OAK RIDGE National Laboratory

ARM + NVIDIA HPC Software Ecosystem Evaluation on Wombat at NCCS

Ross Miller[1], Jack Wells[1], Oscar Hernandez[1], Wayne Joubert[1], Ada Sedova[1], Markus Eisenbach[1], Veronica Vergara,[1] Damien Lebrun-Grandie[1], Arghya Chatterjee[1], Matthew Baker[1], Piotr Luszczek [2], Stan Tomov [2], Patrick Atkinson[3], Tom Deakin[3], Simon McIntosh-Smith[3], Julio Maia[4], John Stone[4], David Hardy[4], Andrea Bocci[5], Vincenzo Innocente[5], Matti Kortelainen[6], Marco Rovere[5], Takuma YAMAGUCHI[7], Kohei FUJITA[7], Tsuyoshi ICHIMURA[7], Christian Trott[8], Akiyoshi Kuroda[13] Duncan Poole[9], Pramod Ramarao[9], Giridhar Chukkapalli[9], CJ Newburn[9], Yukihiko Hirano[9], Doug Miles[9], Matthew Colgrove[9], Raaghav Hebbar[10], Joel Jones[10], Parijat Shukla[10], Matthew Cordery[10], Nicholas Forrington[11], John Linford[11], Andy Warner[12]

Oak Ridge National Laboratory[1], University of Tennessee Knoxville (ICL)[2], University of Bristol[3], University of Illinois Urbana-Champaign[4], CERN[5], Fermi National Laboratory[6], University of Tokyo[7], Sandia National Laboratories[8], RIKEN[13],NVIDIA[9], MARVELL[10], ARM Ltd[11], HPE[12]

November 18-22, 2019

Supercomputing 2019

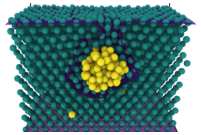
Denver CO, USA

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725

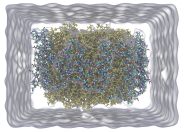
NVIDIA+ARM HPC software stack evaluation using Wombat at NCCS

Applications:

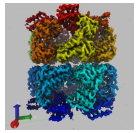
CoMet
Comparative Genomics



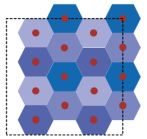
LAMMPS
Molecular Dynamics



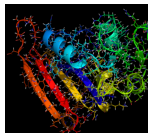
NAMD
Molecular Dynamics



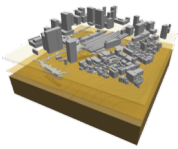
VMD
MD Vizualiz.



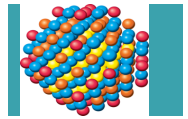
DCA++
Material Science



Gromacs
Molecular Dynamics



Gamera
Earthquake Simulator



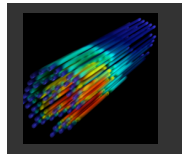
LSMS
Material Science

Benchmarks & Mini-apps:

BabelStream
Memory Transfer

Tea Leaf
Heat Cond.

Clover Leaf
Lagrangian Eulerian hydrodynamic



MiniSweep
Radiation Transport

SNAP
Radiation Transport

Patatrack Pixel Reconstruction

Parallel Prog Models & Sci. Libraries:

Kokkos
C++ Prog. Model

Open MPI
Distributed Prog. Model

CUDA / CUDA Fortran
GPU Prog. Model

UCX
Comm. Framework

Magma SLATE
Sci. Libraries

Evaluation done by:

NVIDIA+ARM: Wombat "testbed" at NCCS

Administrator:
Ross Miller (ORNL)

Hardware:

- HPE Apollo 70 Preproduction nodes
- CPU: ThunderX2
  - 2 Sockets, 28 Cores/socket, 4 threads/per core, 2.0GHz
  - 256 GB RAM
- GPU's: NVIDIA Volta GV100 (2 per node)
  - 32 GB HBM2 each
- 4 nodes with NVIDIA GPUs
- EDR InfiniBand

Software:

- RHEL 8
- CUDA 10.2.107 (aka "Drop 2")
  - Installed Nov 7. Most user's results are with 10.2.91 ("Drop 1")
- GCC 8.2.1is the default compiler
  - GCC 9.2.0 and armclang 19.3 available
- PGI Dev. Compiler
- Open MPI 3.1.4 and 4.0.2rc3
- UCX 1.7.0

Website:

https://www.olcf.ornl.gov/olcf-resources/compute-systems/wombat/

Comparison of Summit vs Wombat Nodes

System Specs	Summit	Wombat
CPU	IBM POWER9™ (2 Sockets / 21 Cores per socket)	Marvell (formerly Cavium) ThunderX2 (2 Sockets / 28 Cores per socket)
GPU	NVIDIA Volta (6 per node)	NVIDIA Volta (2 per node)
RAM	512 GB DDR + 96 GB HBM (16 GB per GPU)	256 GB DDR + 64 GB HBM (32 GB per GPU)
On-node interconnect	NVIDIA NVLINK2 (50 GB/s) Coherent memory across the node	PCIe Gen3 (16 GB/s) No coherence across the node
System Interconnect	Mellanox Dual-port EDR IB network 25 GB/s	Mellanox Single-port EDR IB network 12.5 GB/s
On-node NVM (storage)	1600GB NVME	480GB SATA

Summary of the evaluation:

- Most applications compiled and ran "out of the box" with CUDA, GCC, Open MPI, Redhat 8, etc
  - Except Gromacs which was ported to ARM ThunderX2 by the developers (see Gromacs slide).
  - Other applications needed minor modifications in cmake, Arm scientific libraries APIs, etc.
- Applications produced correct results when using the NV100 "Volta" GPUs + ARM ThunderX2 CPUs
- Preliminary performance results were comparable with Summit using the same number of GPUs, except for explainable differences in memory interconnect and communication bandwidth.
  - E.g. NVLINK2 vs PCIe V3 x16 bandwidth
  - Results need to be rerun on a production system with better support for GPUdirect, etc.
- CUDA toolkit on NVIDIA+ARM was on par as NVIDIA+Power9 on Summit
- Evaluation results were peer-reviewed by application developers and vendors (e.g. NVIDIA, Marvell, etc)

Application Evaluation Acknowledgement:

- **CoMet**: Wayne Joubert (ORNL)
- **GROMACS 2019**: Ada Sedova (ORNL)
- **LAMMPS, KOKKOS**: Damien Lebrun-Grandie (ORNL), Christian Trott (SNL)
- **SLATE, Magma**: Piotr Luszczek (UTK/ICL), Stan Tomov (UTK/ICL)
- **DCA++**: Arghya Chatterjee (ORNL)
- **TeaLeaf, CloverLeaf, SNAP, BabelStream**: Patrick Atkinson (Univ. of Bristol), Tom Deakin (Univ. of Bristol), and Simon McIntosh-Smith (Univ. of Bristol)
- **LSMS**: Markus Eisenbach (ORNL)
- **NAMD & VMD**: Julio Maia (UIUC), John Stone (UIUC), David Hardy (UIUC)
- **Patatrack Pixel Reconstruction**: Andrea Bocci (CERN), Vincenzo Innocente (CERN), Matti Kortelainen (FNAL), Felice Pantaleo (CERN), Marco Rovere (CERN)
- **MiniSweep**: Veronica Vergara (ORNL)
- **GAMERA**: Takuma Yamaguchi (Univ. Tokyo), Kohei FUJITA (Univ. Tokyo), and Tsuyoshi ICHIMURA (Univ. Tokyo)
- **GENESIS, Chainer**: Akiyoshi Kuroda (Riken)

Acknowledgement:(2)

- **Science and Application Leader**: Jack Wells (ORNL)
- **Testbed Administrator**: Ross Miller (ORNL)
- **SW Evaluation Coordinator**: Oscar Hernandez (ORNL), Matt Baker (ORNL)
- **NVIDIA**: Duncan Poole, Pramod Ramarao, Giridhar Chukkapalli, CJ Newburn, Yukihiko Hirano, Doug Miles, Matt Colgrove
- **MARVELL**: Raaghav Hebbar, Joel Jones, Parijat Shukla, Matthew Cordery
- **ARM**: Nick Forrington, John Linford
- **HPE**: Andy Warner

CoMet

Wayne Joubert, ORNL

Application information

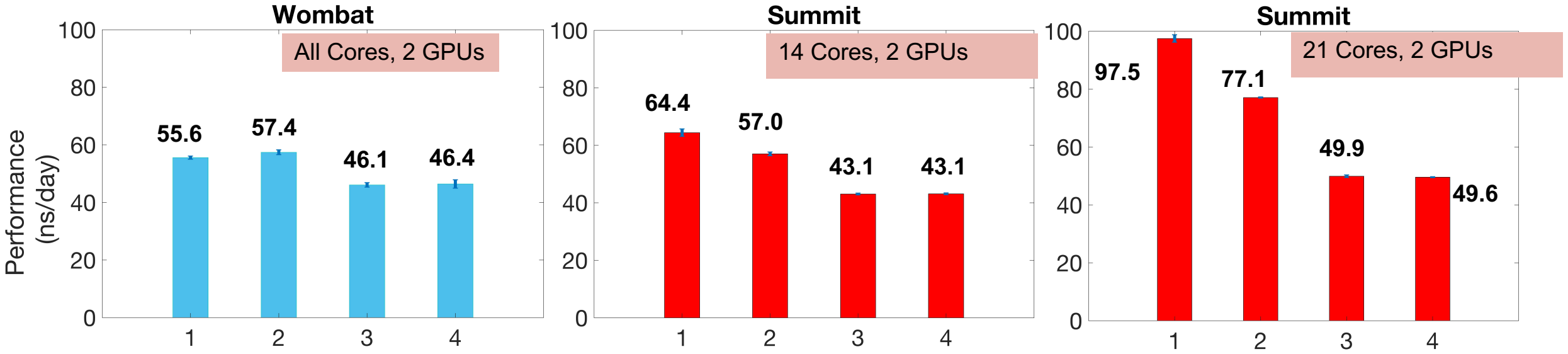
- CoMet: Combinatorial Metrics Application
  - Wayne Joubert lead developer, Dan Jacobson science lead
- Performs genomic epistasis and pleiotropy calculations using vector similarity methods
- Programming Model:
  - MPI + CUDA
  - cuBLAS for GEMM operations using tensor cores; >2 EF performance on Summit
  - Modified MAGMA library is used for other GEMM-like computations
  - OpenMP 3.1 threading for CPU work
  - Asynchronous overlapped computation, communication, transfers

ARM + NVIDIA experience

- Easy initial port of code, runs successfully using one GPU
- CoMet is designed to run efficiently on many nodes -- on small systems, per-GPU performance is limited
- Results for representative run, 2-way CCC method using tensor cores:
  - Summit 1.577 sec, Wombat 1.922 sec
  - Summit 37 TF, Wombat 30 TF
  - Performance limited by CPU for small case
- Next steps:
  - Multi-GPU runs with MPI
  - Runs on many nodes; code tuning
  - OpenMP 3.1 efficiency and CPU speed will be critical to achieving high performance

Single-Node Performance of GROMACS 2019 on ORNL Wombat vs. Summit

Ada Sedova, ORNL



Benchmark: 134K atom ADH (cubic)
1: Using GPU for PME (FFTs), nonbonded, and bonded terms
2: Using GPU for PME and non-bonded only
3: Using GPU for bonded and nonbonded only
4: Using GPU for nonbonded only

*Error bars: standard deviation over 5 runs

GROMACS 2019 needed to be ported to NEON simd to get a significant speedup in the ThunderX2 ARM CPU

Run Configuration-- fastest for each resource using 2 GPUs:
- **Wombat:** 20 MPI ranks, 8 OpenMP threads per rank
- **Summit 14 Cores (--cpu_per_rs):** 2 MPI ranks, 7 OpenMP threads per rank
- **Summit 21 Cores (--cpu_per_rs):** 6 MPI ranks, 7 OpenMP threads per rank

CPU only performance
Wombat: 10.3 ns/day
Summit: 14.4 ns/day

Summit whole node performance (6 GPUs, 42 cores)
1. 127.2 ns/day
2. 107.3 ns/day

Preliminary Results

Bristol Benchmarks

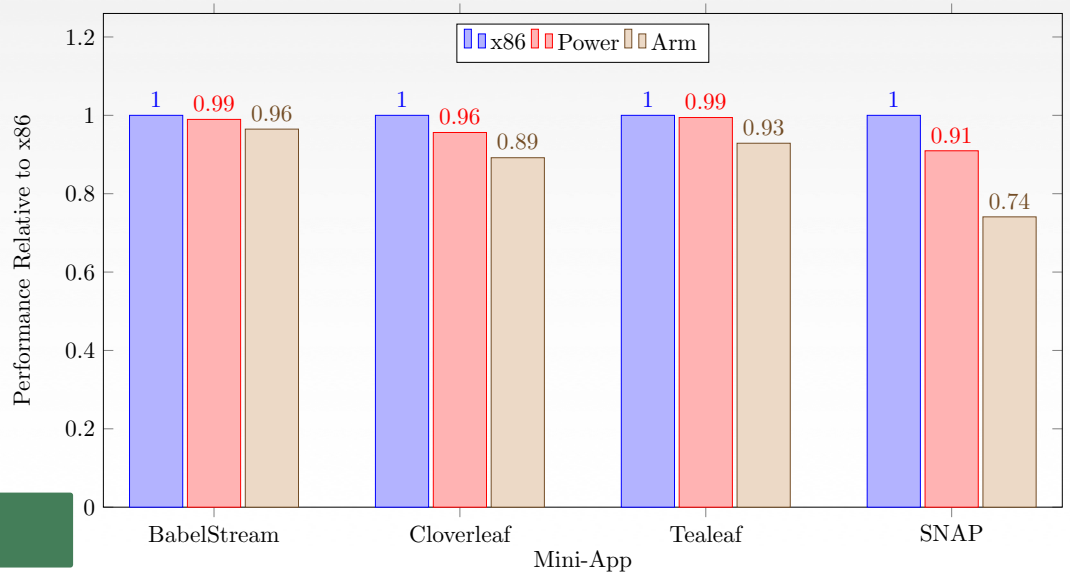
Patrick Atkinson,
Tom Deakin and
Simon McIntosh-Smith

Application information

- BabelStream
  - Memory bandwidth benchmark from UoBristol.
  - Test: STREAM Triad kernel
- CloverLeaf
  - A Lagrangian-Eulerian hydrodynamics benchmark from Mantevo/UK-MAC.
  - Test: clover_bm_16.in
- TeaLeaf
  - A heat conduction benchmark from Mantevo/UK-MAC.
  - Test: tea_bm_5.in
- SNAP
  - Neutral particle transport proxy application from Los Alamos National Lab.
  - Test: 8x8x1024 cells, ng=32, nang=36, nmom=2, ichunk=16, nsteps=1, oitm=iitm=5
- Programming model:
  - Native CUDA implementation.

ARM + NVIDIA experience

- Graph shows comparison of three V100 GPU systems with different hosts: Intel Broadwell, IBM Power 9 and Marvell ThunderX2 (Arm).
  - Performance normalized to x86 host.
- All applications previously optimized for GPUs, and ran on all systems with no modifications.
- Performance of applications very similar across all hosts, as expected.
- NVPROF unavailable on Wombat so can't profile SNAP's host-device interactions further.



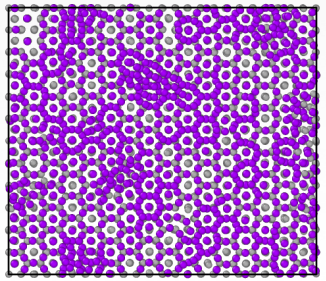
Preliminary Results

LAMMPS

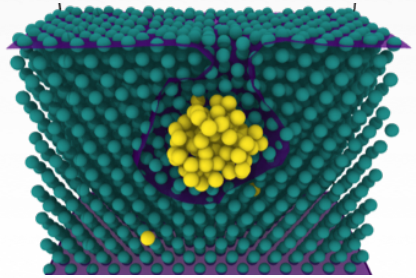
Damien Lebrun-Grandie (ORNL)
Christian Trott (SNL)

Application information

- Authors
  - Lead: Steve Plimpton (Sandia National Labs)
  - Kokkos-PKG-Mgt: Stan Moore
  - SNAP: Aidan Thompson
- Domain & Method
  - Classical Molecular Dynamics Code
  - Domain Decomposition + shared memory parallelism
- Programming Model
  - MPI+Kokkos (https://github.com/kokkos)
  - Kokkos: provides performance portability to diverse node architectures



Beryllium depositing on tungsten surface (SNAP WBe)

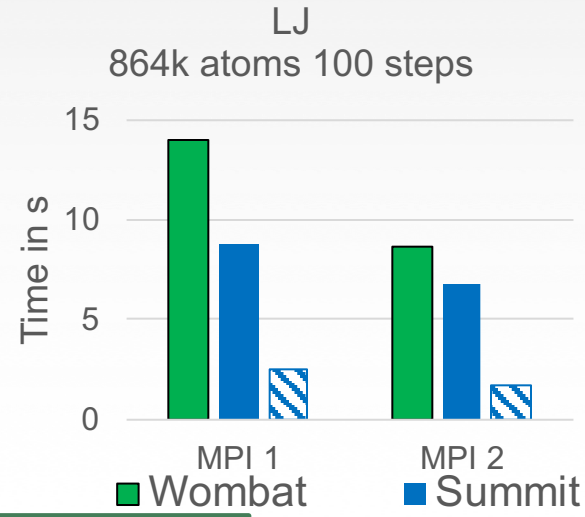


Helium bubble forming near tungsten surface

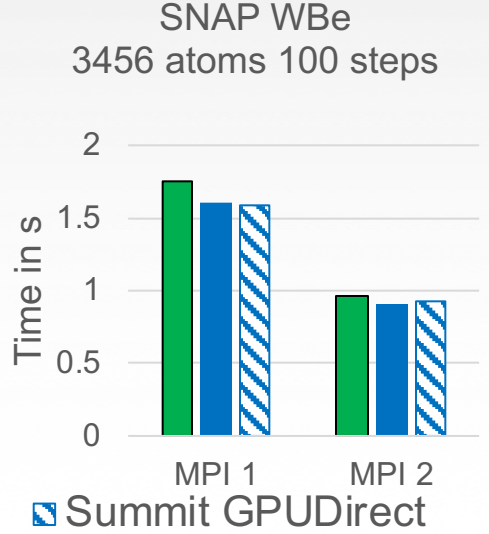
ARM + NVIDIA experience

- LAMMPS simply worked since Kokkos did
  - Use any release of LAMMPS since March 2019
  - Wombat: KOKKOS_ARCH="ARMv8-TX2;Volta70"
  - Summit: KOKKOS_ARCH="Power9;Volta70"
- Summit vs Wombat

Latency/Comm Limited =>
No GPUDirect on Wombat +
No NVLink



Compute Limited =>
GPU Frequency difference



Preliminary Results

Verónica G. Melesse Vergara
Oak Ridge National Laboratory

OAK RIDGE National Laboratory

Minisweep

Application information

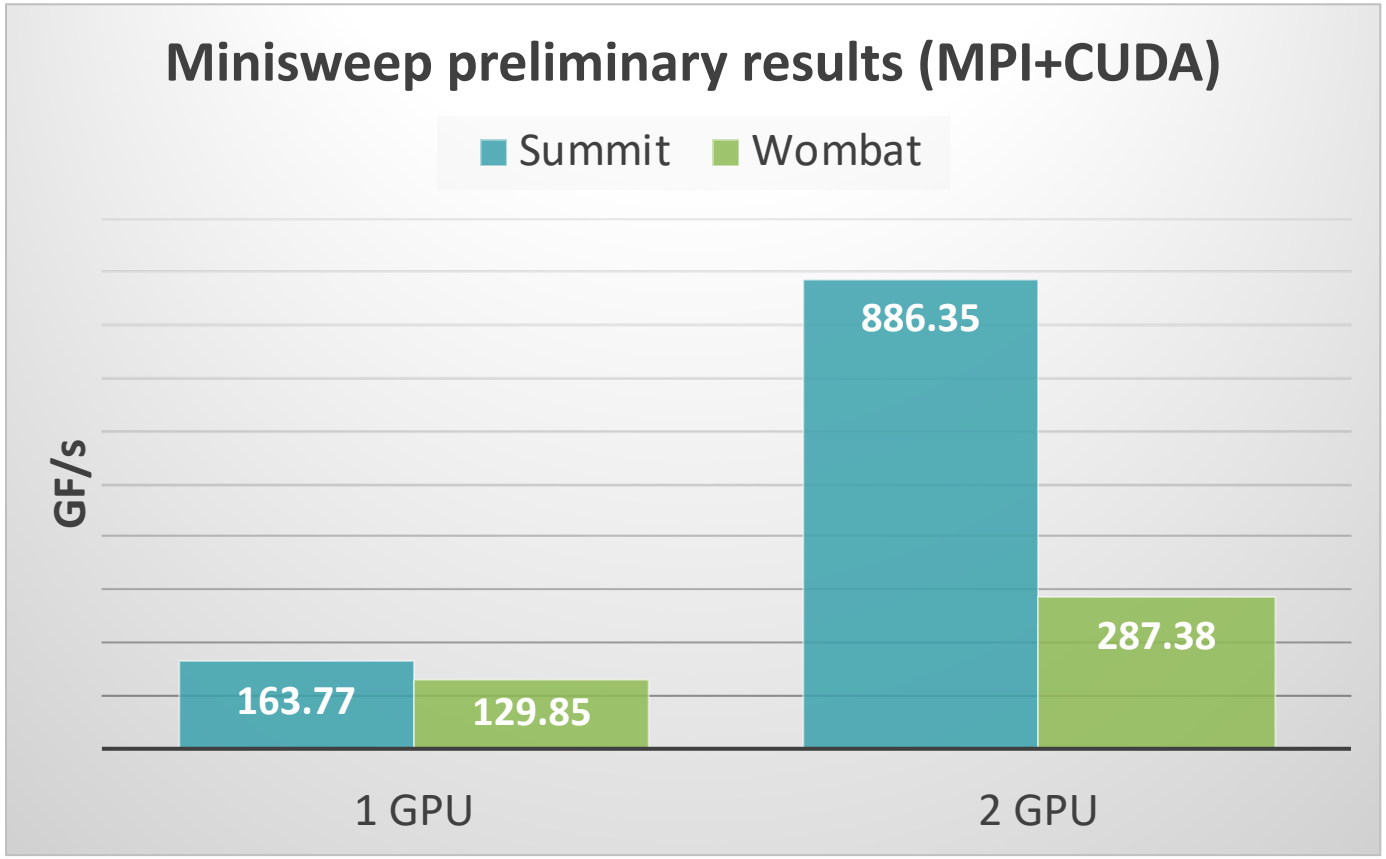
- https://github.com/wdj/minisweep
  Minisweep is a deterministic Sn radiation transport mini-app used for performance optimization and performance model evaluation on HPC architectures.
  - Author: Wayne Joubert (ORNL)
  - Contributors: Robbie Searles (NVIDIA), Verónica G. Melesse Vergara (ORNL)
- Minisweep is a proxy application for the Denovo radiation transport code
- Programming Models: supports OpenMP 3.1, OpenMP 4.5, OpenACC, and CUDA
- Use cases: nuclear reactor core analysis (neutronics), nuclear forensics, radiation shielding and radiation detection.

ARM + NVIDIA experience

- What was the experience for porting the application to ARM + NVIDIA
  - MPI + CUDA version built with GCC 8.2.0 was straightforward
- Software stack readiness:
  - Encountered issues when trying to run using OpenMPI 4.0.2 RC3.
  - Need `module load ARM_Compiler_For_HPC/19.3
  - Need `--host <host>:56`
- Correctness: Minisweep's built-in check was successful
- Next steps:
  - Understand performance difference
  - Attempt to build the OpenMP 4.5 version
  - Having access to NVPROF would be key

Minisweep Preliminary Results

- Results obtained on Summit and Wombat.
- Built with GCC 6.4.0 on Summit and GCC 8.2.0 on Wombat
- CUDA 10.1 on both systems but different CUDA drivers
  - Summit: 418.67
  - Wombat: 435.17.01
- These are preliminary results obtained on a single node using 1 and 2 GPUs.
- Different problem sizes were used for each case.

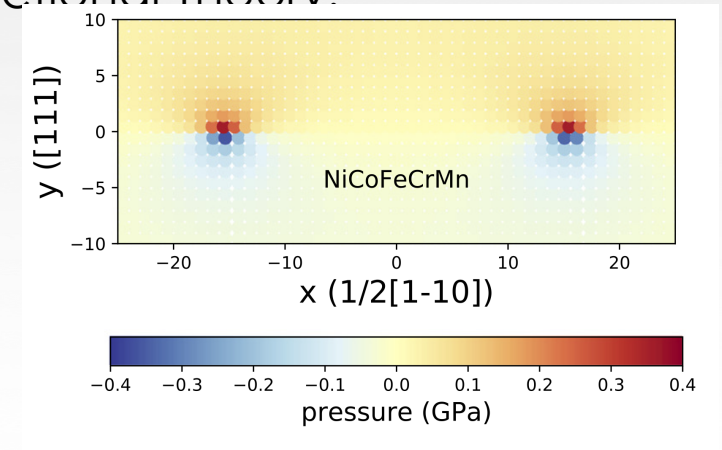


LSMS

Markus Eisenbach, ORNL

Application information

- Scalable First Principles calculations for materials
  - Main Developer and Scientific Lead: Markus Eisenbach
- LSMS (Locally Self-consistent Multiple Scattering) solves the Schrödinger or Dirac equation for electrons in solids within Density Functional Theory.



Dislocation core structure in NiCoFeCrMn alloy

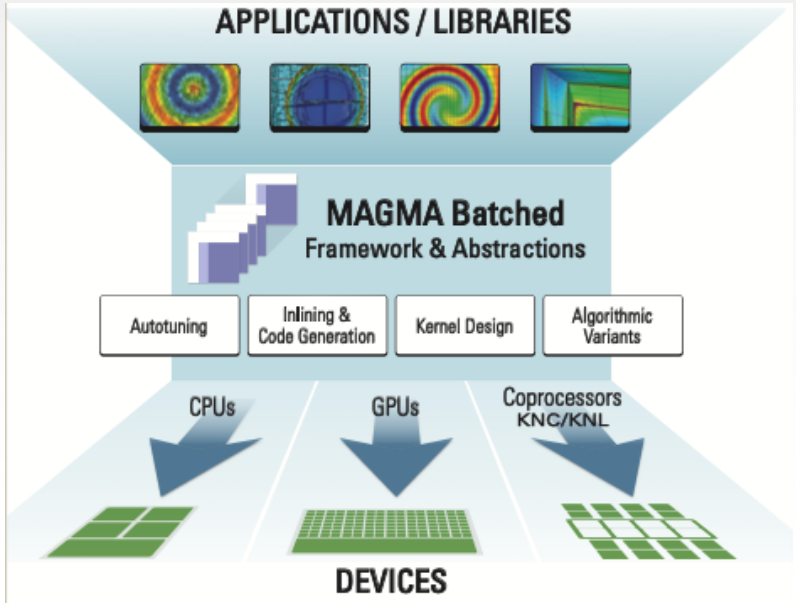
https://github.com/mstsuite/lsms

ARM + NVIDIA experience

- LSMS compiled without problems with the standard build procedure for Linux systems + NVIDIA GPUs
- Experience and performance similar to standard Linux x86 + GPU systems
- Programming model:
  - MPI + OpenMP for Node / CPU parallelism
  - CUDA + cuBLAS for GPU opperartions
- Performance dominated by dense double precision complex matrix operations:
  - ZGEMM; ZGETRF; ZGETRS
- Future work:
  - Multi Node MPI runs
  - Porting additional kernels to GPU to achieve improved performance

MAGMA

Application information

- MAGMA
  - Jack Dongarra, et. al
- Numerical Linear Algebra, Dense & Sparse Matrices, Linear Systems, Eigenvalue Problems
- CUDA, OpenMP CPU threading
- 

ARM + NVIDIA experience

- What was the experience for porting the application to ARM + NVIDIA
  - ARM Compilers worked out-of-the-box
  - OpenMP support worked
  - ARM Performance Library worked
- Summit vs Wombat
  - ARM Performance Library is complete BLAS and LAPACK for ARM
  - IBM's ESSL is incomplete
  - Correctness slow to check on the CPU cores
- Anything that you can say about performance (if it makes sense)
  - 2 NVIDIA Voltas overwhelms ARM cores
- Next steps or forward looking items
  - Wishlist for ARM + NVIDIA: hardware integration, NVLink

SLATE

Piotr Luszczek/ Univ. of Tennessee
run on Wombat (ARM/NVIDIA)

Application information

- SLATE
  - Jack Dongarra et. al
- Distributed Memory Accelerated Numerical Linear, Linear Systems, Eigenvalue Problems, SVD, …
- Programming Models Tested
  - OpenMP 4.5 *tasking*
  - *CUDA*
  - *MPI*

ARM + NVIDIA experience

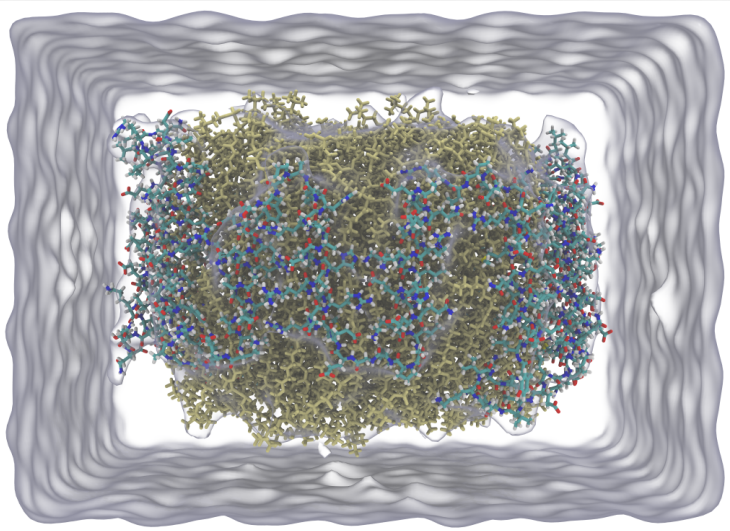
- What was the experience for porting the application to ARM + NVIDIA
  - ARM compilers and CUDA worked
  - ARM headers: not so much
    - API captured by ARM: cblas_zgemm(arm_complex *matrix)
    - Breaks Autoconf/Cmake
    - Requires manual intervention
- Summit vs Wombat
  - ARM Perf. Lib. Is complete BLAS/LAPACK
  - Correctness verified so far
- Anything that you can say about performance (if it makes sense)
  - A lot of options to map MPI-cores-GPUs
- Next steps or forward looking items
  - Wishlist for ARM+NVIDIA: H/W integration

NAMD

Julio Maia, John Stone, David Hardy
University of Illinois at Urbana-Champaign

Application information

- NAMD Molecular Dynamics
  - Julio Maia, John Stone, David Hardy, University of Illinois at Urbana-Champaign
- Domain & Method
  - Biophysics, Structural Biology
- Programming Models
  - Charm++, CUDA

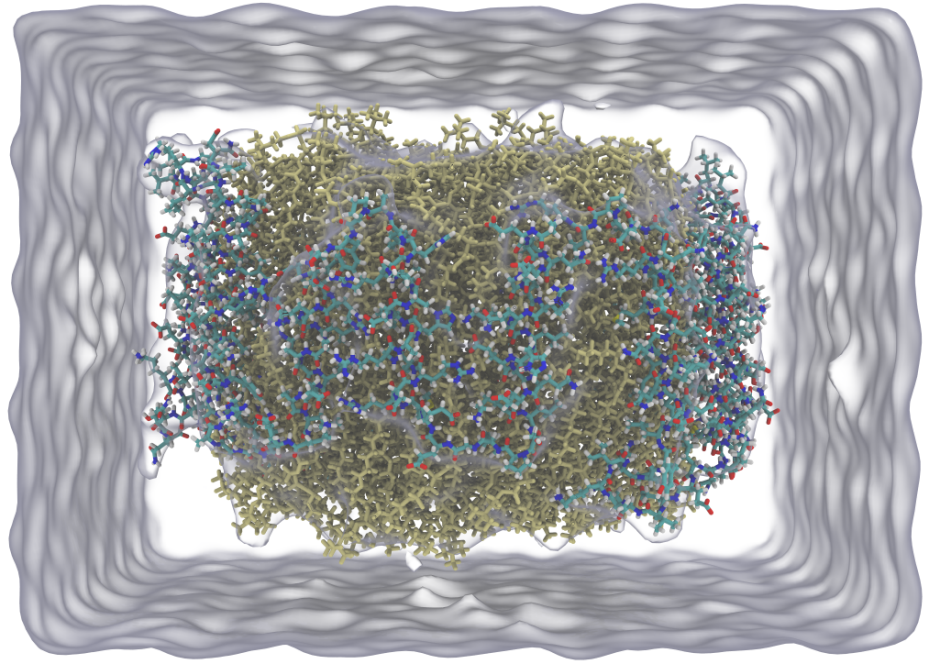


ARM + NVIDIA experience

- What was the experience for porting the application to ARM + NVIDIA
  - Took ~1.5 hours to compile NAMD and all of its dependencies entirely from scratch, including modifications to standard build scripts.
  - No big surprises, just a few tweaks for CUDA 10.2
- Summit vs Wombat
  - Software stack readiness: Good so far.
  - Correctness: Good so far
- Anything that you can say about performance (if it makes sense)
  - **Perf very comparable to x86/P9 systems**
- Next steps or forward looking items
  - **Benchmark IB w/ UCX, etc.**

NAMD 3 Alpha, Tesla V100 Performance, ApoA1 92k Atoms

Hardware platform		ns/day,	Speedup
Intel Xeon 8168	+ 1x Tesla V100	102.1,	1.0x
IBM Power9	+ 1x Tesla V100	99.7,	0.97x
Cavium ThunderX2	+ 1x Tesla V100	98.2,	0.96x



NAMD 3 Alpha Note:

This measured the new NAMD 3 Alpha in single-node mode, yielding greatest overall GPU acceleration for small to moderate sized atomic structure simulations.

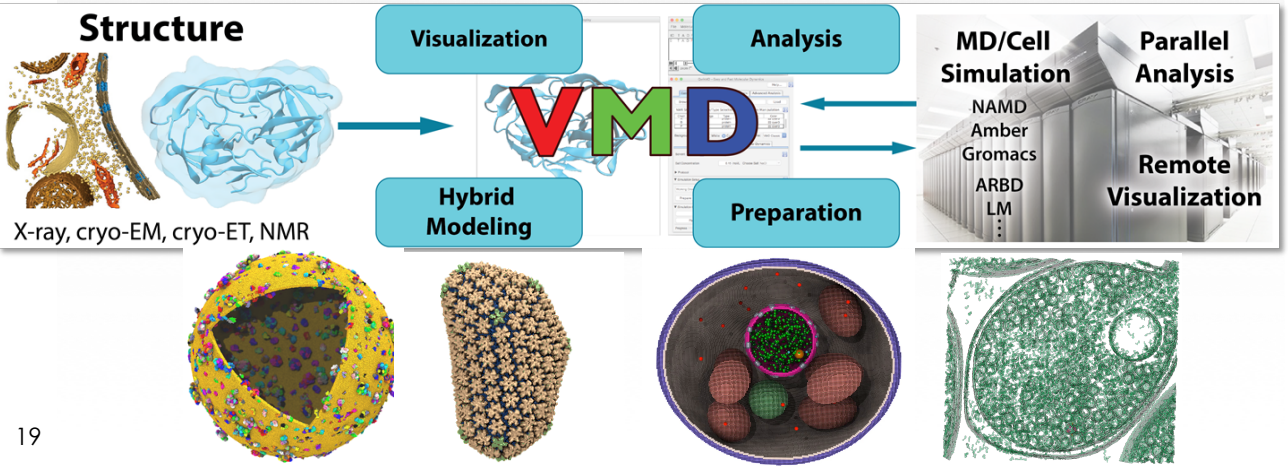
ARM+NVIDIA
Preliminary Results

VMD –Visual Molecular Dynamics

John Stone,
University of Illinois at Urbana Champaign

Application information

- VMD Molecular Dynamics Visualization, Preparation, and Analysis
  - John Stone,
    University of Illinois at Urbana-Champaign
- Domain & Method
  - Biophysics, Computational Biology, Materials Science
- Programming Models
  - MPI, POSIX Threads, CUDA



ARM + NVIDIA experience

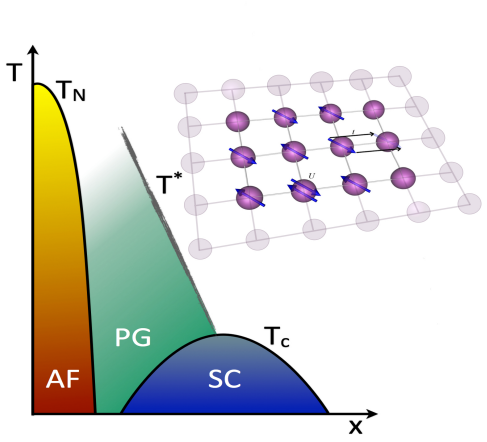
- What was the experience for porting the application to ARM + NVIDIA
  - ARM versions of OptiX, NvEnc/NvPipe not yet generally available
  - Didn't test multi-node MPI builds yet
- Summit vs Wombat
  - Software stack readiness
  - Correctness
- Anything that you can say about performance (if it makes sense)
  - (e.g. compared to Summit)
- Next steps or forward looking items
  - **Better CPU vectorization**
  - Test w/ missing NVIDIA graphics libs

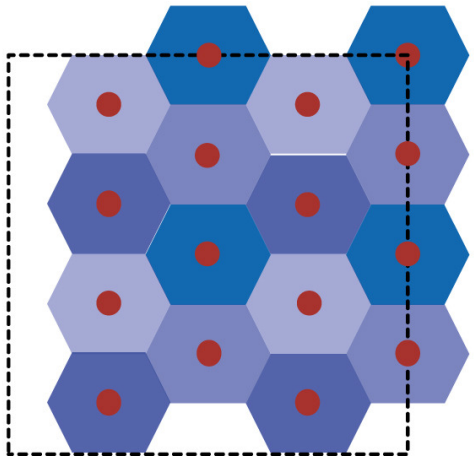
DCA++ (Dynamical Cluster Approximation)

Arghya Chatterjee, SciComp., NCCS.
OLCF INCITE Liaison for DCA++

Application information

- DCA++ :
  - PI: Dr. Thomas Maier (ORNL) / Thomas Schulthess (CSCS, ETH Zúrich).
  - U. Haehner, G. Balduzzi, A. Chatterjee, Y. W. Li, E. D'Azevedo… and others from ORNL and ETH.
- Scientific Motif : Condensed matter physics, predict behaviors of co-related quantum matters (eg. Superconductivity, magnetism)
- Programming Models + libraries : CUDA, pthreads, HPX, OpenMPI, BLAS, Lapack, Parallel HDF5.





ARM + NVIDIA experience

- Experience on porting to ARM + NVIDIA:
  - CMake kept adding "–m64" flags – which were then unrecognized by gcc 8.2 on ARM.
  - ARM + Marvell team assisted promptly to solved these issues.
- Summit vs Wombat:
  - DCA++ has laundry list of dependencies, various packages had to be compiled separately (this is expected for testbed systems)
  - All tests, MPI (2 nodes) and single node execution passes our extensive correctness test suite.
- Performance Analysis:
  - On node small test cases (only tested) performance was comparable to Summit wrt FLOP/s.
- Next steps or forward looking items
  - Bandwidth analysis w/ MPI runs for DCA++
  - Profile and analyze the cpu + gpu code w/ NVProf

VMD Tesla V100 Density Map Segmentation Performance,

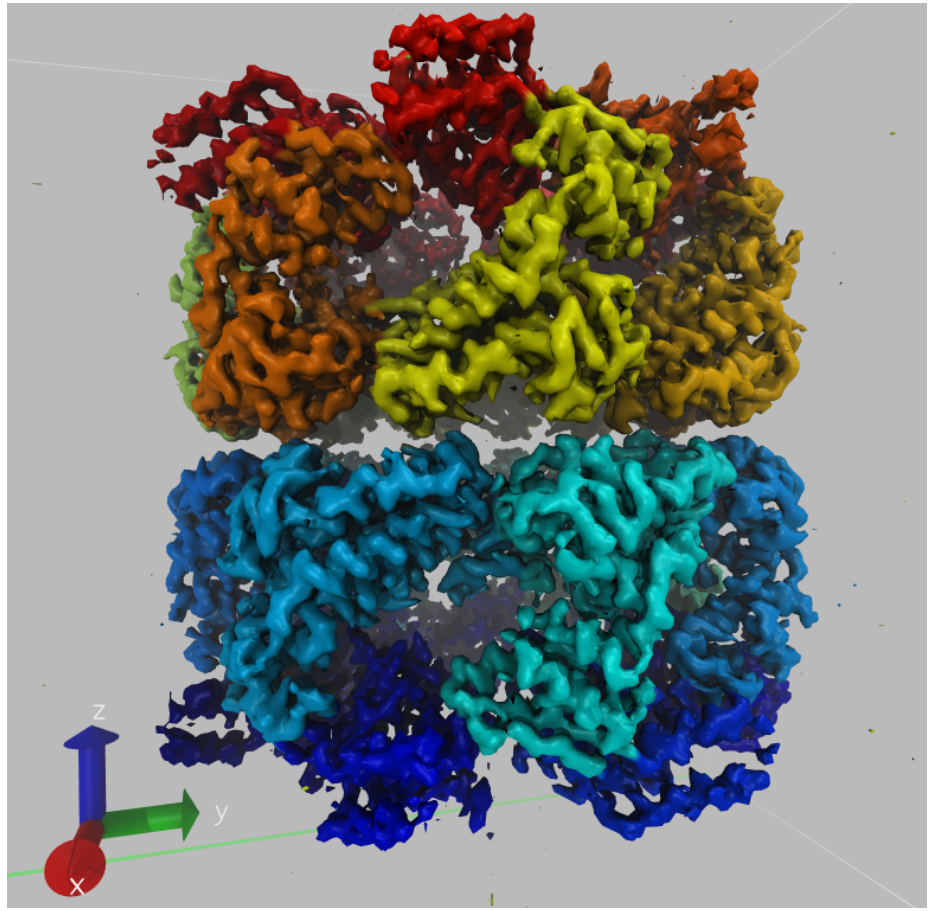
Hardware platform	Runtime,	Speedup
Intel Xeon E5-2660V3 + 1x Tesla V100	0.095,	1.0x
Cavium ThunderX2 + 1x Tesla V100	0.115s,	0.82x
IBM Power9 + 1x Tesla V100	0.118s,	0.80x

General P9 & ARM64 Performance Note:
Intel x86 CPU tests benefit from hand-coded SSE / AVX / AVX-512 vector intrinsics for acceleration of atom selections and density map statistics operations.

I haven't yet developed fully comparable hand-vectorized code for ARM NEON/SVE or POWER9 VSX instructions.

Compilers can't match hand-vectorized performance for these operations.

This covers most of the delta between x86 vs. other arch.



Example: VMD GPU-accelerated density map segmentation of GroEL

ARM+NVIDIA
Preliminary Results

VMD Tesla V100 Cross Correlation Performance

Application and Hardware platform	Runtime, Speedup vs. Chimera, VMD+GPU		
Chimera Xeon E5-2687W (2 socket) [1]	15.860s,	1x	
VMD-CUDA Intel Xeon E5-2687W + 1x Quadro K6000 [1,2]	0.458s,	35x	**1.0x**
VMD-CUDA Intel Xeon E5-2698v3 + 1x Tesla P100	0.090s,	176x	**5.1x**
VMD-CUDA IBM Power8 "Minsky" + 1x Tesla P100	0.080s,	198x	**5.7x**
VMD-CUDA Intel Xeon E5-2697Av4 + 1x Tesla V100	**0.050s,**	317x	**9.2x**
VMD-CUDA IBM Power9 "Newell" + 1x Tesla V100	**0.049s,**	**323x**	**9.3x**
VMD-CUDA ThunderX2 ARM64 + 1x Tesla V100	**0.061s,**	**250x**	**7.5x**

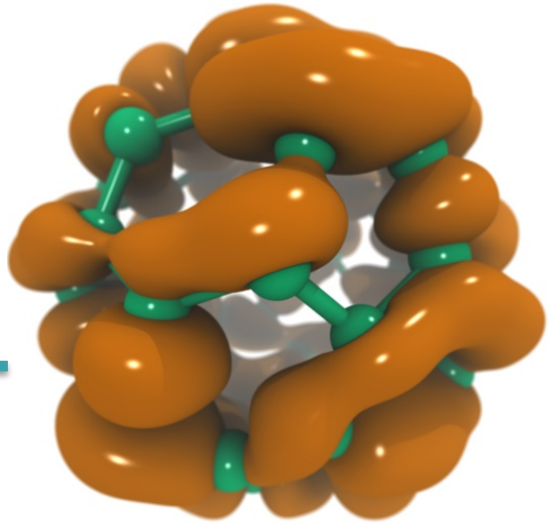
[1] GPU-Accelerated Analysis and Visualization of Large Structures Solved by Molecular Dynamics Flexible Fitting. J. E. Stone, R. McGreevy, B. Isralewitz, and K. Schulten. Faraday Discussions 169:265-283, 2014.

[2] Early Experiences Porting the NAMD and VMD Molecular Simulation and Analysis Software to GPU-Accelerated OpenPOWER Platforms. J. E. Stone, A.-P. Hynninen, J. C. Phillips, K. Schulten. International Workshop on OpenPOWER for HPC (IWOPH'16), LNCS 9945, pp. 188-206, 2016.

ARM+NVIDIA Preliminary Results

VMD Tesla V100 Performance for C_{60} Molecular Orbitals, 516x519x507 grid

Hardware platform		Runtime,	Speedup
IBM Power8 (ORNL 'crest') + 1x Tesla K40 [1]		3.49s,	1.0x
Intel Xeon E5-2697Av4	+ 1x Tesla V100 [2]	0.610s,	5.7x
Intel Xeon E5-2697Av4	+ 2x Tesla V100 [2]	0.294s,	11.8x
Intel Xeon E5-2697Av4	+ 3x Tesla V100 [2]	0.220s,	15.9x
IBM Power9 "Newell"	+ 1x Tesla V100	0.394s,	8.8x
IBM Power9 "Newell"	+ 2x Tesla V100	0.207s,	16.8x
IBM Power9 "Newell"	+ 3x Tesla V100	0.151s,	23.1x
IBM Power9 "Newell"	+ 4x Tesla V100	0.130s,	26.8x
Cavium ThunderX2	+ 1x Tesla V100	0.565s,	6.2x
Cavium ThunderX2	+ 2x Tesla V100	0.284s,	12.2x



NVLink perf. boost

ARM+NVIDIA Preliminary Results

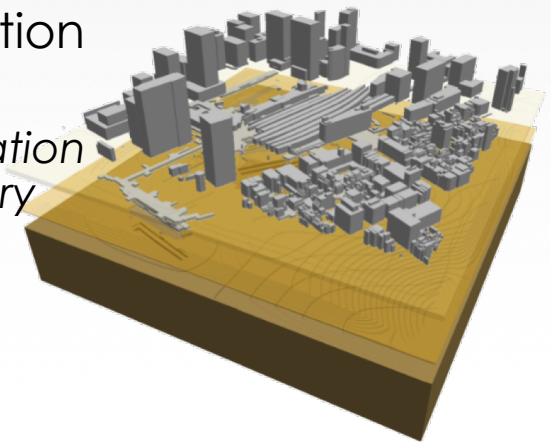
[1] Early Experiences Porting the NAMD and VMD Molecular Simulation and Analysis Software to GPU-Accelerated OpenPOWER Platforms. J. E. Stone, A.-P. Hynninen, J. C. Phillips, K. Schulten. International Workshop on OpenPOWER for HPC (IWOPH'16), LNCS 9945, pp. 188-206, 2016.
[2] NAMD goes quantum: An integrative suite for hybrid simulations. Melo et al., Nature Methods, 2018.

GAMERA

Takuma YAMAGUCHI, Kohei FUJITA, and Tsuyoshi ICHIMURA
The University of Tokyo

Application information

- Low-order unstructured finite element solver for earthquake simulations
  - Main Developer and Scientific Lead: Tsuyoshi ICHIMURA, The University of Tokyo
- Domain & Method
  - Earth Science
- Programming Models
  - CUDA Fortran + MPI
  - OpenACC + MPI
- Diagram of the Application
  - *Target model for Earthquake city simulation with complex geometry*



ARM + NVIDIA experience

- What was the experience for porting the application to ARM + NVIDIA
  - Use CUDA Fortran with PGI compiler
  - The standard build procedure for Linux systems + NVIDIA GPUs works
- Summit vs Wombat
  - Software stack readiness: good
  - Correctness: good
- Comparable performance to x86 systems per GPU in ground shaking analysis (39,191,319 DOF,
  3 types of material properties, 25 timesteps)

Computation environment	time (s)	Speedup
1 node x (2 x Intel Xeon Gold 6148 + 4 x V100 GPU)	47.75	x 1.0
2 node x (2 x Cavium ThunderX2 + 2 x V100 GPU)	50.16	x 0.951

- Next steps or forward looking items
  - Try OpenACC-based codes

ARM+NVIDIA
Preliminary Results

Additional Reports

RIKEN: Apollo 70+NVIDIA GPU testbed

GENESIS
Generalized-ensemble simulation system

GENESIS
(GENeralized-Ensemble Simulation System)

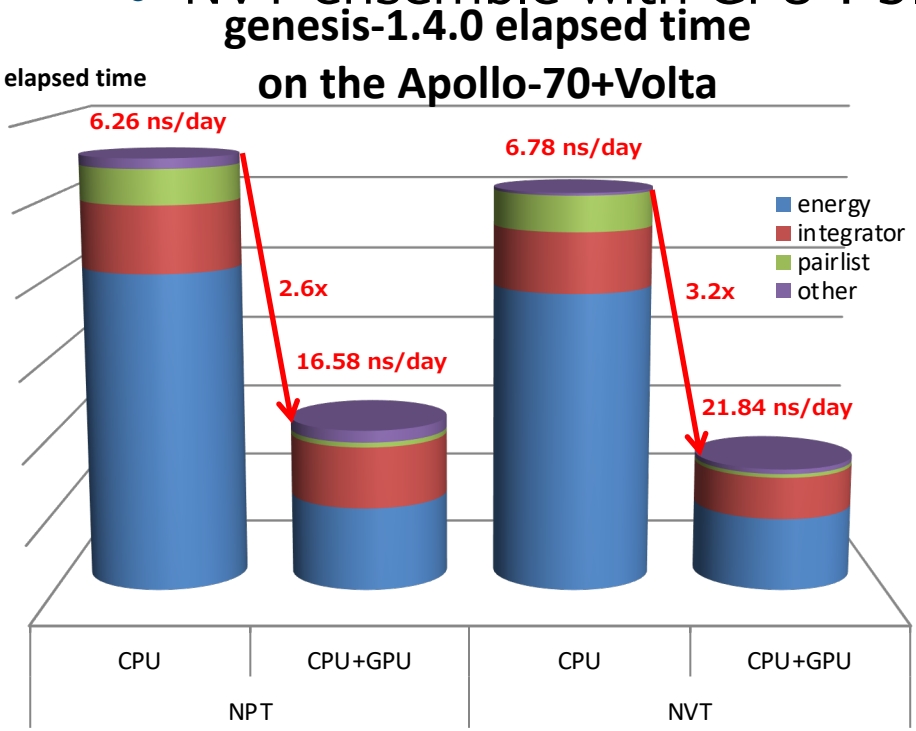
- Molecular dynamics software for simulating biomolecular systems
- Open source software distributed under the GPLv2 license
- Enhanced conformational sampling algorithms for solving the time-scale problem
- Efficient parallelization for simulating large systems on massively parallel supercomputers
- Algorithms for simulating different resolution molecular models such as coarse-grained, all-atom, and QM/MM

References: J. Jung et al., *WIREs Comput. Mol. Sci.*, 5, 310-323 (2015),
C. Kobayashi et al., *J. Comput. Chem.*, 38, 2193-2206 (2017)

Website: https://www.r-ccs.riken.jp/labs/cbrt/

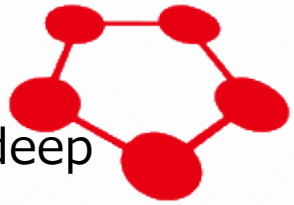
Performance of GENESIS on the Apollo70+GPU

- No optimization done on GENESIS for this measurements.
- Because ThnderX2 is a general purpose CPU, it was easy to build an execution environment of GENESIS with CUDA and gcc / OpenBLAS / OpenMPI.
- Confirmed sample programs accelerated by using GPUs.
  - NPT ensemble with GPU : 2.6x on ns/day
  - NVT ensemble with GPU : 3.2x on ns/day



GENESIS has been developed by Computational Biophysics Research Team in RIKEN Center for Computational Science. Calculations were performed by Dr. Akiyoshi Kuroda from Operations and Computer Technologies Division in RIKEN Center for Computational Science.

Chainer



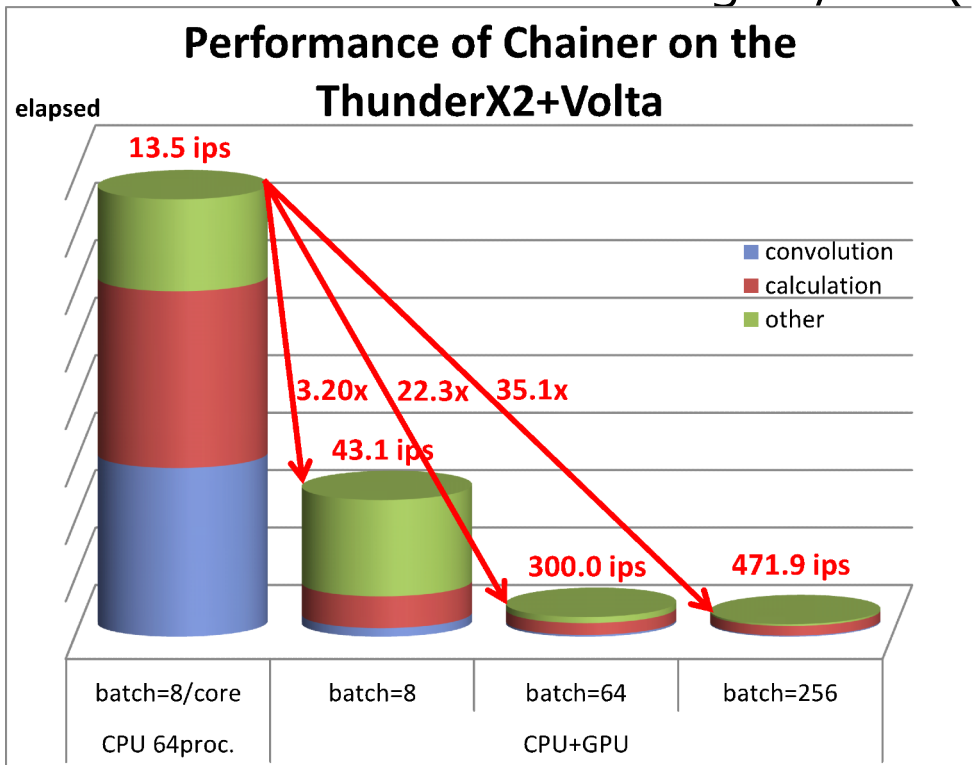
- **About Chainer from web page https://chainer.org/**

  Bridge the gap between algorithms and implementations of deep learning
  - Powerful
    - Chainer supports CUDA computation. It only requires a few lines of code to leverage a GPU. It also runs on multiple GPUs with little effort.
  - Flexible
    - Chainer supports various network architectures including feed-forward nets, convnets, recurrent nets and recursive nets. It also supports per-batch architectures.
  - Intuitive
    - Forward computation can include any control flow statements of Python without lacking the ability of backpropagation. It makes code intuitive and easy to debug.

- **Characteristic of Chainer**
  - For general purposes
  - Described in Python
  - Flexible for various environments by using numpy, mpi4py, …
  - Open source with MIT license

Performance of Chainer on the Apollo70+GPU

- Chainer: Ver.6.5.0 (ChainerX, ChainerMN)
- No optimization / tuning added.
- It was easy to build the environment by using cuda, cublas, cuDNN, NCCL from NVIDIA.
- Results : x 35.1 improvement on GPU (# images / sec)
  - 2 CPUs : 13.5 images/sec
  - 2 GPUs : 479.1 images /sec (maximum)



- CPU measurement (2 CPUs)
  - 64 process flatMPI,
  - batch size/core: 8, batch size/node : 256
- GPU measurement (2 GPUs)
  - 2 process MPI,
  - batch size/process: 8, 64, 256
- No IO time for getting image.

Calculations were performed by Dr. Akiyoshi Kuroda from Operations and Computer Technologies Division in RIKEN Center for Computational Science.