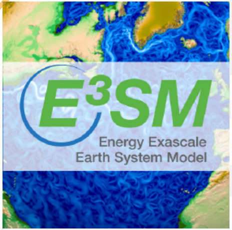
Sandia National Laboratories

Exceptional service in the national interest

E3SM C++ Atmosphere Model
Our journey to Exascale



L.Bertagna

September 28th, 2023

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND NO. 2023-10005PE

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

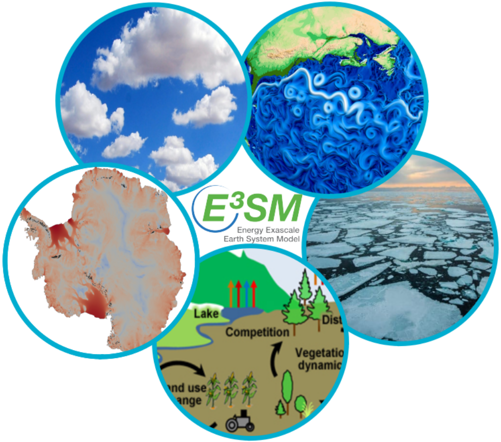
It takes a village to raise a portable model

SNL	Mark Taylor, Andrew Bradley, Oksana Guba, Micheal Deakin, Luca Bertagna, Daniel Sunderland, Jim Foucar, Conrad Clevenger, Ben Hillman, Andy Salinger, Irina Tezaur, Andrew Steyer, Brad Carvey, Chris Eldred, Pete Bosler, Jeff Johnson
LLNL	Peter Caldwell, Aaron Donahue, Chris Terai, Peter Bogenschutz, Hassan Beydoun, Chris Golaz, Finn Rebasso
LBNL	Noel Keen
ANL	Jayesh Krishna, Danqing Wu, Rob Jacob, Xingqiu Yuan
BNL	Wuyin Lin
UC	Paul Ullrich (Davis), Weiran Liu (Davis), Charlie Zender (Irvine)
PNNL	Balwinder Singh, Kyle Pressel
ORNL	Sarat Sreepathi, Matt Norman
HPE	Trey White

And more!

Energy Exascale Earth System Model (E3SM)

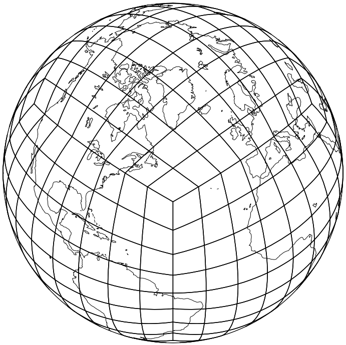
- DOE's state-of-the-science Earth system model
- Several components: atmosphere, land, ocean, land-ice, sea-ice, etc.
- Components can run at different resolutions.
- Broad variety of time/space scales.
- Mostly written in Fortran.
- Forked from CESM in 2014, contains snippets of codes written across several decades.



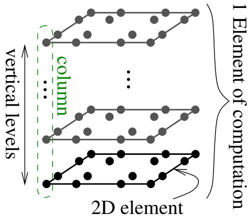
E3SM Atmosphere Model (EAM)

- Broadly speaking, divided in two parts: dynamics and physics
- Dynamics: solves Navier-Stokes equations in rotating spherical framework. It also solves for the transport of tracers in the atmosphere. E3SM uses High Order Methods Modeling Environment (HOMME, M.Taylor) package, which
  - decouples horizontal and vertical differential operators;
  - uses Spectral Element method in horizontal direction;
  - uses Eulerian or Lagrangian methods in vertical direction;
  - uses a Semi-Lagrangian method for tracers transport.
- Physics: approximates unresolved processes, contributing to forcing terms in dynamics equation. Examples include:
  - microphysics: water (vapor, liquid, ice) phase changes and precipitation;
  - macrophysics: subgrid cloud and turbulent processes;
  - radiation: radiative effects on atm temperature;
  - aerosols: cloud and radiative effects of transported particles.
- Original Fortran code highly performant on CPU-based HPC clusters.
- Typical production runs use $\sim$100km horizontal resolution, and 72 vertical layers.

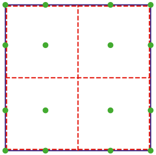
EAM discretization



(a) Example of cubed sphere quadrilateral mesh with ne=8.



(b) Structure of a vertically extruded 2d Spectral Element.

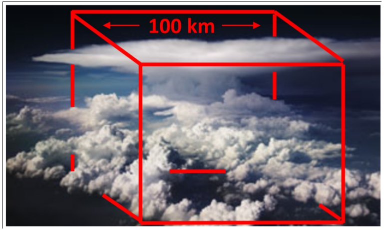


(c) overlap of dynamics (blue line, green dots) and physics (red lines) 2d computational element.

Toward cloud-resolving resolutions

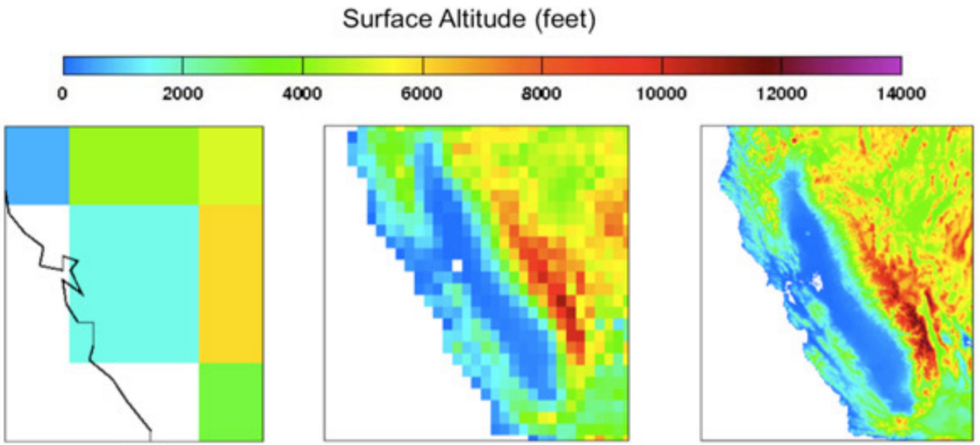
The push for km-scale resolutions in EAM was motivated by a few factors.

- Decision makers need reliable predictions, with reduced/quantifiable uncertainty. Increased resolution can help reduce uncertainty, by eliminating the need for certain subgrid approximations.
- Fine resolution is needed to capture extreme weather, which will be the leading source of climate-change impacts
- Improvements in HPC architectures and platforms made exascale simulation possible.



Note: 3.25km horizontal resolution achieved with ne=1024. With 128 vertical levels, this entails $\sim$7.2B degrees-of-freedom *per variable* on the dynamics grid.

Toward cloud-resolving resolutions



200km
Typical resolution of IPCC AR4 models

25km
Upper limit of climate models with cloud parameterizations

1km
Cloud system resolving models

Performance portability strategies

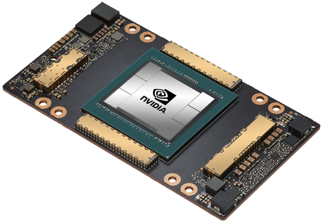
Performance Portability: capability of a code base to run "efficiently" on a variety of computer architectures. Three main approaches:

- **compiler directives**: hint/force compiler on how to optimize (OpenMP, OpenACC).
- **general purpose lib**: delegate architecture-specific choices to a library (Kokkos, Raja, etc.)
- **domain-specific lang/lib**: add intermediate compilation step, to generate optimal source for a given architecture (psyclone, gridtools, etc.).

Note: maintaining multiple versions of E3SM (one for each HPC architecture) is not viable approach. Performance portability **is a must** in the path to exascale.

The Kokkos programming model

- C++ library for on-node parallelism
- Provides constructs for expressing parallelism: execution space, execution policy, parallel operation.
- Provides constructs for multi-dimensional arrays: data type, memory space, layout, memory access/handling.
- Supports several back ends: OpenMP, Pthreads, Cuda, HIP, SYCL, etc.
- Very Reliable: large pool of (world-class) developers, heavily tested, countless downstream apps, closely follows new HPC architectures.



From EAM to EAMxx

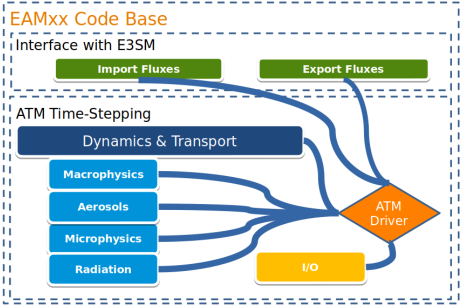
Guiding principles for EAMxx development

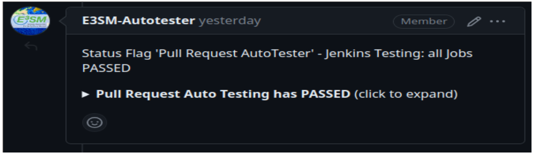
- Expose parallelism: use teams of threads to share common work and minimize index book-keeping.
- Maximize vectorization: use “pack” data structures and “masked” packed operations, to enhance vectorization (for CPU architectures).
- Minimize memory movement:
  - keep data on device, except for I/O or coupling with host-only components.
  - use (and share) minimally sized scratch memory within team parallel loops.

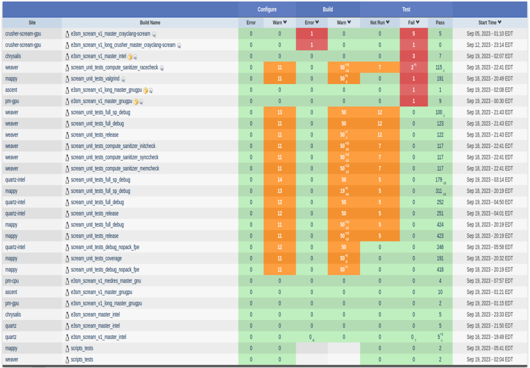
From EAM to EAMxx

Also a great chance to promote

- maintainable, extensible, and robust code design;
- a culture of strong testing and verification.







		Configure		Build		Test			
Site	Build Name	Error	Warn	Error	Warn	Not Run	Fail	Pass	Start Time
crusher-scream-gpu	e3sm_scream_v1_master_crayclang-scream	0	0	1	0	0	5	5	Sep 05, 2023 - 01:10 EDT
crusher-scream-gpu	e3sm_scream_v1_long_crusher_master_crayclang-scream	0	0	1	0	0	1	0	Sep 12, 2023 - 23:14 EDT
chrysalis	e3sm_scream_v1_master_intel	0	0	0	0	0	3	7	Sep 19, 2023 - 02:07 EDT
weaver	scream_unit_tests_compute_sanitizer_racecheck	0	11	0	50^{+13}_{-13}	7	2^{+1}	115_{-1}	Sep 18, 2023 - 22:41 EDT
mappy	scream_unit_tests_valgrind	0	11	0	50^{+5}_{-5}	0	1	191	Sep 18, 2023 - 20:49 EDT
ascent	e3sm_scream_v1_long_master_gnugpu	0	0	0	0	0	1	1	Sep 19, 2023 - 02:08 EDT
pm-gpu	e3sm_scream_v1_master_gnugpu	0	0	0	0	0	1	9	Sep 19, 2023 - 00:30 EDT
weaver	scream_unit_tests_full_sp_debug	0	13	0	50	12	0	100_{-1}	Sep 18, 2023 - 21:43 EDT
weaver	scream_unit_tests_full_debug	0	11	0	50	12	0	123	Sep 18, 2023 - 21:43 EDT
weaver	scream_unit_tests_release	0	11	0	50^{+7}_{-7}	12	0	122	Sep 18, 2023 - 21:43 EDT
weaver	scream_unit_tests_compute_sanitizer_initcheck	0	11	0	50^{+15}_{-15}	7	0	117	Sep 18, 2023 - 22:41 EDT
weaver	scream_unit_tests_compute_sanitizer_synccheck	0	11	0	50^{+13}_{-13}	7	0	117	Sep 18, 2023 - 22:41 EDT
weaver	scream_unit_tests_compute_sanitizer_memcheck	0	11	0	50^{+13}_{-13}	7	0	117	Sep 18, 2023 - 22:41 EDT
quartz-intel	scream_unit_tests_full_sp_debug	0	14	0	50	5	0	179_{-16}	Sep 19, 2023 - 03:14 EDT
mappy	scream_unit_tests_full_sp_debug	0	13	0	19^{+6}_{-43}	5	0	311_{-16}	Sep 18, 2023 - 20:19 EDT
quartz-intel	scream_unit_tests_full_debug	0	12	0	50	5	0	252	Sep 19, 2023 - 04:50 EDT
quartz-intel	scream_unit_tests_release	0	12	0	50	5	0	251	Sep 19, 2023 - 04:01 EDT
mappy	scream_unit_tests_full_debug	0	11	0	50^{+14}_{-14}	5	0	424	Sep 18, 2023 - 20:19 EDT
mappy	scream_unit_tests_release	0	11	0	50^{+13}_{-13}	5	0	423	Sep 18, 2023 - 20:19 EDT
quartz-intel	scream_unit_tests_debug_nopack_fpe	0	12	0	50	0	0	246	Sep 19, 2023 - 05:58 EDT
mappy	scream_unit_tests_coverage	0	11	0	50^{+5}_{-5}	0	0	191	Sep 18, 2023 - 20:32 EDT
mappy	scream_unit_tests_debug_nopack_fpe	0	11	0	50^{+1}	0	0	418	Sep 18, 2023 - 20:19 EDT
pm-cpu	e3sm_scream_v1_medres_master_gnu	0	0	0	0	0	0	4	Sep 19, 2023 - 07:57 EDT
ascent	e3sm_scream_v1_master_gnugpu	0	0	0	0	0	0	10	Sep 19, 2023 - 01:21 EDT
pm-gpu	e3sm_scream_v1_long_master_gnugpu	0	0	0	0	0	0	2	Sep 19, 2023 - 01:15 EDT
chrysalis	e3sm_scream_master_intel	0	0	0	0	0	0	5	Sep 18, 2023 - 23:33 EDT
quartz	e3sm_scream_master_intel	0	0	0	0	0	0	5	Sep 18, 2023 - 21:50 EDT
quartz	e3sm_scream_v1_master_intel	0	0	0_{-6}	0	0	0_{-7}	5^{+3}_{-1}	Sep 16, 2023 - 19:49 EDT
mappy	scripts_tests	0	0			0	0	2	Sep 19, 2023 - 05:41 EDT
weaver	scripts_tests	0	0			0	0	2	Sep 19, 2023 - 02:04 EDT

Items per page All

From EAM to EAMxx

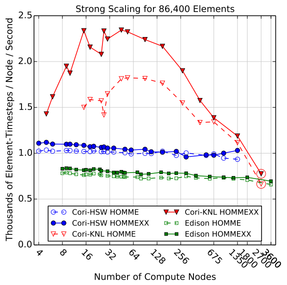
- Climate Modeling Development and Validation (CMDV), 2016-2019 (A.Salinger)
  - Demonstrated use of Kokkos for performance-portabily in dynamics (hydrostatic only)
  - Bertagna L., et al., “HOMMEXX 1.0: a performance-portable atmospheric dynamical core for the Energy Exascale Earth System Model“, Geoscientific Model Development, 2019
- E3SM Phase 2, Non-hydrostatic Atmosphere 2018-2022 (P.Caldwell)
  - Port non-hydrostatic dycore, and start porting of physics
  - Bertagna L., et al., “A performance-portable nonhydrostatic atmospheric dycore for the energy exascale earth system model running at cloud-resolving resolutions“, Sc20: International conference for high performance computing, networking, storage and analysis, 2020
- E3SM Phase 3, EAMxx, 2023-present (P.Caldwell)
  - Full atmosphere component running end-to-end on device (except I/O).
  - Taylor M, et. al., “The Simple Cloud-Resolving E3SM Atmosphere Model Running on the Frontier Exascale System“, ACM Gordon Bell Prize for Climate Modelling finalist, SC23

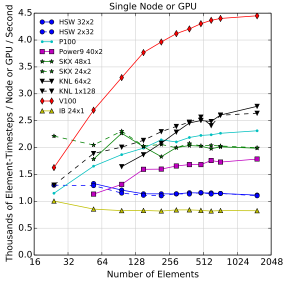
From EAM to EAMxx

- Climate Modeling Development and Validation (CMDV), 2016-2019 (A.Salinger)
  - Demonstrated use of Kokkos for performance-portabily in dynamics (hydrostatic only)
  - Bertagna L., et al., "HOMMEXX 1.0: a performance-portable atmospheric dynamical core for the Energy Exascale Earth System Model", Geoscientific Model Development, 2019
- E3SM Phase 2, Non-hydrostatic Atmosphere 2018-2022 (P.Caldwell)
  - Port non-hydrostatic dycore, and start porting of physics
  - Bertagna L., et al., "A performance-portable nonhydrostatic atmospheric dycore for the energy exascale earth system model running at cloud-resolving resolutions", Sc20: International conference for high performance computing, networking, storage and analysis, 2020
- E3SM Phase 3, EAMxx, 2023-present (P.Caldwell)
  - Full atmosphere component running end-to-end on device (except I/O).
  - Taylor M, et. al., "The Simple Cloud-Resolving E3SM Atmosphere Model Running on the Frontier Exascale System", ACM Gordon Bell Prize for Climate Modelling finalist, SC23

Performance portable hydrostatic dycore

- Established parallel execution patterns and data structures
- Achieved better-than-parity performance with original Fortran on CPU
- In-depth comparison of several architectures for single-node case
- Achieved bit-for-bit comparison with existing F90 (with some optimization disabled)



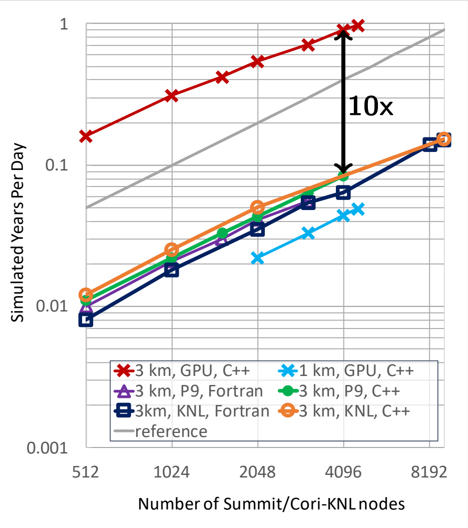


From EAM to EAMxx

- Climate Modeling Development and Validation (CMDV), 2016-2019 (A.Salinger)
  - Demonstrated use of Kokkos for performance-portabily in dynamics (hydrostatic only)
  - Bertagna L., et al., "HOMMEXX 1.0: a performance-portable atmospheric dynamical core for the Energy Exascale Earth System Model", Geoscientific Model Development, 2019
- E3SM Phase 2, Non-hydrostatic Atmosphere 2018-2022 (P.Caldwell)
  - Port non-hydrostatic dycore, and start porting of physics
  - Bertagna L., et al., "A performance-portable nonhydrostatic atmospheric dycore for the energy exascale earth system model running at cloud-resolving resolutions", Sc20: International conference for high performance computing, networking, storage and analysis, 2020
- E3SM Phase 3, EAMxx, 2023-present (P.Caldwell)
  - Full atmosphere component running end-to-end on device (except I/O).
  - Taylor M, et. al., "The Simple Cloud-Resolving E3SM Atmosphere Model Running on the Frontier Exascale System", ACM Gordon Bell Prize for Climate Modelling finalist, SC23

Performance portable non-hydrostatic dycore

- Built on success of hydrostatic dycore work
- Run on full summit system with ne=1024 (3km) and ne=3096 (1km) horizontal resolutions
- Achieved 0.97 Simulated Years Per Day (SYPD) when running on GPU on the whole Summit system
- On Summit, using GPUs gives 10x speedup over CPUs at scale

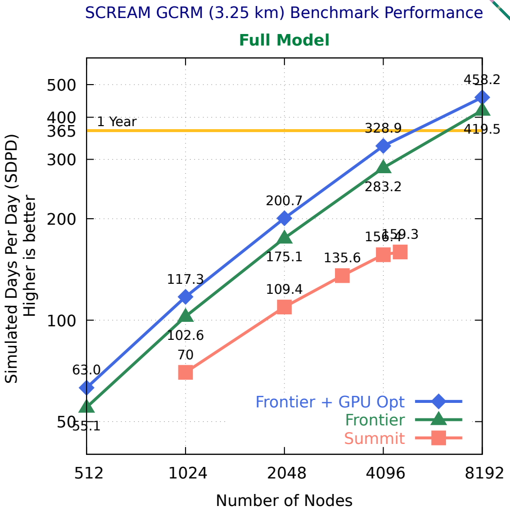


From EAM to EAMxx

- Climate Modeling Development and Validation (CMDV), 2016-2019 (A.Salinger)
  - Demonstrated use of Kokkos for performance-portabily in dynamics (hydrostatic only)
  - Bertagna L., et al., "HOMMEXX 1.0: a performance-portable atmospheric dynamical core for the Energy Exascale Earth System Model", Geoscientific Model Development, 2019
- E3SM Phase 2, Non-hydrostatic Atmosphere 2018-2022 (P.Caldwell)
  - Port non-hydrostatic dycore, and start porting of physics
  - Bertagna L., et al., "A performance-portable nonhydrostatic atmospheric dycore for the energy exascale earth system model running at cloud-resolving resolutions", Sc20: International conference for high performance computing, networking, storage and analysis, 2020
- E3SM Phase 3, EAMxx, 2023-present (P.Caldwell)
  - Full atmosphere component running end-to-end on device (except I/O).
  - Taylor M, et. al., "The Simple Cloud-Resolving E3SM Atmosphere Model Running on the Frontier Exascale System", ACM Gordon Bell Prize for Climate Modelling finalist, SC23

Simple Cloud-Resolving E3SM Atmosphere Model

- Rewrite all atm physics in C++, using Kokkos.
- Highly modular, easy to change/rearrange physics paremetrizations.
- First Global Cloud-Resolving Model to run on an Exascale computer, break the 1 SYPD barrier at cloud resolving resolutions, and run on both AMD and NVIDIA GPUs (as well as conventional CPUs).
- Finalist for the 2023 ACM Gordon Bell Prize in Climate Modeling



Power Consumption

- How to compare CPU vs GPU performance?
  - SYPD per machine purchase cost?
  - SYPD per machine operating costs?
- Exact power consumption of bleeding-edge archs not always clear.
- Based on TDP, 1GPU node uses 4-8x more power than 1CPU node
- On Perlmutter, actual power consumption monitored during SCREAM run showed GPU uses $\sim$1.7x more power than CPU.
- Since PM-GPU is 5.8x faster than PM-CPU, we conclude GPU is 3.5x faster *per Watt*.

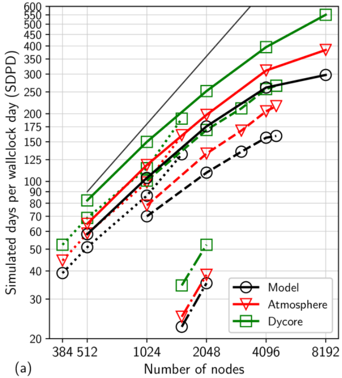


Figure: 3km/128L simulation; solid: Frontier (4 MI250/node), dashed: Summit (6 V100/node); dotted: Perlmutter-GPU (4 A100/node); dash-dot: Perlmutter-CPU (2 Epyc/node).

How is EAMxx used for science?

- Ongoing: hindcasts/storylines for various historical weather events
- Ongoing: doubly-periodic and regionally-refined runs to understand/fix biases
- Completed: 40 day simulations for all 4 seasons
- Imminent: 12 month simulations with current and +4K sea surface temperature (SST) to understand climate sensitivity
- Soon: 13 months nudged simulations with current and pre-industrial aerosols forcing to understand aerosol sensitivity
- Next year: 10-20 year fixed-but-varying SST runs for current and future conditions
- 2025: 1 decade of coupled k-scale simulations

How is EAMxx used for science?

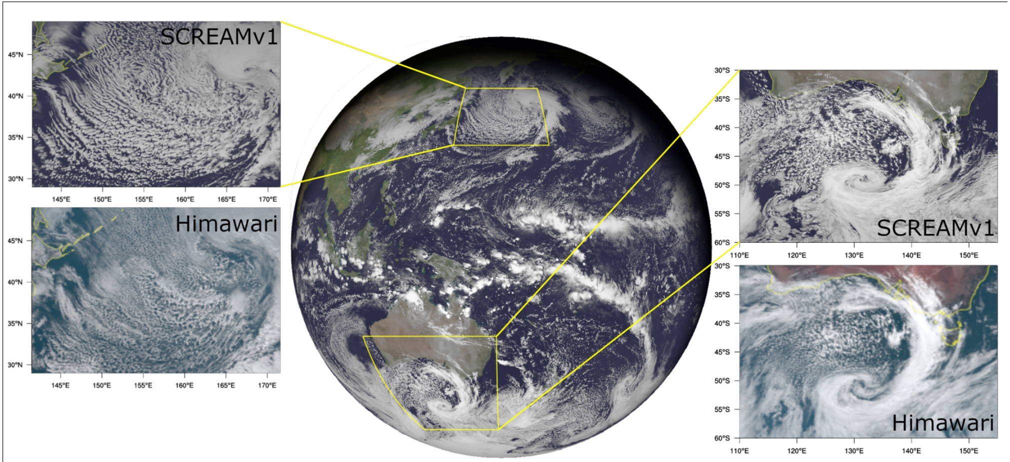


Image from a 40 day SCREAM DYAMOND simulation. Shortwave cloud radiative flux compared to Himawari visible satellite image two days into the simulation (January 22, 2020 at 2:00:00 UTC).

Contributions from other ASCR projects

Several pieces of EAMxx were developed by computational scientists funded by ASCR in collaboration with BER

- Scorpio, a library for (massively) parallel IO (ECP)
- Semi-Lagrangian transport (COMPOSE SciDAC)
- RRTMGP++, the radiation package for EAMxx (ECP)
- Finite volume grid for physics parametrizations (ECP, COMPOSE SciDAC)
- Non-hydrostatic dycore (Non-hydrostatic SciDAC)
- Regionally Refined Meshes (Multiscale SciDAC)

Conclusions

- With EAMxx we kept faith to the 'Exascale' part of the E3SM name.
- EAMxx established itself as a reference point for other atmosphere code bases when it comes to exascale performance.
- Initial study of performance-portable implementation strategy was time consuming, but lessons learned considerably sped up subsequent work.
- Kokkos efficiently maps to current architectures, as well as reduces disruption from future ones.
- C++ syntax is *much* heavier than Fortran, which can be a barrier for field scientists. Good software design is crucial to separate concerns, increase productivity, and make code maintenance manageable.
- C++ with performance-portable layers is also used in the E3SM land-ice component (MALI, SciDAC), and in the next-generation E3SM ocean component (Omega).