Massively Parallel Performance of the HOMME Spectral Element Atmosphere Model

The 8th International Workshop on Next Generation Climate Models for Advanced High-Performance Computing Facilities

February 24, 2006
Bill Spotz
Mark Taylor
Sandia National Laboratories

In collaboration with NCAR Scientific Computing Division
A Little Background…

- Sandia has a long history as a leader in advanced, high-performance computing
- A significant percentage of Sandia’s scientific applications are based on finite elements
- Historically, Sandia computing has focused on engineering-related problems and has had only minimal involvement in climate modeling
- I joined Sandia in 2001 after 5+ years at NCAR (atmospheric dynamical cores)
- Mark Taylor joined Sandia in 2004 after 5+ years at LANL (ocean modeling) and 5+ at NCAR (SEAM)
- HOMME/SEAM element-based numerical method and scalability made it attractive for a collaborative effort
SEAM: Spectral Element Atmosphere Model

- Atmospheric global circulation model (GCM)
  - NCAR/DOE CHAMMP
- Quadrilateral spectral elements in horizontal
- Finite differences in vertical
- Cubed sphere, 2D domain decomposition
- 6 transforms between spherical and local cartesian coordinates
  - No pole problem
  - Edge conditions
- 2001 Gordon Bell Honorable Mention
Atmospheric Dynamical Cores

- Spherical Harmonics
- Accuracy (wave propagation)
- Local Conservation (moisture advection)
- Parallel Scalability
- Finite Volumes

SEAM
• **Accuracy**: can achieve same accuracy as spherical harmonic models.

• **High order** representation allows for high order scale selective dissipation (like hyper viscosity used in S.H.)

• **Unstructured Grid**: Can handle AMR

• **Unstructured Grid**: No pole problem, so excellent parallel scalability

• **Unstructured Grid**: New challenges for existing physics parameterizations?

• **Local Conservation**: Less oscillatory than S.H., but does not have exact local conservation (DG?)
Why do Spectral Elements Scale Well?

- Scalability depends on doing enough computational work relative to communication.
- For fixed-resolution problems, this is always a losing battle: as processors $\uparrow$, communication $\uparrow$, computational work $\downarrow$.
- Even as spectral elements approach their granularity limit, they still represent a reasonable amount of computational work.
- 1 mat-vec represents $64 \times (64+63) = 8128$ FLOPS/element.
- Equivalent FV patch: $64 \times (9+8) = 1088$ FLOPS/patch.
How Well Does SEAM Scale?

How Well Does SEAM Scale?

BlueGene/L

Parallel Scalability

MFLOPS per CPU

NCPU

1 16 256 4096 64K

How Well Does SEAM Scale?

Parallel Scalability

Red Storm

MFLOPS per CPU

NCPU

- 156km L26, 384 elements
- 40km L50, 6144 elements
- 20km L70, 24576 elements
- 10km L100, 98304 elements
Integration Rates

• Atmosphere For Earth Simulator (AFES)
  – Global spectral model (spherical harmonics: Legendre transforms, all-to-all transpositions)
  – Full physics
  – 10km (24TF) 57 sim days/day

• Red Storm (SEAM)
  – Spectral elements: local computations and communications
  – Aquaplanet (reduced physics)
  – 40km (3TF) 7-30 sim years/day
  – 10km (5TF) 32-128 sim days/day
Why the Range of Integration Rates?

• Explicit vs. Semi-implicit time stepping
• Explicit (lower bound):
  – Small $\Delta t$ for stability
  – Efficient numerical kernel
• Semi-implicit (upper bound…):
  – Larger $\Delta t$ for stability ($\sim x8$)
  – Helmholtz solve: communication required every iteration
  – How many iterations?
• Research questions:
  – What is the best algebraic preconditioner for spectral elements?
  – What is the optimal $\Delta t$?
  – What is the best integration rate we can obtain?
Current Semi-Implicit Scheme

• 3D primitive equations
• Hydrostatic assumption
• Loosely coupled “stack” of shallow water equations
• Eigenmode decomposition in vertical direction (split semi-implicit)
  – For $L$ levels, this is an $L \times L$ eigensystem
  – Results in $L$ independent 2D systems to solve
  – Most energy & linear system solver work occurs in lowest eigenmodes
  – Decomposition is pre-processing step
  – Transforms: physical space $\Leftrightarrow$ eigenspace
  – Quantity of interest: post-spinup average iterations for each eigenmode (function of resolution)
SEAM Split Semi-Implicit Iterations

Modal Iterations, N=8 (180 km)

Simulation Time (Days)

Iterations

- Mode 0 (red)
- Mode 1 (green)
- Mode 2 (blue)
- Mode 3 (cyan)
- Mode 4 (magenta)
Description of Numerical Experiment

• For each resolution of interest
  – Run model until it reaches “solver equilibrium” (>100 days)
  – Obtain iteration profile (# iterations vs. eigenmode)
  – Hard-code the solver algorithm to perform these specified numbers of iterations (ignoring convergence)

• For each processor count of interest
  – Run short (5 time step) performance experiments for semi-implicit and explicit
  – Compute ratio of integration rates: semi-implicit/explicit
SEAM Split Semi-Implicit Acceleration

![Graph showing Acceleration Rates vs Number of Processors]

- **Ne = 5 (256 kn)**: Red line
- **Ne = 8 (160 kn)**: Green line
- **Ne = 16 (80 kn)**: Blue line
- **Break Even**: Purple line

The graph illustrates the acceleration rates for different numbers of processors, showing how the rates change as the number of processors increases.
Red Storm Demonstration Run

• Polar vortex problem: Strong circumpolar jet that traps air over the poles

• Numerical Statistics
  – 13km grid spacing, 300 levels in the vertical (1 billion grid points)
  – Integrated for 288,000 time steps using 7200 CPUs for 36 hours
  – Produced 1TB of data
Polar Vortex on Red Storm
Red Storm Demonstration Run

Isosurface and contours of potential vorticity over north pole
Spectral Elements in Triangles

• Quadrilateral spectral elements
  – Choice of tensor-product Gauss-Lobatto points for nodal basis and quadrature leads to diagonal mass matrix and excellent interpolation properties

• Triangular spectral elements
  – Choice of nodal grid points is much more complicated . . . hard to get good interpolation and quadrature simultaneously

• Taylor: perform numerical optimization to look for suitable points