A Portable Software Environment for Ultrahigh-Resolution ELM Development on GPUs

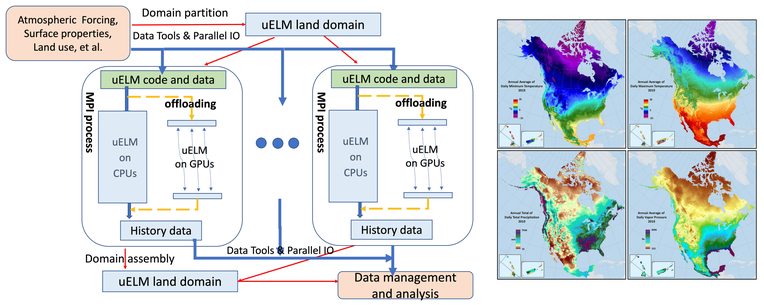
F. Eaglebarger[a,b], D. Wang[c], F. Yuan[c], P. Schwartz[c], C. Layton[c], R. Weiskopff[c], D. Riccuito[c], P. Thornton[c]
[a]Pellissippi State Community College, [b] Oak Ridge Institute for Science and Education, [c]Oak Ridge National Laboraotry

Abstract

A software tool called SPEL has been developed to port and optimize the ultrahigh-resolution ELM (uELM) code onto GPUs within in a functional unit test framework. To promote the widespread adoption of this approach for community-based uELM development, this poster presents an initiative to create a portable software environment that enables efficient development of ultrahigh-resolution ELM (uELM) code on GPUs. Technically, the standalone software environment, which utilizes Docker, contains all the necessary code, libraries, and system software required for uELM development using SPEL. The process involved in this study includes identifying a Docker image that supports GPU, configuring and simulating ELM at the site level, capturing reference solutions, testing uELM functional units, and generating and optimizing code that is compatible with GPUs. The effectiveness of this method is demonstrated through a case study.

Project Background

Ultrahigh-resolution E3SM Land Model (ELM) Development [1-3]



Portable Software Environment for uELM

Procedure for uELM code porting/optimization via SPEL

1: Identify a Docker image with Nvidia GPU support (nvcr.io)

2: Customize docker image for ELM simulation at site level

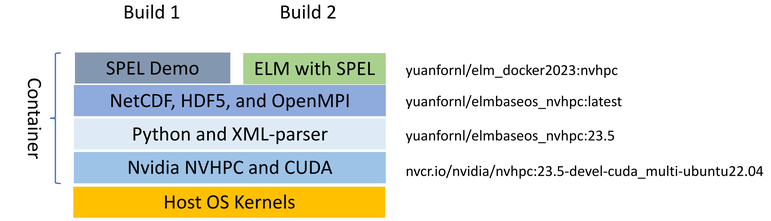
3: Capture reference solution

4: uELM functional unit testing using SPEL

5: GPU-ready code generation and optimization via SPEL

Docker Layer Architecture

- Dockerfiles are text files that contain a set of instructions for building a Docker image, used to automate the process of creating and distributing applications.
- Docker uses a layered file system to build and manage images, enabling its copy-on-write mechanism, where changes made to a container are stored in a new layer without modifying the underlying layers.



A Case Study Demonstration

This software environment has significantly shortened the time required to complete tasks that previously took weeks to months, now making them achievable within in hours. By utilizing the Github and DockerHub repositories, developers and researchers can promptly access uELM/SPEL and develop uELM code for GPUs. These repositories offer precise instructions and concise background information, facilitating user navigation.

A case study to generate and optimize a uELM module (LakeTemperature) on GPUs is available at github.com/daliwang/Docker4SPEL. Users can

1. Download Docker system
2. Pull the necessary Docker image
3. Run the image (using nvidia-docker)
4. Execute LakeTemperature unit test module within the running container

LakeTemperature Unit Test Demonstration on a DGX station

Docker Layer Architecture: AMD x86-64

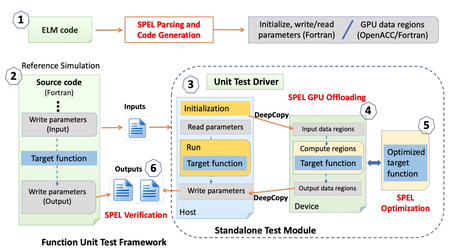
github.com/Eaglebarger/SPEL-Docker

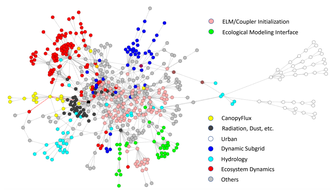
1. `nvcr.io/nvidia/nvhpc:23.5-devel-cuda_multi-ubuntu22.04`
   - DGX Machine NVIDIA GPU utilizing Docker image
2. `yuanfornl/elm_baseos_nvhpc:23.5`
   - Initial dependencies, locale configuration, Python/pip3, ex
3. `yuanfornl/elm_baseos_nvhpc:latest`
   - HDF5, netCDF, netCDF-Fortran, OpenMPI
4. `fje1223/spel_docker_nvhpc`
   - SPEL repository & software necessary dependencies
   - SPEL Origination/Documentation: SPEL_OpenACC

hub.docker.com/r/fje1223/spel_docker_nvhpc

SPEL: Software toolkit for Porting ELM using Compiler Directives

SPEL in the ELM Functional Unit Testing Framework[4-6]

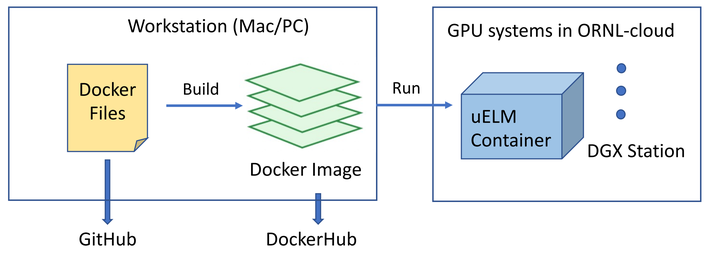




½ million lines of code
Auto code generation (> 98%)
2 X speedup GPU/CPU

Development and Distribution System

Several computing resources are used in this study, including Mac and Windows laptops/desktops, an NVidia DGX station (with 4 V100) and a customized GPU system (2 A100s) in the ORNL Research Cloud using the OpenStack platform.



Conclusion

The portable software environment is a self-contain, standalone system for uELM code development using SPEL on hybrid CPU/GPU systems. Through the GitHub/Docker Hub repositories, this system provides a convenient solution to support community-based uELM code development on various platforms.

Acknowledgment: Mr. Franklin Eaglebarger is funded by the Community College Internships (CCI) program in the US Department of Energy (DOE) through the Oak Ridge Institute for Science and Education. This research leverage previous efforts that were supported as part of the Energy Exascale Earth System Model (E3SM) project and the Next Generation Ecosystem Experiment-Arctic (NGEE-Arctic) project, funded by DOE's Office of Science, Office of Biological and Environmental Research. This research used resources of the Oak Ridge Leadership Computing Facility and Computing and Data Environment for Science at the Oak Ridge National Laboratory, which are supported by the DOE's Office of Science under Contract No. DE-AC05-00OR22725.

References

1. Wang, D., Schwartz, P., Yuan, F., Thornton, P., & Zheng, W. (2022). Toward Ultrahigh-Resolution E3SM Land Modeling on Exascale Computers. *Computing in Science & Engineering*, 24(6), 44-53. DOI: 10.1109/MCSE.2022.3218990
2. Thornton, P.E., Shrestha, R., Thornton, M. , Kao, S.C., Wei, Y., Wilson, B. (2021) Gridded daily weather data for North America with comprehensive uncertainty quantification. *Scientific Data* **8**, 190 (2021). https://doi.org/10.1038/s41597-021-00973-0
3. Yuan, F., Wang*, D., Kao, S., Thornton, M., Ricciuto, D., Sulmon, V., Iversen, C., Schwartz, P., & Thornton, P. (2023). An ultrahigh-resolution E3SM Land Model simulation framework and its first application to the Seward Peninsula in Alaska, *Journal of Computational Science*, Vol73, pp02145. DOI: 10.1016/j.jocs.2023.102145
4. Schwartz, P., Wang, D., Yuan, F., & Thornton, P. (2022). SPEL: Software Tool for Porting E3SM Land Model with OpenACC in a Function Unit Test Framework. In the 9th *Workshop on Accelerator Programming Using Directives*. DOI: 10.1109/WACCPD56842.2022.00010.DOI: 10.1109/WACCPD56842.2022.00010
5. Schwartz, P., Wang, D., Yuan, F., & Thornton, P. (2022). Developing an ELM Ecosystem Dynamics Model on GPU with OpenACC. In *Lecture Notes on Computer Science*, D. Groen et al. (Eds.): ICCS 2022, LNCS 13351, 1–13. DOI:10.1007/978-3-031-08754-7_38
6. Schwartz, P., Wang, D., Yuan, F., & Thornton, P. (2023). Developing Ultrahigh-Resolution E3SM Land Model for GPU Systems. In et al. (Eds.), *Computational Science and Its Applications* – ICCSA 2023 (pp. 13956). Lecture Notes in Computer Science. Springer, Cham. DOI: 10.1007/978-3-031-36805-9_19

© Poster Template by Genigraphics® 1.800.790.4001 www.genigraphics.com