2021/6 Spring All-Hands Meeting

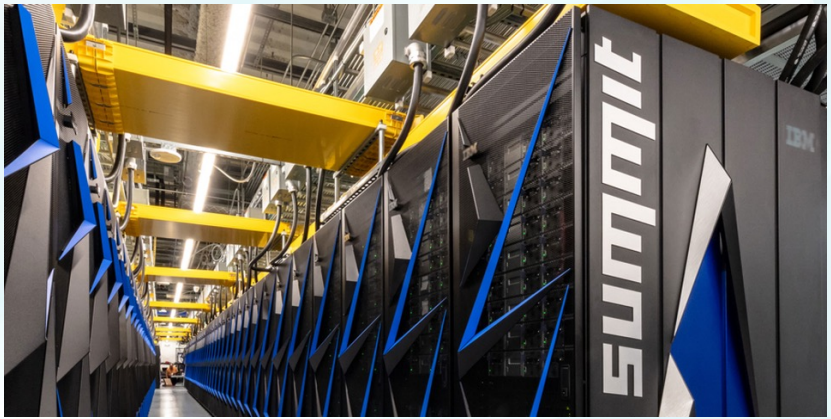
Day 1, Executive Committee Overview

Mark Taylor

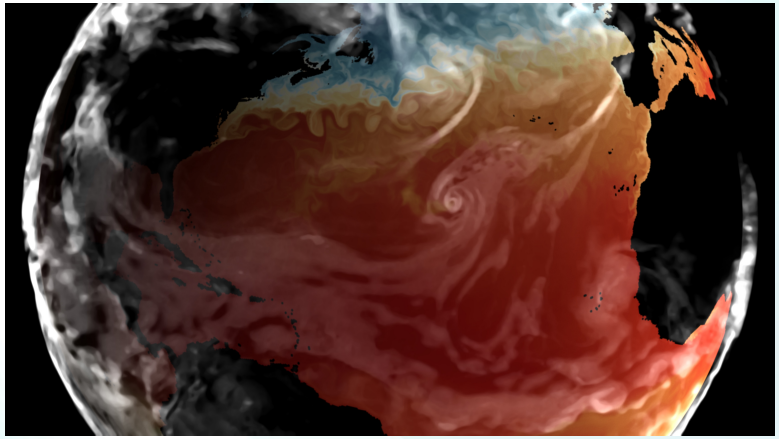
E3SM Exascale strategy: Running on GPUs

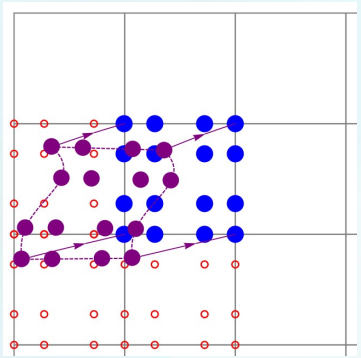
All new DOE SC machines will be GPU based:

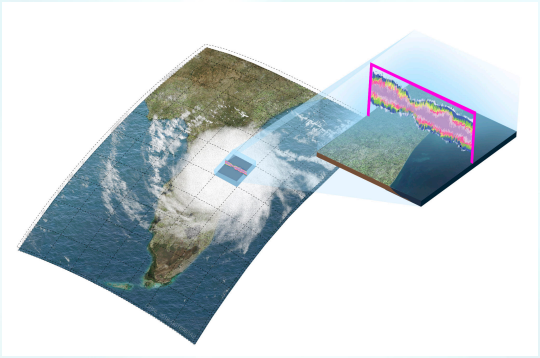
- 2021: NERSC Perlmutter
  - 6000 GPUs. 8 MW
  - 3000 CPU nodes (2022)
- 2022: OLCF Frontier
  - 1 CPU + 4 GPUs per node. 30 MW
- 2023: ALCF Aurora
  - Intel GPU nodes. ~40MW



E3SM Exascale Strategy







- **BER: E3SM Project**
- ~70 FTEs, 8 labs + Universities
- **E**nergy **E**xascale **E**arth **S**ystem **M**odel
- DOE-SC science mission: Energy & water issues looking out 40 years
- Ensure E3SM will run well on upcoming DOE exascale computers

- **ASCR/BER SciDAC**
- ~10 FTEs over multiple projects
- Large focus on new algorithms

- **ASCR ECP Project**
- ~10 FTEs
- E3SM-MMF: ”superparameterization”
- MPAS porting

Timeline and GPU Readiness

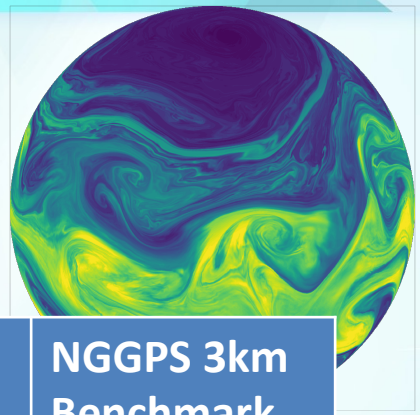
- E3SM Phase 2: Thru 2022
  - CPU: All major V2 simulation campaigns, Fortran code base
  - GPU: MMF (“super-parameterization”) atmosphere
  - GPU: SCREAM (prescribed aerosols) (late 2021)
- E3SM Phase 3: 2022-2025
  - CPU: All major V3 simulation campaigns, Fortran code base
  - GPU: MPAS components (Fortran/openMP)
  - GPU: ELM highres North America
- E3SM Phase 4: 2025-2028
  - Full Earth System Model running on both CPU and GPU systems

E3SM Dedicated Resources

V2/V3 model mostly requires CPU resources

- 2016: Anvil
  - 240 nodes Intel Broadwell
  - 0.05 MW
- 2019: Compy
  - 460 nodes, Intel Skylake
  - BER resource - 50% for E3SM
  - 0.2 MW
- 2020: Chrysalis
  - 512 nodes, AMD Epyc
  - 0.15 MW
  - E3SM V2 traditional resolution: 40 SYPD!
  - E3SM V1 high-resolution: 1 SYPD!

V2 Highlight: Scaling to all of Summit



- NGGPS 3km (cloud resolving) benchmark problem. Benchmark from the National Weather Service
- Atmosphere dycore with realistic configuration, 10 tracers and idealized physics
- Highlights from several generation of computers and dycores.
- SCREAM C++ dycore (HOMME-NH): 1.38 SYPD on Summit. Best performance we know of for a global dycore+tracers at 3km resolution

Model	Computer (Linpack rating)	NGGPS 3km Benchmark
NOAA FV3 2015	Edison (2.6PF)	0.16 SYPD
HOMME (CESM) 2017	TaihuLight (125 PF)	0.34 SYPD
HOMME++ 2021	Summit (200 PF)	1.38 SYPD

L. Bertagna et. al., SC '20: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2020

Path to V4: Single code base and GPU support

- Land Model
  - Not a performance bottleneck in coupled simulations
  - Evolve existing Fortran code based on science drivers
  - openACC for GPU acceleration of ultra-high resolution land simulations
- MPAS Ocean/Ice
  - Refactor MPAS Fortran framework code (ECP project)
  - openACC acceleration, transitioning to openMP for Frontier/Aurora
  - Phase 3 Ocean NGD will consider rewriting subcomponents in C++
- Atmosphere
  - Complete rewrite from scratch in C++

Path to V4: Single code base and GPU support

- V2 Atmosphere
  - Large Fortran code base, many gaps in E3SM expertise
  - During phase 2: decided not to pursue porting this code base to GPUs
- V4 Atmosphere
  - Rewrite from scratch in C++/kokkos
  - Emphasis on both GPU and CPU performance
  - Emphasis on low level testing for all subcomponents: unit tests, property tests, convergence tests

How do we get from V2 to V4?

Path to V4: Single code base and GPU support

Fortran Code Base

- V2 Atmosphere
  - Fortran code running on CPUs
  - NH dynamical core, SL transport
  - predominantly V1 physics
- V3 Atmosphere
  - Fortran code running on CPUs
  - V2 physics + NGD physics

Evolution to C++ code base

- SCREAM v0
  - Fortran code running on CPUs
  - V2 dynamics, V2 driver
  - SCREAM cloud resolving physics
- SCREAM v1
  - 100% C++: driver, dycore, physics
  - Prescribed aerosols
- V4 Atmosphere
  - Aerosols from EAGLES
  - V3 physics for climate simulations:
    - Rewrite in C++
  - Option: MMF physics from ECP

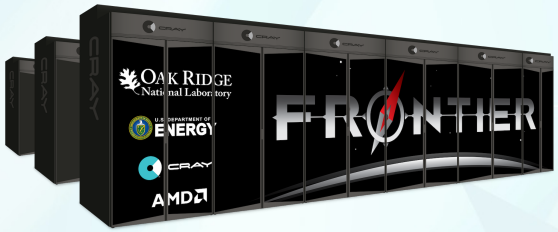
Backup Slides

V2 Exascale Highlights

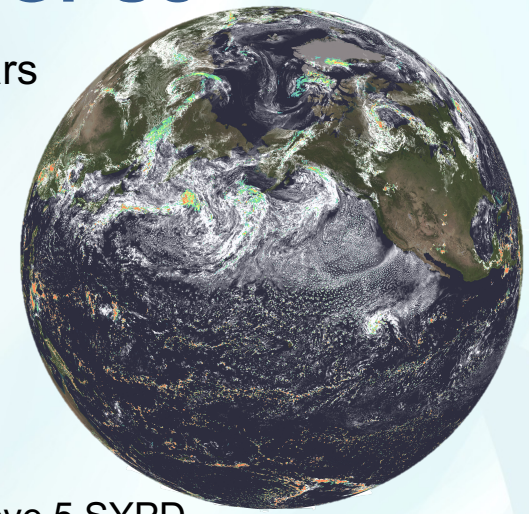
- Consolidated on two successful programming models
  - Fortran/openACC for existing modern Fortran code (MPAS)
  - C++/kokkos for new code, replacing legacy code
- Full understanding of GPU potential:
  - What runs well on GPUs, what should be run on CPUs
  - Guides our V3 and V4 strategy
- Progress on GPU ports of many subcomponents
  - Nonhydrostatic dycore, most physics, many MPAS subcomponents
  - Good understanding of the work needed for remaining subcomponents

Key Points for Earth System Models

- CPU performance (per watt) has nearly stagnated
  - 2x speedup over the last 6 years
- GPUs: 3x speedup (per watt) over today's CPUs
  - But only in the high-workload regime
  - Traditional IPCC style climate simulations run in low-workload regime
  - Need major code rewrite or refactor
- DOE Exascale era: ~80MW of GPU cycles
- E3SM phase 3: Focus on several new types of simulation campaigns where GPUs will allow us to run simulations not possible on CPU systems:
  - Large ensembles & cloud resolving simulations



High Workload Simulations for GPUs



- ✗ Earth System Model running at 5 SYPD for ~300 simulated years
  - A $5M commodity CPU cluster is most efficient
- ✓ **Ultra high resolution "Cloud Resolving" model**
  - BER: E3SM's "SCREAM" project
  - On track for E3SM V3 (2021-2022)
  - Typical INCITE award: 10-30 simulated year
- ✓ **Increase "local" complexity**
  - ECP: E3SM-MMF project: "superparameterization"
  - Achieve many aspects of a cloud resolving simulation and also achieve 5 SYPD
  - Running on Summit since OLCF Early Access Program
- ✓ **Large Ensembles**
  - Each ensemble member is running slower but more efficiently on GPU systems.
  - E3SM V4 capability: large ensembles on GPU systems

Performance Portability Strategy

- Large investments in two approaches
  - Both strategies require complete code refactor or rewrite, with careful coding to obtain competitive performance
  - Takes us 1–2 years per major component
- C++/Kokkos:
  - Rewriting from scratch allows us to take the opportunity to: replace legacy code, introduce low level unit and property tests
- Fortran/openMP
  - Vectorization (for CPUs) remains superior
  - Fortran GPU Support now lagging C++ substantially