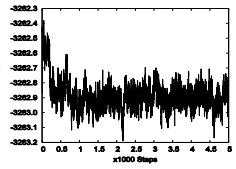
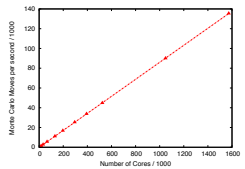
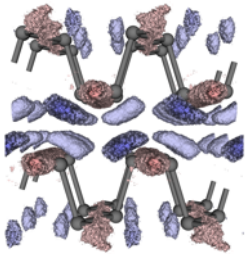
SAND2017-6625C







Extending Strong Scaling of Quantum Monte Carlo (QMC) to the Exascale

L. Shulenburger

Sandia National Laboratories

Sandia National Laboratories

Exceptional service in the national interest

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

QMC solves quantum many-body electronic structure

- Solve the Schrodinger equation for the electrons in a material to determine its properties

$$\left[-\frac{\hbar^2}{2m}\vec{\nabla}^2 + V_{e-e}(\vec{r},\{\vec{R}\}) + V_{e-I}(\vec{r},\{\vec{R}\})\right]\psi_q(\vec{r}) = E_q^{BO}\psi_q(\vec{r})$$

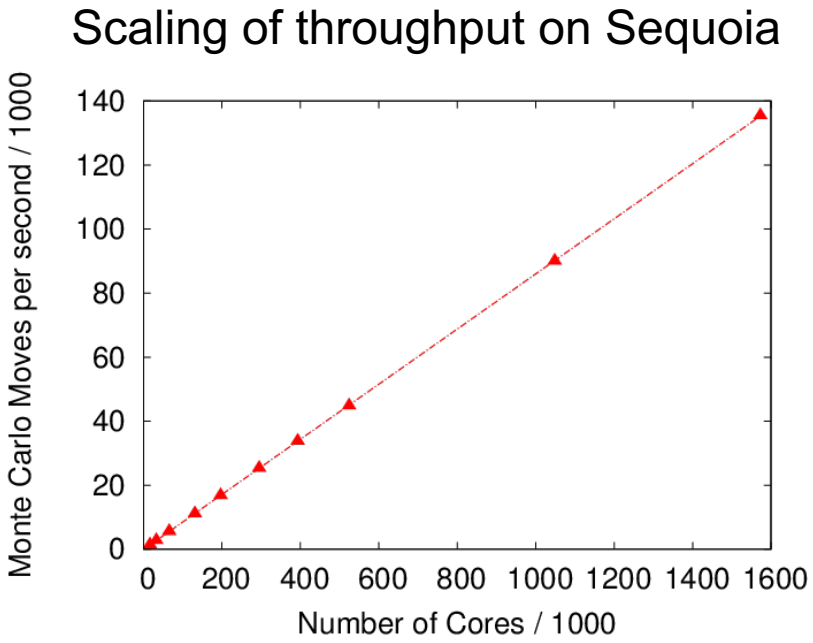
→ Only input: N, $\vec{R}$ and Z

- This involves integrals in 3N dimensions
  - QMC uses a stochastic process to avoid curse of dimensionality
  - Green's function is sampled using a Markov process (many nearly independent walkers)
  - Results in highly accurate solutions, but at high computational cost
- Mature codes exist, eg. QMCPACK, Casino, QWALK, CHAMP…

QMC algorithm is ideally suited to leadership computing

Sandia National Laboratories

- Current parallelization strategy relies on distributing walkers among processing elements
- Highly scalable algorithm, as demonstrated on Sequoia at LLNL
  - Nearly perfect parallel efficiency in moving walkers to over 1.5M cores
- This is commonly taken to mean that QMC has perfect scaling

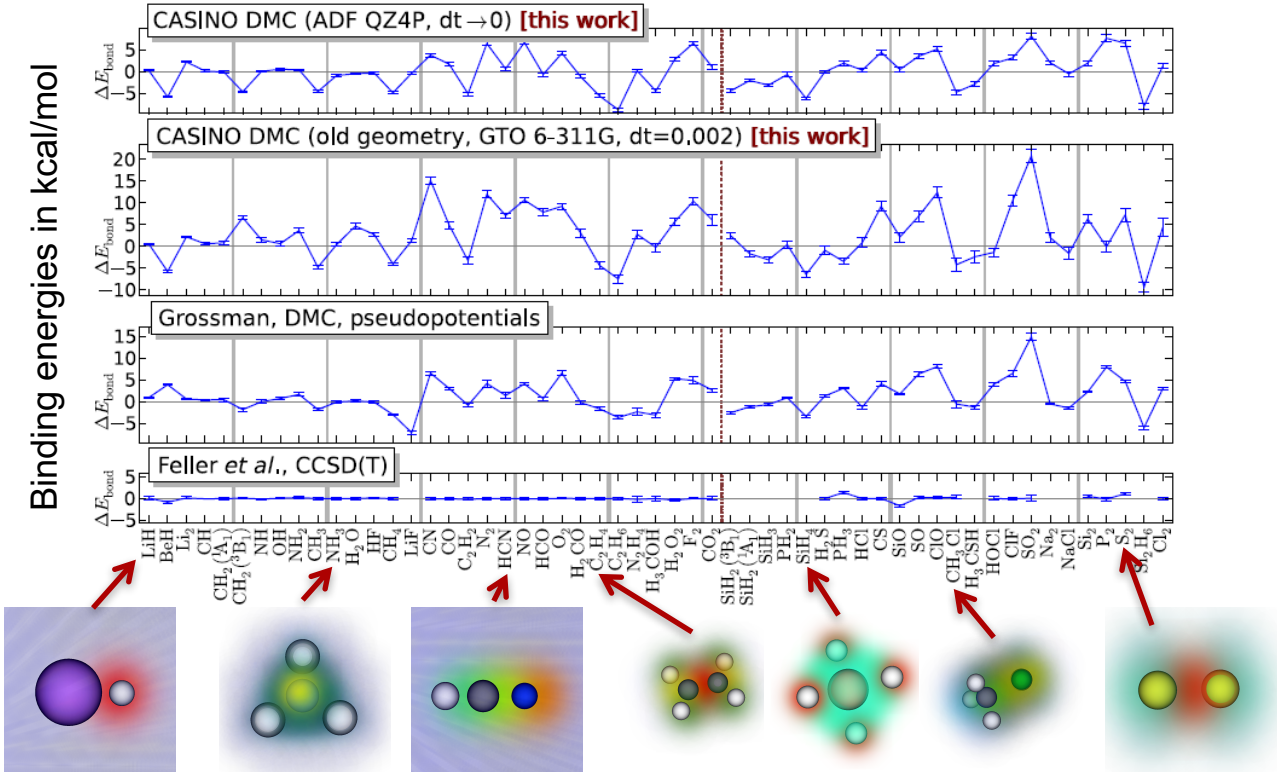


How is this currently used?

Sandia National Laboratories

- Three example cases
  - Accuracy in general molecules
  - General solids
  - Dispersion dominated systems

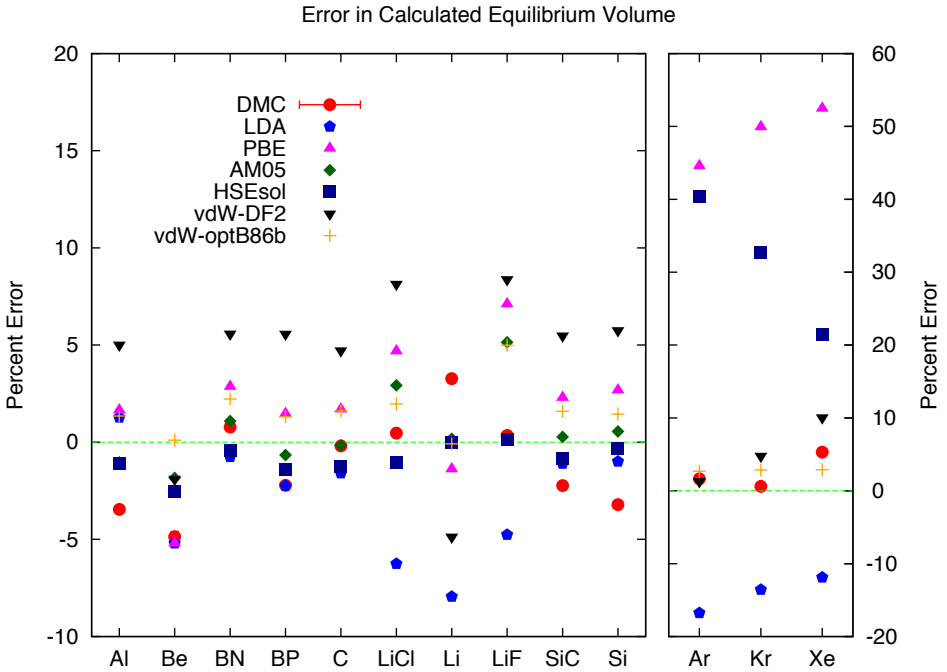
General molecules



•from Nemec et al, JCP. **132**, 034111 (2010)

Has also been applied to a wide range of solids

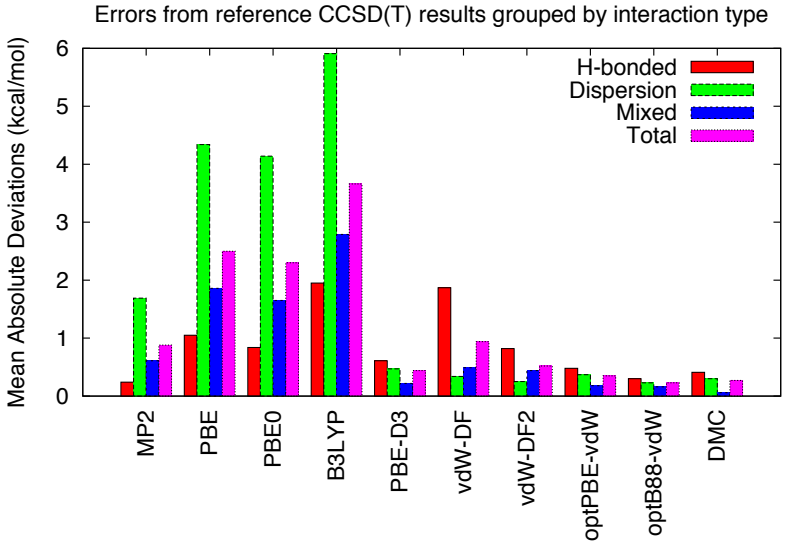
- Calculate energy vs lattice constant for a wide range of molecules
- Compare to a wide range of DFT functionals
- Accuracy is good and is quite general
- This pointed to areas where the methodology could be improved



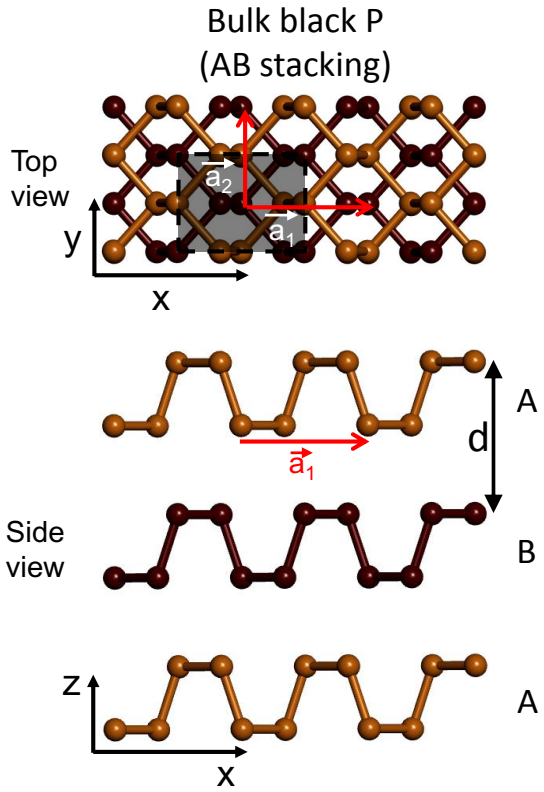
LNS and TRM, PRB 88, 245117 (2013)

Cases where dispersion interactions dominate are particularly interesting

- For example S22 test set collects complexes of molecules bound by dispersion
  - Hydrogen bonded complexes
  - Dispersion bonded complexes
  - Mixed binding complexes
- DMC with a single slater jastrow trial wavefunction is applied
- Performance compared to CCSD(T) is equal to or better than competing methods



Example: layered materials

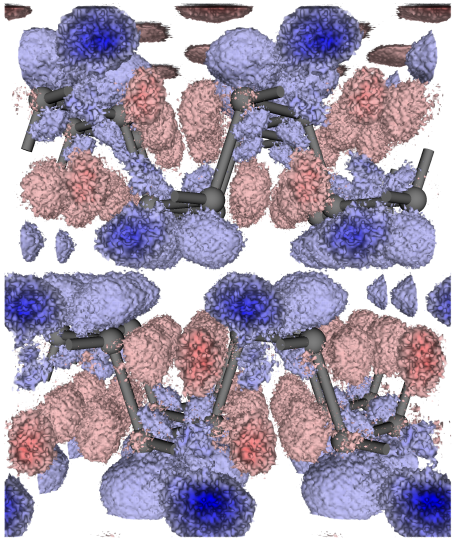


- Black phosphorus is a layered material
- It has recently been exfoliated
- It has a direct band gap
  - Ranges from ~0.3 eV to 2 eV (semiconductor)
- It has recently been made into semiconductor devices
- Interactions within layers are covalent
- Interlayer binding is thought to be mediated by van der Waals forces

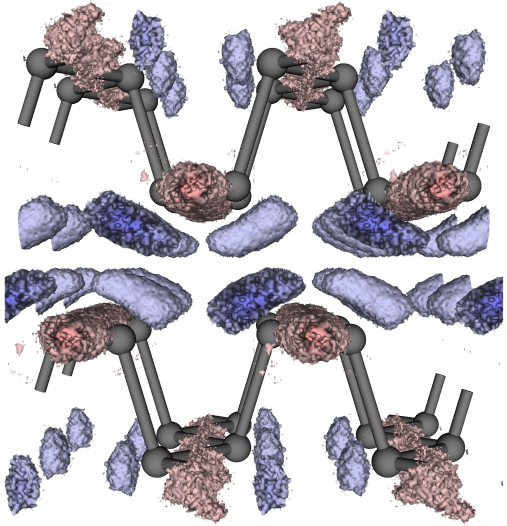
It was found that Phosphorene exhibits large charge redistribution due to the effects of the environment

Sandia National Laboratories

Bulk charge density difference



bilayer charge density difference



- Red is area of increased electron density, blue decreased electron density
- Mixed estimator

Where can we go with exascale?

Sandia National Laboratories

- Strongly correlated materials / more complicated wavefunctions
- Treat the complexity of real materials
  - Need to move from 10's of atoms to many 100's

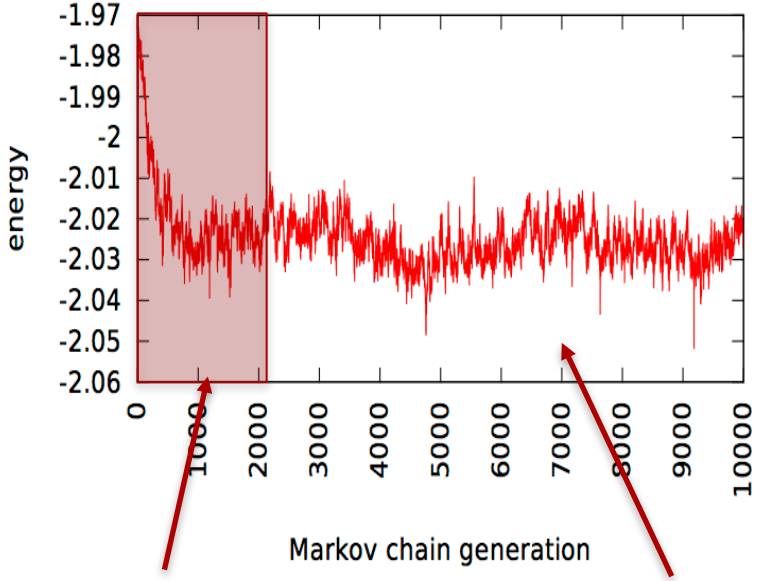
Where can we go with exascale?

Sandia National Laboratories

- Strongly correlated materials / more complicated wavefunctions
- Treat the complexity of real materials
  - Need to move from 10's of atoms to many 100's

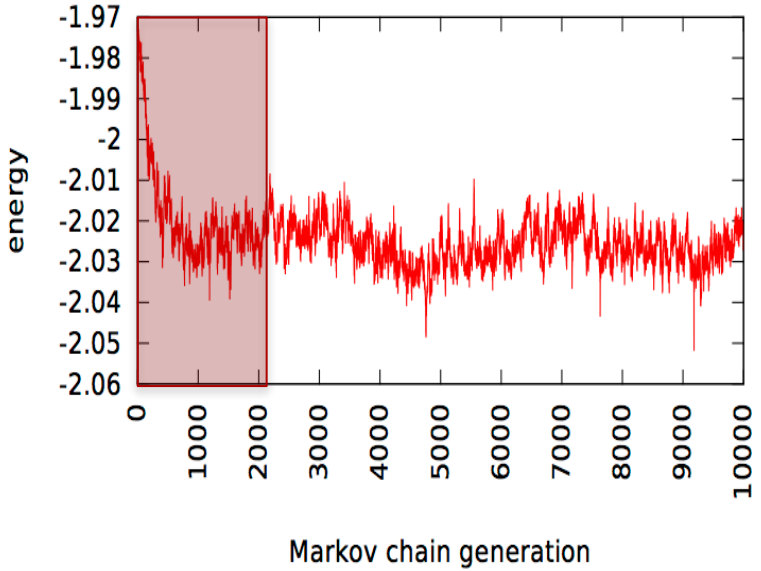
- First, can we just continue with our current algorithm?

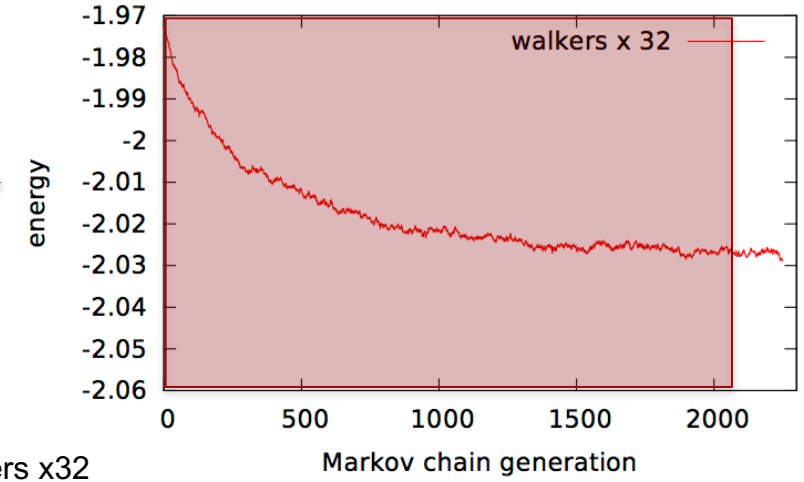
Looking at a calculation in detail



Increasing the number of cores…

- What actually happens as the calculation is scaled?

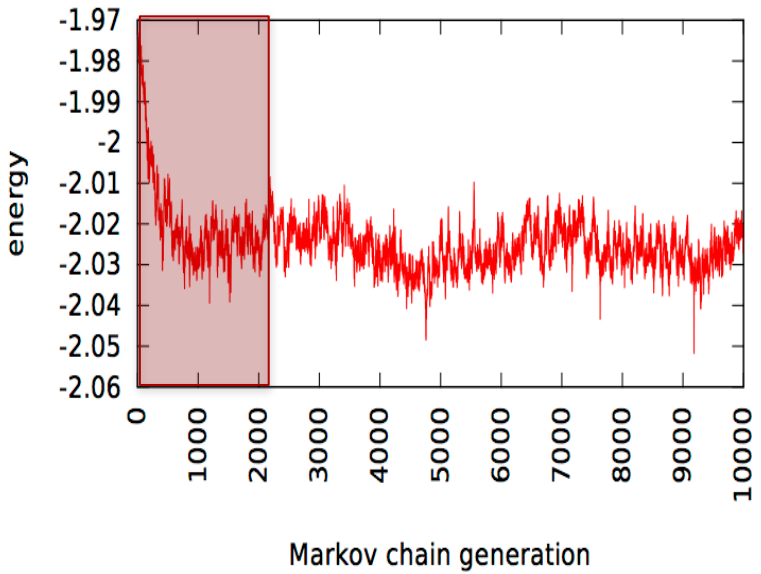


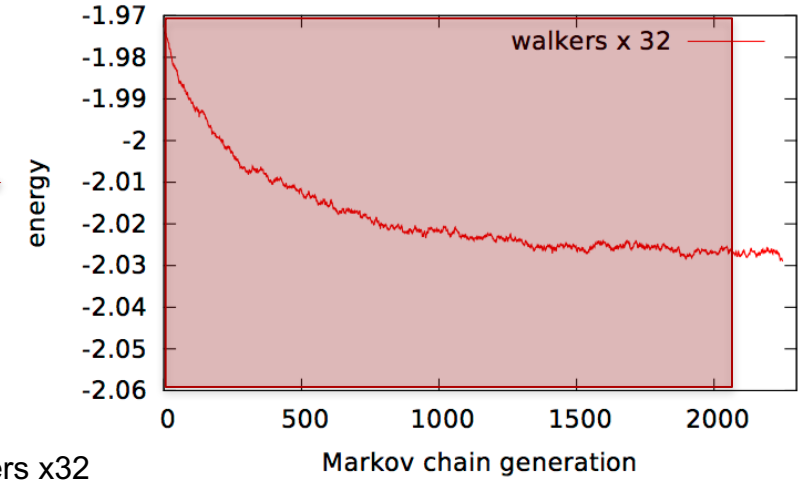


- Number of walkers x32
- Processors x 32
- Throughput x 32
- Wall time / 5
- Parallel Efficiency **~ 15%**

Increasing the number of cores…

- What actually happens as the calculation is scaled?





- Number of walkers x32
- Processors x 32
- Throughput x 32
- Wall time / 5
- Parallel Efficiency **~ 15%**

Equilibration dominates
Naïve strategies will soon fail!

How can we do better?

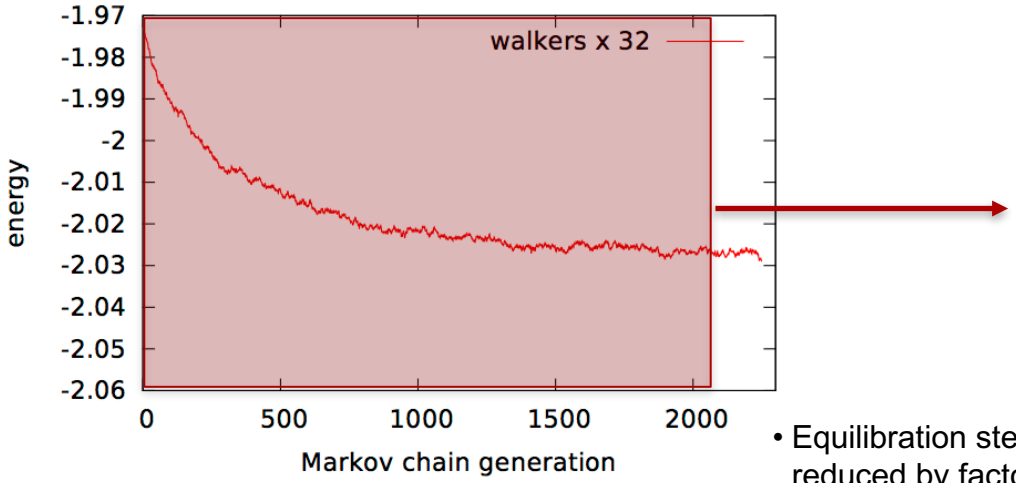
- Go back to the algorithm
- We are propagating the following PDE in imaginary time

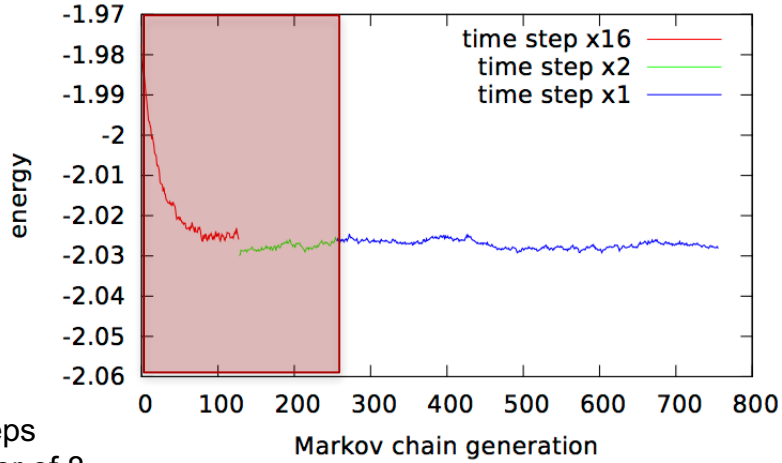
$$|\Psi(t)\rangle = e^{-(\hat{H}-E_T)t}\,|\Psi_T\rangle$$

- A short time approximation is used: t → n Δt
- Δt has to be taken to 0 to minimize discretization errors
  - This is not true in the equilibration phase

Introduce a variable time step

- Take larger steps during equilibration

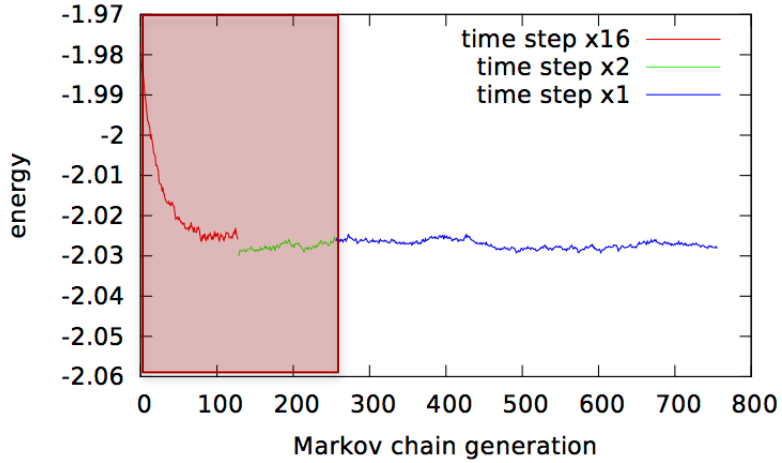


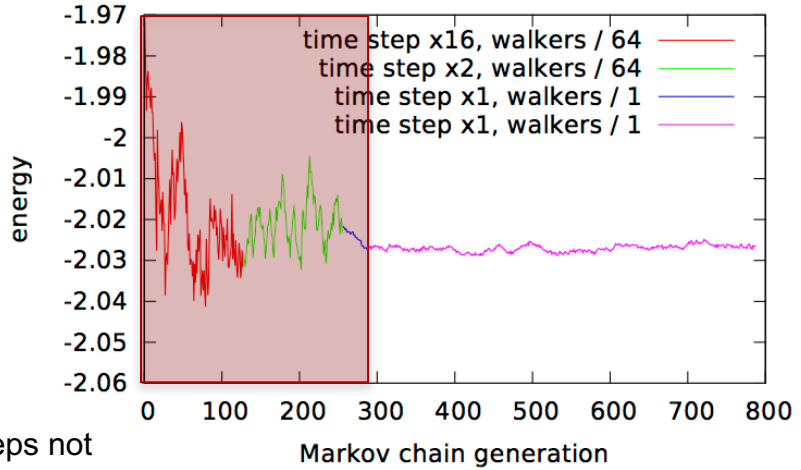


- Equilibration steps reduced by factor of 8
- Wall time / 3
- Parallel Efficiency **~40%**

Population (number of walkers) need not be constant

- Equilibrate a smaller population and then decorrelate
  - Similar to method proposed in Gillan et el. “Petascale computing opens new vistas for quantum Monte Carlo”, Psi-k newsletter, Feb. 2011



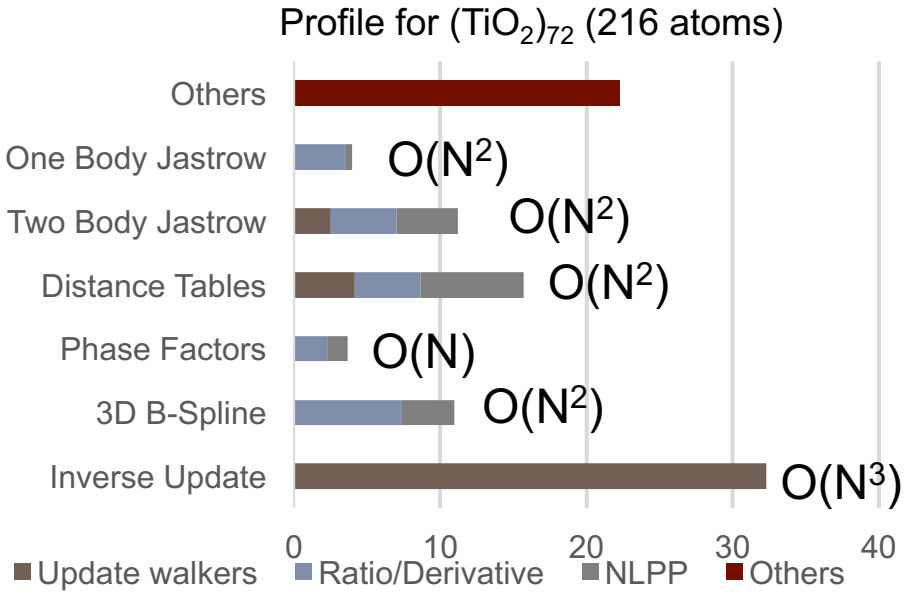


- Equilibration steps not reduced
- Cost to equilibrate decreases by factor of ~ 7
- Parallel Efficiency **> 90%**

How does this relate to larger systems?

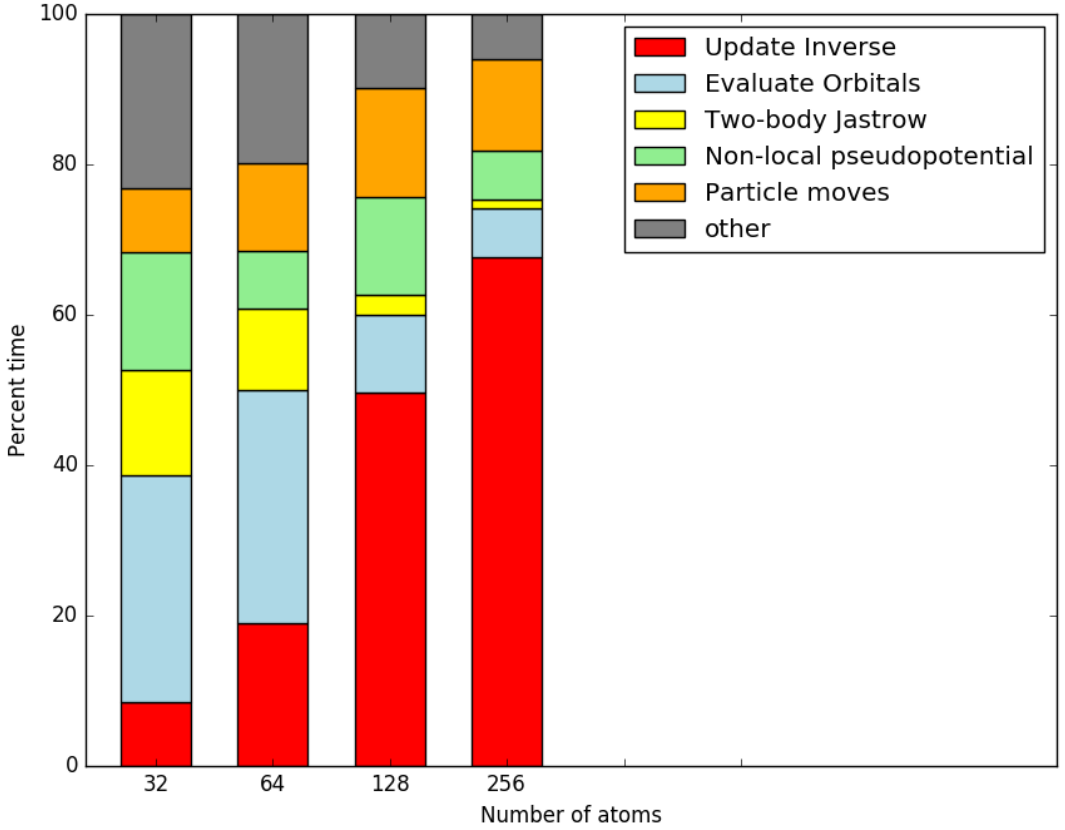
- Time to propagate a walker and example profile

Problem size (atoms)	Time per step
32	0.2 s
64	0.7 s
128	7.8 s
256	57 s
512	455 s
1024	3640 s



Profile confirms that matrix dominates

- Look at the profile for propagating a single walker as a function of system size
- Already for 256 atoms the $O(N^3)$ matrix operations dominate



Memory analysis tells a similar story

Sandia National Laboratories

- Look at a current version of QMCPACK
- Walker memory was $30*N^2*8$ Bytes per walker
- Wavefunction (splines) scales linearly with problem size for this problem

NiO Cell Size (atoms)	Per Walker	Wavefunction
128	135 MB	6.7 GB
256	540 MB	13.4 GB
512	2.11 GB	26.8 GB
1024	8.43 GB	53.6 GB

Need to change parallelization

Sandia National Laboratories

- Memory demands are growing faster than core count
- Time to propagate a walker is growing with the cube of the system size
- Equilibration dominates at large core counts

What can we do?

- In QMCPACK, the code is parallelized so that the walker is the minimum unit of parallel work
- For large calculations, target is currently one walker per thread
- In order to reduce population by a factor of 64, would have to be able to parallelize the walker over 64 processing elements
  - Can’t currently do this, but it is plausible for large enough calculations

Summary

Sandia National Laboratories

- QMC throughput is nearly linear with number of processing elements on world's largest computers
  - This does not translate into strong scaling!
- The warm-up phase of the Markov chain dominates when number of processors gets too large
- Small changes to the algorithm (variable time step) can help
- Even larger increases of efficiency are possible if a variable population is used
  - A finer level of parallelization should be implemented in QMC codes