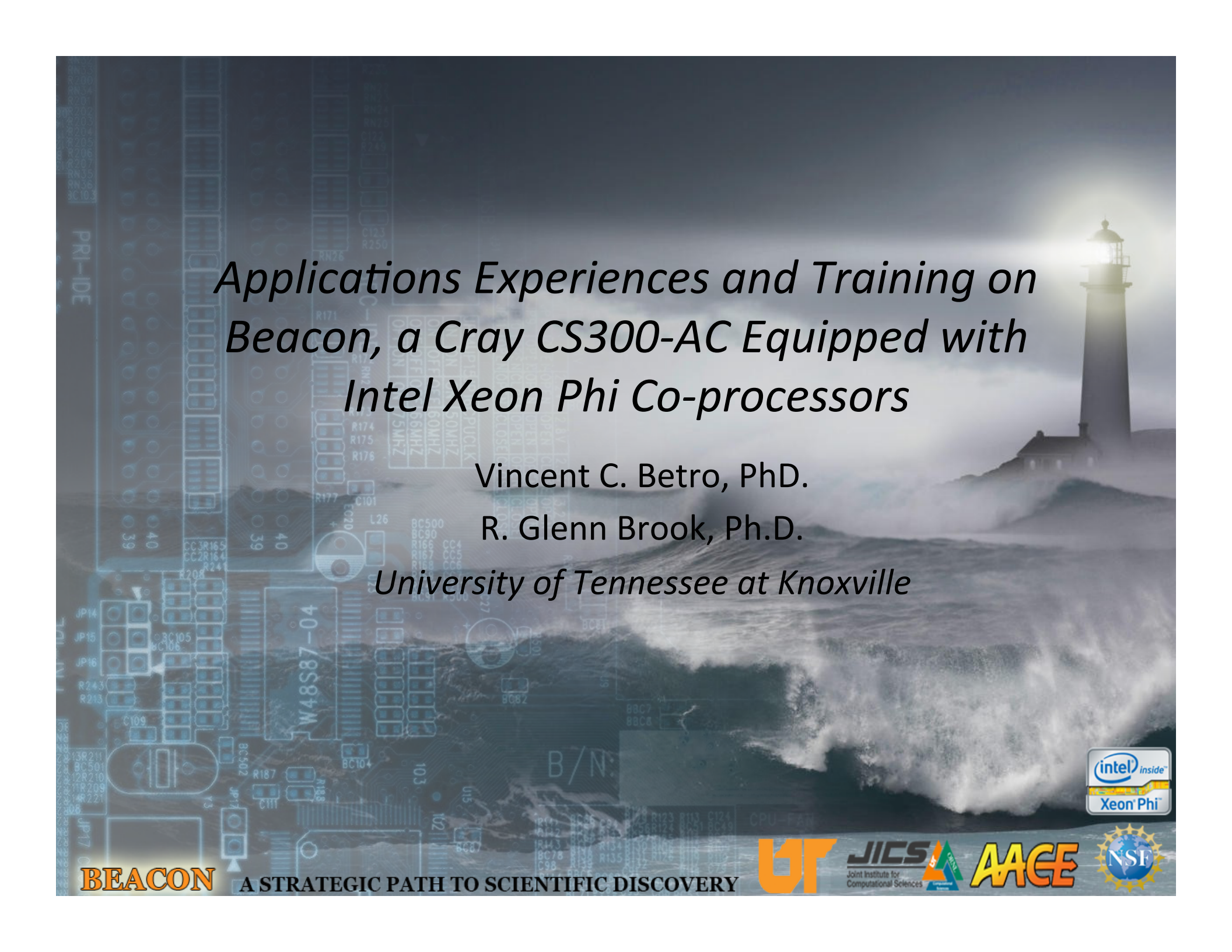




Applications Experiences and Training on Beacon, a Cray CS300-AC Equipped with Intel Xeon Phi Co-processors

Vincent C. Betro, PhD.

R. Glenn Brook, Ph.D.

University of Tennessee at Knoxville



BEACON

A STRATEGIC PATH TO SCIENTIFIC DISCOVERY



The Application Acceleration Center of Excellence (AACE)

- Established by the University of Tennessee in 2011 alongside the National Institute for Computational Sciences (NICS) within the Joint Institute for Computational Sciences (JICS) at Oak Ridge National Laboratory (ORNL)
- Tasked with exploring the application of emerging computing technologies to the acceleration of science and engineering
- Facilitates strategic progress on several fronts by acting to establish a feedback loop between the scientific community, industry partners, and operations research.
- **An essential element of a sustainable software infrastructure for simulation in science and engineering**

Managed by UT-Battelle
for the U.S. Dept. of Energy

BEACON

A STRATEGIC PATH TO SCIENTIFIC DISCOVERY



AACE accomplishments and the Beacon Project

- **March 2011** – Founded in partnership with NICS
- **Spring 2011** – Established multi-year strategic engagement with Intel
- **Fall 2011** – Deployed first academic cluster of Knights Ferry (KNF) outside of Intel
- **Fall 2011** – Demonstrated ported scientific libraries and applications running on KNF at SC11
- **Spring 2012** – Received \$1M award from NSF to support continued porting and optimization of scientific codes through the Beacon Project
- **Summer 2012** – Upgraded Beacon-dev to pre-production Knights Corner (KNC) coprocessors
- **Summer 2012** – Demonstrated scientific applications running in multiple modes on KNC at ISC 2012
- **Fall 2012** – Received additional funding from UT to replace the original Beacon with a significantly larger cluster equipped with Intel Xeon Phi coprocessors
- **Fall 2012** – Engaged Appro / Cray and Intel in a strong collaboration that led to Beacon ranking #1 on the November 2012 Green500 List



Beacon

A strategic path to scientific discovery

Moving scientists, users, software vendors and hardware manufacturers forward towards emerging HPC architectures

Original Beacon cluster by Appro / Cray	
Nodes	2 service, 16 compute
CPU model	Intel Xeon E5-2670
CPUs per node	2 8-core, 2.6 GHz
RAM per node	64 GB
SSD per node	80 GB
Intel® Xeon Phi™ coprocessors per node	2 x pre-production
Cores per Intel® Xeon Phi™ coprocessor	50+
RAM per Intel® Xeon Phi™ coprocessor	8 GB GDDR5

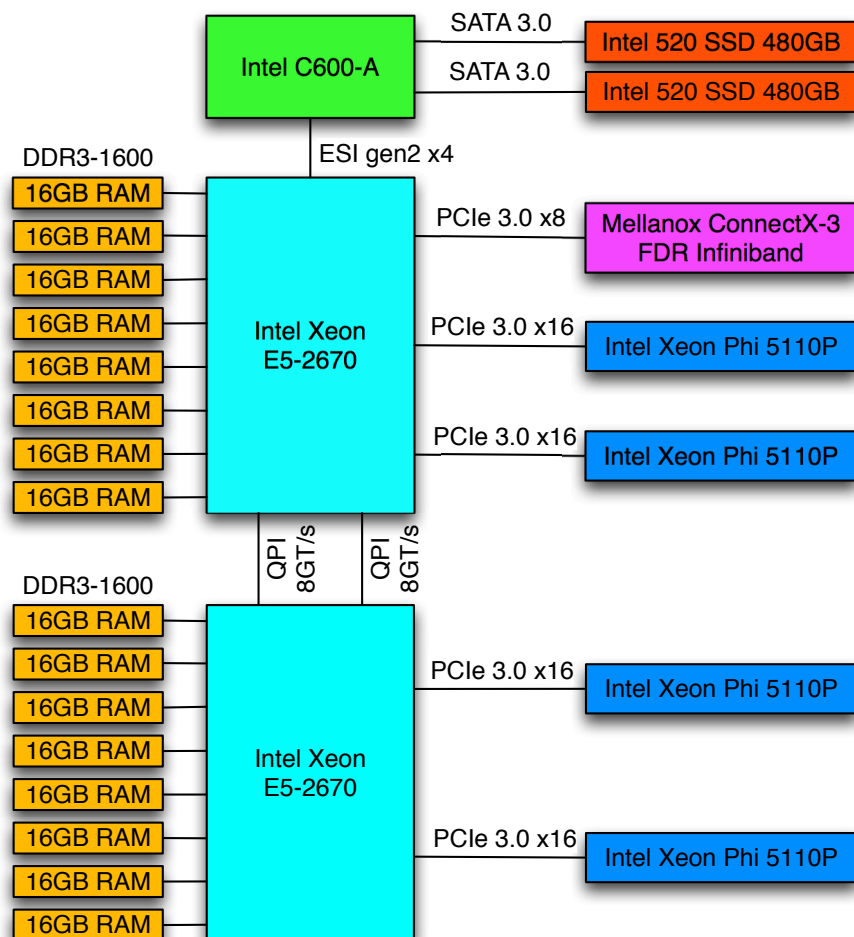
Full Beacon cluster by Appro / Cray	
Nodes	4 service, 6 I/O, 48 compute
CPU model	Intel Xeon E5-2670
CPUs per node	2 8-core, 2.6GHz
RAM per node	256 GB
SSD per node	960 GB
Intel® Xeon Phi Coprocessors per node	4 x 5110P
Cores per Intel® Xeon Phi™ coprocessor	60
RAM per Intel® Xeon Phi™ coprocessor	8 GB GDDR5



Beacon

A strategic path to scientific discovery

Next-generation HPC environment



- Provide to the scientific community, today, with the next generation of HPC architectures.
- Catalyze application deployment readiness for crucial scientific codes
- Impact future HPC roadmaps, via feedback and user experiences, with current toolsets and exemplar scientific codes.

Science Story : MAGMA MIC

LAPACK for the Intel Xeon Phi Coprocessors

MAGMA MIC : Innovative Computing Laboratory, University of Tennessee, Knoxville
J. Dongarra, M. Gates, A. Haidar, Y. Jia, K. Kabir, P. Luszczek and S. Tomov

Methodology

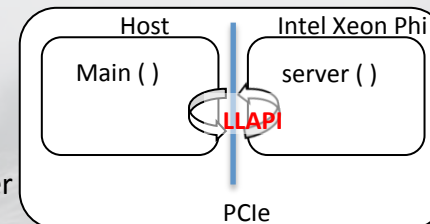
- Hybrid algorithms
 - Use both CPUs and Coprocessors
- Hardware capability task distribution
 - Panel is best for CPU
 - Updates on Coprocessors

Programming Model

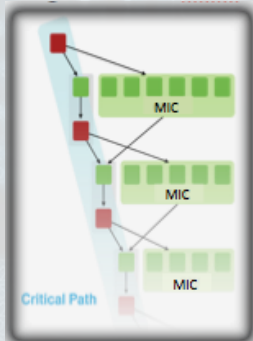
Hybrid algorithms : offloading work to MIC

Two approaches for offload with unified APIs are developed

- Server based
 - Using LLAPI
- Pragma offload
 - Intel Compiler



Algorithms as DAGs



Example algorithm:
Two-phase factorization of
 $A = [P_1 P_2 \dots]$ with **lookahead** :

```
PanelStartReceiving on CPU( $P_1$ ) ;
for  $P_i = P_1, P_2, \dots$  do
    PanelFactorize on CPU( $P_i$ ) ;
    PanelSend to MIC( $P_i$ ) ;
    TrailingMatrixUpdate on MIC( $P_{i+1}$ ) ;
    PanelStartReceiving on CPU( $P_{i+1}$ ) ;
    TrailingMatrixUpdate on MIC( $P_{i+2}, \dots$ ) ;
```

Two approaches for scheduling

- Static
- Light-weight runtime (QUARK)

Azzam Haidar, Piotr Luszczek, Stanimire Tomov, and Jack Dongarra, *Heterogenous Acceleration for Linear Algebra in Mult-Coprocessor Environments*, UTK Technical report UT-EECS-14-724, February, 2014.



Science Story : MAGMA MIC

LAPACK for the Intel Xeon Phi Coprocessors

Functionality [MAGMA MIC 1.1 <http://icl.cs.utk.edu/projectsfiles/magma/mic/magmamic-1.1.0.tar.gz>]

- Linear system factorizations and solvers
- Two-sided factorizations
- Eigenproblem solvers and SVD

Performance & Scalability

Beacon system

48 Nodes, each node with:

CPU: 2x8-core Intel Xeon E5-2670

[16 cores @ 2.6 GHz

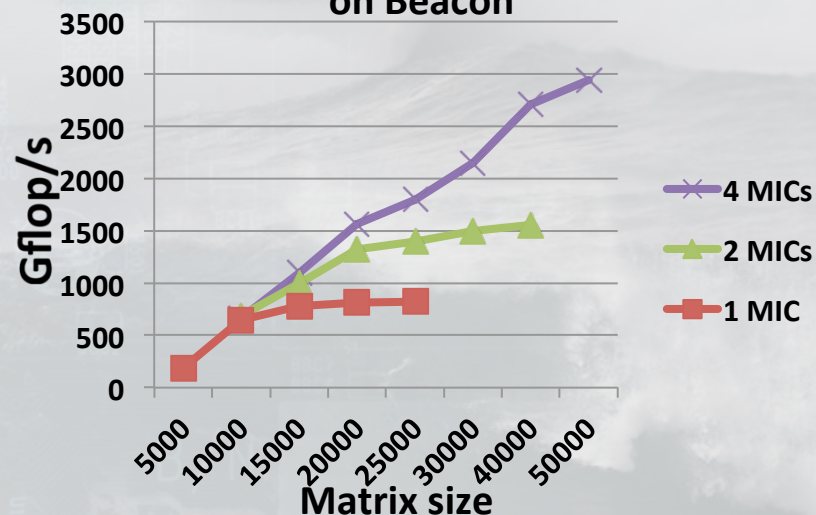
[DP Peak: $16 * 2.6 * 8 = 332 \text{ Gflop/s}$]

MIC: 4 Intel Xeon Phi 5110P

[60 cores @ 1.053 GHz, 1 core for OS

DP peak: $4 * 59 * 1.053 * 16 =$
3,976 Gflop/s]

MAGMA MIC LU in Double Precision
on Beacon



Science Story : HSP-BLAST

The HSP-BLAST approach on the BEACON cluster

HSP-BLAST : Joint Institute for Computational Sciences, Univ. of Tennessee, Knoxville
B. Rekepalli

- The **B**asic **L**ocal **A**lignment **S**earch **T**ool (BLAST) is an integral part of the Bio-informatics tool chain. The tool searches for local alignments in protein or nucleotides in order to determine a position of a novel sequence in the tree of life.
- The field of genomics is currently undergoing exponential growth. Databases of known sequences are growing much faster. This leads to substantial growth in search times.
- One of the primary difficulties in the current software tool chain is the lack of **scalability** of the approaches
- Based on the preliminary work of Dr. Bhanu Rekepalli : “HSPp-BLAST: Highly Scalable Parallel psi-BLAST for Very Large-scale Position Iterative Sequence Searches”

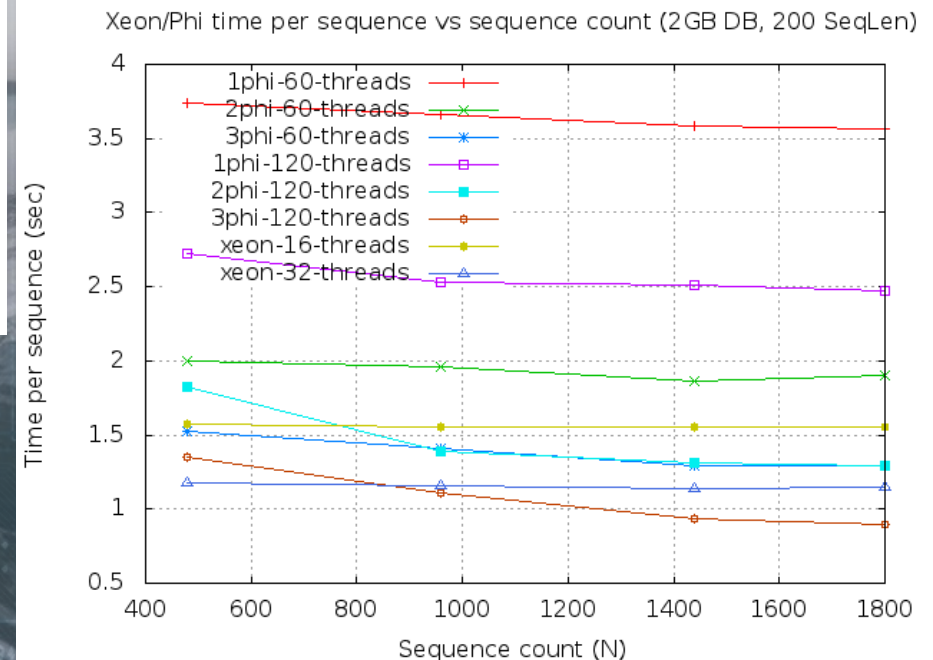
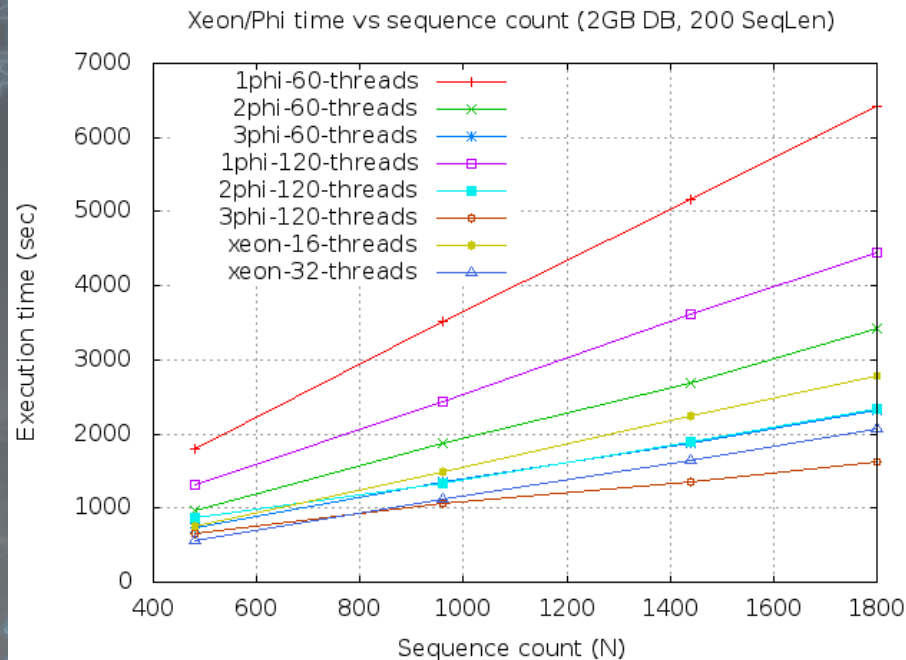


Science Story : HSP-BLAST

The HSP-BLAST approach on the BEACON cluster

Approach

- A database of sequences is distributed
- Queries are then run against the distributed database in a scalable fashion



Science Story : HPC-BLAST

The HPC-BLAST approach on the BEACON cluster

HSP-BLAST : Joint Institute for Computational Sciences, Univ. of Tennessee, Knoxville

R. Brook, B. Rekepalli, S. Sawyer

- funded by Intel through Intel Parallel Computing Center at JICS/UTK
- hybrid, parallel task-centric implementation of NCBI-BLAST based on the BLAST+ C++ toolkit.
- A hierarchical framework of task decomposition along the subject database and query set
- Emphasizes inter-thread cooperation rather than inter-process
- In less than a year of development, performance is on par with the mature mpiBLAST implementation.



BEACON

A STRATEGIC PATH TO SCIENTIFIC DISCOVERY

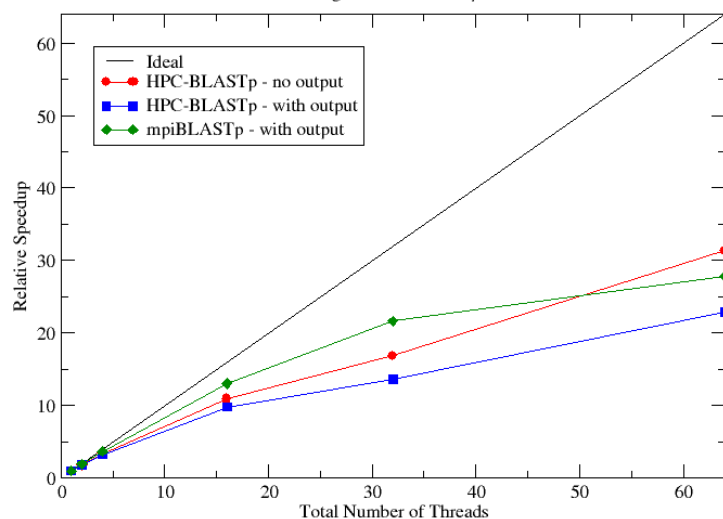


Science Story : HPC-BLAST

The HPC-BLAST approach on the BEACON cluster

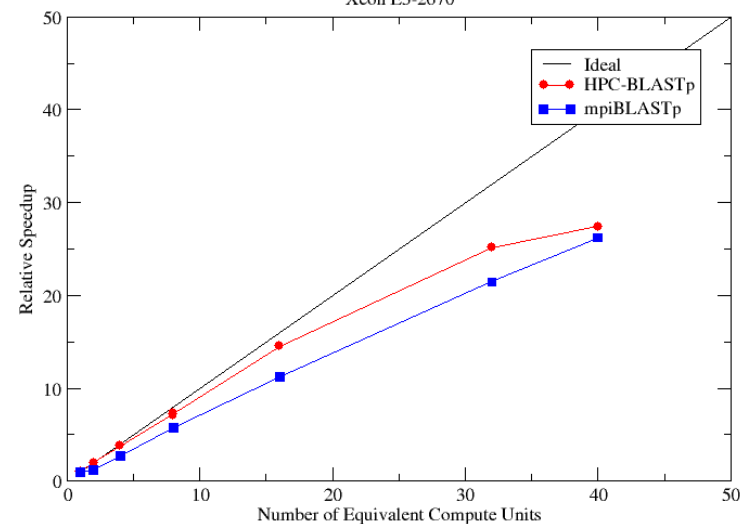
HPC-BLASTp Strong Scaling with SwissProt

Single Xeon Phi 5110p



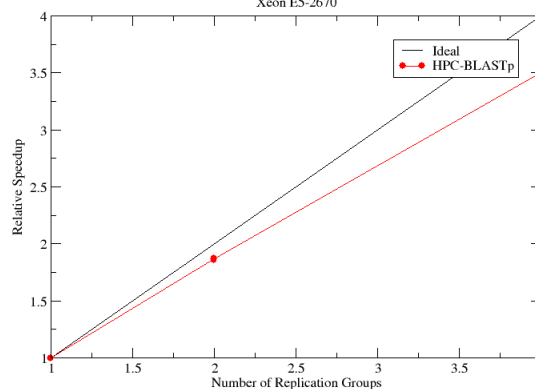
HPC-BLASTp Strong Scaling with SwissProt

Xeon E5-2670



HPC-BLASTp Strong Scaling with NR

Xeon E5-2670



BEACON

A STRATEGIC PATH TO SCIENTIFIC DISCOVERY



Science Story : MIC @ MUC

MIC porting and optimization at Technische Universität München and Leibniz Supercomputing Centre

M. Bader, G. Brietzke, A. Breuer, A. Heinecke, W. Eckhardt, S. Rettenberger

Governing Hardware-Concepts:

- Many levels of task parallelism:
 - thread/core
 - sockets
 - node
 - rack/island/leaf
 - full machine
- Data parallelism and local storage:
 - SIMD/SIMT
 - caches/scratchpad
- Heterogeneity
 - CPUs + GPUs
 - CPUs + coprocessors

Optimized Applications:

- SeisSol
 - novel simulation code for earthquake scenarios
 - element-local operations boil down to (sparse-)DGEMMs
- LS1/Mardyn
 - molecular dynamics simulation for chemical engineering
 - N-body solved by Linked-Cells
- SG++
 - direct discretization of high-dimensional Spaces ($d > 4$)
 - used in data mining and computation finance



Science Story : Pseudo spectral code benchmarking on the MIC

B.K. Muite, B. Cloutier, P. Rigge

- FFT methods can be very efficient in solving partial differential equations
- High degrees of global communication make these problems difficult for distributed architectures
- Goal : Compare FFT methods across CPU, GPU and Intel Xeon Phi architectures

Model: Time dependent
Schrödinger equation
Technique: second order
splitting scheme

$$iu_t + u_{xx} + u_{yy} = |u|^2 u$$

Results : time to completion for 20 time steps of size $dt = 10E-3$.

M2070 : CUDA Fortran

Xeon Phi Coprocessor: Intel Fortran

*The Xeon Phi Coprocessor code was fully optimized

Grid Size	Xeon Phi	Nvidia M2070
256x256	0.0122	0.00802
512x512	0.0291	0.0234
1024x1024	0.118	0.0851
2048x2048	0.422	0.334
4096x4096	1.626	1.49

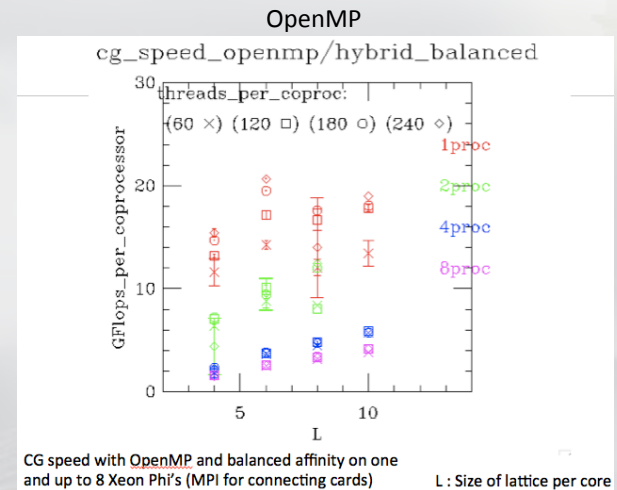
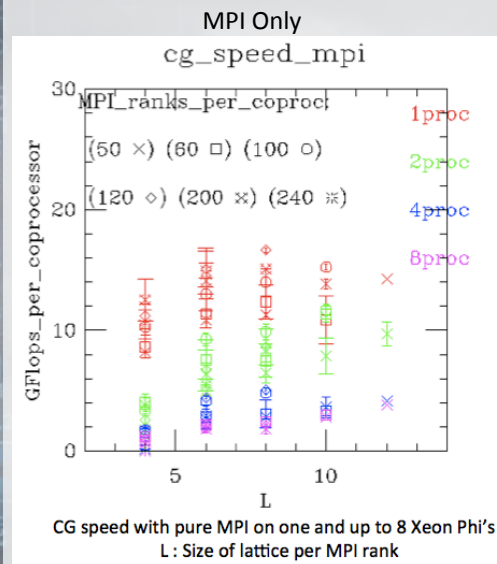


Science Story : MILC and DSLASH

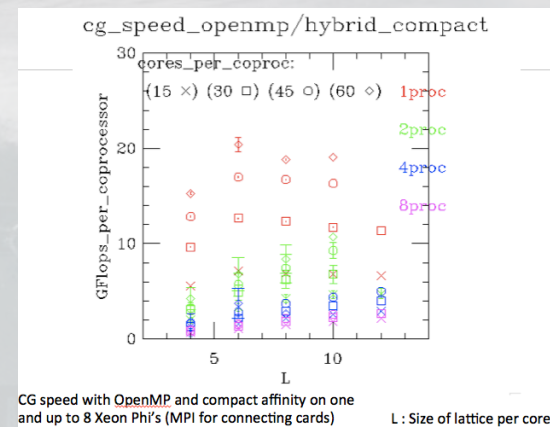
S. Gottlieb, B. Joo, and R. Li

- Two codes tested on the Intel Xeon Phi
 - MILC (Quantum Chromodynamics)
 - Pure MPI version
 - Threaded (OpenMP) version
 - Hybrid (MPI + OpenMP)
 - Dslash
 - Vectorized code on one Xeon Phi with OpenMP

Results : MILC



Hybrid MPI+OpenMP



Science Story : Oak Ridge Leadership Computing Facility (OLCF)

B. Messer, J. Hill, E. D'Azevedo, R.G. Brook

Development of MiniApps for performance testing at the Oak Ridge Leadership Computing Facility

Project Goals:

- We will formulate a set of ORNL-owned MiniApps and associated tools and use these in a series of performance studies.
- We will produce versions of each MiniApp using several specific programming models (e.g., OpenACC, CUDA, OpenMP).
- This suite of MiniApps will become the heart of a new performance testing harness for HPC.
- We will use the resultant MiniApps to conduct a study of performance portability across both programming models and hardware platforms.
- We will concentrate on ensuring that the modeled performance of the derived MiniApps is congruent to the production application performance.
 - Can the MiniApp be divorced from the underlying physics being modeled in the main application and still be an effective proxy?
- The same MiniApp implemented with multiple programming models on multiple hardware will also be compared.
- A primary research outcome of this task will be findings regarding performance portability, which will likely be applicable to many other codes



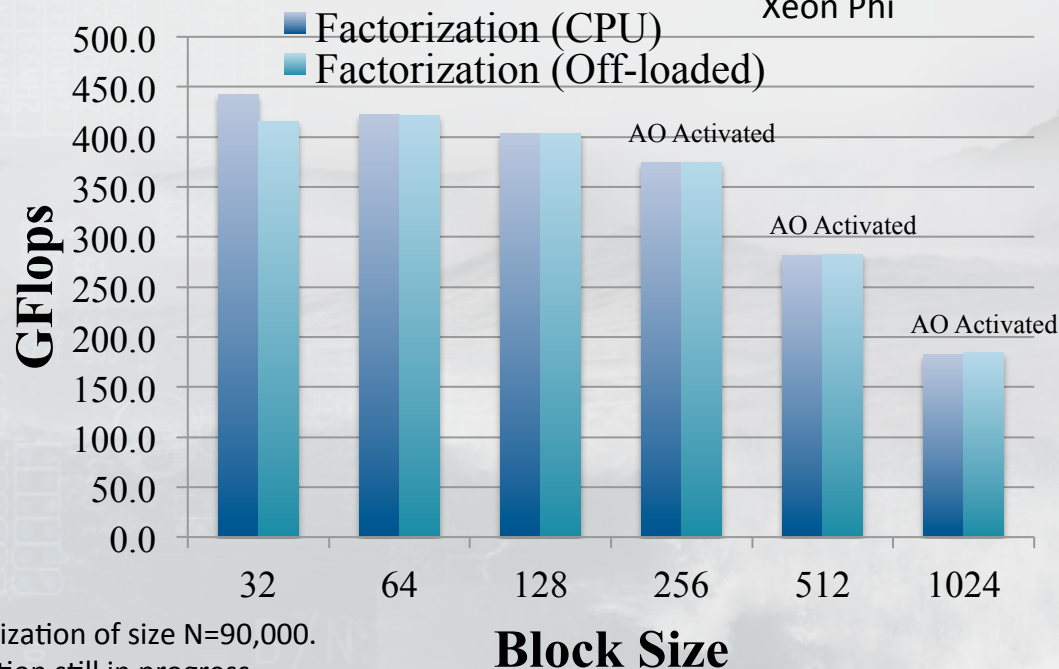
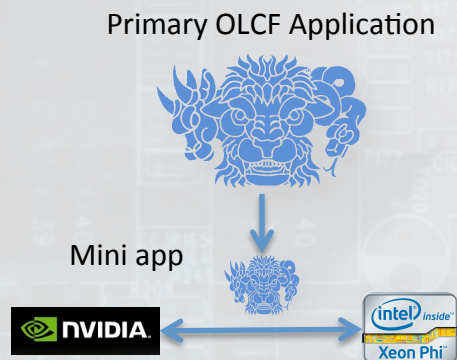
Science Story : Oak Ridge Leadership Computing Facility (OLCF)

B. Messer, J. Hill, E. D'Azevedo, R.G. Brook

Development of MiniApps for performance testing at the Oak Ridge Leadership Computing Facility

- Each of the Mini-Apps will be ported to the Intel many integrated core architecture
- The porting and optimization will occur on BEACON

AORSA MiniApp
Performance: Intel
Xeon Phi



Observations:
AORSA mini app on Beacon

- Preliminary results shown for a factorization of size $N=90,000$. Scalability comparisons to full application still in progress
- Significant work towards performance is underway



Science Story : VisIt

S. Ahern, H. Childs, J. Huang

Development of VisIt for tight integration with the Xeon Phi at AACE

- Two primary foci of brand new IPCC project:
 - Lay the groundwork for efficient data pipeline execution on the many cores of the Phi. Hand optimize common filters.
 - Develop highly parallel rasterization/rendering on Phi. Leverage Intel technologies where appropriate. Explore parallel compositing solutions.

THE UNIVERSITY of
TENNESSEE 
KNOXVILLE


UNIVERSITY
OF OREGON



BEACON

A STRATEGIC PATH TO SCIENTIFIC DISCOVERY



Science Story : GROMACS

R. Schulz, J. Eblen, J. Smith

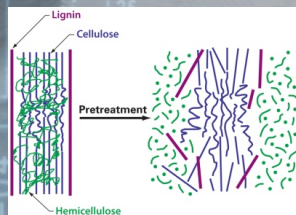
- MD is used to understand biological function on the atomic scale: e.g. protein function, or recalcitrance

Cellulosic Biofuel

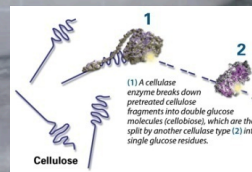
Lignocellulosic Biomass



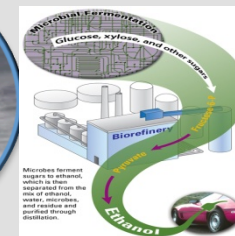
→ **Pre-treatment** →



→ **Enzymatic Hydrolysis** →



→ **Fermentation**



bottleneck: **biomass recalcitrance**
plant's resistance to deconstruction



SIMD in Gromacs (1/2)

- Gromacs has optimized SIMD for SSE2, SSE4.1, AVX, AVX-FMA, AVX2, QPX (Bluegene), Phi
- Requires abstraction to be manageable
- Different SIMD dialects are abstracted, e.g.

```
#define gmx_simd_load_f          _mm512_load_ps
#define gmx_simd_load1_f(m)     _mm512_extload_ps(m,
_MM_UPCONV_PS_NONE, _MM_BROADCAST_1X16,
_MM_HINT_NONE)
```

- 230 functions for basics (double, single, full width, 4 width, arithmetic, load-store, compare and blend, transcendentals)
- Adding support for an extra architecture requires only adding the mapping



SIMD in Gromacs (2/2)

- Unroll factors are a preprocessor constants and depends on SIMD width (SIMD width is obviously not abstracted)
- Preprocessor is used if some details of the algorithm depends on the SIMD width or the unroll factors
- Documented and OpenSource. Questions and Contributions welcome: www.gromacs.org



Gromacs on Phi

- Porting Gromacs to Phi only required one source line change (inline asm for memory fence)
- Development tools need some improvement e.g.
 - Compiler: bug and missing obvious optimizations (prefetch)
 - Vtune: GUI very slow, incorrect asm-source mapping
 - Inspector: False positives for simple OpenMP loops
 - Limited 3rd party tools: e.g. valgrind or address-sanitizer
- Offload isn't working yet. Progress slow because of the missing support for offloading complex data structures (e.g. pointers to pointers)
- Implemented SIMD for Phi
- Fast tree reduction with pair-wise synchronization



GROMACS Performance

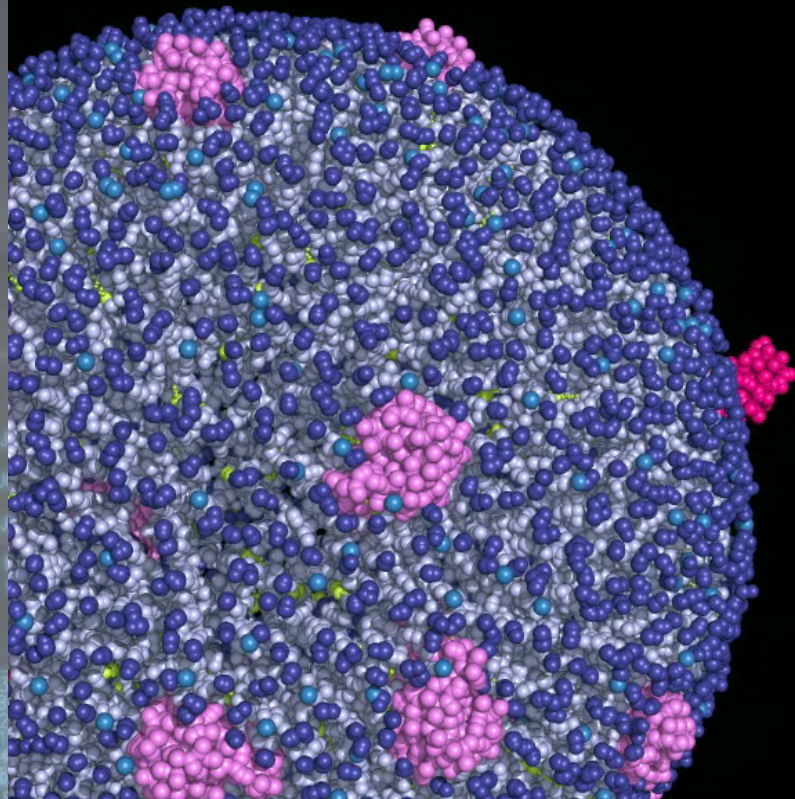
- SIMD implementation improved overall performance by 3.9x
- The performance of the most compute intensive n-body non-bonded section was improved by 8x
- The tree-reduction has improved the time for force reduction by 1.9x
- Force calculation 60% faster than single E5-2670 and 17% slower than two CPUs
- Not all code in Gromacs is vectorized yet and has some scalability issues for small inputs
- For relative large input 25% slower than 2 E5-2697
- Offloading just the force calculation should give good performance



Science Story : Protein-in-Water

GROMACS in use at Mount Sinai

Understanding How Membrane Proteins Mediate Cellular Functions Leading to Disease



Simulations used to discover how membrane protein complexes process cell signaling so that we can make drugs more specific and design more efficient medicines (e.g. non-addictive painkillers)

Implemented in Gromacs

Johnston, J.M., Wang, H., Provasi, D., Filizola, M.
"Assessing the Relative Stability of Dimer Interfaces in G Protein-Coupled Receptors." *PLOS Computational Biology* (2012) 8(8):e1002649.

Computational
Scientist:
A. Costa

- J. Johnston
- H. Wang
- D. Provasi
- M. Filizola



Science Story : Protein-in-Water

GROMACS in use at Mount Sinai

Research Towards Non-Addictive Painkillers

Use newly available structural information of opioid receptors in computer simulations to discover/design novel painkillers with reduced abuse liability.

Filizola & Devi, "How opioid drugs bind to receptors" *Nature* (2012) 485: 314-317

Filizola & Devi, "Grand Opening of Structure-Guided Design for Novel Opiates" *TIPS* (2013) 34(1):6-12

Negri, A. et al. "Discovery of a Novel Selective Kappa-Opioid Receptor Agonist Using Crystal Structure-Based Virtual Screening" *J.Chem. Inf. & Model.* (2013) 53: 521-526

Computational
Scientist:
A. Costa

- M. Filizola
- L. Devi
- A. Negri



BEACON

A STRATEGIC PATH TO SCIENTIFIC DISCOVERY



JICS
Joint Institute for
Computational Sciences



AACE



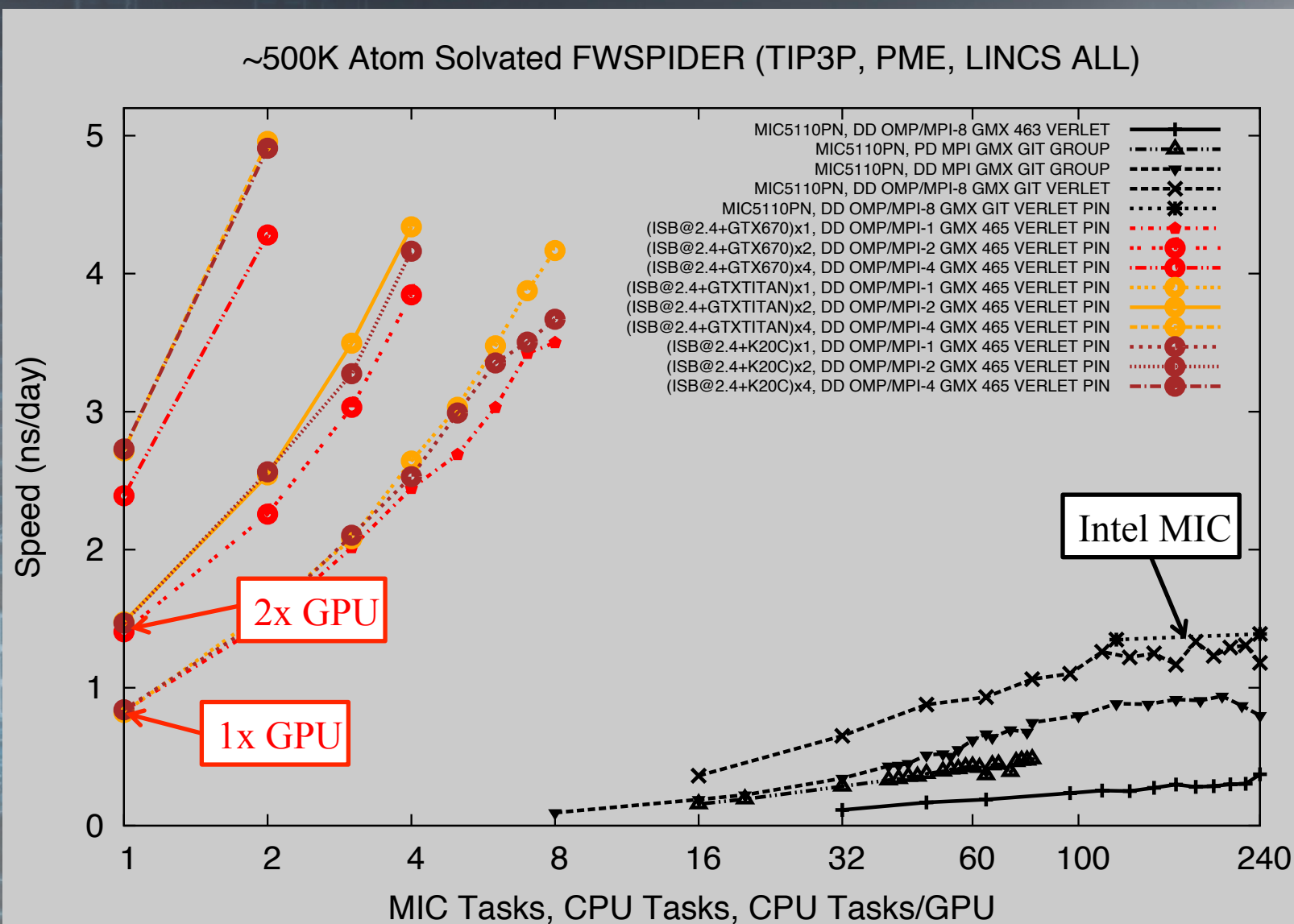
Initial work in MIC-accelerated molecular dynamics

- Last year we saw a number of attempt to investigate MD on MIC
- Native 512-wide SIMD for force evaluation, distance calculations, neighbor list building
- Work demonstrated significant speedups for short-range molecular dynamics simulations (no electrostatics)
- Sandia's miniMD benchmark a $\sim 5x$ speedup over scalar execution on an Intel Xeon E5 with excellent weak scaling.
- Xiao and coworkers demonstrated $>7x$ speedup over a single E5 core using even more fine-grained strategies.
 - This beat their equivalent implementations in a hybrid CPU-GPU code.
- Of course, we would rather have native acceleration of established packages with broad feature sets and are immediately useful to researchers.
- Existing GROMACS provides frameworks for implementing new intrinsics approach for MIC

S. J. Pennycook, C. J. Hughes, M. Smelyanskiy, S. A. Jarvis, Exploring SIMD for Molecular Dynamics, Using intel Xeon Processors and Intel Xeon Phi Coprocessors
Q. Wu, C. Yang, T. Tang, L. Xiao, MIC Acceleration of Short-Range Molecular Dynamics Simulations



Comparison to modern GPU implementations



Limitations and future steps

- For most typical protein-in-water calculation, native-mode MIC performance outperforms 1 Xeon CPU and a modern NVIDIA GPU accelerator.
- 5x Thread-MPI performance improvement in Gromacs 5.0 over 4.X MIC due to native vectorization.
- Many current limitations must be overcome to see first major adoption of MIC for MD.
 - Native mode limits peak performance dramatically.
 - No scaling to multiple MICs for large nonbonded systems.
 - PME on the device, cannot load balance electrostatics against nonbonded forces with multiple CPU MPI ranks or threads.
 - This work is ongoing, goal is general framework for nonbonded interaction offload similar to CUDA implementation but via 512-wide vector intrinsics for MIC.

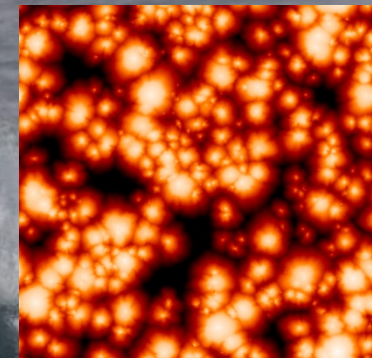
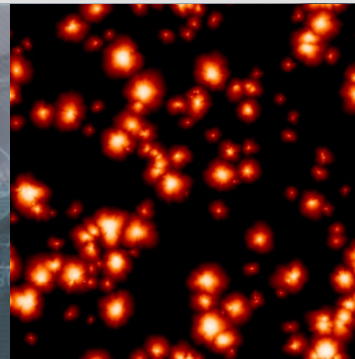
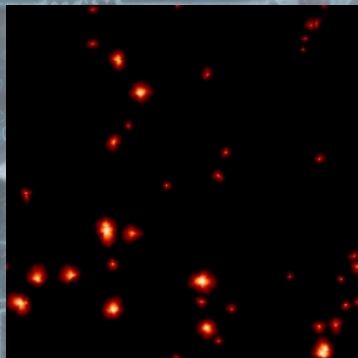


Other Codes Investigated by AACE

- 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)

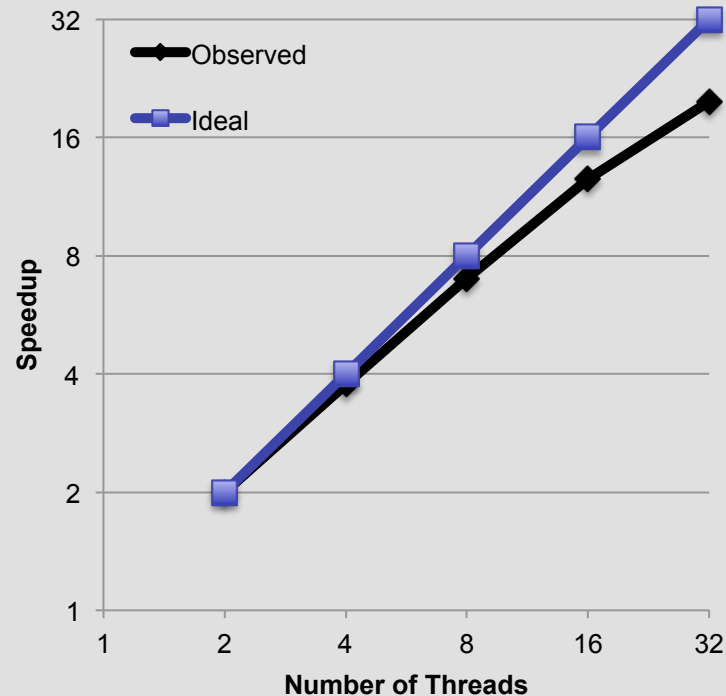
Enzo

- Community code for computational astrophysics and cosmology by Dr. Robert Harkness (MIC Port)
- More than 1 million lines of code
- Uses powerful adaptive mesh refinement
- Highly vectorized with a hybrid MPI + OpenMP programming model
- Utilizes HDF5 and HYPRE libraries
- Multiple MPI tasks per coprocessor and many threads per MPI task



Preliminary Scaling Study: Native

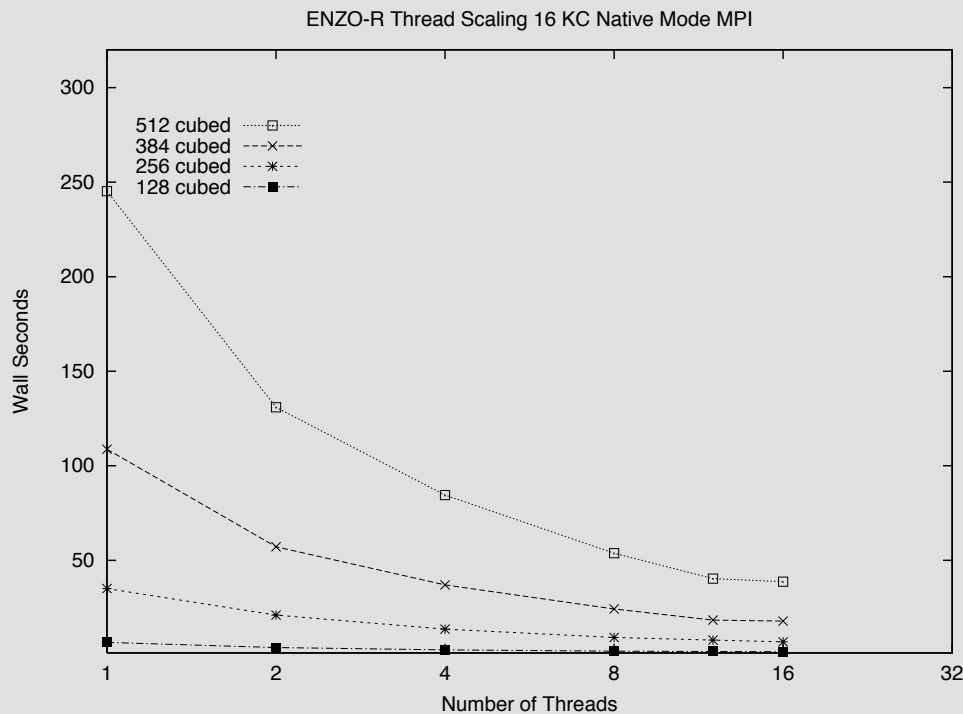
- ENZO-C
- 128^3 mesh (non-AMR)
- pure MPI
- native mode
- Fortran, C, C++



Results were generated on the Intel® Knights Ferry software development platform



Multi-KNC Scaling Study: Native



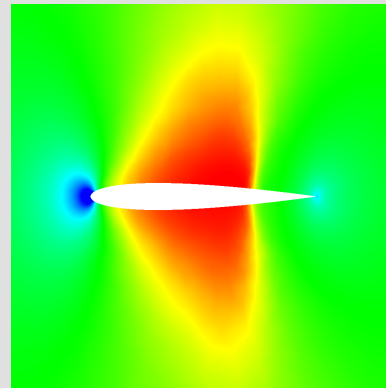
Results were generated on a Pre-Production Intel® Xeon Phi™ coprocessor
Beta software, 61 cores @ 1.09 GHz and 8 GB of GDDR5 RAM @ 2.75 GHz

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



Model Boltzmann Equation Solver

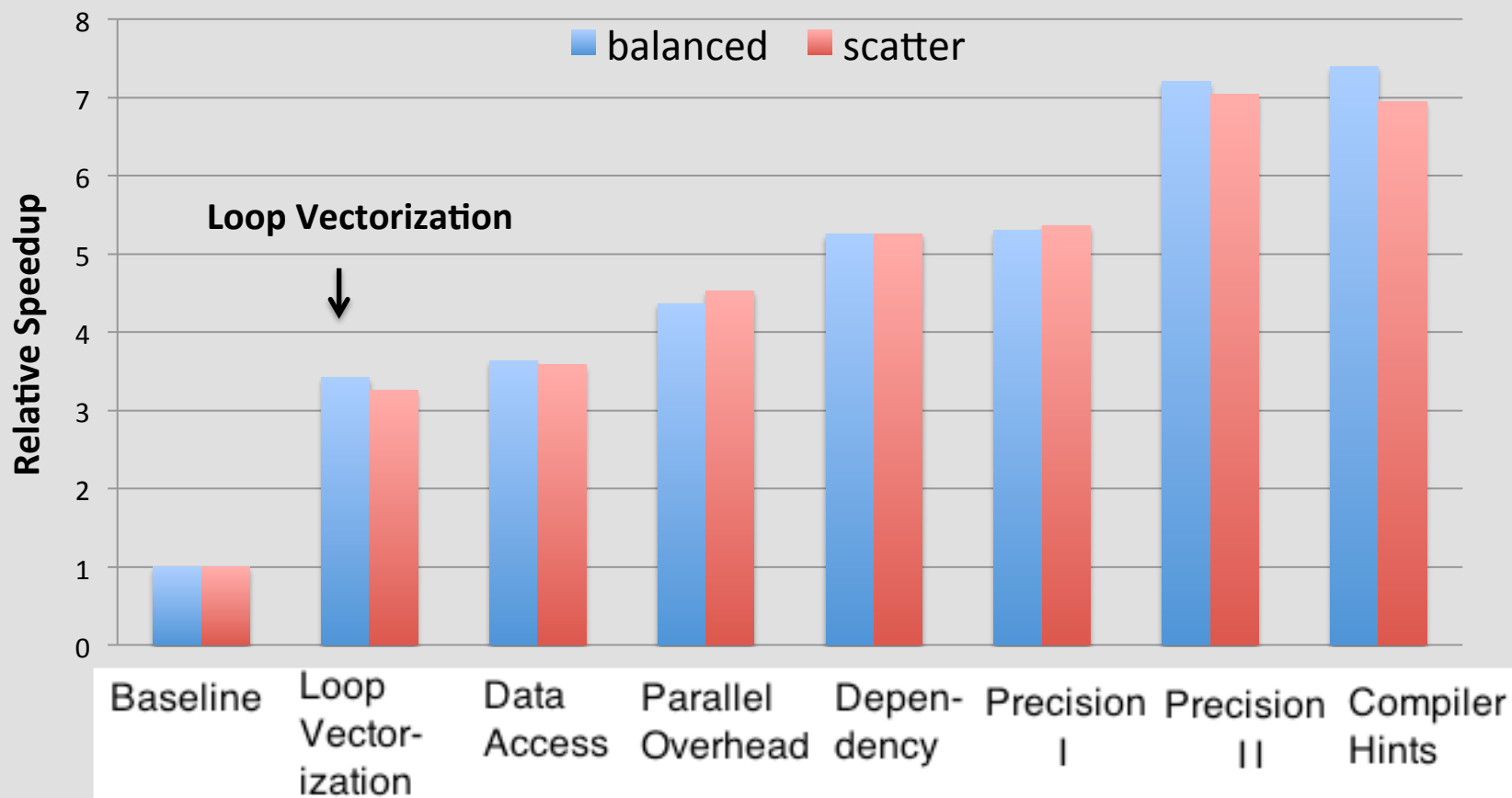
- Solves the BGK Boltzmann equations, a kinetic model for rarefied gases
- Solves and Euler problem (non-viscous), but can show viscous effects
- Gas-kinetic model allows for highly asynchronous computation and ease of vectorization



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

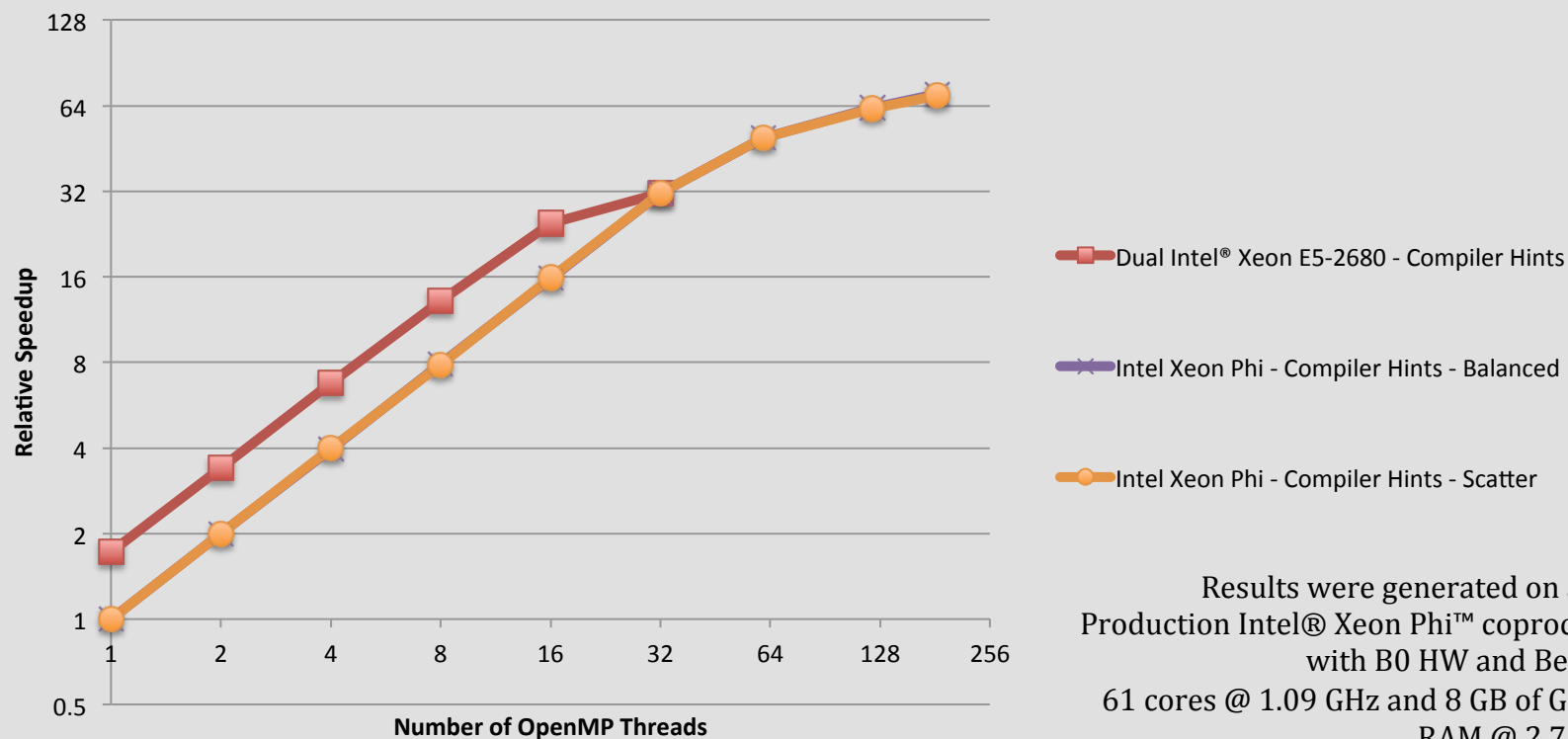


Results were generated on a Pre-Production Intel® Xeon Phi™ coprocessor
with B0 HW and Beta SW
61 cores @ 1.09 GHz and 8 GB of GDDR5 RAM @ 2.75 GHz



Model Boltzmann Equation Solver Performance

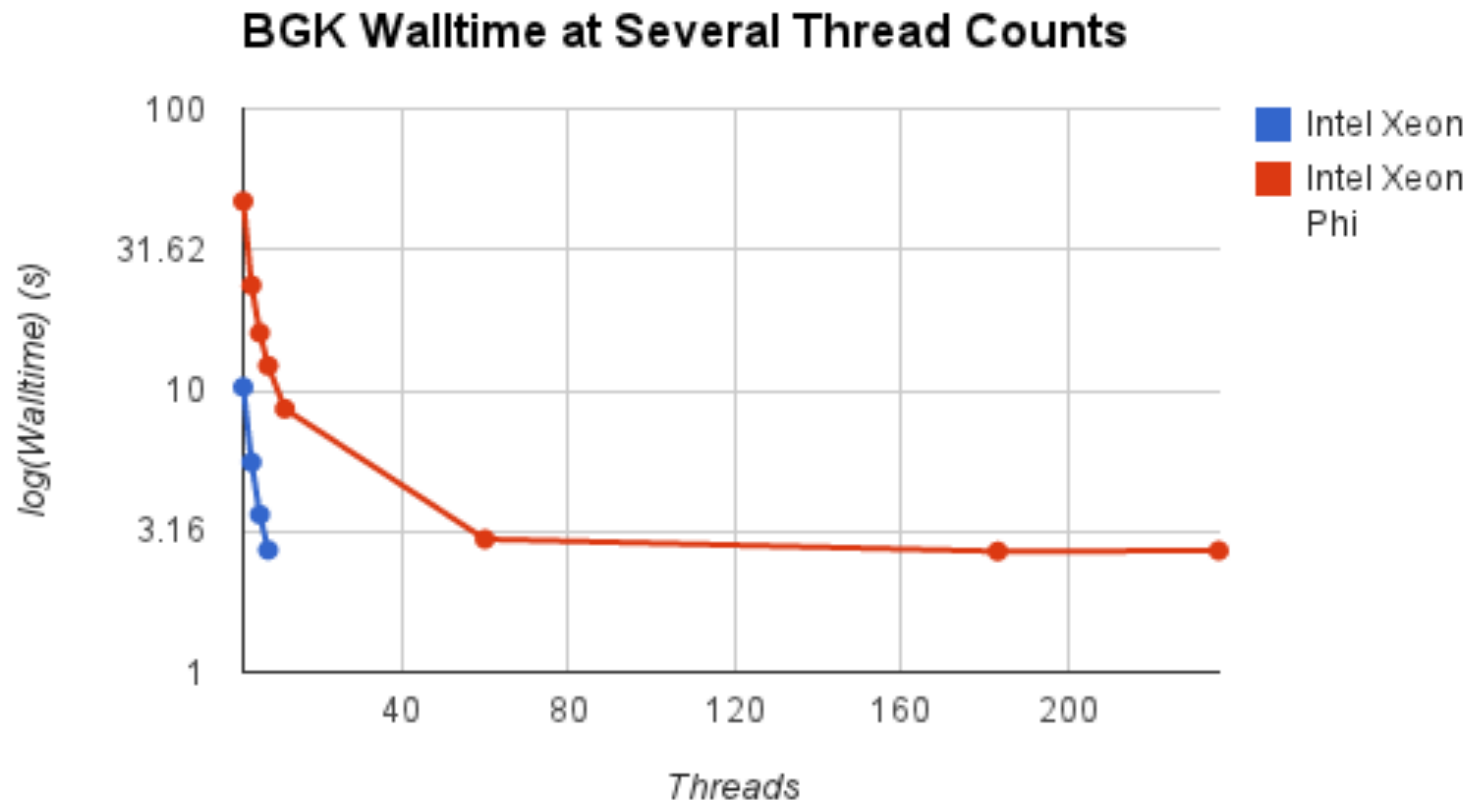
Relative Speedup of two 8-core 3.5 GHz Intel® Xeon E5-2670 Processors Versus an Intel® Xeon Phi™ Coprocessor



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



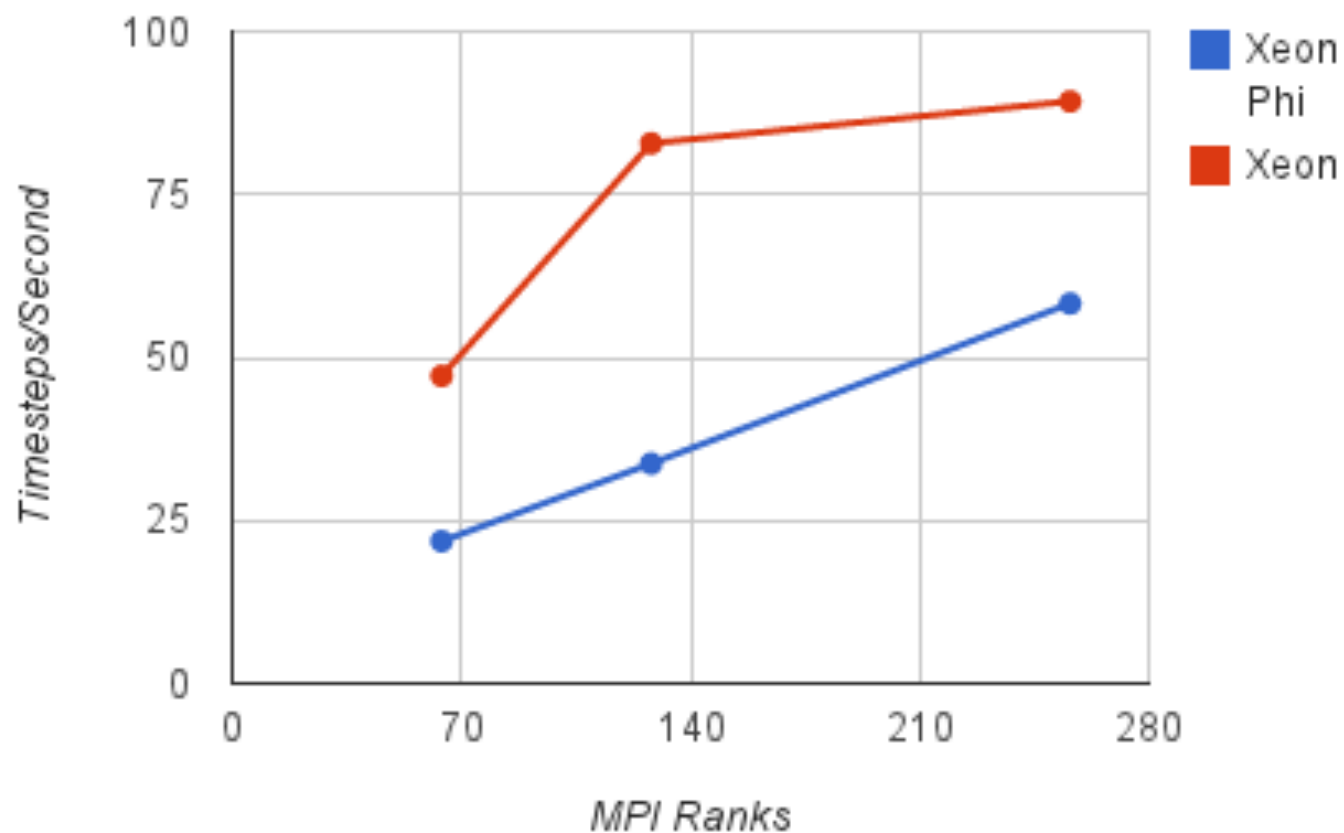
Gyro: Tokamak Plasmas

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-explicit Runge-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

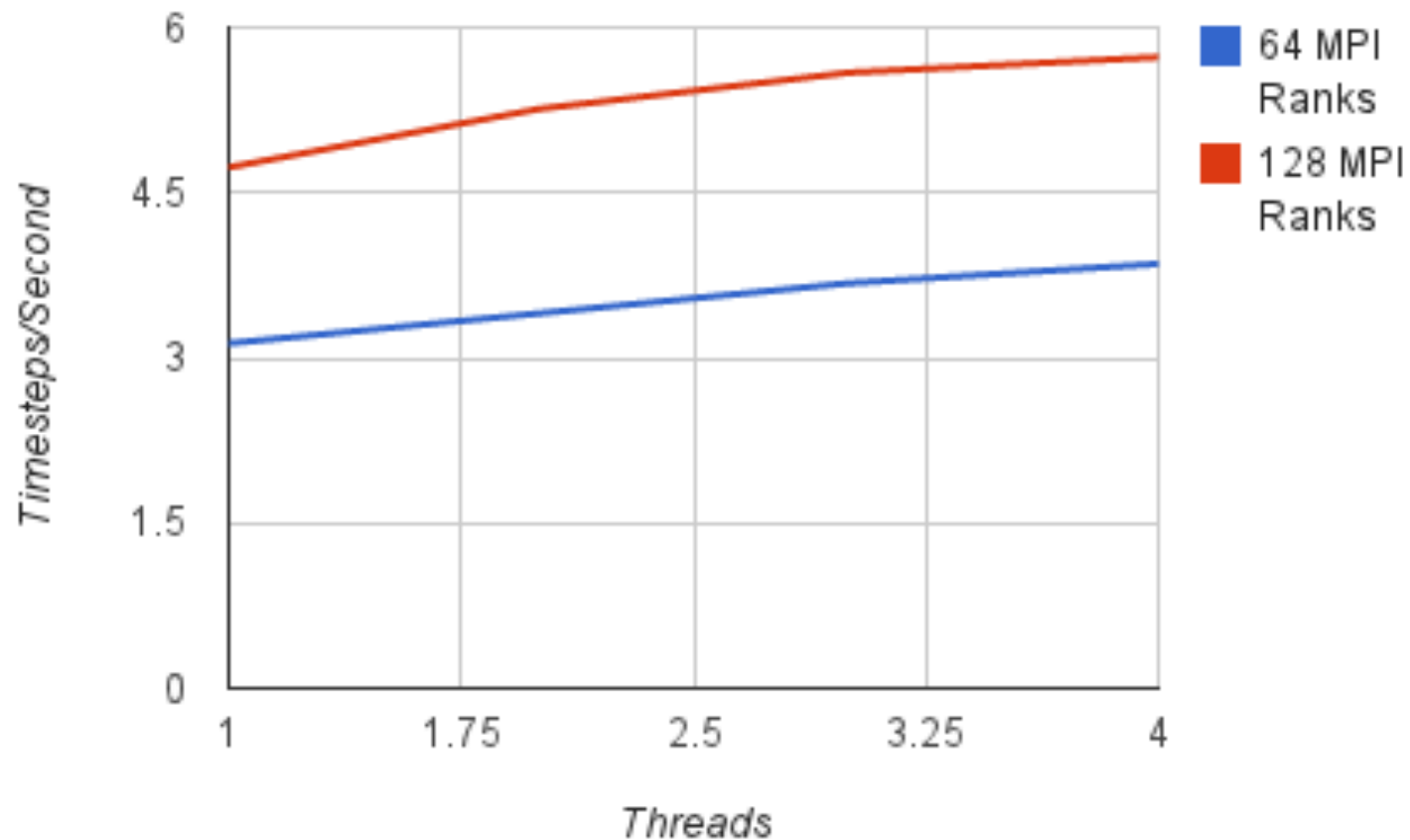
Gyro: MPI Performance

Gyro Performance on Beacon



Gyro: Threaded Performance

Gyro Performance on Beacon at Different Thread Counts



Other codes ported to the Phi

Code	MPI Ranks	Walltime	Speed Up	% Peak
HOMME-CAM	32	82.65 s	---	---
HOMME-CAM	64	54.91 s	1.51	76%
PSC	16	894.41 s	---	---
PSC	32	679.58 s	1.32	66%
AWP-ODC	32	932 s	---	---
AWP-ODC	64	524 s	1.78	89%
AWP-ODC	128	287 s	3.25	81%
TRANSIMS	1	5904 s	serial	---
ASCAPE	1	30.197 s	serial	---



Beacon

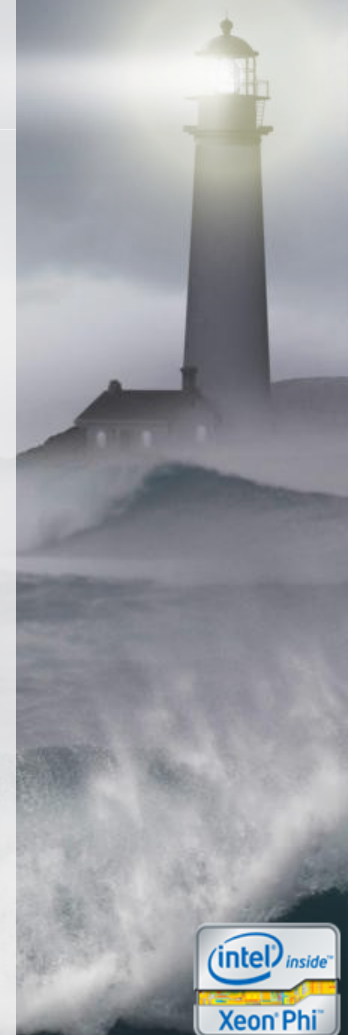
A strategic path to scientific discovery

Scientific research “lessons learned”

- Hardware-aware algorithm design allows for tremendous speedup
- Increased return on investment: Optimization techniques targeting the MIC architecture are valid for both standard CPUs and highly-parallel coprocessors
- Important to utilize all of the coprocessors architectural capabilities to achieve maximum performance.
 - Utilizing the full feature set of the MIC architecture yields significant performance but can require substantial code re-design.
 - A solid performance increase can be achieved by capturing low-hanging fruit
- User teams expressed interest in energy awareness. Profiling memory usage and developing energy aware algorithms were topics of future interest.

AACE is driven by **results**.

Our academic partners, their graduate students, and our computational scientists are the backbone of our **scientific research success**



Beacon

A strategic path to scientific discovery

Training

- The Beacon project holds at least two full-day trainings per year on taking advantage of the heterogeneous Intel Xeon Phi architecture contained in the Cray CS300-AC(TM).
- The material is created here at NICS based on experiences with users, and it is integrated with the latest, cutting edge Intel training to make for a particularly useful combination of broadly applicable power techniques and system-specific work-arounds and best practices.



BEACON

A STRATEGIC PATH TO SCIENTIFIC DISCOVERY



Beacon

A strategic path to scientific discovery

Training

- Selected Topics:
 - porting code to the Xeon Phi
 - running code in native, offload, and symmetric paradigms
 - using Intel's MKL libraries on the Xeon Phi
 - debugging on Xeon Phi
 - profiling and optimizing code for the Xeon Phi
 - working with the Xeon Phi in a cluster environment



Beacon

A strategic path to scientific discovery

Training: Best Practices and Lessons Learned

- Trainings are open to the public and are always simulcast on Webex
- They are recorded and posted for asynchronous viewing by users
- Having labs interspersed with content helps users internalize the often new concepts of offload and native modes in a meaningful way
- Optimizing for the Xeon and using Xeon-based tools is always the best place to begin
- Working with Intel to keep training fresh is crucial

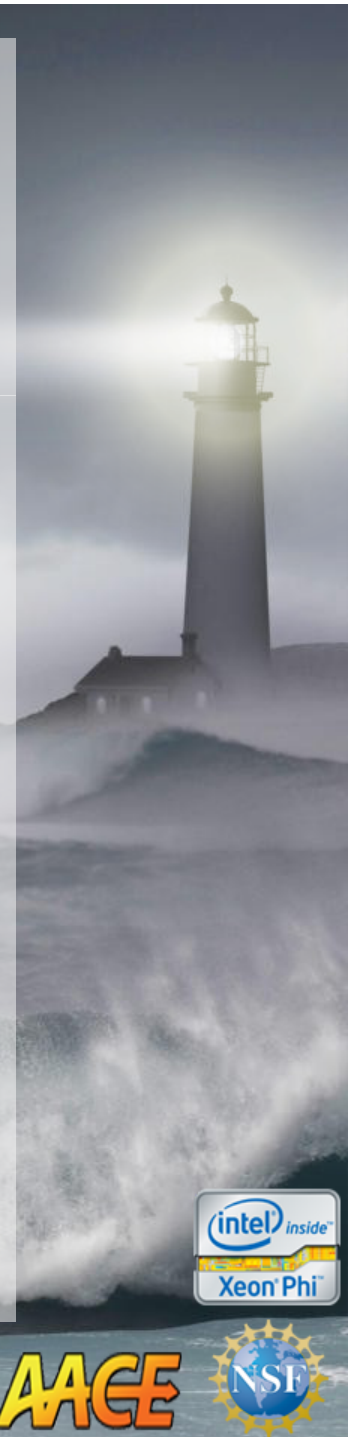


Beacon

A strategic path to scientific discovery

Conclusions

- The BEACON project is leading the way in operations research aimed at deployments of Intel MIC architectures at a range of scales
- The BEACON project has established an invaluable feedback cycle incorporating the needs of National Science Foundation researchers, industry concerns, and third party software tools providers
- The BEACON project is engaged. Our affiliated researchers are leading the way in providing highly threaded, vectorized codes and capability to the National Science Foundation community
- The BEACON project is increasingly facilitating scientific discovery at scale through both the resource itself and training



BEACON

A STRATEGIC PATH TO SCIENTIFIC DISCOVERY



Beacon

A strategic path to scientific discovery

THANK YOU!

For more information about the **Beacon** project, the Application Acceleration Center of Excellence (AACE) or the Joint Institute for Computational Sciences (JICS) please contact:

Dr. R. Glenn Brook at glenn-brook@tennessee.edu

or

Dr. Vincent C. Betro at vbetro@tennessee.edu

BEACON

A STRATEGIC PATH TO SCIENTIFIC DISCOVERY

