16th Annual CESM Workshop 7/23/2011

OLCF
Oak Ridge Leadership Computing Facility

Center for Accelerated Application Research(CAAR)

Application Readiness Team	
Rick Archibald	ORNL
Jeff Larkin	Cray
Ilene Carpenter	NREL
Paulius Micikevicius	Nvidia
Matthew Norman	ORNL
Valentine Anantharaj	ORNL

PROGRESS TOWARDS ACCELERATING CAM-SE ON HYBRID MULTI-CORE SYSTEMS

Presented by: Rick Archibald

Necessity of Hybrid Multicore Systems

ORNL Supercomputing Path:

- Today: Two Petaflop system with thousands of multicore nodes that uses 6MW
- 2018: 1000X speed-up in flops (Exaflop), 10X-100X increase in hybrid nodes and only 3X increase in power

GPU Properties:

- High flop per watt
- Reliability at scale
- Leverage commodity
- GPU augments CPU
- Complex memory hierarchy
- Boost parallelizm

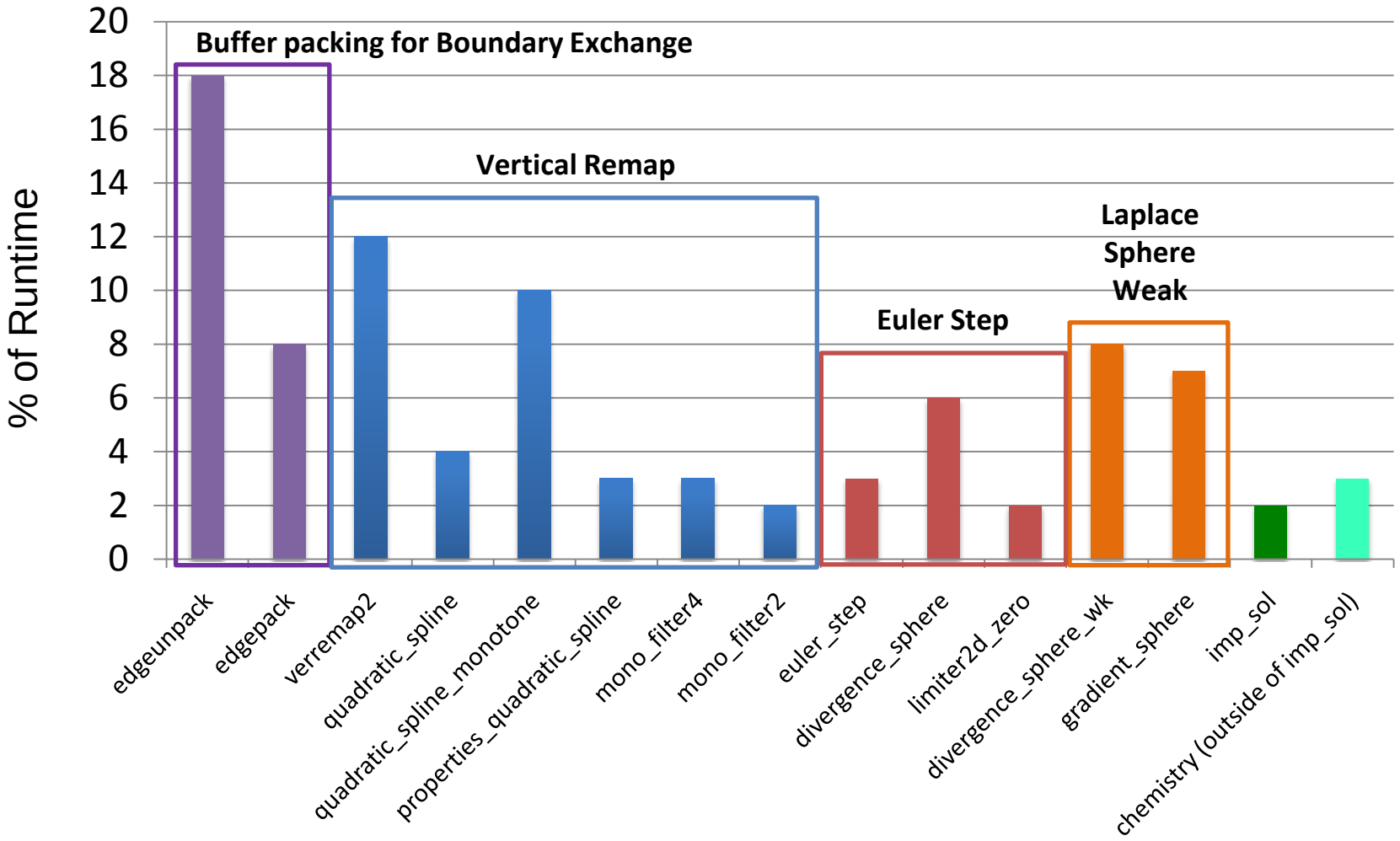
Target Problem

- 1/8 degree CAM, using CAM-SE dynamical core and Mozart tropospheric chemistry. Land model will be run on CPUs.
- Why is acceleration needed to “do” the problem?
  - Mozart atmospheric chemistry is too expensive to include in high resolution models on today’s systems.
- What unrealized parallelism needs to be exposed?
  - The most time consuming part of the Mozart chemistry is the implicit solver. Currently, parallelism is over columns of grid points, as it is in the rest of the “physics”. Parallelism over vertical levels within a column of grid points needs to be exploited in this phase of the computation.
  - In many parts of the dynamics, parallelism needs to include levels and chemical constituents.

Profile Generation

- The science problem of interest cannot be run today because Mozart chemistry uses a latitude/longitude grid and CAM-SE uses a cubed-sphere grid.
  - Work is underway in the community to remove the dependence on lat/lon grids.
- A set of jobs that span the aspects of this problem have been run, which allowed us to make a performance model for the target problem:
  - ¼ deg Finite Volume CAM with Mozart chemistry
  - 1/8 deg CAM-SE with 3 advected constituents
  - Stand-alone CAM-SE with 3 and 101 advected constituents (the number we'd have if we could run Mozart chemistry with HOMME)

Profile of Runtime



Kernels extracted

- Vertical remapping
- Euler_step + limiter2d_zero
- Laplace_sphere_wk
- Small kernels from dynamics (gradient, divergence, vorticity etc.), used by euler_step and laplace_sphere_wk
- Tendency physics before coupling (for compiler analysis, not run-able).
- Chemistry implicit solver (smaller than expected % of final runtime, further development put on hold)

Method of acceleration

- CUDA
  - Use CUDA Fortran for HOMME
- Directives
  - Use directives for CAM physics
- Libraries
  - No significant math library use
- Explore what works best, what is needed for atmospheric chemistry (Mozart).

Method of acceleration: Why?

- CAM-SE dynamical core has the same loop and index patterns appear repeatedly, which makes CUDA work tractable.
- Mozart chemistry code developers are likely to accommodate code changes. Both CUDA and directives will be explored. Implicit solver code is generated by a pre-processor, so GPU-specific code can be generated without affecting CPU version.
- Physics code will need to accelerated using directives
  - to prevent version bifurcation, involves many routines written by many different people who have no knowledge accelerators
  - profile for the physics is extremely flat
  - physics is less than 1% of target job with current (un-optimized) vertical remapping routines
  - Directives are the only method that potentially allow one to use GPUs without causing changes in the CPU base code.

Plan of work – what has been done?

- Examination of code structure and data types. The code is about a million lines and the profile is relatively flat so this has been a very time consuming process.
- Created test cases for profiling and performance model development.
- Created macro-kernels for some of the most time consuming parts of the code to enable people unfamiliar with CESM to assist with project.
- Created low level kernels from CAM-SE, converted to CUDA and to use accelerator directives with PGI and HMPP compilers.

Plan of work – what remains to be done?

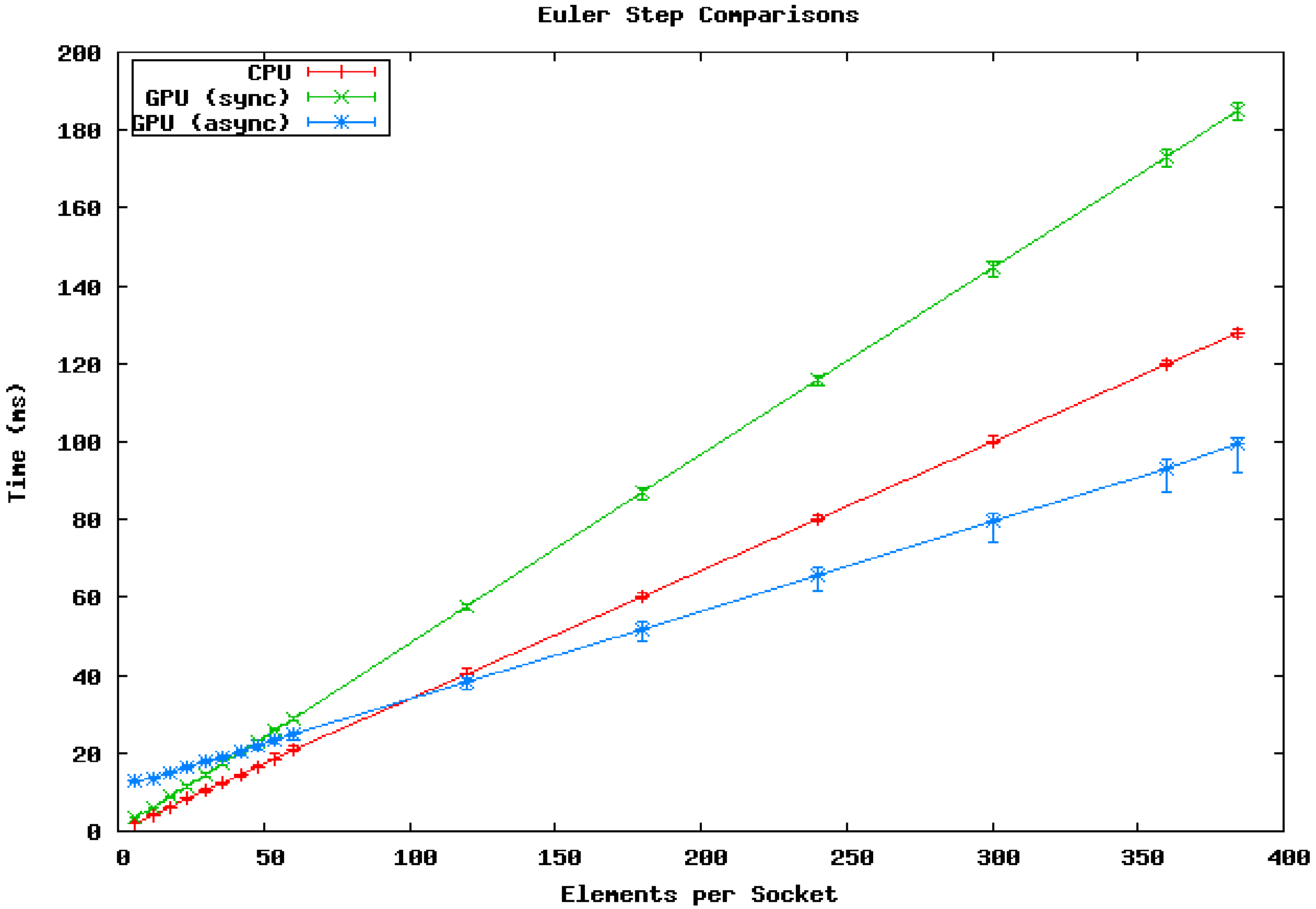
- Continue to refine profiles as inefficient code is rewritten (vertical remap, chemistry implicit solver).
- Examine performance of buffer packing and unpacking routines at scale.
  - Gain experience with new CAM-SE version with OpenMP hybrid parallelism for buffer packing/unpacking.
  - Explore doing packing/unpacking for boundary exchange on GPU.
- Continue CUDA work in CAM-SE
  - restructure code as needed to work around compiler limitations to create higher level kernels
  - continue up the dynamics call stack to run more on the accelerator and reduce data transferred between CPU and GPU
  - use CUDA streams in tracer advection to overlap data transfer and computation
  - Integrate GPU kernels into CAM (in CESM)

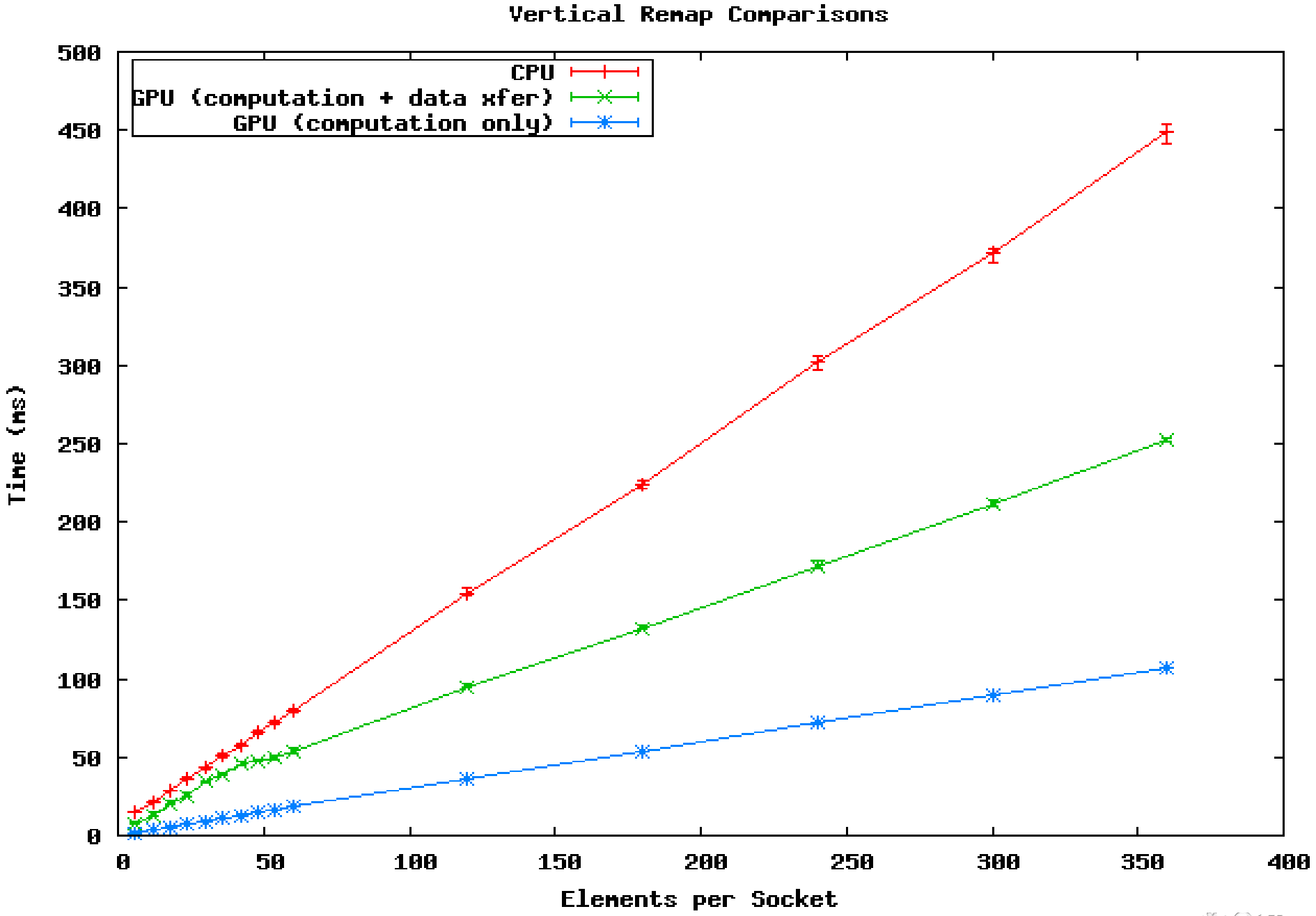
CPU code:

```
Loop over elements
  Loop over advected constituents (q)
    Loop over j
      Loop over i
        Loops over levels
          all computations from quadratic_spline and
          quadratic_spline_monotone manually inlined
```

GPU code:

```
Transfer element-independent data to GPU
Create chunks of 6 elements

Decompose work into thread blocks:
 tblock=dim3(nv,nv,6) – columns associated with 6 elements
 tgrid=dim3(qsize,1,1)  - each block has a different value of q

Loop over chunks:
 Copy data from element structure to local arrays
 Transfer data to GPU
 Call remap_Q_kernel<<<tgrid,tblock>>>(…)
 Transfer results back to CPU
 Copy data back to elem structure

attributes(global) subroutine &
      remap_Q_kernel(Qdp,hyai,hybi,ps_v,dpdn,ps0,dt)

  i  = threadIdx%x
  j  = threadIdx%y
  ie = threadIdx%z
  q  = blockIdx%x

 ! each thread has one column of points for one value of q

   Loops over levels

     all computations from quadratic_splime and
     quadratic_spline_monotone manually inlined

end subroutine remap_Q_kernel
```





Summary

- The community is moving towards high-resolution modeling with atmospheric chemistry
- We have demonstrated that such a run is dominated by dynamics calculation
- We have demonstrated that the dynamics phase is capable of running efficiently on an accelerator and given a plan to proceed forward

Additional help from

Jim Rosinski, Mark Taylor, John Dennis, Kate Evans, Oscar Hernandez, Jim Schwarzmeier, Tom Court, Abdulla Bataineh