ANL/ALCF/ESP-13/2

Materials Design and Discovery: Catalysis and Energy Storage (Mira Early Science Program Final Technical Report)

ALCF-2 Early Science Program Technical Report

Argonne Leadership Computing Facility

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Availability of This Report
This report is available, at no cost, at http://www.osti.gov/bridge. It is also available on paper to the U.S. Department of Energy and its contractors, for a processing fee, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone (865) 576-8401
fax (865) 576-5728
reports@adonis.osti.gov

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

ANL/ALCF/ESP-13/2

Materials Design and Discovery: Catalysis and Energy Storage (Mira Early Science Program Final Technical Report)

ALCF-2 Early Science Program Technical Report

prepared by
Anouar Benali, Nichols A. Romero
Argonne Leadership Computing Facility, Argonne National Laboratory

May 7, 2013

Mira Early Science Program
Final Technical Report
Materials Design and Discovery: Catalysis and Energy Storage

Anouar Benali and Nichols A. Romero
Leadership Computing Facility, Argonne National Laboratory

April 3, 2013

Introduction

The investigation and design of new classes of materials for energy and catalysis requires a multi-facetted approach to simulation. Multiple methods are needed to study materials on the length scale 0.1 nm - 10 nm. For simulations where the atomic (and electronic) degrees of freedom are relevant, the methods of choice in the surface science, condensed matter physics, and material science communities are classical molecular dynamics (CMD), Density Functional Theory (DFT), and quantum Monte Carlo (QMC).

The original scope for this Early Science Program (ESP) project was to perform fast-accurate DFT calculations on materials for energy and catalysis using the GPAW[1, 2, 3] code on Blue Gene/Q. The types of calculations included significantly reduced time-to-solution on systems sizes accessible on Blue Gene/P ($\sim$10, 0000 valence electrons), but also systems which were were at least a factor of two larger ($\sim$20, 000 valence electrons). GPAW is a real-space DFT code using the projector augemented wave (PAW) method. DFT calculations on Blue Gene/P were executed on over >100, 000 cores using GPAW; thus it was consider a success on Mira's predecessor system, Intrepid.

One of the co-PIs (NAR), determined that the work necessary to allow the GPAW code for these aformentioned types of calculations could not be accomplish within the time frame of the ESP. This was not simply due to human time required to implement OpenMP parallelism, but also from intrinsic algorithmic limitations in supporting libraries, most notably ScaLAPACK. Additionally, the return-on-investment on $\mathcal{O}(N^3)$ DFT code has become some what tenable at best. NAR argues that what is really needed in preparation for exascale computing and to enable high-fidelity materials research is robust reduced scaling, $\mathcal{O}(N)$ or $\mathcal{O}(N)\log(N)$, DFT approaches.

Thus, in aggreement with the other co-PIs, the decision was made to purse QMC as a complimentary method on Mira since it would be vaulable for the scientific community and could easily leverage the massive parallelism that would be provided by the Blue Gene/Q. We note that the two other atomic-scale methods mentioned here, CMD and DFT, are being pursued by other ESP projects on Mira. The remainder of this report will focus on our progress on QMC.

Beyond Density Function Theory

DFT provides *qualitative* accuracy for many well-behaved systems but lacks *quantitative* accuracy for most materials. One example where DFT consistently performs poorly is van der Waals dominated systems; additionally, chemical accuracy, generally considered to be 1 kcal/mol (=4 kJ/mol or 1 meV) cannot be achieved. This accuracy can only be achieved by an accurate description of the electronic correlations of the system and therefore making it difficult to use mean field methods, such as DFT or Hartree Fock (HF).

Accurate many-body methods, such as Coupled Cluster (CC), provides accurate estimates of the energies by solving the many body Schroedinger equation, but becomes rapidly computationally intractable as the number of electrons increases, scaling as poorly as N^7. QMC, within the variational Monte Carlo (VMC) and diffusion Monte Carlo (DMC) methods are "stochastic approaches for evaluating quantum mechanical expectation values with many-body Hamiltonians and wave functions. [..] The main attraction of these methods is that the computational cost scales as some reasonable power (normally from N^2 to N^4) of the number of particles N. This scaling makes it possible to deal with hundreds or even thousands of particles, allowing applications to condensed matter."[4]

We therefore solve the Schröedinger equation with the manybody Hamiltonian. QMC formalism is the usual Monte Carlo (MC) formalism in the sense that it solves multi-dimensional integrals by sampling randomly the space and allowing the system to evolve in the imaginary time using MC steps. Only moves that lower the energy are accepted. The obtained total energy comes at a cost of a variance and an error bar due to its stochastic nature.

$$\sigma^2 = \left\langle E_T^2 \right\rangle - \left\langle E_T \right\rangle^2, \qquad \delta = \frac{\sigma}{\sqrt{M}} \tag{1}$$

It becomes evident that to reduce the error bar, one should run the simulation longer or simply increase the number of samples, which suits particularly well large supercomputers systems.

A good sampling requires starting close to the right answer. In order to do so, we use a trial wave function that is created from a Slater determinant of single-particle orbitals (obtained from a previous DFT calculation) in combination with Slater-Jastrow parameter that explicty incorporate electron correlation effects.

Splines Single-particle orbitals (SPO) are a set of functions in (R^3), one function per orbital that describes its quantum state. During simulation, a walker (R^{3N}) samples spatial regions for the many-body state at that point, requiring an evaluation of all orbital functions. The Slater determinant of single-particle orbitals depends on the choice of basis set (molecular orbitals, plane waves etc...). For the class of materials we are interested in, the most practical choice is the use of plane waves. The B-pline approximation in QMC reports significant reduction of time of calculation while maintaining plane wave level of precision. The mesh size in the X, Y, and Z dimensions determines the accuracy of the representation. When a point in space is evaluated, a minicube of coefficients (64 coefficients) surrounding that point is required to interpolate its value.

Twists Generating a QMC trial wavefunction can be accomplished by generating a DFT wavefunction that contains all of the k-points necessary to express the many-body wavefunction with the different boundary conditions in the QMC simulation cell. Then the resulting QMC calculations will be performed with all of these different boundary conditions. Practically speaking, one needs to specify the boundary conditions for the QMC calculations using a k-point grid of $n \times n \times n$ twists. This also has the effect to change the type of the wave function from real when no twists are used, to complex when they are present.

Workflow

A common workflow for QMC consists on generating a trial Slater-Jastrow wave function, running a VMC optimization and finally running a DMC to reach the desired accuracy. VMC treats the square of the wave function as the probability distribution on which to do Monte Carlo. This means the form of the wave function limits the minimum energy you can reach in VMC. DMC, uses a branching, birth/death process based on the imaginary-time version of the Schröedinger equation to guide the random walkers in their random walk. Its minimum energy is limited only by the nodal surface, derived from the trial wave function, which prevents random walkers from moving between different-signed areas of the configuration space.

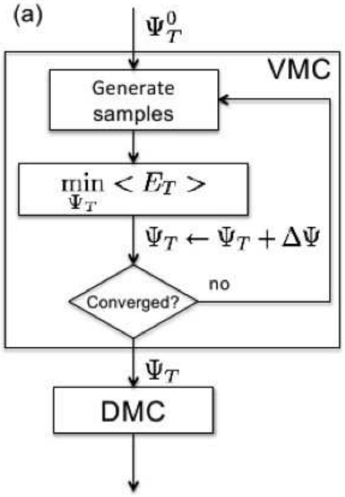

Figure 1: QMC workflow chart.

QMCPACK on MIRA

We use the QMCPACK[5, 6] simulation package for this project. The code was developed starting 2002 by the Ceperley Group at UIUC. Since then, the community of developers led by Dr. Jeongnim Kim (ORNL) has grown and spread amongst different institutions and National Laboratories.

QMCPACK is a heavily templated C++ code using all aspects of object-oriented coding, STL libraries and MPI and OpenMP for communication. This should have made it ready for porting on BGQ, however, as we experienced with our early access, many C++ standard were not compatible between different compilers and the use of OpenMP led to some serious problems (race conditions and thread unsafe regions) that had to be addressed before going further.

Once we were certain that the code was producing the right answers we proceeded to optimizing it.

Profiling We used HPM and GPROF as main profiling tools. Most of the systems of importance to us use twists and therefore we focused on the complex-valued wave functions which exercises the complex-valed code paths in QMCPACK.
Running GPROF showed that 71% of the application time was spent in

the spline evaluation of the wavefunction. This time was spread between two functions, `Eval-Z` evaluating the spline and `Eval-Z-VGH` evaluating the spline, the gradient and the hessian. (These functions exist also for the real-valued type in two versions, a double precision and a single precision version). Computationally, each function consists on 4 nested loops ($4{\times}4{\times}4{\times}N$) where $4{\times}4{\times}4$ corresponds to the number of coefficients in the minicube and N is the number of orbitals in the system, as described in the splines section.
In order to optimize the code we modified these two types of algorithms (for complex type and double/single precision type) using two general algorithms, then adding a layer of QPX and finally prefetching when it was possible.

Algo M. Algorithm M. consists of fusing the $4{\times}4{\times}4$ loops and unrolling the inner loop with a stride of 8. For Mira, we used QPX instructions to manually load and store data after using the fused multiply-add functions. As a last step we added prefetching on the spline-only evaluation function to improve memory management.

Algo B. Algorithm B. consists on reversing the order of the loops to $N{\times}4$ and unrolling the other loops. The mathematical expression of the problem is modified decreasing substantially the number of floating-point operations. For Mira we managed to use a similar algorithm to replace most of the instructions by QPX functions but were not able to benefit from prefetching.

As said previously, according to the type of system one can study (solid, nanocluster, molecules etc...) the use of twists will make the wavefunction either complex or real. This will exert 2 different parts of the code, a complex-valued type and a real-valued type (with a double precision and a single precision version). We applied the same methodology to optimize the spline evaluation functions and show the results in table-1. Results show that Algo. M is more efficient for the simple evaluation of the spline, while Algo B. is more efficient when spline, gradient and hessian are evaluated. The increased latency in Algo B. is hidden by the very important decrease of the total number of instructions (see table-2. However, Algo M doesn't have a better management of memory access and does not reduce as effectively the number of instructions, but for a smaller function, it is far more efficient than Algo. B.

When profiling the same problem with the new optimized algorithms we

Speed up	Eval-Z	Eval-D	Eval-S
Algo. B	0.38	0.81	0.39
Algo. M	2.48	0.91	1.02
Algo. (X) with QPX	3.94 (M)	1.08 (M)	1.26 (M)
QPX + Prefetch.	4.5	1.23	1.81
Speed up	Eval-Z-VGH	Eval-D-VGH	Eval-S-VGH
Algo. B	1.59	0.93	1.62
Algo. M	2.15	1.01	0.95
Algo. (X) with QPX	7.62 (B)	1.58 (B)	1.31 (B)

Table 1: QMCPACK speed up for three different types of wave functions exerting the complex-valued part (`Eval-Z`, `Eval-Z-VGH`),the real part (double precision) (`Eval-D` and `Eval-D-VGH`) and its single precision version (`Eval-S` and `Eval-S-VGH`).

see that the percentage of peak, the memory management and the number of instructions per cycle completed per core dropped significantly (see table-2). However, the number of instructions has decreased significantly which hides the latency in the memory management and most of all, the time spent on the spline evaluation and the time to solution was reduced by a factor 2.67 (see Fig-2). We selected the most efficient algorithms and applied them to QMCPACK. Results are shown in Fig-2.

Conclusion

Quantum Monte Carlo algorithms (due to their stochastic nature and the independence between samples) can benefits greatly from massively parallel supercomputers. The large number of cores on Mira can be leverage by QMC to study very large systems at chemical accuracy in extremely short time by using a very large number of samples, corresponding to a very large number of cores. The use of QPX and prefetching improved substantially the time to solution making the code even more efficient with a 2.67 speedup. Working with QMCPACK on BGQ Mira allows us to study a larger spectrum

Profiling	Original Version	Mira Optimized
Time spent on Spline evaluation	70.97 (%)	22.33 (%)
Percentage of Peak	6.55 %	5.33%
All XU Instructions (in Billion)	27,644	8,581
All AXU Instructions (in Billion)	22,786	4,896
FP Operations (in Billion)	43,043	13,017
Instructions/cycle completed/core	0.6138	0.4417
L1 d-cache hits	94.03 (%)	88.60 (%)
L1P buffer hits	5.36(%)	5.92 (%)
L2 cache hits	0.35 (%)	4.50 (%)
DDR hits	0.26 (%)	0.98 (%)

Table 2: Performance Comparison between Original version of QMCPACK and Mira modified version of QMCPACK

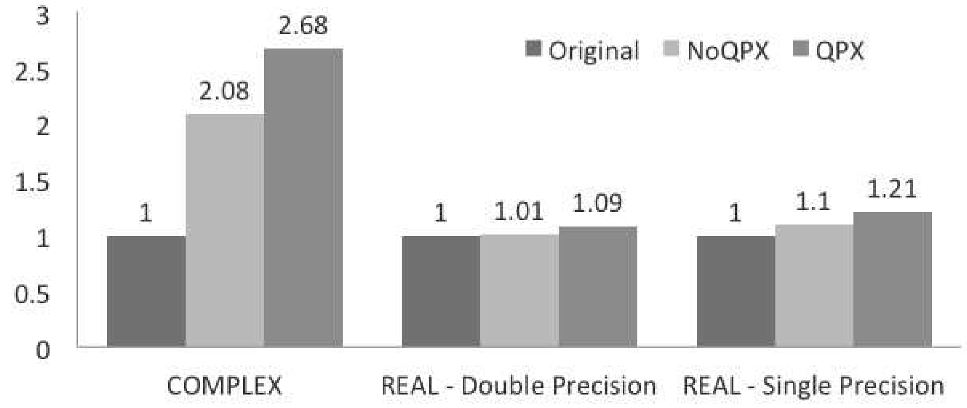

Figure 2: QMCPACK Speed up using compared to the original version using our cross platform algorithm (NoQPX) and QPX for all three types of wavefunctions.

of materials at the chemical accuracy which is a great achievement. Many applications, from material design to biochemistry are being investigated and should soon be submitted to high impact journals.

The work on QMCPACK, specially on Mira is far from being over. As one can notice on this report, most of the efforts were focused on porting

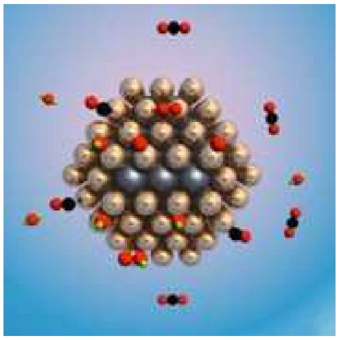

Figure 3: Pt solids and Nanoclusters for Catalysis

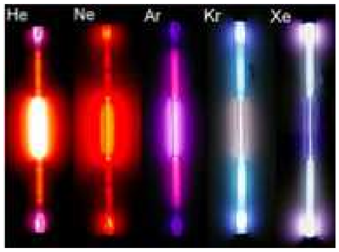

Figure 4: Ar, Kr and Xe Solid (Simulation of Van der Waals dominated solids

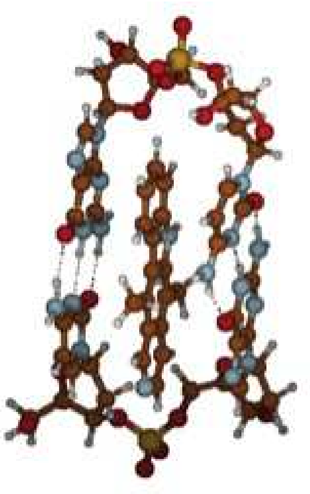

Figure 5: Molecule of Ellipticine with DNA fragments

the code to BGQ, optimizing the main kernel using QPX and ”prefetching”. However very little has been done in implementing nested open OpenMp (hybrid paralellization) which in theory, could reduce the time to solution by a factor 4. Hopefully this next step will be undertaken in the near future.

Bibliography

[1] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen, "Real-space grid implementation of the projector augmented wave method," *Phys. Rev. B*, vol. 71, p. 035109, Jan 2005.

[2] J. Enkovaara, C. Rostgaard, J. J. Mortensen, J. Chen, M. Dułak, L. Ferrighi, J. Gavnholt, C. Glinsvad, V. Haikola, H. A. Hansen, H. H. Kristoffersen, M. Kuisma, A. H. Larsen, L. Lehtovaara, M. Ljungberg, O. Lopez-Acevedo, P. G. Moses, J. Ojanen, T. Olsen, V. Petzold, N. A. Romero, J. Stausholm-Möller, M. Strange, G. A. Tritsaris, M. Vanin, M. Walter, B. Hammer, H. Häkkinen, G. K. H. Madsen, R. M. Nieminen, J. K. Nørskov, M. Puska, T. T. Rantala, J. Schiøtz, K. S. Thygesen, and K. W. Jacobsen, "Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method," *J. Phys.: Condens. Matter*, vol. 22, p. 253202, 2010.

[3] J. Enkovaara, N. A. Romero, S. Shende, and J. J. Mortensen, "GPAW - massively parallel electronic structure calculations with Python-based software," *Procedia Computer Science (2011)*, vol. 4, pp. 17–25, 2011.

[4] Needs, M. D. Towler, N. D. Drummond, and P. L. Ríos, "Continuum variational and diffusion quantum monte carlo calculations," *Journal of Physics: Condensed Matter*, vol. 22, no. 2, p. 023201, 2010.

[5] J. Kim, K. Esler, J. McMinis, and D. M. Ceperley, "QMCPACK simulation suite." unpublished.

[6] J. Kim, K. Esler, J. McMinis, and D. M. Ceperley, "Quantum monte carlo algorithms: making most of large-scale multi/many-core clusters," Conference Series, (Chattanooga, TN), Scientific Discovery through Advanced Computing (SciDac), J. of Physics, Jun 2010.

Argonne Leadership Computing Facility
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 240
Argonne, IL 60439

www.anl.gov

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC