

Trinity Center of Excellence

*I can't promise to solve all your problems,
but I can promise you won't face them alone*



Hai Ah Nam
Computational Physics & Methods (CCS-2)

Presented to:
Salishan Conference on High Speed Computing
April 25-27, 2017



Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

Trinity

Advanced Technology System (ATS-1)

Trinity Cray XC40 Specifications

Intel Xeon E5-2698v3 "Haswell"	Intel Xeon Phi 7250 "Knights Landing"
9436 nodes	9984 nodes
Dual socket, 16 cores/socket, 2.3 GHz	1 socket, 68 cores, 1.4 GHz, > 3 Tflops/KNL
128 GB DDR4	96 GB DDR4 + 16GB HBM
1.15 PB on-node memory	1.12 PB on-node memory



#6 on Top500 (11/2015)
8.1 Pflops (11 PF Peak)

CIELO
8944 nodes
285 TB of total
on-node mem.
(retired 7/2016)



Cray Aries 'Dragonfly'
Advanced Adaptive Routing
All-to-all backplane & between groups



**Cray Sonexion
Storage System**
78 PB Usable, ~1.6 TB/s

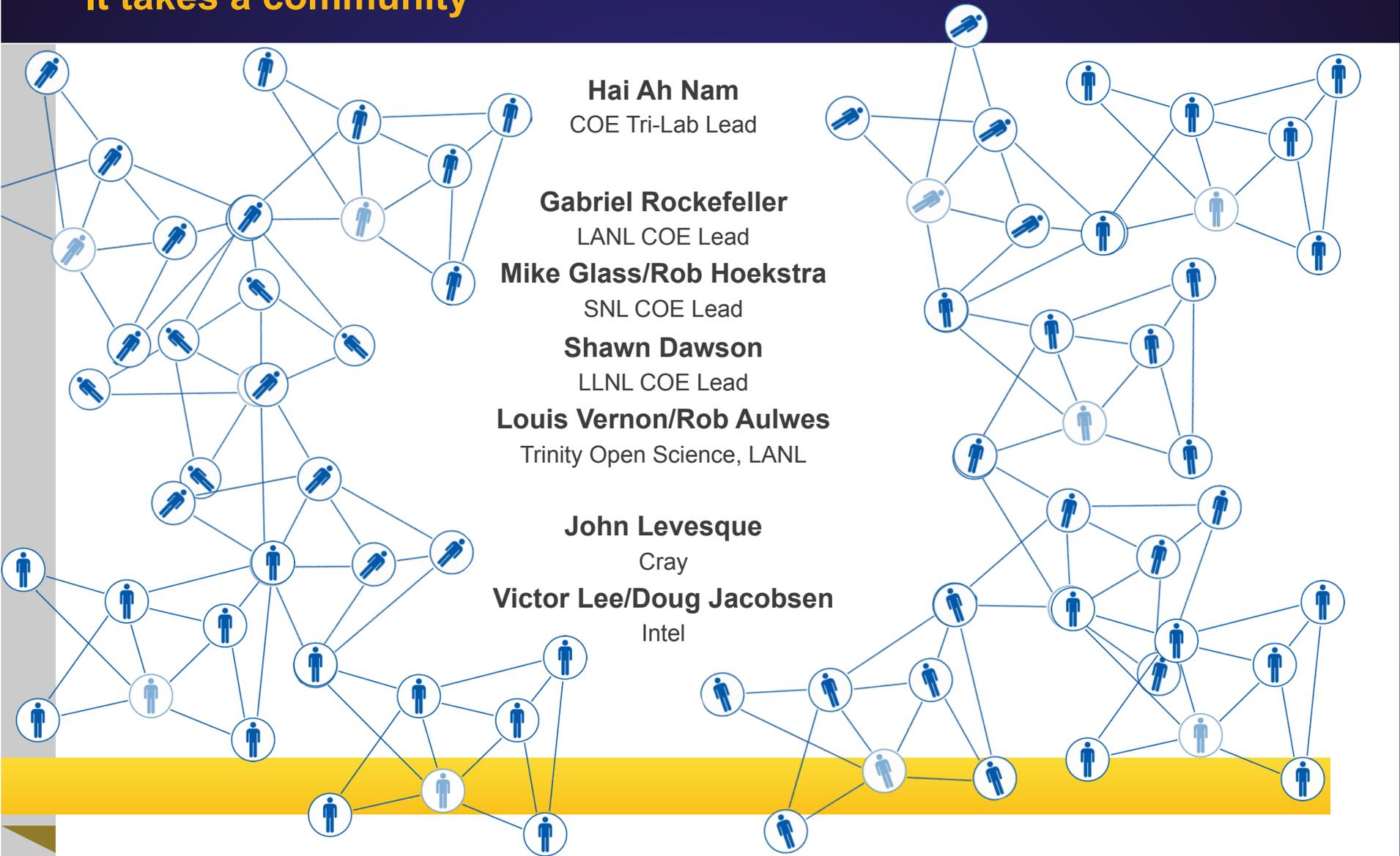


Cray DataWarp
576 Burst Buffer Nodes
3.7 PB, ~3.3 TB/s

*You can design & create, and build the most wonderful place in the world.
But it takes people to make the dream a reality. ~ Walt Disney*

Trinity Center of Excellence

It takes a community



Hai Ah Nam
COE Tri-Lab Lead

Gabriel Rockefeller
LANL COE Lead

Mike Glass/Rob Hoekstra
SNL COE Lead

Shawn Dawson
LLNL COE Lead

Louis Vernon/Rob Aulwes
Trinity Open Science, LANL

John Levesque
Cray

Victor Lee/Doug Jacobsen
Intel

Identify Challenges Facing Reality

Cielo
 16 cores [16]
 SSE 128-bit [2 doubles]



Trinity HSW
 32 cores + HT [64]
 AVX 256-bit [4 doubles]

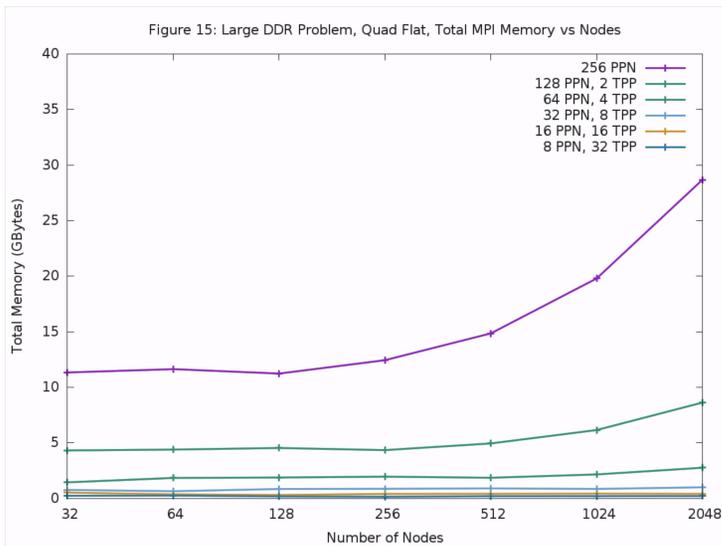


Trinity KNL
 68 cores + 4 HW [272]
 AVX-512 512-bit [8 doubles]

- **On-node parallelism**

- MPI Overhead

- 68 ranks/node @8792 nodes (~600K ranks) used 8.3 GB/node, however...



[Contributed by: Dave Nystrom, LANL]

- At some point you'll have to fill in the blank: MPI + _____

- **Vectorization**

- Data structure/access patterns for compiler auto-vectorization
 - Fortran & C vectorize better than C++

- **Memory hierarchy(DDR+HBM)**

- Cache vs. explicit management
 - 5x memory BW with HBM over DDR

- **KNL cluster/memory modes**

- Only 20?

- **I/O (2X BW from Burst Buffer)**

- Scheduling yet another resource

- **Exploit new opportunities**

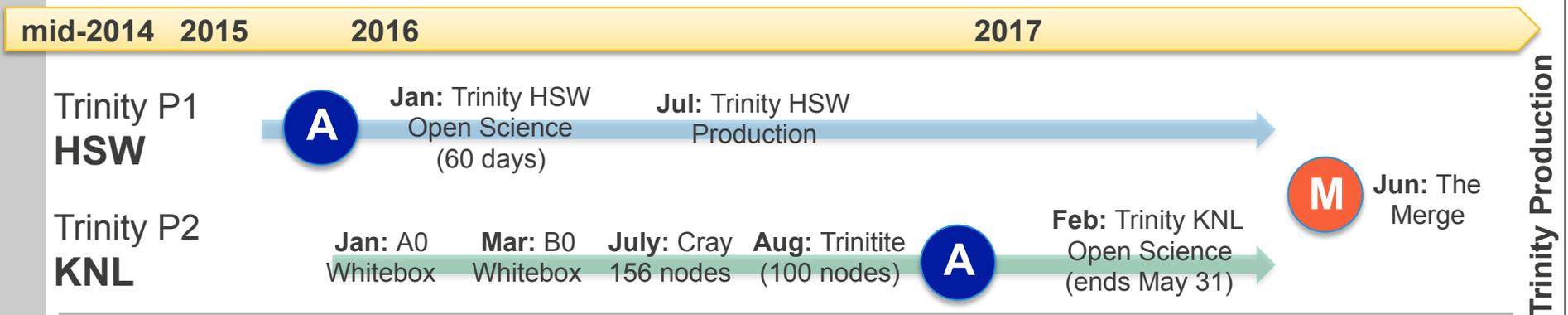
- Scale & heterogeneous system

You can avoid reality, but you cannot avoid the consequences of avoiding reality.

~ Ayn Rand

Hardware Timelines & COE Goals

Setting Expectations



- Pre- hardware delivery & software hardening
- No defined (contractual) metric for success to enable high risk prototyping and a desire to foster collaboration on the real issues
- COE GOALS: “Shared fate approach” where vendor, application developers and software developers work collaboratively
 - Port and achieve performance on key ASC applications to attain simulation scales that were not previously possible
 - Developer productivity & education
 - Provide vendors deeper insight into real production application needs (complex!)

Metrics are few because constraints are many

COE Activities

Creating Opportunity for Collaboration

Strike Force

(2-5 days of intense collaboration)

- Intel: Hackathon, Discovery, Dungeon
- Cray: Deep dive, Bootcamp
- Identify bottlenecks, explore improvements
- Presentations/training w/ each activity to update users on new features/tools

10 activities in 2016!

Sharing Best Practices

- KNL Working Group – + ANL/NERSC
- COE Seminars
- Tri-lab coordination meeting
- Lots of mailing lists, wikis, confluence
- Acceptance lessons learned
- DOE COE Performance Portability Workshop (April 2016)

Training

- Cray/Intel Trinity Phase 1 Workshop (2015)
- 4 Days (compiler, tools, hardware, file system, MPI)

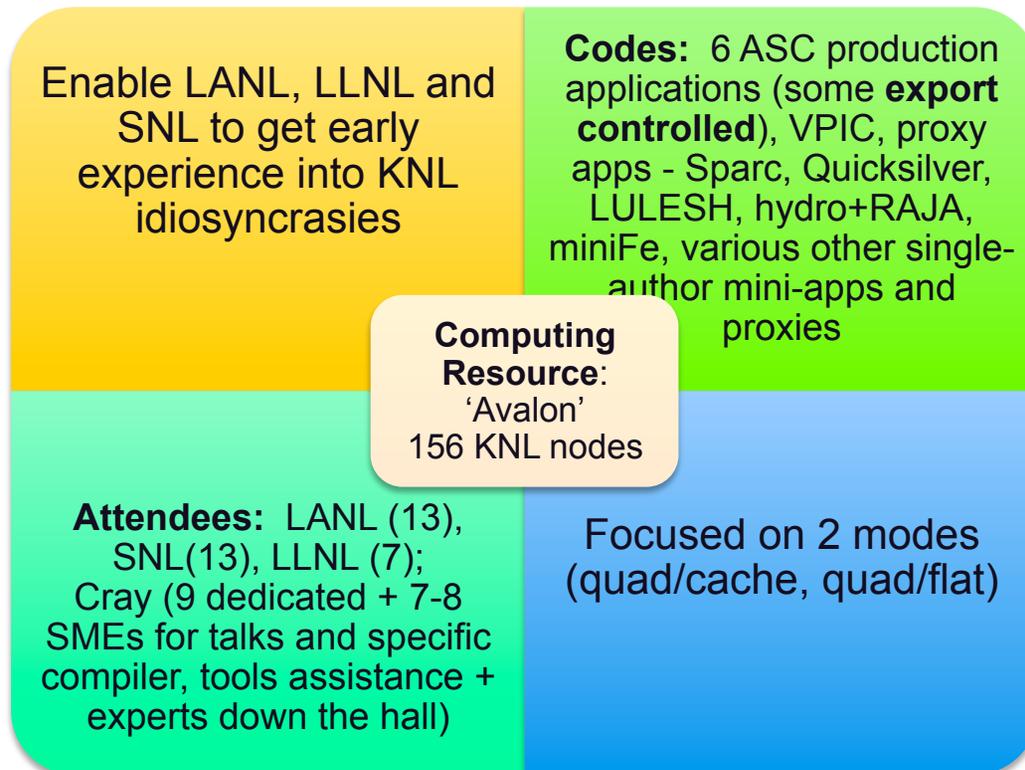
Embedded Support

- Cray (Wagenbreth) @ SNL
- Cray (Levesque) @ LANL
- Intel (Jacobsen) @ LANL+SNL+LLNL

You have to get along with people, but you also have to recognize that the strength of a team is different people with different perspectives and different personalities. ~ Steve Case

The Mother of all Bootcamps (July, 2016)

• Tri-Lab Cray Trinity Bootcamp at Cray in St. Paul, MN



Outcomes

- All codes able to build and run with at least one compiler on KNL and run on a single node (performance ranges from good to bad out of the box)
- ~80% of codes able to run on more than 1 KNL node to do scaling studies, some using up to 100+ nodes
- Out of the box performance improvements for some codes compared to Moonlight production system
- Running 1 MPI process per tile (32 MPI Processes + Threads) looks promising for a range of hybrid MPI+OpenMP applications

Exploration without boundaries... or at least not as many as usual

Exposing Application Complexity

Building long-term advocacy for our application development

SNL – Sierra Solid Mechanics

- 2 Intel Discovery Sessions
- Intel ‘Super’ Dungeon (2/2016)
 - Sierra NALU, Trilinos MG Solver, Sierra Solid Mechanics/Structural Dynamics Domain Decomposition Solver
- 6 weeks preparation meetings
- 12-15 Intel SMEs
- REAL code & dependencies, not proxies

Beware: reduced productivity

LLNL – Proxy Applications & Performance Portable Abstraction Layer (RAJA)

- 2 Discovery Sessions
- Intel Dungeon (5/2016)
 - Quicksilver, LULESH w/ RAJA, Kripke
 - Improved hybrid MPI+threads performance by 32%

Expose vendors to real code issues (e.g. long compile times, vTune analysis, compiler issues)

Improved compiler

Improved tools

Faster Bug Fixes

Expose OpenMP bugs in compiler, RAJA long compile times and correctness issues

Vendor Partner

Once you get that two-way energy thing going, everyone benefits hugely.

~ James Taylor

It all started with a COE Seminar

COE Seminar @ LANL & SNL

- Peter Mendygral (Cray) - *High-level OpenMP and Thread Scalable MPI-RMA: Application Study with the Wombat Astrophysical MHD Code*
- Illustrated the benefit of individual threads performing their own MPI using MPI-3 one sided RMA message passing in a SPMD model

Inspired Jim Schwarzmeier (Cray) Implement SPMD OpenMP into SNAP (Sn transport proxy) with good performance improvement (10-30%) and evidence it could work at production scale.

- 3 versions
- P2P MPI replaced with SHMEM-like MPI_PUT
- rma_buf size doubled – 2 outstanding receives
- OMP PARALLEL region lowered

Several other LANL proxy applications implementing SPMD OpenMP (one-sided MPI)

Jim Schwarzmeier & Peter Mendygral (Cray);
Randy Baker & Joe Zerr (LANL)

Power of proxy apps → 'high-risk' prototyping

Trinity Open Science

- **LANL**

- SPaSM/CoMD: Molecular Dynamics

- C++, Semi-Explicit Vectorization, OpenMP, MPI
- PI: Tim Germann; **IC-APT member(s): Louis Vernon, Xiaoying Pang**

- Genesis: Molecular Dynamics

- Fortran, Implicit Vectorization, OpenMP, MPI
- PI: Karissa Sanbonmatsu; **IC-APT member(s): Mike Wall, Toks Adedoyin**

- PetaVision: Neural networks

- C++, Implicit Vectorization, OpenMP, MPI
- PI: Garrett Kenyon; **IC-APT member(s): Boram Yoon, Ron Green**

- VPIC: Particle-in-cell plasma code

- C++, Explicit Vectorization, Pthreads (OpenMP), MPI
- PI: Brian Albright; **IC-APT member(s): Bill Rust, Xiaoying Pang**

- **SNL (QMCPack, LAMMPS, CTH)**

- **LLNL (Mercury)**

Limited
Access

Funded

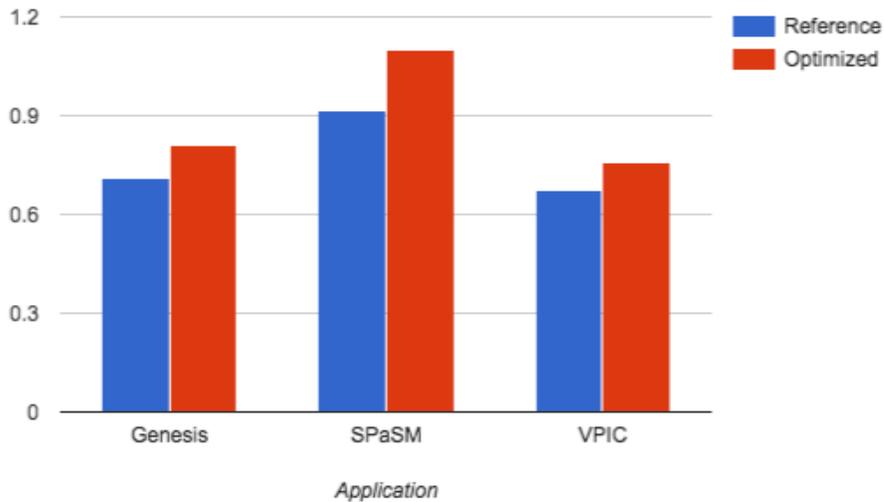


Human
Supported

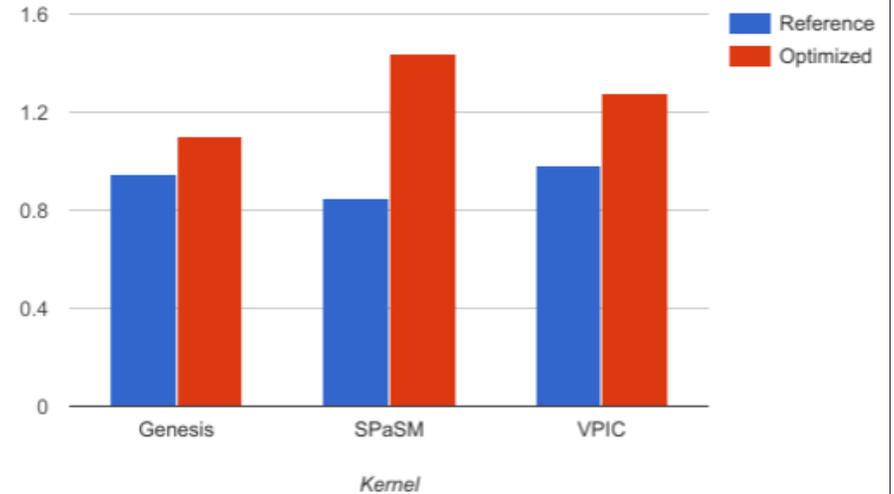
2 Gordon Bell Submissions

Open Science Codes – Early Performance Results

Application KNL Performance Relative to Haswell - Before and After Optimization



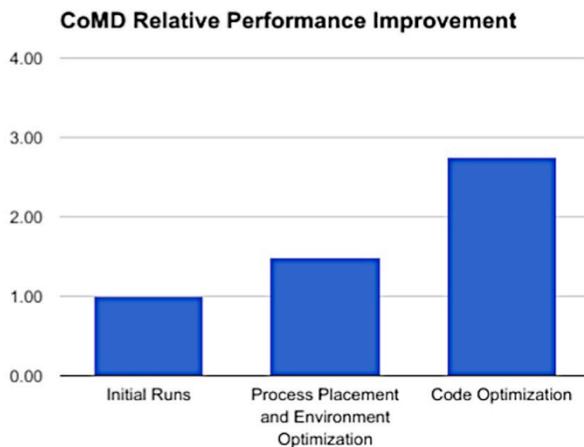
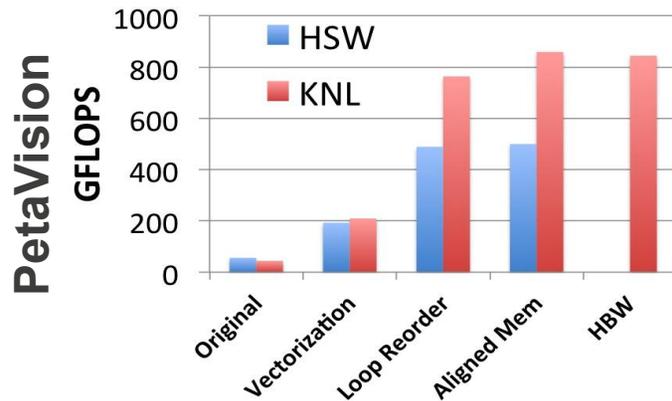
Kernel KNL Performance Relative to Haswell - Before and After Optimization



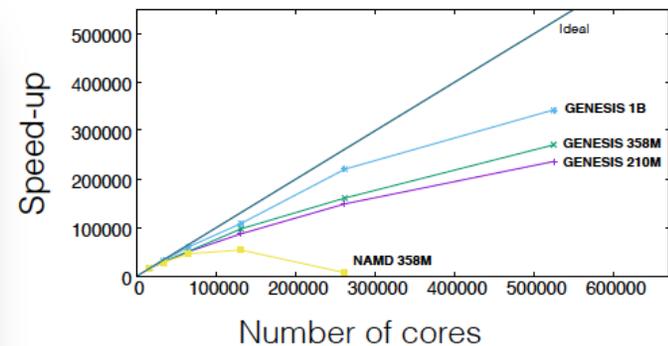
Improvements for the KNL will also reap benefits on the Haswell (performance portability)

Provided by Louis Vernon, LANL

Open Science Codes Early Performance/Scaling Results



(a) Strong Scaling - Trinity KNL Phase 2



(b) Strong Scaling - Trinity KNL Phase 2

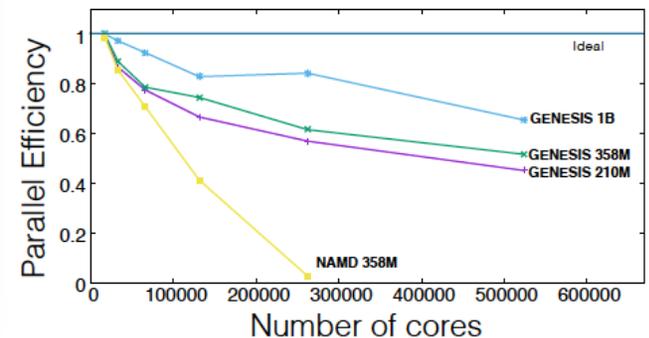


Figure 3. Strong scaling of GENESIS vs. NAMD on LANL Trinity Phase 2 KNL platform. (a) Speed-up. (b) Efficiency.

Provided by Louis Vernon, LANL

Satellite Tobacco Mosaic Virus Benchmark
Karissa Sanbonmatsu et al.

Lessons learned, parting words, musings

- **Applications (on-going)**

- Lots of good progress, foundation laying, relationship building
 - Bottlenecks understood, time to roll up your sleeves

- **Trinity System (on-going)**

- Quad cache is a good starting point, then explore other modes
- Dynamic provisioning (user-driven mode change) is not allowed (reboot times and stability are still a work in progress)
- Cross-compiling is always fun and needs more attention

- **COE (on-going)**

- Everyone is integrated into the communication network to find the help they need to make progress on their problem.
- Impact to future systems, system software design lifecycle, procurement choices
 - Really? You're latency bound?
 - What the devil are you doing with these templates?

Keep on movin' on

This work was performed using the Cray Trinity system of the Alliance for Computing at Extreme Scale (ACES), a partnership between Los Alamos National Laboratory and Sandia National Laboratories for the U.S. Dept. of Energy's NNSA