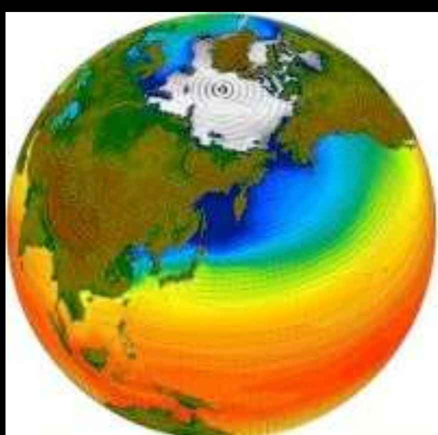
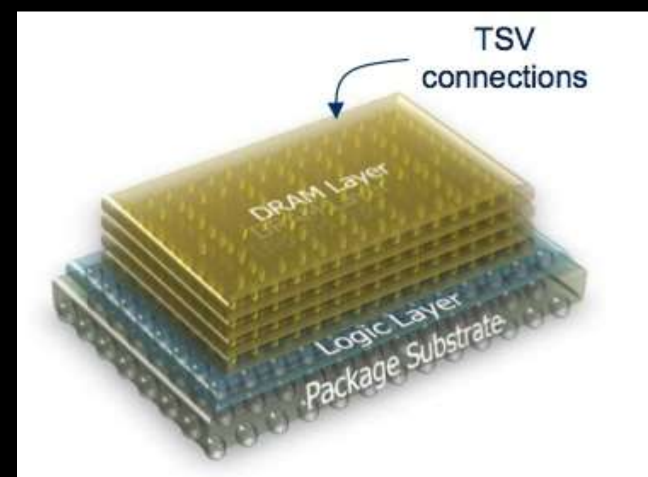
Refactoring Climate Models for the Exascale

Dr. Richard D. Loft
loft@ucar.edu
Computational and Information Systems Laboratory
National Center for Atmospheric Research



Earth System Model



Future architecture:
massively parallel and complex

NCAR NSF

September 9, 2015

Outline

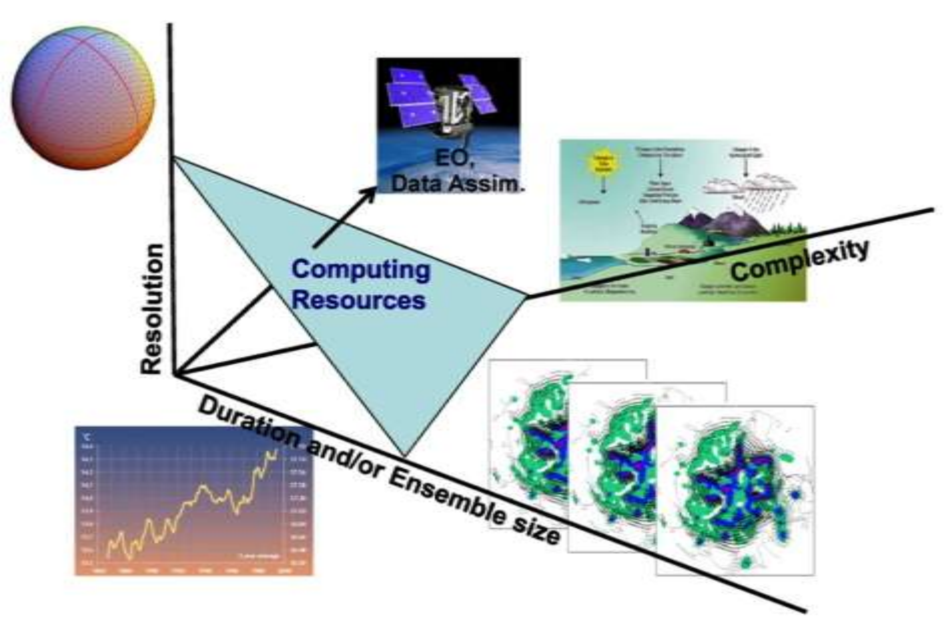
- What climate experiments need exascale?
- The performance gap
- A case study: more peak flops won’t help
- Performance refactoring 1.5 million SLOC

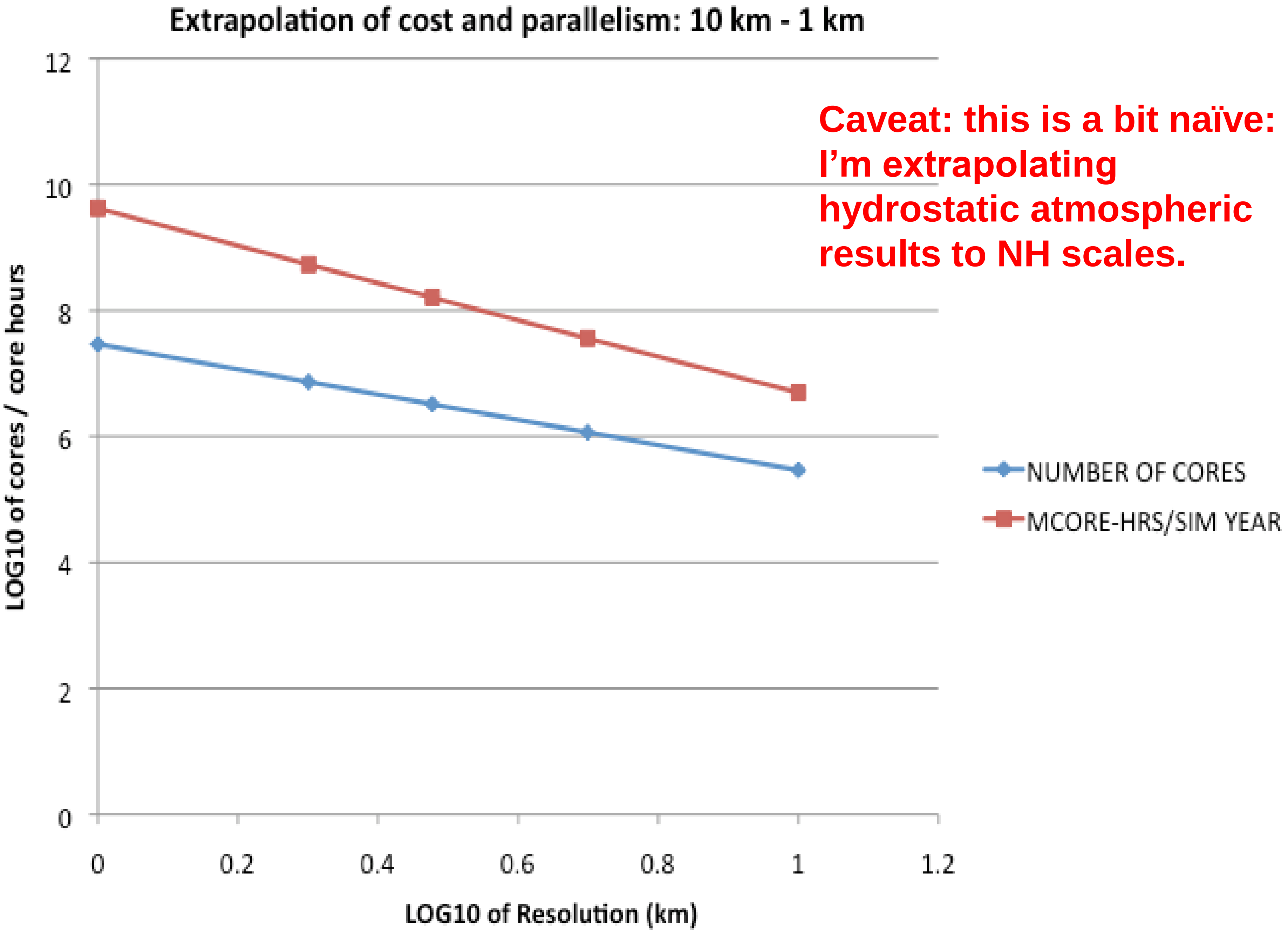
What will we run at exascale?

- Single ultra high-resolution models
  - Capture convective-scale (cloud) processes
  - Non-hydrostatic; 1 km resolution
  - Most vulnerable to a system component failure
- Climate system data assimilation
  - Study predictability on seasonal/decadal scales
  - Ensemble Kalman Filter is scalable but very data-intensive!
  - EKF can (in principle) be made fault resilient

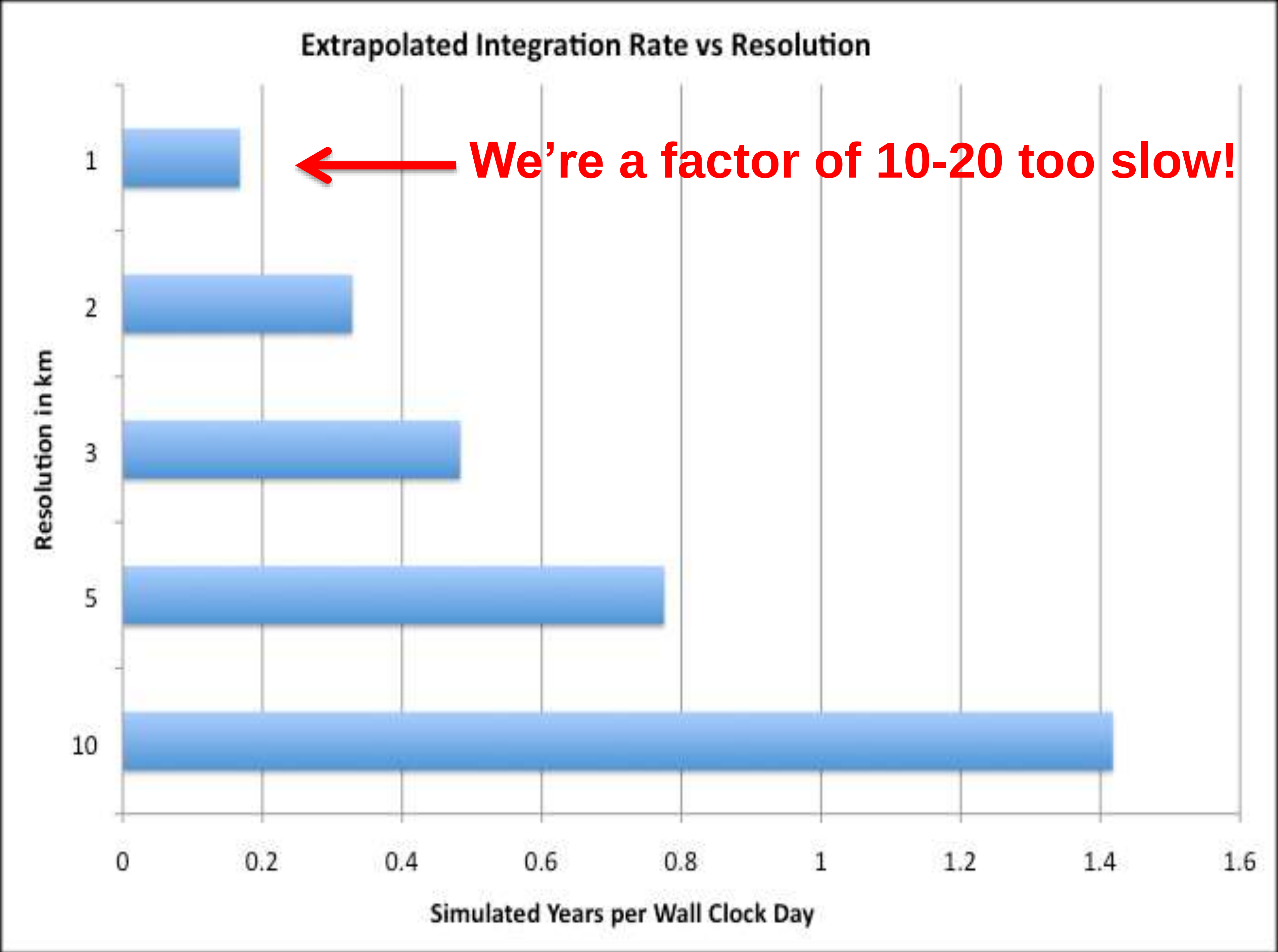
What will we run at exascale?

- Very large ensembles of "low-res" models
  - Study natural variability and extremes
  - Higher I/O per flop than high resolution
  - Free fault tolerance – resubmit failed ensemble members.



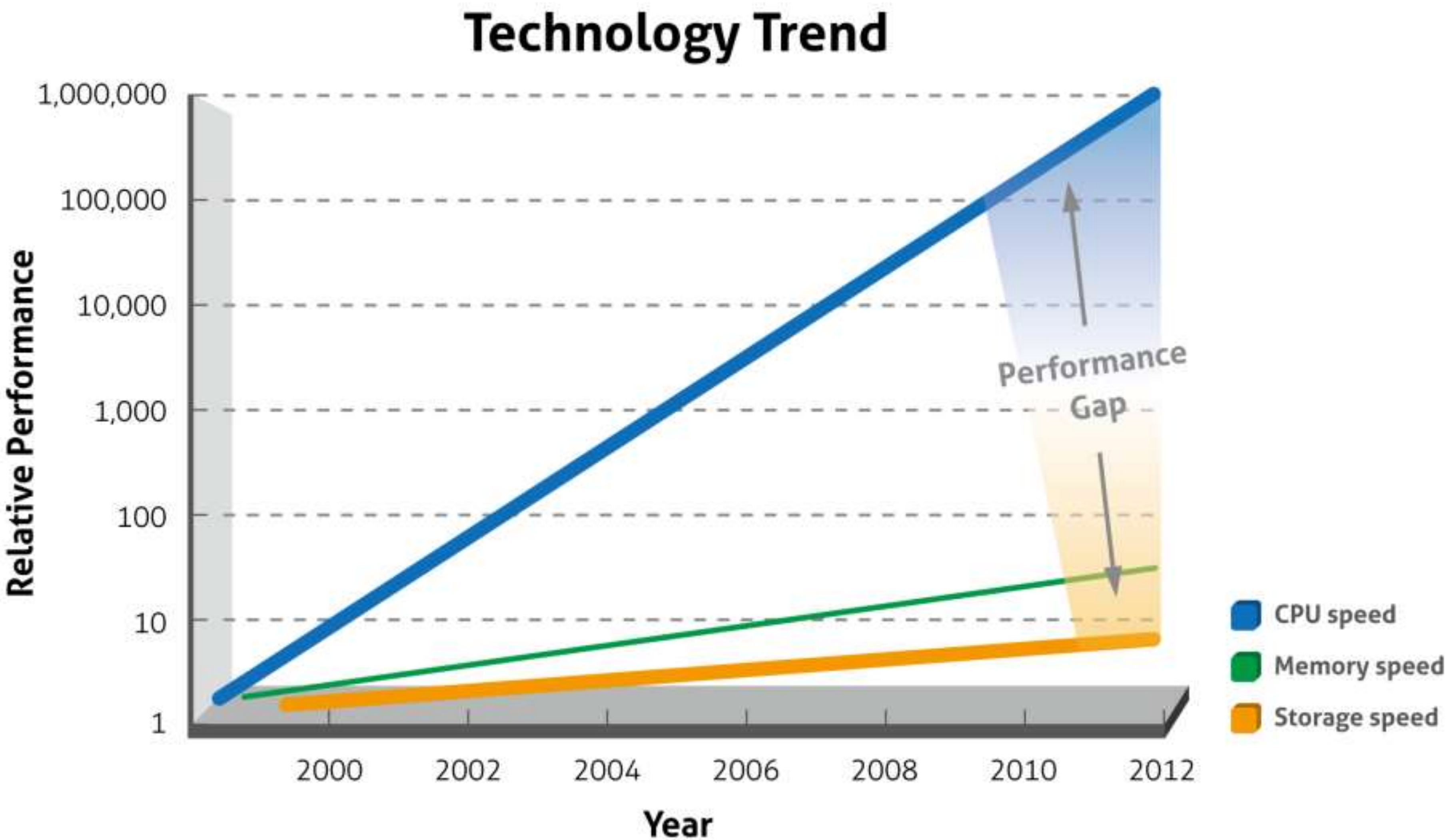


Caveat: this is a bit naïve: I'm extrapolating hydrostatic atmospheric results to NH scales.



Extrapolated Integration Rate vs Resolution
We're a factor of 10-20 too slow!
Resolution in km
1
2
3
5
10
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
Simulated Years per Wall Clock Day

the performance gap



What can we do to make up the performance gap(s)?

- Better algorithms!
  - Local vs non-local
  - Some evidence that simpler numerics is better
- Better code!
  - Optimized for efficient use of scarce resources
  - Memory heirarchy, divides, SIMD vectorization

What can we do to fill the performance gap(s)?

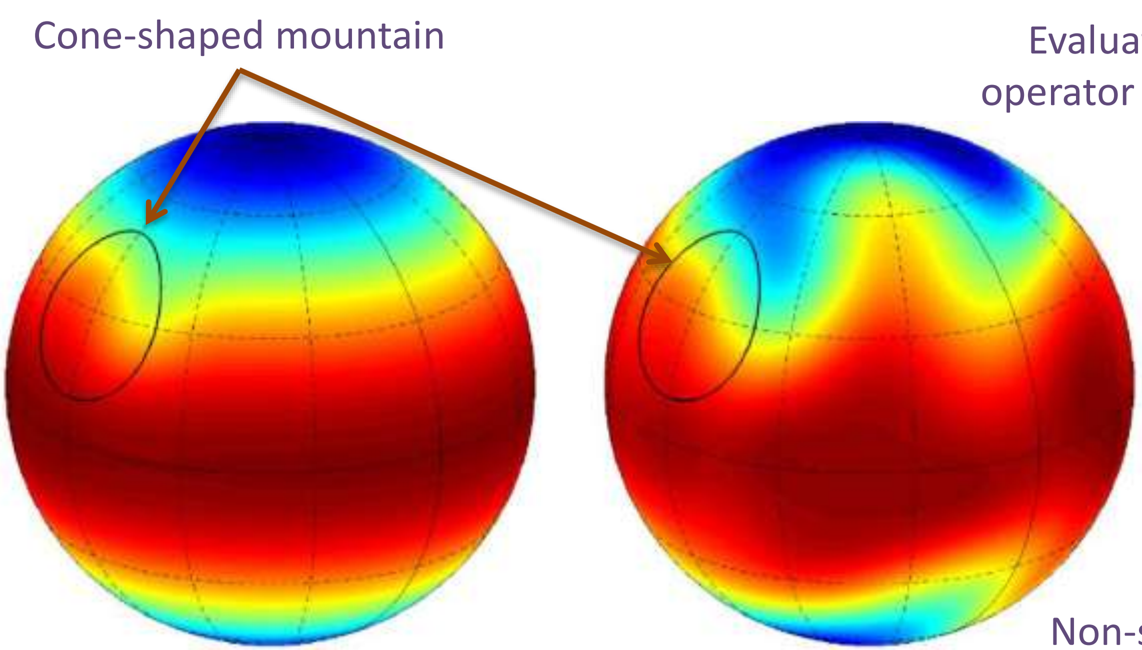
- More parallelism!
  - Nested data/task parallelism
  - Time parallelism
- Better hardware!
  - Faster memory (e.g. stacked)
  - Memory-like disk; disk-like memory
- Quit obsessing about flops!

A simple illustration of why flops are not relevant…

Work by NCAR SIParCS Intern, Tuan Ta (Ole Miss)

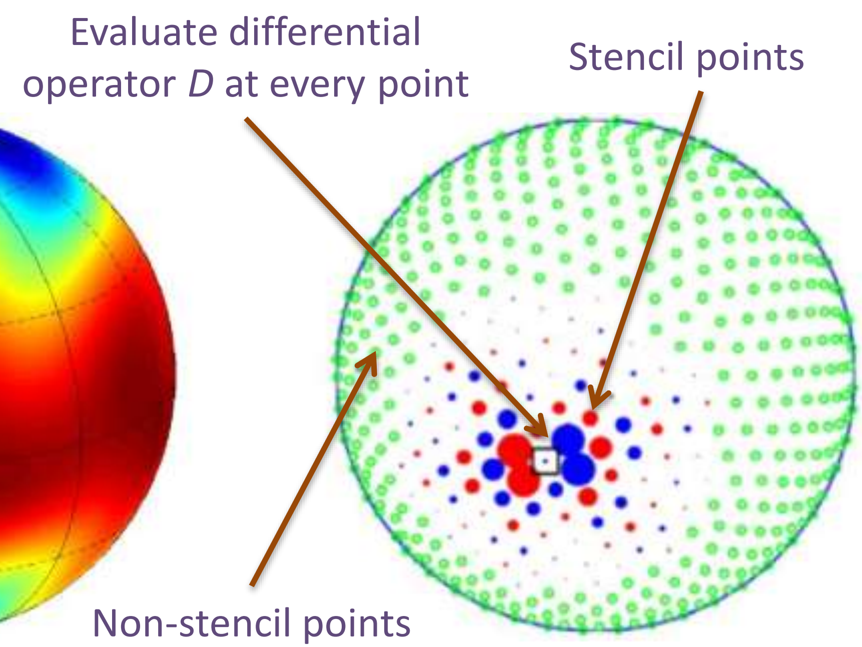
Benchmark Problem

- Shallow Water Equations (SWE)
  - A set of non-linear partial differential equations (PDE)
  - Capture features of atmospheric flow around the Earth
- Radial basis function-generated finite difference (RBF-FD) methods



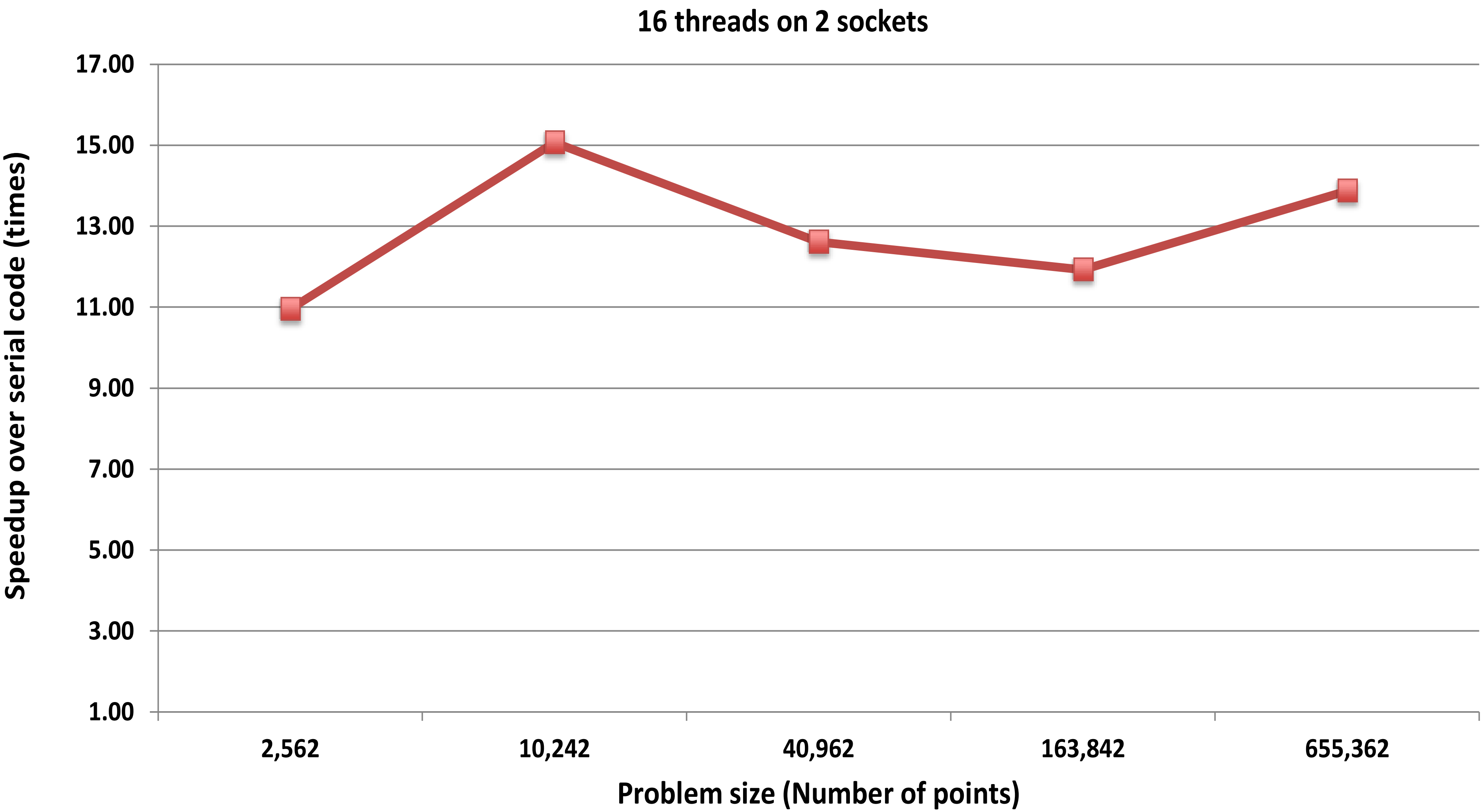
Day 1 Day 15

RBF-FD solution to SWE test case "Flow over an isolated mountain" using 655,532 points [1]



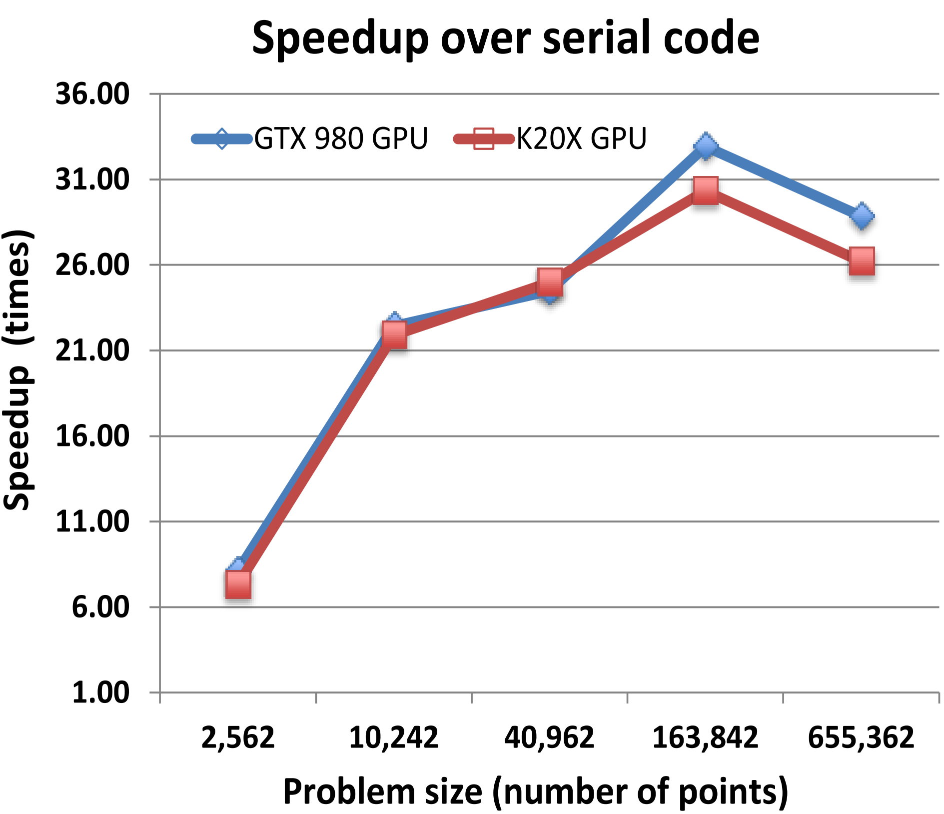
An example of 75-point stencil on a sphere [1]

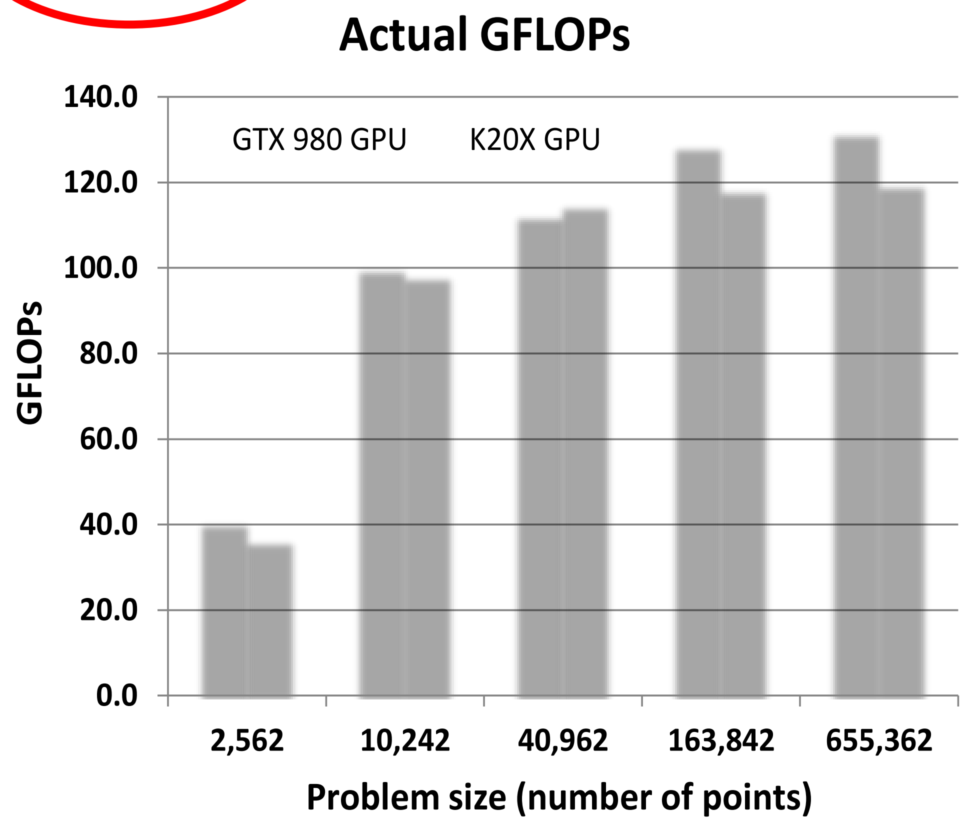
Best-effort OpenMP Implementation



More GFLOPs -> Faster?

	GTX 980 GPU	K20X GPU
Peak GFLOPs (double precision)	144	1310
Memory bandwidth (GB/s)	224	250
Market price (U.S. dollars)*	≈ $500	≈ ~ $3,000





More GFLOPs do not help unless we optimize memory access more aggressively!

A grossly unfair price performance comparison

- RBFs on dual GTX980 node $25K/TFLOPS
- Yellowstone workload ~$1M/TFLOPS
- Why are we so far off?
- System bandwidth ~6x
  - Dual Intel Xeon SB ~75 GB/sec
  - Dual GTX-980 ~450 GB/sec
- Production code inefficiency ~2x
- Interconnect/Infrastructure overhead ~1.3x
- Remaining: ~2.6x reflects “real” GPU advantage

Model efficiency

- CESM is 1.5 million source lines of code
- The computational intensity of CESM is well modeled by a simple daxpy operation:
  - Y(:) = a*X(:)+Y(:)
- Socket bandwidth is a critical parameter in determining its performance.

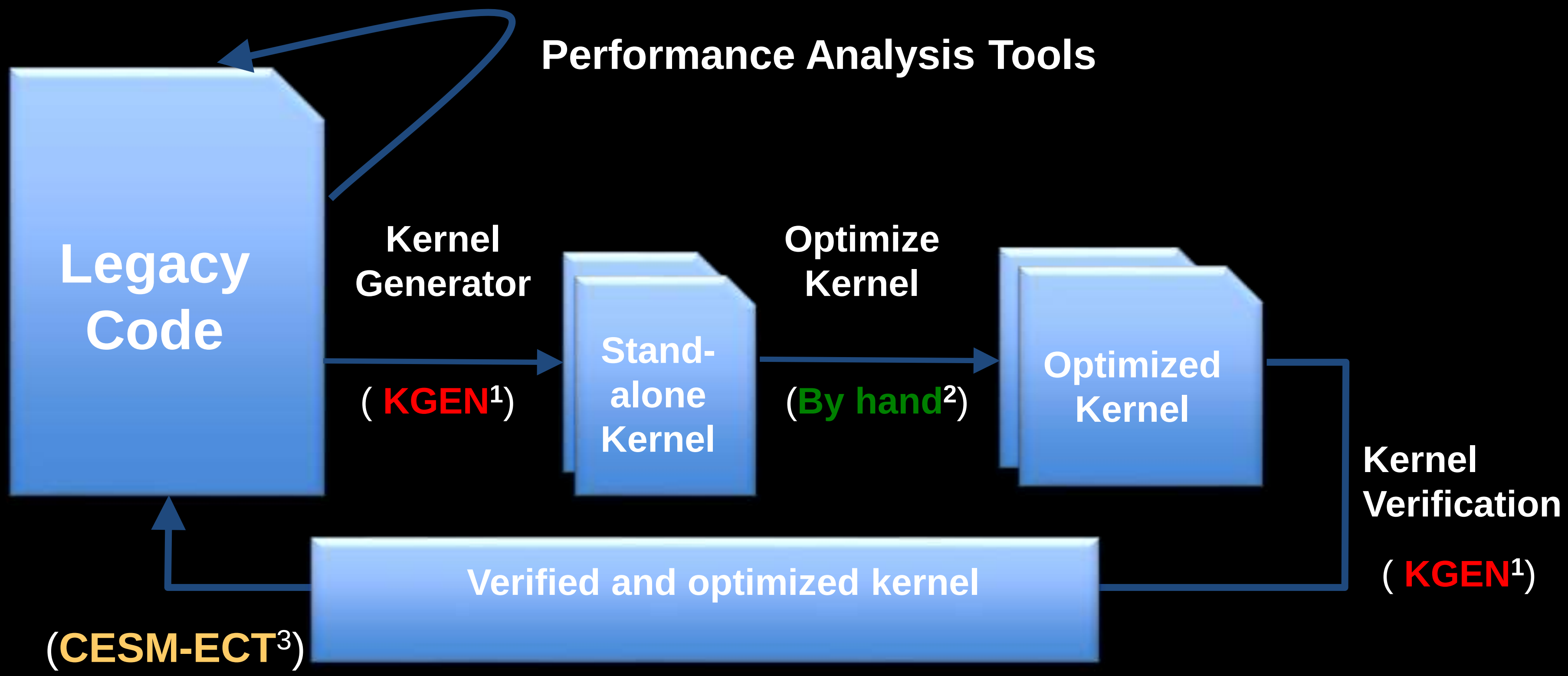
Leaks in model performance

We have a scarce resource and we're wasting it!

Our approach to improving efficiency: catch and release

- Learn how to optimize code
- Diagnose the problem with profilers
- Capture the problematic routine(s) in an auto-generated kernel
- Optimize code/fix compiler bugs
- Validate results
- Release back into the wild

Workflow for rapidly adapting code for many core architectures

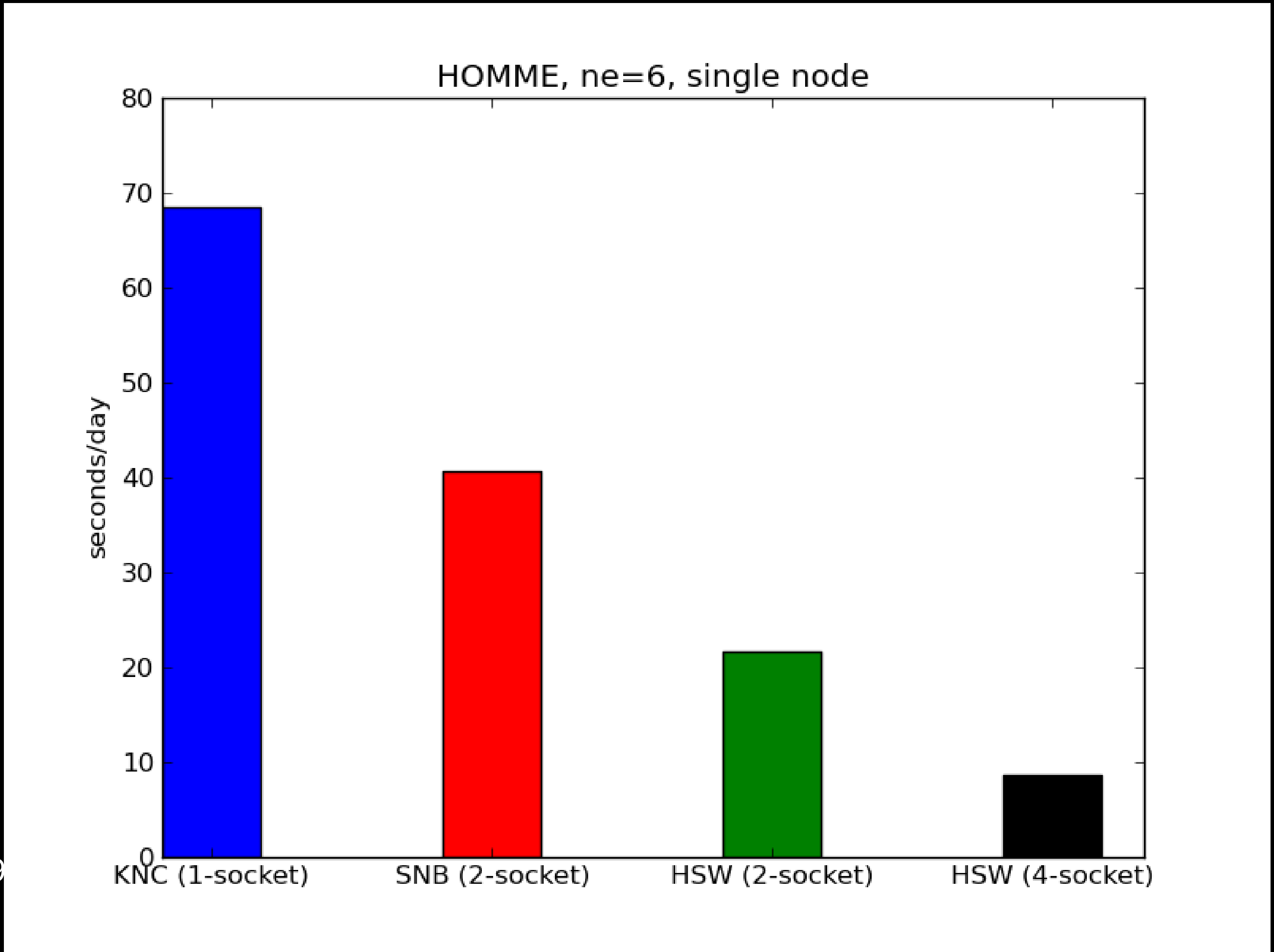


KGEN[1]: Semi-automated Kernel Generator
By hand[2]: Optimize & work with vendors to fix compiler problems
CESM-ECT[3]: Verify CESM with Ensemble Consistency Test

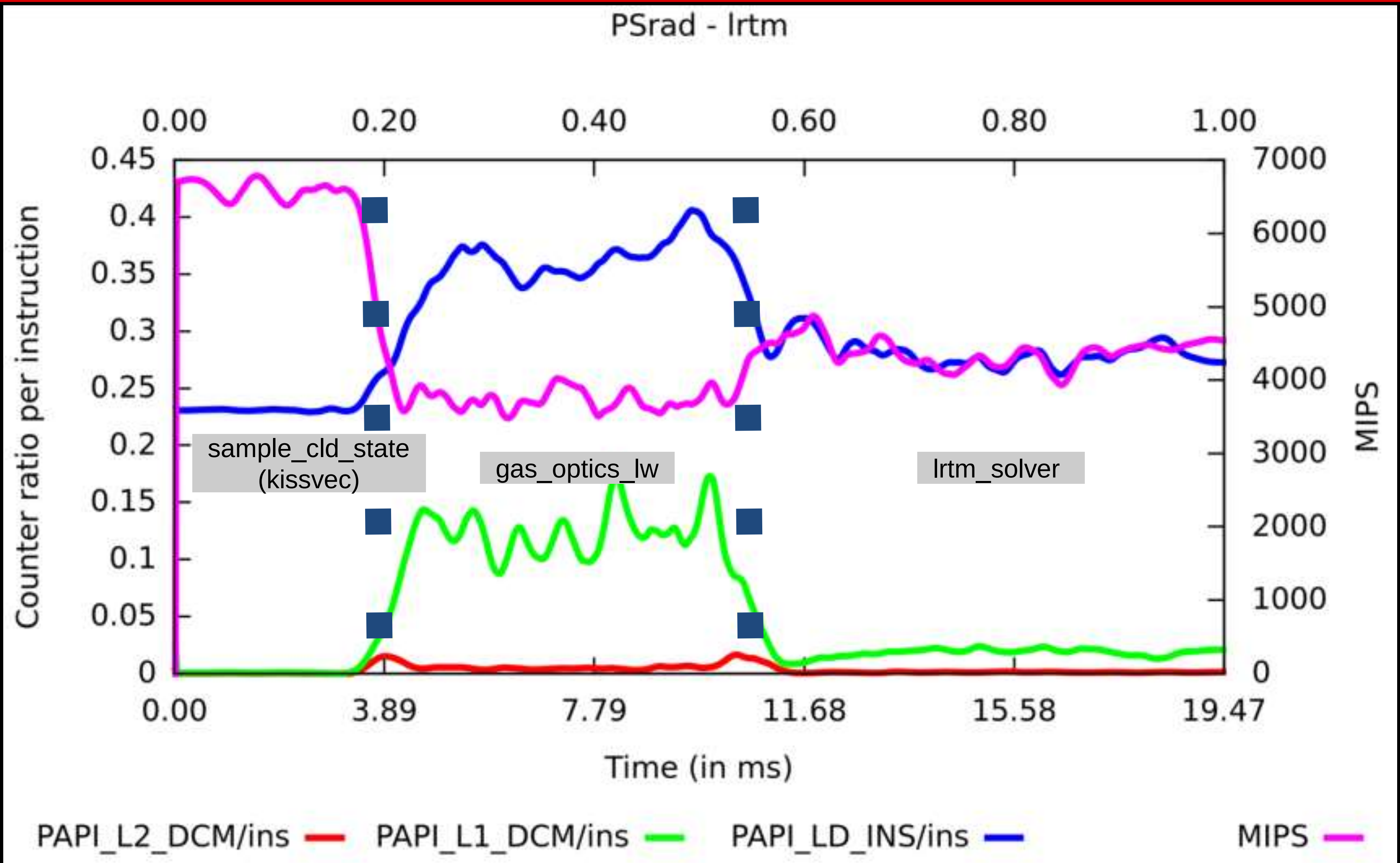
Where is time going in the atmospheric climate component?

- Spectral element PDE solver (35%)
- Radiative transport (13%)
- MG2 microphysics (13%)
- Convection (11%)
- Other (28%)

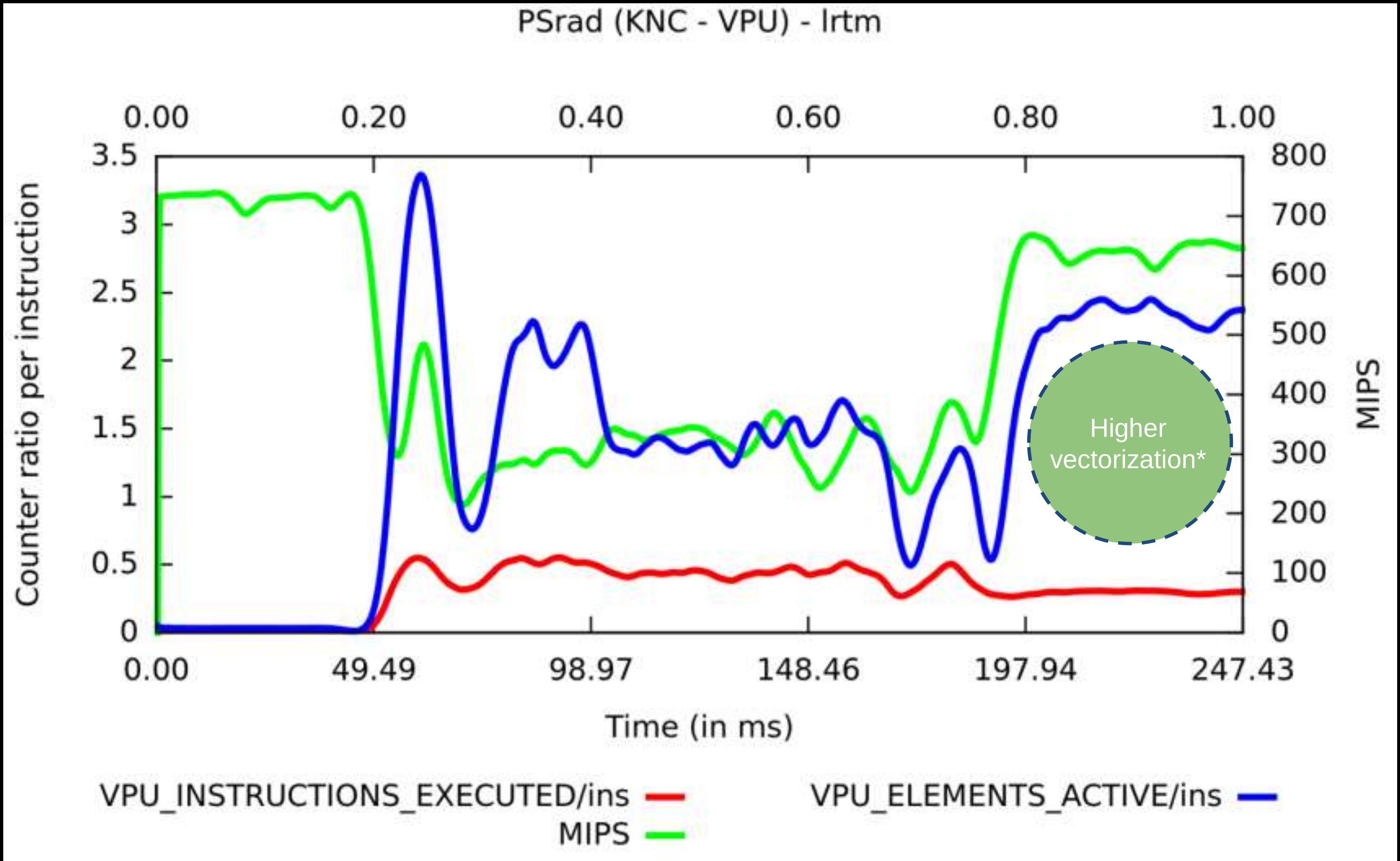
State of optimization of spectral element PDE solver on Xeon and Xeon Phi



‘EKG’ of Radiation Code Memory System Usage

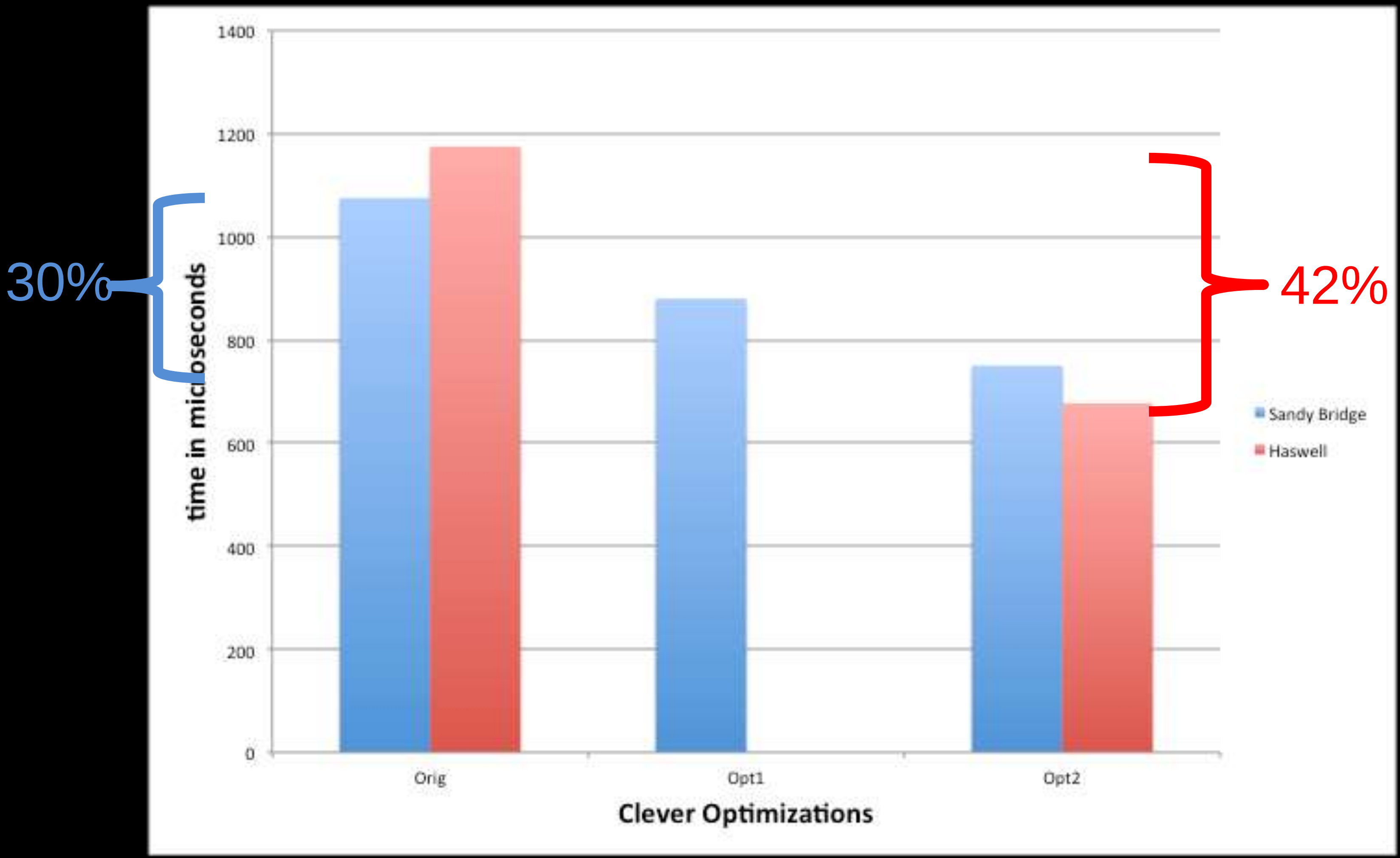


‘EKG’ of Radiation Code
Vector Unit Usage



vectorized data movement including broadcast

Optimization of MG2 microphysics KGEN kernel on Intel Xeon



Thanks! Question?

Dr. Richard Loft

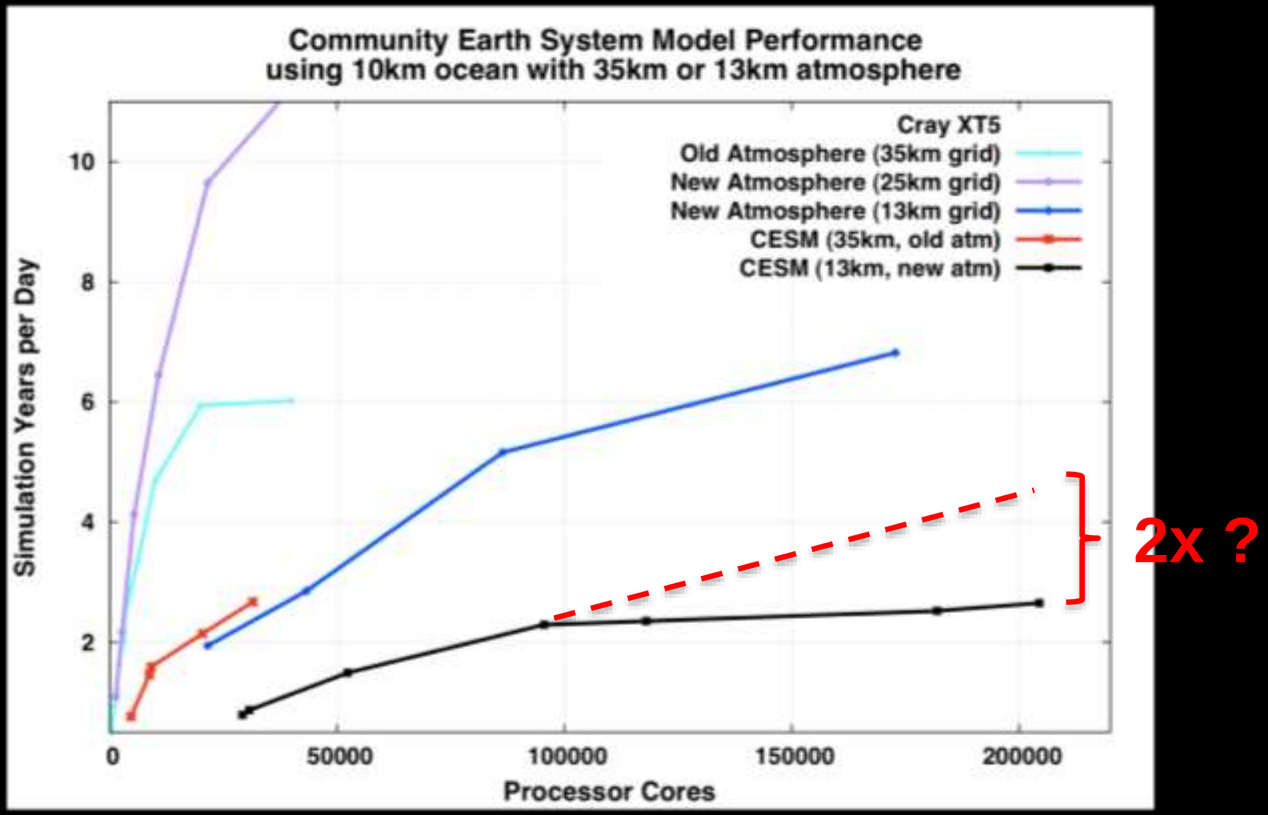
NCAR

loft@ucar.edu

January 13, 2015

CESM Computational Performance

(courtesy of Pat Worley)

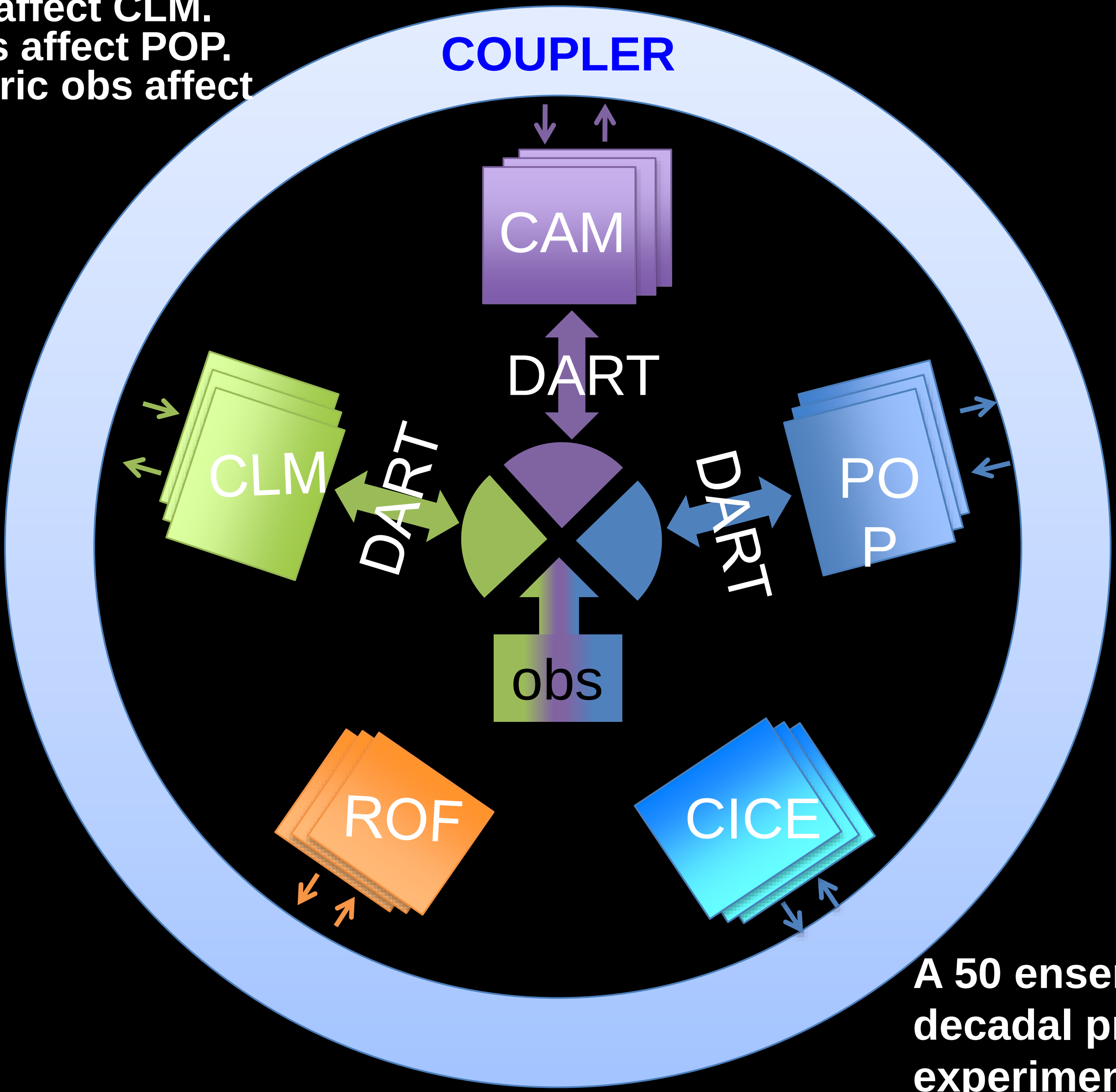


Spectral element-based atmospheric dynamics permits scalable CESM performance at high resolution.

Further improvements will require optimization of other components

Climate Data Assimilation with DART…

- Land obs affect CLM.
- Ocean obs affect POP.
- Atmospheric obs affect CAM.



A 50 ensemble member decadal prediction experiment with CESM 0.1˚ needs O(10^6 cores)