Received: 29 April 2021 | Accepted: 12 October 2021
DOI: 10.1002/cpe.6727

SPECIAL ISSUE PAPER

WILEY

Building a portal for climate data—Mapping automation, visualization, and dissemination

Jared McLean[1,2] | Sean B. Cleveland[1,2] | Michael DodgeII[1,2] | Matthew P. Lucas[3] | Ryan J. Longman[3,4] | Thomas W. Giambelluca[3] | Gwen A. Jacobs[1,2]

[1]Hawai'i Data Science Institute, University of Hawai'i, Honolulu, Hawai'i, USA

[2]Cyberinfrastructure, University of Hawai'i, Honolulu, Hawai'i, USA

[3]Water Resource Research Center, University of Hawai'i at Manoa, Honolulu, Hawai'i, USA

[4]East West Center, Honolulu, Hawai'i, USA

Correspondence
Jared McLean, 1659 Oneawa Pl, Hilo, HI 96720, USA.
Email: mcleanj@hawaii.edu

Funding information
National Science Foundation, Grant/Award Numbers: ACI #1450413, CSSI #1931439, CSSI #1931575, OIA #1557349

Summary

This article discusses the technologies and implementation of a climate data portal. This portal provides researchers and community stakeholders access to climatological data and resources, currently focusing on the state of Hawai'i. The portal provides interactive access to and visualization of hosted historical and near-real-time gridded maps and aggregated sensor station observational data. Climate data (currently precipitation and temperature) from sensor stations are collected, quality controlled, and processed daily to produce high-resolution gridded maps of climate data. A publicly available web application allows users to navigate the available data and visualize the produced gridded data products and sensor stations data on an interactive map element, view information about the sensor stations used to produce a given map, and generate time series from sensor station data. The portal can also generate packages of data to export from the application. The portal is designed to host and disseminate any climatological variables that can be processed into a set of observational data and gridded value maps. The established workflow and automation procedure is generally extensible to additional variables and will be used similarly to expand the scope of the portal.

KEYWORDS

climate, hydrology, precipitation, science gateways, sustainability, Tapis

1 | INTRODUCTION

A sustainable supply of clean water is essential for all aspects of life and there are multiple threats to maintaining this resource for future generations. Climate change is decreasing rainfall and increasing sea levels while population growth, urban development, and agricultural use continue to increase throughout Hawai'i, bringing into question what is sustainable. Knowledge of mean rainfall patterns in Hawai'i is critically important in support of the research and understanding of watershed modeling as well as for the management and protection of groundwater and surface water resources. Further, rainfall information can inform methods to control and eradicate invasive species, methods to protect and restore native ecosystems, and planning for the effects of global climate change.

The Hawai'i Established Program to Stimulate Competitive Research (EPSCoR) 'Ike Wai (Water Knowledge) project in partnership with the University of Hawai'i Water Resource Research Center (WRCC) has spent the last several years developing gridded monthly rainfall products. The gridded monthly rainfall products are highly useful for water resources analysis and planning, ecological modeling, and other applications. Prior to this work, maps were produced at irregular intervals of at least several years, and maps for recent months were generally not available. By automating and streamlining data acquisition, quality assurance and quality control (QA/QC), gap filling, interpolation, and dissemination this initiative intends to produce results in regularly updated maps with the goal of releasing a preliminary product within one month of the end of the most recent month. Having more up-to-date analyses available will potentially facilitate additional applications for these datasets, such as drought planning and wildfire management.

In addition to producing near-real-time data products going forward, available historical data can be processed using the same workflow to fills the gaps in pre-existing rainfall maps. Historical sensor station data are available at a monthly interval from 1920, and at a daily interval from 1990. The current production effort focuses on the monthly and daily rainfall data from 1990 onward; however, the larger monthly only data set extending back to 1920 will be used to produce a more extensive historical record to facilitate analyses of longer-term trends.

This workflow requires execution everyday, not only to generate near-real-time daily rainfall map products, but also to ensure the collection of transient precipitation data that becomes unavailable from some resource providers if not harvested within a window of availability. This use case could fit a cloud virtual machine (VM) architecture, like Jetstream,[1] that is always running, but the fact that the tasks complete within a few hours each day leads to inefficient use of the resource overall. There is also the additional overhead of patching, securing, and managing a VM server—in the case of Jetstream some of this is covered by the environment administrators but this is not the case for most VM hosting services. The Tapis actor based containers (Abaco) function-as-a-service (FaaS), or serverless computing service, provides a more suited architecture for this use case and was adopted by this workflow. This service allows short computational jobs to be executed more or less on-demand each day using containers, like Docker, integrated with CRON-like scheduling features. This architecture satisfies the execution frequency required and makes efficient use of computational resources.

In addition to the generation of these rainfall products, the dissemination and visualization of these products is crucial for engaging stakeholders and community members in utilizing this data. The portal includes an interactive web application interface for data exploration and decision support as part of the 'Ike Wai Gateway,[2] providing public access to the annotated data products of this workflow. This article presents the end-to-end workflow that leverages geographically distributed cyberinfrastructure, gateways, data repositories, and computational infrastructure with methods, such as containerization, that enhance provenance, tracking, and reproducibility to support climate research in Hawai'i.

2 | BACKGROUND

2.1 | 'Ike Wai

The 'Ike Wai project is the current Hawai'i EPSCoR Track 1 project. This project was established to help ensure Hawai'i's future water security through an integrated program of research, education, community engagement, cyberinfrastructure development and support, and decision support. The vision of the project is to promote water resource management in Hawai'i that is sustainable, responsible, and data driven. The scientific, cultural, and social dimensions of the problem of water security are integrated to provide a transparent, stakeholder-driven, and robust water research enterprise.

To support this mission a robust cyberinfrastructure solution such as a science gateway was required. Science gateways and their frameworks have become integral in providing and lowering the barrier of research infrastructure. These science gateways have become more popular and are being supported by organizations such as XSEDE[3] and the Science Gateways Community Institute (SGCI)[4-6] to accelerate science. There are several efforts to provide "as a service" gateways such as WS-PGRADE/gUSE project,[7] the HUBzero project,[8] Apache Airavata,[9,10] Galaxy,[11] and Globus.[12] Existing hydrology gateways and tools such as the CUAHSI Hydrological Information System (HIS),[13] Hydroshare,[14] and Virtual Observatory and Ecological Informatics System (VOEIS)[15] provide server side support, data entry, and quality control client tools. These systems, however, lack tools for managing and validating multi-domain metadata, integration for multiple data stores, and non-integration of computational resources or flexibility in the access and selection of resources.

Tapis[16] provides solutions to these shortcomings and was chosen as the framework to construct the 'Ike Wai gateway due the wide range of supported functionality enabling the above needs. Additionally, Tapis is supported by the Texas Advanced Compute Center (TACC), a partner of the University of Hawai'i, assisting with collaboration and development of the framework to suit the needs of the project. Tapis is an open source, scientific API middleware for hybrid cloud computing and data management that is powering a number of current community science gateways such as CyVerse,[17,18] which provides life scientists with powerful computational infrastructure and the ability to handle large datasets and complex analyses to enable data-driven discovery.[19]

The current 'Ike Wai gateway[2] supports research in hydrology and water management and provides tools to address questions of water sustainability in Hawai'i. The gateway provides centralized web-based user interfaces and representational state transfer (REST) APIs for multi-domain data management, computation, analysis, and visualization tools. These tools provide reproducible science, modeling, data discovery, and decision support for the 'Ike Wai research team and wider Hawai'i hydrology community. By leveraging the Tapis framework[16] UH has constructed a gateway that ties data management and advanced computing resources together to support diverse research domains including microbiology, geochemistry, geophysics, economics, and humanities. The data from this serverless workflow are housed, annotated, and indexed within the 'Ike Wai gateway.

2.2 | Tapis

Tapis is a National Science Foundation (NSF) funded web-based API framework for securely managing computational workloads across infrastructures and institutions, so that experts can focus on their research instead of the technology needed to accomplish it. As part of work funded by the NSF starting in 2019, Tapis is delivering a version 3 ("V3") of its platform with several new capabilities, including a multi-site security kernel, streaming data APIs, and high-level support for containerized applications.

Tapis provides FaaS through Abaco, which is based on the actor model of concurrent computation and Docker; users define computational primitives called *actors* with a Docker image, and Abaco assigns each actor a unique URL over which it can receive messages. Users send the actor a message by making an HTTP POST request to the URL. In response to an actor receiving a message, Abaco launches a container from the associated image, injecting the message into the container. Typically, the container execution is asynchronous from the message request, though Abaco does provide an endpoint for sending a message to an actor and blocking until the execution completes, providing synchronous execution semantics. Abaco maintains a queue of messages for each actor, and is capable of launching containers in parallel for a given actor when the actor is registered as *stateless*. The functions run with an authenticated context that allows them to make requests to other Tapis APIs to perform actions such as data transfer or job submission.

2.3 | Climate data in Hawai'i

The 'Ike Wai Gateway houses monthly and daily rainfall and temperature data obtained from a number of sources (see Longman et al.[20] for a comprehensive description of the climate data networks in Hawai'i). The near-real-time data utilized in the portal are obtained from several national, online data repositories including: the National Centers for Environmental Information (NCEI),[21] the Hydrometeorological Automated Data System (HADS),[22] and the Soil Climate Analysis Network (SCAN).[23] For analysis, all data are quality controlled, screened, and subsequently flagged for extraneous outlying values. Gaps are then filled using an optimized normal ratio approach.[20,24] Lastly, daily data are aggregated to the monthly time step and then used to create gridded monthly and daily rainfall and temperature maps for the State of Hawai'i at a 250 m × 250 m resolution. This workflow (Figure 1) produces rainfall maps using methods described by Lucas et al.,[25] for the seven major islands Kaua'i, O'ahu, Moloka'i, Lana'i, Maui, Kaho'olawe, and Hawai'i.

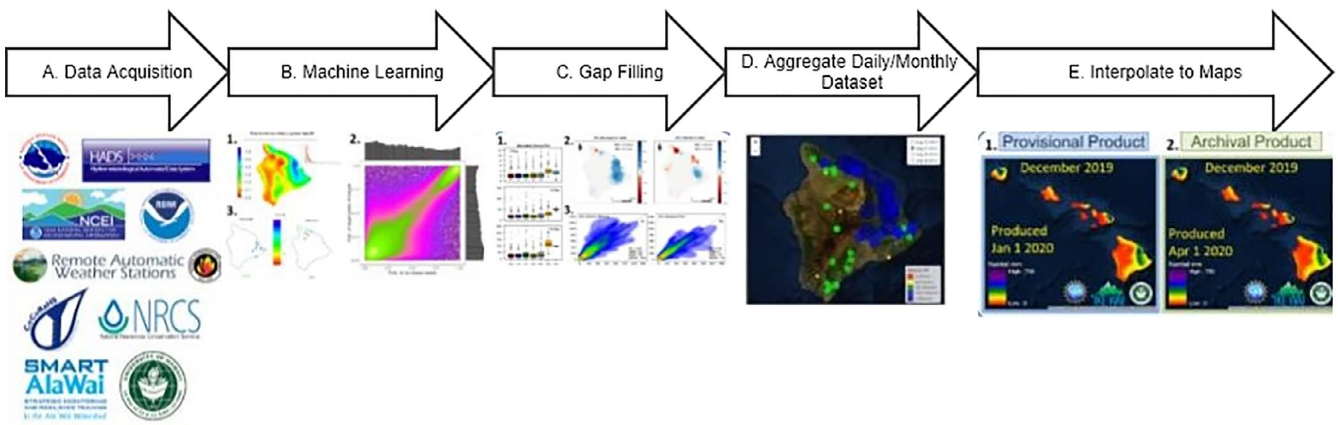

FIGURE 1 Hawai'i statewide rainfall data mapping workflow. (A) Rainfall station data are collected from multiple sources and processed to daily totals with online resources to make a provisional product. Additionally, after some delay, manually collected rainfall is used to produce an archival product. (B) A custom, random-forest machine learning algorithm uses the 10 closest stations to estimate if an observed rainfall value is true or a data collection error. (C) Data-driven gap filling of removed and missing values using best available methods to fill missing station values. (D) Daily processed, raw, and gap filled station data are produced. (E) Gap filled monthly station values are used to create 250 m × 250 m resolution gridded maps of monthly rainfall amount, anomaly, and respective standard errors

3 | NEAR-REAL-TIME DATA AGGREGATION AND MAPPING WORKFLOW

3.1 | Script development on Jetstream

The Jetstream platform has provided an effective development system for the rainfall workflow. A single VM (m1.small 2 CPUs, 4 GB of memory, and 20 GB of disk) using the base image "Ubuntu 18.04 Devel and Docker v1.11" (Jetstream image ID 717, Instance ID 26850, Alias: 93522185-0c41-4d9e-97c9-24ea6e2717e8) was used to host a development environment with R and RStudio. This VM enabled domain scientists to develop a series of R scripts for each step in the precipitation workflow (Figure 1) and test each component with tools familiar to their existing development workflow. The VM also provided a system that could run continuously for executing the daily workflows for fetching data with CRON tasks. Additionally, the VM provides a layer of reproducibility for the development environment as it can be exported, using an environment variable approach for smooth containerization. Each script requires access to several file paths, which are configured with a simple switch of environment variables between the staging VM and a container.

3.2 | Github and Webhook automation

The project's code was organized using a three-branch layout, with a production branch for active deployments, a branch for container development, and a branch for workflow script updates (source: https://github.com/ikewai/precip_pipeline_container). This allows for testing of each stage before introduction into the regular daily/monthly operations. Substantial changes to workflow are usually supported by appropriate modifications to the containers, such as dependency updates. Github's webhook system is used to trigger automatic container rebuilding on Docker Hub. Each branch of the repository is tracked and tagged separately, allowing for streamlined testing and automation where possible.

3.3 | Automated container management

The containers are organized in a two-layer structure: a base container and execution container. The base container consists of a debian image with R and all workflow dependencies pre-installed, and the aforementioned three branches are built on the base container (container: https://hub.docker.com/r/ikewai/rainfallbase, source: https://github.com/ikewai/rainfallbase_container/). The execution container leverages this base container by building on top of it with the addition of the workflow scripts. This layered containerization structure streamlines the build process and saves a tremendous amount of build time, as new testing and production execution containers can be built by DockerHub's infrastructure in a matter of a few minutes, rather than 3+ hours needed for dependency compilation of the base container.

3.4 | Tapis and Abaco

The containerized workflow runs using Abaco with its custom CRON-scheduling system, where parameters (environment variables) of the container can be adjusted ahead of each run, and different tasks can be handled at arbitrary intervals as needed by researchers (Figure 2).

The container arrangement is designed for three primary intervals: daily, monthly, and post-monthly. The daily operations handle aggregation of data from various rainfall sources, and processing them for daily mapping and use by later workflows. The monthly operations handle creates detailed monthly maps. The post-monthly operations handle gap-filling (using models to fill in data where certain sensors may have been offline) and QA/QC to detect and resolve anomalous values reported by sensor stations. Each of these three workflows is run on a single container. Using Abaco's message system, the container is passed all of the required environment variables related to file paths and workflow type in a JSON-formatted string, and the container's entry script uses the variables to determine which workflow scripts to run. We note that the daily, monthly, and post-monthly workflows could be organized as three separate containers corresponding to a unique actor for each, but, since a large portion of the workflow codes are the same up to the creation of mapping products, it is simpler to manage a single actor in the manner described over increasing the complexity in source code, container, and actor management.

3.5 | Workflow output, ingestion, and annotation

The mapping workflow[25] outputs a set of geoTIFF files defining the gridded precipitation data, a metadata file containing information about the data interpolation procedure, and comma-separated value (CSV) files representing the individual rainfall station values and metadata. Four types

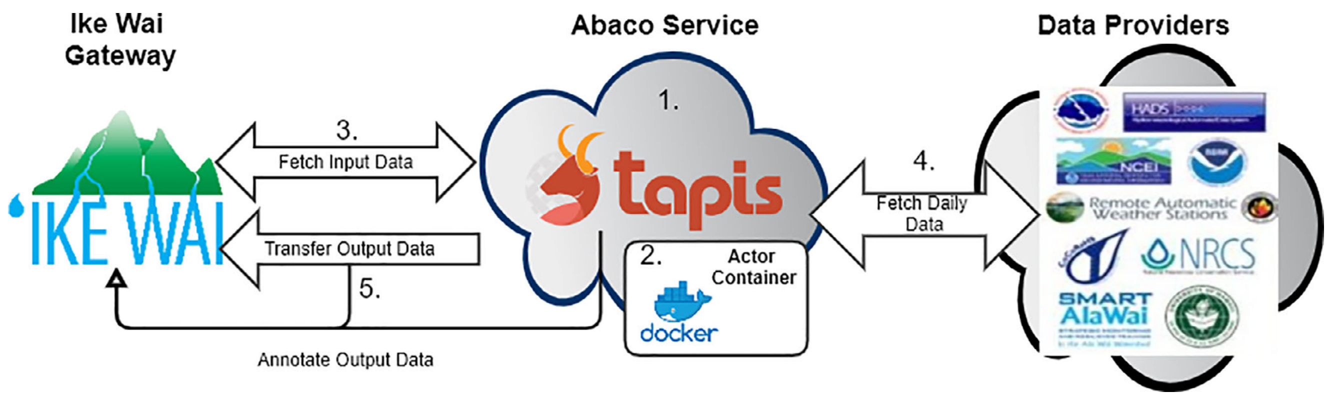

FIGURE 2 Serverless workflow. (1) Abaco's cron-scheduler executes the container and passes it a JSON-formatted message containing environment variables for file paths. (2) The container starts up and the container's internal control scripts parse the message, setting environment and input variables. (3) The control scripts use the Tapis Files API to transfer necessary metadata and monthly data files from the 'Ike Wai Gateway. (4) The domain portion of the workflow executes to collect and process precipitation data. (5) Workflow output files and execution logs are uploaded using the Tapis Files API to the 'Ike Wai Gateway

of geoTIFF files are produced: the gridded map of interpolated precipitation values, a standard error map, an anomaly map, and a leave one out cross-validation (LOOCV) error map. The latter three of these files provide error and probabilistic likelihood metrics for the values of the gridded map values yielded by this workflow. These files are created for each county as well as for the whole state. The set of produced files are stored and made available through the portal frontend web application.

The primary data files—the geoTIFF gridded data map and the rainfall station CSVs—are converted into JavaScript Object Notation (JSON) objects and ingested into a MongoDB database via the Tapis Metadata service. MongoDB is a no-SQL database that is based on JSON. The Tapis Metadata service provides an API for interacting with a MongoDB backend, providing the ability to create records with unique identifiers and methods for querying the database for specific records via REST requests.

The rainfall station data CSV files are formatted as a table of the rainfall station values by date and unique station identifier (SKN). Each value is parsed into a separate JSON document containing this information. The Tapis metadata service requires documents to include a name and value field and assigns each document a UUID. The station value documents were named "hcdp_station_value" and the value field was populated with information for identifying the dataset the value belonged to, the date of the measurement, the metadata associated with the station, and the value of the station. The metadata field includes a reference to a master metadata document for the full set of sensor stations, and the station identifier which references the station's entry in the metadata document (Figure 3A).

The statewide geoTIFF files are similarly converted into documents with the name "hcdp_raster_value." Each geoTIFF is converted into a single document. Since a large number of the grid cells in these maps fall in the ocean and have no associated value, the data are compressed by converting it from a sparse, continuous value grid to a mapping of grid indices to values; by doing this, ocean cells can be removed. The value field of these documents is then populated with information to identify the dataset, the date associated with the map, a geospatial reference document, and the aforementioned value map (Figure 3B). Since all of rainfall maps are positioned similarly, a single geospatial reference document is generated. This document contains header information from the geoTIFF file indicating the grid's geospatial position, dimensions, and cell resolution.

The use of the Tapis Metadata API for data annotation within the 'Ike Wai gateway and therefore the portal, supports FAIR data principles guideline aiming to enhance the reusability of data and metadata.[26] The FAIR principles put specific emphasis on enhancing the ability of machines to automatically find and use the data, in addition to supporting its reuse by individuals—which is a major aim of the portal capabilities.

3.6 | Sustainability and provenance

The use of automated continuous integration to deploy containerized software into production workflows with Github and Dockerhub that automatically track provenance can make updating the domain portion of the workflow as simple as committing updates via git. Containerizing the workflow also allows it to be moved to other Tapis tenants or container supporting computational resource with minimal reconfiguration.

Furthermore, with all the provenance tracking and access to the various provenance information of the workflow (versioned source, versioned containers, actor execution logs, linked annotations) there is robust support for tracking and debugging issues in the container or source code that may arise and rolling-back and redeploying the execution container, which is a priority since the workflow needs to run everyday.

```
"name": "hcdp_station_value",
"value": {
  "metadata": {
    "ref": "hawaii_station_master"
    "station_id": "83"
  },
  "datatype": "rainfall",
  "period": "monthly",
  "date": "2019-12",
  "ext": {
    "fill": "partial"
  },
  "value": 345.43
}
```

(A)

```
"name": "hcdp_raster_value",
"value": {
  "georef": "hawaii_statewide",
  "datatype": "rainfall",
  "period": "monthly",
  "date": "2019-12",
  "ext": {},
  "value": {
    ...
    "21003": 154.2,
    "21004": 132.8,
    ...
  }
}
```

(B)

FIGURE 3 (A) Example JSON document schema for station values. (B) Example JSON document schema for gridded map products

4 | WEB PORTAL

To facilitate public dissemination of the produced data products a public facing web application was developed.[19] This application provides climatological resources to users and provides access to all of the data produced by the data aggregation and mapping workflow as well as the ability to navigate and visualize the data in an interactive environment. The visualization and data dissemination portion of the application was developed using Angular, a web-application framework based on Typescript, a typed superset of JavaScript. This Angular application is embedded in a Wordpress website containing additional resources such as a library of publications related to climate research and modeling efforts in Hawai'i and references to historical climate mapping efforts.

4.1 | Data visualization

The data visualization component of the portal allows users to navigate through the entire set of gridded and sensor station time-series data and displays results on an interactive map in the application (Figure 4). The application leverages the JSON documents ingested into the Tapis Metadata service discussed in Section 3.5 for identifying and retrieving data. Requests for the documents containing the data to be displayed by the application are submitted to the metadata service's API endpoint to efficiently retrieve the values. Since the documents containing the

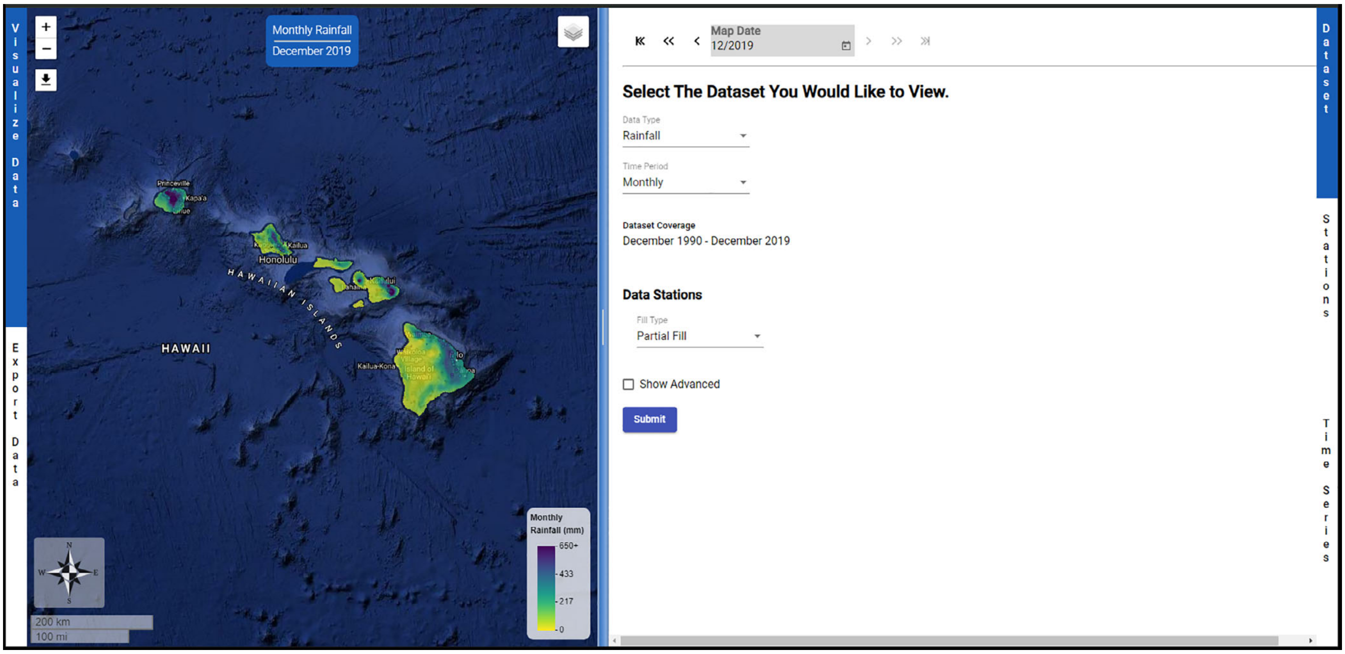

FIGURE 4 The portal visualization interface for selecting the dataset to be viewed and move between dates within the dataset

gridded values are large and take time to retrieve, the application caches several maps surrounding the date currently in use by the application. This improves wait times while stepping though data at sequential time intervals, as may be a common use pattern for observing trends over time. Additionally, these documents being generated and updated as part of the automated workflow, the application can identify the most up-to-date data via the date tag associated with the value documents and automatically update the range of coverage for near-real-time datasets.

The map visualization is generated using Leaflet, a JavaScript library for creating interactive maps. The gridded data are mapped to a color gradient and injected into a custom layer that renders the grid cells onto the map. The sensor stations are placed on the map as a set of circle markers that are sized according to their value and colored based on the same color scheme as the gridded data for an easy visual comparison to the values produced by the interpolation procedure. Multiple color schemes are provided by the application and custom color schemes can be uploaded to accommodate different user preferences and facilitate various types of visual analysis. The custom color scheme system uses an extensible markup language (XML) format established by the TACC SciVisColor tool,[27] and several of the color schemes established by this tool are preloaded into the initial set of available color schemes.

Basic information about the available sensor stations is displayed in a table and can be filtered on the various properties of the metadata associated with the stations to identify specific stations of interest. Clicking a station in the table or its respective map marker will bring up additional information about that station, displaying all of the metadata associated with it and displaying its value at the selected date (Figure 5). This will also trigger a request for all of the values associated with the station to generate a time series which is presented as a graph of the station's values over different periods of time. This graph can be used to view trends of a specific station, or to identify potential periods of interest (Figure 6).

Both the interactive map element and the time series graphs generated from station data provide controls to export as an image. This allows users to easily generate graphics from the data without necessarily needing to transfer the data into a third-party application to accomplish this.

4.2 | Data export

The portal provides access to all of the data files output by the data aggregation and mapping workflow. These files are made accessible through the Tapis files service, which provides API access to file storage systems. The interface allows users to generate custom export packages by first selecting the dataset of interest—this includes the climatological variable (e.g., rainfall) and the temporal granularity of the production (e.g., daily or monthly maps)—then selecting the set of files and date range of interest that are available for the specified dataset. The interface allows multiple file sets to be created for export at once (Figure 7).

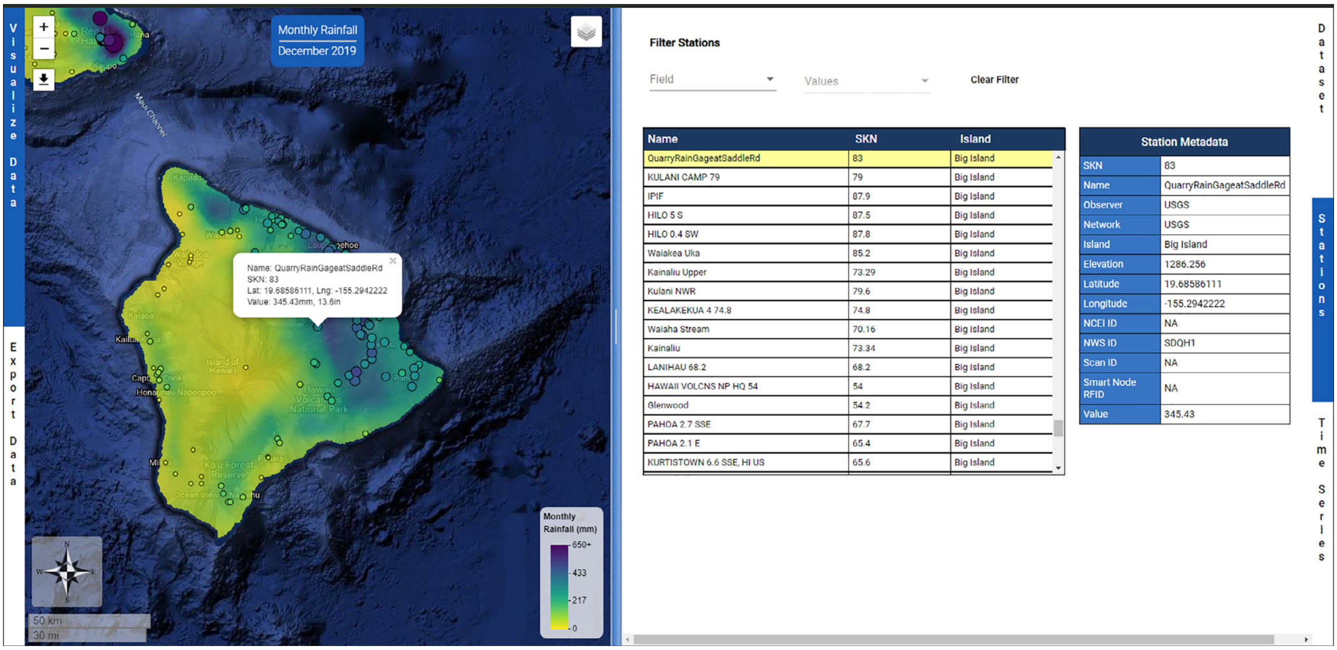

FIGURE 5 Sensor station informational view. Selecting a sensor station from the map or the data table displays detailed information about the selected station

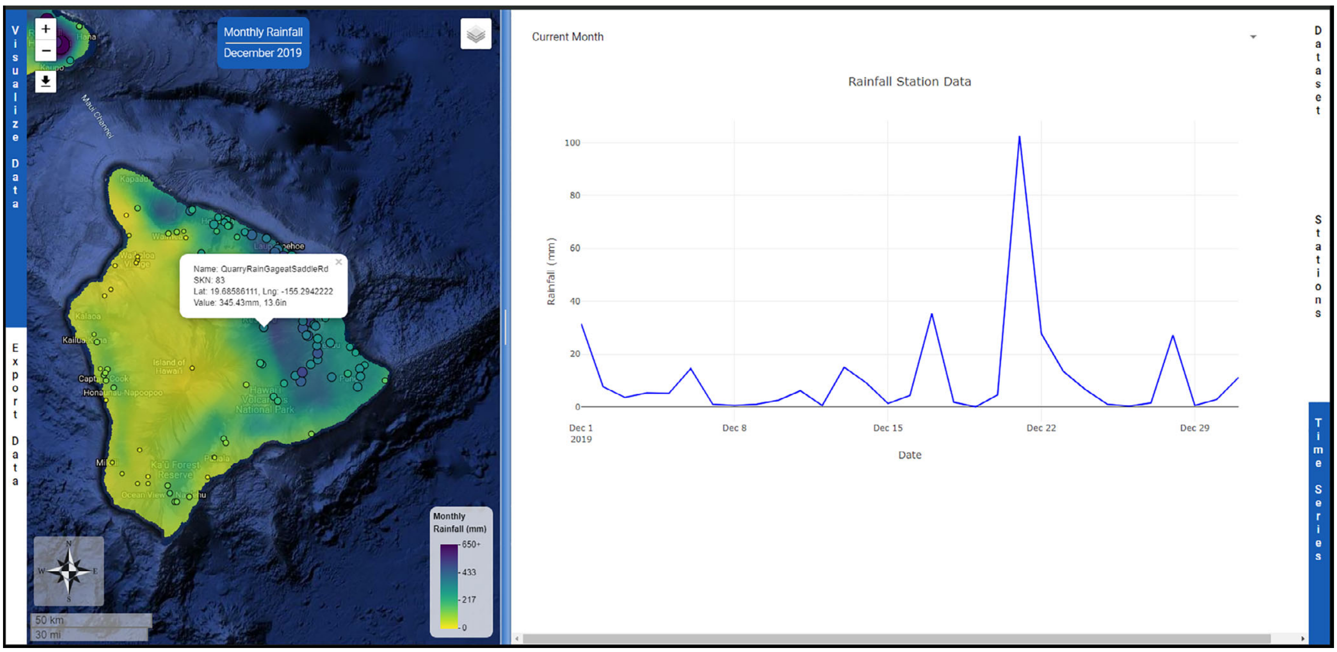

FIGURE 6 Time series graph for a sensor station for December 2019. Also allows users to view the time series for the selected year and the entire range of the dataset

Since the portal handles a large volume of data, the requested export package could potentially be very large. On export the size of the requested set of files will be evaluated and handled in one of two ways. If the user is only requesting a few files and the set is reasonably small, the application will directly generate a request to the Tapis files service and generate an on-demand download package in the application. This is infeasible for larger file sets, so if the set of files is too large the application will require the user to provide an email that can receive a download link after the package is generated. The application will then dispatch a request to create a Tapis Abaco container that will generate the download package. Abaco will be provided with the set of files and user email address and use these to retrieve the requested files, generate a download package, and email the download package link to the user on completion.

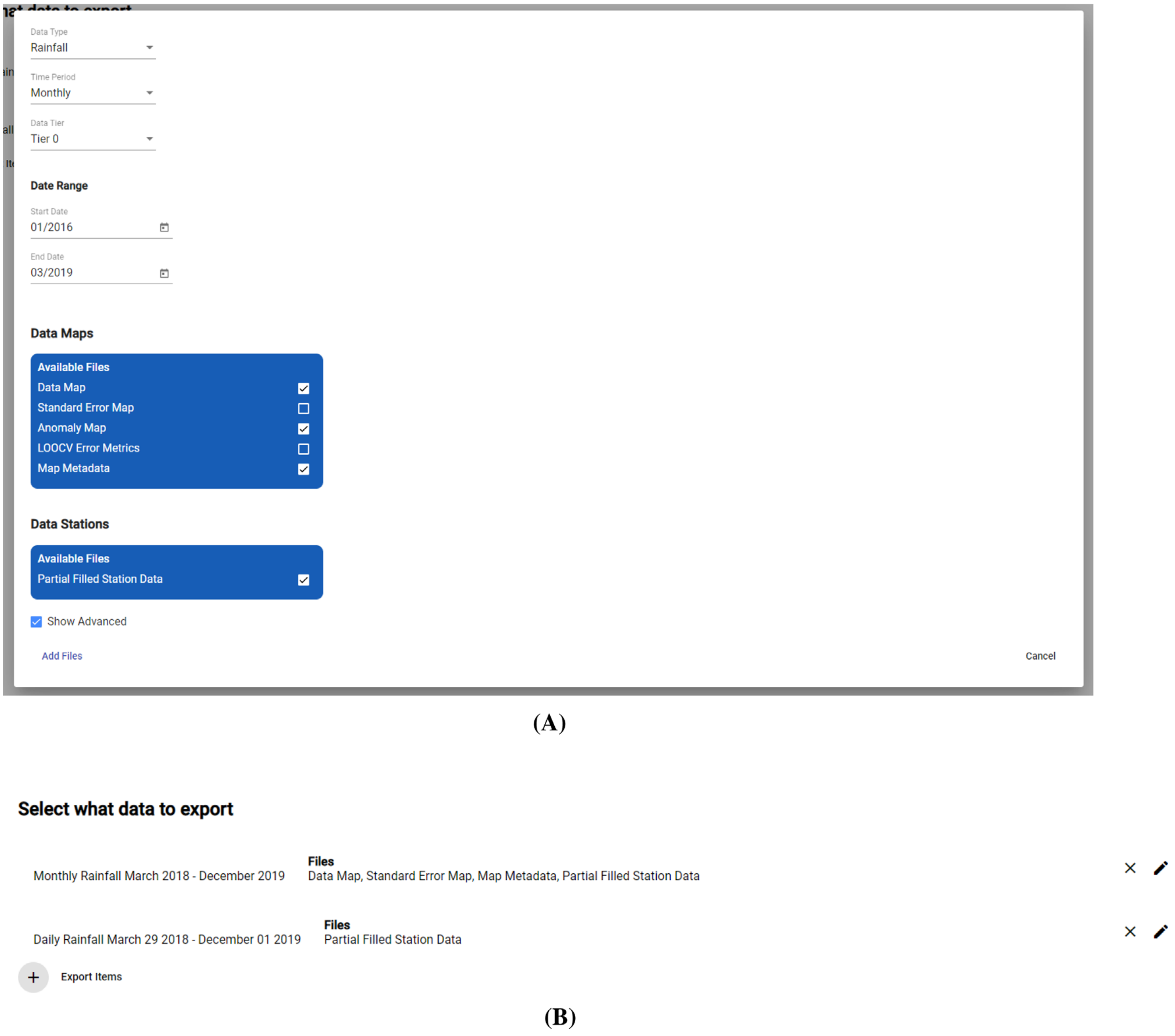

FIGURE 7 The portal export interface. (A) The form used to select the set of files to export from the portal. Selecting the dataset production details (e.g., data type, time period) populates lists of associated files and allows a subset of the date range the dataset covers to be selected. (B) Multiple sets of files can be added to the export package

5 | CONCLUSION

This work demonstrates the successful implementation of a "serverless" continuous workflow and dissemination platform for rainfall data in Hawai'i that utilizes distributed storage and computational resources while decreasing maintenance and upkeep of infrastructure for the researcher. Further, the levels of annotation and provenance allow for increased transparency and integrity of the data products, which is important for stakeholders and community researchers that utilize these datasets in models and additional studies. Providing simplified and single-source access to observational data as well as interpolated-gridded data maps in near-real-time should also serve to dramatically reduce the overhead of modeling and analyses that requires rainfall data. The workflow laid out also provides a good framework for providing similarly robust datasets for other climatological variables that can further streamline the process of data acquisition for research efforts requiring or building on historical or up-to-date climatological data for the state of Hawai'i.

6 | FUTURE WORK

This project will explore implementing new algorithms for automated daily precipitation map generation which will only require updates of the domain workflow source code in Github. Additionally, adding Abaco workers to a University of Hawai'i associate site deployment of Tapis will be explored to decrease file transfer latency from the 'Ike Wai gateway to the Abaco worker executing the workflows.

While some of the algorithmic details may vary, the general data aggregation, mapping, and automation workflows discussed in this article are extensible to climatological variables other than rainfall. The addition of other climate data types (temperature, evapotranspiration, etc.) will be enabled through similar Tapis powered workflows making the products available in near real-time as the current rainfall products are. The web interface has also been designed to handle any dataset providing geospatial point data from sensor stations or gridded data maps. Data produced by the workflow are tagged with information about the dataset they belong to, such as the data type they represent, to allow for easy differentiation as more data products are generated. This will allow for easy integration of additional climatological variables as they become available so the scope of the portal can be expanded without significant modifications.

To help further facilitate easy analysis of the produced data, a more robust suite of analysis tools can also be added to the web portal. Currently only station time series are included; however, the ability to create "virtual" stations anywhere on the map will be added. This will produce time series of the values at the underlying grid cell. This would allow users to view trends through time of any location in the state based on the produced data maps rather than being limited to sensor station locations. Additionally, more advanced comparison tools will be added based on common data use cases. Further, development of a public python library will make programmatic data access as well as integration into interactive environments such as Jupyter notebooks simpler for customized analyses.

ACKNOWLEDGMENTS

This work was made possible by grant funding from the following: RII Track 1: 'Ike Wai Securing Hawai'i's Water Future NSF OIA #1557349, Project Tapis: Next Generation Software for Distributed Research NSF CSSI #1931439, #1931575, Collaborative Research: SS2-SSI: The Agave Platform: An Open Science-As-A-Service Cloud Platform for Reproducible Science NSF ACI #1450413. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in the Hawaii Climate Data Portal at https://www.hawaii.edu/climate-data-portal/data-portal/.

ORCID

Jared McLean https://orcid.org/0000-0002-6347-0734
Sean B. Cleveland https://orcid.org/0000-0002-7130-3434

REFERENCES

1. Stewart CA, Turner G, Vaughn M, et al. Jetstream. Proceedings of the 2015 XSEDE Conference on Scientific Advancements Enabled by Enhanced Cyberinfrastructure—XSEDE '15; 2015; ACM Press.
2. Cleveland SB. The 'Ike Wai gateway—a science gateway for the water future of Hawai'i". Proceedings of Science Gateways; September 2018; Science Gateways Community Institute, Austin TX.
3. Dahan M, Towns J, Cockerill T, et al. XSEDE: accelerating scientific discovery. *Comput Sci Eng*. 2014;16(5):62-74. doi:10.1109/MCSE.2014.80
4. Wilkins-Diehr N, Crawford TD. The science gateways community institute and the molecular sciences software institute. *Comput Sci Eng*. 2018;20(5):26-38. doi:10.1109/MCSE.2018.05329813
5. Gesing S, Lawrence K, Dahan M, Pierce ME, Wilkins-Diehr N, Zentner M. Science gateways: sustainability via on-campus teams. *Future Gener Comput Syst*. 2019;94:97-102. doi:10.1016/j.future.2018.09.067
6. Gesing S, Wilkins-Diehr N, Dahan M, et al. Science gateways: the long road to the birth of an institute. Proceedings of the 50th Hawaii International Conference on System Sciences; 2017.
7. Kacsuk P, Farkas Z, Kozlovszky M, et al. WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities. *J Grid Comput*. 2012;10(4):601-630. doi:10.1007/s10723‐012‐9240‐5
8. Mclennan M, Kennell R. HUBzero: a platform for dissemination and collaboration in computational science and engineering. *Comput Sci Eng*. 2010;12(2):48-53. doi:10.1109/MCSE.2010.41
9. Pierce M, Marru S, Gunathilake L, et al. Apache airavata: design and directions of a science gateway framework. *Concurr Comput Pract Exper*. 2015;27(16):4282-4291. doi:10.1109/IWSG.2014.15
10. Marru S, Gunathilake L, Herath C, et al. Apache airavata: a framework for distributed applications and computational workflows. Proceedings of the 2011 ACM Workshop on Gateway Computing Environments; 2011:21-28; ACM, New York, NY.
11. Afgan E, Goecks J, Baker D, Coraor N, Nekrutenko A, Taylor J. Galaxy: a gateway to tools in e-science. *Guide e-Science*. 2011;145-177. doi:10.1007/978‐0‐85729‐439‐5&uscore;6
12. Ananthakrishnan R, Chard K, Foster I, Tuecke S. Globus platform-as-a-service for collaborative science applications. *Concurr Comput Pract Exp*. 2014;27(2):290-305. doi:10.1002/cpe.3262
13. Tarboton DG, Horsburgh JS, Maidment DR, et al. Development of a community hydrologic information system. Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation; 2009:988-994.

14. Tarboton D, Idaszak R, Horsburgh J, et al. HydroShare: advancing collaboration through hydrologic data and model sharing. Proceedings of the 7th International Congress on Environmental Modelling and Software, International Environmental Modelling and Software Society (iEMSs); 2014; San Diego, CA.
15. Mason SJ, Cleveland SB, Llovet P, Izurieta C, Poole GC. A centralized tool for managing, archiving, and serving point-in-time data in ecological research laboratories. *Environ Model Softw*. 2014;51:59-69.
16. Stubbs J, Cardone R, Packard M, et al. Tapis: an API platform for reproducible, distributed computational research. Proceedings of the Future of Information and Communication Conference; 2020.
17. Merchant N, Lyons E, Goff S, et al. The iPlant collaborative: cyberinfrastructure for enabling data to discovery for the life sciences. *PLoS Biol*. 2016;14(1). doi:10.1371/journal.pbio.1002342
18. Goff SA, Vaughn M, McKay S, et al. The iPlant collaborative: cyberinfrastructure for plant biology. *Front Plant Sci*. 2011;2(34). doi:10.3389/fpls.2011.00034
19. McLean JH, Cleveland SB, Lucas M, et al. The Hawai'i rainfall analysis and mapping application (HI-RAMA): decision support and data visualization for statewide rainfall dat. *Pract Exper Adv Res Comput*. 2020;239-245.
20. Longman RJ, Giambelluca TW, Nullet MA, et al. Compilation of climate data from heterogeneous networks across the Hawaiian Islands. *Sci Data*. 2018;5:180012. doi:10.1038/sdata.2018.12
21. National centers for environmental information. Accessed February 17, 2020. https://www.ncei.noaa.gov/
22. Hydrometeorological automated data system. Accessed February 17, 2020. https://hads.ncep.noaa.gov/
23. Soil climate analysis network. Accessed February 17, 2020. https://www.wcc.nrcs.usda.gov/scan/
24. Longman RJ, Newman AJ, Giambelluca TW, Lucas M. Characterizing the uncertainty and assessing the value of gap-filled daily rainfall data in Hawaii. *J Appl Meteorol Climatol*. 2020;59(7):1261-1276. doi:10.1175/jamc‐d‐20‐0007.1
25. Lucas MP, Longman RJ, Giambelluca TW, et al. Optimizing automated kriging to improve spatial interpolation of monthly rainfall over complex terrain. *Rev J Hydrometeorol*. 2020;12(4):2381-2409.
26. Cleveland SB, Geis J, Jacobs Cleveland GA. Scientific data annotation and dissemination: using the 'Ike Wai gateway to manage research data. *Practice and Experience in Advanced Research Computing*. Association for Computing Machinery; 2020:188-197.
27. TACC. SciVisColor portal. Accessed April 26, 2021. https://sciviscolor.org/

How to cite this article: McLean J, Cleveland SB, Dodge M, et al. Building a portal for climate data—Mapping automation, visualization, and dissemination. *Concurrency Computat Pract Exper*. 2021;e6727. doi: 10.1002/cpe.6727