Making of High-Performance Parallel Applications

Jeongnim Kim, PhD
HPC **E**cosystem **A**pplication Engineering **T**eam/CMO/E&G/DCG, Intel
ADAC Institute Workshop, July 2017

Legal Disclaimers

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.
Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Knights Landing and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user

Intel, Look Inside, Xeon, Intel Xeon Phi, Pentium, Cilk, VTune and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Legal Disclaimers

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel.

Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

Legal Disclaimers

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to http://www.intel.com/performance.

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more athttp://www.intel.com/go/turbo.

Estimated Results Benchmark Disclaimer:
Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

Software Source Code Disclaimer:
Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Legal Disclaimers

The above statements and any others in this document that refer to plans and expectations for the third quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as “anticipates,” “expects,” “intends,” “plans,” “believes,” “seeks,” “estimates,” “may,” “will,” “should” and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel’s actual results, and variances from Intel’s current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the company’s expectations. Demand could be different from Intel's expectations due to factors including changes in business and economic conditions; customer acceptance of Intel’s and competitors’ products; supply constraints and other disruptions affecting customers; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Uncertainty in global economic and financial conditions poses a risk that consumers and businesses may defer purchases in response to negative financial events, which could negatively affect product demand and other related matters. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Revenue and the gross margin percentage are affected by the timing of Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel’s response to such actions; and Intel’s ability to respond quickly to technological developments and to incorporate new features into its products. The gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; start-up costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; product manufacturing quality/yields; and impairments of long-lived assets, including manufacturing, assembly/test and intangible assets. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. Intel’s results could be affected by the timing of closing of acquisitions and divestitures. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel’s ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. A detailed discussion of these and other factors that could affect Intel’s results is included in Intel’s SEC filings, including the company’s most recent reports on Form 10-Q, Form 10-K and earnings release.

Rev. 7/17/13

Systems and toolchains

- KNC: Intel® Xeon Phi™ coprocessor 7120P
  - 61 cores @ 1.238 GHz, 4-way Intel® Hyper-Threading Technology, Memory: 15872 MB
  - Intel® Many-core Platform Software Stack Version 3.6.1
  - OS Version : 3.10.0-229.el7.x86_64
- Intel® Xeon Phi™ 7250P (code-named Knights Landing, KNL), 68 cores, 1.4GHz with 16GB MCDRAM (used in flat mode), cluster boot mode=Quad, Turbo=enable.
- Intel® Xeon® E5-2697v4(BDW) node single socket, 18 cores HT Enabled @2.3GHz 145W (E5-2697v4 w/128GB RAM DDR4 2400 8*16GB DIMMS
- Compilers and MPI and math library on Intel platforms
  - icc version 16.0 (gcc version 4.8 compatibility)
  - Intel® MPI Library for Linux* OS, Version 5.1.3 Build 20160120 (build id: 14053)
- BGClang on IBM Blue Gene/Q
  - H. Finkel, bgclang: creating an alternative, customizable toolchain for the Blue Gene/Q, IEEE/ACM International conference for high performance computing, networking, storage and analysis, 2014.

Acknowledgements

- Many thanks to MKL, OpenMP and MPI developers at Intel
- Intel® Parallel Compute Centers at ANL/Sandia and ZIB
- QMCPACK collaboration http://www.qmcpack.org
  - Especially, **Ye Luo** and Anouar Benali (ANL), Luke Shulenburger and Raymond Clay (Sandia), and **Amrita Mathuriya** (HEAT, Intel)
  - DOE BES and ASCAR funding
  - Resources at DOE leadership computing centers
- VASP collaboration: **Martijn Marsman** (Univ. Vienna), **Florian Wende** (ZIB), Zhengji Zhou and Stephen Leak (NERSC), Fedor Vasliev and Nadezhda Plotnikova (INNL, Intel)

Contact: jeongnim.kim@intel.com

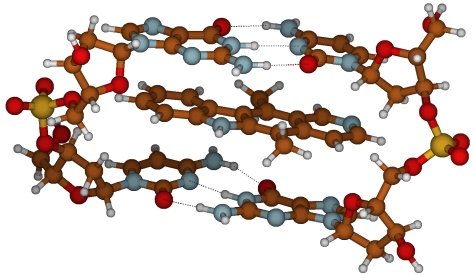
Outline

- Introduction
  - QMCPACK and parallel computers
  - Overview of Intel® Xeon Phi™ Processor Family
- Transforming QMCPACK, a parallel application
  - Data layout transformation, a portable way on cache-based shared memory systems
- Conclusions

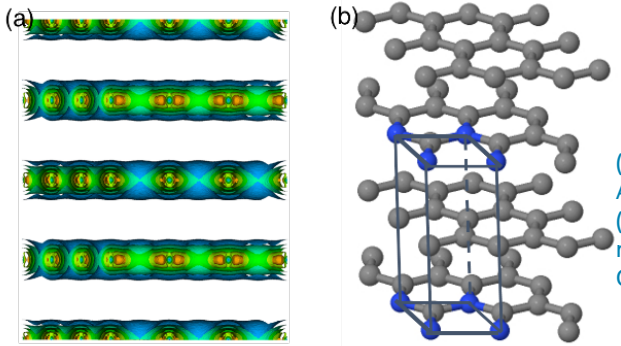
QMCPACK http://www.qmcpack.org/

An open-source US-DOE flagship many-body *ab initio* quantum Monte Carlo (QMC) code for computing the electronic structure of atoms, molecules, and solids.

- C++, MPI+X (OpenMP or CUDA)
- UIUC/NCSA Open-source
- Research code developed and maintained by QMC experts
- Used to explore new programming models and architectures
- Supported by DOE
  - Exascale Application Project (ECP)
  - CORAL benchmark application
  - QMC endstation (completed)
  - > 300 M allocations by 3 projects per year



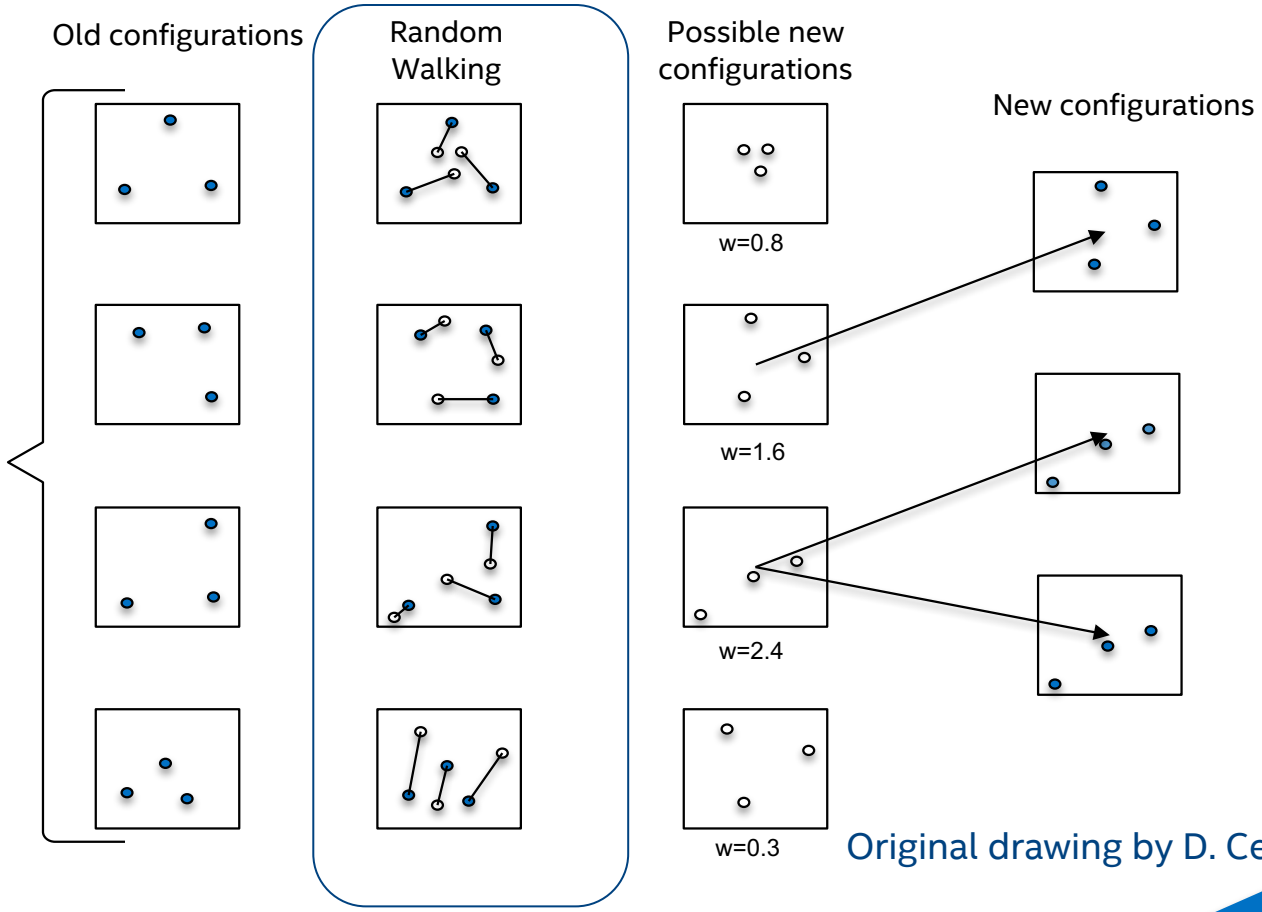
An anti-cancer drug (ellipticine) intercalated in between two DNA basepairs, Anouar *et al.*



(a) DMC charge-density of AB stacked graphite and (b) the ball-and-stick rendering and the 4-Carbon unit cell in blue.

Diffusion Monte Carlo Schematics

Important sampling guided by a trial wavefunction



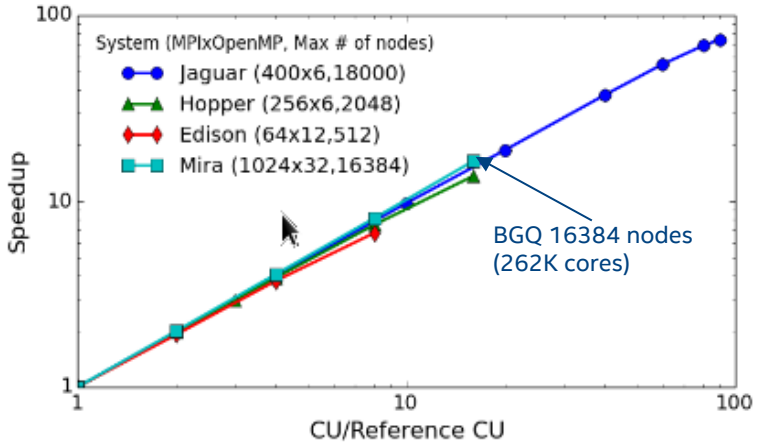
Ensemble evolves according to

- Diffusion
- Drift
- Branching

Original drawing by D. Ceperley

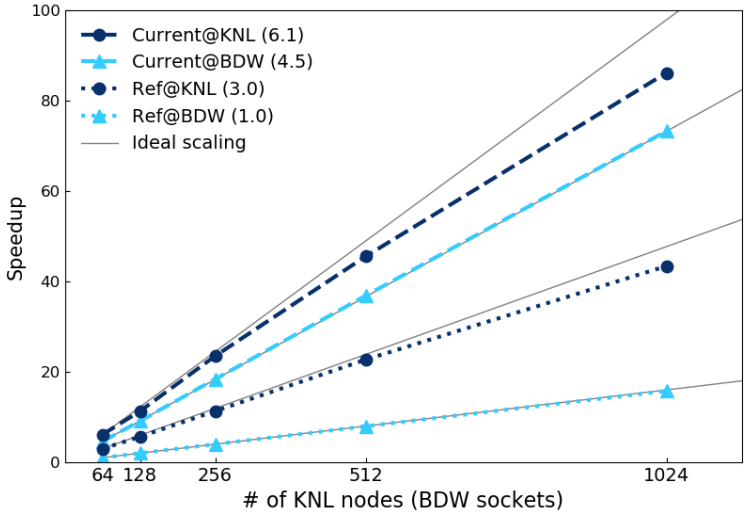
QMCPACK evolution since 2005

Parallel efficiency on DOE leadership computing facilities
Various Carbon systems, N~200



Parallel efficiency of QMCPACK on US-DOE facilities. The legend shows the MPI tasks and OpenMP threads of the reference computing unit (CU) and the maximum number of nodes on each platform.

Parallel efficiency on KNL and BDW clusters
NiO: 64-atom cell, N=768 electrons



On Trinity at LANL (KNL) and Serrano at SNL (BDW) systems. The performance is normalized by a reference throughput using 64 BDW sockets. A. M. et al, SC17

Evolution in computation, memory and communication

	Cray XT 5 (Jaguar@ORNL)	IBM BG/Q (Mira@ANL)	Penguin OPA (Serrano@Sandia)	Cray XC40 (Cori@NERSC)
Processor Clock	2.3 GHz	1.6 GHz	2.1 GHz	1.4 GHz
Node	8 cores (2 sockets)	16 cores	36 cores (2 sockets)	68 cores
Peak GF/s	73.6 /node 18.4/core	204.8 /node 12.8/core	1209.6/node 33.6/core	3000/node 44 /core
SIMD length (D/S)	128 Bytes (2/4)	256 Bytes (4/8)	256 Bytes (4/8)	512 Bytes (8/16)
Peak Memory BW	25.6 GB/s	42.6 GB/s/node	126 GB/s/node	460 GB/s/node*
Memory (DDR)	16 GB/node 2 GB/core	16 GB/node 1 GB/core	128 GB/node 3.6 GB/core	96 GB/node 1.4 GB/core
On-package HBM				16 GB/node 235 MB/node
Cache	L1 (P, 64 kB) L2 (P, 512 kB) L3 (S, 2 MB)	L1 (P, 16kB/16kB) L2 (S, 32 MB)	L1 (P, 32 kB) L2 (P, 256 kB) L3 (S, 45 MB)	L1(P, 32 kB) L2(P/tile, 1 MB)
Node-to-node BW	9.6 GB/node	4 GB/s/node	20 GB/s/node	11 GB/s/node
	2009	2012	2016	2017

https://cug.org, 2009
https://www.alcf.anl.gov/files/IBM_BGQ_Architecture_0.pdf

* MCDRAM on KNL

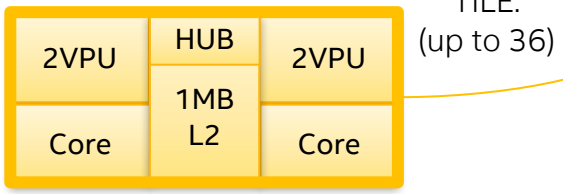
Intel® Xeon Phi™ Processor Family Architecture Overview

ISA
Intel® Xeon® Processor Binary-Compatible (w/Broadwell)

On-package memory (MCDRAM)
Up to 16GB, ~500 GB/s STREAM memory bandwidth

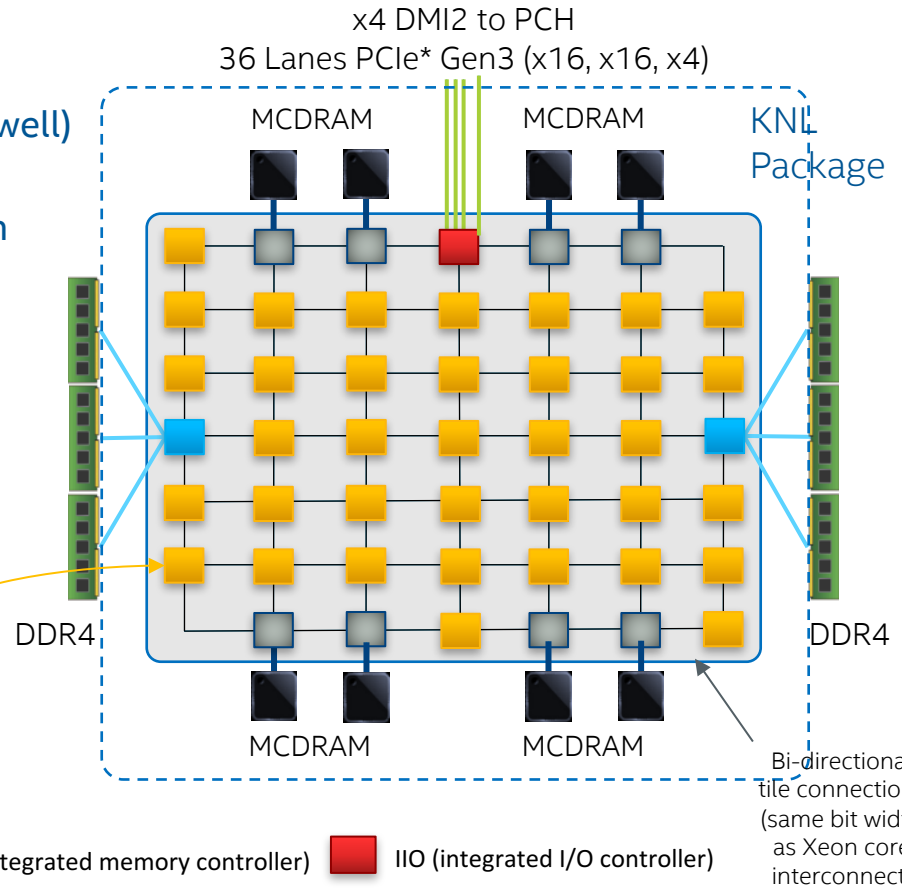
Platform Memory (DDR)
Up to 384GB (6ch DDR4-2400 MHz)

Notable features
- 2D Mesh Architecture
- Out-of-Order Cores



TILE: (up to 36)

Enhanced Intel® Atom™ cores based on Silvermont Microarchitecture



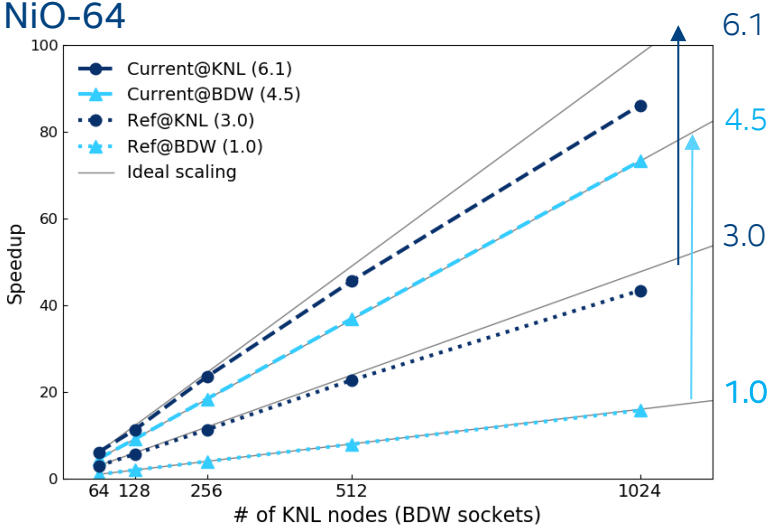
What is KNL for QMCPACK community?

- **Discovery of new materials through faster, bigger and better quantum simulations!**
- Many cores, many threads, SIMD, high-band memory, large capacity memory, multi-level cache: just another CPU, building blocks of parallel computers
- MPI over distributed shared-memory domains
- Threading within a shared-memory domain
- # of MPI per node must be controlled on KNL
  - MPI costs resources – memory, bandwidth, computation, communication
  - Why distribute data when one can share?
  - Why communicate when one can compute?
- **Amdahl's law dictates: cannot accelerate parts; everything needs to be parallel**
- Have to use KNL resources at ANL, NERSC, Sandia/LANL, CINECA ….

The same for any high-performance parallel application and its community!

QMCPACK “revolution” in 2017

NiO-64



Node Performance Increase

	Graphite	Be-64	NiO-32	NiO-64
BG/Q*	1.6	1.3	1.3	2.4
BDW	2.9	3.4	2.6	5.2
KNL	2.2	2.9	2.4	2.4

Constraints

- Portable and maintainable and expandable by QMCPACK developers (physicists and chemists)
- Sustain excellent parallel programming model based on MPI and OpenMP standards
- Small memory footprint per node
- Leverage ecosystem: compilers, standards, libraries

Approaches

- Create miniapps representing QMC computation and data access; continuous integration of new data structures and algorithms with validation
- **Introduce data layout transformation from Array-of-Struct (AoS) to Struct-of-Arrays (SoA) of key physics abstractions for QMC algorithms**
- Improve algorithms for modern platforms: compute-on-the-fly and reduced memory operations
- Expand single-precision use while preserving fidelity of calculations

* Baseline on BG/Q used QPX intrinsics

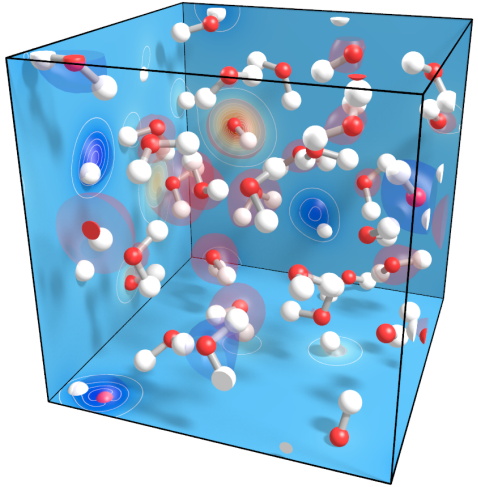
QMC: Single particle orbital (SPO) representation with B-spline basis set

One Dimensional cubic B-spline function

$$f(x) = \sum_{i'=i-1}^{i+2} b^{i',3}(x)\; p_{i'},$$

Tensor product in each Cartesian direction, Representation for 3D orbital,

$$\phi_n(x,y,z) = \sum_{i'=i-1}^{i+2} b_x^{i',3}(x) \sum_{j'=j-1}^{j+2} b_y^{j',3}(y) \sum_{k'=k-1}^{k+2} b_z^{k',3}(z)\; p_{i',j',k}$$

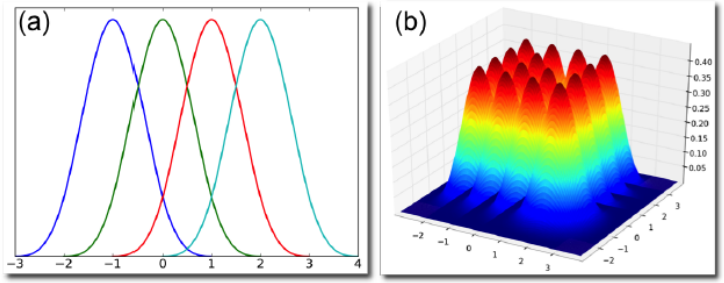


Isosurface of $|\phi_n|^2$ in H_2O

Determinant of A(N,N)

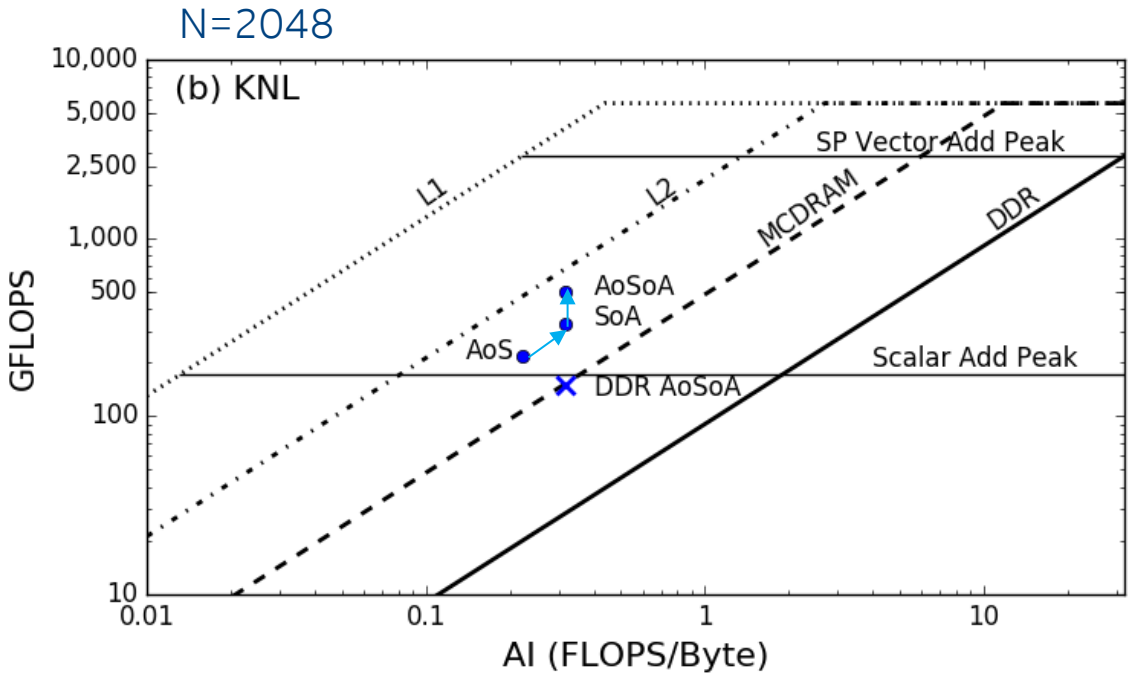
$$\begin{vmatrix} \phi_1(\mathbf{r}_1) & \phi_2(\mathbf{r}_1) & \cdots & \phi_N(\mathbf{r}_1) \\ \phi_1(\mathbf{r}_2) & \phi_2(\mathbf{r}_2) & \cdots & \phi_N(\mathbf{r}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{r}_N) & \phi_2(\mathbf{r}_N) & \cdots & \phi_N(\mathbf{r}_N) \end{vmatrix}$$

basis



- Precomputed coefficients
- 4D read-only array
- Stored in SoA format, P[nx][ny][nz][N]
- Provided by DFT or HF computations using Quantum Espresso

Computation of values, gradients and Hessians (Bspline-vgh) on KNL



Mathuriya et al, IPDPS 2017

R = positions of N particles

AoS R[N][3]

x	y	z	x	y	z
x	y	z	x	y	z
x	y	z	x	y	z

SoA R[3][N]

x	x	x	x	x	x
y	y	y	y	y	y
z	z	z	z	z	z

AoSoA

x	x	…	y	y	…	z	z	…
x	x	…	y	y	…	z	z	…
x	x	…	y	y	…	z	z	…

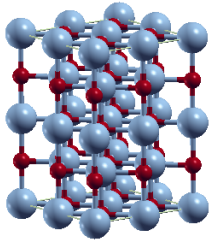
Data layout transformation: AoS-to-SoA

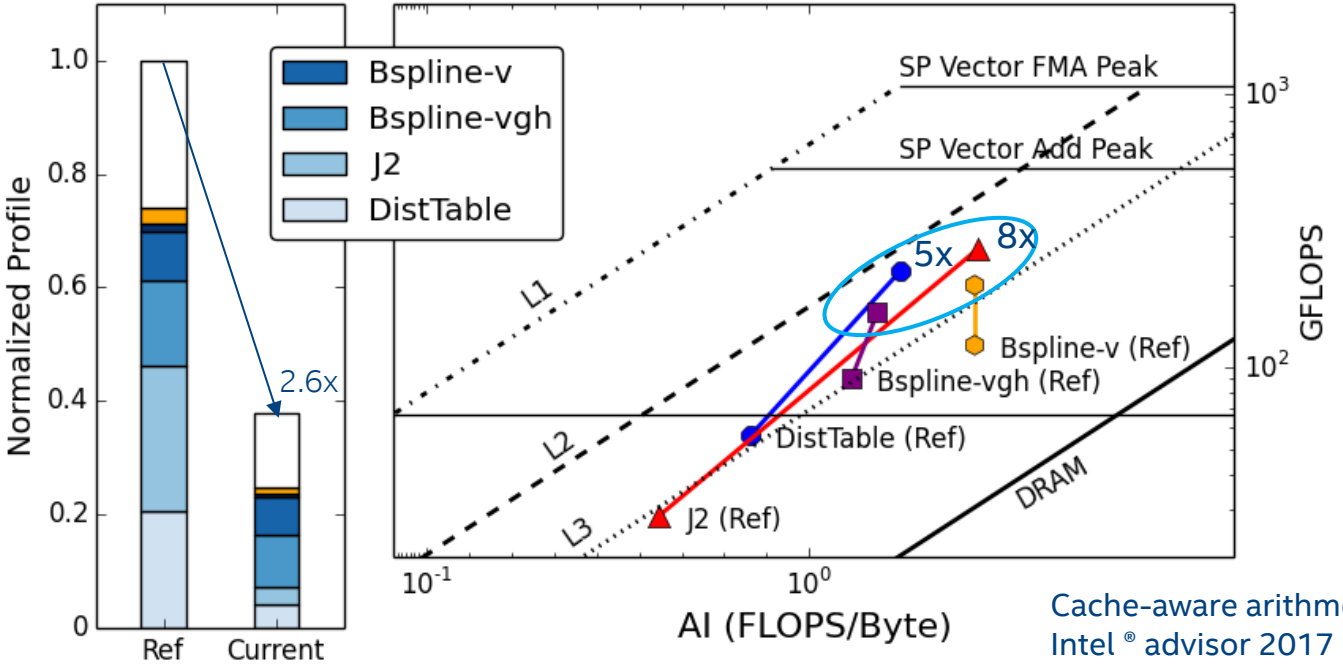
- Need disruptive and structural changes, affecting many code paths including computations of
  - Distance relationship among electrons and ions
  - Correlation functions (aka Jastrow) using distance tables (distances, displacements)
  - Interaction energies, Coulomb potentials and model potentials
- Naively, one has to rewrite QMCPACK!
- No need to panic
  - QMC algorithms for big problems (N>10) move one electron at a time (aka Particle-by-particle update) : only the inner-most loops matter
  - Need AoS for cache locality and physics abstractions: MPI communications, I/O, statistics
  - Core compute kernels are few and simple
  - Use miniapps to debug, test, performance analysis, and algorithm development
- **Solution: use intermediate helper classes and methods to facilitate SIMD-friendly loops; let the compilers optimize; help compilers with alignment and OMP SIMD**
- SoA-to-AoSoA transformations use SoA objects

Performance profiles before and after

Mathuriya et al, SC17

NiO-32 on a 1S E5-2698 v4 (BDW, 20 core)

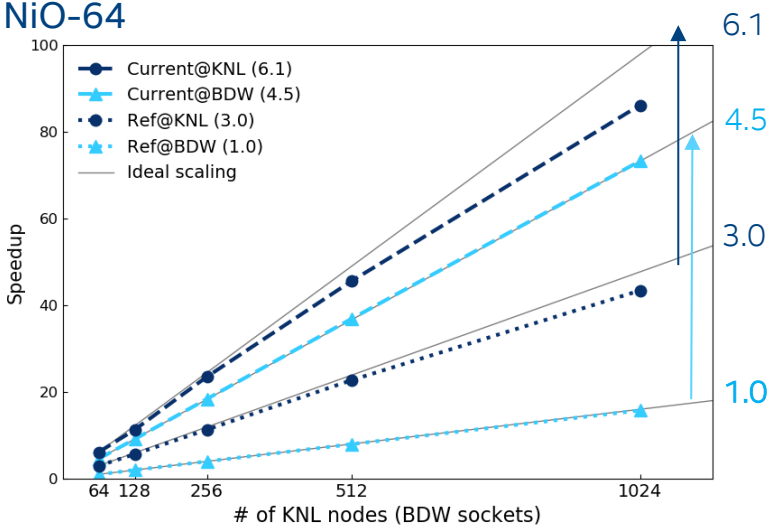


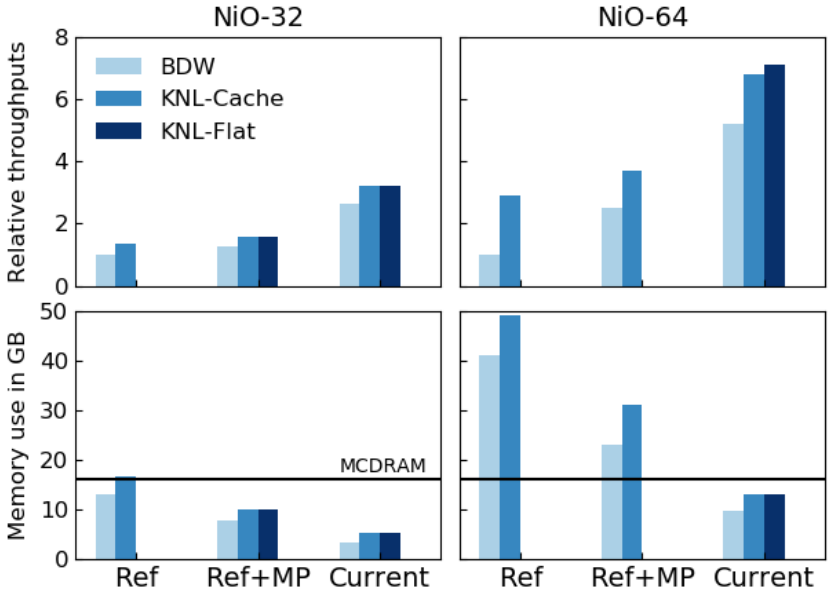


Cache-aware arithmetic intensity (AI), Intel ® advisor 2017

QMCPACK Performance in 2017

NiO-64





Node Performance Increase

	Graphite	Be-64	NiO-32	NiO-64
BG/Q*	1.6	1.3	1.3	2.4
BDW	2.9	3.4	2.6	5.2
KNL	2.2	2.9	2.4	2.4

* Baseline on BG/Q used QPX intrinsics

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations.

Increasing parallelism: OO way

Same classes and methods

Assume OpenMP
Any *Nw* walkers
N ~ *L* > 1000

```
1  #pragma omp parallel
2  {
3  int var=0;
4  int ip=tid();
5  int kI=L/crews*ip;
6  int kF=L/crews*(ip+1);
7
8  for(i=0;i<Nw;i++){
9   for(j=0;j<N;j++){
10
11   //explicit
12   for(k=kI; k<kF;k++)
13   {
14     ...
15   }
16
17
18   var++;
19  }
20
21
22 }}
23   (a) baseline
```

```
#pragma omp parallel
{
int var=0;

#pragma omp parallel
{
for(i=0;i<Nw;i++){
 for(j=0;j<N;j++){

#pragma omp for nowait
  for(k=0; k<L;k++)
  {
    ...
  }

#pragma omp master
  var++;
 }
//N plain barriers
#pragma omp barrier
}}}
    (b) nested
```

```
#pragma omp parallel
{
int var=0;

for(i=0;i<Nw;i++){
 for(j=0;j<N;j++){

#pragma omp parallel for
  for(k=0; k<L;k++)
  {
    ...
  }

//NxM fork-join constructs
  var++;
 }
}}
   (c) parallel-for
```

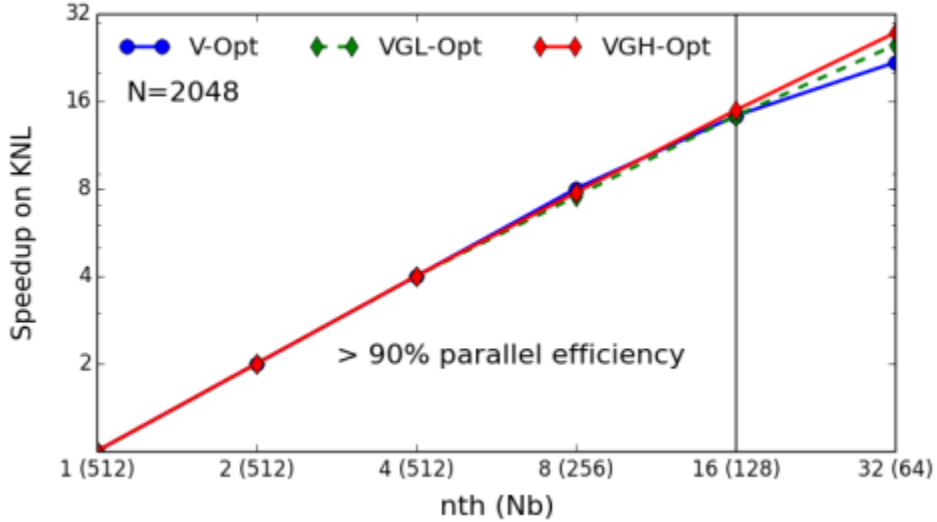
Serial

Parallel

crews=# of cooperative threads

Doodi *et al.*, IWOMP2017

Strong Scaling of Bspline routines on KNL



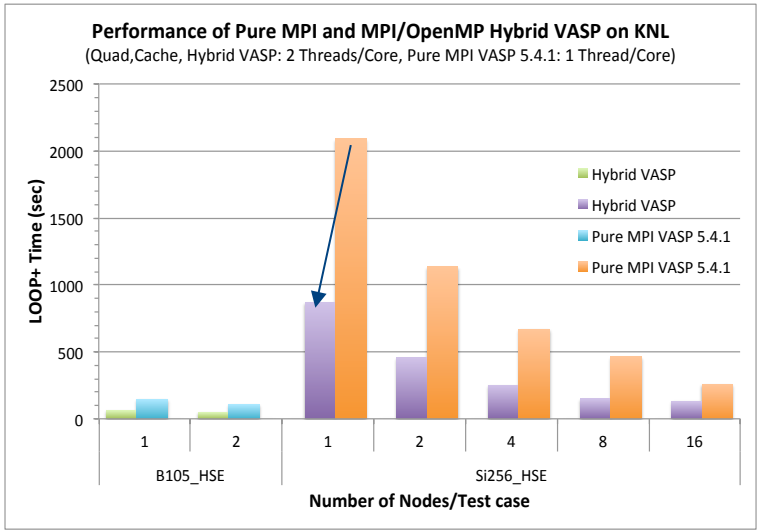
Speedup on KNL w.r.t. number of walkers per thread.

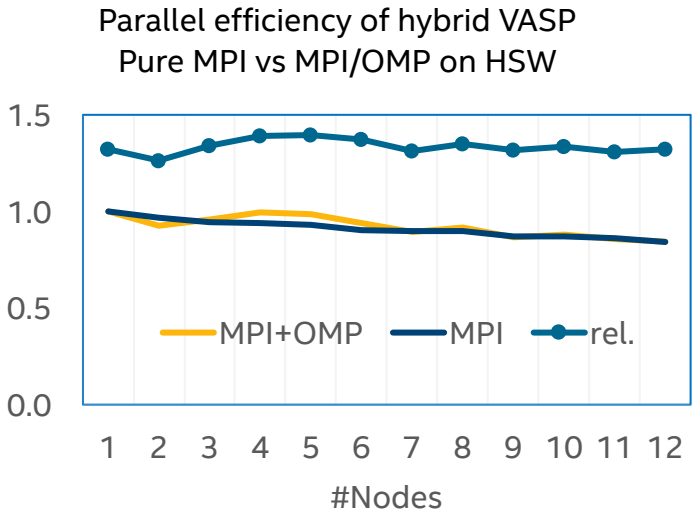
Reduces time to solution by ~14x with 16 crews per walker

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations.

MPI/OpenMP implementation in VASP

Marsman (Univ. Vienna) and Wende (ZIB IPCC)





- More than 2x improvement from 5.4.1
- Using the same compilers and MKL
- Includes 2x for both with MKL optimization

- 30% speedup with hybrid on HSW
- More than 2x speed up on KNL
- Hybrid allows use of KNL flat/quad mode

Zhou (NERSC) et al., NERSC Cori, Intel® Xeon Phi™ processor 7250
VASP 5.4.1, current release
Hybrid VASP: beta testing in progress at NERSC et al

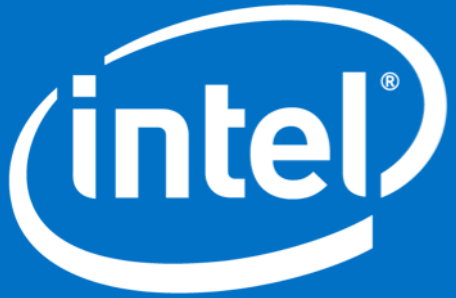
Marsman, HPC DevCon 2016
Intel® Xeon E5-2660v3@2.60 GHz, dual sockets, 10 cores/socket
Local InfiniBand cluster at Univ. Vienna

Conclusions

- Making of high-performance parallel applications
  - Each application is different and knowing the application is the first step
  - Consider algorithms, data structures, programming languages, parallel programming models, libraries
- KNL is an instance of cache-based, many-core, SIMD processors
  - Exercise parallel and good programming 101
- Set the goals and priorities and execute them
- Code is a living organ: evolutionary and revolutionary transformations of the data structures and algorithms
- How about performance portability?

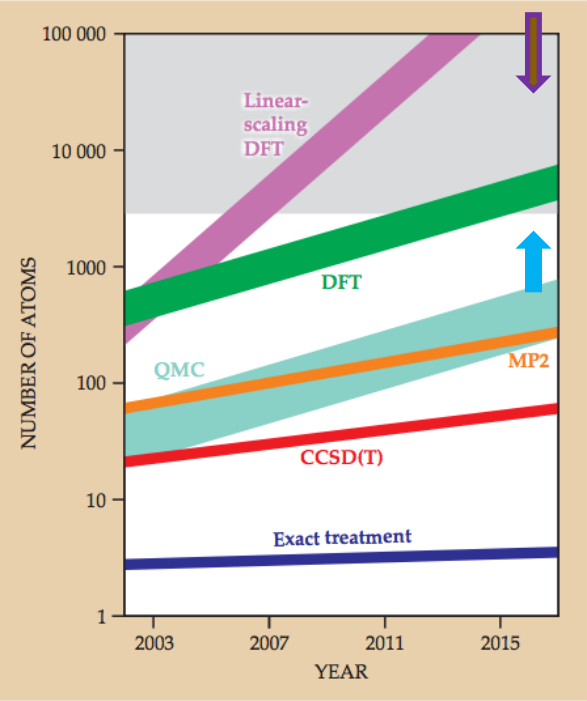
Performance portable practices

- Parallelize at all levels: nodes, sockets, cores and SIMD; MPI, OpenMP, threads
- Analyze performance of representative workloads and build performance model
- Improve data movement and BW utilization: AoS vs SoA vs AoSoA
- Select optimal data structures for the algorithms and problems to solve
  - One-size-fits-all, not practical nor productive
- Increase compute density/bytes
  - Do not store, read or write unless you must
- Leverage generic and object-oriented programming
  - Types, operators, containers, algorithms, concepts … all good things about C++
  - Specialization on target platforms, if profitable
- Improve algorithms
- Expect changes and prepare for disruptive technology

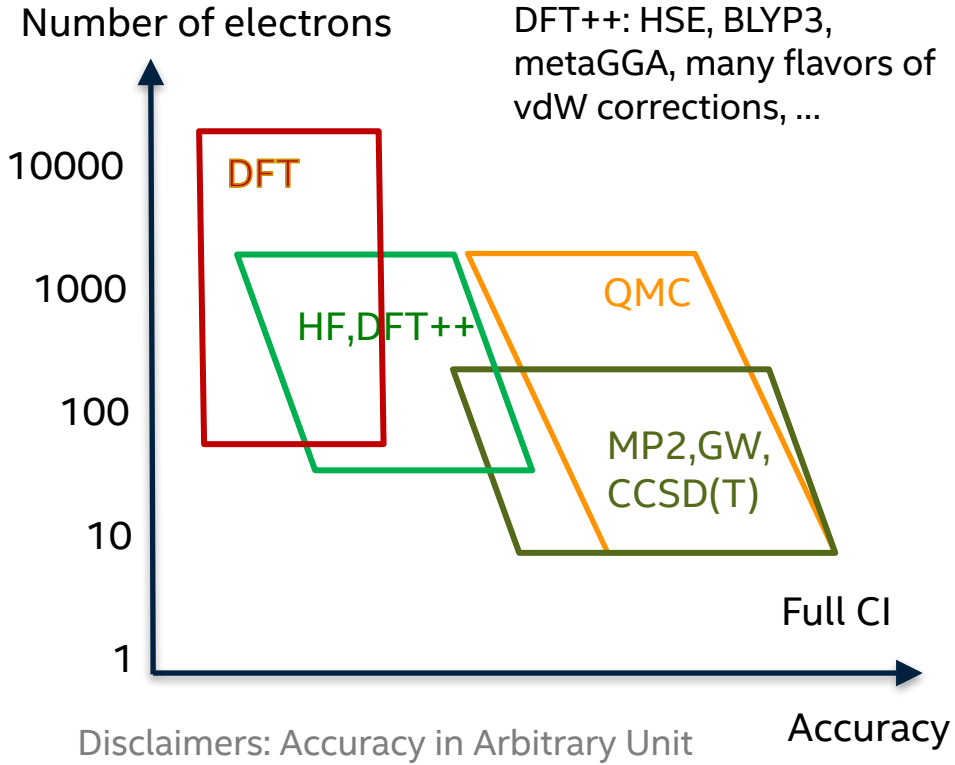

intel

QMMM on Parallel Computers

Predictions in 2008



Head-Gordon & Atacho, Physics Today (2008), http://dx.doi.org/10.1063/1.2911179

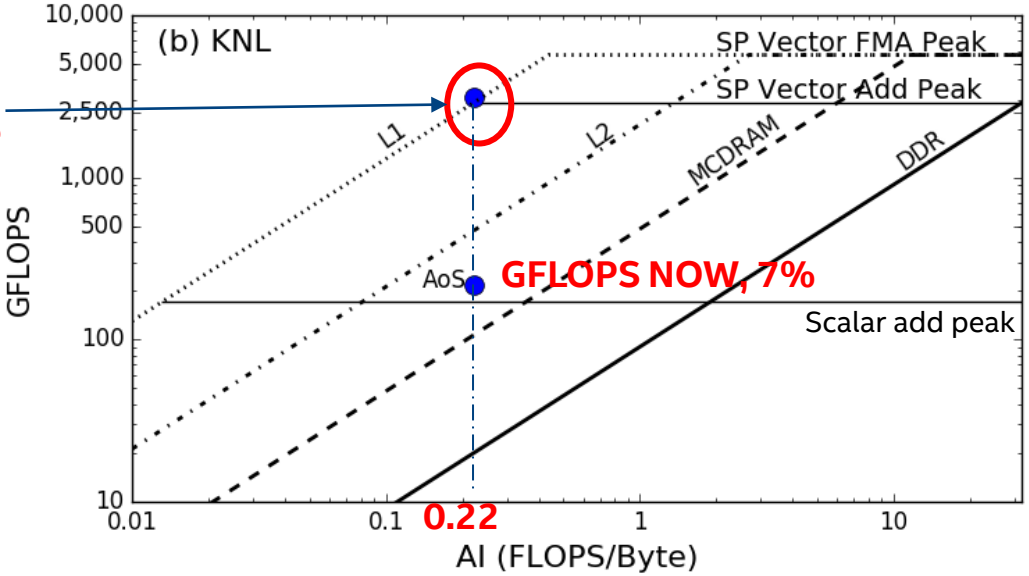


DFT++: HSE, BLYP3, metaGGA, many flavors of vdW corrections, ...

Disclaimers: Accuracy in Arbitrary Unit

Roofline Performance Analysis of bspline on KNL

Peak GFLOPS at (0.22 Flops/Byte



VGH roofline performance model for N=2048. Circles denote GFLOPS at the cache-aware AI and X (b) the best performance (AoSoA) on DDR.

7% of peak GFLOPS on KNL achieved with the current AoS version

Plans and executions of QMCPACK

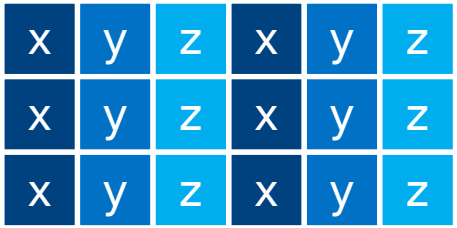
- Keep the current parallelization scheme and focus on improving efficiency on a node
- Reduce memory use by expanding single-precision use
  - Careful accuracy control for all problem sizes and quality of the trial wavefunction
- Increase SIMD efficiency with new data structures
  - Use CacheAlignedAllocator : TBB, boost, c++11
  - AoS-to-SoA transformations for particle attributes
  - Expose SIMD-friendly loops**: pragma omp simd**
- Change algorithms: compute over store
  - Improve BW and cache utilization
  - Reduce memory use and operations
- Develop miniapps to test, build and debug and integrate to the full app

Done

- Tackle fine-grained parallelism per walker and explore threading methods
- SoA-to-AoSoA for cache optimization and parallelization

Data Layout – Performance Considerations

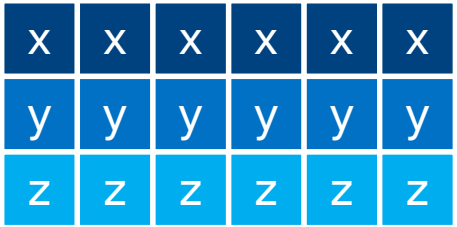
Array-of-Structs (AoS)



- Pros:
  Logical for expression of physical abstractions in 3D or higher dimensions.

Original QMCPACK

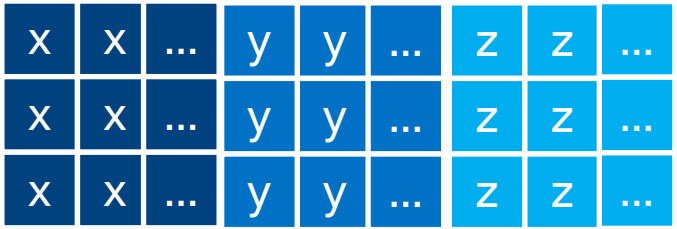
Struct-of-Arrays (SoA)



- Pros:
  Contiguous loads/stores for efficient vectorization.

Optimized QMCPACK

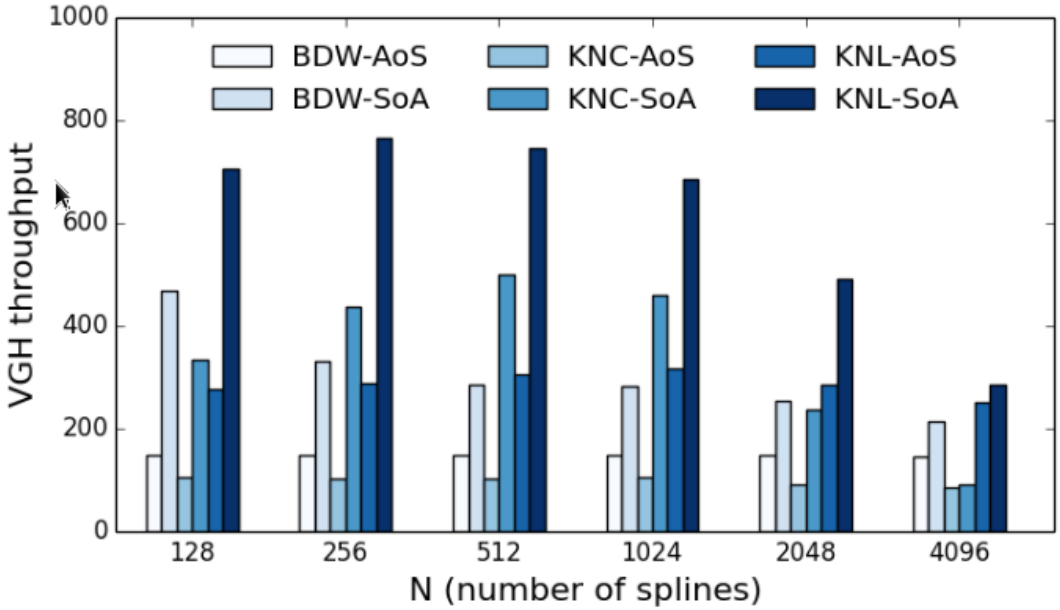
Hybrid (AoSoA)



- Pros:
  Potentially useful for increasing cache locality. Also supports efficient vectorization.

Optimized QMCPACK for future problems

Bspline-ValueGradientHessian throughput by AoS-to-SoA transformation

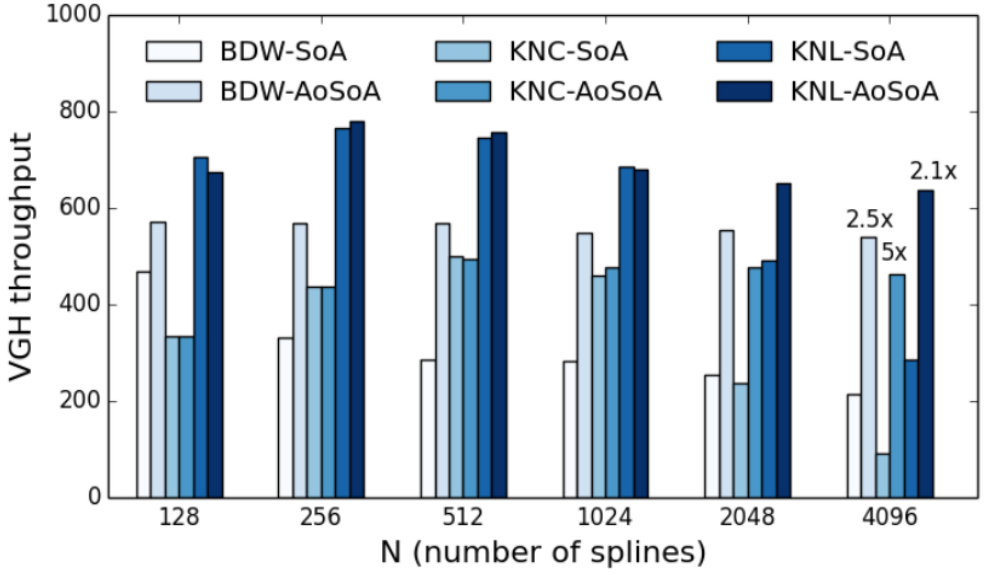


Higher the better

2x-4x Performance improvement for small to medium problem sizes.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations.

Performance gain with tiling/AoSoA - Higher the better



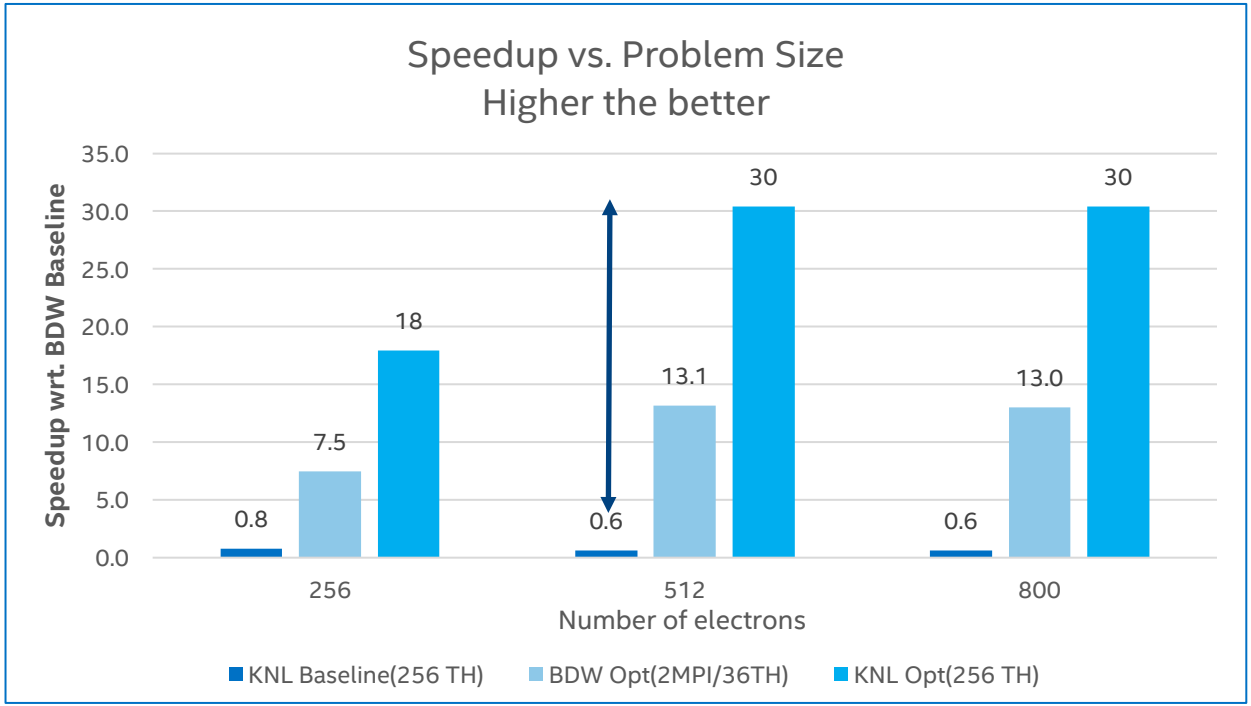
VGH Performance with SoA to AoSoA transformation (tiling)

AoSoA helps achieve sustained throughput across problem sizes for all architectures.

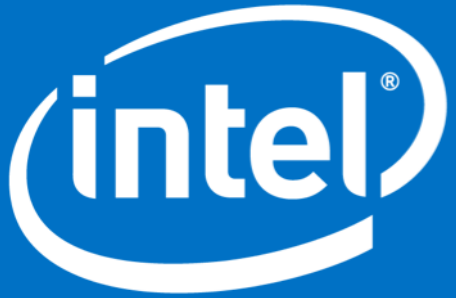
Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

Electron-electron Distance table computation miniapps

AoS to SoA transformation: R[N][3] -> R[3][N]



- KNL used in Quad/Cache mode for these experiments
- Here, TH = threads.
- BDW has 2 sockets for these experiments.


intel