CyberGIS-Compute: Geospatial Middleware for Simplifying Access to High Performance Computing

Anand Padmanabhan and Alexander Michels
University of Illinois Urbana-Champaign
CyberGIS-Compute Team

- Mit Kotak
- Zhiyu Li
- Alexander Michels
- Anand Padmanabhan
- Shaowen Wang
- Zimo Xiao
- Taylor Ziegler
Motivation

• Geospatial discovery and innovation are increasingly computation and data intensive
• Personal computing environments are limited to resolve such computational intensity
• High-performance computing (HPC) environments are needed to enable computation- and data-intensive geospatial scientific workflows
But …

• The learning curve to access and use HPC is very steep!
Examples

Spatial Accessibility Calculation

Estimate Height Above Nearest Drainage

WRFHydro Model
What is CyberGIS-Compute?

- Simplify access to HPC
- Bridge the gap between interactive computing environments (e.g. CyberGIS-Jupyter) and HPC
- Enable computation- and data-intensive geospatial workflows
User Interface

University of Illinois at Urbana-Champaign
College of Liberal Arts and Sciences
School of Earth, Society and Environment (SESE)

REMINDER:
This is the login node for keeling, and it is shared among multiple users. Do not run parallel programs requiring more than seven compute threads on this machine; please use the batch system for such programs instead.

Welcome to CyberGIS-Compute
Some description about CyberGIS-Compute

Job Configurator: Your Job Status: Download Job

Job Templates:
hello_world Job Template
keeling_community

Estimated Runtime:

Computing Resources:

- Computing Resource: keeling_community

Slurm Computing Configurations
All config are optional. Please refer to Slurm official documentation at ...
num_of_tasks
num_of_nodes
Bridging Ease of Use with Powerful Computing

Geospatial Middleware

A **scalable middleware framework** for enabling high-performance and data-intensive geospatial research and education
Key Components

• **Core**: middleware server that automates job submission to HPC

• **SDK**: interactive client for Jupyter Notebook with code-less UI support

• **Contribution**: developer API that enables workflow contribution with little to no modification of existing code
Architecture
Middleware Server

- Manages job submission to HPC
- Three layers
  - API Server Space
  - Maintainer Pool
  - Connection Pool

API Server

• Provides a **user-facing RESTful web interface** (API Server) for authentication and interaction with internal components of CyberGIS-Compute

• Pushes computation jobs into an internal job queue that the Maintainer Pool consumes
Maintainer Pool

• Maintainer Pool has a multithreaded life cycle that spawns and oversees worker processes called **Maintainer Workers** that contain **logic** on submitting, stopping, resuming, and ending a task executing on a remote HPC resource.

• A Maintainer Worker can use a **Connector** to interact with remote HPC via SSH.
Connectors

- A Connector is a **long-living SSH connection** between the CyberGIS-compute core and HPC.
- To reduce the connection rate, connectors are shared between Maintainer Workers and a **Mutex** is implemented to avoid race conditions in executing commands.
SDK: Client Package

- A **Python-based Jupyter Notebook client** that integrates CyberGIS-Compute Core functionalities into CyberGIS-Jupyter
- Provides **seamless** interaction with HPC
- Provides **code-less** interactive UI

Large Datasets

Big Data + globus online = Fast & Reliable Transfer
Seamless Access to HPC

**Containerization**: Run code in familiar environments

**Transparencyly interfaces with batch systems (e.g. Slurm)**: Manage Slurm on behalf of developers
CyberGIS-Compute Contribute allows users to submit workflow code hosted on GitHub repositories to be executed on HPC resources.

Submissions are verified through a checking process.

Provides configurations, system environment, and developer API.

Git commit version lock for security.
Contribution Process

```python
import json
import os

print('running main...

print('./job.json

job = json.load(open('./job.json',))

print('SLURM_NODEID

print(os.environ['SLURM_NODEID'])

print('SLURM_PROCID

print(os.environ['SLURM_PROCID'])

print('job_id')

print(os.environ['job_id'])
```

+ **manifest.json**  
  
  =  
  
  **ready for HPC**
Links

Project Development doc:  
http://github.com/cybergis/cybergis-compute-core

SDK doc:  
http://github.com/cybergis/cybergis-compute-python-sdk

Hello World doc:  
http://github.com/cybergis/cybergis-compute-hello-world
Acknowledgments

- National Science Foundation
  - 2118329
  - XSEDE
- Virtual ROGER
Thanks!

Comments / Questions?

Email: apadmana@illinois.edu