Porting The Community Atmosphere Model – Spectral Element (CAM-SE) To Hybrid GPU Platforms

Matthew Norman (ORNL)

Richard Archibald (ORNL)

Jeffrey Larkin (Cray)

Valentine Anantharaj (ORNL)

Ilene Carpenter (NREL)

Paulius Micikevicius (Nvidia)

What is CAM-SE?

- Climate-scale atmospheric simulation for capability computing
  - Decades to centuries of global simulation at high resolution
  - Utilize up to 200,000 cores
- Maintained and developed by many institutions
- Comprised of (1) a dynamical core and (2) physics packages

1. Dynamical core

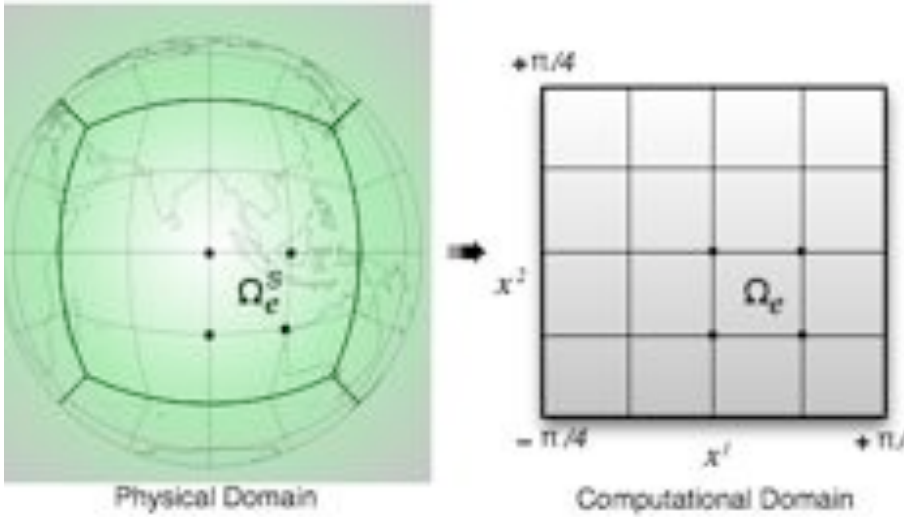
   (a) "Dynamics": Solve for wind, energy, & mass
   (b) Transport "tracers" (water vapor, CO2, O3, etc)

2. Physics packages: Resolve physical phenomena not included in dynamical core (moist convection, radiation, chemistry, etc)

- We used CUDA Fortran for our port

OAK RIDGE National Laboratory

What is CAM-SE?

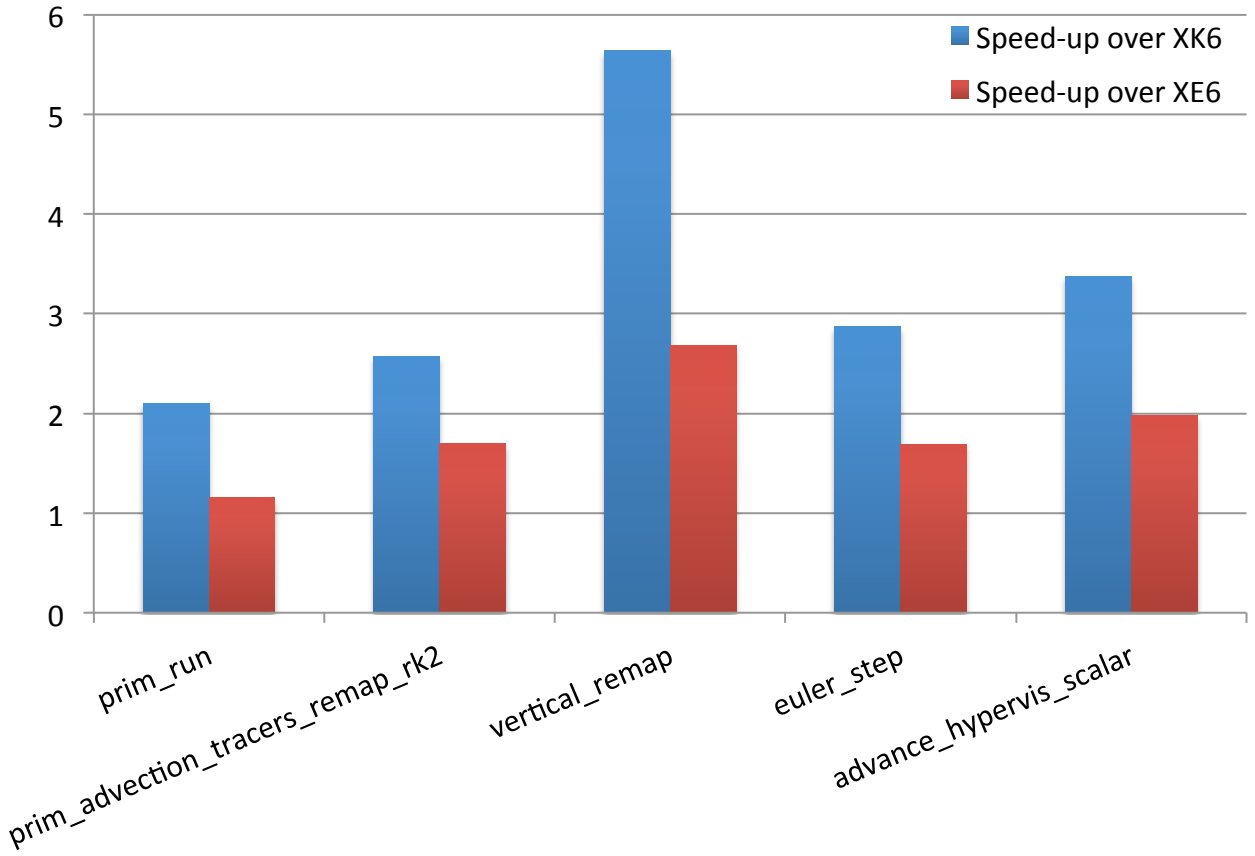


Courtesy of Ram Nair
http://www.image.ucar.edu/staff/rnair/research09.html

- Cubed-Sphere grid
- Each face divided into elements
- Elements spanned by 4x4 nodal basis functions
- Only nearest-neighbor comms. required between elements
- 26 Vertical Levels

- Global 14km configuration
  - 240 x 240 elements per panel, 4 x 4 basis functions per element
  - Strong scales to 172,800 XT5 cores with 60% parallel efficiency
  - Target is 64 columns of elements per compute node
  - With Mozart chemistry, runs at 0.25 simulated years per day on XT5
  - About 1.2 billion degrees of freedom total

Current Performance Status



- 96 x 96 elements per panel, 864 nodes, 64 elements per node
- Only prim_advection_tracers_remap_rk2 ported so far
- Expect 1.75x improvement over XE6 node with Fermi GPUs

OAK RIDGE National Laboratory

Challenges Unique to CAM-SE

- Throughput requirement (1–5 simulated years per day, SYPD)
- Time-explicit simulation
  - Elements per node must decrease with grid refinement
- Typical available threading per-GPU:
  - 8 x 8 elements per GPU, 4 x 4 bases per element, 26 vertical levels
  - 26,624 threads available when vertical threading possible
  - 1,024 threads when vertical threading impossible (e.g., physics)
- Why add tracers?
  - Roughly 3 million threads per GPU
  - Puts a spike in the profile to port

OAK RIDGE National Laboratory

Codebase Challenges

- Eventually, we desire use of directives
  - CUDA Fortran suitable for key kernels, but not sustainable in general
- Even directives require extensive code changes (at first)
  - "It is often wise to represent an array of structures as a structure of arrays" <*http://developer.nvidia.com/content/openacc-directives-gpus*>
  - Strided and irregular memory accesses must be contained
  - Logic which diverges over fastest varying loop indices is bad
- A single instruction stream makes a difference

Cache Awareness

```
do s = 1 , 3
  coefs(s,i,j,k,q,ie) = …
enddo
```

Thread Awareness: Block over i,j,k

```
do s = 1 , 3
  coefs(i,j,k,s,q,ie) = …
enddo
```

- L1 cache unpredictable, use shared memory when possible

OAK RIDGE National Laboratory

Future Challenges

- Must increase data-parallel work without reducing time step
  - More resolution, more uncertainty, more ensembles?
  - Use capability to allow more tracers
- Ultimately, we need new relaxed-time-step methods
  - Typically, added moments (i.e., p-refinement) are data-parallel
  - CAM-SE: time step reduces quadratically with added moments
  - Multi-Moment, Finite-Volume methods show some promise
- Interacting with the user community
  - Single precision in the dynamical core? In the physics?
  - Increased number of vertical levels
  - Run dynamics and physics in parallel (slightly loosen coupling)

OAK RIDGE National Laboratory