CyberGIS-Cloud: A unified middleware framework for cloud-based geospatial research and education

FURQAN BAIG, CyberGIS Center for Advanced Digital and Spatial Studies, UIUC, U.S.A
ALEXANDER MICHELS, CyberGIS Center for Advanced Digital and Spatial Studies, UIUC, U.S.A
XIMO ZIAO, CyberGIS Center for Advanced Digital and Spatial Studies, UIUC, U.S.A
SU YEON HAN, CyberGIS Center for Advanced Digital and Spatial Studies, UIUC, U.S.A
ANAND PADMANABHAN, CyberGIS Center for Advanced Digital and Spatial Studies, UIUC, U.S.A
ZHIYU LI, CyberGIS Center for Advanced Digital and Spatial Studies, UIUC, U.S.A
SHAOWEN WANG, CyberGIS Center for Advanced Digital and Spatial Studies, UIUC, U.S.A

Interest in cloud-based cyberinfrastructure continues to grow within the geospatial community to tackle contemporary big data challenges. Distributed computing frameworks, deployed over the cloud, provide scalable and low-maintenance solutions to accelerate geospatial research and education. However, for scientists and researchers, the usage of such resources is highly constrained by the steep curve for learning diverse sets of platform-specific tools and APIs. This paper presents CyberGIS-Cloud as a unified middleware to streamline the execution of distributed geospatial workflows over multiple cloud backends with easy-to-use interfaces. CyberGIS-Cloud employs bringing computation-to-data model by abstracting and automating job execution over distributed resources hosted in the cloud environment where the data resides. We present details of CyberGIS-Cloud with support for popular distributed computing frameworks backed by research-oriented JetStream Cloud and commercial Google Cloud Platform.

CCS Concepts: • **Computer systems organization → Cloud computing**; • **Information systems → Geographic information systems**.

Additional Key Words and Phrases: cloud-computing, gis, scientific-gateway, middleware

ACM Reference Format:
Furqan Baig, Alexander Michels, Ximo Ziao, Su Yeon Han, Anand Padmanabhan, Zhiyu Li, and Shaowen Wang. 2022. CyberGIS-Cloud: A unified middleware framework for cloud-based geospatial research and education. In . ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The use of computational and storage resources in cloud is becoming increasingly popular in academia and industry alike. This shift from on-premise resource usage is due to a number of factors, including but not limited to, low setup and maintenance cost, pay-per-use model, high accessibility, better collaboration, and built-in security & privacy guarantees. While the trend is more noticeable in the private sector due to competition and monetary stakes, academia is actively catching up. Initiatives such as NSF CloudBank, European Open Science Cloud, and National Research Cloud are paving the way to providing cutting-edge computing power to academic researchers.

PEARC '22, July 10–14, 2022, Boston, MA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Following the trend, the geospatial community has also started to adopt cloud resources from two major perspectives; storage and computation. Recent advancements in mobile phones, Internet of Things (IoT), and sensory measurement technologies have contributed to generating multi-dimensional geospatial data at an unprecedented rate and scale. To collaboratively work with and store such data, cloud is an optimal option to offload setup, maintenance, and technical expertise for nominal costs. Similarly, for geospatial problem-solving at scale cyberGIS (i.e. geographic information science and systems based on advanced cyberinfrastructure (CI)) has become a key tool. However, despite increasing availability, effective usage of cloud resources is constrained by required technical expertise and a diverse set of platform-specific tools and APIs. Domain experts, including geospatial researchers and scientists, are among the most disadvantageous sub-group since they have to spend extra time learning and familiarizing themselves with different technologies in order to perform identical tasks in different cloud environments. While there exist studies [13, 14] exploring the potential of geospatial cloud computing and bringing HPC workloads to cloud [9, 10], limited work have focused on ease of access and availability of cloud resources to accelerate distributed workloads for geospatial domain experts.

In this study, we present CyberGIS-Cloud; a user-centered easy-to-use middleware for executing distributed geospatial big data analysis at scale. CyberGIS-Cloud abstracts away the details of accessing and utilizing multiple cloud backends and presents a unified interface to the end-users. The middleware supports popular geospatial libraries and employs a multi-tiered architecture to streamline different stages of distributed workflows on the user-selected cloud backend. As a proof of concept, we've automated executions of geospatial workflows utilizing Apache Hadoop, Apache Spark, and Dask frameworks deployed on distributed on-demand clusters in XSEDE JetStream Cloud and Google Cloud Platform environments.

2 ARCHITECTURE

CyberGIS-Cloud is part of an ecosystem [9, 15] mainly designed for enabling cutting edge data and compute-intensive geospatial analytics at scale. CyberGISX [15] provides easy-to-use Jupyter Notebook interfaces to perform geospatial analytics. CyberGIS-Compute [9] augments the capabilities of CyberGISX to take advantage of high-performance computing systems. CyberGIS-Cloud further extends the capabilities of CyberGIS-Compute and allows data scientists, researchers, and domain experts to scale geospatial workflows over cloud computing resources with the least amount of effort.

Figure 1 illustrates the architectural components of CyberGIS-Cloud. These components can broadly be categorized into three logical layers; the interface, middleware, and cloud resources. The interface layer provides a unified front end for the user to access backend cloud resources. This can be achieved either through python SDK or Jupyter notebook via a web browser.

The middleware comprises intermediary services between the user front end and the backend resources. Authentication service generates, validates, and authenticates users' credentials for interaction with cloud resources. In addition, the middleware hosts RESTful services to interactively register, submit and monitor distributed jobs on cloud backends. Instead of reinventing the wheel, we make use of open source services such as Apache Livy, sparkmagic, and DataprocSpawner. Data movement costs can quickly overwhelm any system, especially in the context of big data. To minimize data movement, for CyberGIS-Cloud, we assume data is already accessible in the cloud environment where the user specifies to execute the workflow. However, for all practical purposes, the middleware optionally provides a data management service to move data to/from cloud storage.

The middleware behaves as an abstraction layer to multiple backend cloud environments. Different cloud providers have different (and sometimes competing) capabilities and APIs. For instance, Google Cloud, Amazon AWS, and Microsoft Azure, well-known cloud providers, have their own set

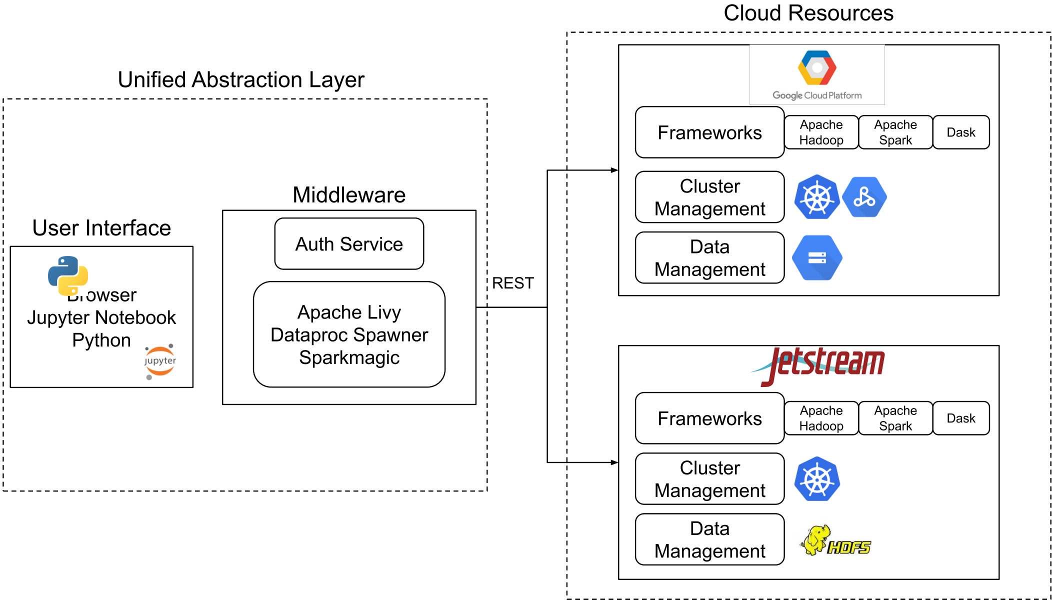

Fig. 1. CyberGIS-Cloud Architecture with unified abstraction layer for multiple cloud backends

of tools i.e. Cloud SDK, Amazon CLI, and Azure CLI respectively, for interacting with their services. Similarly, google file system, S3, and Azure Storage are storage mechanisms to manage data on the cloud having their own strengths and features. Learning to utilize multiple platform-specific utilities to perform the same set of tasks in different cloud environments can prove to be prohibitive, especially for domain experts who want to focus on their areas of expertise. CyberGIS-Cloud defines two sets of abstract interfaces; data and cluster management, that needs to be implemented for integrating every cloud provider with the system. The implementation details of these interfaces are hidden from the user. At the same time, this layer separation makes the CyberGIS-Cloud system easily extensible to other cloud platforms in the future.

3 IMPLEMENTATION

We implemented CyberGIS-Cloud to accelerate geospatial big data analysis using distributed computing. Programming models such as Map-Reduce [4] and software packages built on top of such models e.g. Hadoop, Spark, etc. have become readily available to facilitate distributed computing. In the geospatial domain, extensions of these software packages [2, 3, 5, 8, 12] have been proposed and implemented both in academia as well as industry. Despite the ease of using programming interfaces provided by these packages, deploying, executing, and managing them on actual infrastructure is non-trivial and can prove to be a major barrier for developers and researchers alike. We aim to bridge this gap by implementing CyberGIS-Cloud for Google Cloud Platform (GCP) and XSEDE JetStream. While GCP provides publicly available commercial services, JetStream is more research-oriented and popular in scientific communities.

3.1 Distributed Cluster Management

Automating software and package deployment is an essential part of CyberGIS-Cloud. Distributed workflows are generally accelerated on a cluster of workers. These workers can be individual physical nodes, independent virtual machines, or containers deployed and managed over orchestration frameworks. Recent developments in this domain have produced systems and software suitable for a number of use cases.

3.1.1 Google Cloud. GCP provides multiple managed services to set up and manage cluster resources in the cloud environment. Dataproc is primarily designed for big data stack and has out-of-the-box support for Apache Hadoop and Spark. In addition, Dataproc provides a mature API for cluster creation, management, and deletion. Google Kubernetes Engine (GKE), on the other hand, is relatively new and provides native support for auto-scaling. It is more suitable for long-running jobs. For CyberGIS-Cloud, we implemented cluster management using Dataproc API due to its maturity, ease of use, and cost-efficiency.

3.1.2 JetStream. Since JetStream provides access to more bare-bones virtual machines, we decided to use Kubernetes to set up and manage the distributed cluster. Kubernetes [1] is a container orchestration framework for deploying and managing resources, especially in cloud environments. It follows a master-slave architecture building on the concept of pods. Masters are generally user-accessible, whereas slave workers are managed by the master. Unlike traditional resource managers like YARN, Mesos, etc. Kubernetes provide native container orchestration support. In addition to providing the capability to execute multiple distributed frameworks on the same cluster, this also allows easy scaling mechanisms which are essential to cloud computing.

3.2 Framework Support & Dependency Management

During the past decade, Map-Reduce-based frameworks have become a popular choice for scalable and cost-effective big data processing over a cluster of commodity machines. Iterative in-memory frameworks take distributed processing a step further and promise 10x-100x performance boosts. To cover the maximum range of applications over this paradigm, we provide support for Apache Hadoop [7], Apache Spark [16], and Dask [11] based geospatial workflows in CyberGIS-Cloud.

3.2.1 Google Cloud. As mentioned earlier, GCP Dataproc by default provides support for many popular big data frameworks including Hadoop and Spark. For Dask workflows, we used cluster initialization actions to set up Dask over the Dataproc cluster. Similarly, we used Dataproc initialization actions to install user-specified dependencies on distributed cluster nodes.

3.2.2 JetStream. Latest versions of Apache Spark provide support for Kubernetes operators. We use helm charts to deploy Hadoop and Dask workloads over Kubernetes cluster in the cloud. CyberGIS-Cloud launches every distributed workflow using a default docker container deployed over each Kubernetes pod. This default docker container comes pre-configured with a set of popular geospatial libraries and packages. However, to avoid unnecessarily bloating the default image, we provide options for users to specify their additional required dependencies. This can easily be done by simply including *Dockerfile* commands for general packages or providing a *requirements.txt* for python dependencies. In either case, the additional packages are installed on the Docker containers deployed over Kubernetes pods in the cluster.

3.3 Data Management

Most of the contemporary solutions aiming at bridging the gap between users and sophisticated technical infrastructure tend to consider data as second-class citizens. The focus generally is on providing access to computational frameworks and data is either copied to accessible storage systems or shared using unified systems such as globus [6]. Despite its applicability, this approach is not scalable and comes with inherent costs. Especially, in the context of big data, data copying or accessing it over the internet can prove to be prohibitive in terms of performance.

CyberGIS-Cloud proposes to bring computation to data instead of the usual other way around. If a user's data resides in a commercial cloud e.g. google cloud storage buckets, (s)he can grant appropriate permissions to access and process it in the CyberGIS-Cloud environment. Similarly, for research-oriented cloud systems e.g. JetStream, users can execute their distributed workflows by simply selecting JetStream as their execution environment.

3.4 Pricing

Research-oriented clouds such as XSEDE JetSteam generally do not have direct financial implications. On the other hand, cost and pricing are critical factors while working with commercial clouds. The decision about the entity responsible for the cost can make or break the system. CyberGIS-Cloud takes a hybrid approach. Resource utilization caps and fairness guarantees are enforced by the middleware. We believe that available cluster options are generalized enough to handle typical big data workflows.

Despite its free-of-cost availability, CyberGIS-Cloud has limits in terms of resources and funding. However, we envision CyberGIS-Cloud to operate beyond such limits. To this end, we plan to provide options to integrate user-provided cloud resources with CyberGIS-Cloud. While specifying a job in CyberGIS-Cloud, the user will have an option to provide details about their own cloud resources. The functionality of CyberGIS-Cloud will remain unchanged except for the fact that in such a case, the actual cluster resources will be provisioned on the user's cloud and the workflow will be executed on it. This particular feature is currently under development and will be one of the major features of our future work.

4 HUMAN MOBILITY ANALYSIS - A TYPICAL CYBERGIS-CLOUD WORKFLOW EXAMPLE

We studied the effectiveness of CyberGIS-Cloud by taking social media-based human mobility analysis as a representative use case. Past geospatial research has shown that coordinates of Twitter users' locations can be used to examine different trajectories of some users and their travel patterns within and between cities. These geotagged tweets can reveal spatiotemporal patterns of human movements that can especially be useful for studying evacuees' travel patterns during natural disasters and disease spread in pandemic situations such as covid-19.

To perform the analysis, we pulled 500,000 tweets from Twitter Streaming API on distributed HDFS (Hadoop File System) setup on a preconfigured cluster on JetStream Cloud and more than 10 million tweets on the google cloud storage bucket. We wrote *pyspark* code[1] for the distributed mobility analysis pipeline. Executing the pipeline on a local setup with 4 CPU cores and 8 GB of memory was able to process up to several thousands of tweets only. However, this was only sufficient for initial testing and debugging and could not handle actual larger datasets.

Using CyberGIS-Cloud, we were able to connect to a 3 node cluster deployed on JetStream cloud to perform the analysis on the 100,000 geotagged tweets. Using the same code and CyberGIS-Cloud environment, we were able to further scale our analysis over a much larger dataset in the Google Cloud environment. The major advantage of CyberGIS-Cloud is the ability to use the same distributed processing code, with minimal changes, and execute it on on-demand clusters deployed over multiple cloud backends via a unified user interface through the users' browser environment.

5 CONCLUSION AND FUTURE WORK

In this paper, we presented the design and implementation details of CyberGIS-Cloud; a middleware that provides easy-to-use interfaces to accelerate distributed geospatial workloads over cloud-based

[1] https://github.com/cybergis/CyberGIS-BigData/blob/main/twitter_mobility/code/tweet_processing.py

cyberinfrastructure resources. CyberGIS-Cloud allows users to execute Hadoop, Spark, and Dask based analytics on geospatial data stored in cloud storage. Depending on the cloud storage provider, CyberGIS-Cloud enables the user to perform distributed computation on a cluster deployed over an appropriate cloud backend i.e. JetStream Cloud or Google Cloud. To address data privacy, cost, and pricing concerns, in the future, we plan to explore options to integrate CyberGIS-Cloud with user-provided cloud resources to deploy and execute distributed workflows. Another important future aspect for CyberGIS-Cloud is to provide the ability for automated resource estimation and job scaling based on dataset size.

ACKNOWLEDGMENTS

The material is based upon work supported in part by the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) that is supported by the National Science Foundation (NSF) under award No. 2118329. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF. Our computational work used NSF XSEDE and Virtual ROGER. Virtual ROGER is a geospatial supercomputer supported by the CyberGIS Center for Advanced Digital and Spatial Studies and the School of Earth, Society and Environment at the University of Illinois Urbana-Champaign.

REFERENCES

[1] 2022. *Production-Grade Container Orchestration.* Retrieved March 19, 2022 from https://kubernetes.io

[2] Ablimit Aji, Fusheng Wang, Hoang Vo, Rubao Lee, Qiaoling Liu, Xiaodong Zhang, and Joel Saltz. 2013. Hadoop GIS: A High Performance Spatial Data Warehousing System over Mapreduce. *Proc. VLDB Endow.* 6, 11 (Aug. 2013).

[3] Furqan Baig, Hoang Vo, Tahsin Kurc, Joel Saltz, and Fusheng Wang. 2017. SparkGIS: Resource Aware Efficient In-Memory Spatial Query Processing. In *Proceedings of the 25th ACM SIGSPATIAL*. ACM.

[4] Sanjay Dean, Jeffrey & Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* (2008).

[5] Ahmed Eldawy. 2014. SpatialHadoop: Towards Flexible and Scalable Spatial Processing Using Mapreduce. In *Proceedings of the 2014 SIGMOD PhD Symposium (SIGMOD'14 PhD Symposium).* ACM, New York, NY, USA.

[6] Ian Foster. 2011. Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. *IEEE Internet Computing* 15, 3 (may 2011), 70–73. https://doi.org/10.1109/MIC.2011.64

[7] Apache Software Foundation. 2022. *Hadoop.* Retrieved March 19, 2022 from https://hadoop.apache.org

[8] Jinxuan Wu Jia Yu, Mohamed Sarwat. 2015. GeoSpark: A Cluster Computing Framework for Processing Large-Scale Spatial Data. In *Proceedings of ACM SIGSPATIAL 2015.*

[9] Anand Padmanabhan, Ximo Ziao, Rebecca C. Vandewalle, Furqan Baig, Alexander Michel, Zhiyu Li, and Shaowen Wang. 2021. CyberGIS-Compute for Enabling Computationally Intensive Geospatial Research. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on APIs and Libraries for Geospatial Data Science* (Beijing, China) *(SpatialAPI '21).* Association for Computing Machinery, New York, NY, USA, Article 3, 2 pages. https://doi.org/10.1145/3486189.3490017

[10] Yuxing Peng, Jonathan Skone, Callista Christ, and Hakizumwami Runesha. 2021. *Skyway: A Seamless Solution for Bursting Workloads from On-Premises HPC Clusters to Commercial Clouds.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3437359.3465607

[11] Matthew Rocklin. 2015. Dask: Parallel computation with blocked algorithms and task scheduling. In *Proceedings of the 14th python in science conference.* Citeseer.

[12] Dong Xie, Feifei Li, Bin Yao, Gefei Li, Liang Zhou, and Minyi Guo. 2016. Simba: Efficient In-Memory Spatial Analytics. In *In Proceedings of 35th ACM SIGMOD International Conference on Management of Data (SIGMOD'16).*

[13] Chaowei Yang, Robert Raskin, Michael Goodchild, and Mark Gahegan. 2010. Geospatial Cyberinfrastructure: Past, present and future. *Computers, Environment and Urban Systems* (2010).

[14] Chaowei Yang, Manzhu Yu, Fei Hu, Yongyao Jiang, and Yun Li. 2017. Utilizing Cloud Computing to address big geospatial data challenges. *Computers, Environment and Urban Systems* (2017).

[15] Dandong Yin, Yan Liu, Anand Padmanabhan, Jeff Terstriep, Johnathan Rush, and Shaowen Wang. 2017. A CyberGIS-Jupyter Framework for Geospatial Analytics at Scale. In *Proceedings of the PEARC'17.* ACM.

[16] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. 2016. Apache Spark: A Unified Engine for Big Data Processing. *Commun. ACM* (oct 2016).