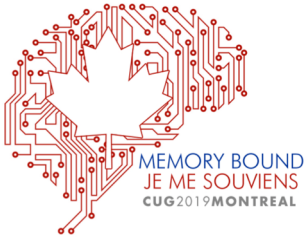
CRAY USER GROUP MEETING 2019 (CUG 2019)



Argonne NATIONAL LABORATORY

ROOFLINE-BASED PERFORMANCE EFFICIENCY OF HPC BENCHMARKS AND APPLICATIONS ON CURRENT GENERATION OF PROCESSOR ARCHITECTURES

JAEHYUK KWACK*, THOMAS APPLENCOURT, COLLEEN BERTONI, YASAMAN GHADAR, HUIHUO ZHENG, CHRISTOPHER KNIGHT, AND SCOTT PARKER

Argonne Leadership Computing Facility
Argonne National Laboratory

U.S. DEPARTMENT OF ENERGY Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

May 5th - 9th , 2019 in Montreal, Canada

INTRODUCTION

Supercomputers with the cutting-edge technology

- **TOP500 – mostly about Peak Flop-rates**

Rank	System	Cores	Rmax (TFlop/s)
1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,397,824	143,500.0
2	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0

- **HPCG list – mostly about memory bandwidths**

Rank	TOP500 Rank	System	Cores
1	1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,397,824
2	2	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480

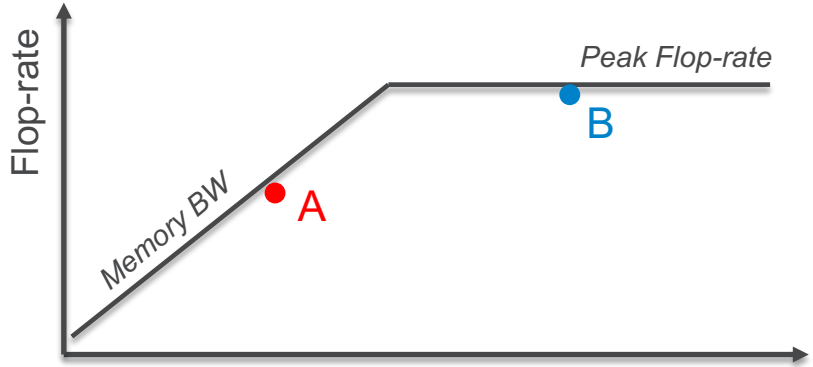
- **Argonne, Cray and Intel have collaborated for an Exa-scale system (Aurora)**

- **We are interested in general state of processor performance for future**

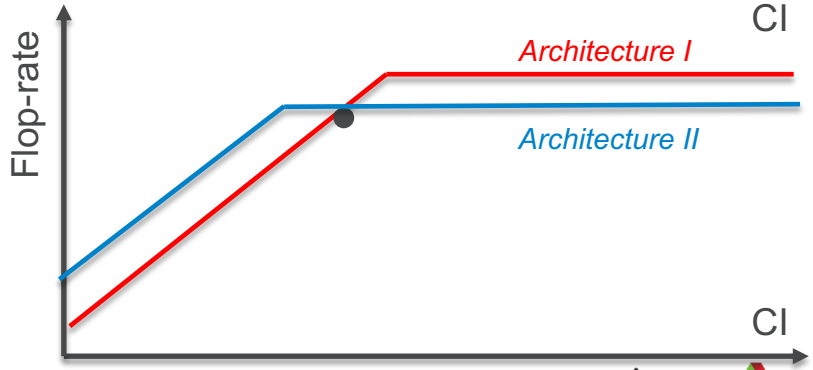
INTRODUCTION

Why does Roofline-based Performance Efficiency matter?

- On a given processor architecture, applications' performance can be bound by
  - Memory Bandwidth (e.g., application A), or
  - Peak Flop-rates (e.g., application B).



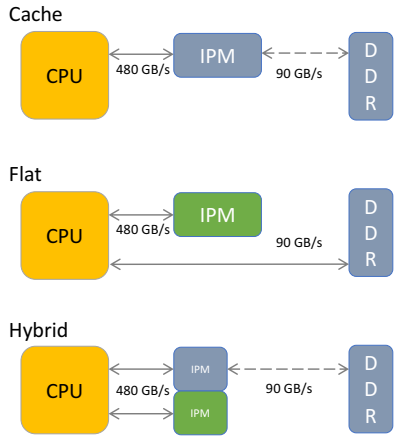
- Your application performance can be bound by
  - Memory Bandwidth on architecture I, or
  - Peak Flop-rates on architecture II.

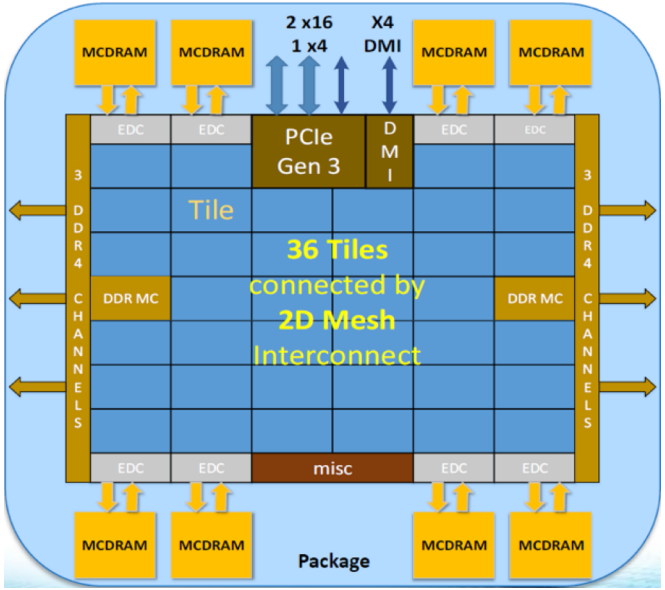


EMPLOYED PROCESSOR ARCHITECTURES

INTEL XEON PHI KNL PROCESSOR

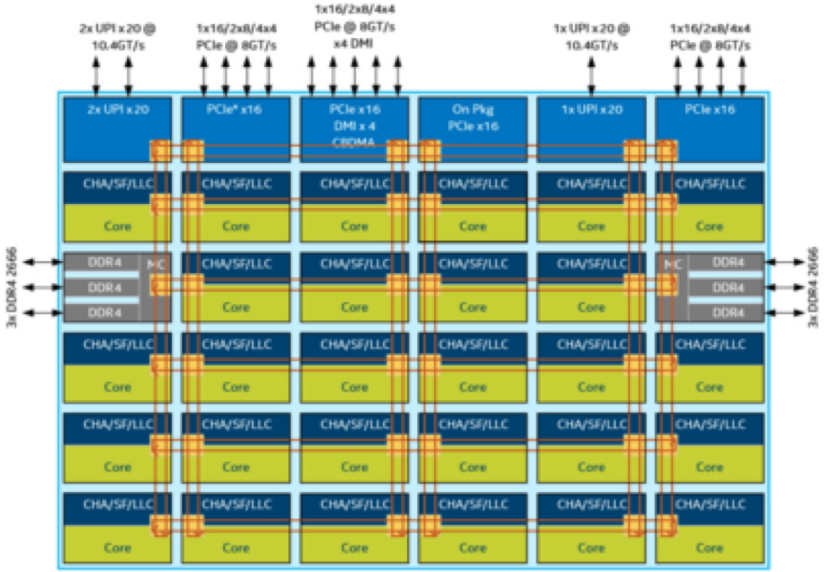
- ALCF Theta system
  - Cray XC40 system
  - 4,392 KNL 7230 processors w/ a peak of 11.69 PF
  - 192GB DDR / node
- Intel KNL 7320 processor
  - 32 tiles with 2 cores/tile (64 cores in total) (14nm)
  - 32 KB L1 data cache/core
  - 1 MB L2 data cache/tile
  - 16 GB MCDRAM on chip
  - AVX-512 instructions
  - Two instructions/clock cycle
  - SMT-4 mode (i.e., 4 hyper-treads/core)
  - 1.3 GHz reference frequency
- Memory configurations
  - Cache / Flat / Hybrid mode





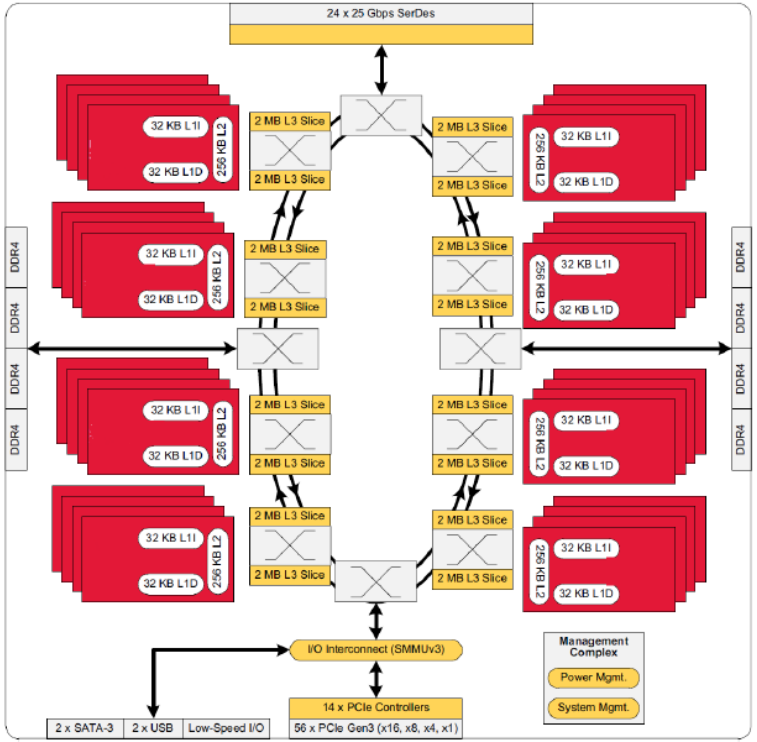
INTEL XEON SKYLAKE PROCESSOR

- ANL JLSE (Joint Laboratory for System Evaluation) system (Skylake partition)
  - Dual-socket Intel Xeon 8180M processor node
  - 395 GB DDR / node
- Intel Xeon Platinum 8180M processor
  - 28-core x86 Skylake processor (14 nm+)
  - 2 AVX-512 FMA units / core
  - Three UPI (Ultra Path Interconnect) links
  - 2.5 GHz reference frequency
  - 205W / socket
  - 32 KB L1 data cache/core
  - 1 MB L2 data cache/core
  - 38.5 MB L3 data cache/socket
  - 6 memory channels
  - SMT-2 mode (i.e., 2 hyper-threads/core)



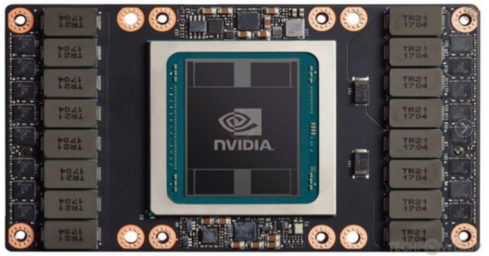
ARM MARVELL THUNDER X2 PROCESSOR

- ANL JLSE (Joint Laboratory for System Evaluation) system (Comanche partition)
  - Dual-socket Marvell ThunderX2 processor nodes
  - 217GB DDR / node
- Arm Marvell ThunderX2 CN9975 processor
  - 28-core Arm v8.1 processor (16nm)
  - 2 NEON 128-vectors engines/core
  - CCPI2 (Cavium Coherent Processor Interconnect) link
  - 2.2 GHz reference frequency (2.5 GHz on Turbo mode)
  - 170W / socket
  - 32 KB L1 data cache/core
  - 256 KB L2 data cache/core
  - 32 MB L3 data cache/socket
  - 8 memory channels
  - SMT-2 mode (i.e., 2 hyper-threads/core, up to 4 hyper-threads/core (SMT-4) available)



NVIDIA TESLA V100 SXM2 GPU

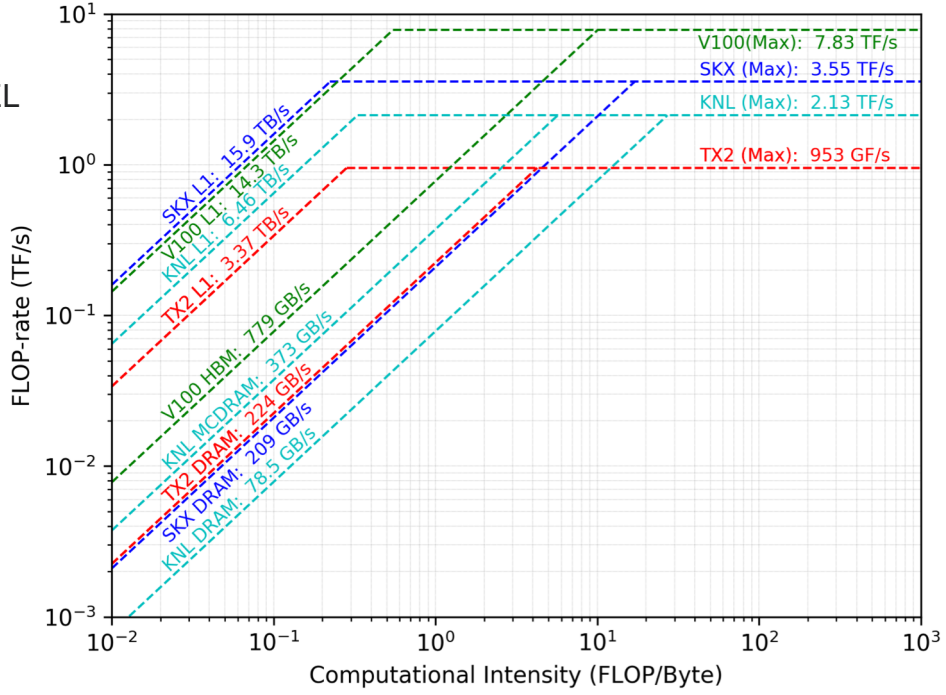
- ANL JLSE (Joint Laboratory for System Evaluation) system (NVIDIA V100 SXM2 GPU partition)
  - Dual-socket Intel Xeon Gold 6152 processors
  - 4 NVIDIA Tesla V100 SXM2 GPUs
  - NVLINK among 4 GPUs
  - PCIe 3.0 between GPUs and CPUs
  - 197GB DDR / node
- NVIDIA V100 SXM2 GPU
  - 80 Streaming Multiprocessors (SMs) per GPU (12nm)
    - 32 FP64, 64 FP32, 64 INT32 CUDA cores/SM
    - 8 tensor cores/SM
  - 1.53 GHz maximum frequency
  - 250W / socket
  - 128 KB L1 data cache/SM
  - 6 MB L2 data cache/socket
  - 4 stacks of HBM2 (32GB)/socket



MEASURE PEAK PERFORMANCE

- Via Empirical Roofline Tool [1]
  - ERT CFLAGS for KNL: -O3 -fno-alias -fno-fnalias -xMIC-AVX512 -DERT INTEL
  - ERT CFLAGS for SKX: -O3 -fno-alias -fno-fnalias -xCORE-AVX512 -qopt-zmm-usage=high -DERT INTEL
  - ERT CFLAGS for TX2: -Ofast -mcpu=thunderx2t99 -fsimdmath
  - ERT CFLAGS for V100: -O3
  - ERT GPU CFLAGS for V100: -x cu
- TX2 peak flop-rate from DGEMM
- V100 L1 is the theoretical peak.

	Flop-rate (TF/s)	L1 (TB/s)	L2 (TB/s)	LLC (GB/s)	DRAM (GB/s)
KNL	2.13	6.46	1.911	373	78.5
Dual SKX	3.55	15.91	4.55		209
Dual TX2	0.953	3.37	2.63	1091	224
V100	7.83	14.336	3.35		779



BENCHMARK/APPLICATION PERFORMANCE

THE EMPLOYED TEST SUITE

HPC Benchmarks and Applications

- HPGMG: an ECP proxy application
- NEKBONE: an ECP proxy application and DOE CORAL-2 benchmark
- GAMESS: an ECP application
- LAMMPS: an ECP application and DOE CORAL-2 benchmark
- QMCPACK: an ECP application and DOE CORAL-2 benchmark
- Qbox: an ECP application

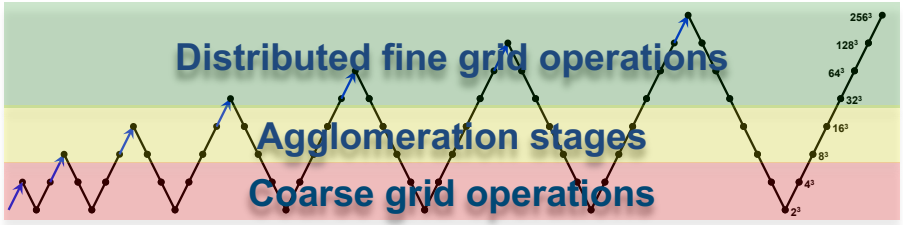
  - DOE: U.S. Department of Energy
  - ECP: Exascale Computing Project
  - CORAL: Collaboration of Oak Ridge, Argonne, and Livermore

PoC: JaeHyuk Kwack

HPGMG

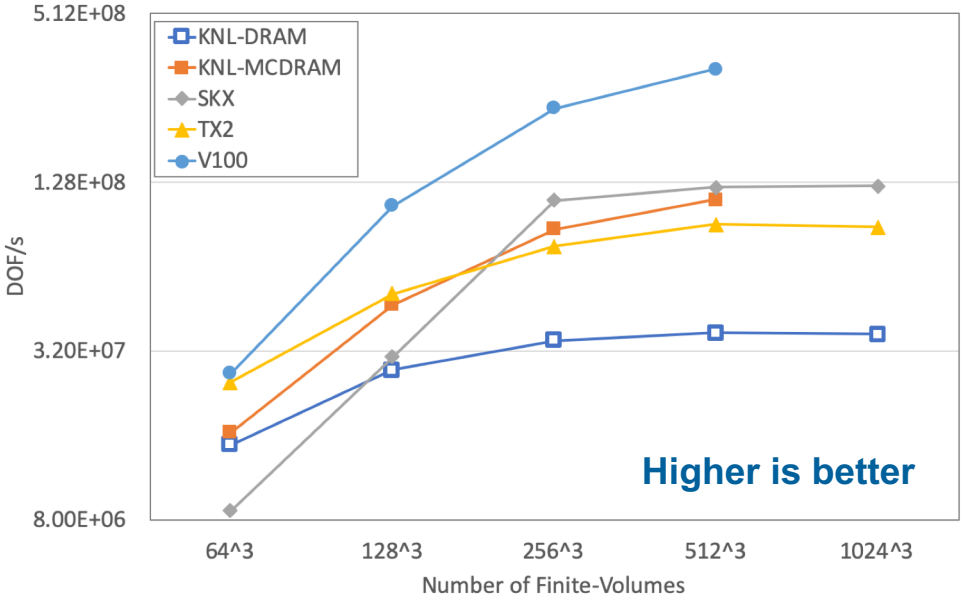
High Performance Geometric Multi-Grid Benchmark [2][3][4][5]

- HPGMG-FE(Finite Element): compute-intensive and cache-intensive
- HPGMG-FV(Finite Volume): memory bandwidth-intensive
  - Used for the official list
  - Solving an elliptic problem on isotropic Cartesian grids with 4th order accuracy
  - 4× FP ops, 3× MPI messages, 2× MPI message size w/o DRAM data movement compared to 2th order HPGMG-FV
  - Employing the Full Multi-grid (FMG) F-cycle
  - A series of progressively deeper geometric multi-grid V-cycles



HPGMG-FV

- Source
  - MPI+OpenMP version (commit: a0a5510) [6]
  - MPI+CUDA version (commit: 5ad473d) [7]
- Compilers
  - KNL / SKX : Intel 19.0.3.199
  - TX2: Arm Compiler version 19.0
  - V100: CUDA V10.0.130



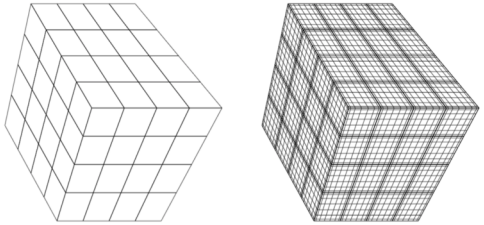
- Inputs

Number of Finite-Volumes	Multi-grid Levels	Degrees-of-Freedom	Numerical Errors
64^3	6	2.62E+05	6.93E-05
128^3	7	2.10E+06	7.45E-06
256^3	8	1.68E+07	5.14E-07
512^3	9	1.34E+08	4.15E-08
1024^3	10	1.07E+09	5.15E-09

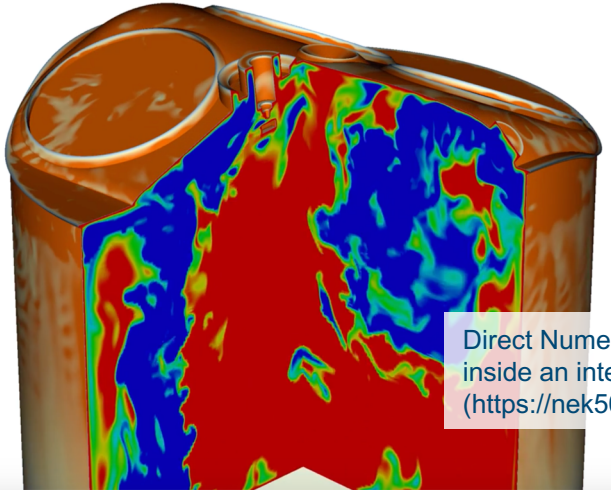
- Runtime configurations

Processor	Number of MPI ranks	Number of Threads per MPI rank	Total Threads
KNL	64	1	64
SKX	16	7	112
TX2	16	7	112
V100	1	7	all GPU cores

NEKBONE [8]



- A mini-app derived from the Nek5000 [9] CFD code which is a high order, incompressible Navier-Stokes CFD solver based on the spectral element method.
- Standard Poisson equation in a 3D box domain with a block spatial domain decomposition among MPI ranks.
- Solution phase: conjugate gradient iterations in an element-by-element fashion.
  - Vector operations
  - Matrix-matrix multiply operations
  - Nearest- neighbor communication
  - MPI Allreduce operations.
- Source:
  - written in C and Fortran
  - MPI+OpenMP



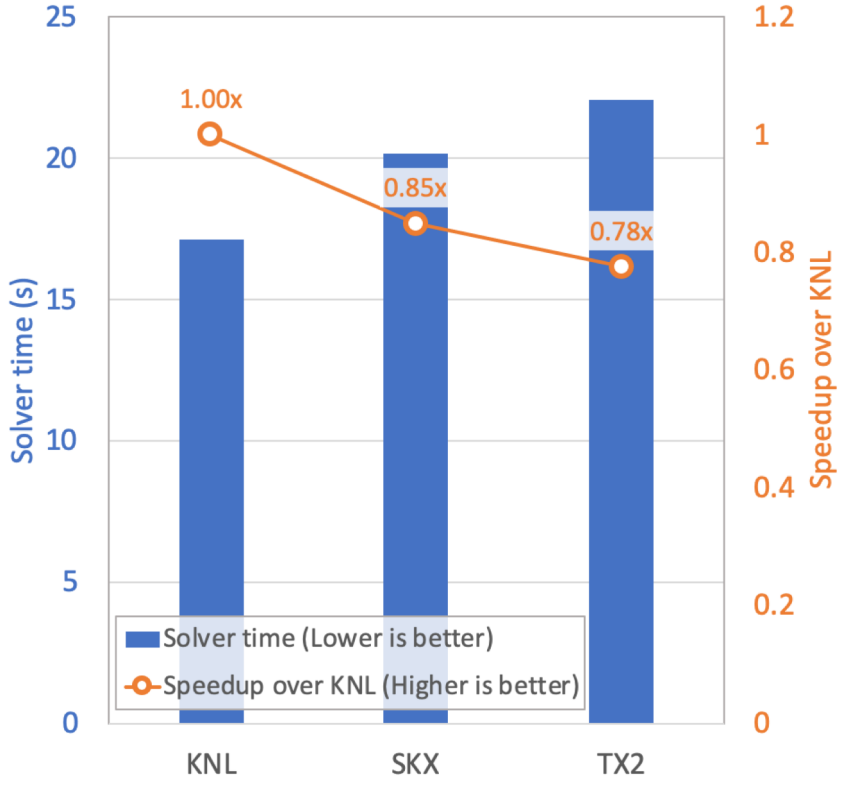
Direct Numerical Simulation of the flow inside an internal combustion engine (https://nek5000.mcs.anl.gov)

NEKBONE

- Input
  - a total of 8960 spectral elements
  - 12 grid points in each direction within an element
- Runtime configurations
  - KNL: 1 MPI rank + 128 OpenMP threads/MPI
  - SKX: 2 MPI ranks + 56 OpenMP threads/MPI
  - TX2: 2 MPI ranks + 56 OpenMP threads/MPI

- NEKBONE solver time

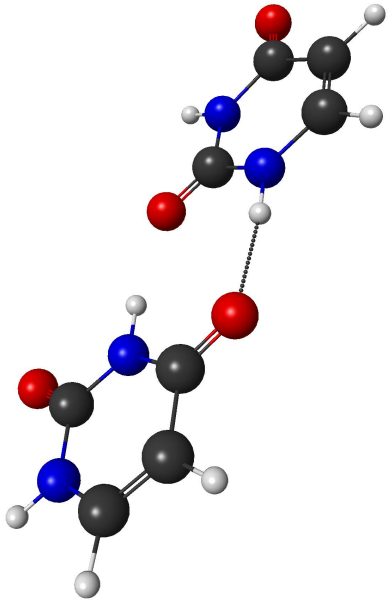
Processor	Solver Time (s)	Ranks	Thds/Rank	El./Rank
KNL	17.11	1	128	8960
SKX	20.15	2	56	4480
TX2	22.07	2	56	4480



GAMESS

General Atomic and Molecular Electronic Structure System

- A general quantum chemistry and *ab initio* electronic structure code [10][11].
  - *ab initio* SCF energies (e.g. RHF and MCSCF)
  - Force fields (e.g., the Effective Fragment Potential)
  - Perturbative corrections to Hartree-Fock (e.g., MP2 and RI-MP2)
  - Near-linear scaling fragmentation methods (e.g., Fragment Molecular Orbital (FMO) method)
  - *ab initio* gradients, hessians, and geometry optimizations.
- Source
  - Mainly written in Fortran
  - A MPI parallelization library (DDI library) written in C
  - An optional C++ library with re-implementations of certain methods
  - MPI + X
    - OpenMP for CPU cores
    - CUDA for GPU accelerators.



Generated by wxMacMolPlt [12]

GAMESS

Runtime configurations

- Two groups of MPI ranks
  - A half for ”compute processes” to perform the chemistry algorithms
  - Another half for ”data servers” to handle distributed memory and dynamic load-balancing.

- Via over-subscription, a physical core serves as a compute process as well as a data server

- MPI-only
  - 128 MPIs (64 compute + 64 data) for KNL
  - 112 MPIs (56 compute + 56 data) for SKX/TX2

- MPI+X: 2 MPIs (1 compute + 1 data)
  - 256 OpenMP threads for compute on KNL
  - 112 OpenMP threads for compute on SKX/TX2

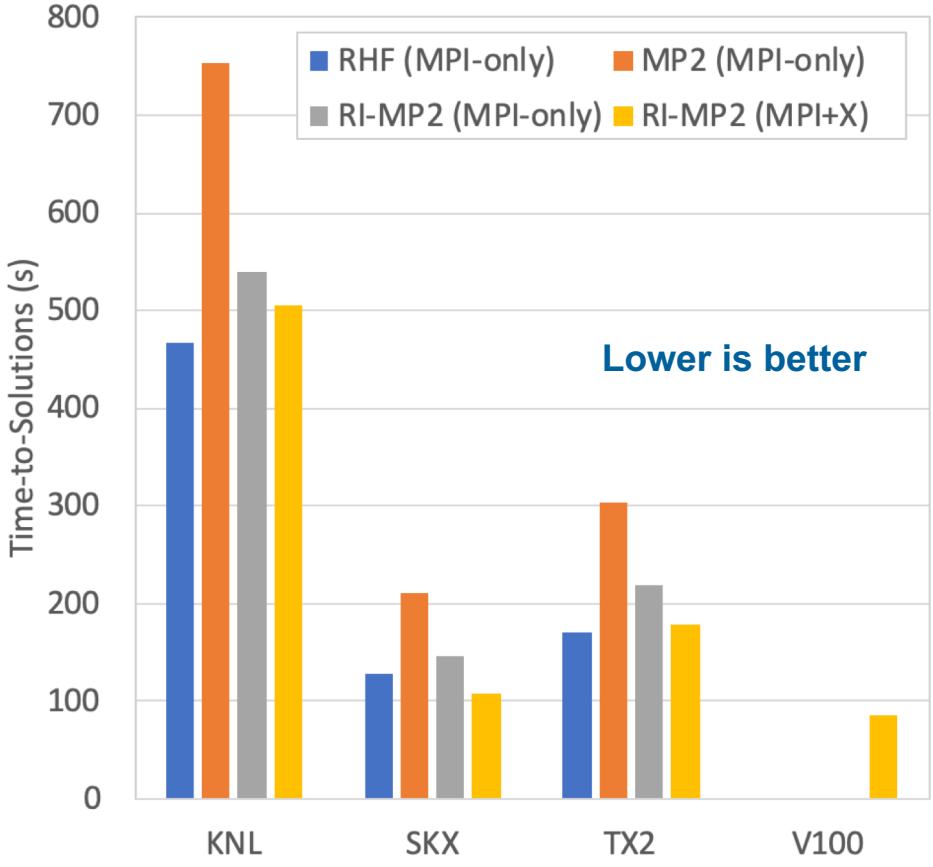
GAMESS

Benchmark results

- Inputs
  - RHF (energy) for KNL/SKX/TX2
  - MP2 (energy) for KNL/SKX/TX2
  - RI-MP2 (energy) for KNL/SKX/TX2/V100

- Average Speedup over KNL

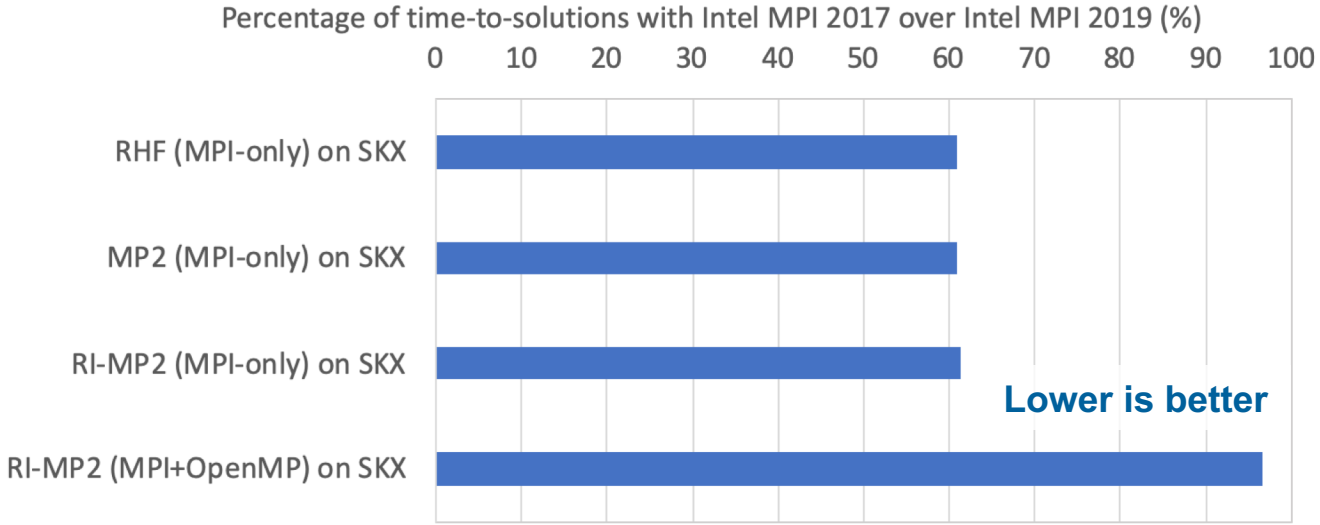
	Average Speedup over KNL
KNL	1.0 X
SKX	3.9 X
TX2	2.6 X
V100	5.9 X



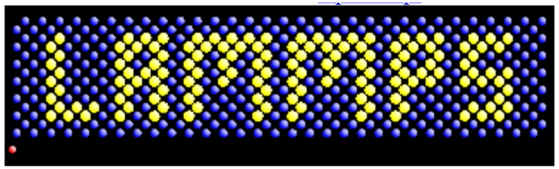
GAMESS

Intel MPI 2017 vs. 2019

- In Intel MPI 2019, the flag “I_MPI_WAIT_MODE” has been removed.
- This flag has an effect on GAMESS performance when oversubscribing cores, since it allows the data servers to wait for messages instead of polling the fabric.



PoC: Yasaman Ghadar, Christopher Knight

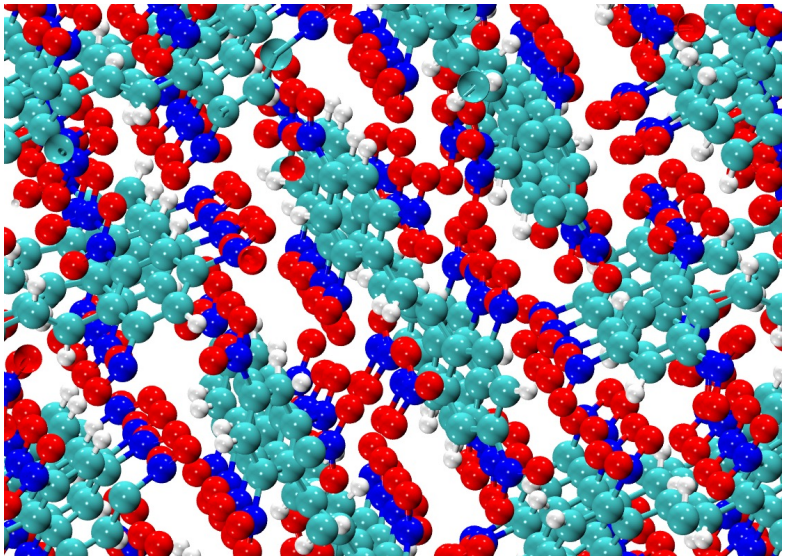


LAMMPS

A Molecular Simulation Code

- A molecular simulation code commonly used for modeling various states of matter (liquids, surfaces, solids, biopolymers) and supports multiple physical models, particle types, and sampling methods [13][14].

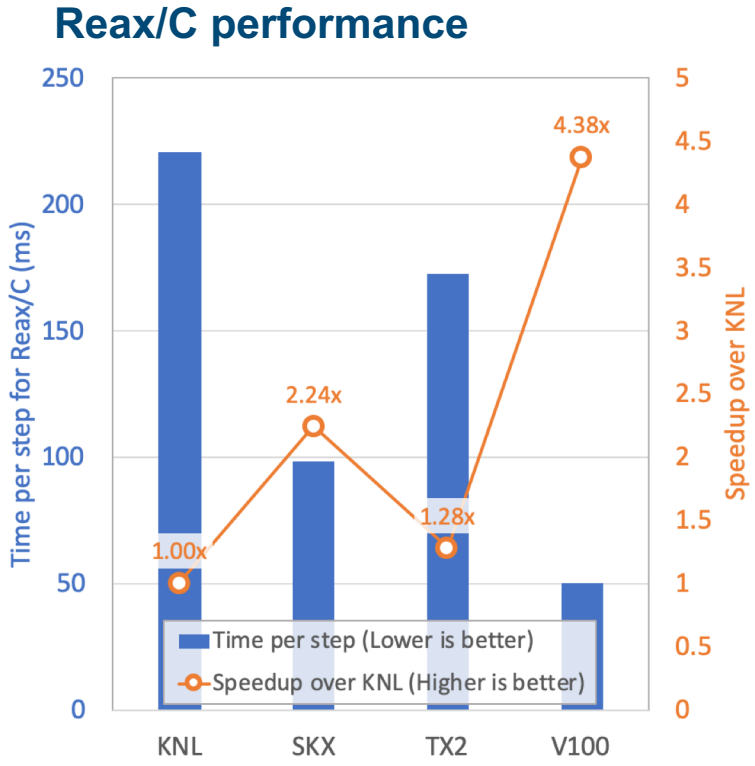
- Source
  - Written in C/C++
  - Parallelized with MPI + X
    - X for OpenMP, CUDA/OpenCL, Kokkos and explicit vectorization.
  - An unaltered version of LAMMPS, 19Feb19,
  - Used for analysis of the reactive forcefield ReaxFF using the DOE CORAL-2 LAMMPS benchmark.



LAMMPS

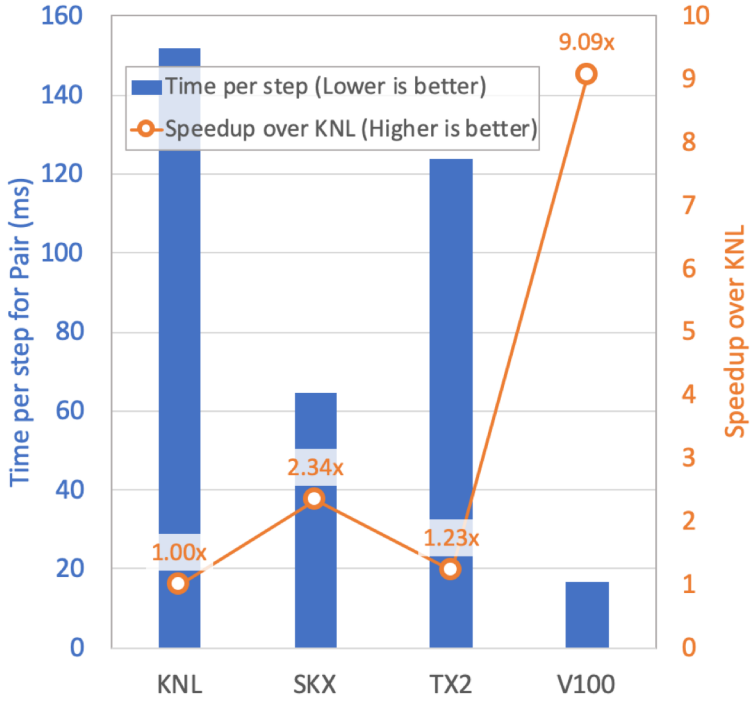
Benchmark results

- Input
  - Analysis of the reactive forcefield ReaxFF using the DOE CORAL-2 LAMMPS benchmark
  - 36,480 particles
- Runtime configurations
  - KNL: 32 MPIs + 4 OpenMP threads/MPI
  - SKX: 28 MPIs + 4 OpenMP threads/MPI
  - TX2: 14 MPIs + 8 OpenMP threads/MPI
  - V100: 1 MPI with Kokkos

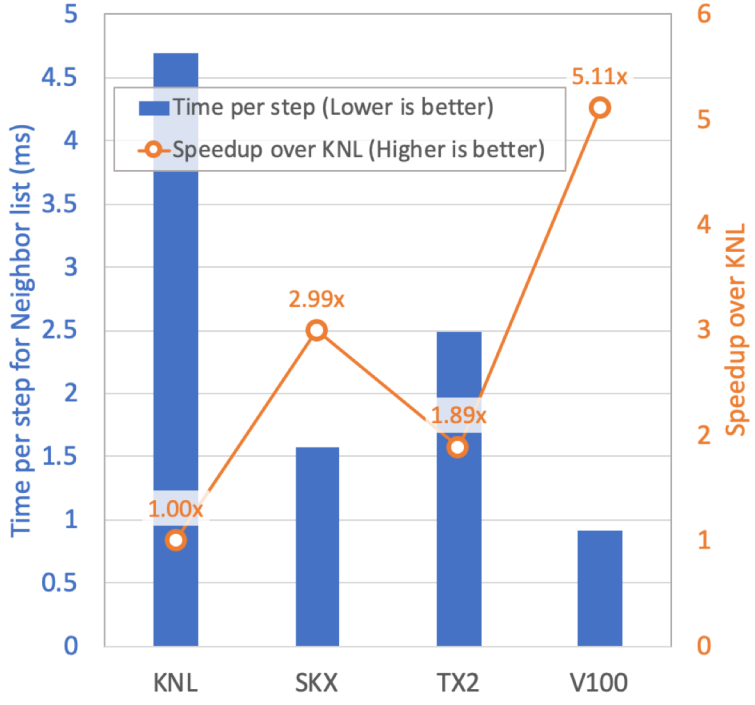


LAMMPS

Pair performance



Neighbor list performance



QMCPACK

Quantum Monte Carlo PACKage

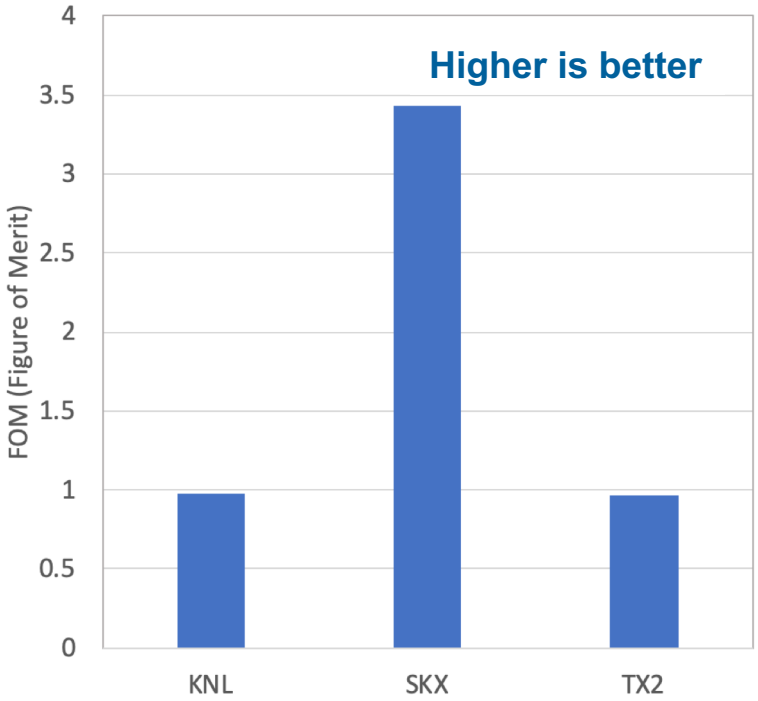
- An open source quantum Monte Carlo package [15] for *ab-initio* electronic structure calculations.
- It supports calculations of metallic and insulating solids.
- It uses a Metropolis Monte Carlo algorithm who generates samples sequentially via a random walk along a Markov chain.
- Each OpenMP thread executes an independent Markov chains or walkers. After each walker has completed a number steps, the simulation is completed. Hence, the more worker you have, the more computation you will do.
- Our figure of merit (FOM) measures how many walkers have been moved in one second.
- Version: QMCPACK v3.7.0 with SoA (i.e., Structure-of-Array)
- Input (a.k.a. S32)
  - 32 repeats of a NiO primitive cell leading to 128 atoms and 1536 electrons

QMCPACK

Benchmark results

- FOM measurement

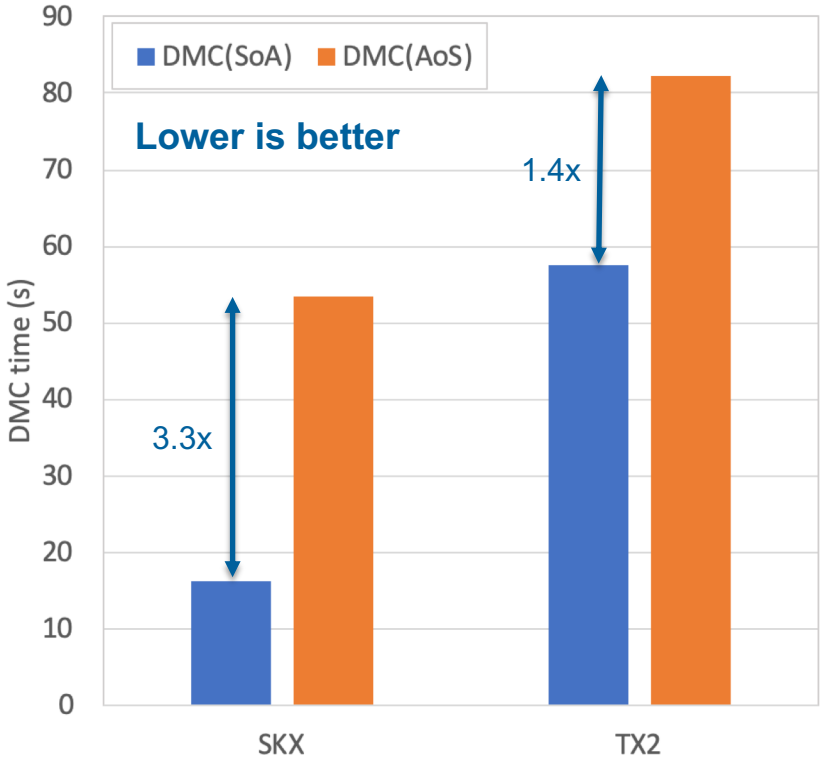
	DMC Time	Walker	Socket	FOM
KNL	65.01	64	1	0.98
SKX	16.173	28	2	3.43
TX2	57.52	28	2	0.97



QMCPACK

AoS vs. SoA

- The performance of QMCPACK has been improved by adopting SoA (Structure-of-Array) instead of AoS (Array-of-Structure).
- Since the SoA approach improves data cache hit ratio, the performance gain by SoA depends on the data cache performance.
- The speedup by SoA is much higher on SKX than on TX2, because the data cache performance of SKX is much better than the cache performance of TX2.



PoC: Huihuo Zheng

QBOX

First-Principles Molecular Dynamics

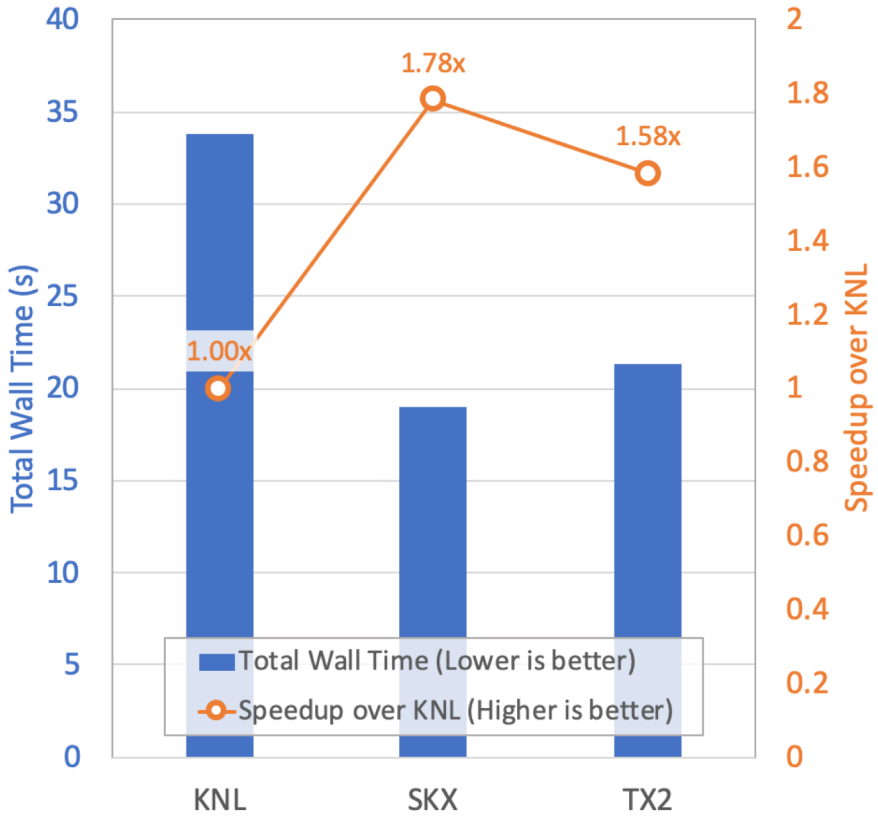
- A C++ MPI/OpenMP scalable parallel implementation of first-principles molecular dynamics based on the plane-wave, pseudopotential density functional theory formalism
- It uses FFTW for 3D Fast Fourier Transformation and ScaLAPACK for parallel dense linear algebra.
- Linking against the vendor provided libraries for FFT and ScaLAPACK
  - MKL on SKX and KNL
  - ArmPL on TX2
- Input
  - A silicon carbide periodic solid system which contains 64 atoms (32 silicon and 32 carbon atoms) and 256 electrons
  - Performing the ground state calculation using PBE0 hybrid functional
  - Total number of self-consistent iterations set to be 5
- Runtime environments
  - OMP_NUM_THREADS=1 and 1 MPI rank per core on all architectures
  - MPI processes are arranged in a two dimensional array (8 × 7 on SKX/TX2, 8 × 8 on KNL).

QBOX

Benchmark results

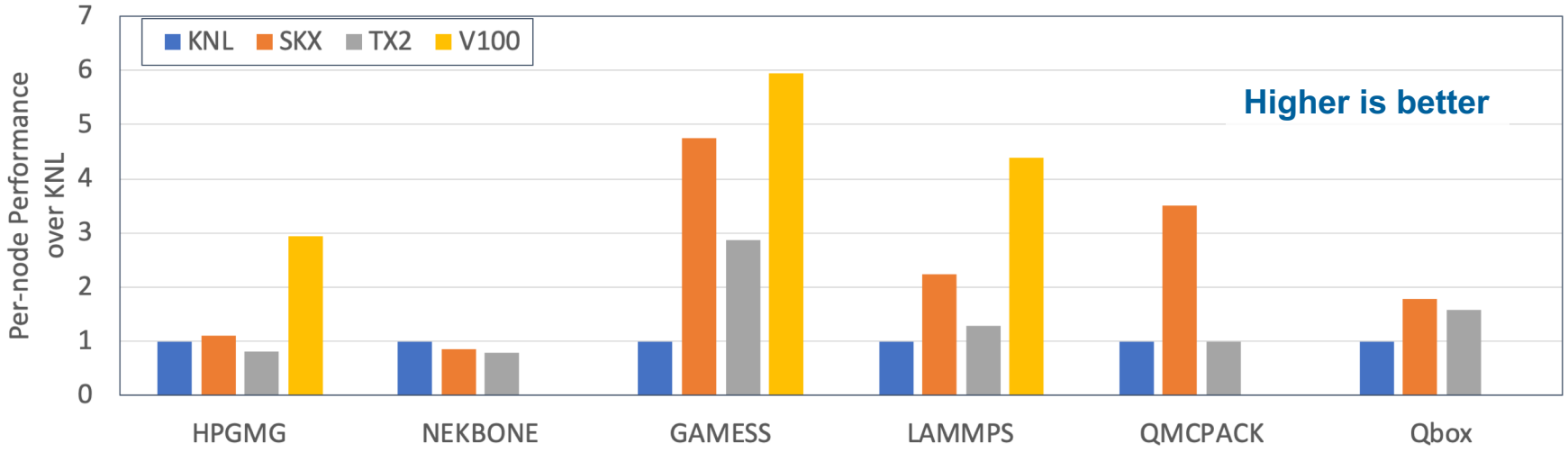
- Time-to-solutions

Kernel	KNL	SKX	TX2
exc	24.15	16.76	19.278
hpsi	2.06	0.47	0.74
wf_update	1.63	0.40	0.38
Total Walltime	33.76	18.94	21.32



SUMMARY

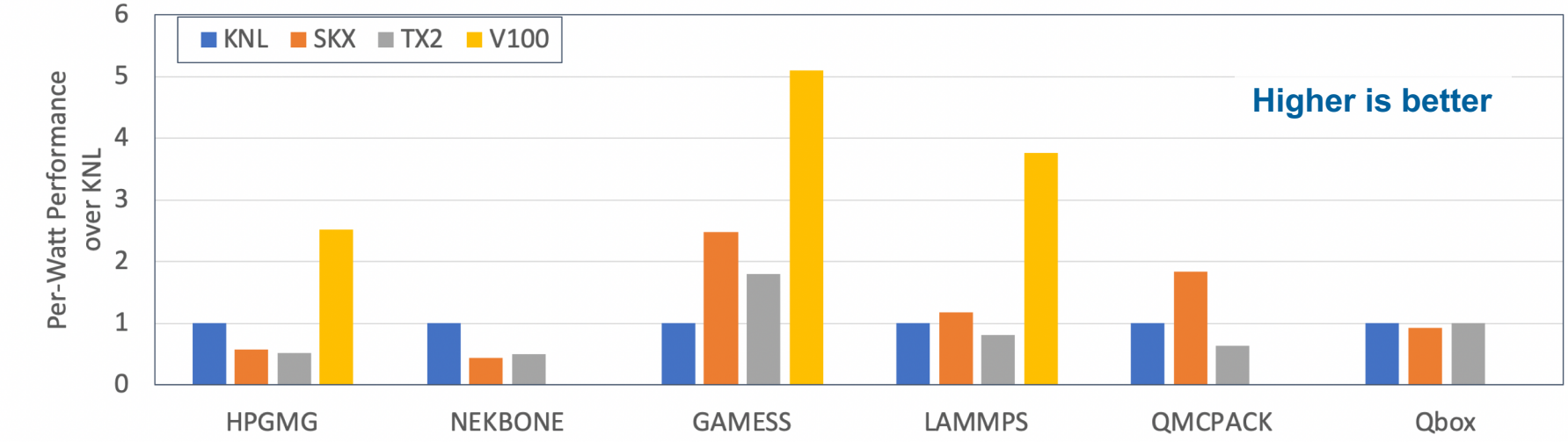
Per-node performance



SUMMARY

Per-watt performance

- TDP (Thermal Design Power)
  - KNL: 215W/socket, 215W/node
  - SKX: 205W/socket, 410W/node
  - TX2: 170W/socket, 340W/node
  - V100: 250W/socket



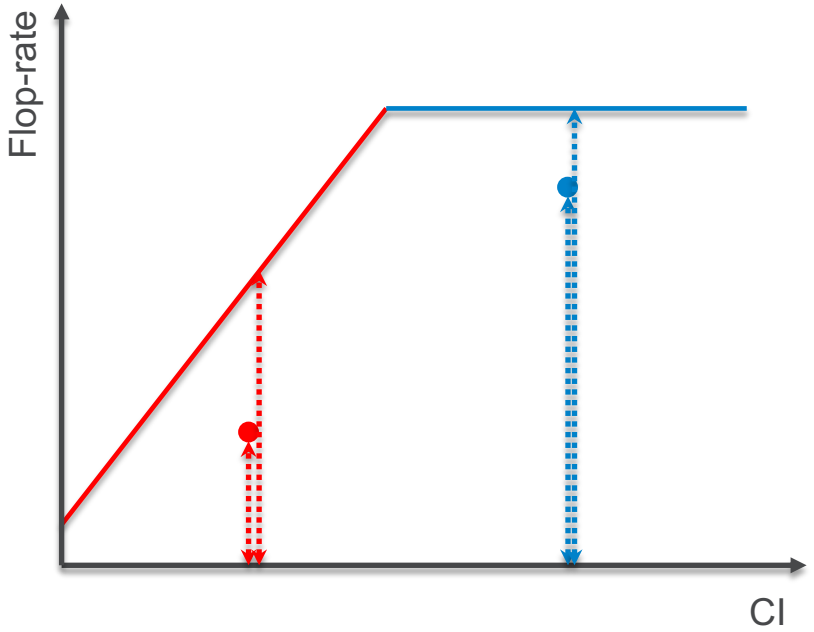
ROOFLINE-BASED PERFORMANCE EFFICIENCY

ROOFLINE-BASED PERFORMANCE EFFICIENCY

- Computational Intensity (CI)
  - CI = FLOP measurement / Data transfer
- DRAM-based CIs

	GFLOP	Memory Read/Write (GiB)	Memory-based Computational Intensity
HPGMG-FV	13303.9	13440.0	0.99
NEKBONE	2666.6	3838.0	0.69
GAMESS	9618.9	548.1	17.55
LAMMPS	4997.3	32075.7	0.16
QMCPACK	16653.5	3038.8	5.48
Qbox	997.2	2913.2	0.34

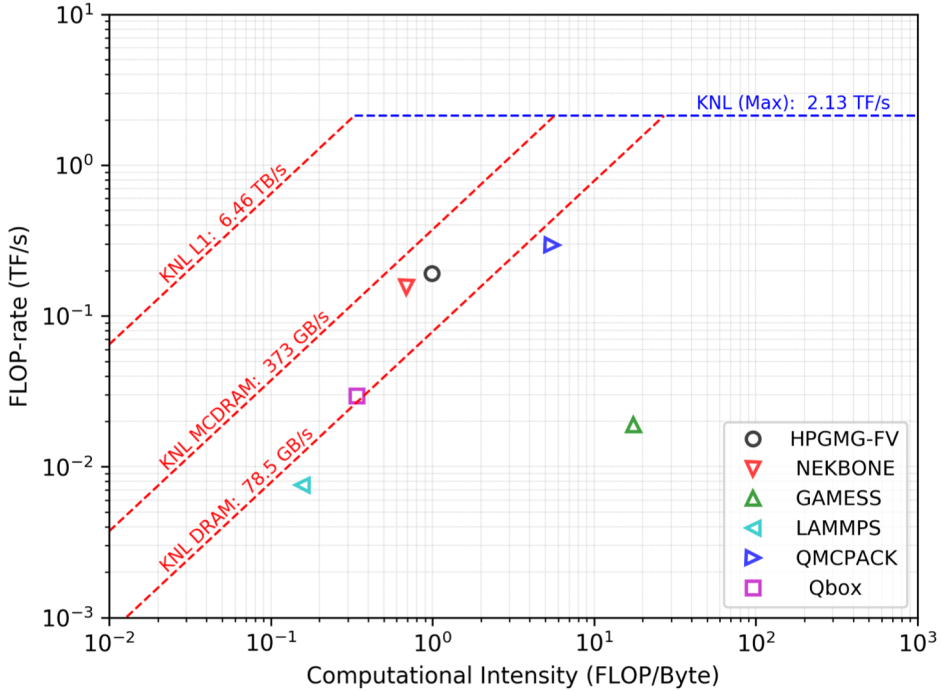
- Roofline-based Performance Efficiency [16-19]
  - Compute-bound applications
    - $Efficiency = \frac{Application\ Flop-rate}{Peak\ Flop-rate}$
  - Memory-bound applications
    - $Efficiency = \frac{Application\ Flop-rate}{Application\ CI * Memory\ BW}$



ROOFLINE-BASED PERFORMANCE EFFICIENCY

Intel Xeon Phi 7230 processor

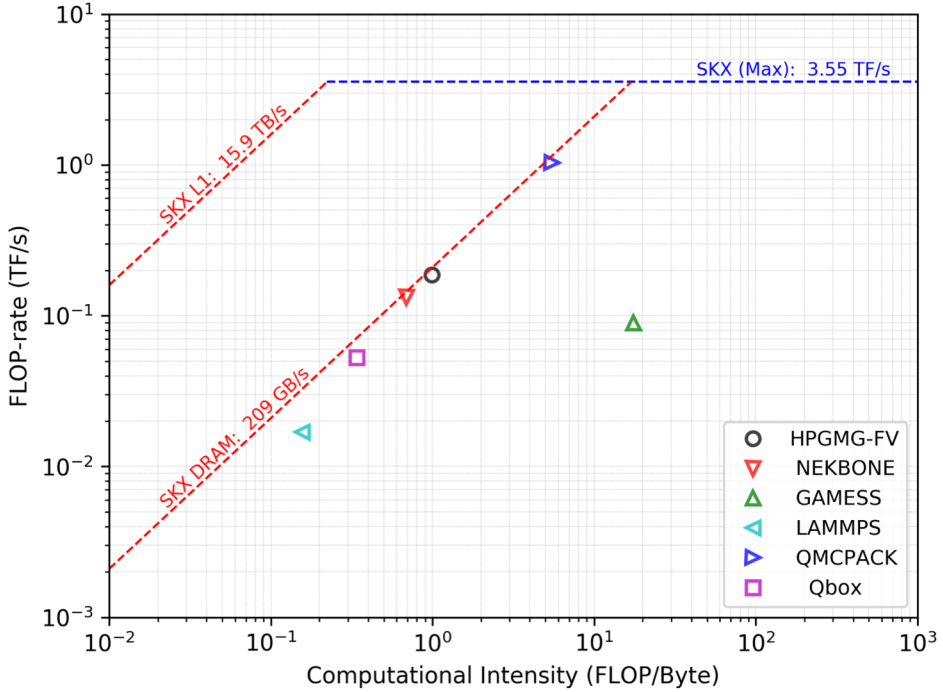
	FLOP-rates (GFLOP/s)	KNL Peak (GFLOP/s)	Efficiency (%)
HPGMG-FV	191.5	369.2	51.9%
NEKBONE	155.9	259.2	60.1%
GAMESS	19.0	2130.0	0.9%
LAMMPS	7.5	58.1	13.0%
QMCPACK	295.86	2044.2	14.5%
Qbox	29.5	127.7	23.1%



ROOFLINE-BASED PERFORMANCE EFFICIENCY

Intel Xeon Skylake 8180M processors

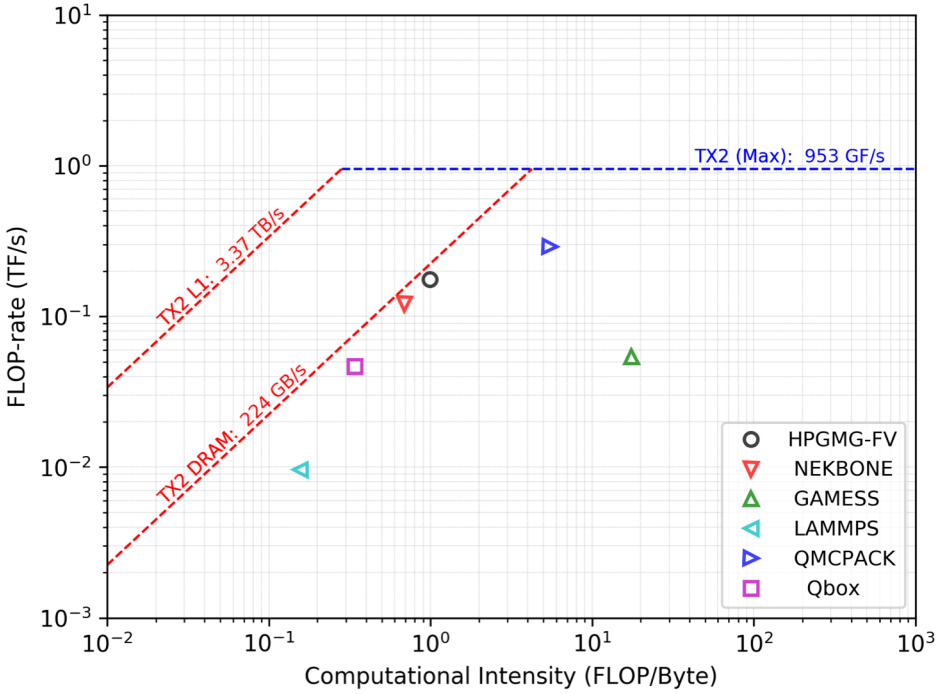
	FLOP-rates (GFLOP/s)	SKX Peak (GFLOP/s)	Efficiency (%)
HPGMG-FV	186.0	206.9	89.9%
NEKBONE	132.3	145.2	91.1%
GAMESS	90.1	3550.0	2.5%
LAMMPS	16.9	32.6	51.9%
QMCPACK	1029.7	1145.4	89.9%
Qbox	52.6	71.5	73.6%



ROOFLINE-BASED PERFORMANCE EFFICIENCY

Arm Marvell ThunderX2 processors

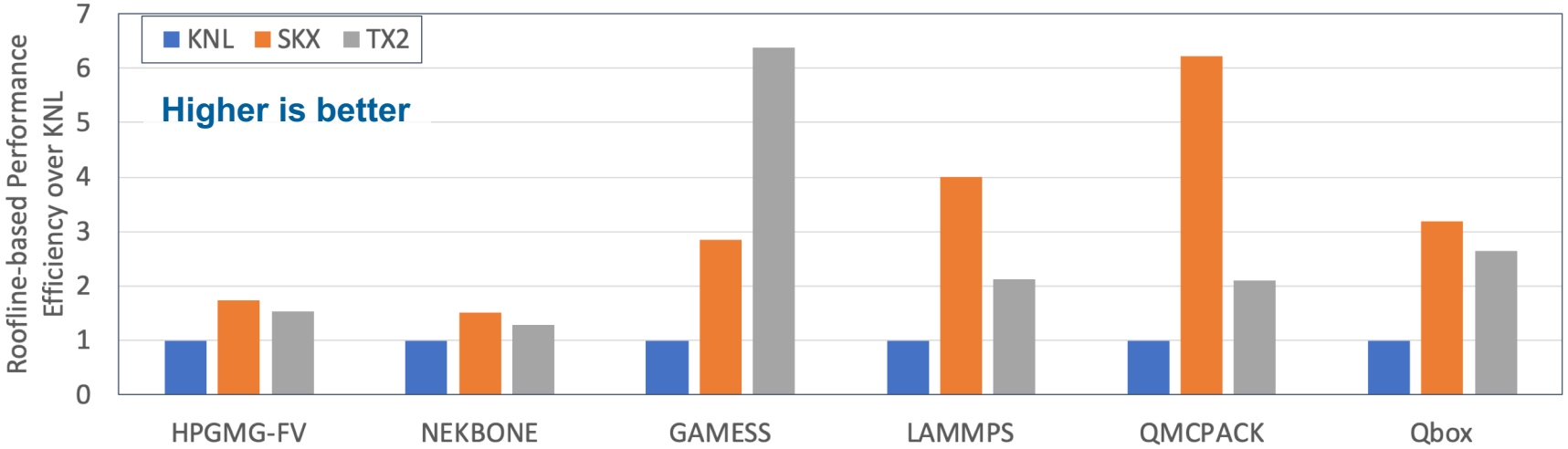
	FLOP-rates (GFLOP/s)	TX2 Peak (GFLOP/s)	Efficiency (%)
HPGMG-FV	176.9	221.7	79.8%
NEKBONE	120.8	155.6	77.6%
GAMESS	54.3	953.0	5.7%
LAMMPS	9.6	34.9	27.6%
QMCPACK	289.5	953.0	30.4%
Qbox	46.8	76.7	61.0%



ROOFLINE-BASED PERFORMANCE EFFICIENCY

Relative Roofline-based Performance Efficiency

	KNL	SKX	TX2
HPGMG-FV	1.00	1.73	1.54
NEKBONE	1.00	1.52	1.29
GAMESS	1.00	2.85	6.39
LAMMPS	1.00	4.00	2.13
QMCPACK	1.00	6.21	2.10
Qbox	1.00	3.18	2.64

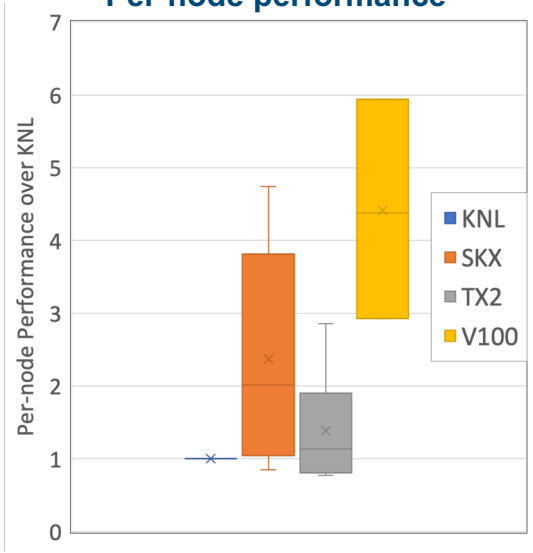


CONCLUDING REMARKS

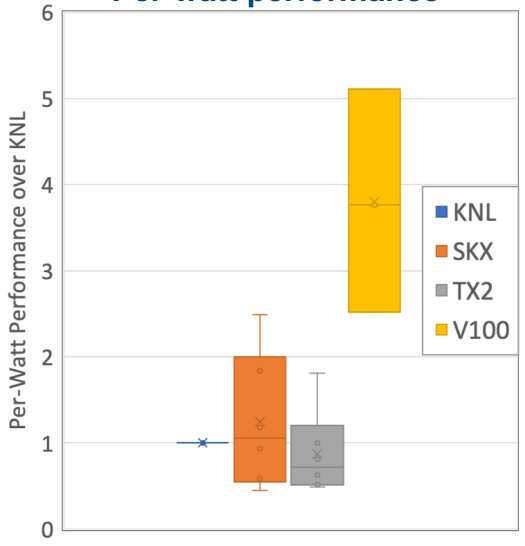
CONCLUDING REMARKS

- Executed performance tests
  - for 2 HPC benchmarks (i.e., HPGMG-FV, and NEKBONE) and 4 HPC applications (i.e., GAMESS, LAMMPS, QMCPACK, and Qbox)
  - on four types of processor architectures (i.e., KNL, SKX, TX2 and V100)

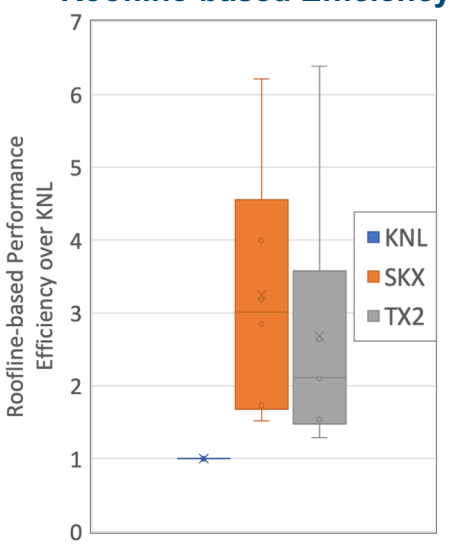
Per-node performance



Per-watt performance



Roofline-based Efficiency



CONCLUDING REMARKS

- Core Affinity issues on TX2
  - “-bind-to socket” should be used with MPI. Otherwise, OpenMP threads are spread out to multiple sockets, or MPI processes are not equally distributed to multiple sockets.

ACKNOWLEDGEMENT

- This Work was supported by the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.
- We also gratefully acknowledge the computing resources provided and operated by the Joint Laboratory for System Evaluation (JLSE) at Argonne National Laboratory.

REFERENCES

1. “Empirical Roofline Tool Web page,” https://crd.lbl.gov/departments/computerscience/PAR/research/roofline/software/ert/, 2019.
2. “HPGMG Web page,” https://hpgmg.org, 2019.
3. S. Williams, “4th order hpgmg-fv implementation,” HPGMG BoF, Supercomputing, 2015.
4. M. Adams, J. Brown, J. Shalf, B. Straalen, E. Strohmaier, and S. Williams, “Hpgmg 1.0: A benchmark for ranking high performance computing systems,” LBNL Technical Report, LBNL 6630E, 2014.
5. J. Kwack and G. H. Bauer, “HPCG and HPGMG benchmark tests on multiple program, multiple data (MPMD) mode on Blue Watersa Cray XE6/XK7 hybrid system,” Concurrency Computat: Pract Exper., 2017.
6. “HPGMG Github,” https://github.com/hpgmg/hpgmg, 2019.
7. “HPGMG-CUDA Bitbucket,” https://bitbucket.org/nsakharnykh/ hpgmg- cuda.git, 2019.
8. “Nekbone repository,” https://asc.llnl.gov/CORAL-benchmarks/.
9. P. Fischer, J. Lottes, D. Pointer, and A. Siegel, “Petascale algorithms for reactor hydrodynamics,” Journal of Physics: Conference Series, 2008.

REFERENCES

10. M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis, and J. A. Montgomery, “General atomic and molecular electronic structure system,” Journal of Computational Chemistry, vol. 14, no. 11, pp. 1347–1363, 1993.
11. M. S. Gordon and M. W. Schmidt, “Chapter 41 - advances in electronic structure theory: Gamess a decade later,” in Theory and Applications of Computational Chemistry, C. E. Dykstra, G. Frenking, K. S. Kim, and G. E. Scuseria, Eds. Amsterdam: Elsevier, 2005, pp. 1167 – 1189.
12. Bode, B. M. and Gordon, M. S. J. Mol. Graphics Mod., 16, 1998, 133-138.
13. S. Plimpton, “Fast parallel algorithms for short-range molecular dynamics,” Journal of Computational Physics, vol. 117, pp. 1– 19, 1995.
14. “LAMMPS Web page,” https://lammps.sandia.gov, 1995.
15. J. Kim, et al, “QMCPACK: an open source ab initio quantum monte carlo package for the electronic structure of atoms, molecules and solids,” Journal of Physics: Condensed Matter, vol. 30, no. 19, p. 195901, apr 2018.

REFERENCES

16. S. Williams, A. Waterman, and A. Patterson, “Roofline: an insightful visual performance model for floating-point programs and multicore architectures,” Commun ACM., vol. 53, pp. 65–76, 2009.
17. A. Ilic, F. Pratas, and L. Sousa, “Cache-aware roofline model: upgrading the loft,” IEEE Comput Archit Lett., vol. 13, pp. 21–24, 2014.
18. J. Kwack, G. Arnold, C. Mendes, and G. H. Bauer, “Roofline analysis with Cray performance analysis tools (CrayPat) and roofline- based performance projections for a future architecture,” Concurrency Computat Pract Exper., 2018.
19. “General Roofline Evaluation Gadget Webpage,” https://github.com/ ncsa/GREG, 2019.

THANK YOU!