EarthWorks

Scaling a Fully Coupled Climate Model to Run at Ultra-High Resolutions

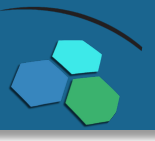
Sheri Voelz[1]
PIs: David Randall[2], Jim Hurrel[2]
Richard Loft[3], Thomas Hauser[1], Michael Duda[1], Dylan Dickerson[1], Supreeth Suresh[1], Jian Sun[1], Chris Fisher[1], Donald Dazlich[2], Gunther Huebler[4], Jim Edwards[1], Brian Dobbinsr[1], Raghu Raj Kumar[5], Pranay Reddy Kommera[5]

1 National Center for Atmospheric Research
2 Colorado State University
3 AreandDee, LLC
4 University of Wisconsin, Milwaukee
5 NVIDIA Corporation

June 4, 2024

by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

What is EarthWorks?

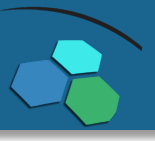


EarthWorks is a five-year funded project, led by Colorado State University, with participation from half of the laboratories within the NSF National Center for Atmospheric Research, NVIDIA, and the University community

This project is funded by the National Science Foundation (NSF) through their Cyberinfrastructure for Sustained Scientific Innovation (CSSI) program

We are just starting the last year of this project

The Goals of EarthWorks



Science Goals

- Begin to resolve storms at ~4-km grid.
- Eliminate deep convection or gravity-wave drag parameterizations.
- Include a resolved stratosphere.
- Enable new science (extreme events!) for both weather and climate.
- Provide a critical capability to the climate community for guiding adaptation at global, regional and local levels.

Computational Goals

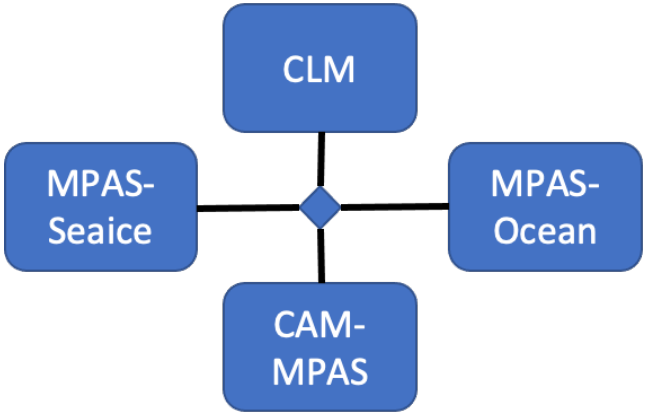
- The EarthWorks ESM will run on CPUs for low resolution experiments and for testing short ultra-high resolution setups.
- Provide GPU-enabled implementations of ocean and atmosphere for tackling high resolutions.
- EarthWorks will put huge demands on computational and data systems. Thus the project incorporates infrastructure development efforts for both big data and machine learning inference.

Infrastructure Targets for EarthWorks

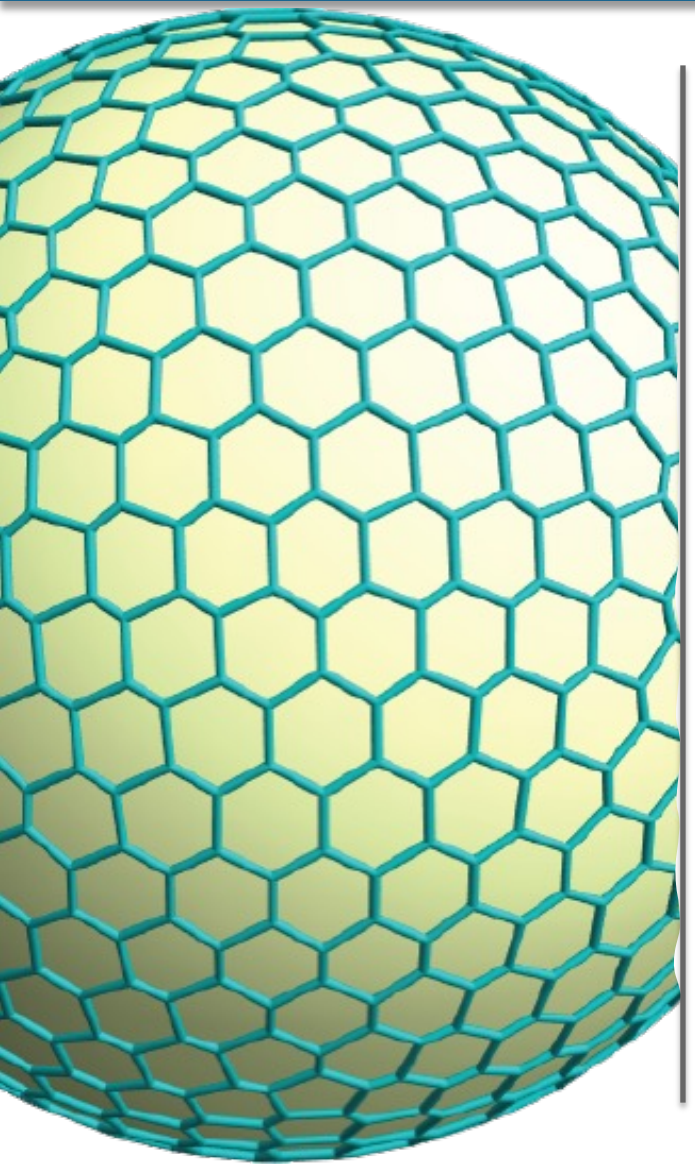
Based on the Community Earth System Model (CESM), but it is NOT CESM

1. The MPAS non-hydrostatic dynamical core, with a resolved stratosphere and CAM physics
2. *The MPAS ocean model, developed at Los Alamos
3. *The MPAS sea ice model, based on CICE
4. The Community Land Model (CLM)
5. The Community Mediator for Earth Prediction Systems (CMEPS)

* indicates where EarthWorks differs from CESM



Infrastructure Targets for EarthWorks

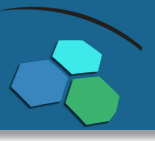


EarthWorks uses the same *geodesic mesh and cell spacing* for all components.

A mesh spacing of 3.75 km works well for the atmosphere, the ocean, and the land surface.

Using a single mesh reduces the operation count, message-passing overhead, and memory requirements.

Supported Model Configurations



- **Resolutions (120 km, 60 km, 30 km, 15km, 7.5km, 3.75km)**

- **Five configurations with MPAS-dynamical core in CAM**
  - Idealized Held Suarez climate
  - Moist baroclinic wave + KESSLER microphysics
  - Aquaplanet - full up atmosphere with data ocean, no land
  - CAM6-MPAS atmosphere with CTSM (land) + data ocean
  - CAM-6 MPAS atmosphere + CTSM + MPAS-Ocean + MPAS Sea-Ice

End-to-end Workflow Portability

Objective	Tools
Revision Control	Github
Containers for portability	Singularity and Docker
Performance portability	OpenACC, OpenMP, OpenMPI
Scalable I/O	PIO
Analysis	Atmospheric Diagnostic Framework and Raijin
Data Transfer	Globus
Science Gateway	Containerized Gateway

We're Working Along Four Fronts

Running short ultra-high resolution simulations on CPUs to flush out potential issues

Porting the atmospheric model to run on GPUs

Maintaining the repository and making sure it's in sync with CESM

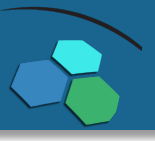
Developing workflows for data visualization and analysis on the native MPAS grid and at scale

Running short ultra-high res simulation to flush out potential issues

Issues encountered (and fixed!)

- **Initialization (abnormally long times)**
  - Traced to an issue in the ESMF framework (needed to turn off subcomponents), resulted in a patch release.
  - **Impact: 5.7x speedup, 2x reduction in memory use in initialization.**
- **Initialization (abnormally long times, MPI communication)**
  - Traced to an issue in the ESMF framework, Alltoall MPI call (~164MB on 164K cores)
  - Workaround was created to eliminate the need for this call, long term fix still needed
  - **Impacts: 41K cores: 45 mins to ~2 mins**
- **Large slowdown in history I/O bandwidth**
  - Traced to the ROMIO MPI-IO implementation in PnetCDF, resulted in a problem report and workaround.
  - **Impact: expected history I/O performance restored**
- **Run after restart errors**
  - Traced to an issue with the PIO2 (parallel I/O) infrastructure in CESM, resulted in a patch release.
  - **Impact: correct model restarts restored**
- **IO Performance**
  - Sort variables by size and write all similar variables together
  - **Impact: IO made several times faster**

Our Approach to Porting Onto GPUs



- We are porting using the OpenACC directive approach
- We then use a tool developed by Intel to add OpenMP offload directives. This tool looks at the OpenACC directives and adds the equivalent OpenMP offload directive.
- We are continuously evaluating different methods, we found there are pros/cons for all methods
- Determine what is important to you and chose method based on that

Intel(r) Application Migration Tool written by Harald Servat
https://github.com/intel/intel-application-migration-tool-for-openacc-to-openmp

https://github.com/larson-group/clubb_release/blob/master/compile/convert_acc_to_omp.bash

Porting Status

Component	Subcomp	Package	Porting Status	Offload Paradigm
Atmpsphere			In progress	
	Dycore	MPAS-7.x	completed	OpenACC
	Physics	CAM	In progress	
		PUMAS	completed	OpenACC + OpenMP
		RRTMGP	completed	OpenACC + OpenMP
		CLUBB	completed	OpenACC + OpenMP
Ocean	MPAS-O		completed	OpenACC
Sea-ice	MPAS-SI		deferred	OpenACC

We have successfully ran a model configuration with PUMAS, RRTMGP, MPAS dycore GPU combined run
We will be testing the same configuration with CLUBB/GPU shortly

MPAS Dycore Performance

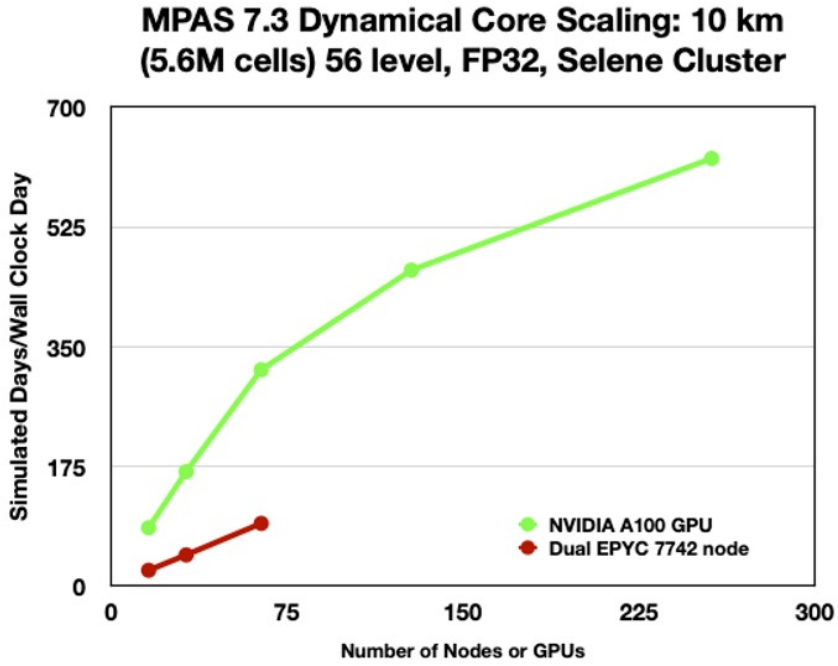
Experiment: MPAS-7 (5.9M cell mesh; 56 levels; FP32) ran dry baroclinic test case for 10 simulated days

Equipment: Selene supercomputer; nodes = AMD Dual socket EPYC 7742 “Rome” CPUs with 8x NVIDIA A100 GPUs; 10 HDR links/node.

Resources: Benchmark of 128-core ROME CPU node vs A100 GPU

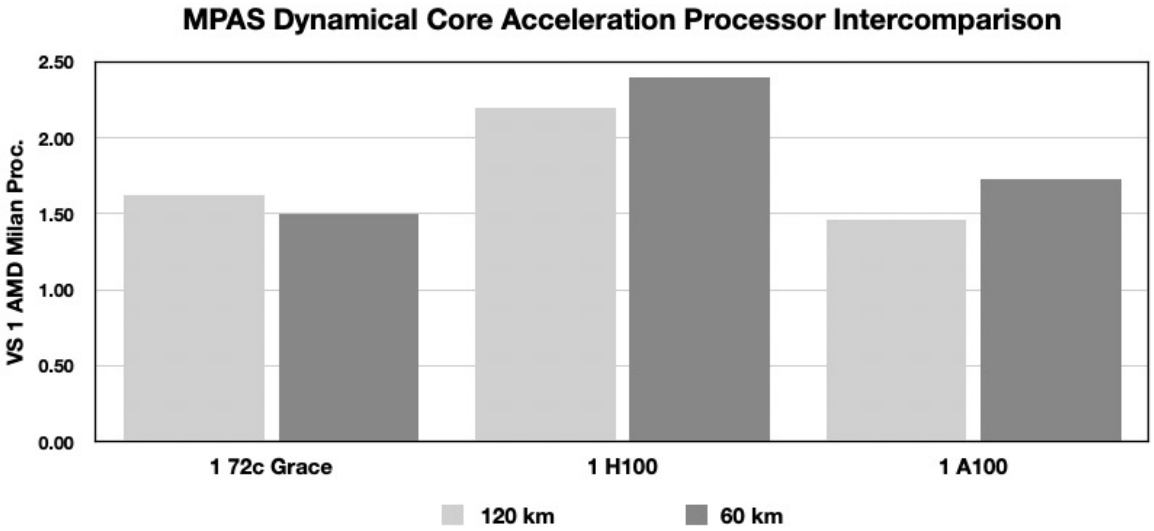
Takeaways:

- **3.5x faster than CPU node.**
- **Slowdown of MPAS-7 compute (m) was isolated to not declaring new variables GPU resident.**



Results courtesy of Raghu Raj Kumar of NVIDIA for benchmarking MPAS-7

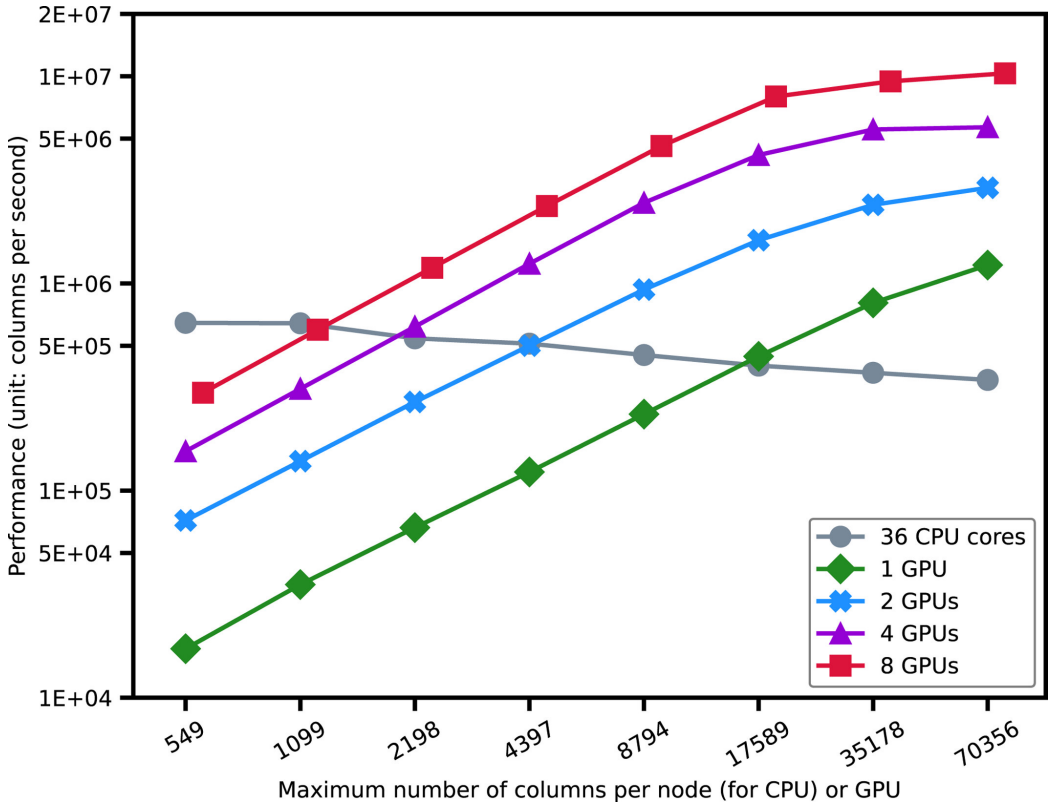
Grace and Hopper Performance Comparison



- Here we are doing a "socket to socket" comparison of a Grace processor, H100, and A100 against a single Milan Processor.
- CAM-MPAS dynamical core on a quasi-uniform global grid with 32 levels is offloaded with OpenACC directives.
- All GPU experiment were run with a single Host rank offloading to the GPU. In other words, no MPS (Multi-Process Service) was used.
- **Notable features:**
  - **Grace is slightly faster core for core than Milan**.
  - **H100 is about 1.5 faster than A100** on the MPAS dynamical core workload.
  - The ratio increase seen for GPUs between 120 km (40K columns) and 60 km (160 columns) could be attributable to either CPU cache or GPU occupancy (data parallelism) effects.
- **Conclusions and Next Steps:**
  - Single H100 results are encouraging but need Multi-A100/H-100 benchmarks.
  - Integrated GPU-dycore + GPU-physics testing will push us to multi-ranks per device. Where's the sweet spot?
  - Multi-GPU, multi-node results at higher resolutions coming soon.

Results courtesy of Rich Loft of AreandDee, LLC

PUMAS/MG3 Performance



This plot compares the OpenACC performance on the NVIDIA V100 against one full CPU node (36 Intel Skylake CPU cores) for different numbers of columns per node in the standalone PUMAS kernel (does not include memory transfer time).

(Results courtesy of Jian Sun)

https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2022MS003515

RRTMGP

- Utilizing code from Robert Pincus, et al
https://github.com/earth-system-radiation/rte-rrtmgp

- We have incorporated RRTMGP into the latest CAM development version

- Verified that answers match between CPU and GPU

- Very preliminary results show about a 10x speedup on GPUs in the stand alone version (without data transfers), though we are not currently seeing this performance within CAM

- We are currently working to optimize the code and the memory movement to improve performance within CAM

CLUBB Performance

Goal: Create performance portability across compilers, architectures, and OpenACC/OpenMP offload

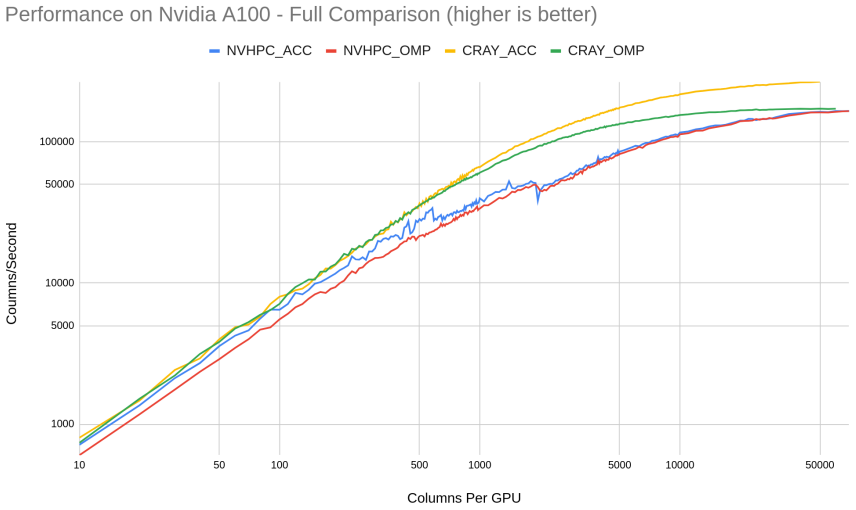
A100 results were collected on Derecho

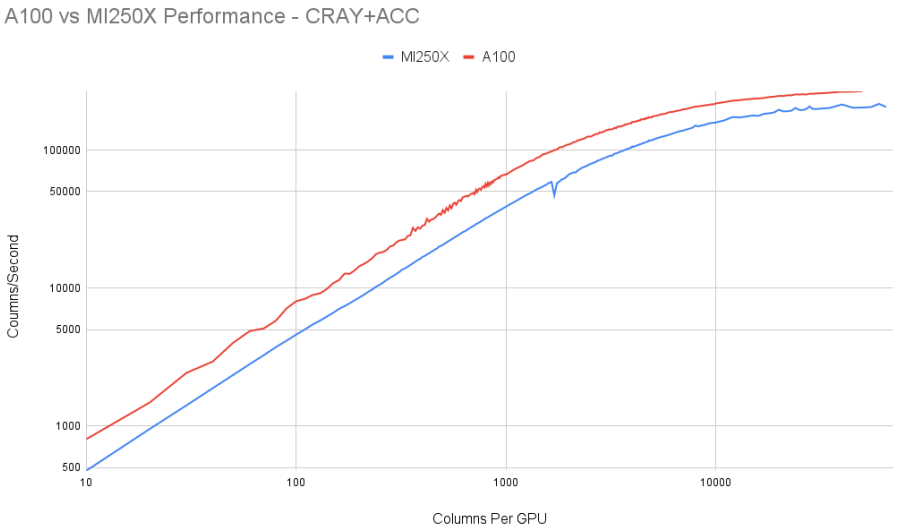
MI250X results were collected on Frontier

Findings after code was optimized for each compiler and architecture:

- Cray out performed NVHPC for both OpenACC/OpenMP offload. At 2K columns, Cray is 2x faster
- This *could* be because Cray seems to be using more GPU memory, nvhpc looks to be deallocating/allocating more frequently
- With the Cray/OpenACC combination, we see better performance on A100 than on MI250X

Results courtesy of Gunther Huebler





EarthWorks Release 2.2 (released yesterday)

- Multi-Platform Support
  - We have added support for the GH1, a Grace-Hopper system at the Texas Advanced Computing Center
  - GH1 Grace (CPU) testing has been performed for the FHS94, FKESSLER, and QPC6 (Aquaplanet) compsets only.
  - GH1 Hopper (GPU) offload testing has been performed for the FHS94 (Held-Suarez) test case only.

- Multi-Component GPU Offload
  - Includes MPAS dynamical core, PUMAS microphysics, and RRTMG-P radiative transfer physics code (CLUBB/GPU will be in next release)
  - Note: performance of the GPU offload has not been fully optimized

- Easier to run “out-of-the-box” model configurations

- More integrated testing

- Developers guide

Next Steps

- Optimize the memory movement in the CAM interfaces for RRTMGP and CLUBB
- Continue to improve performance at the ultra-high resolution scale on CPU/GPU
- Continue to flush out IO performance issues
- Study performance of fully GPU ported atmosphere model on Grace-Hopper

Questions?

Code Availability

https://github.com/EarthWorksOrg/EarthWorks

Contact information mickelso@ucar.edu

Thanks to the NSF CSSI program for their support

Thanks to TACC and the NCAR/CISL/CSG teams for their support

Thanks to the full EarthWorks team for their efforts and guidance

David Randall[1], James Hurrell[1], Donald Dazlich[1], Lantao Sun[1], Andrew Feder[1], William Skamarock[2], Andrew Gettelman[2], Brian Medieros[2], Xingying Huang[2], Sheri Voelz[2], Supreeth Suresh[2], Thomas Hauser[2], Ming Chen[2], Dylan Dickerson[2], Brian Dobbins[2], Michael Duda[2], Jim Edwards[2], Chris Fisher[2], Jihyeon Jang[2], Mariana Vertenstein[2], Richard Loft[3], Phil Jones[4], Luke Van Roeckel[4], John Cazes[5],Gunther Huebler[6]

[1]Colorado State University, [2]National Center for Atmospheric Research, [3]AreandDee, LLC, [4]Los Alamos National Laboratory, [5]University of Texas at Austin, [6]University of Wisconsin, Milwaukee