CyberGIS-Compute: Middleware for Democratizing Scalable Geocomputation

Alexander Michels[1], Anand Padmanabhan[1], Zimo Xiao[2], Mit Kotak[3], Furqan Baig[1], and Shaowen Wang[1]

[1] CyberGIS Center for Advanced Digital and Spatial Studies, University of Illinois at Urbana-Champaign; [2] Carnegie Mellon University; [3] Massachusetts Institute of Technology

INTRODUCTION

NSF Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) enables transformative discovery and innovation for tackling fundamental scientific and societal challenges that are at the cusp of achieving significant breakthroughs by harnessing the vast, diverse, and ever-growing corpus of geospatial data. As most challenging sustainability and resilience problems today require expertise from multiple domains and geospatial data science, I-GUIDE plays a central role in successfully spanning domains, leveraging its multidisciplinary collaboration team and partnerships to achieve geospatial data-intensive discovery and innovation. Collaboration is key to tackling cross-cutting challenges, but it is not always easy. Technical expertise, technology stacks, expectations, and best practices vary widely across domains.

To help address the issues of collaborating on, reproducing, and sharing complex computational workflows, we have developed CyberGIS-Compute. CyberGIS-Compute provides a simple user interface in Jupyter, streamlines the process of integrating domain-specific models with HPC, and simplifies the process of reproducing and sharing computational workflows.

CONTRIBUTING MODELS

Contributing models to CyberGIS-Compute is designed to be as streamlined as possible. You simply need:

- Your code in Github
- A JSON manifest with instructions to run the model
- A container (Docker/Singularity) that can run your model

A manifest (example to the right) provides metadata, the steps of the model, and basic computational requirements for the model which allows CyberGIS-Compute to correctly execute it!

```
{
    "name": "COVID-19 spatial accessibility",
    "description": "Calculates travel-time from hospitals and then calculates spati
    "estimated_runtime": "~20 minutes",
    "container": "cybergisx-0.4",
    "execution_stage": "MPLBACKEND=Agg python main.py",
    "slurm_input_rules": {
        "time": {
            "max": 180,
            "min": 60,
            "default_value": 120,
            "step": 1,
            "unit": "Minutes"
        },
        "cpu_per_task": {
            "max": 4,
            "min": 4,
            "default_value": 4,
            "step": 1
        },
        "memory_per_cpu": {
            "max": 20,
            "min": 16,
            "default_value": 20,
            "step": 1,
            "unit": "GB"
        }
    },
    "require_upload_data": false,
    "supported_hpc": ["keeling_community", "expanse_community", "anvil_community"],
    "default_hpc": "expanse_community"
}
```

Models support a variety of SLURM parameters, accepting parameters and input data from users, and other customizations!

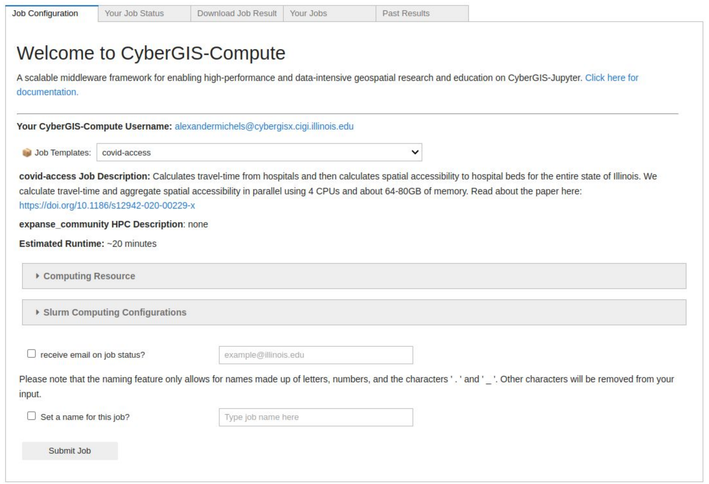
For many model developers, contributing a model is as simple as creating a container that can run the model, making some minor changes to the code to ensure that model outputs can be retrieved, and creating a manifest.

To learn more about contributing models to CyberGIS-Compute check out the guide at: https://cybergis.github.io/cybergis-compute-python-sdk/model_contribution/

USER INTERFACE

CyberGIS-Compute's graphical user interface is written in Python on top of Jupyter widgets. It provides a simple, point-and-click interface to submit jobs.

```
[1]: from cybergis_compute_client import CyberGISCompute
     cybergis = CyberGISCompute(suffix="v2")
[2]: cybergis.show_ui()
```



The CyberGIS-Compute user interface.

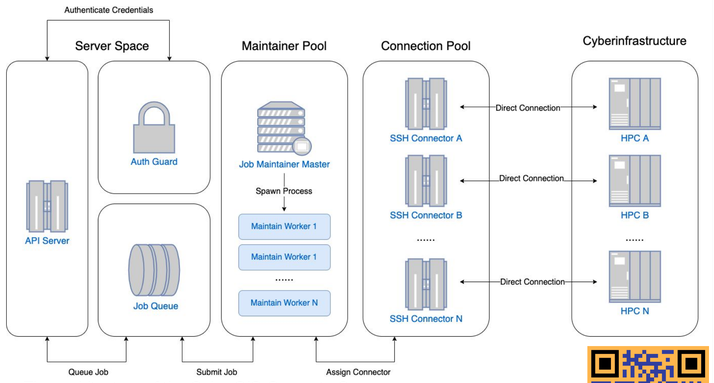
The QR code to the right brings you to the Github page for the CyberGIS-Compute Python SDK.
https://github.com/cybergis/cybergis-compute-python-sdk

CyberGIS-Compute is supported on a few JupyterHub instances:

- CyberGISX: cybergisxhub.cigi.illinois.edu
- CyberGIS-Jupyter for Water (CJW): go.illinois.edu/cybergis-jupyter-water
- I-GUIDE Platform: iguide.illinois.edu/platform

CORE SERVER

CyberGIS-Compute Core provides an API that handles the execution of models on HPC systems. Requests are authenticated using JupyterHub before sending jobs in a pool and finally submitting the jobs to our supported HPC (Expanse, Bridges-2, ACES, and more!). When the job finishes, we use Globus to transfer the results to the user from the HPC.

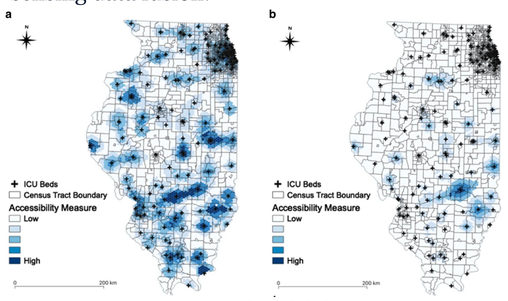


The architecture of the CyberGIS-Compute Core server.

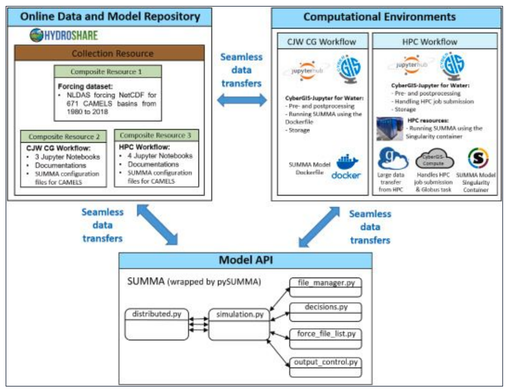
Check out the CyberGIS-Compute Core github with the QR code to the right or use the URL: https://github.com/cybergis/cybergis-compute-core

MODELS

CyberGIS-Compute currently supports a growing and diverse set of models including spatial accessibility, hydrology, and remote sensing data fusion.

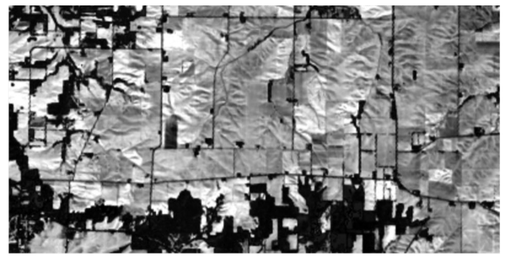


Kang et. al. (2020) utilized CyberGIS-Compute to provide replicable analyses of access to ICU beds for (a) those over 50 years of age and (b) COVID-19 patients



Workflow used by Maghami et. al. for hydrological modeling with CyberGIS-Compute

Input MODIS



Predicted Result

Example of the Remote Sensing Image Fusion Model by Lyu et. al. (2022). The work utilizes a robust hybrid deep learning model based on a super-resolution convolutional neural network (SRCNN) and long short-term memory (LSTM) for spatiotemporal data fusion to integrate Landsat images with MODIS.

NEXT STEPS

We are actively working to improve and increase adoption of CyberGIS-Compute and we welcome all kinds of contributions:

- New models/workflows/analyses
- Features for the User Interface / Python SDK
- Improvements to the Core server

The CyberGIS-Compute development team is currently working on the following improvements:

- Integrating the Cern Virtual Machine File System (CVMFS) to exactly replicate the compute environments on the supported JupyterHubs
- Creating better documentation, more models, and additional features to better support geospatial machine learning/AI applications
- Supporting private Github repositories
- Providing a simplified user interface for model end-users
- Allowing for provide allocations (currently all jobs utilize our community allocation from ACCESS)
- Integrating cloud providers to allow for more diverse computational workflows.

REFERENCES

Maghami, I., et. al. Building cyberinfrastructure for the reuse and reproducibility of complex hydrologic modeling studies. Environmental Modelling & Software. 164 (2023). https://doi.org/10.1016/j.envsoft.2023.105689

Kang, JY., Michels, A., Lyu, F. et al. Rapidly measuring spatial accessibility of COVID-19 healthcare resources: a case study of Illinois, USA. Int J Health Geogr 19, 36 (2020). https://doi.org/10.1186/s12942-020-00229-x

Lyu, F., Yang, Z., Xiao, Z., Diao, C., Park, J., and Wang, S. 2022. CyberGIS for Scalable Remote Sensing Data Fusion. In Practice and Experience in Advanced Research Computing (PEARC '22). Association for Computing Machinery, New York, NY, USA, Article 35, 1-4. https://doi.org/10.1145/3491418.3535145

Padmanabhan, A., Xiao, Z., Vandewalle, R., Baig, F., Michels, A., Li, Z., and Wang, S.. CyberGIS-Compute for Enabling Computationally Intensive Geospatial Research. SpatialAPI'21: Proceedings of the 3rd ACM SIGSPATIAL International Workshop on APIs and Libraries for Geospatial Data Science, 2021. https://doi.org/10.1145/3486189.3490017

Acknowledgements: This material is based upon work supported by the National Science Foundation 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 the National Science Foundation.

Contact: Shaowen Wang • University of Illinois Urbana-Champaign • shaowen@illinois.edu • https://iguide.Illinois.edu

I-GUIDE Institute for Geospatial Understanding through an Integrative Discovery Environment