2022 Computational Infrastructure for Geodynamics Developers Workshop

Brad T. Aagaard, U.S. Geological Survey
Rene Gassmoeller, University of Florida
Lorraine Hwang, University of California, Davis

Summary

The Computational Infrastructure for Geodynamics (CIG) held its second Developers Workshop February 28 and March 1, 2022. The workshop focused on the following specific topics and recommendations:

- Alignment of CIG Best Practices with Journal of Open Source Software (JOSS) review criteria as a pathway to publishing a peer-reviewed paper associated with open source software;
- Updates to CIG Best Practices for documentation and testing to encourage more dynamic and interactive html documentation and the use of continuous integration testing tools such as GitHub Actions and Azure Pipelines;
- Using libraries to leverage accelerator architectures, Kokkos as an example;
- Building containers for high-performance computing systems, e.g. Shifter, Singularity, Charliecloud, and Sarus - this is not easy and is still undergoing significant development;
- Using the hIPPYlib library for Bayesian inference; and
- Introduction to SCOPED (Seismic COmputational Platform for Empowering Discovery) project for Improving accessibility and portability of seismic inversion software.

The 46 workshop participants represented developers of CIG community codes, those interested in contributing to the community codes, and researchers interested in improving their own codes. The workshop was held online and consisted of a mixture of presentations (6) and discussion.

Introduction to the workshop

The Computational Infrastructure for Geodynamics (CIG) held its second Developers Workshop February 28 and March 1, 2022. Workshop participants (46) were from 7 countries. Approximately 17% (8) identified as female and 28% (13) were early career researchers (graduate students and postdoctoral fellows). See Appendix A.

The workshop followed the same general themes as in 2021:

- Expanding the CIG software developer community;
- Making CIG software more accessible to new users;
- Identifying and leveraging common infrastructure; and
- Leveraging collective wisdom to make CIG software better and easier to develop and maintain.

The complete agenda is given in Appendix B. To follow are the summaries of the presentations and discussions.

Alignment of CIG Best Practices with Journal of Open Source Software (JOSS) review criteria

Daniel Katz provided an overview of JOSS and the review process and criteria. JOSS strives to make it easy to publish a peer-reviewed paper associated with open source software. The goal is to improve the quality of the software associated with manuscripts submitted to JOSS. The JOSS review criteria focus on minimum standards for the software in the areas of licensing, scholarly effort, documentation (statement of need, installation instructions, examples, API documentation, and community guidelines), functionality, and testing.

CIG Best Practices address software development and training and focus on encouraging developers to adhere to high levels of best practices. The general consensus from the discussion was that CIG Best Practices minimum Best Practices are similar to the JOSS review criteria and CIG should review the minimum level to identify inconsistencies with JOSS review criteria. In areas with inconsistencies, CIG should consider updating the Best Practices to better align with JOSS review criteria.

Updates to CIG Best Practices for documentation and testing

Software development tools continue to evolve, especially in the areas of documentation and testing. CIG Best Practices could provide more guidance in these areas to help developers deliver better documentation for users and improve the quality of software through better testing.

Menno Fraters demonstrated the use of Markedly Structured Text (MyST) and Sphinx to generate dynamic (html files) and static (PDF file) documentation for the Geodynamic World Builder. Tools like Sphinx produce documentation that is more dynamic and interactive than tools like LaTeX. Additionally, Jupyter notebooks (discussed at the 2021 CIG Developers workshop) can also produce interactive documentation. CIG should update its software development Best Practices at the target level to encourage developers to leverage these types of tools to deliver dynamic documentation that addresses user needs.

Many online repositories, such as GitHub and GitLab, provide integration with continuous integration (CI) testing tools, either through their own services, third-parties, or private infrastructure. CI tools automate the process of running tests when code is changed; usually the testing is triggered by a pull request (GitHub) or merge request (GitLab). This ensures consistent testing on a variety of platforms and helps detect bugs before they are introduced into stable development branches. In the early years of CIG, CIG provided testing infrastructure to developers that was integrated with the CIG subversion repository. Some CIG projects are currently leveraging CI testing infrastructure provided by GitHub Actions and Azure Pipelines. The consensus from the discussion was that CIG should update the software development Best Practices at the standard level to include using CI tools to automate running tests.

Additional discussion focused on using Integrated Development Environments (IDEs) in training and how to encourage developers to follow the CIG Best Practices at the standard and target levels. Brad Aagaard summarized his positive experience using the Visual Studio (VS) Code IDE and Docker containers during the 2021 PyLith Hackathon. The use of Docker containers eliminated the need for participants to build dependencies and VS Code provides a development experience with the containers that is almost identical to working with code locally. Additionally, VS Code provides sharing capabilities that allow multiple people in different locations to view and potentially edit code. One idea for encouraging developers to follow CIG Best Practices involves expanding the current labeling system to a badge

system to highlight the support level and adherence to CIG Best Practices. The consensus from the discussion was that CIG should form a small group to develop a badging system that could then be discussed by the CIG community.

Using libraries to leverage accelerator architectures

The workshop included presentations on Kokkos (Christian Trott), a C++ library that provides portable data structures and parallel execution for high-performance computing, support for CUDA in deal.II (Bruno Turcksin), and support for Kokkos in PETSc (Patrick Sanan). All three presentations noted the variety of accelerator architectures will likely continue to grow and application developers should leverage libraries like Kokkos that provide excellent performance across a variety of high-performance computing architectures. The presentations also highlighted the differences in design between codes written for CPUs and those designed to potentially leverage accelerator architecture. The discussion raised the issue that deal.II and PETSc need to provide additional documentation describing how to design applications that use deal.II or PETSc to best leverage high-performance computing architectures.

Building containers for high-performance computing systems

Shane Canon gave a presentation on using containers on high-performance computing systems.
He mentioned that there are multiple high-performance computing container runtimes, including Shifter, Singularity, Charliecloud, and Sarus. Singularity is very popular, Charliecloud is very light-weight, and Shifter converts Docker images to an optimized internal format and integrates with batch scheduler for scalability. The runtimes typically exploit application binary interface (ABI) compatibility to achieve native MPI performance; it strikes a balance between portability and performance and works very well for the MPICH MPI implementation. One can apply similar techniques for GPUs. Builds for multiple architectures are more experimental. Docker buildx can be used to build and publish multiple architecture images; it assumes the Dockerfile itself is portable and works on the different target architectures. Shane's tips for building containers for high-performance computing systems include:

- Use an OCI image format and recipe;
- Consider tools like Spack, Easybuild, or HPC Container Maker (nVidia) to generate and manage build recipes;
- Avoid using volume mounts to inject code (except for development);
- Avoid approaches that require modifying the image at runtime;
- Design images to be read-only without privileges; and
- Use a container-enabled workflow tool, like Nextflow, CWL, or WDL.

The consensus from the discussion is that building portable containers for high-performance computing systems is not easy and the tools are still undergoing significant development. As a result, some experimentation will be needed. Addressing this issue could be a goal for CIG in the next few years.

Using the hIPPYlib library for Bayesian inference

Most codes developed by CIG have focused on solving partial differential equations for specified sources or forcing. Interest in using these codes in inversion frameworks is growing. Umberto Villa provided an overview of the hIPPYlib (inverse Problem PYthon) library for Bayesian inference. In general, the Bayesian approach is difficult and computationally expensive because it involves exploring a high dimension space which is exponentially sparse. Conventional Monte Carlo methods are prohibitively expensive. The approach used by hIPPYlib exploits the structure of the posterior probability over the parameter space. The number of model solves is independent of parameter and data dimensions; it only

depends on the “information” dimension. However, this approach requires more information from the forward model (gradient and Hessians).

The hIPPYlib library provides a user-friendly, compact, mathematical notation to express the partial differential equation and likelihood in weak form. It includes automatic generation of efficient code for discretization of weak forms using FEbiCS, scalable algorithms, maximum a posteriori point computation, low rank representation of the posterior covariance via randomized algorithms, scalable sampling from prior and posterior distributions, forward and inversion propagation of uncertainties.

The discussion highlighted the importance of an efficient forward model as well as the utility of the general approach used by hIPPYlib even if it is implemented using other tools. In developing software for inversions and data assimilation, the CIG community may want to consider using hIPPYlib or implementing its approach or similar approaches.

The SCOPED improving accessibility and portability of seismic inversion software

Inversions for source parameters and Earth structure remains a principal research area in seismology. However, the workflow for compiling and processing data, computing sensitivity kernels, and running inversions often involves manually interfacing multiple pieces of software and considerable effort. Carl Tape and Ian Wang gaven an overview of the Seismic COmputational Platform for Empowering Discovery (SCOPED) project, which provides a novel hybrid cloud and high-performance computing platform intended for use by the scientific community to combine processing and modeling of seismic data. SCOPED goals include:

- Enable connectivity with other modeling codes;
- Improve documentation for existing tools;
- Improve training;
- Improve software portability through the use of containers; and
- Enable scientific discovery.

The toolkit is based on the Massive Parallel Analysis System for Seismologists. The toolkit includes data management with SeismoDB, big data processing with Dask or Spark, H5Coro and object store in the cloud, high–performance input and output using ThemisIO, and job scheduling using Terraform. The project plans to release the big data analysis tools as Common Container images and the simulation applications as Performance-Optimized Container images. The CIG community may want to leverage some of these tools or approaches in other disciplines or other projects.

Links

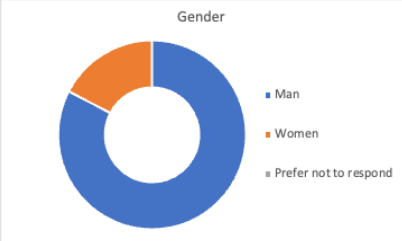
2022 CIG Developers Workshop webpage
CIG Best Practices
Journal of Open-Source Software review criteria
Kokkos Lecture Series
Texas Advanced Computing Center (TACC) containers
hIPPYlib
Seismic COmputational Platform for Empowering Discovery (SCOPED)

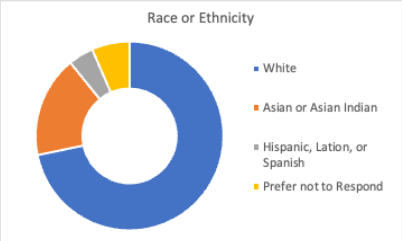
Disclaimer

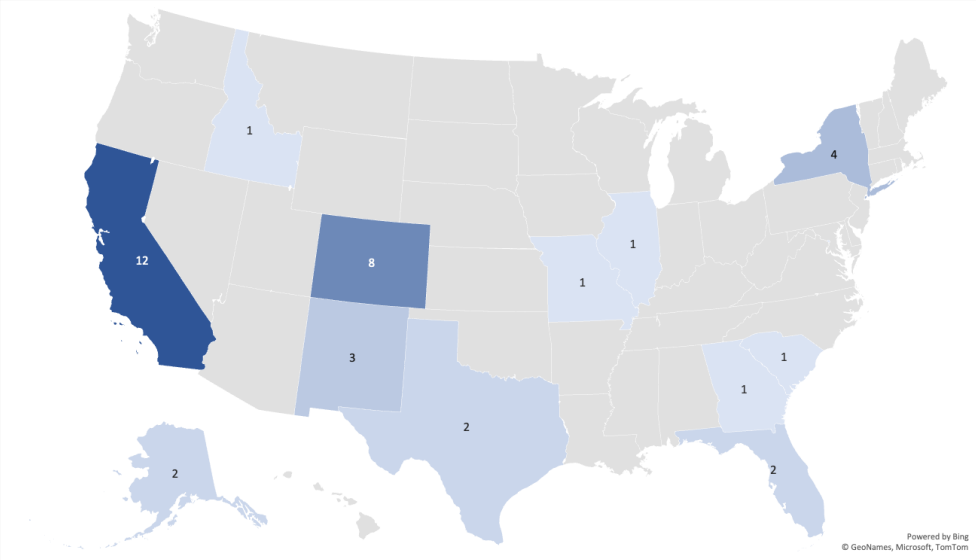
Appendix A: Participants

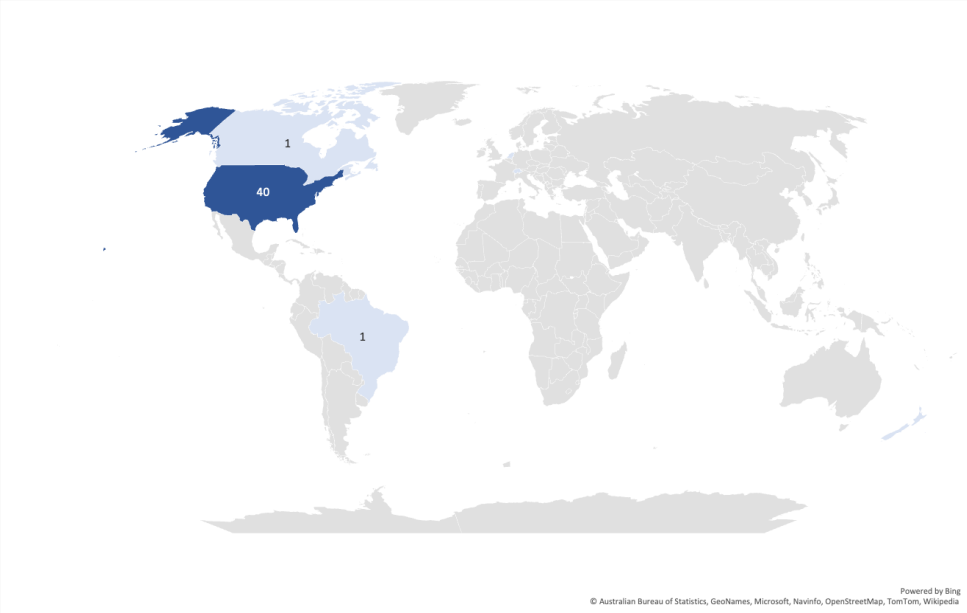
Total Attendees: 46 *minimum*

Demographics









Appendix B: Agenda

All times are Pacific Time (UTC-8)

Day 1: Monday February 28

8:00 Opening Remarks and Workshop Objectives

SESSION I: CIG BEST PRACTICES

8:10 Brief review of CIG Best Practices for software - *Rene Gassmoeller, University of Florida*

8:20 Journal of Open Source Software (JOSS): Developing a software review community - *Daniel Katz, University of Illinois at Urbana-Champaign*

8:40 Discussion of JOSS and CIG Best Practices for software - *Moderator: Rene Gassmoeller*

9:00 Updates to CIG Best Practices

- Usage of VS Code and Docker for training - *Brad Aagaard, US Geological Survey*
- Continuous Integration using Azure Pipelines - *Brad Aagaard, US Geological Survey*
- Online Documentation with Sphinx - *Menno Fraters, UC Davis*

9:20 Discussion - *Moderators: Rene Gassmoeller and Lorraine Hwang*

9:40 Break

SESSION II: HPC FRAMEWORKS

10:10 Kokkos: A brief overview - *Christian Trott, Sandia National Laboratories*

10:40 GPU Support in deal.II - *Bruno Turcksin, Oak Ridge National Laboratory*

10:55 Using GPUs with PETSc - *Patrick Sanan, ETH Zurich*

11:10 Discussion - What is the best path for using accelerated architectures? *Moderator, Juliane Dannberg*

Day 2: Tuesday March 1

8:00 Recap of Day 1

SESSION III: BUILDING CONTAINERS FOR HPC

8:10 Building Portable Containers for HPC - *Shane Canon, Lawrence Berkeley Laboratory*

8:40 Discussion - *Moderator: Jed Brown*

SESSION IV: SCOPED

9:00 SCOPED: Introduction and Project Updated Seismic COmputational Platform for Empowering Discovery - *Carl Tape, University of Alaska Fairbanks and Ian Wang, the University of Texas at Austin*

9:20 Discussion - *Moderator: Shi Joyce Sim*

9:30 Break

SESSION V: BAYESIAN INFERENCE

10:00 HIPPYlib: An Extensible Software Framework for Large-Scale Inverse Problems Governed by PDEs - *Umberto Villa, Washington University, St. Louis*

10:20 Discussion - *Moderator, Ebru Bozdag*

WRAP UP

10:40 Summary Discussion