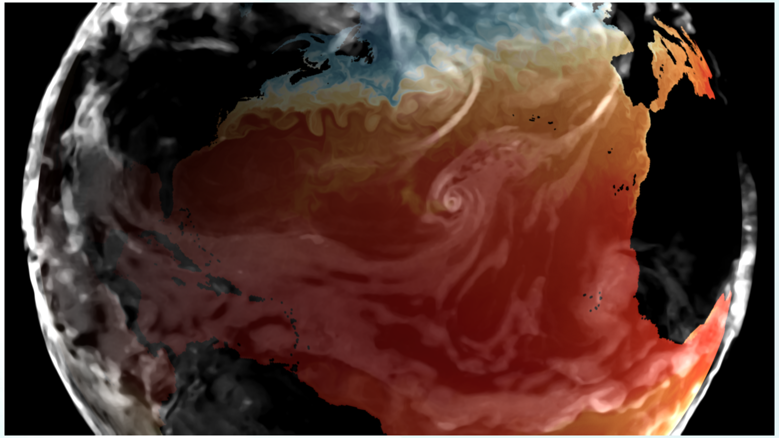
Overview of Phase II progress from Executive Committee

Day 1, Session 1, 11am-11:40am

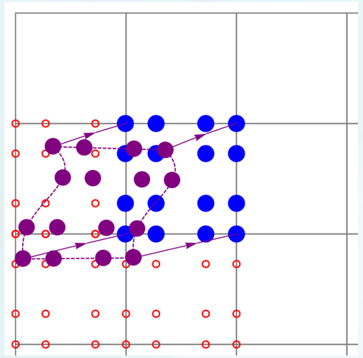
Guidance: In the overview, the presentation should elaborate on the specific progress that the project has made toward the goals in the Phase II proposal. In particular, what specific progress has the E3SM Phase II project made to:

3. Make E3SM GPU capable

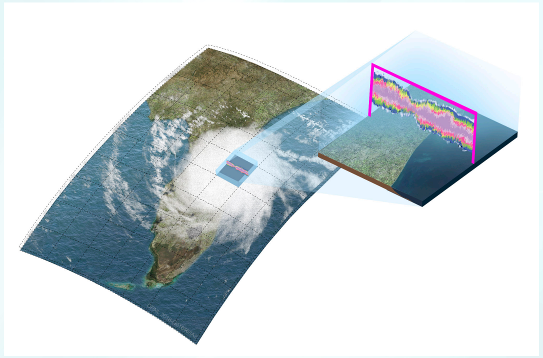
E3SM Exascale Strategy



- **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”

E3SM Exascale strategy: Running on GPUs

All new DOE SC machines will be GPU based:

- 2021: NERSC Perlmutter
  - 3000 CPU nodes, 6000 GPUs. 8 MW
- 2021/2022: OLCF Frontier
  - 4 GPUs per node. 30 MW
- 2022? ALCF Aurora
  - Intel GPU nodes. ~40MW

E3SM dedicated CPU resources:

- Anvil, Compy, Chrysalis
- 1200 CPU nodes, ~0.6 MW

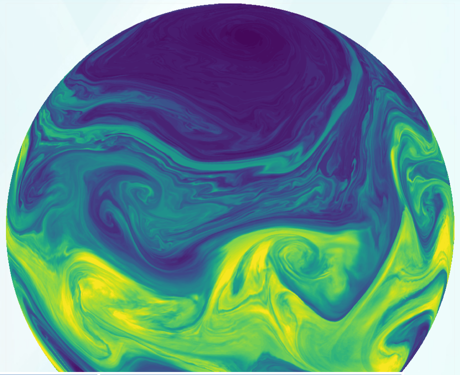
Timeline and GPU Readiness

- V2: 2019 code freeze, 2020 begin simulations
  - CPU: All major simulation campaigns, Fortran code base
  - GPU: NH atmosphere dycore.   partial: MPAS-O, ATM physics
- V3: 2022 code freeze, 2023 begin simulations
  - CPU: All major simulation campaigns, Fortran code base
  - GPU: MPAS components (Fortran/openMP)
  - GPU: SCREAM (prescribed aerosols) NH atmosphere
  - GPU: MMF ("super-parameterization") atmosphere
- V4: 2025 code freeze, 2026 begin simulations
  - Full Earth System Model running on both CPU and GPU systems

V2 Highlights

- Consolidated on two successful programming models
  - Fortran/openMP 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, some physics, MPAS (40%)
  - Good understanding of the work needed for remaining subcomponents

NGGPS cloud resolving (3km) benchmark Scaling to all of Summit



- Standardized benchmark from the National Weather Service
- Atmosphere model with realistic configuration and idealized physics
- Highlights from several generation of computers and Global cloud resolving models (GRCMs)
- Double precision results (reported real*4 results ~1.6x faster)

GCRM Model	Computer (Linpack rating)	NGGPS 3km Benchmark
NOAA FV3	Edison (2.6PF)	0.16 SYPD
HOMME (CESM)	TaihuLight (125 PF)	0.34 SYPD
E3SM's HOMMEXX_NH	Summit (200 PF)	0.97 SYPD (1.14 hydrostatic)

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

Phase 3 Vision

- Day 1, Session 4, 4pm-5pm

V2 Highlights

- Consolidated on two successful programming models
- **Full understanding of GPU potential:**
  - **What runs well on GPUs, what should be run on CPUs**
  - **Guides our V3 and V4 strategy**
- Partial progress on GPU ports of many subcomponents
  - Nonhydrostatic dycore, some physics, much of MPAS
  - Good understanding of the work needed for remaining subcomponents
- Continued work on CPU performance to support V2/V3 science campaigns

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
  - Consider openMP for GPU acceleration of sub 1km simulations
- MPAS Ocean/Ice
  - Relatively new code base, extensive E3SM expertise
  - 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
  - V2 code base, running on CPUs
  - V2 aerosols + NGD physics

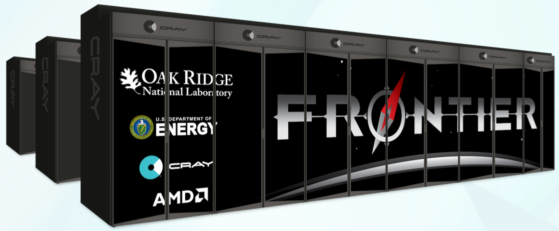
C++ code base

- SCREAM v0
  - V2 Fortran code running on CPUs
  - V2 dynamics
  - SCREAM cloud resolving physics
- SCREAM v1
  - 100% C++: driver, dycore, physics
  - Prescribed aerosols
- V4 Atmosphere (SCREAM v2)
  - Aerosols from EAGLES
  - V3 physics for climate simulations:
    - Rewrite in C++
  - Option: MMF physics from ECP

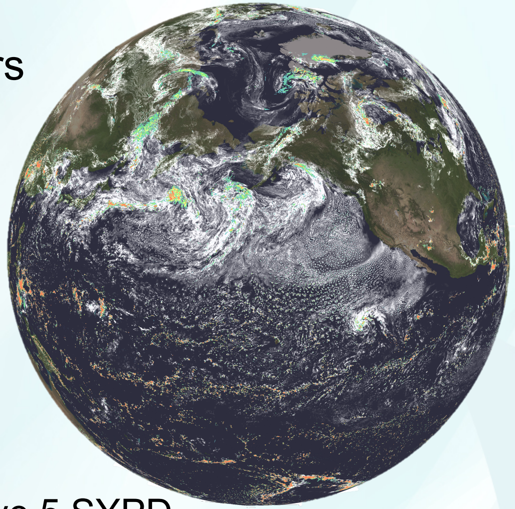
Backup Slides

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