM Download	Forecast Table	Forecast Map	HUC6 TMS	CONUS TMS
Time	49	603	1779	27,845	13,892
Data size	780MB unzipped (52MB x 15 forecasts)	889MB (60MB x 15 forecasts)	223GB (4901 maps for 331 HUC6 units)	45GB (4,140,833 tiles; 8 zoom levels)	35GB (2,405,624 tiles; 8 zoom levels)

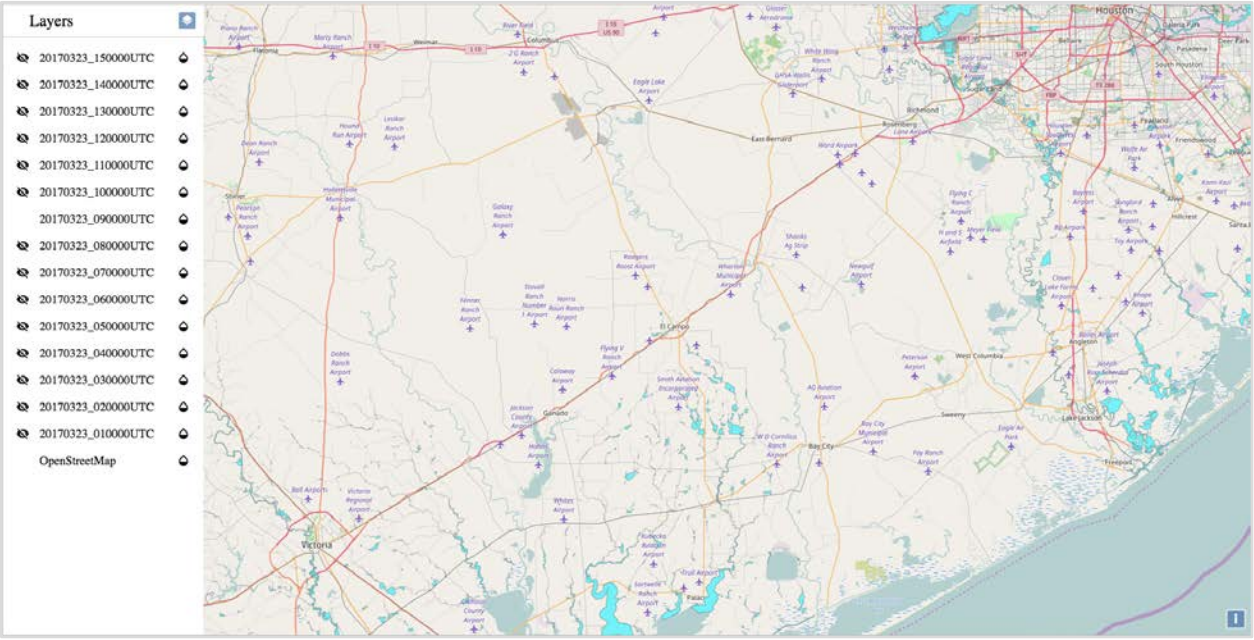

FIGURE 12. Inundation forecast map for NOAA NWM forecast initialization time 12:00:00am, March 23, 2017 UTC. Fifteen hourly maps based on the short-range forecast are generated. Map shows only the southern Houston area for visual purpose. A national view is visually insignificant because inundation occurs at local level.

The TMS-based visualization computation, however, required almost 12 hours to generate the CONUS view. In the TMS computation, the 4,140,833 tile images (each a 256x256 PNG file) were first created on zoom level 5-12 for each HUC6 unit. A merging process was then applied to generate the CONUS view with 2,405,624 tile images, in which 1,211,949 needed an image merging operation because rectangle tiles overlap across multiple HUC6 units. The visualization computation performed poorly for two reasons. First, pyramid tiling in GDAL does not have parallel computing functionality. Second, although we applied embarrassingly parallel computing on the 331 HUC6 units for TMS tile pyramiding using 15 computing nodes, tile pyramiding creates many small tile images, resulting in poor I/O performance on the GPFS file system. Significant time was spent creating 6,546,457 small tile image files at the HUC6 and CONUS level combined. In addition, tile pyramiding wastes copious disk space. Each map requires a TMS tile pyramid that consists of tens of thousands of small image tiles, taking much more disk space than the sum of the file sizes. This occurs because, on our GPFS system, the

smallest allocable file block is 32KB. If a tile image is less than 32KB, disk space is wasted. The size of most of the tile images ranges from 1KB to 50KB. Since the number of accumulated forecasts increases along the temporal dimension, the current solution exhibits poor scalability.

An immediate goal is to improve the responsiveness of the inundation forecast visualization solution. We are working on a dynamic mapping process that is based on online analytical processing (OLAP. Accessed March 10, 2017, http://olap.com) and does not require physical map and tile storage. The dynamic mapping will query HAND and a few auxiliary rasters, the hydraulic property table, and the inundation forecast table directly, and render the inundation map on the fly. Once implemented, the last three steps in the current process will not be needed, saving 12.5 hours of computing time and 303GB disk space. The requirements for the inundation mapping at CONUS scale will only include 10 minutes 52 seconds and 1.669GB for the first two steps in the current solution and a mapping service that renders the visualization map dynamically upon user request. Since the data to be transferred to a browser screen is a linear factor of the spatial extent in the browser window at a certain zoom level, the performance of our new solution will be sufficient to meet the requirements for real-time inundation forecast, e.g., the animation of a series of forecast maps. A potential technical solution is to use Esri's mosaic dataset technique (Esri, Mosaic Dataset. Accessed March 10, 2017, http://desktop.arcgis.com/en/arcmap/latest/manage-data/raster-and-images/what-is-a-mosaic-dataset.htm) to organize raster input for inundation mapping and develop raster functions (Esri, Raster Function. Accessed March 10, 2017, http://desktop.arcgis.com/en/arcmap/latest/manage-data/raster-and-images/python-raster-function.htm) for dynamic mapping. Esri has collaborated with NOAA and the research community since the inception of NFIE and NWM development and provides free interactive temporal map services in ArcGIS Online (Esri, ArcGIS Online. Accessed March 10, 2017, https://www.arcgis.com/home/) to visualize the forecast results and forecast anomaly. Our collaborators at Esri are now developing tools for applying the NWM forecasts to the HAND to create interactive temporal image services for use in web mapping applications. Using the runoff forecast and rating curve, a python raster function dynamically determines if a given cell in the HAND will be flooded or not. To make this scalable to CONUS, two referenced mosaic dataset are created. HUC6 inundation maps for a given forecast timestamp is organized into a virtual raster as the first mosaic dataset. The second mosaic dataset organizes all forecast timestamps along temporal dimension. Raster functions are then invoked upon user requests, which specify both the spatial extent and the temporal range for inundation computation.

DISCUSSION AND CONCLUSION

We have developed a cyberGIS framework for achieving integrated high-performance data processing, analysis, modeling, and visualization that enable continental-scale flood inundation mapping on advanced cyberinfrastructure. The framework exploits ROGER, the first cyberGIS supercomputer with a hybrid supercomputing architecture, to provide a prototype

platform for efficient data, software, and computation integration, as well as collaboration for methodology development. A scalable computing solution based on two-level parallelization is developed to compute three major data products of CFIM: HAND, the hydraulic property table, and inundation forecast table and map.

We successfully demonstrated the computational feasibility of continental-scale flood inundation mapping with the cyberGIS framework. The computation of HAND for the relevant 331 HUC6 units on CONUS achieved a turnaround time of 1.5 days on the ROGER supercomputer. An additional 2.5 hours was taken to compute the hydraulic property table and store it in NetCDF4 format. The inundation forecast process took 45 minutes for producing 15 inundation tables and maps (excluding the TMS tile pyramiding step) on a short-range NOAA NWM forecast initialization time stamp for CONUS coverage.

The availability of HAND at 1/3rd arc-second resolution and CONUS scale has auspicious, broad, and significant research implications, opening the door for pertinent research communities to conduct large-scale flood inundation mapping research by pertinent research communities. The CFIM collaboration resulted in significant scalability and performance improvement of cyberGIS and TauDEM software. The CFIM computational model is based on open source geospatial and hydrologic software that is able to harness massive cyberGIS computing power for enabling the computation of the CFIM workflow. The computation on ROGER seamlessly exploits its HPC and cloud components for workflow methodology development and CFIM workflow computation, visualization, and validation. The CFIM practice demonstrated the unique capability of ROGER as an integrated cyberGIS infrastructure for large-scale geospatial computation to support scientific research.

Future Work

Our ongoing work continues to improve the CFIM workflow to couple related hydrologic modeling processes for producing flood inundation forecasts at high spatial and temporal resolutions. Since our current framework provides a scalable solution, turnaround time for the computation of major CFIM components can be reduced by simply employing additional computing power. Our next goal is to accelerate the current CFIM computational workflow. We are working with TauDEM team to incorporate more performant strategies in data decomposition and runtime communication. We will support block-wise decomposition with VRT output to provide a set of decomposition choices (row-, column-, block-wise) that can be applied according to spatial data characteristics. The VRT output will generate a collection of raster tiles of moderate size for researchers to evaluate on desktop GIS, since an output raster of tens of gigabytes in size is difficult to use even when our framework can generate it. We will use MPI non-blocking calls to support asynchronous communication, eliminating potential synchronization bottlenecks in large-scale runs on thousands of processors (Liu *et al.*, 2015). The continuous acceleration of TauDEM is important for expanding our computational capabilities

for higher resolution DEM analysis at the CONUS scale. For instance, it is desirable to run the enhanced TauDEM on the entire 1/3rd arc-second 3DEP DEM to calculate HAND, which has the benefit of removing boundary issues between adjacent HUC units. The HUC-based DEM domain decomposition mechanism is still useful in analyses at 3-meter, 1-meter or LiDAR-derived sub-1-meter DEM resolutions. A more efficient inundation forecast visualization solution is being developed to address the responsiveness requirements for real-time inundation forecast data and visual analytics.

One of the design goals of our CFIM framework is to facilitate future continental hydrology research. Preliminary results obtained for HAND, the hydraulic property table, and the prototype inundation mapping process have identified a rich set of research topics concerning the underlying models and methodologies. We are currently working to improve related methodologies and conducting rigorous validation processes to quantify uncertainties of the approach from both geospatial and hydrology perspectives. We expect the further development of our framework and future data products to engage broader communities for serving various research and educational purposes.

The preliminary data, software, and results produced in this work have been made available online at http://nfie.roger.ncsa.illinois.edu/nfiedata/. We will continue to improve the open source software code to further componentize and automate the CFIM workflow. The data, software, and computing environment are being containerized for better broad community evaluation and outreach. An interactive online environment based on CyberGIS Jupyter and HydroShare is being developed to provide a powerful online problem-solving environment for the co-development of methodologies, education and training, and other community engagement activities.

ACKNOWLEDGEMENTS

This research is supported in part by USGS under grant number G14AC00244 and NSF under grant numbers: 1047916, 1343785, and 1443080. The work used the ROGER supercomputer, which is supported by NSF under grant number: 1429699. HydroShare is being developed under NSF grants ACI 1148453 and 1148090. This work is part of the ECSS project (award number ENG140009) of XSEDE that is supported by NSF under grant number 1053575. TauDEM was enhanced to support parallel computing and integrate with GDAL libraries with support from the US Army Corps of Engineers contract numbers W912HZ-11-P-0338 and W91238-15-P-0033 and XSEDE ECSS allocation EAR130008. The authors are grateful for the insightful discussions with Steve Kopp and Dean Djokic at Esri, and Larry Stanislawski at USGS. The authors would like to thank Dandong Yin at the University of Illinois at Urbana-Champaign for his development of the illustrative HAND Jupyter notebook.

LITERATURE CITED

Bockelman, B., T. Cartwright, J. Frey, E. M Fajardo, B. Lin, M. Selmeci, T. Tannenbaum, and M. Zvada, 2015. Commissioning the HTCondor-CE for the Open Science Grid. *Journal of Physics: Conference Series* (664).

David, C.H., D.R. Maidment, G.-Y. Niu, Z.-L. Yang, F. Habets, and V. Eijkhout, 2011. River Network Routing on the NHDPlus Dataset. *Journal of Hydrometeorology* 12(5):913-934.

Fan, Y., Y. Liu, S. Wang, D. Tarboton, A. Yildirim, and N. Wilkins-Diehr, 2014. Accelerating TauDEM as a Scalable Hydrological Terrain Analysis Service on XSEDE. In: *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*, Atlanta, GA, USA, July 13-18. ACM, New York. DOI: 10.1145/2616498.2616510

Horsburgh, Jeffery S., Mohamed M. Morsy, Anthony M. Castronova, Jonathan L. Goodall, Tian Gan, Hong Yi, Michael J. Stealey, and David G. Tarboton, 2016. HydroShare: Sharing Diverse Environmental Data Types and Models as Social Objects with Application to the Hydrology Domain. *Journal of the American Water Resources Association (JAWRA)* 52(4):873–889. DOI: 10.1111/1752-1688.12363

Hu, H., X. Hong, J. Terstriep, Y. Liu, M. Finn, J. Rush, J. Wendel, and S. Wang. 2016, TopoLens: Building A CyberGIS Community Data Service for Enhancing the Usability of High-resolution National Topographic Datasets. In: *Proceedings of the 2016 Annual Conference on Extreme Science and Engineering Discovery Environment (XSEDE'16)*. Miami, Florida. July 17-21. ACM, New York, pp. 39:1--39:8. DOI: 10.1145/2949550.2949652

Jones, N., J. Nelson, N. Swain, S. Christensen, D. Tarboton, and P. Dash, 2014. Tethys: A Software Framework for Web-Based Modeling and Decision Support Applications. In: D.P. Ames, N.W.T. Quinn, and A.E. Rizzoli (Editors), *Proceedings of the 7th International Congress on Environmental Modelling and Software*, June 15-19, San Diego, California, ISBN: 978-88-9035-744-2.

Lawrence, Katherine A., Michael Zentner, Nancy Wilkins-Diehr, Julie A. Wernert, Marlon Pierce, Suresh Marru, Scott Michael, 2015. Science gateways today and tomorrow: positive perspectives of nearly 5000 members of the research community. *Concurrency and Computation: Practice and Experience* 27: 4252–4268. DOI: 10.1002/cpe.3526

Liu, Yan Y. and Shaowen Wang, 2015. A Scalable Parallel Genetic Algorithm for the Generalized Assignment Problem. *Parallel Computing* 46: 98-119. DOI: 10.1016/j.parco.2014.04.008

Maidment, David R., 2016a. Conceptual Framework for the National Flood Interoperability Experiment. *Journal of the American Water Resources Association (JAWRA)* 53(2): 245-257. DOI: 10.1111/1752-1688.12474

Maidment, David R., 2016b. Open Water Data in Space and Time. *Journal of the American Water Resources Association (JAWRA)* 52(4):816–824. DOI: 10.1111/1752-1688.12436

Maidment, David R., Harry Evans, Xing Zheng, David Arctur, Dean Djokic, and Matt Ables, 2017. Texas Flood Response System. *Journal of the American Water Resources Association (JAWRA)*, Draft for review.

Nobre, A. D., L. A. Cuartas, M. Hodnett, C. D. Rennó, G. Rodrigues, A. Silveira, M. Waterloo and S. Saleska, 2011. Height Above the Nearest Drainage – a hydrologically relevant new terrain model. *Journal of Hydrology*, 404(1–2): 13-29. DOI: 10.1016/j.jhydrol.2011.03.051

Nobre, Antonio Donato, Luz Adriana Cuartas, Marcos Rodrigo Momo, Dirceu Luís Severo, Adilson Pinheiro, and Carlos Afonso Nobre. 2016. HAND contour: a new proxy predictor of inundation extent. *Hydrological Process*es, 30: 320–333. DOI: 10.1002/hyp.10581

Rennó, Camilo Daleles, Antonio Donato Nobre, Luz Adriana Cuartas, João Vianei Soares, Martin G.

Hodnett, Javier Tomasella, and Maarten J. Waterloo, 2008. HAND, a new terrain descriptor using SRTM-DEM: Mapping terra-firme rainforest environments in Amazonia. *Remote Sensing of Environment* 112(9) 3469-3481. DOI: 10.1016/j.rse.2008.03.018

Snow, Alan D., Scott D. Christensen, Nathan R. Swain, E. James Nelson, Daniel P. Ames, Norman L. Jones, Deng Ding, Nawajish S. Noman, Cedric H. David, Florian Pappenberger, and Ervin Zsoter, 2016. A High-Resolution National-Scale Hydrologic Forecast System from a Global Ensemble Land Surface Model. *Journal of the American Water Resources Association (JAWRA)* 52(4): 950-964, DOI: 10.1111/1752-1688.12434

Survila, K., A.A. Yildirim, T. Li, Y. Liu, D.G. Tarboton, and S. Wang, 2016. A Scalable High-performance Topographic Flow Direction Algorithm for Hydrological Information Analysis. In: *Proceedings of the 2016 Annual Conference on Extreme Science and Engineering Discovery Environment (XSEDE'16)*. Miami, Florida. July 17-21. ACM, New York, pp. 11:1--11:7. DOI: 10.1145/2949550.2949571

Tarboton, D.G., 1997. A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water resources research* 33(2): 309-319. DOI: 10.1029/96WR03137

Tarboton, D.G. and M.E. Baker, 2008. Towards an algebra for terrain-based flow analysis. In: *Representing, modeling and visualizing the natural environment: innovations in GIS* 12, NJ Mount, GL Harvey, P. Aplin and G. Priestnall (Editors). CRC Press, Florida. pp: 167-194. DOI: 10.1201/9781420055504.ch12

Tarboton, D. G., K. A. T. Schreuders, D. W. Watson, and M. E. Baker, 2009. Generalized terrain-based flow analysis of digital elevation models. In: *18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*, July 2009. R. S. Anderssen, R. D. Braddock and L. T. H. Newham (Editors). Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, pp. 2000-2006. URL: http://www.mssanz.org.au/modsim09/F4/tarboton_F4.pdf

Tarboton, D. G., R. Idaszak, J. S. Horsburgh, J. Heard, D. Ames, J. L. Goodall, L. Band, V. Merwade, A. Couch, J. Arrigo, R. Hooper, D. Valentine and D. Maidment, 2014. HydroShare: Advancing Collaboration through Hydrologic Data and Model Sharing. In: *Proceedings of the 7th International Congress on Environmental Modelling and Software*, San Diego, California, USA. D. P. Ames, N. W. T. Quinn and A. E. Rizzoli (Editors). International Environmental Modelling and Software Society (iEMSs), ISBN-13: 978-88-9035-744-2

Tesfa, Teklu K., David G. Tarboton, Daniel W. Watson, Kimberly A.T. Schreuders, Matthew E. Baker, and Robert M. Wallace, 2011. Extraction of hydrological proximity measures from DEMs using parallel processing, *Environmental Modelling & Software* 26(12): 1696-1709. DOI: 10.1016/j.envsoft.2011.07.018.

Tomlin, Dana C., 2012. Geographic information systems and cartographic modeling. Esri Press, California, ISBN-13: 978-1589483095

Towns, John, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, and Nancy Wilkins-Diehr, 2014. XSEDE: Accelerating Scientific Discovery. *Computing in Science & Engineering* 16(5): 62-74. DOI:10.1109/MCSE.2014.80

Wang, Shaowen, and Marc P. Armstrong, 2009. A Theoretical Approach to the Use of Cyberinfrastructure in Geographical Analysis. *International Journal of Geographical Information Science*, 23 (2): 169-193.

Wang, Shaowen, 2010. A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis. *Annals of the Association of American Geographers* 100(3): 535-557

Wang, Shaowen, Luc Anselin, Budhendra Bhaduri, Christopher Crosby, Michael F. Goodchild, Yan Liu, and Timothy L. Nyerges, 2013. CyberGIS Software: A Synthetic Review and Integration Roadmap. *International Journal of Geographical Information Science* 27(11): 2122-2145

Wang, Shaowen, Yan Liu, and Anand Padmanabhan, 2015. Open CyberGIS Software for Geospatial Research and Education in the Big Data Era. *SoftwareX* 5:1-5. DOI:10.1016/j.softx.2015.10.003

Wang, S., 2017. CyberGIS. In: *The International Encyclopedia of Geography*, Douglas Richardson, Noel Castree, Michael F. Goodchild, Audrey Kobayashi, Weidong Liu, and Richard A. Marston (Editors). John Wiley & Sons, Ltd. DOI: 10.1002/9781118786352.wbieg0931

Wilkins-Diehr, N., S. Sanielevici, J. Alameda, J. Cazes, L. Crosby, M. Pierce, and R. Roskies, 2015. An Overview of the XSEDE Extended Collaborative Support Program. In: *High Performance Computer Applications 6th International Conference, ISUM 2015*. Mexico City, Mexico, March 9-13. Gitler, Isidoro, Klapp, Jaime (Eds.) Springer International Publishing. ISBN-13: 978-3-319-32243-8, pp: 3-13. DOI: 10.1007/978-3-319-32243-8

Yildirim, A.A., D.G. Tarboton, Y. Liu, N.S. Sazib, S. Wang, 2016. Accelerating TauDEM for Extracting Hydrology Information from National-Scale High Resolution Topographic Dataset. In: *Proceedings of the 2016 Annual Conference on Extreme Science and Engineering Discovery Environment (XSEDE'16)*. Miami, Florida. July 17-21. ACM, New York, pp. 3:1--3:2. DOI: 10.1145/2949550.2949582

Zheng, Xing, David Tarboton, David R. Maidment, Yan Y. Liu, and Paola Passalacqua, 2017. River Channel Geometry and Rating Curve Estimation Using Height Above the Nearest Drainage. *Journal of the American Water Resources Association (JAWRA)*, Draft for review.