# Streamlined HPC Environments with CVMFS and CyberGIS-Compute

Alexander Michels
CyberGIS Center for Advanced Digital and Spatial Studies
University of Illinois at Urbana-Champaign
Urbana, IL, USA
michels9@illinois.edu

Mit Kotak
Center for Computational Science and Engineering
Massachusetts Institute of Technology
Cambridge, MA, USA
mkotak@mit.edu

Anand Padmanabhan
CyberGIS Center for Advanced Digital and Spatial Studies
University of Illinois at Urbana-Champaign
Urbana, IL, USA
apadmana@illinois.edu

Shaowen Wang
CyberGIS Center for Advanced Digital and Spatial Studies
University of Illinois at Urbana-Champaign
Urbana, IL, USA
shaowen@illinois.edu

***Abstract*— High-Performance Computing (HPC) resources provide the potential for complex, large-scale modeling and analysis, fueling scientific progress over the last few decades, but these advances are not equally distributed across disciplines. Those in computational disciplines are often trained to have the necessary technical skills to utilize HPC (e.g. familiarity with the terminal), but many disciplines face technical hurdles when trying to apply HPC resources to their work. This unequal familiarity with HPC is increasingly a problem as cross-discipline teams work to tackle critical interdisciplinary issues like climate change and sustainability. CyberGISCompute is middle-ware designed to democratize HPC services with the goal of empowering domain scientists, but a key challenge facing model developers on CyberGIS-Compute is creating a containerized software environment for their models. In this paper, we discuss our work to integrate the Cern Virtual Machine File System (CVMFS) into CyberGIS-Compute to provide consistent software environments across science gateways and HPC resources.**

***Keywords—CyberGIS, CyberGIS-Compute, CVMFS***

## I. Introduction

We are faced with complex, large-scale, and unprecedented sustainability challenges that require immense amounts of computation to adequately analyze, model, and simulate. High-Performance Computing (HPC) resources are capable of helping us tackle these challenges, but domain experts working on these sustainability challenges often lack the technical training required to access and fully utilize the power that HPC resources provide. Lowering the technical barriers to HPC is critical to bring together domain experts working on our most pressing problems and HPC resources capable of handling the immense complexity of these problems.

CyberGIS-Compute is a middle-ware designed to simplify and streamline utilizing HPC resources, with a focus on geospatial workflows like those in geography, hydrology, and remote sensing [1; 2]. CyberGIS-Compute is able to execute models from Github on HPC resources and relies on containerization technology to provide a consistent execution environment across HPC centers [3]. Despite its benefits, our reliance on containers introduces a significant technical hurdle by requiring that those developing models for CyberGIS-Compute learn the basics of Singularity. In this paper, we discuss our work integrating the Cern Virtual Machine File System (CVMFS) [4] into CyberGIS-Compute to streamline HPC environments for users.

## II. Background

### A. CyberGIS-Compute

CyberGIS [5; 6] has the potential to help analyze and model the complex systems driving today's sustainability challenges just as cyberGIS has fueled research in fields like economics [7], evacuation [8], and public health [9; 10]. CyberGIS-Compute was designed to help realize this potential by eliminating the technical hurdles faced by domain experts working on our most pressing sustainability challenges. CyberGIS-Compute has helped domain experts in hydrology [11], public health [12], and remote sensing [13] by providing a user-friendly graphical user interface (GUI), managing operations with cyberinfrastructure batch systems, and streamlining the process for model development.

CyberGIS-Compute consists of two main components: a Software Development Kit (SDK) that provides a Graphical User Interface (GUI) for end-users and a Core server that accepts user requests and executes jobs on HPC resources [1; 2]. We found that Python and Jupyter notebooks [14] were familiar to our target audience, so the SDK is written in Python with the GUI built on Jupyter widgets. The Core server is written in Typescript and its architecture is illustrated in Figure 1. When a user submits a job from the GUI, that request is sent to our Core server which authenticates the user using a Jupyter token, creates the necessary script to execute the job on the requested HPC resource, submits the job, monitors it, and then returns any results to the user using Globus [15].

Environments on HPC centers differ drastically, so CyberGIS-Compute uses Singularity containers [3] to ensure that jobs can run on a variety of HPC resources we support. Despite its benefits, containers are the most frequently cited pain point among model developers because those unfamiliar with HPC are usually not familiar with containerization. We have created Singularity images that closely mimic the software

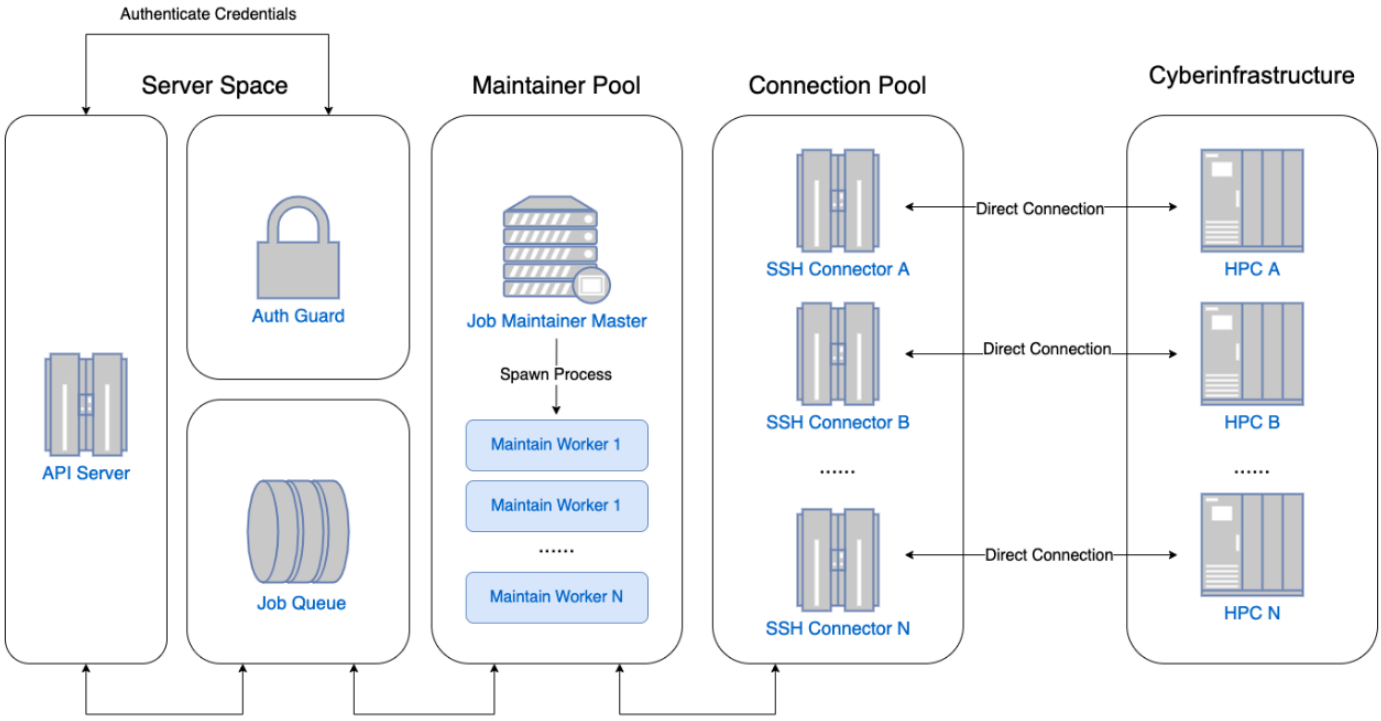


Fig. 1. The architecture of the CyberGIS-Compute Core server. Requests are received by the API server, users are authenticated, and then jobs enter a job queue. Our maintainer pool submits the job to an HPC center through the connection pool and monitors it until it completes.

environment provided by kernels on the CyberGISX [16; 17] and CyberGIS-Jupyter for Water (CJW) [18], but recreating software environments is imperfect. Our users want the exact same software environment on our science gateways and HPC.

### *B. Cern Virtual Machine File System*

The Cern Virtual Machine File System (CVMFS) is a distributed file system designed for efficiently delivering software on HPC centers [4]. CVMFS achieves efficiency through Content-Addressable Storage that provides content de-duplication and enables fast data integrity verification. Rather than a single server providing access to the software, CVMFS relies on a content delivery network for scalability and fault tolerance. Software is written to a Stratum 0 server, Stratum 1 servers are geographically distributed read-only copies of Stratum 0, and a network of proxies provide end-users with access. The CyberGISX and CJW science gateways have recently transitioned to utilizing CVMFS for their software as shown in Figure 2.

## III. Methods

We have developed a Connector[1] for CyberGIS-Compute that utilizes CVMFS to provide consistent software environments for CyberGIS-Compute jobs across the HPC resources we support. This could eliminate the need for model developers to create Singularity containers and allow models to rely on the exact same software utilized on the CyberGISX and CJW gateways, drastically lowering the technical barriers to

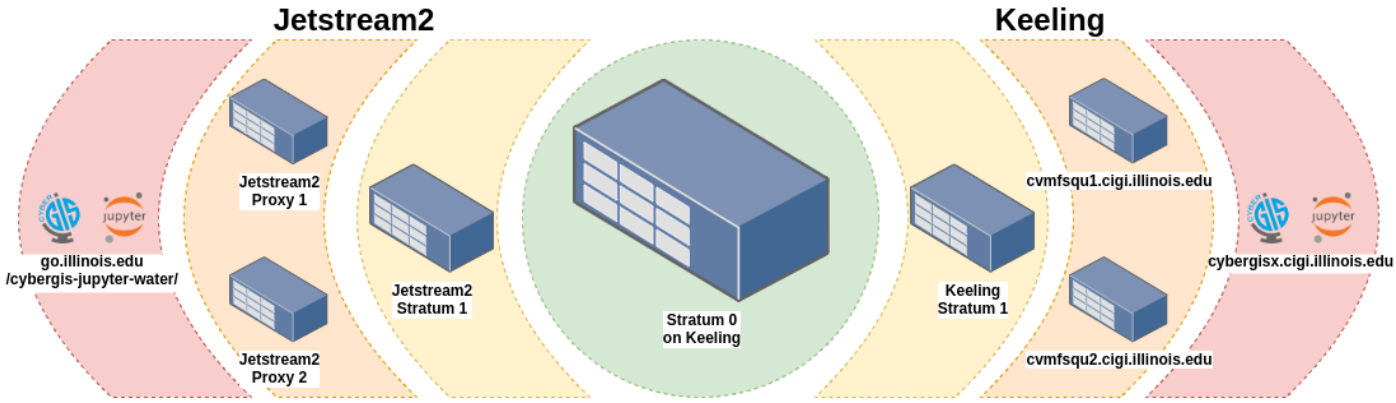


Fig. 2. The CVMFS content delivery network used by the cybergis.illinois.edu CVMFS repository.

[1] https://github.com/cybergis/cybergis-compute-core/pull/64

model contribution. Each job still executes in a container, but the same container is used for all jobs and rather than specifying a Singularity container for their model, model developers just specify a kernel on either of our science gateways that they would like to run the model with.

Our CVMFS Connector relies on singcvmfs[2] to ensure we can provide the CVMFS software on all HPC centers without requiring that administrators add our CVMFS repository. The singcvmfs tool is a drop-in replacement for the Singularity command that allows us to launch a Singularity container with CVMFS repositories bind-mounted in them. Our next step was to streamline the process of recreating the software environments from CyberGISX and CJW on HPC. singcvmfs gives us access to our software and a simple container with Lmod [19] which allows users to load the software, but users should be able to specify the kernel they want rather than loading all of the software themselves. To achieve this, we configured our CVMFS Connector to generate a kernel-init.sh script that recreates the software environment used by the kernel specified. The script loads modules, alters the path, and sets environment variables before the model executes.

## IV. Results

We have successfully converted and executed some existing models with our new CVMFS Connector. One such example is the pysal-acess model[3] which utilizes the Pysal access Python package [20] to calculate spatial accessibility to doctors in Chicago, IL, USA using a variety of methods. Access to healthcare is key to the U.N. Sustainable Development Goal 3 of ensuring healthy lives and promoting wellbeing for all at all ages [21]. Converting the model to use our CVMFS Connector[4] was as simple as specifying that we want to use our new connector and then specifying the CyberGISX Python3 0.9.0 kernel in the container field. Figure 3 shows a side-by-side comparison of the manifests for the CVMFS-enabled model (left) and the original model (right).

## V. Concluding Discussion

This paper discusses our work to integrate CVMFS into CyberGIS-Compute to streamline model development and further lower the technical barriers faced by domain experts trying to utilize HPC resources. We have demonstrated the ability to use our CVMFS Connector to execute an existing CyberGIS-Compute model with minimal changes to the model. This would allow us to remove the largest remaining technical barrier to those developing models for CyberGIS-Compute and

```
{
    "name": "Pysal Access Example",
    "description": "Calculates spatial accessibil
ity using a variety of metrics using the Pysal ac
cess package: https://github.com/pysal/access",
    "estimated_runtime": "3-6 minutes",
    "container": "cybergisx/python3-0.9.0",
    "connector": "SingCVMFSConnector",
    "execution_stage": "python ChicagoAccess.py",
    "slurm_input_rules": {
          "time": {
               "max": 30,
               "min": 15,
               "default_value": 20,
               "step": 1,
               "unit": "Minutes"
           },
           "memory": {
               "max": 4,
               "min": 2,
               "default_value": 4,
               "step": 1,
               "unit": "GB"
           }
    },
    "require_upload_data": false,
    "supported_hpc": ["anvil_community", "expanse
_community", "keeling_community"],
    "default_hpc": "keeling_community"
}
```

```
{
    "name": "Pysal Access Example",
    "description": "Calculates spatial accessibil
ity using a variety of metrics using the Pysal ac
cess package: https://github.com/pysal/access",
    "estimated_runtime": "3-6 minutes",
    "container": "pysal-access",

    "execution_stage": "python ChicagoAccess.py",
    "slurm_input_rules": {
          "time": {
               "max": 30,
               "min": 15,
               "default_value": 20,
               "step": 1,
               "unit": "Minutes"
           },
           "memory": {
               "max": 4,
               "min": 2,
               "default_value": 4,
               "step": 1,
               "unit": "GB"
           }
    },
    "require_upload_data": false,
    "supported_hpc": ["anvil_community", "expanse
_community", "keeling_community"],
    "default_hpc": "keeling_community"
}
```

Figure 3. Comparison of the manifests for the CVMFS-enabled model (left) and the original model (right). Only two lines have changed: a new line specifies the SingCVMFSConnector and the container is changed from "pysal-access" to "cybergisx/python3-0.9.0"

[2] https://github.com/cvmfs/cvmfsexec

[3] Original Pysal Access model: https://github.com/cybergis/pysal-access-compute-example

[4] CVMFS-enabled Pysal Access model: https://github.com/cybergis/pysal-access-compute-example-cvmfs

provide the software environments on our science gateways across multiple HPC centers.

While we have been able to successfully convert a handful of models, there is more work to do to ensure that our solution is stable and reliable. We have found that the CVMFS Connector occasionally fails because the model attempts to access software that CVMFS has not yet received from the content delivery network. Additionally, the models we have tested so far have been Python-based and we need to extensively test if models in other languages or that utilize tools like MPI will work with our Connector. Additional work will also be required to determine how well our CVMFS solution works on commercial clouds [22]. Lastly, the CVMFS integration will allow us to duplicate environments on CyberGIS-Jupyter [16] and CJW [18], but the the I-GUIDE Platform [23] is still working to move to our CVMFS repository.

## ACKNOWLEDGMENT

This paper and associated materials are based in part upon work supported by the National Science Foundation under grant numbers: 2118329 and 2112356. The work also received support from the Taylor Geospatial Institute. Our computational experiments used Virtual ROGER that 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] A. Padmanabhan, Z. Xiao, R. Vandewalle, A. Michels, and S. Wang, "Enabling computationally intensive geospatial research on CyberGIS-Jupyter with CyberGIS-Compute," Oct. 2021.

[2] A. Padmanabhan, X. Ziao, R. C. Vandewalle, F. Baig, A. Michels, Z. Li, and S. Wang, "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, SpatialAPI '21, (New York, NY, USA), pp. 1–2, Association for Computing Machinery, Nov. 2021.

[3] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," PLOS ONE, vol. 12, p. e0177459, May 2017.

[4] J. Blomer, P. Buncic, and T. Fuhrmann, "CernVM-FS: Delivering scientific software to globally distributed computing resources," in Proceedings of the First International Workshop on Network-aware Data Management - NDM '11, (Seattle, Washington, USA), p. 49, ACM Press, 2011.

[5] S. Wang, "A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis," Annals of the Association of American Geographers, vol. 100, pp. 535–557, June 2010.

[6] S. Wang, L. Anselin, B. Bhaduri, C. Crosby, M. F. Goodchild, Y. Liu, and T. L. Nyerges, "CyberGIS software: A synthetic review and integration roadmap," International Journal of Geographical Information Science, vol. 27, pp. 2122–2145, Nov. 2013.

[7] A. Michels, J.-Y. Kang, and S. Wang, "An Exploration of the Effect of Buyer Preference and Market Composition on the Rent Gradient Using the ALMA Framework," Proceedings of the 3rd ACM SIGSPATIAL International Workshop on GeoSpatial Simulation, pp. 48–51, 2020.

[8] R. Vandewalle, J.-Y. Kang, D. Yin, and S. Wang, "Integrating CyberGIS-Jupyter and spatial agent-based modelling to evaluate emergency evacuation time," in Proceedings of the 2nd ACM SIGSPATIAL International Workshop on GeoSpatial Simulation, pp. 28–31, 2019.

[9] J.-Y. Kang, B. F. Farkhad, M.-p. S. Chan, A. Michels, D. Albarracin, and S. Wang, "Spatial accessibility to HIV testing, treatment, and prevention services in Illinois and Chicago, USA," PLOS ONE, vol. 17, p. e0270404, July 2022.

[10] A. Michels, J.-Y. Kang, and S. Wang, "Particle Swarm Optimization for Calibration in Spatially Explicit Agent-Based Modeling," Journal of Artificial Societies and Social Simulation, vol. 25, no. 2, p. 8, 2022.

[11] I. Maghami, A. Van Beusekom, L. Hay, Z. Li, A. Bennett, Y. Choi, B. Nijssen, S. Wang, D. Tarboton, and J. L. Goodall, "Building cyberinfrastructure for the reuse and reproducibility of complex hydrologic modeling studies," Environmental Modelling & Software, vol. 164, p. 105689, June 2023.

[12] J.-Y. Kang, A. Michels, F. Lyu, S. Wang, N. Agbodo, V. L. Freeman, and S. Wang, "Rapidly measuring spatial accessibility of COVID-19 healthcare resources: A case study of Illinois, USA," International journal of health geographics, vol. 19, no. 1, pp. 1–17, 2020.

[13] F. Lyu, Z. Yang, Z. Xiao, C. Diao, J. Park, and S. Wang, "CyberGIS for Scalable Remote Sensing Data Fusion," in Practice and Experience in Advanced Research Computing, PEARC '22, (New York, NY, USA), pp. 1–4, Association for Computing Machinery, July 2022.

[14] T. Kluyver, B. Ragan-Kelley, F. P´erez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter Notebooks – a publishing format for reproducible computational workflows," in Positioning and Power in Academic Publishing: Players, Agents and Agendas (F. Loizides and B. Schmidt, eds.), pp. 87–90, IOS Press, 2016.

[15] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," The International Journal of Supercomputer Applications and High Performance Computing, vol. 11, pp. 115–128, June 1997.

[16] D. Yin, Y. Liu, H. Hu, J. Terstriep, X. Hong, A. Padmanabhan, and S. Wang, "CyberGISJupyter for reproducible and scalable geospatial analytics," Concurrency and Computation: Practice and Experience, vol. 31, no. 11, p. e5040, 2019.

[17] A. Michels, A. Padmanabhan, Z. Li, and S. Wang, "Towards Reproducible Research on CyberGISX with Lmod and Easybuild," Oct. 2021.

[18] Z. Li, A. Michels, A. Padmanabhan, A. Nassar, D. G. Tarboton, and S. Wang, "CyberGISJupyter for water - an open geospatial computing platform for collaborative water research," in AGU Fall Meeting Abstracts, vol. 2022, pp. IN32A–05, Dec. 2022.

[19] R. McLay, K. W. Schulz, W. L. Barth, and T. Minyard, "Best practices for the deployment and management of production HPC clusters," in State of the Practice Reports, SC '11, (New York, NY, USA), pp. 1–11, Association for Computing Machinery, Nov. 2011.

[20] J. Saxon, J. Koschinsky, K. Acosta, V. Anguiano, L. Anselin, and S. Rey, "An open software environment to make spatial access metrics more accessible," Journal of Computational Social Science, vol. 5, pp. 265–284, May 2022.

[21] UN General Assembly, "Transforming our world: The 2030 Agenda for Sustainable Development," United Nations: New York, NY, USA, 2015.

[22] F. Baig, A. Michels, Z. Xiao, S. Y. Han, A. Padmanabhan, Z. Li, and S. Wang, "CyberGISCloud: A unified middleware framework for cloud-based geospatial research and education," in Practice and Experience in Advanced Research Computing, PEARC '22, (New York, NY, USA), pp. 1–4, Association for Computing Machinery, July 2022.

[23] R. Kalyanam, L. Zhao, C. X. Song, A. Padmanabhan, F. Baig, and Z. Li, "The I-GUIDE CI Platform Enabling Integrative Discovery," in 15th International Workshop on Science Gateways, June 2023.