THE UNIVERSITY of TENNESSEE
NATIONAL LEADERSHIP IN HPC

Performance of the fusion code GYRO on ~~three~~ four generations of Cray Computers

Vincent C. Betro, Ph.D.
Computational Scientist
National Institute for Computational Sciences/
Application Acceleration Center of Excellence
vbetro@tennessee.edu

Mark R. Fahey, Ph.D.
Deputy Director
Industrial and Information Engineering Department
National Institute for Computational Sciences
mfahey@tennessee.edu

R. Glenn Brook, Ph.D.
Chief Technology Officer
Director, Application Acceleration Center of Excellence
Joint Institute for Computational Sciences
University of Tennessee & Oak Ridge National Laboratory
glenn-brook@tennessee.edu

Outline

- Introduction to GYRO
- Problem statement (NL02A Benchmark)
- Machine information
- Gotchas/Things to worry about
- Results from porting Gyro to several architectures
- Results of porting other codes to the Xeon Phi
- Interesting things to discuss/Conclusions

Gyro: Tokamak Plasmas

GYRO is a code for the numerical simulation of fusion tokamak microturbulence

- Computes the turbulent radial transport of particles and energy in tokamak plasmas
- Solves 5-D coupled time-dependent nonlinear gyrokinetic-Maxwell equations with gyrokinetic ions and electrons
- Developed by Jeff Candy and Ron Waltz at General Atomics
- GYRO can operate as a flux-tube (local) code, or as a global code, with electrostatic or electromagnetic fluctuations.
- Propagates system forward using a second-order, implicit-explict Runga-Kutta integrator and a fourth-order, explicit Eulerian algorithm
- Runs on a variety of machines: IBM Power, Cray XT and XE, SGI ICE and UV, Intel and Opteron Clusters

Gyro: Formulation

$$\frac{\partial f}{\partial t} = \mathcal{L}_a f + \mathcal{L}_b \langle \Phi \rangle + \{ f, \langle \Phi \rangle \}$$

$$\mathcal{F}\Phi = \int \int dv_1\, dv_2\, \langle f \rangle$$

- f is the gyrocenter distribution (measures the deviation from a Maxwellian), and $\Phi(\mathbf{r}) = [\varphi, A_{||}]$ are EM fields
- $\mathcal{L}_a$, $\mathcal{L}_b$ and $\mathcal{F}$ are linear operators
- <•>is a gyroaveraging operator
- The function $f(\mathbf{r}, v_1, v_2)$ is discretized over a 5-dimensional grid

Gyro: Discretization/MPI Formulation

- Eulerian schemes (e.g., GYRO) solve the gyrokinetic Maxwell equations on a fixed grid

$$f(r,\tau,n_{tor},\lambda,E) \Rightarrow f(i,j,n,k,e)$$

- For different code stages, the distribution of an index across processors is incompatible with the evaluation of operators on that index
  - for example, a derivative in r requires all i to be on processor
- Therefore, 2 and 3 index transpose operations must be executed
- Transpose operations use MPI ALL TO ALL.
  - Performance of these routines is the a scaling limiter
  - TRANSPOSE and SSUB libraries perform 3-index row transposes and 2-index column transposes
  - utilizing subcommunicators COMM_ROW and COMM_COLL
- code has been run up to 49,152 MPI processes on Cray XE6 at OLCF (Jaguar)

Gyro: OpenMP Implementation

- In December of 2011, preliminary hybrid parallelization (MPI+OpenMP) modifications were made to GYRO.
- For a given MPI task, we can have additional OpenMP threads which share memory
- The total core count is the product (number of MPI tasks) times (number of OpenMP threads per task).
- All the OpenMP code tends to target loops over radius, which are left undistributed by MPI. These loops tend to have the structure

```
do 1=1,n_x
   do ip=-n_band,n_band
      do (distributed stuff)
```

- For large radial grids and large gyro bandwidth there's quite a lot of work for OpenMP.

Gyro: Benchmark

NL02A Test Case

- 100 timesteps (real simulation would require at least 250,000)
- Kinetic electrons and electron collisions
- Electromagnetic effects
- Radial annulus with non-periodic boundary conditions and flat profile
- An 8-toroidal-mode electrostatic (ions and electrons) case
- 8 x 400 x 12 x 8 x 28 x 2 grid
    - Grid is typical of production runs and represents roughly the minimum grid size to obtain physically accurate results

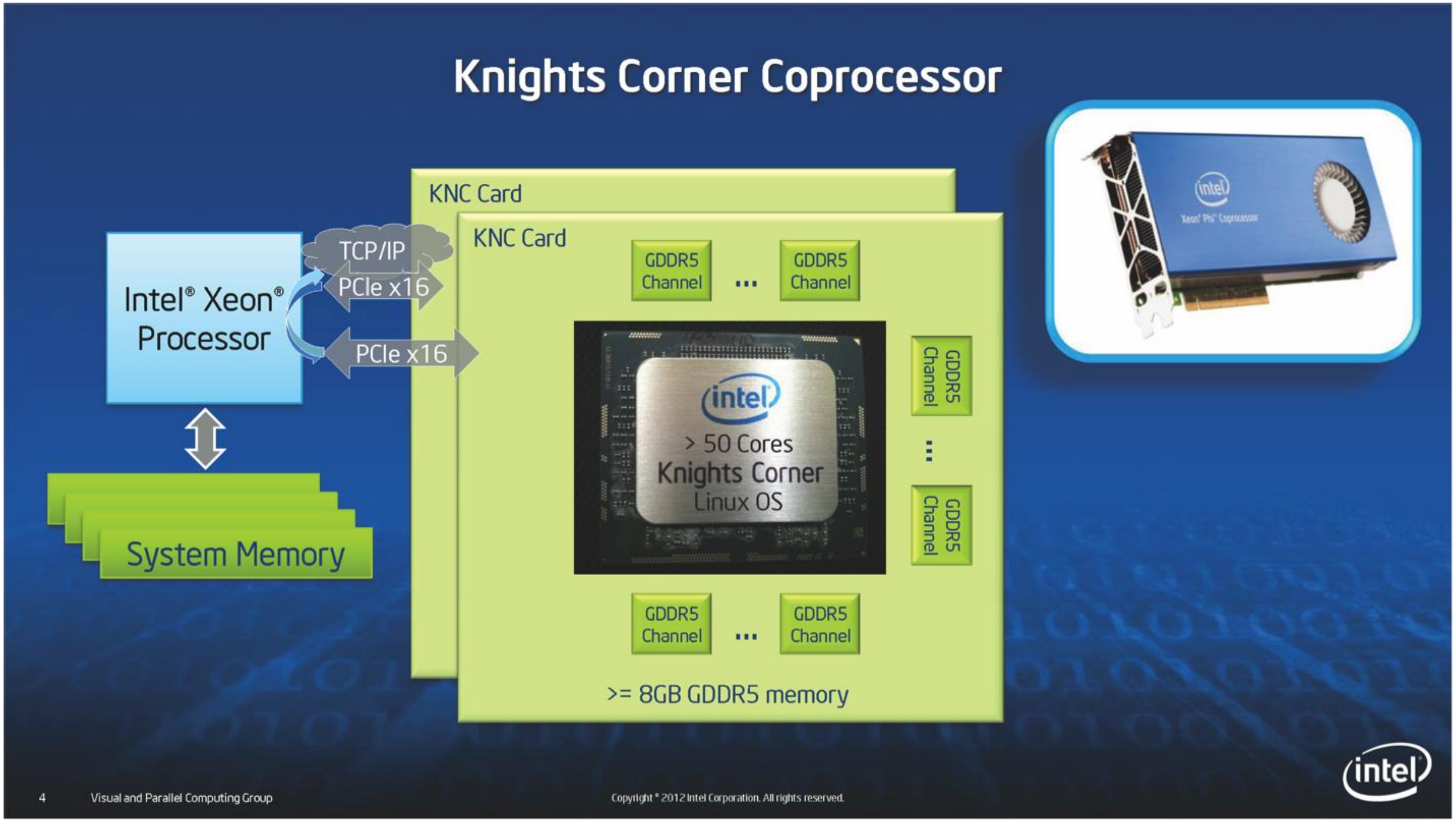
Machine Information

Kraken: Cray XT5 Peak Performance: 1.17 PFLOP/s	
Nodes	9,408
Interconnect	Cray SeaStar2+
Interconnect Bandwidth	57.6 GB/s (total)
CPU model	AMD Opteron (Istanbul)
Memory Bandwidth	21 GB/s (peak)
CPUs per node	2 6-core, 2.6GHz
RAM per node	16 GB

Ares: Cray XE6/XK6 Peak Performance: 18.5 TFLOP/s	
Nodes	36
Interconnect	Cray Gemini
Interconnect Bandwidth	99.6 GB/s (total)
CPU model	AMD Opteron (Bulldozer)
CPUs per node	2 16-core, 2.2Hz
Memory Bandwidth	83.5 GB/s (peak)
RAM per node	32 GB
NVIDIA TESLA X2090 GPUs	1 x 16 nodes, 6 GB RAM

Beacon – Phase 2 Cray CS300-AC™ Cluster Supercomputer Peak Performance: 210.1 TFLOP/s	
Nodes	4 service, 6 I/O, 48 compute
Interconnect	FDR IB Fat Tree
Interconnect Bandwidth	56 GB/s (total)
CPU model	Intel Xeon E5-2670
CPUs per node	2 8-core, 2.6GHz
Memory Bandwidth	128 GB/s (peak)
RAM per node	256 GB
SSD per node	2 x 480 GB (compute), 16 x 300 GB (I/O)
Intel® Xeon Phi Coprocessors per node	4 x 5110P 60-core, 1.053GHz 8 GB GDDR5 RAM

Machine Information:
The Intel Xeon Phi Coprocessor



Gotchas/Things to Worry About

- How to compare fairly across systems?

  - Different compilers
  - Different processors
  - Different configurations
  - Different libraries
  - Different interconnects

- Try to remove as many variables as possible!

  - Used newest available Intel compilers, since these are a must on the Xeon Phi
  - Only needed to link to FFTW/2.1.5, NETCDF, MKL libraries; all compiled with newest available Intel compilers on a given machine
  - Tried to run with saturated sockets to make memory bandwidth comparisons accurate
  - Tried to minimize number of nodes used to make interconnect bandwidth less important

Gotchas/Things to Worry About

- How we still got caught in snares:
  - Compiler and library versions on Kraken were older due to older architecture and lack of updated software stack for XT machines from Cray
  - Had to make sure not to use hyperthreading options since unavailable on Kraken

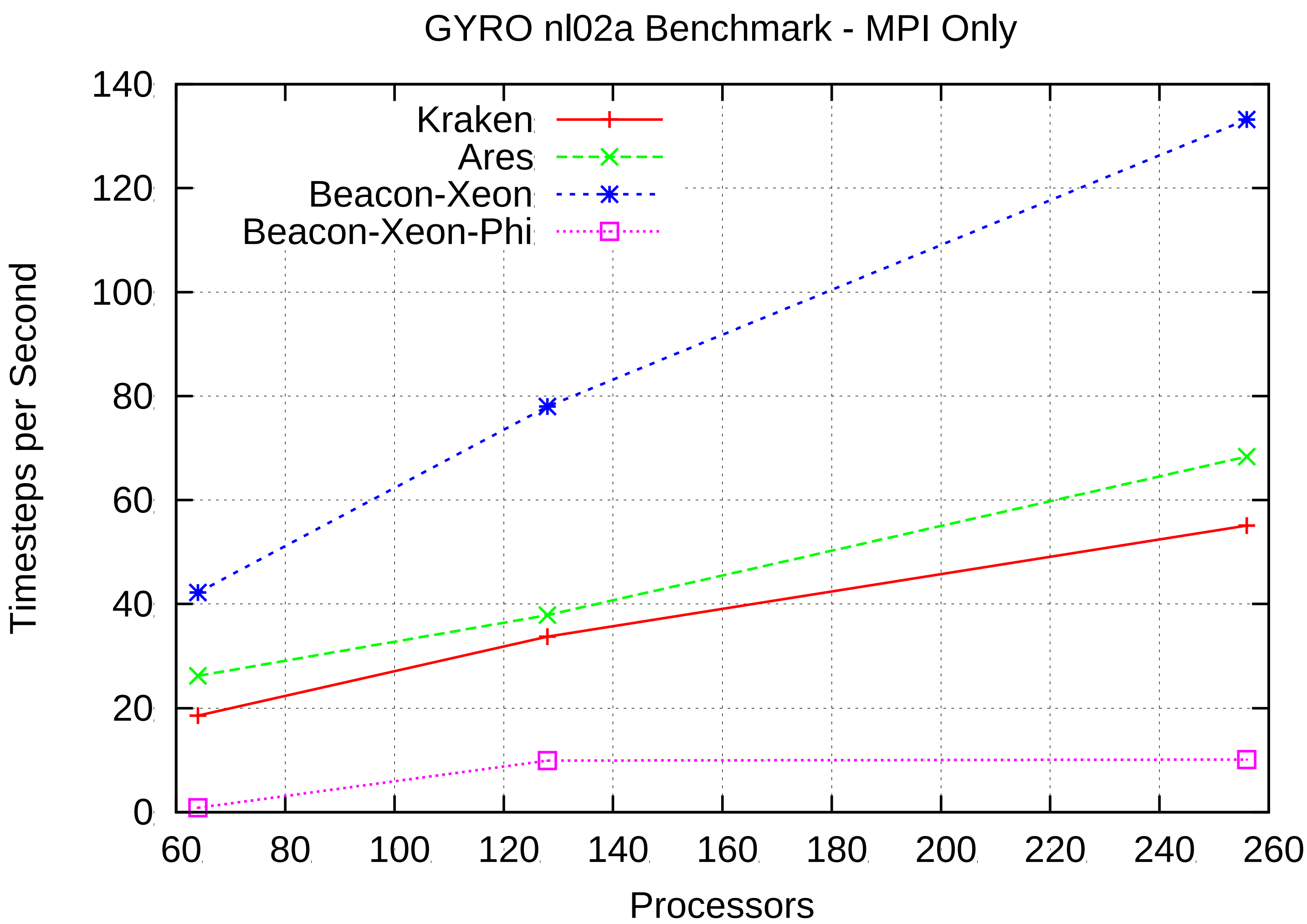
Gotchas/Things to Worry About

Still, these levelers were reasonably easy to implement for MPI only cases….

EXCEPT….

Xeon Phi has less memory per core than any of the CPUs, which required us to run less ranks per Phi “node” in order to fit the problem in memory

Gyro: MPI Performance



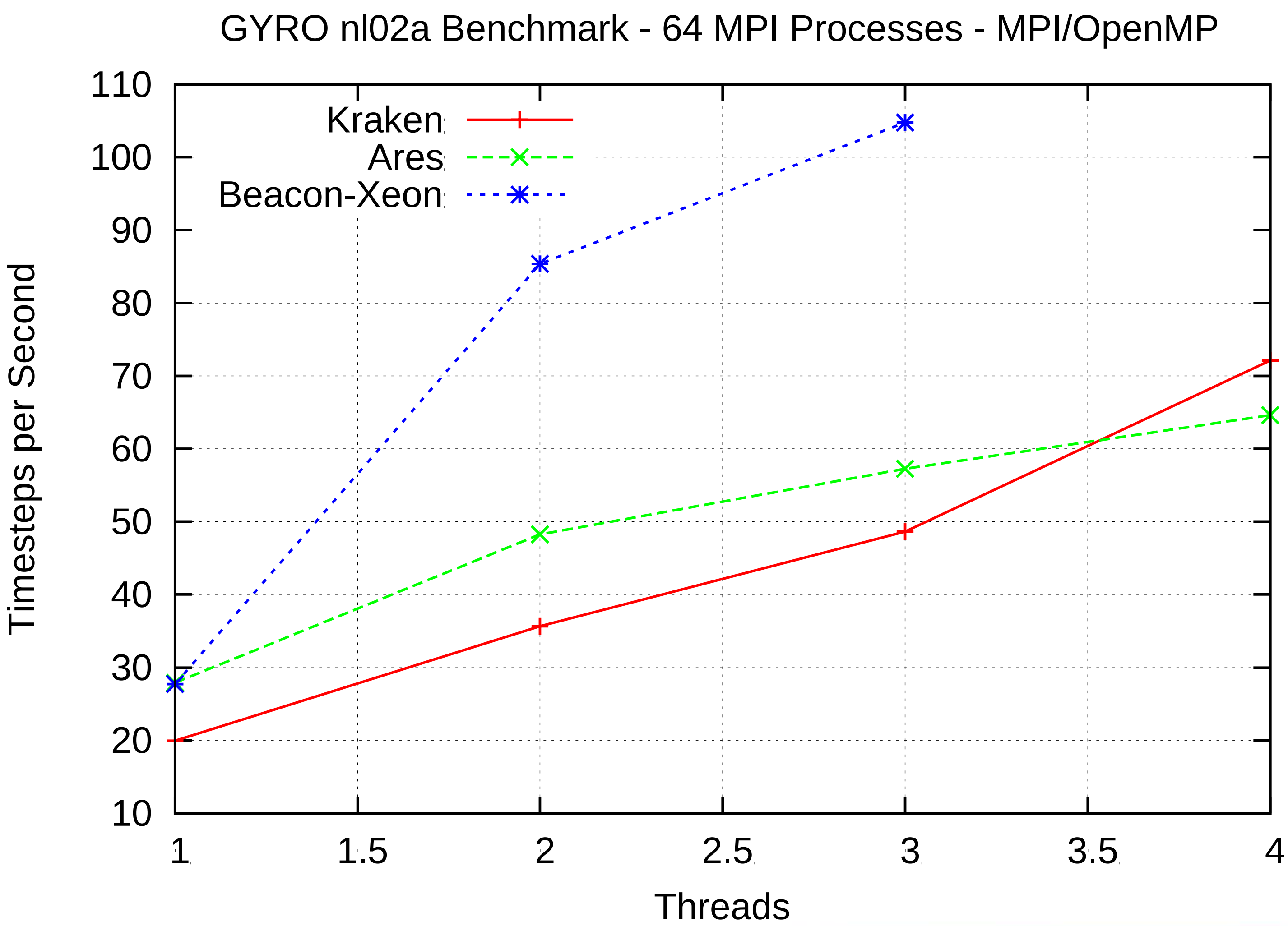
Gotchas/Things to Worry About

Things get even harder when implementing the MPI/OpenMP hybrid version…..

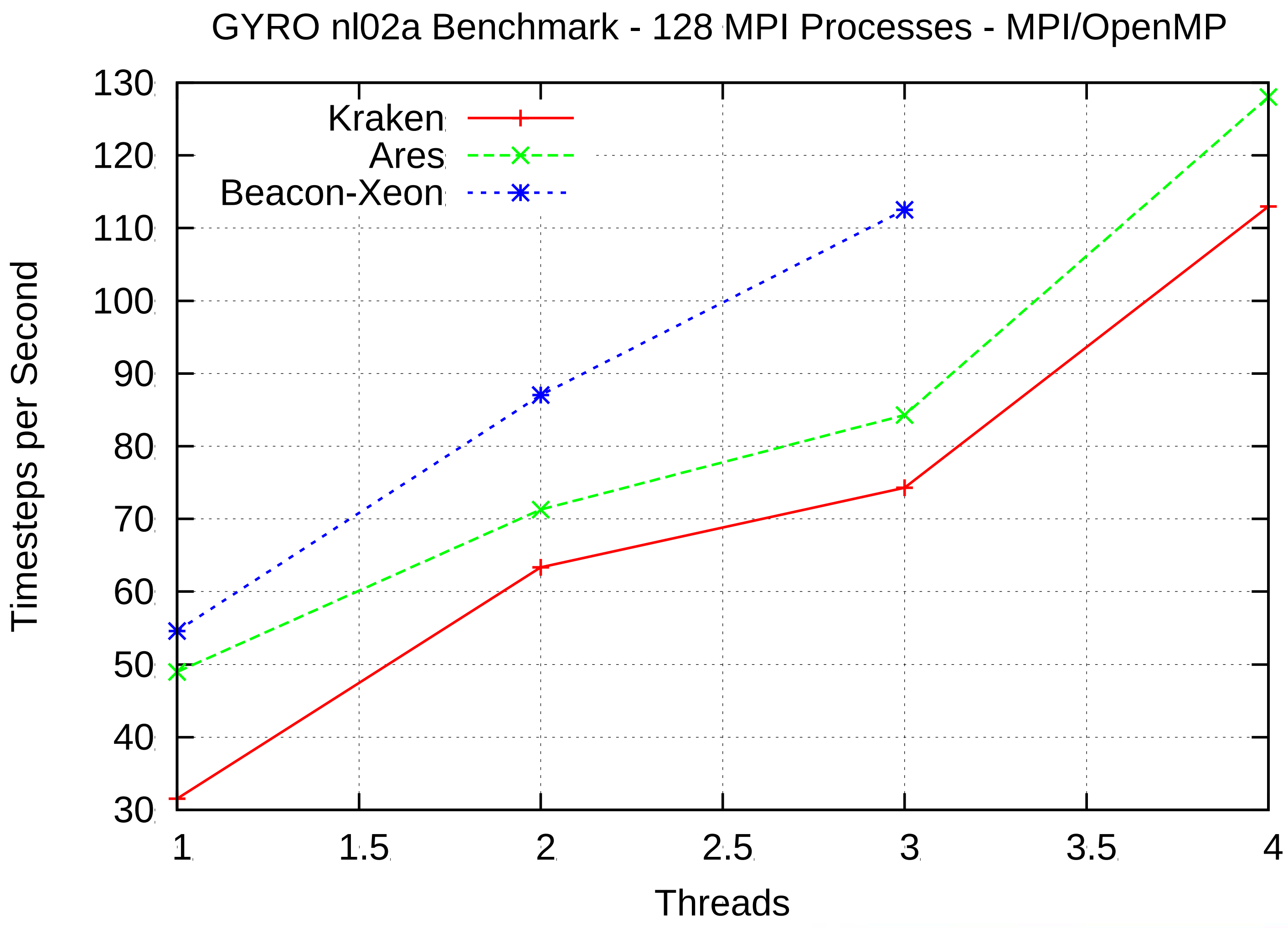
Kraken has 12 cores per node and the others have multiples of 16, which made it impossible to saturate the nodes at some core/thread counts

Had to run Xeon Phi threaded cases with same number of threads per node for 64 and 128 MPI ranks, as memory bandwidth caused the 128 MPI rank case to be slower at all counts due to contention. This broke the rule about having the same number of MPI ranks per node.

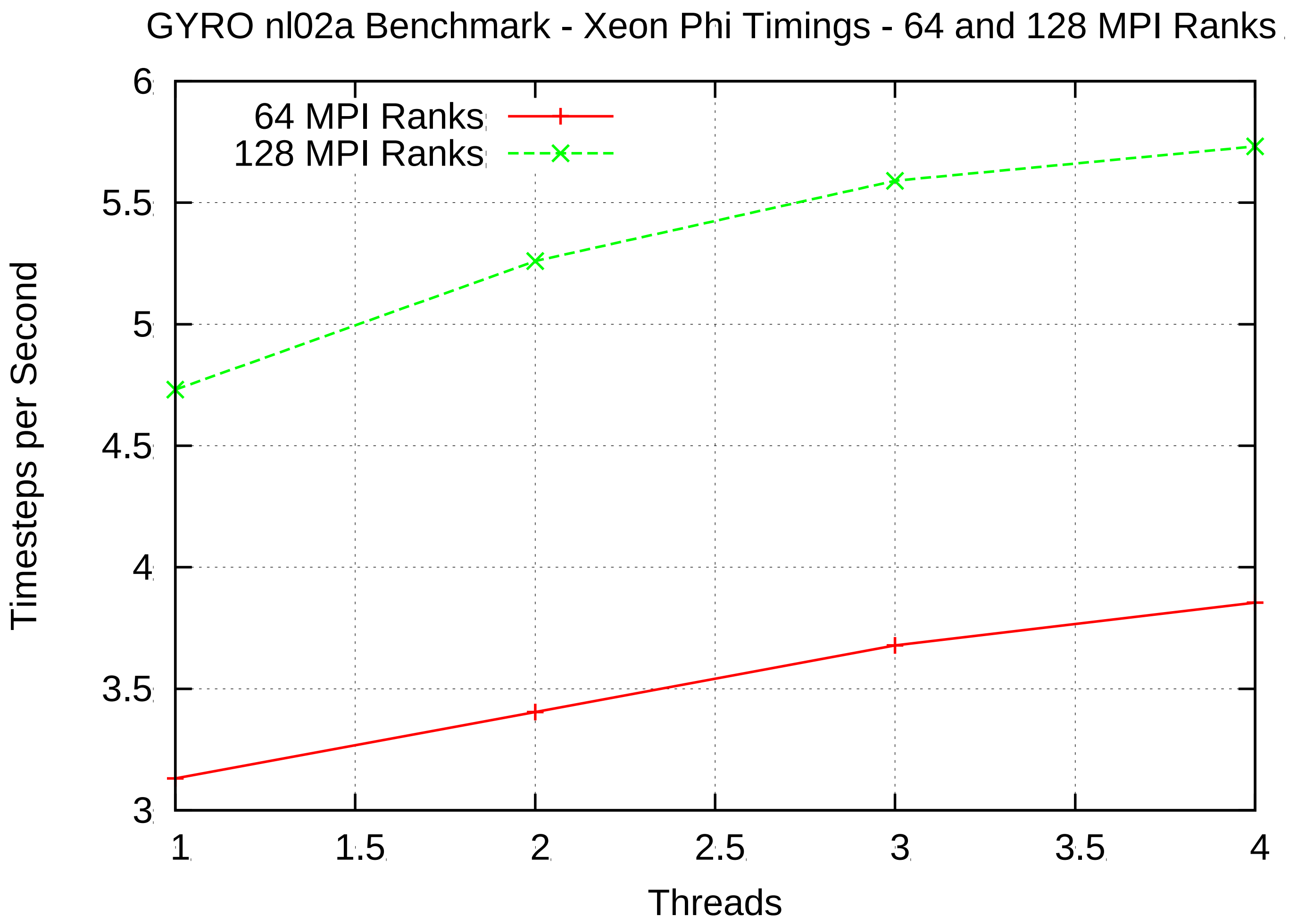
Gyro: Threaded Performance



Gyro: Threaded Performance

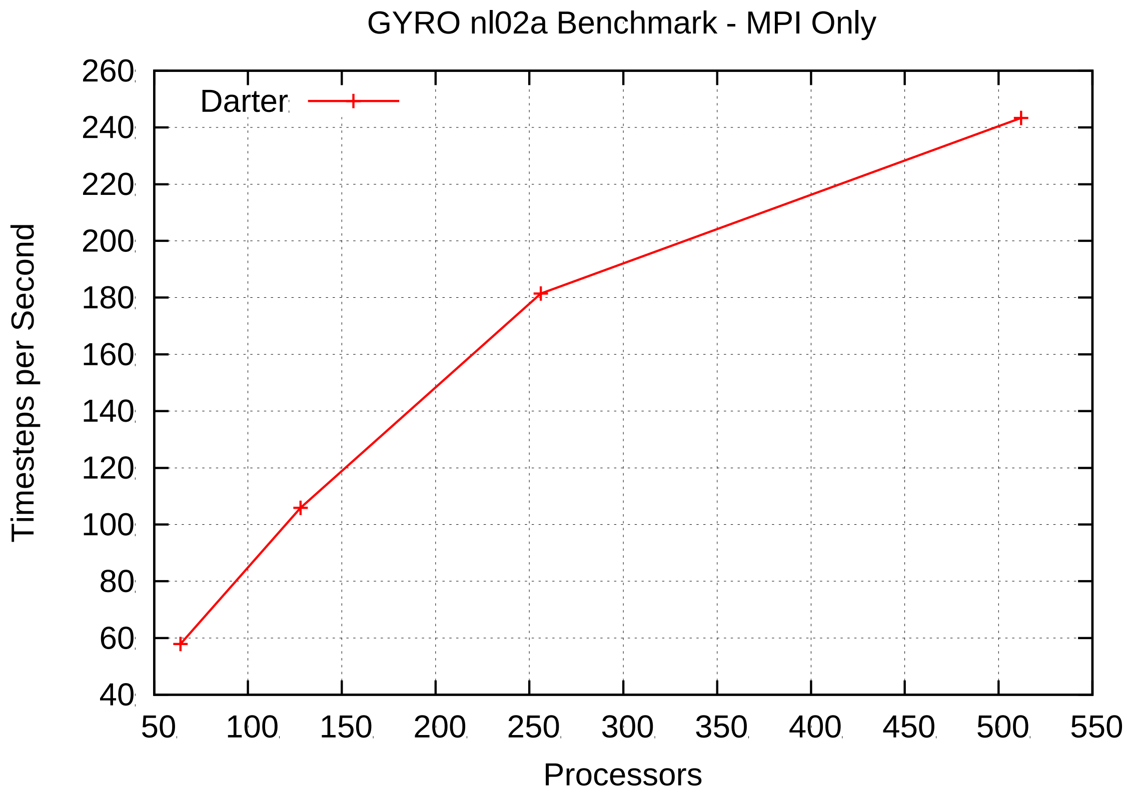


Gyro: Threaded Performance



Interesting Things to Discuss

- Things continue to improve on an XC30 with Intel E5-2670 processors and a Cray Aries interconnect



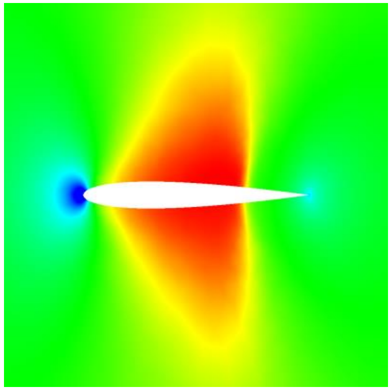
Comparing Results from Intel Xeon Phi

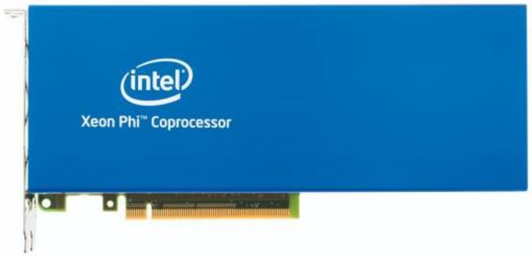
In order to have an apples to apples comparison on the Xeon Phi, we need to consider the following:

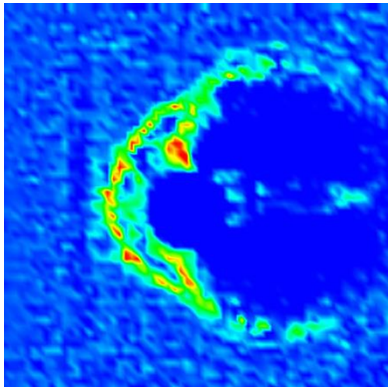
- Compare 2 Sandybridges to 1 Xeon Phi
  - 125W x 2 vs 255 W
  - 256 bit vector x 2 vs 512 bit vector
  - Run same code on both
- Doing some quick math, this means that we have similar power and vector processing, except that the Xeon is ~2.5 times the clock speed
- So, how do we make the Xeon Phi worth our while?
  - Many more threads
  - Programming versatility

Codes Investigated by AACE

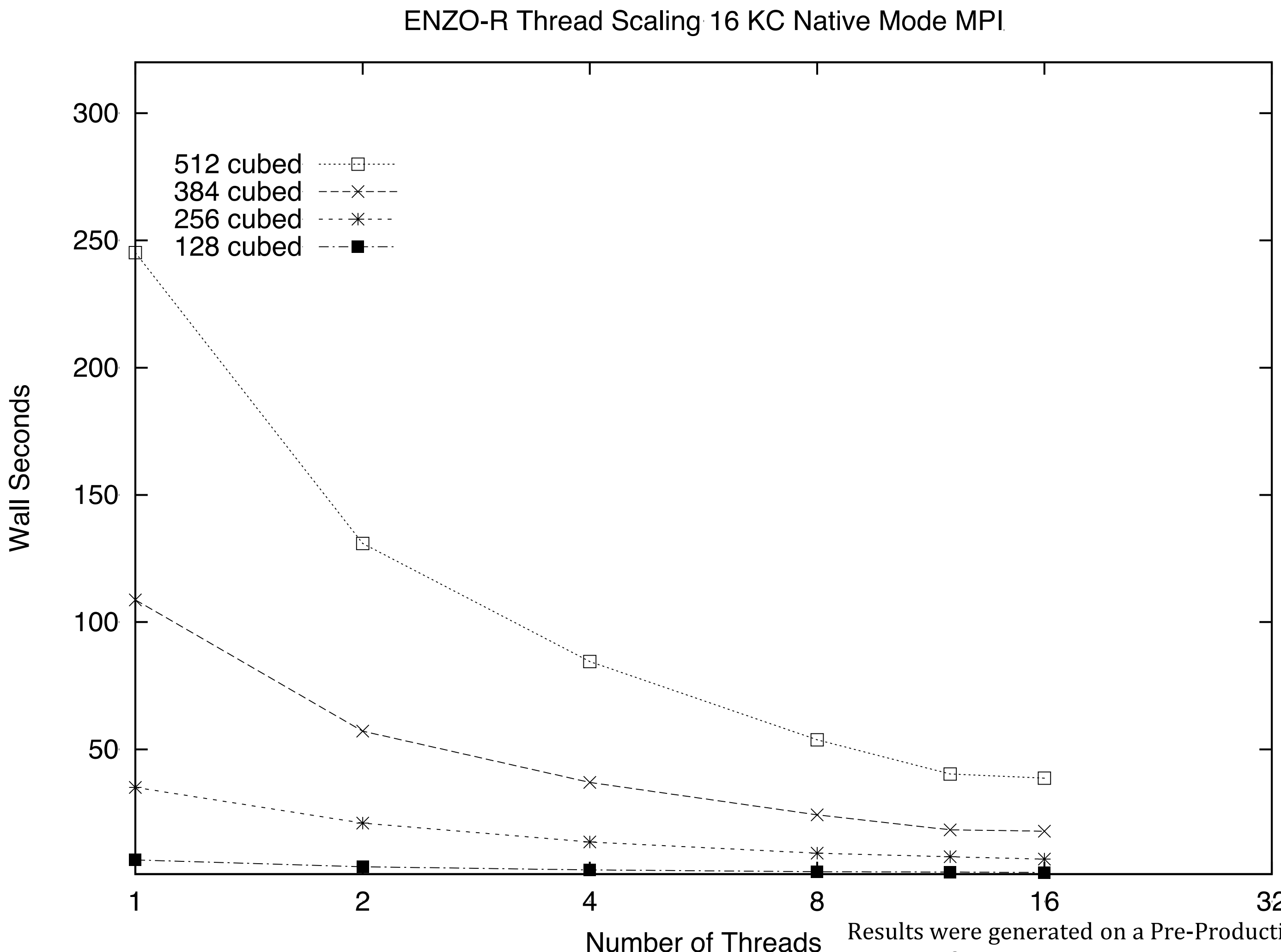
- Atmospheric modeling – HOMME-CAM (ported)
- Astrophysics – Enzo (ported and partially optimized)
- Computational Fluid Dynamics (CFD) – BGK-Boltzmann Solver (ported and optimized)
- Earthquake modeling – AWP-ODC (ported)
- Magnetospheric Physics – H3D (ported and partially optimized) and PSC (ported)
- Tokamak Plasmas – Gyro (ported)
- Agent Based Modeling – Transims and ASCAPE (ported)







Multi-KNC Scaling Study: Native



- ENZO-R
- N^3 mesh
- Decomposed into 4x4x4 blocks
- Native mode on 16 KNC
- 4 MPI ranks per KNC
- 1,2,4,8,12,16 threads per rank

Results were generated on a Pre-Production Intel® Xeon Phi™ coprocessor Beta software, 61 cores @ 1.09 GHz and 8 GB of GDDR5 RAM @ 2.75 GHz

Hybrid3d (H3D)

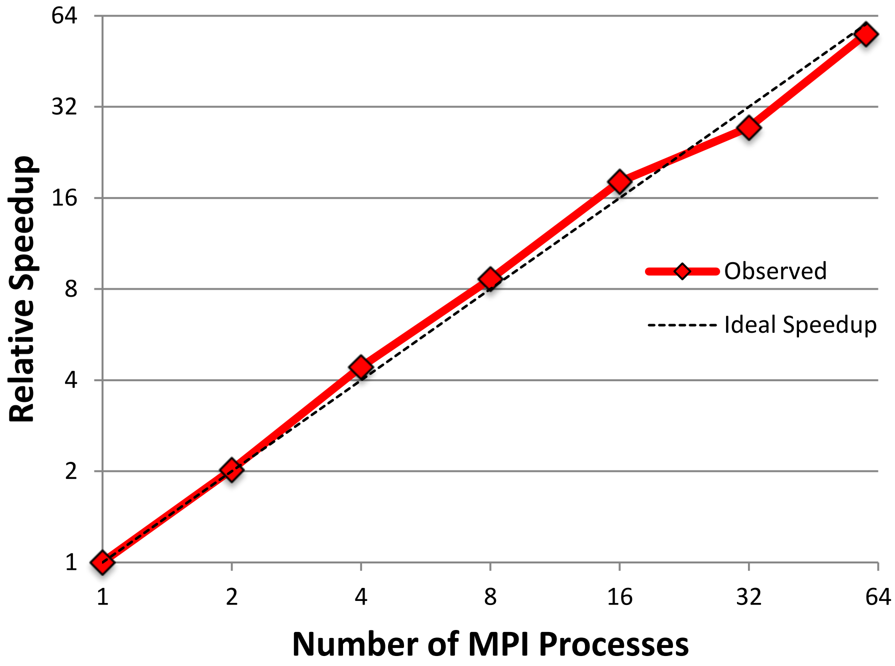
- Provides breakthrough kinetic simulations of the Earth's magnetosphere
- Models the complex solar wind-magnetosphere interaction using both electron fluid and kinetic ions
  - Unlike magnetohydrodynamics (MHD), which completely ignores ion kinetic effects
- Contains the following HPC innovations:
  1. multi-zone (asynchronous) algorithm
  2. dynamic load balancing
  3. code adaptation and optimization to large number of cores

Hybrid3d (H3D) was provided for porting to the the Intel® Xeon Phi™ Coprocessor by
Dr. Homa Karimabadi
hkarimabadi@ucsd.edu

Hybrid3d (H3D) Performance

Optimizations were provided by Intel senior software engineer Rob Van der Wjingaart.

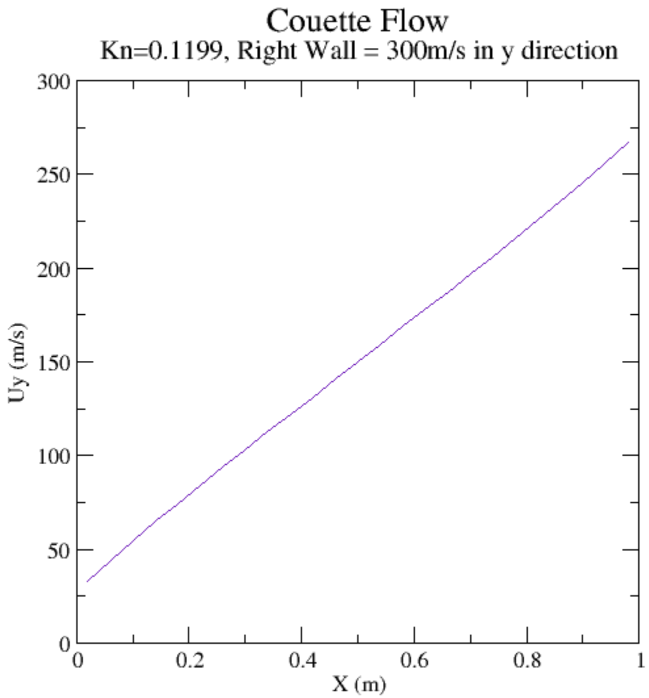
H3D Speedup on the Intel® Xeon Phi™ Coprocessor (codename Knights Corner)

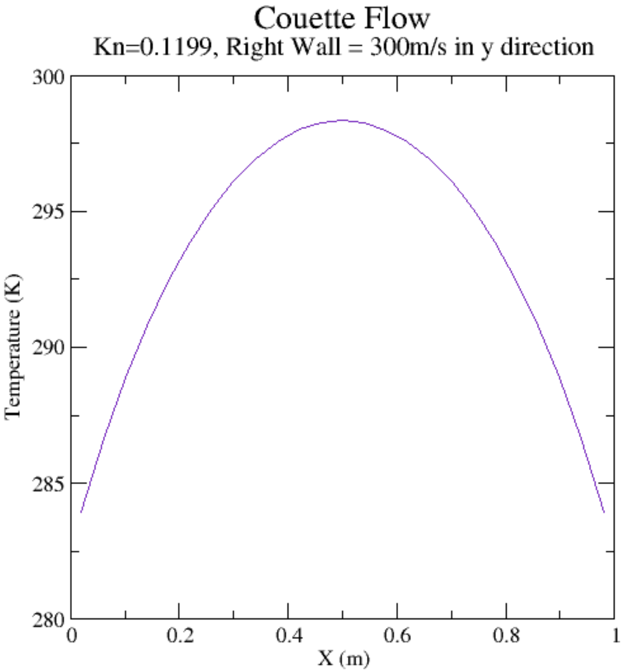


Results were generated on a Pre-Production Intel® Xeon Phi™ coprocessor with B0 HW and Beta SW
61 cores @ 1.09 GHz and 8 GB of GDDR5 RAM @ 2.75 GHz

Computational Fluid Dynamics (CFD)

Steady-state solution of a Couette flow using the Boltzmann equation with BGK collision approximation





The above CFD solvers were developed for the Intel® Xeon Phi™ Coprocessor by

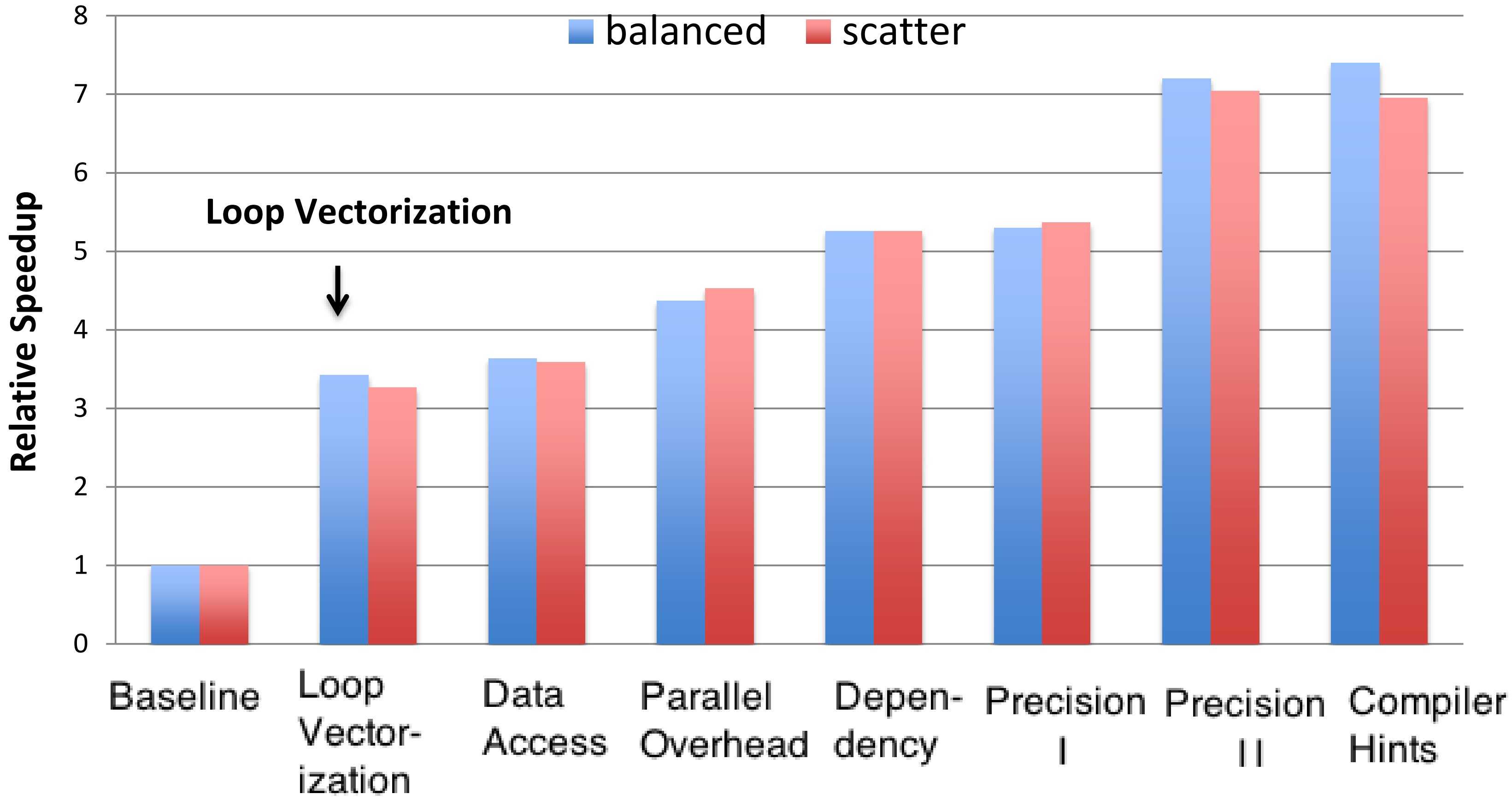
Ryan C. Hulguin
ryan-hulguin@tennessee.edu

Impact of Various Optimizations on the Model Boltzmann Equation Solver

- Optimized by Intel software engineer Rob Van der Wjingaart
- Base-line solver – all loops were vectorized except for one

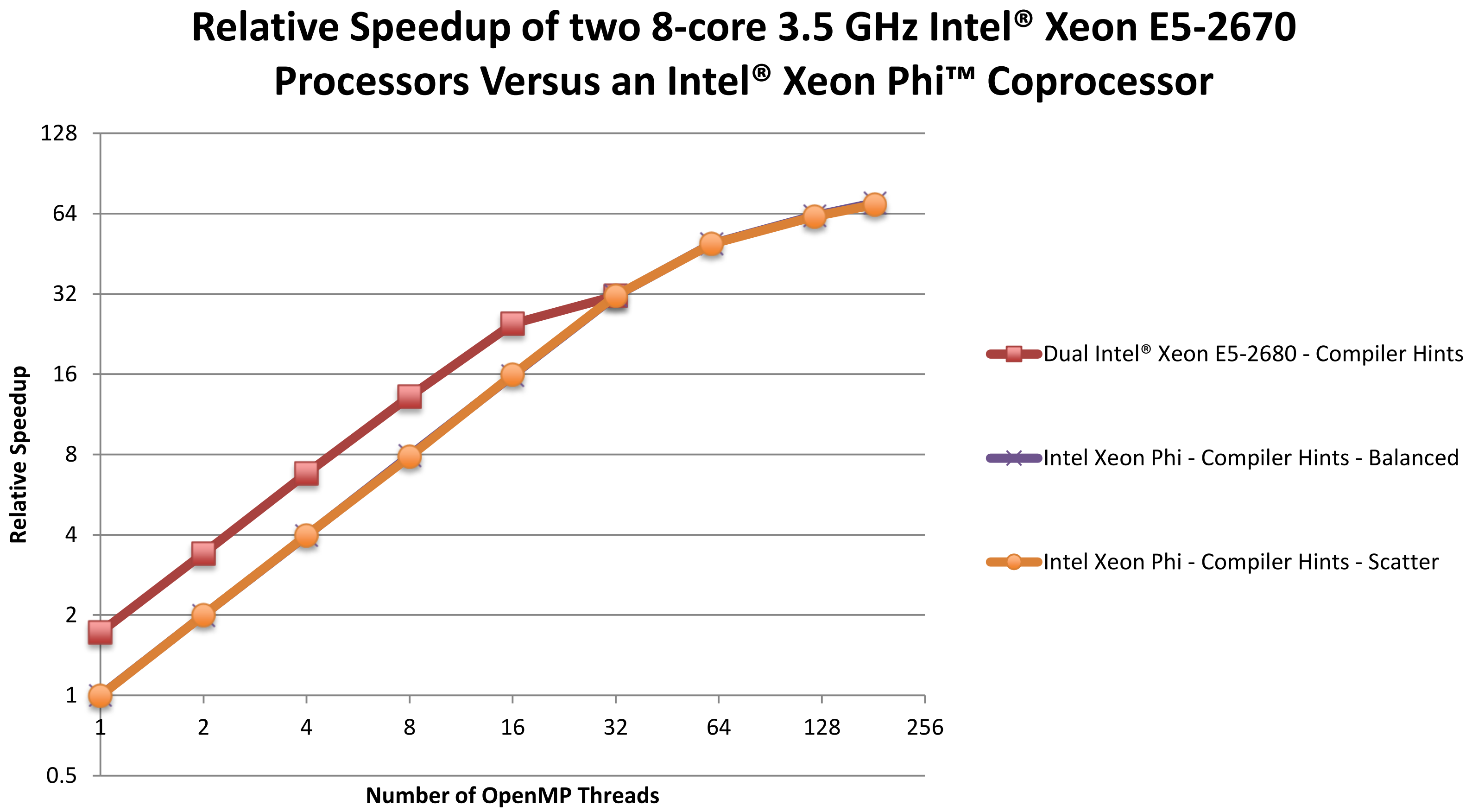
- Set I — Loop Vectorization
  - Stack variable pulled out of the loop
  - Class member turned into a regular structure
- Set II — Data Access
  - Arrays linearized using macros
  - Align data for more efficient access
- Set III — Parallel Overhead
  - Reduce the number of parallel sections
- Set IV — Dependency
  - Remove reduction from computational loop by saving value into a private variable
- Set V — Precision
  - Use medium precision for math function calls (-fimf-precision=medium)
- Set VI — Precision
  - Use single precision constants and intrinsics
- Set VII — Compiler Hints
  - Use #pragma SIMD instead of #pragma IVDEP

Optimization Results from the Model Boltzmann Equation Solver



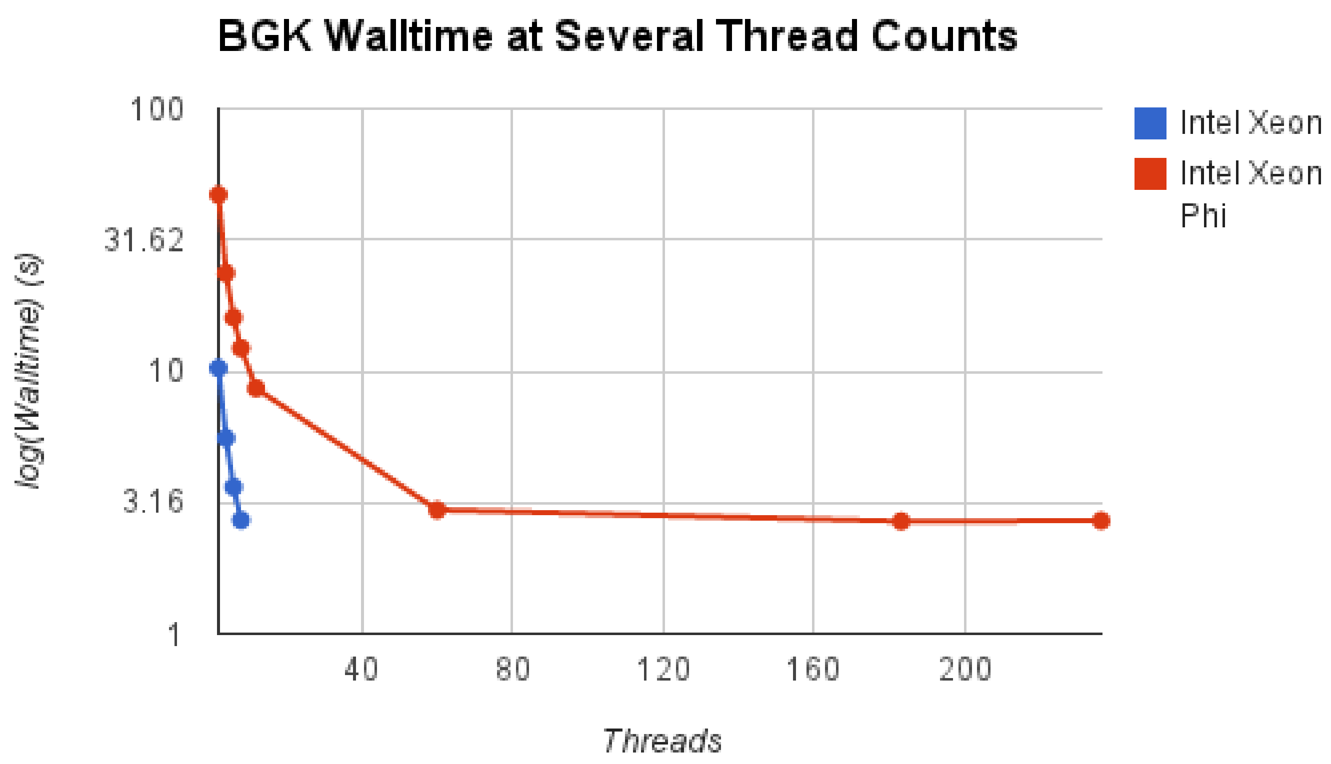
Results were generated on a Pre-Production Intel® Xeon Phi™ coprocessor with B0 HW and Beta SW
61 cores @ 1.09 GHz and 8 GB of GDDR5 RAM @ 2.75 GHz

Model Boltzmann Equation Solver Performance



Results were generated on a Pre-Production Intel® Xeon Phi™ coprocessor with B0 HW and Beta SW
61 cores @ 1.09 GHz and 8 GB of GDDR5 RAM @ 2.75 GHz

Model Boltzmann Equation Solver Performance: Another View



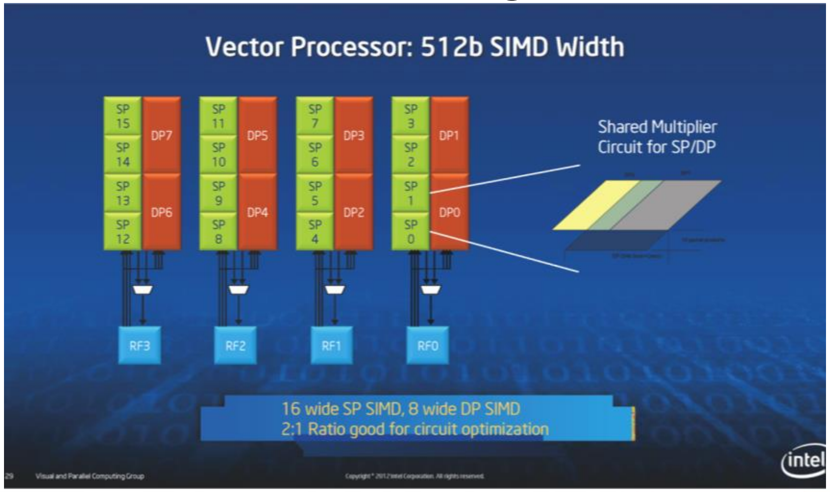
Results were generated on a 5100P Intel® Xeon Phi™ coprocessor with MPSS Gold
60 cores @ 1.053 GHz and 8 GB of GDDR5 RAM @ 2.75 GHz

Conclusions

- Proprietary interconnects scale better
- Code scales well in both MPI and OpenMP dimensions
- Large jump in performance from Opterons to Xeons
- With no optimizations on Xeon Phis, code is still doing respectably

Future Work

- Investigate scaling anomalies
  - Look into leveling memory bandwidth playing field
  - Work on collective communication bottleneck
- Try out early experimental OpenACC work done on GYRO
- Optimize code for Xeon Phi
  - Vectorization will allow scaling to 10s/100s of threads



WORLD RECORD!
"Beacon" at NICS

Intel® Xeon® + Intel Xeon Phi™
Cluster
First to Deliver
2.499 GigaFLOPS / Watt
71.4% HPL efficiency
#1 on current Green500

Acknowledgements

R. Glenn Brook, Director AACE

John Candy, General Atomics, GYRO author

Jim Jeffers, James Reinders, and everyone at Intel

Contact Information

Vincent C. Betro, Ph.D.

Computational Scientist

National Institute for Computational Sciences/

Application Acceleration Center of Excellence

vbetro@tennessee.edu

Mark R. Fahey, Ph.D.

Deputy Director, National Institute for Computational Sciences

Industrial and Information Engineering Department

mfahey@tennessee.edu

R. Glenn Brook, Ph.D.

Chief Technology Officer

Director, Application Acceleration Center of Excellence

Joint Institute for Computational Sciences

University of Tennessee & Oak Ridge National Laboratory

glenn-brook@tennessee.edu