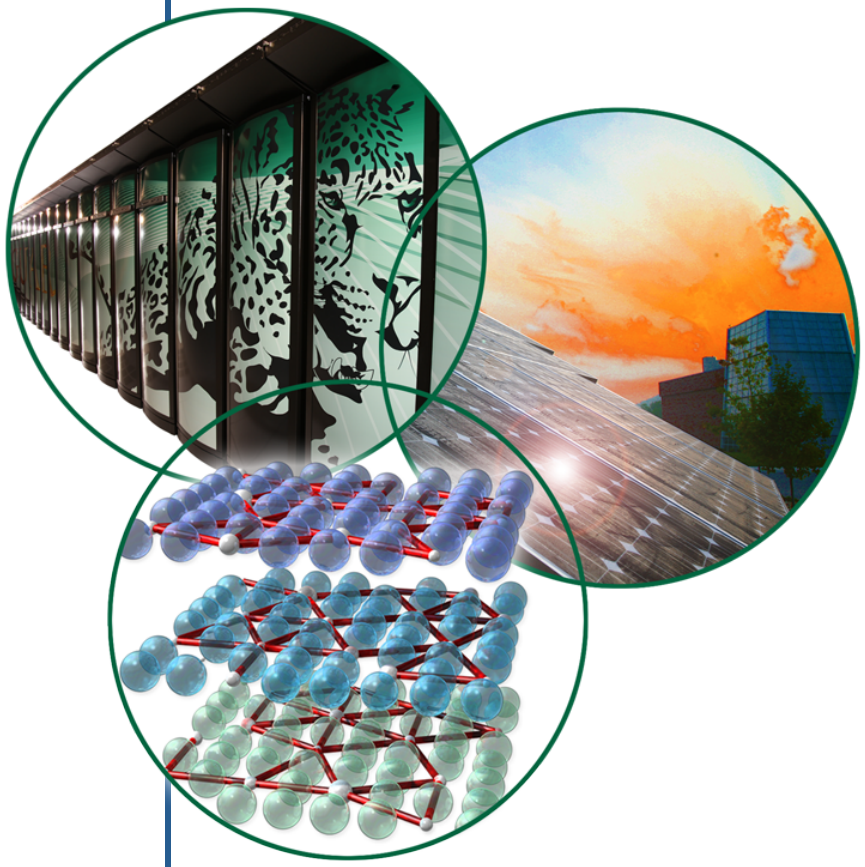
Preparing Quantum Monte Carlo for Exascale

Presented at

Productive Programming Models for Exascale, 2012

Portland, OR



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)
- Anour Banali (ANL)
- Nick Romero (ANL)
- Simone Chiesa (W&M)
- Kris Delaney (UCSB)

& David M. Ceperley (UI)

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

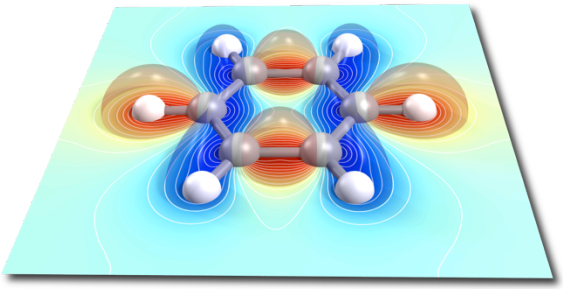
- INCITE and ALCC @ OLCF, ALCF
- NSF Teragrid facilities at NCSA, NICS, PSC and TACC

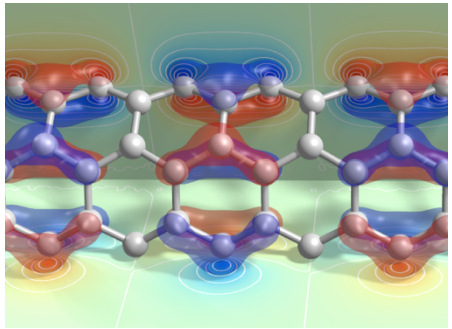
http://qmcpack.cmscc.org

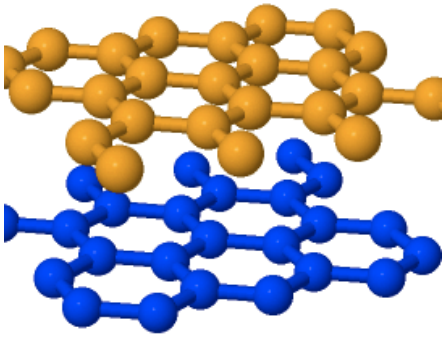
QMC advantages: accuracy and scalability

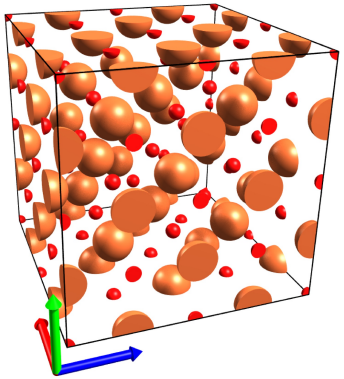
- Applicable to a wide range of problems
  - Any boundary conditions: molecular and solid-state systems
  - Dimensionality: 1D, 2D, and 3D
  - Representation: atomistic to model Hamiltonians
- Scale with a few powers in system size: $O(N^3)$-$O(N^4)$
  - Routine calculations of 100s-1000s electrons
-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

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

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: $\sigma \rightarrow 0 \;\; \text{as} \;\; \Psi_T \rightarrow \Psi$ (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)

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

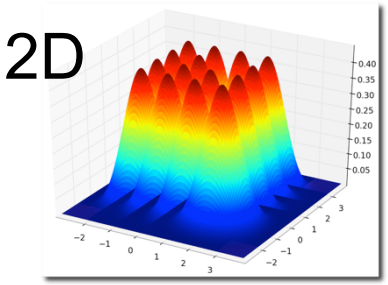
Single-particle orbitals

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

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

Accelerating QMC

- Better Ψ_T



- 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

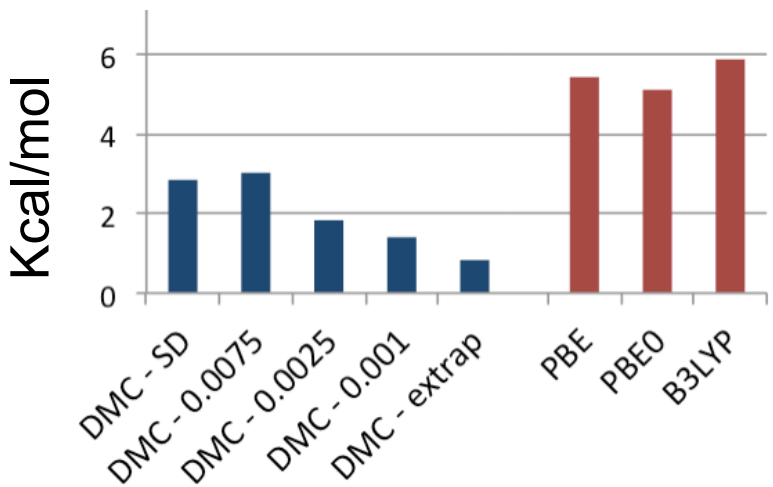
More science

State-of-art QMC

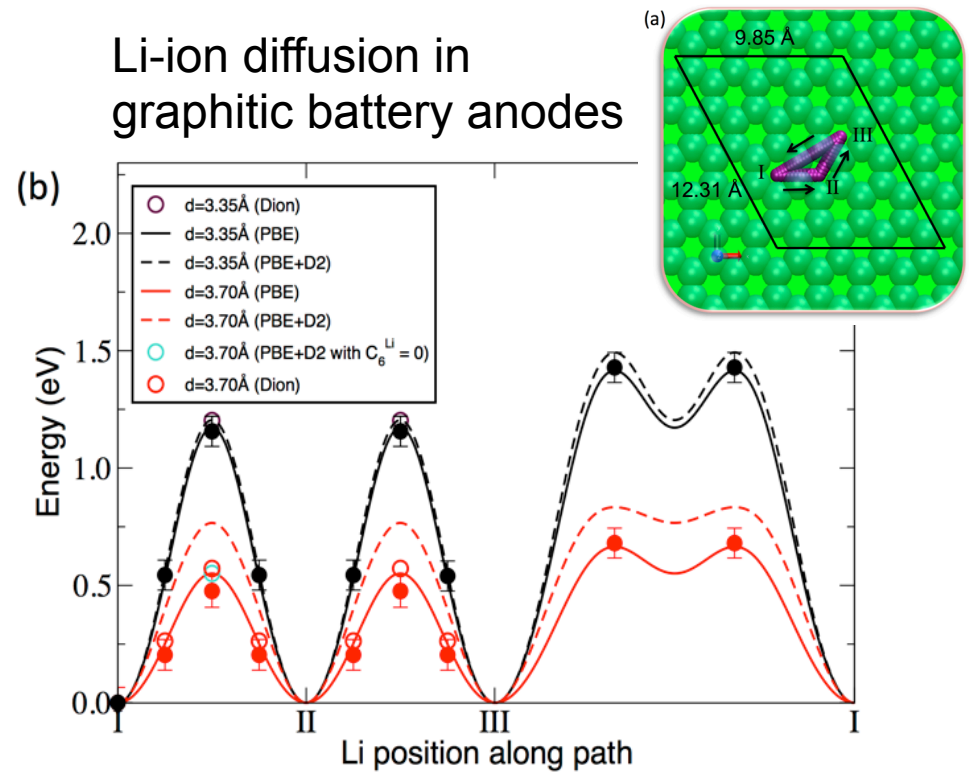
- Fast algorithm for multi-determinant evaluation
- Improved optimization of trial many-body wavefunctions
- QMCPACK: efficient and scalable QMC for large clusters of multi-core and GPUs

More Science, High Fidelity

MAE of atomization energies*
55 of G1 set

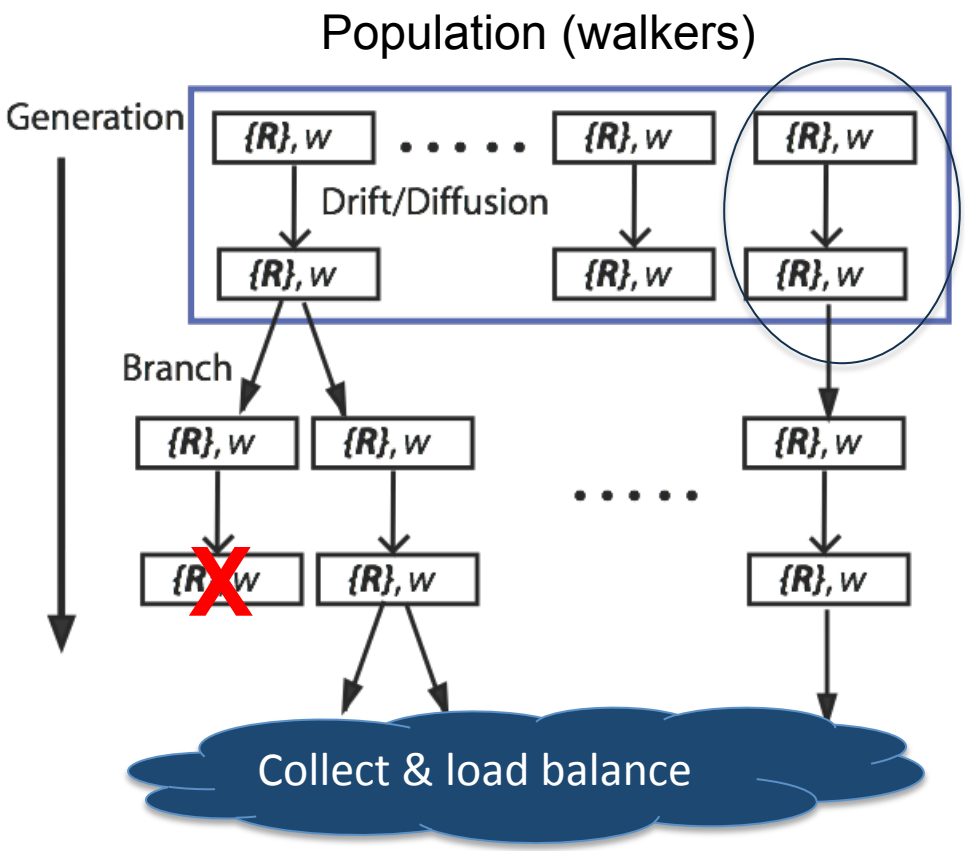


Li-ion diffusion in graphitic battery anodes



*Morales & Kim, JTCP 2012

DMC: computational view



Make a move

Random

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

"Quantum Force"

Accept/reject a move

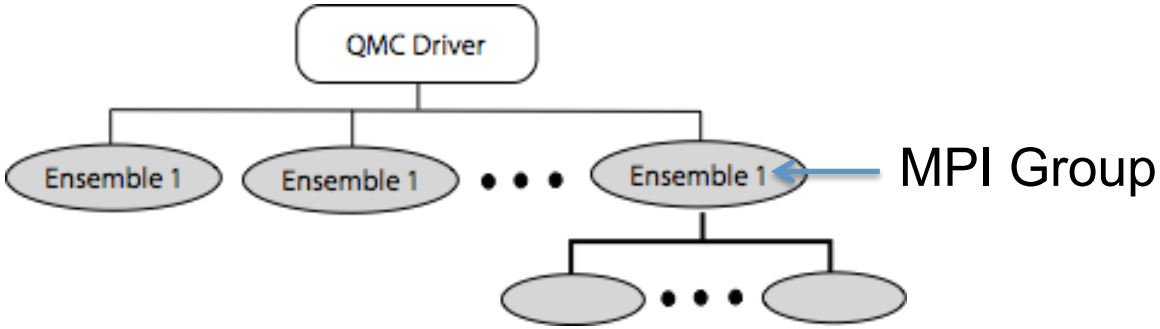
$$\frac{|\Psi_T(\mathbf{R}')|^2}{|\Psi_T(\mathbf{R})|^2} \frac{G_d(\mathbf{R} \to \mathbf{R}'; \tau)}{G_d(\mathbf{R}' \to \mathbf{R}; \tau)}$$

Branch with the weight

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

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

MPI+X Model for QMC

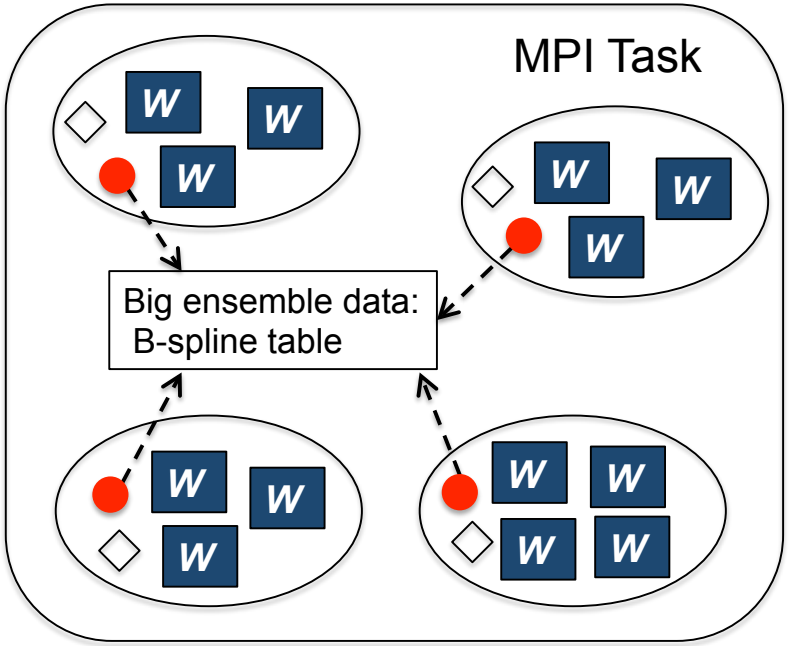


X on SMP
OpenMP, CUDA,
Threads ….

Each group

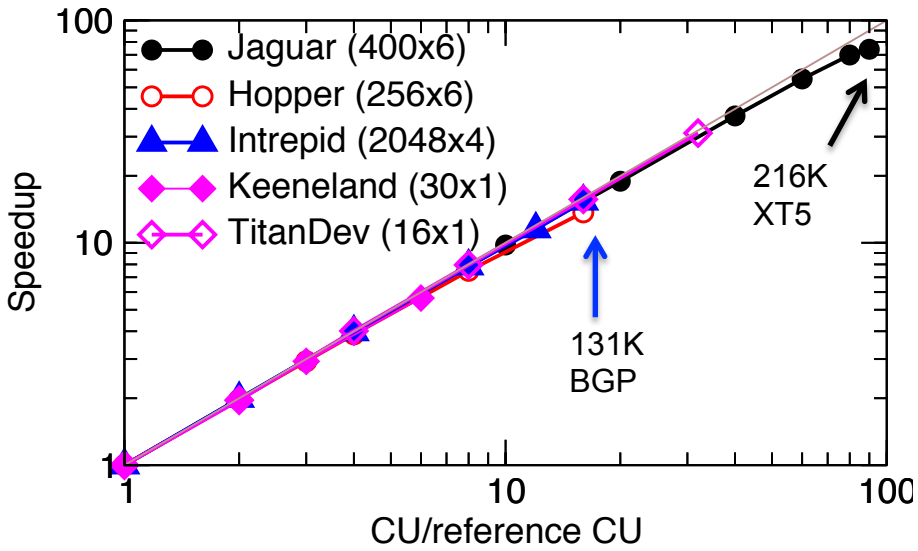
for generation $= 1 \cdots N_{\mathrm{MC}}$ **do**
  for walker $= 1 \cdots N_w$ **do**

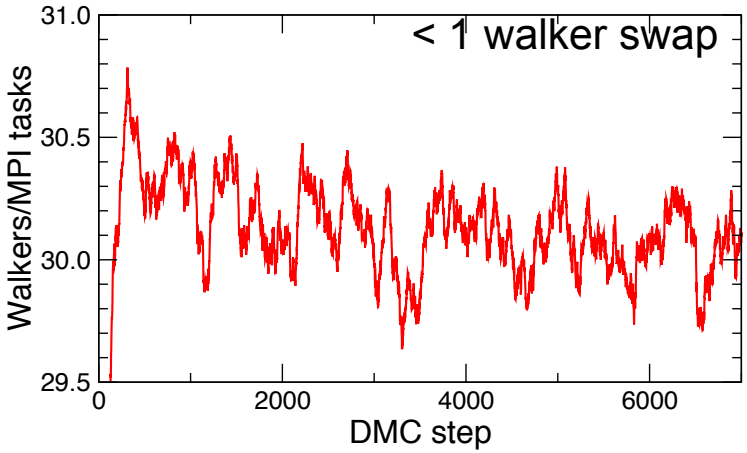
  end for{walker}
  Reweight and branch walkers
  Update E_T and collect properties
end for{generation}

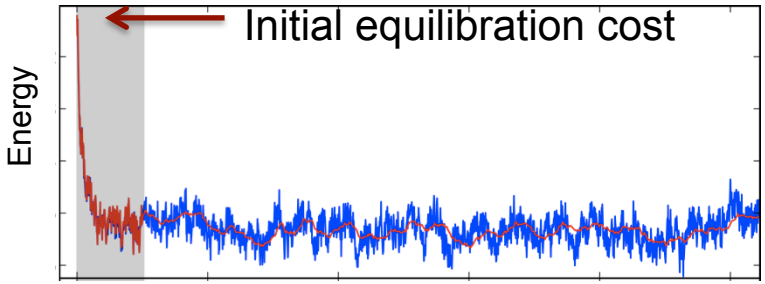


Parallel Performance

- 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
  - Gain in Collectives
  - 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

(MPI+X)/C++ has worked (so far)

- Simple but effective for QMC
- Productive solution for developers (physicists)
  - Abstraction for physics concepts, methods and algorithms
  - Extensible and efficient
- Object-oriented and generic programming
  - Allow separation of science from coding
  - Lazy implementations and optimizations
- Efficient memory and thread management
  - Oversubscription on Intel & IBM BGQ increases performance
- Portable and scalable from a laptop to HPC
- Optimal use of current multi/many-core SMP architectures
- Exploit steady improvement of HPC ecosystem
  - Compilers, open standards, systems, numerical libraries, I/O

Managing Complexity

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

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

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

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

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

Single-particle orbitals

- Multiple forms
- Multiple precisions
- Nested parallelism
- Scaling with *N* & *M*

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

QMC on Exascale

Better QMC simulations with more resources

- Increase fidelity of QMC simulations: better
- Beyond ground-state simulations: excited states, finite temperatures
- Multi-code execution: MC with QMC
- Increase complexity and cost per sample: constant time-to-solution

General plans

- Probably (MPI*+X)/C++ will work
  - Clusters of powerful nodes
  - Each node has ample memory and a lot of concurrency
- What is X on node: options abound
- Doing a lot better within X
  - Most of the speedup will come from vector units
  - Intrinsic specialization: e.g.,SSE, AVX and QPX in numerical library
  - Need and extend specialized containers and operators on target architectures

For Productivity

- Multiple solutions that can be composed and replaced
  - Too hard for a single programming model to deal with complexity and to support diverse use cases
- Ecosystem improvement : compilers, libraries, tools and well-trained/trainable programmers
- Division of labor
  - Applications manage data and abstractions
  - Performance enhancement by professionals
    - Smart allocators, iterators, auto code generation
- Optimized execution for each run
  - Input is “compiled” for a problem size, representations, solutions and execution tree
  - Profiling and auto tuning

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
- (MPI+X)/C++ for now

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)