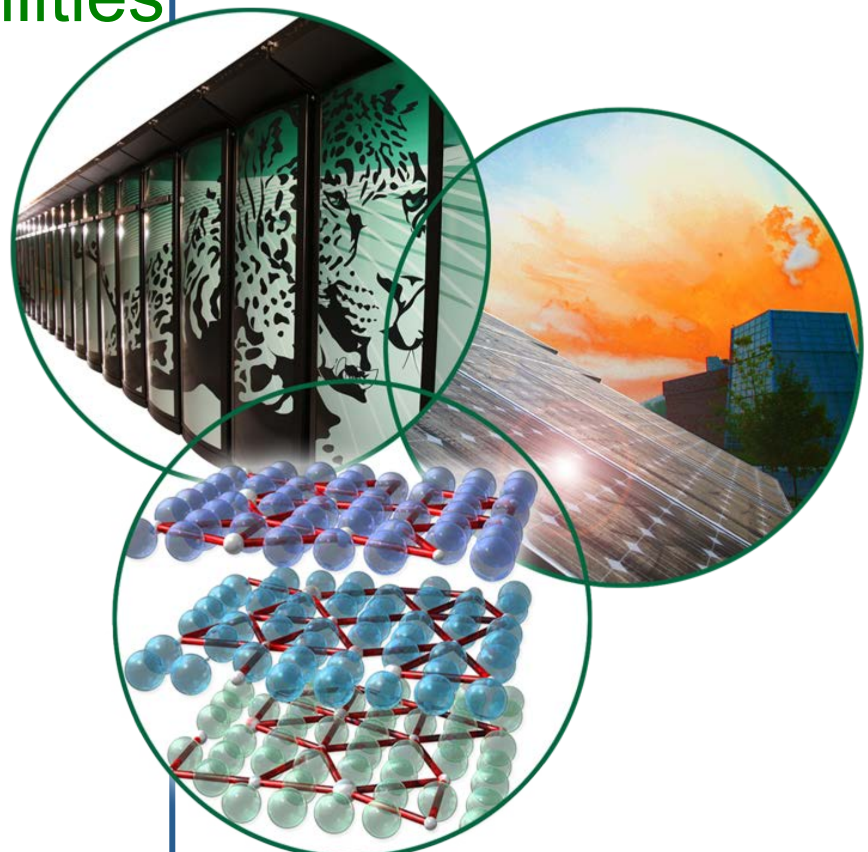
QMCPACK: Enabling Breakthrough QMC Simulations at Leadership Computing Facilities



Presented at

Accelerating Computational Science Symposium 2012

Jeongnim Kim

R&D Staff

Materials Science and Technology Division

Computational Chemistry and Materials Division

Oak Ridge National Laboratory

U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Acknowledgements

QMCPACK developers

- **Kenneth P. Esler (Stoneridge)**
- Jeremy McMinis (UI)
- Miguel Morales (LLNL)
- Bryan Clark (Princeton)
- Luke Shulenburger (Sandia)
- Jaron Krogel (UI)
- Simone Chiesa (W&M)
- Kris Delaney (UCSB)
- others

http://qmcpack.cmscc.org

QMC Endstation

- David M Ceperley (UI)
- S. Zhang & H. Krakauer (W&M)
- P. Kent (ORNL)
- L. Mitas (NCSU)
- Umrigar & Hennig (Cornell)
- A. Srinivasan (FSU)

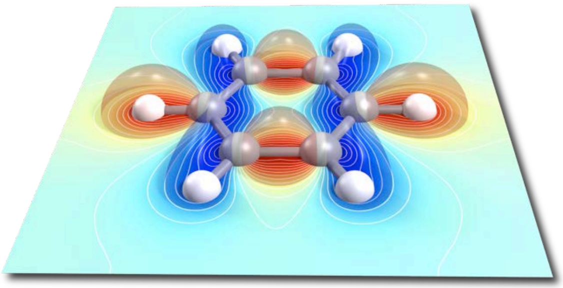
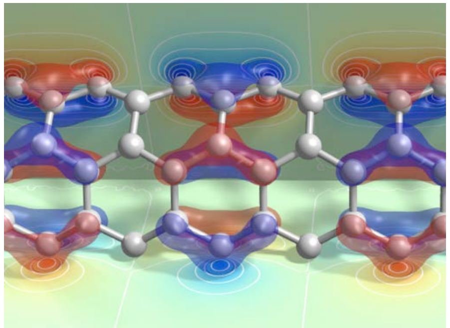
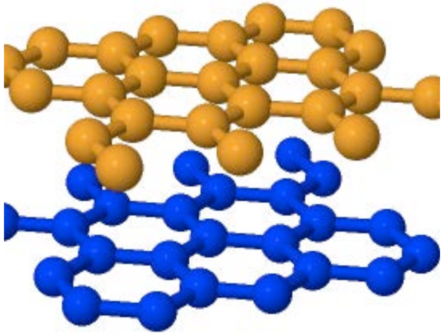
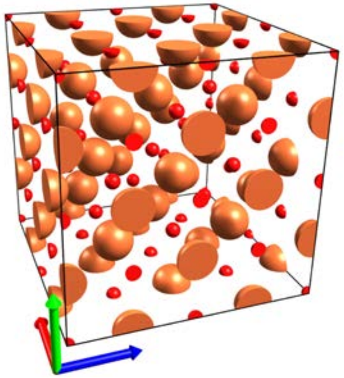
Special thanks to

- T. C. Schulthess (CSCS,ORNL)
- Richard M. Martin (UI)
- John W. Wilkins (OSU)

QMC advantages: accuracy and scalability

- Applicable to a wide range of problems
  - Any boundary conditions: molecular and solid-state systems
  - Dimensionality: 1D, 2D, and 3D
  - Representation: atomistic to model Hamiltonians
- Scale with a few powers in system size: *$O(N^3)$-$O(N^4)$*
  - Routine calculations of 100s-1000s electrons
- Ample opportunities of parallelism

QMC has enabled **accurate, many-body** predictions of electronic structures of atoms, molecules to solids; molecular solids to highly correlated metals

Basics of QMC for ES

For *N*-electron system $\{\mathbf{R}\} = (\mathbf{r}_1, \cdots, \mathbf{r}_N)$

Many-body Hamiltonian $\hat{H} = \sum_i \frac{1}{2m_e}\nabla^2 + \frac{1}{2}\sum_{i \neq j} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_i V_{ext}(\mathbf{r}_i)$

Many-body trial wavefunction $\Psi_T(\mathbf{R})$

$$E_T = \frac{\int d^{3N}\mathbf{R}\ \Psi_T^*(\mathbf{R})\hat{H}\Psi_T(\mathbf{R})}{\int d^{3N}\mathbf{R}\ |\Psi_T(\mathbf{R})|^2}, \quad E_T \geq E_0$$

QMC

$$\boxed{< E_T >= \frac{\sum_i^M w(\mathbf{R}_i)E_L(\mathbf{R}_i)}{\sum_i^M w(\mathbf{R}_i)}, \quad E_L = \frac{\hat{H}\Psi_T(\mathbf{R})}{\Psi_T(\mathbf{R})}}$$

Efficiency of QMC

- QMC employs sampling to obtain

$$< E_T >= \frac{\sum_i^M w(\mathbf{R}_i) E_L(\mathbf{R}_i)}{\sum_i^M w(\mathbf{R}_i)}, \quad E_L = \frac{\hat{H}\Psi_T(\mathbf{R})}{\Psi_T(\mathbf{R})}$$

with an error bar $\delta = \frac{\sigma}{\sqrt{M}}$, $\boxed{\sigma^2 =< E_T^2 > - < E_T >^2}$

variance

- Efficiency of QMC simulations is high, when
  - Variance is small: $\sigma \rightarrow 0$ as $\Psi_T \rightarrow \Psi$ (zero-variance)

    Physical insights & improved optimization

  - M/τ, the rate of MC sample generation is high

    Parallelism, compact form of Ψ_T & optimized kernels

Accelerating QMC

- Better Ψ_T

Accelerating QMC

- Better $\Psi_T = e^{J_1 + J_2 + \cdots} \sum_k^M C_k D_k^{\uparrow}(\phi) D_k^{\downarrow}(\phi)$

$$N = N^{\uparrow} + N^{\downarrow}$$

Correlation (Jastrow)

$$J_1 = \sum_i^N \sum_I^{N_{ions}} u_1(|\mathbf{r}_i - \mathbf{r}_I|)$$

$$J_2 = \sum_{i \neq j}^N u_2(|\mathbf{r}_i - \mathbf{r}_j|)$$

Anti-symmetric function (Pauli principle)

$$D_k^{\sigma} = \left| \begin{matrix} \phi_1(\mathbf{r}_1) & \cdots & \phi_1(\mathbf{r}_{N^\sigma}) \\ \vdots & \vdots & \vdots \\ \phi_{N^\sigma}(\mathbf{r}_1) & \cdots & \phi_{N^\sigma}(\mathbf{r}_{N^\sigma}) \end{matrix} \right|$$

Single-particle orbitals

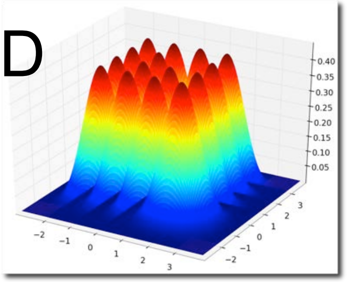
$$\phi_i = \sum_l^{l=N_b} c_l^i \boxed{\Phi_l}$$

Basis sets: molecular orbitals, plane-wave, grid-based orbitals …

Accelerating QMC

- Better Ψ_T

2D



- Improved algorithms

[1] $$\phi_i = \sum_{l}^{l=N_b} c_l^i \Phi_l$$

[2] $$\sum_{i}^{M\to\infty} C_i D_i^{\uparrow} D_i^{\downarrow}$$

[3] Optimization of Ψ_T

[1] einspline library, Esler, http://einspline.svn.sourceforge.net/
[2] Clark *et al.*, JCP **135** 244105 (2011); Morales *et. al.* (2012)
[3] Umrigar, *et. al.*, PRL **98**,110201 (2007)

Accelerating QMC

- Better Ψ_T
- Improved algorithms
- Faster computers
- Bigger computers

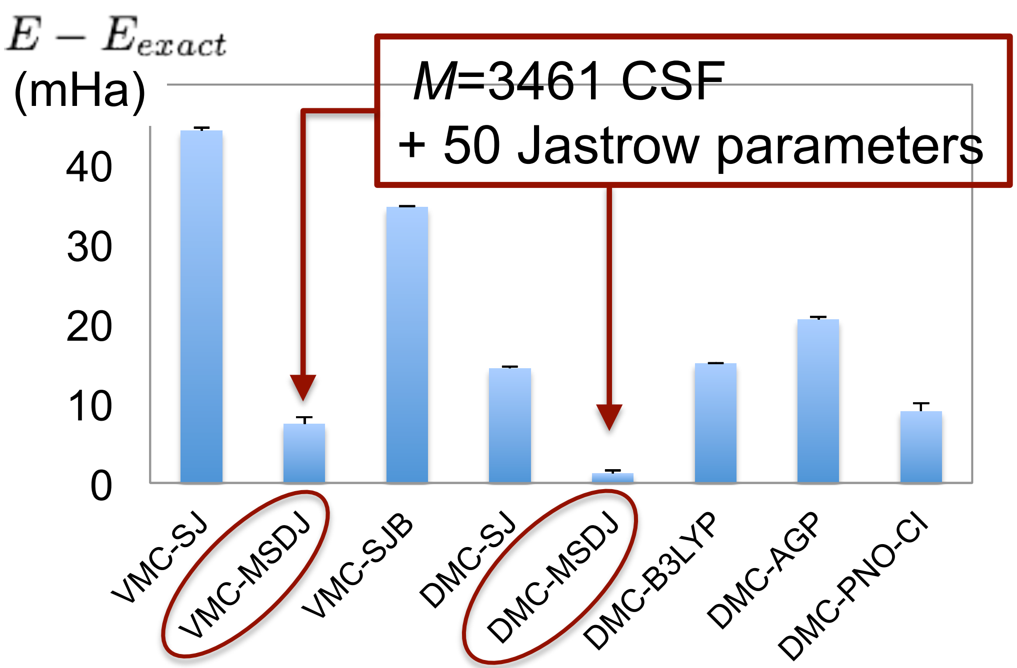
Increase the QMC efficiency
Minimize time-to-solution (wall-clock time) to reach a target error bar

More science

State-of-art QMC

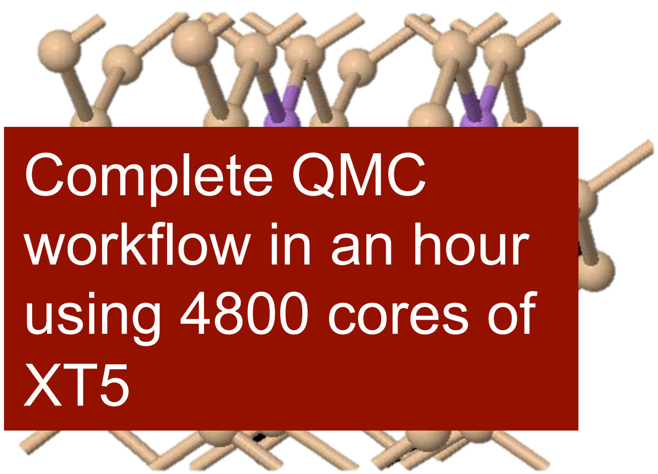
- Fast algorithm for multi-determinant evaluation
- Improved energy minimization in VMC
- QMCPACK: efficient and scalable QMC for large clusters of multi-core and GPUs

Energy of H_2O



Formation energy of a defect in bulk Si (64 atoms)

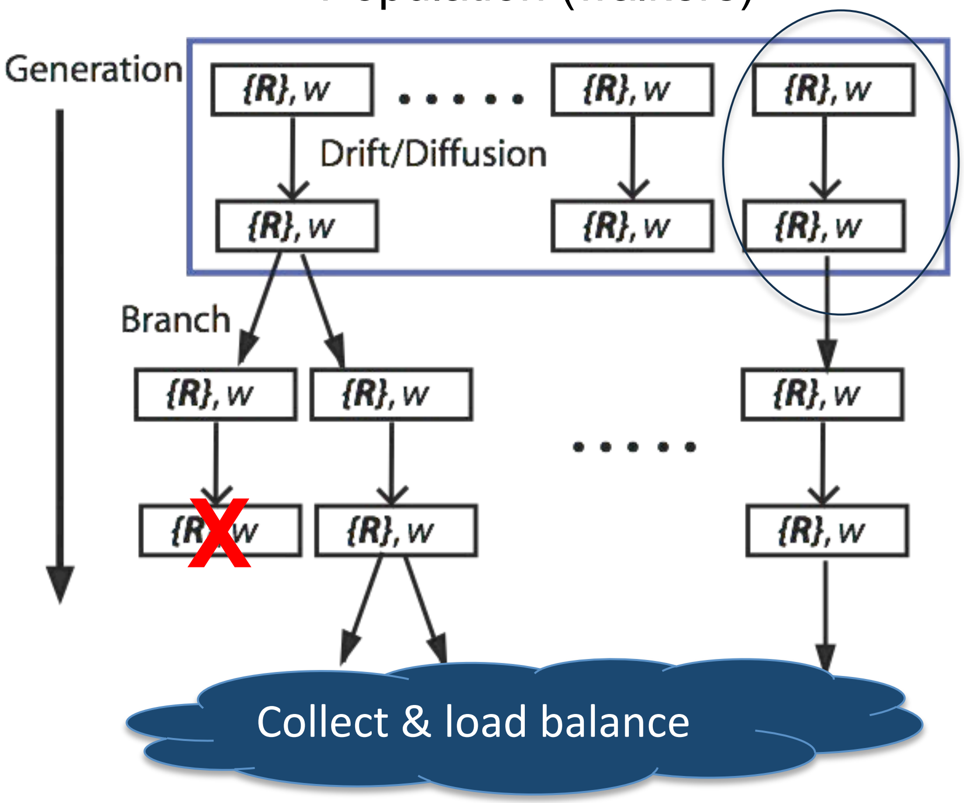
E_f = 3.07 (11) eV



Complete QMC workflow in an hour using 4800 cores of XT5

DMC: computational view

Population (walkers)



Make a move

Random

$$\mathbf{R}' = \mathbf{R} + \tau \boxed{\nabla \ln \Psi_T(\mathbf{R})} + \chi$$

"Quantum Force"

Accept/reject a move

$$\boxed{\frac{|\Psi_T(\mathbf{R}')|^2}{|\Psi_T(\mathbf{R})|^2}} \frac{G_d(\mathbf{R} \rightarrow \mathbf{R}'; \tau)}{G_d(\mathbf{R}' \rightarrow \mathbf{R}; \tau)}$$

Branch with the weight

$$\exp^{-\tau[(E_L(\mathbf{R}) + \boxed{E_L(\mathbf{R}')})/2 - \tilde{E}_T]}$$

- Computationally Intensive: Quantum Force, Ratio, Local Energy
- Communication light
- Ample parallel opportunities : configurations, k-point, walkers

QMC on GPUs: Why & How

- Major performance limiting factors for QMC
  - Random access
  - Memory bandwidth
  - Mostly BLAS I/II operations
- GPUs provide higher bandwidth & FLOPS/s than conventional CPUs : acceleration possible
- But, how to exploit GPUs' power? Need
  - Keep GPUs busy
  - Minimize/hide cost of data transfer between CPUs & GPUs
  - Expose fine-grained parallelisms

QMC on GPU

for walker $= 1 \cdots N_w$ **do**
  let $\mathbf{R} = \{\mathbf{r}_1 \ldots \mathbf{r}_N\}$
  for particle $i = 1 \cdots N$ **do**
    set $\mathbf{r}_i^{'} = \mathbf{r}_i + \delta$
    let $\mathbf{R}^{'} = \{\mathbf{r}_1 \ldots \mathbf{r}_i^{'} \ldots \mathbf{r}_N\}$
    ratio $\rho = \boxed{\Psi_T(\mathbf{R}^{'})/\Psi_T(\mathbf{R})}$
    if $\mathbf{r} \rightarrow \mathbf{r}'$ is accepted **then**
      update the state of a walker
    end if
  end for{particle}
  Compute $E_L = \boxed{\hat{H}\Psi_T(\mathbf{R})/\Psi_T(\mathbf{R})}$
end for{walker}
Reweight and branch walkers
Update E_T and collect properties

Loops

- Restructure the algorithm and data structure to expose & exploit parallelisms

multiple walkers per kernels

* Esler, Kim, Shulenburger & Ceperley, CISE (2010)

QMC on GPU

for walker $= 1 \cdots N_w$ **do**
  let $\mathbf{R} = \{\mathbf{r}_1 \ldots \mathbf{r}_N\}$
  for particle $i = 1 \cdots N$ **do**
    set $\mathbf{r}_i' = \mathbf{r}_i + \delta$
    let $\mathbf{R}' = \{\mathbf{r}_1 \ldots \mathbf{r}_i' \ldots \mathbf{r}_N\}$
    ratio $\rho = \boxed{\Psi_T(\mathbf{R}')/\Psi_T(\mathbf{R})}$
    if $\mathbf{r} \rightarrow \mathbf{r}'$ is accepted **then**
      update the state of a walker
    end if
  end for{particle}
  Compute $E_L = \boxed{\hat{H}\Psi_T(\mathbf{R})/\Psi_T(\mathbf{R})}$
end for{walker}
Reweight and branch walkers
Update E_T and collect properties

Loops

- Restructure the algorithm and data structure to expose & exploit parallelisms

  multiple walkers per kernels

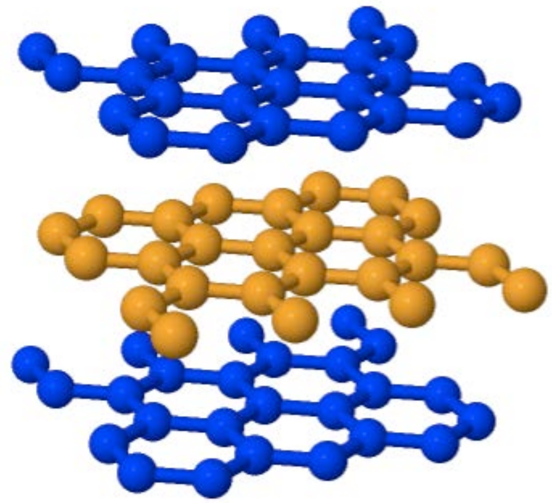
- Mixed precision sufficient for the target accuracy
- Higher-level implementation intact
- MPI for load balancing & reductions : sustain high parallel efficiency

* Esler, Kim, Shulenburger & Ceperley, CISE (2010)

Graphite Benchmark Problem

- DMC simulations of graphite
  N=256 electrons (4x4x1, 64 carbons)
- Efficiency = # of MC samples / sec
  Reflects node performance
- Walkers distributed over MPI
  - CPU: 4 MPI and 8 threads per node; each thread works on different walkers
  - GPU: 1 MPI per GPU; work on all the walkers concurrently

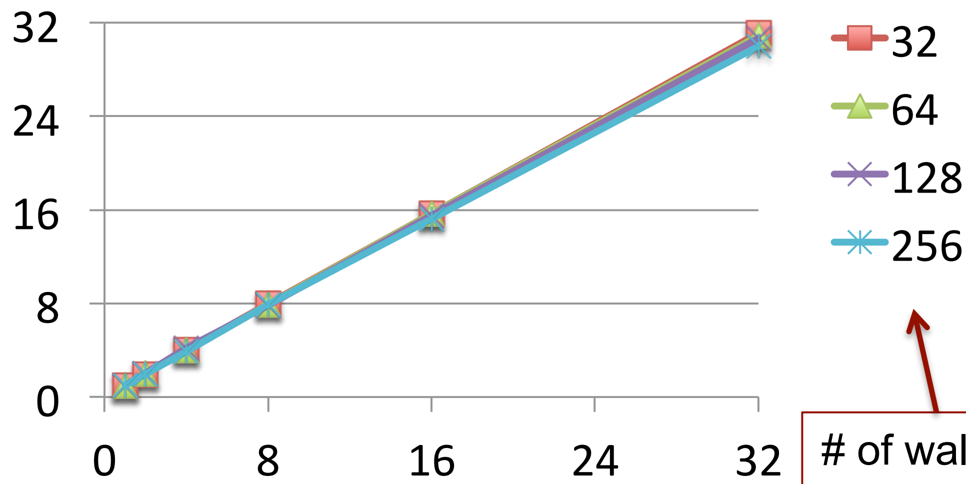
Graphite
using 24x24x4 k points



- Weak-scaling: fixed samples (works) per node per MC step
- Strong-scaling: fixed total samples per MC step
- Parallel efficiency of weak-scaling has to be perfect, unless the target samples are set to too low (unrealistic) or system problems (MPI,…)
- Node-to-node comparison of XK6 w or wo GPUs & XE6

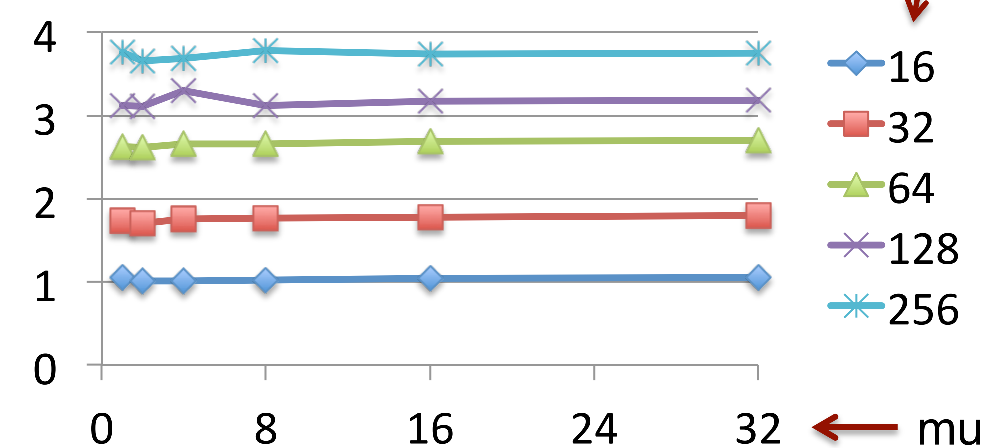
Performance on Titandev@OLCF

Parallel efficiency of XK6 w GPU



# of walkers per GPU

Speedup of GPU (node-to-node)

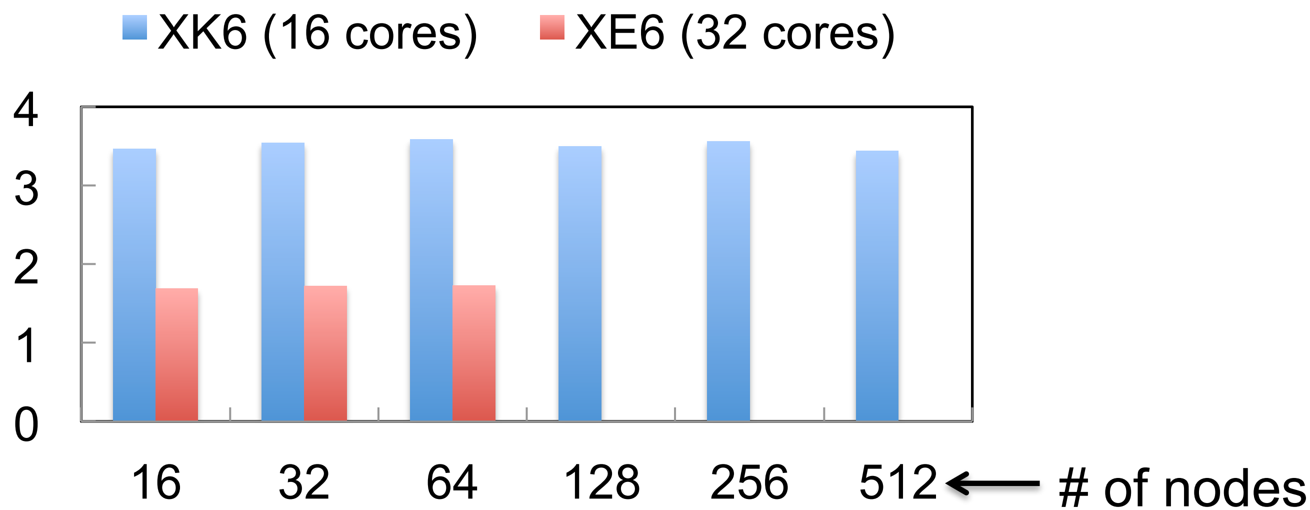


multiples of 16 nodes

- Fix # of walkers per node = "average" work per node
- Time-to-solution decreases with increasing MPI nodes: strong scaling in science
- GPU Speedup compared to 4-walker per core runs on Titan

Comparisons of XK6 and XE6*

Speedup on XK6g over XK6c & XE6



- The number of walkers per node is fixed at 128

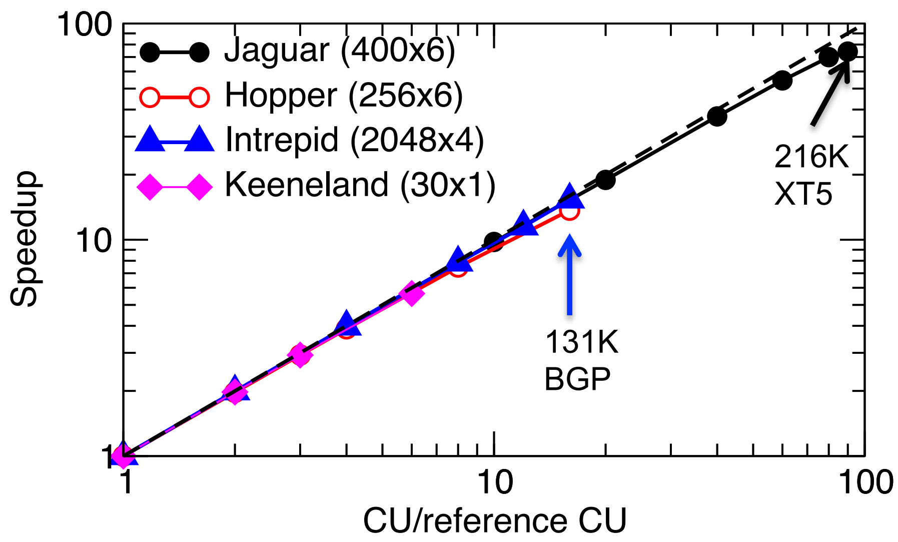
  [XK6g] 128 per GPU [XK6c] 8 per core [XE6] 4 per core

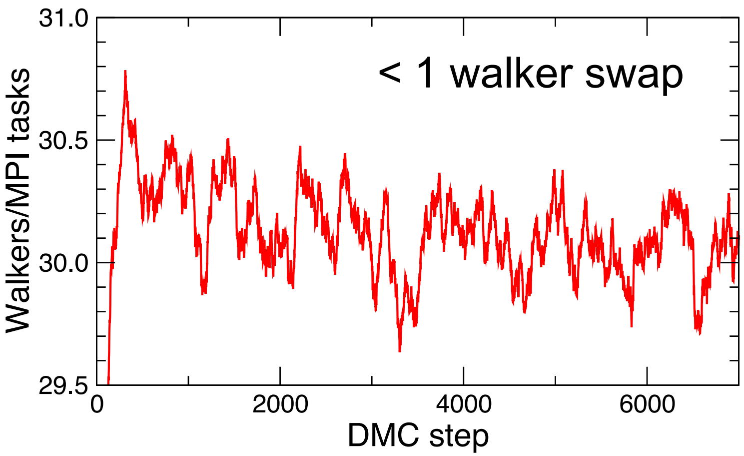
- CPU in double precision, GPU in mixed precision
- XK6g: optimal throughput with large systems, many walkers
- XK6c & XE6, insensitive to the number of walkers

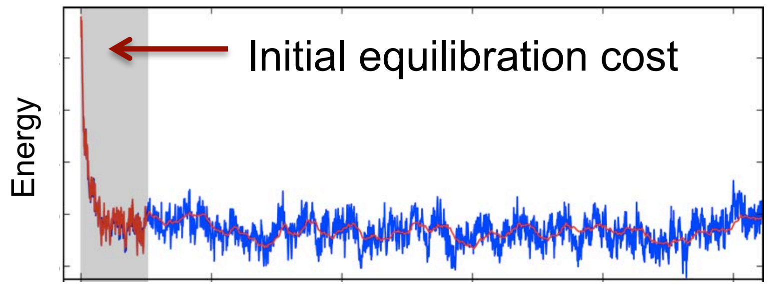
*Titan/Titandev @OLCF and Monte Rosa @CSCS

Parallel Performance

- DMC scaling is almost perfect , > 90% efficiency
  - Limited by collectives for $E_T,\ N_p^w - < N^w >$
- 1 MPI to 1 GPU or NUMA mapping
  - Large average number of walkers per MPI task, thus small fluctuations : easy to balance walkers per node







- (MPI x OpenMP) for the reference Compute Unit
- Keeneland@NICS, Fermi (3 cards per node)
- Strong scaling on CPUs; weak scaling on GPUs

Status Update on XK6

- Obtained the speedup as expected by the bandwidth and peak FLOPS of GPUs

More Accurate Answers, Faster

Breakthrough QMC Simulations

- Works in progress
  - Maximize utilizations : asynchronous, non-blocking operations*
    - Launch several kernels in parallel
    - Fast atomics to communicate between blocks
    - Facilitated by newer GPUs and CUDA
  - Utilize CPU as well: potentially > 25% gains on XK6
  - Enable larger systems to be simulated: Use GPU peer-to-peer to allow distribution of read-only orbital dataset between GPUs on same node

*Cliff Woolley & Chris Cameron, NVIDIA

QMC on 10-100 PF

Better QMC simulations with more resources

- Increase fidelity of QMC simulations: better Ψ_T
- Beyond ground-state simulations: excited states, finite temperatures
- Accelerate discovery

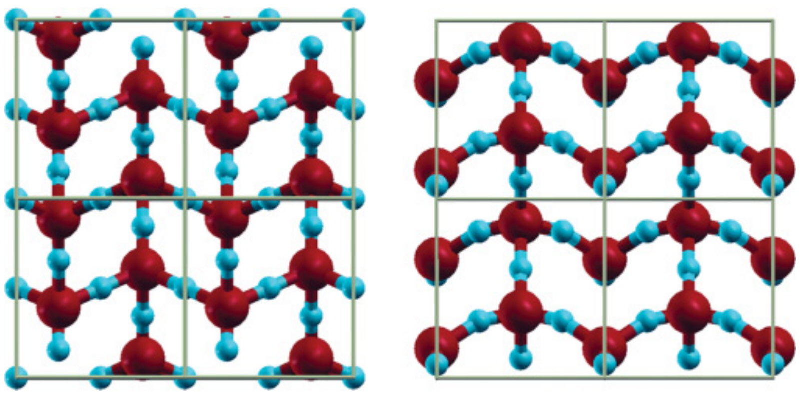
Materials Genome Initiative*

In the same way that the Human Genome Project accelerated a range of biological sciences by identifying and deciphering the basic building blocks of the human genetic code, the Materials Genome Initiative will speed our understanding of the fundamentals of material science, providing a wealth of practical information that entrepreneurs and innovators will be able to use to develop new products and processes.

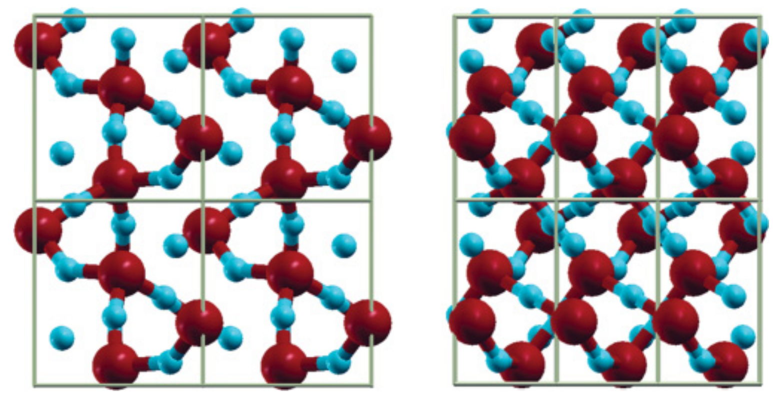
http://www.whitehouse.gov/blog/2011/06/24/materials-genome-initiative-renaissance-american-manufacturing

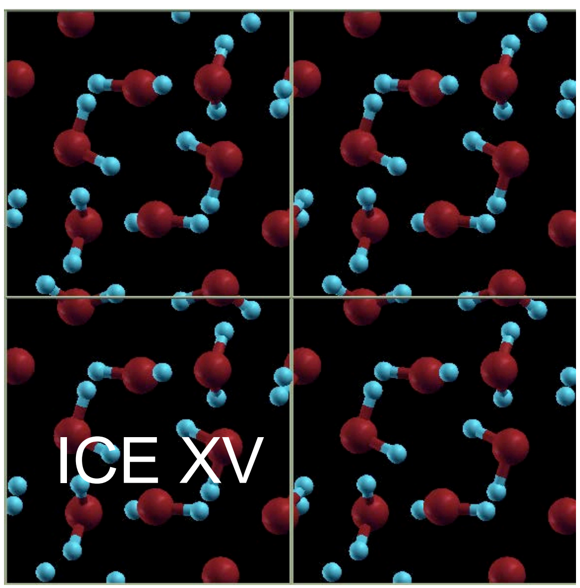
High-pressure Phases of H_2O*

(a) 1 TPa



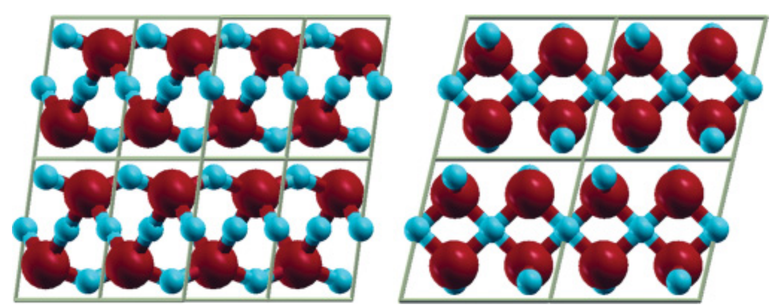
(b) 1.4 TPa





ICE XV:Salzmann et al, PRL (2009)

(c) 5 TPa



*Figures by J. M. McMahon, Phys. Rev. B **84**, 220104(R) (2011)

Challenges

- New methods and algorithms

  E.g., current algorithms and data structure on GPUs are not ideal for the fast algorithm for massive multi-determinant expansions

$$\sum_{i}^{M\to\infty} C_i D_i^{\uparrow} D_i^{\downarrow} \longleftarrow$$

10-100 times more expensive than M=1

- New bottlenecks: e.g., general eigen-value solvers, I/O
- Deeper memory, communication and algorithmic hierarchy
- Mixed precisions: single to quad
- Exploit increase node performance & parallelism

Programming models and software environments for productivity: need portable and efficient solutions on multiple platforms, now and future

Conclusions

- QMC has kept up with the HPC evolution and will continue improving predictive powers in physics, materials and chemistry

  ✓ Clusters of multi- and many-core SMP

  ✓ Clusters of GPU

  😰 Clusters of hybrid

  😱 What is next

- More to be done improve science productivity
  - Reduce impacts of application-level, software and hardware faults: Algorithms for robust and fault-tolerant simulations
  - Faster off-node communication and I/O

Acknowledgements

Supported by

- QMC Endstation (DOE-ASCR)
- PetaApps & PRAC (NSF-DMR, OCI)
- Materials Computation Center, University of Illinois (NSF-DMR)
- Center for Defect Physics, ORNL (DOE-BES)
- ORNL-LDRD (DOE-BES/ASCR)

Computing resources provided by

- INCITE and ALCC
  - Oak Ridge Leadership Computing Facility (OLCF)
  - Argonne Leadership Computing Facility (ALCF)
- NSF Teragrid facilities at NCSA, NICS, PSC and TACC
- National Energy Research Scientific Computing Center (NERSC)