Quantum Mechanical Material Simulations on Blue Gene/Q

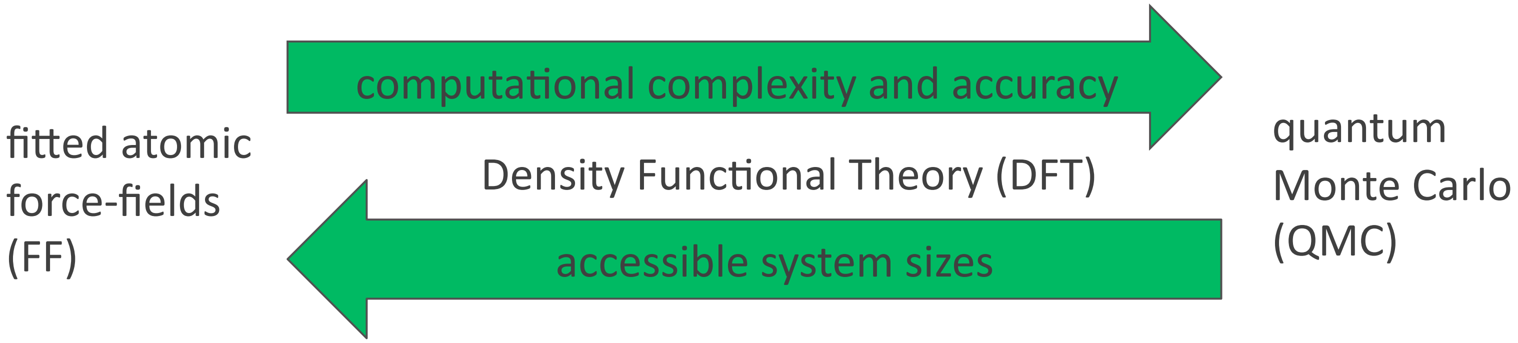
Nichols A. Romero

Assistant Computational Scientist

Leadership Computing Facility

Atomic simulations methods in a nutshell

Primarily for ground-state simulations in material science, chemistry, and condensed matter physics



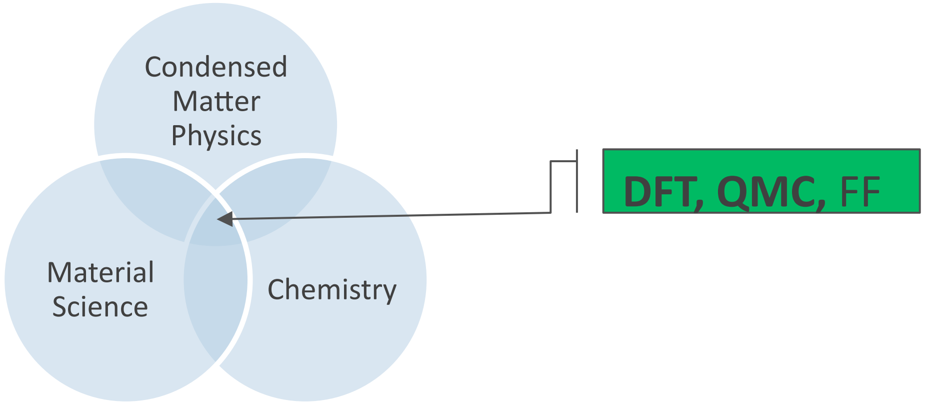
computational complexity and accuracy

fitted atomic force-fields (FF)

Density Functional Theory (DFT)

quantum Monte Carlo (QMC)

accessible system sizes



Condensed Matter Physics

Material Science

Chemistry

DFT, QMC, FF

Motifs in Quantum Mechanical Simulation

- Type 1A: Capability due to high accuracy in the level of theory
  - QMC $\mathcal{O}(N^{3\text{-}4})$
  - Hybrid DFT $\mathcal{O}(N^4)$
  - quantum chemistry $\mathcal{O}(N^{5\text{-}7})$
- Type 1S: Capability due to problem size (mostly pure DFT $\mathcal{O}(N^3)$)
  - Canonical DFT has a number issues, especially with non-local representations of Ψ
    - Dense diagonalization exhibits poor performance (see http://arxiv.org/pdf/1205.2107v1)
    - Instabilities in SCF algorithms (see Phys. Rev. B 64, 121101(R) (2001))
    - $O(N^3)$ “wall”
  - Reduced-scaling methods are needed in quantum mechanical approaches (mostly in DFT, but also many-body methods)
    - Fragment-type methods (GAMESS, LS3DF) or localization methods (CONQUEST, MADNESS, SIESTA, and many others)
    - Lots of progress, but also many remaining challenges (metals vs. insulators, precision, etc.)

Motifs in Quantum Mechanical Simulation (cont'd)

- Type 2: Loosely coupled, but not too loosely coupled
  - Might be doable in the cloud or on the grid, but communication between "images" needs to occur in a "lockstep" fashion
  - Nudge-elastic band calculations, Phonon spectrum, replica-exchange calculation, etc.
  - Even some aspects of quantum Monte Carlo
- Type 3: Capacity due to large parameter sweeps (i.e. High-throughput computing)
  - No control or data dependencies among calculations
  - Still not suitable for SETI@Home since each calculation may require many MPI tasks or lots of memory
  - Managing jobs with very different wall-clock times
  - Analysis of large data sets

What will you get out of this talk?

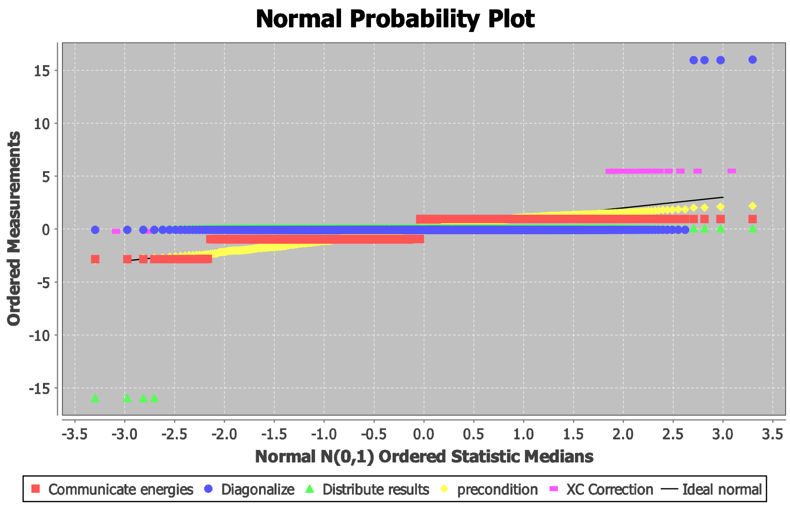
- It won’t answer:
  - Should I buy a Blue Gene/Q for my institution?
- It will begin to answer:
  - Is my electronic structure calculation well-suited for Blue Gene/Q?
- It will probably answer:
  - What types of electronic structure calculations are running well now?
  - What is involved in getting your code running on Blue Gene/Q?
    - Porting is easy
    - Using all the processing elements is easy
  - What is involved in getting your code optimized on Blue Gene/Q?
    - Getting good performance will require extra work
    - ALCF has a really excellent performance team (shameless self-promotion)

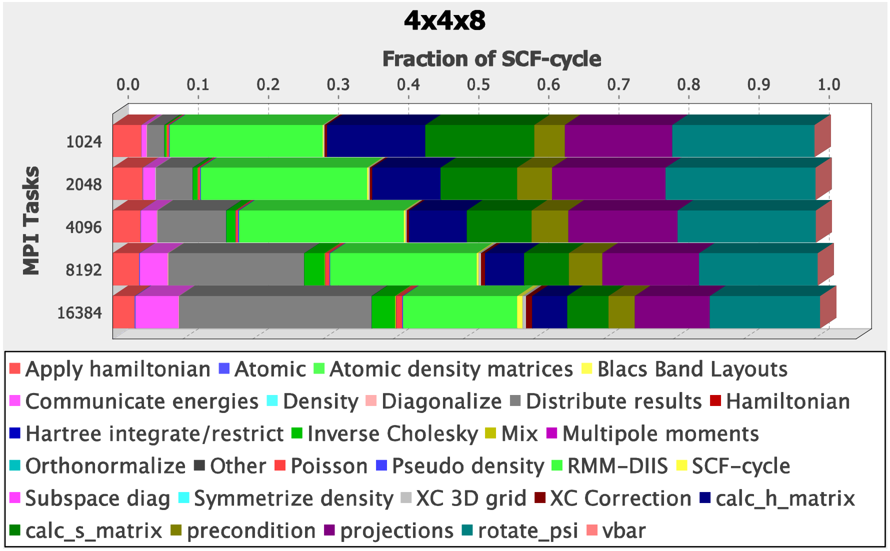
Quick code overview

- GPAW
  - Mira acceptance test code
  - Mainly used for Type 1S on Intrepid; largest calculations were 8 - 32-racks on 10,000 – 14,000 valence electrons
  - Consumed ~100 million core hours over the lifetime of Intrepid (includes INCITE/ALCC/ Discretionary)
  - Porting time on Blue Gene/Q ~ 1 day
- CPMD
  - INCITE/ALCC code
  - Used for Type 2, some Type 1S on Intrepid; largest Type 2 calculations were 8 - 32-racks
  - Will be used for Type 1A (hybrid DFT) on Mira
- QMCPACK
  - Early Science Program code
  - Not used on Intrepid for any scientific calculations due to insufficient memory per node
  - Will be used for Type 1A (QMC) on Mira
  - Porting time to Blue Gene/Q ~ 1 day, early issues with XL compiler, but resolved now

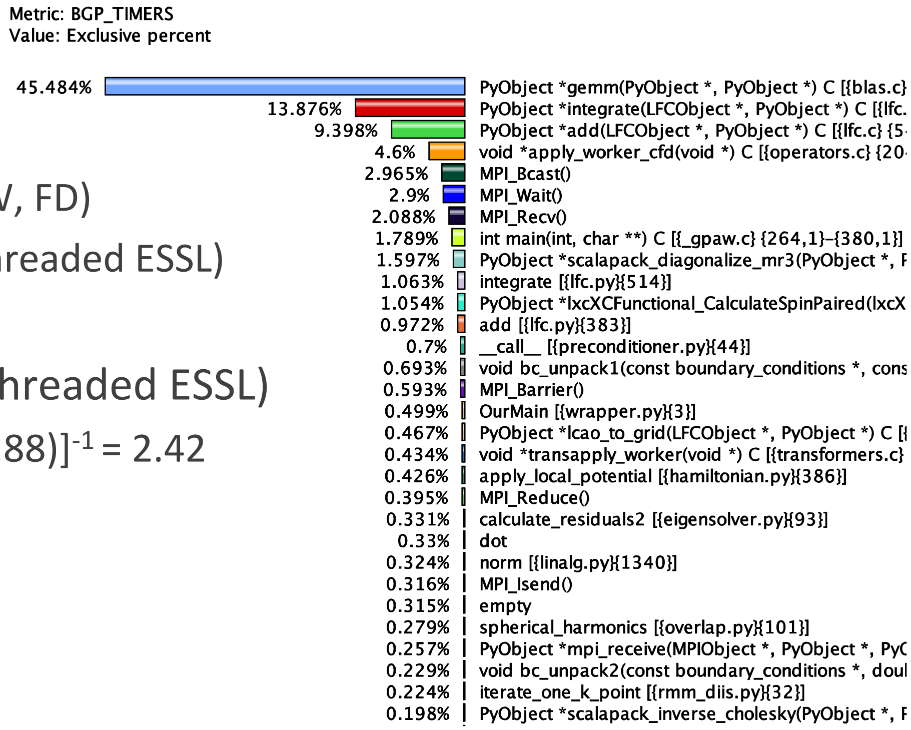
GPAW - performance on Blue Gene/P

- Density functional theory
  - Projector Augmented Wave (PAW) using finite-difference (FD) stencils on uniform real-space grid
  - MPI parallelism only, written in Python and C
- Algorithmic issues for Type 1S (> 10,000 electrons)
  - Large number of SCF cycles (>200) to obtain convergence (not shown).
  - Poor performance of dense diagonalization (inferred from dark grey on right figure)





GPAW - performance on Blue Gene/Q



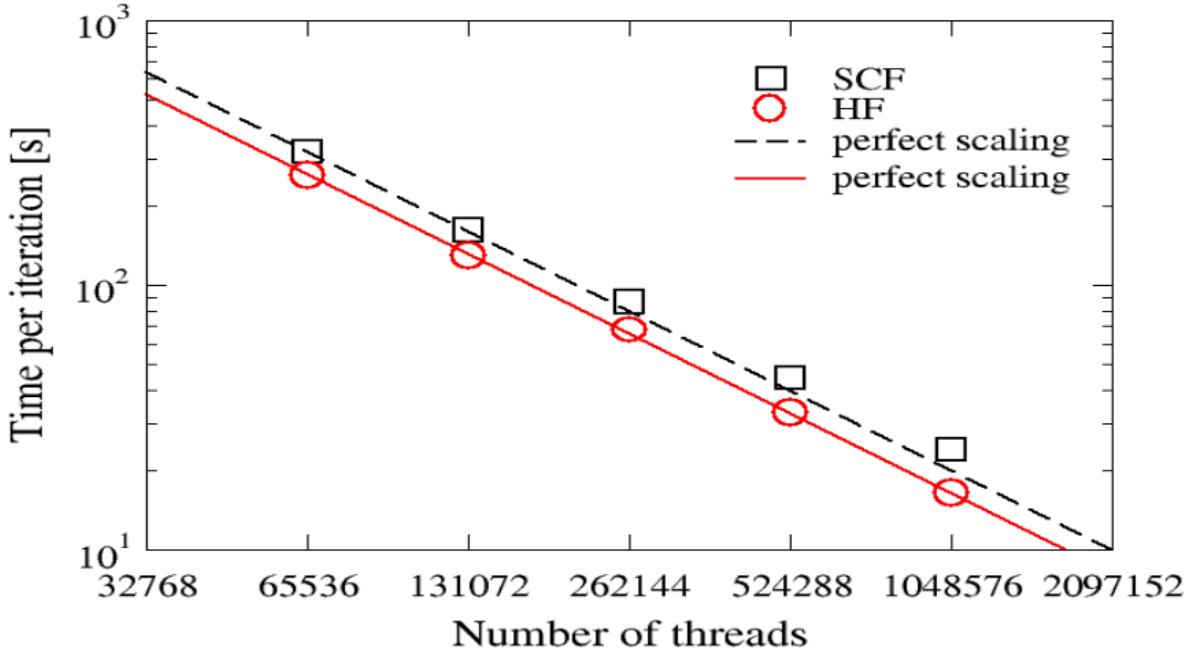
- Performance characteristics:
  - Simple flat profile
  - Representative of other DFT codes (PW, FD)
  - TAU profile from BG/P (4 rpn, single-threaded ESSL)
  - 45% DGEMM and 55% other stuff
- Core-to-core speed-up (16 rpn, multi-threaded ESSL)
  - Theoretical: $[(0.45/2.0/1.88) + (0.55/1.88)]^{-1} = 2.42$
    - Factor of 2.0 from quad-FPU
    - Factor of 1.88 from clock-rate
  - Actual: 1.57
  - Reason?
    - ESSL's DGEMM requires some tuning
      - 45% of peak on BG/P at 4 rpn single-threaded
      - 18% of peak on BG/Q at 16 and 32 rpn multi-threaded
    - Factor of 1.88 is not free:
      - Most applications do not get ideal instruction mix. GPAW gets 18% FPU / 82% FXU.
      - Two instruction streams per core required, one instruction stream per hardware thread

GPAW - moving forward

- Status on Blue Gene/Q
  - Possible to get 2X core-to-core speed-up with 16 rpn on some cases
    - Operate on very large chunk of matrix increases peak performance (40%) of ESSL's multi-threaded DGEMM
    - Leads to large messages size in parallel matrix multiply but network seems to keep up
  - Working with IBM to improve DGEMM
  - Core-to-core speed-up:
    - Speculation/potential : $[(0.45/2.0/1.88*2.0) + (0.55/1.88)]^{-1} = 2.88$
    - Assumptions:
      - >80% of peak for DGEMM on BG/Q,
      - Three remaining kernels can be threaded and have ideal instruction mix
  - Scalable to 8-racks but still suffers from the aforementioned algorithmic issues.
- Challenges ahead:
  - Reduced-scaling DFT code needed for Type 1S calculations
  - LCF staff (Jeff Hammond, Alvaro Vazquez) are working on MADNESS for the Type 1S calculations.

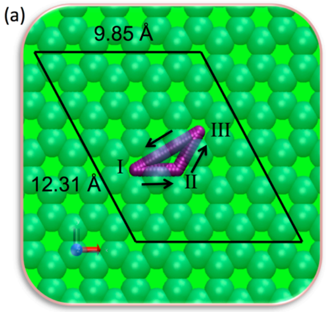
CPMD - performance on Blue Gene/Q

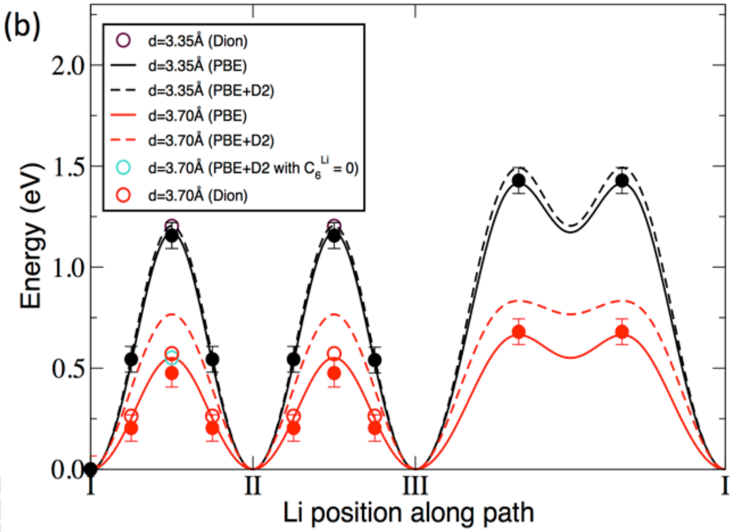
- Type 1A calculation (data from Curioni (IBM Zurich)):
  - hybrid DFT (PBE0) in a PW basis
  - Precondition conjugate gradient for minimization of wave function
  - 1000 atoms



QMCPACK - scientific motivation

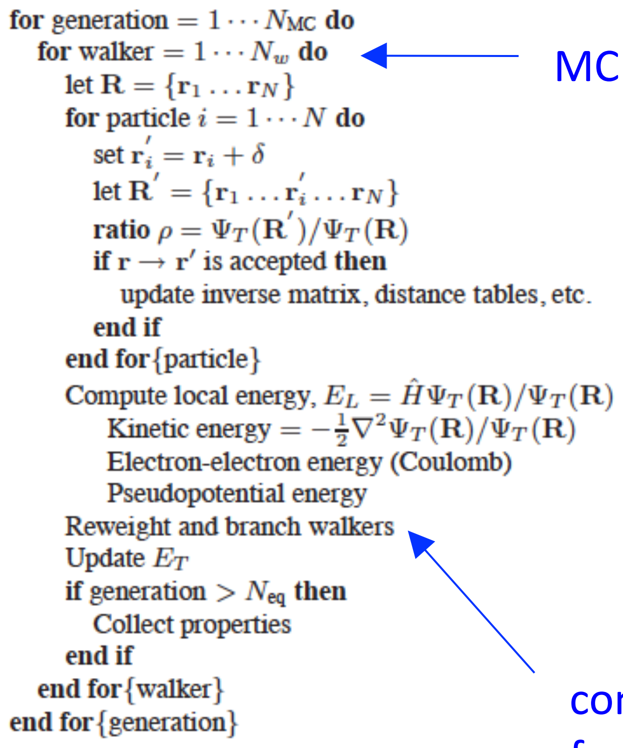
- quantum Monte Carlo (QMC)
  - Variational Monte Carlo (VMC) and Diffusion Monte Carlo (DMC).
    - Supports many basis: PW, B-splines, LCAO
    - Supports many boundary conditions
  - MPI and OpenMP parallelism, written in C++ and C.
  - Mainly used for Type 1A calculations, especially where DFT is problematic
    - vdW systems
    - Strong-correlated systems
- Li-ion diffusion and binding in graphitic battery anodes, by Ganesh, Kent, Reboredo & Kim





QMCPACK - performance on Blue Gene/P

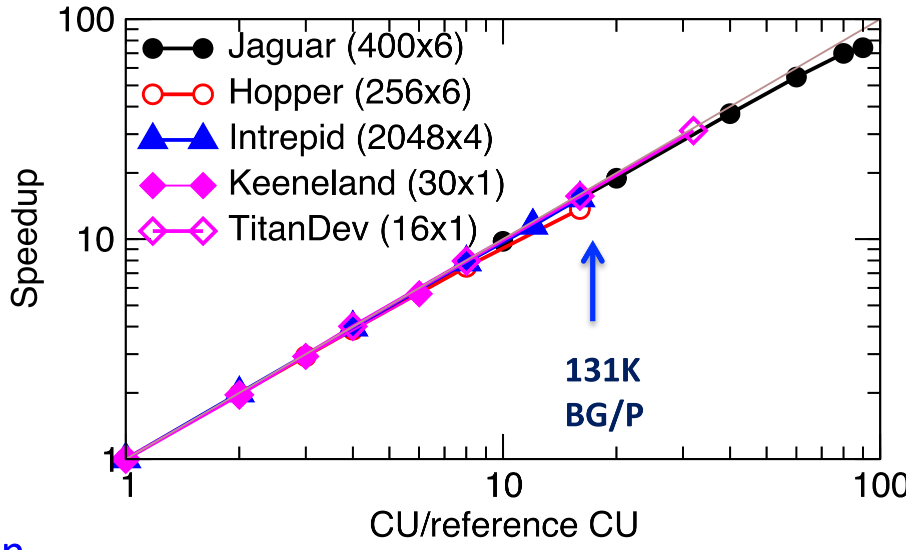
- Performance characteristics
  - Little communication, scales nearly perfectly
  - Easier to talk about node-to-node speed-up and thus realize full potential of BG/Q



```
for generation $= 1 \cdots N_{\mathrm{MC}}$ do
  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 $\rho = \Psi_T(\mathbf{R}^{'})/\Psi_T(\mathbf{R})$
      if $\mathbf{r} \rightarrow \mathbf{r}'$ is accepted then
        update inverse matrix, distance tables, etc.
      end if
    end for{particle}
    Compute local energy, $E_L = \hat{H}\Psi_T(\mathbf{R})/\Psi_T(\mathbf{R})$
       Kinetic energy $= -\frac{1}{2}\nabla^2\Psi_T(\mathbf{R})/\Psi_T(\mathbf{R})$
       Electron-electron energy (Coulomb)
       Pseudopotential energy
    Reweight and branch walkers
    Update $E_T$
    if generation $> N_{\mathrm{eq}}$ then
      Collect properties
    end if
  end for{walker}
end for{generation}
```

MC Walker is basic unit of parallelism

communication for DMC



131K BG/P

QMCPACK - performance on Blue Gene/Q

Eval_Z: Evaluation of spline

Eval_Z_VGH: Evaluation of spline derivatives

Symmetric_DTD: Distance table

- Original Esler Algorithm:

  4 nested loops (x,y,z,spline)

  No optimization of operations
- Killough Algorithm:

  2 nested loops (spline, x)

  2 unrolled loops (y,z)

  Optimization of operations
- Morozov Algorithm:

  4 nested loops (x,y,z,spline)

  Unrolling loops (4 times less iterations)
- Best Algorithms (GNU)

  - Eval_z: Morozov Algo.

  - Eval_z_vgh: Killough Algo.
- Best Algorithms (XL)

  - Eval_z: Morozov Algo.

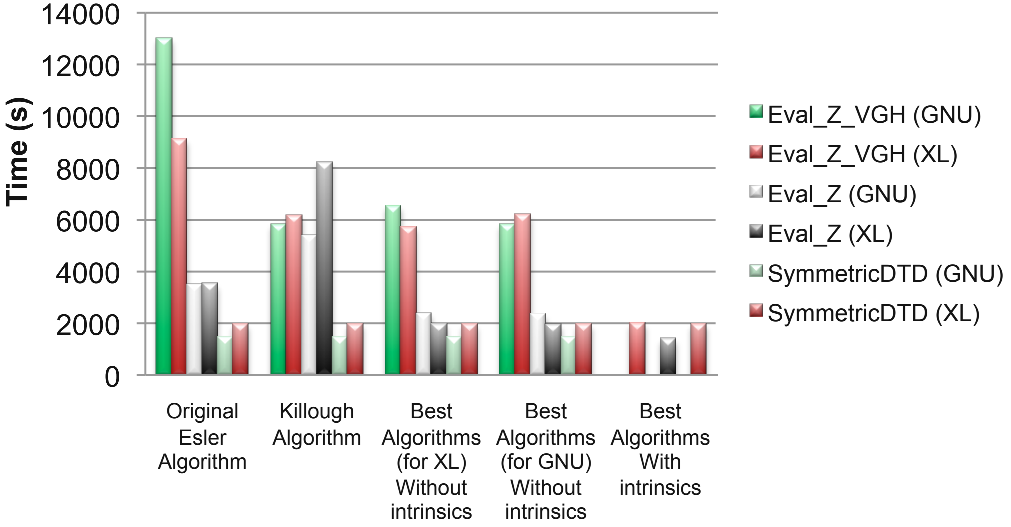
  - Eval_z_vgh: Benali Algo (Based on Morozov).
- Intrinsics (XL QPX)

  - Eval_z: Morozov Algo (with intrinsics).

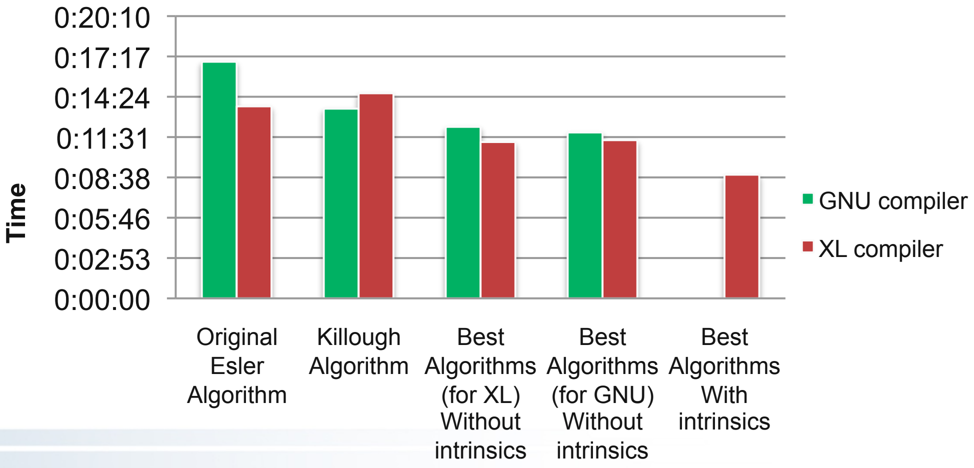
  - Eval_z_vgh: Benali Algo (Unrolling two loops

and adding intrinsics).

Kernels timing



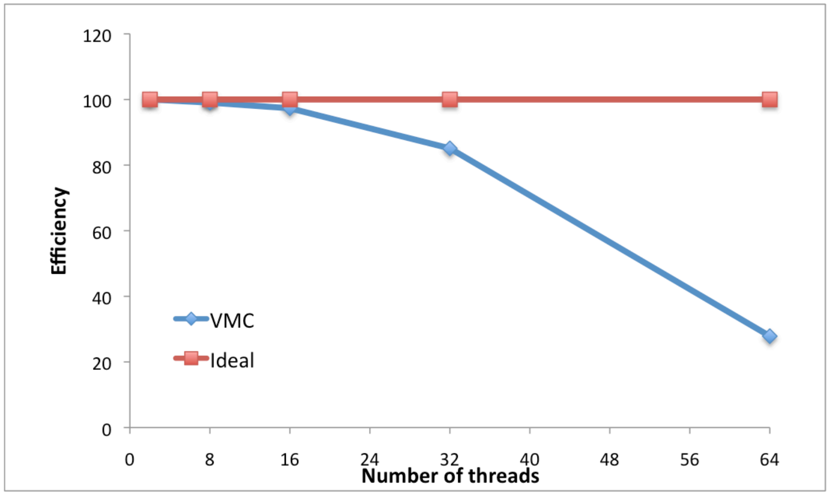
Total Application times



QMCPACK - performance on Blue Gene/Q (cont'd)

- Node-to-node speed-up (4 tpn on BG/P, 64 tpn on BG/Q):
  - Theoretical: 2.0*7.52 = 15.04
    - Factor of 2.0 from quad-FPU
    - Factor of 1.88 from clock-rate AND hardware threads
    - Factor of 4 from higher density of cores
  - Actual: 12.38
    - QPX speed-up of EVAL_Z_VGH by a factor of 3
    - 9.8 speed-up from OpenMP parallelism at 64 tpn compared to 4 tpn
    - Backward-port Blue Gene/P reference number based on best algorithm

Round-robin assignment of OpenMP threads to cores



QMCPACK - moving forward

- Status on Blue Gene/Q:
  - Science runs will begin soon.
  - Working on single precision version of Einspline
    - Needed for time-to-solution to be competitive with GPU version
    - Will reduce memory requirements for storing wave function, allows simulation of larger systems
  - We will scale QMCPACK to all 48-racks of Mira as proof-of-principle in the near future
- Challenges:
  - Percentage of peak is very low ~ 3%
    - Many kernels bandwidth limited (i.e. QPX will not help)
  - Large QMC problems (> 1000 valence electrons) will require nested OpenMP parallelism
  - More compact representations of the wave function needed
    - LAPW (in progress)
    - Distribute/tessellate wave function without a performance hit
  - Trial wave function currently optimized with serial LAPACK (general non-symmetric EVP)

Acknowledgements

- Technical University of Denmark (DTU), Center for Atomic-scale Molecular Design (CAMd)
  - Jens Jørgen Mortensen (lead developer of GPAW), Marcin Dulak , Ask Larsen (now at Universidad del País Vasco), Christian Glinsvad
- CSC, the Finnish IT Center for Science, Ltd.
  - Jussi Enkovaara – support staff
- Argonne National Laboratory
  - Anouar Benali, Vitali Morozov, Lee Killough (now at Appro)
- Oak Ridge National Laboratory:
  - Jeongnim Kim (lead developer of QMCPACK)
- University of Oregon, Performance Research Laboratory
  - Sameer Shende – TAU Performance System™
- IBM (Rochester, Watson, Sweden, Toronto)
  - Paul Coffman, Bob Walkup, Nils Smeds, Basil Kenneth, Wang Chen

This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.