AFRL-AFOSR-VA-TR-2019-0177

BRI A Co-Design Approach for Advances in Software and Hardware

Mark Gordon
IOWA STATE UNIVERSITY

01/03/2019
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTB1
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
02-07-2019	Final Performance	30 Sep 2012 to 30 Nov 2018

4. TITLE AND SUBTITLE
BRI A Co-Design Approach for Advances in Software and Hardware

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-12-1-0476

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Mark Gordon

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
IOWA STATE UNIVERSITY
1350 BEARDSHEAR HALL
AMES, IA 50011 US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTB1

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-VA-TR-2019-0177

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Investigations were performed on energy savings for GAMESS and HPC Benchmarks on various Intel processor architectures and their analysis using the Empirical Model Decomposition (EMD) method. A novel energy-saving strategy, which takes into account the DVFS granularity of the different processor types while choosing a ceiling frequency (or power limiting level) has been proposed and studied. Under the auspices of the AFOSR BRI grant, we conducted multiple studies that explore the viability of multiple ARMv8 processor architectures in terms of performance and energy efficiency.

15. SUBJECT TERMS
Co-design, software, hardware, high performance computing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			POMRENKE, GERNOT
Unclassified	Unclassified	Unclassified	UU		19b. TELEPHONE NUMBER *(Include area code)* 703-696-8426

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

DISTRIBUTION A: Distribution approved for public release.

Results from AFOSR Award No. FA9550-12-1-0476
12/1/2014-11/30/2018
PI: Mark S. Gordon
Iowa State University
Program Officer: Gernot Pomrenke

Accomplishments.

Energy savings. Investigations were performed on energy savings for GAMESS and HPC Benchmarks on various Intel processor architectures and their analysis using the Empirical Model Decomposition (EMD) method. The findings and developments were as follows:

● Runtime system is proposed that (1) predicts the micro-instructions executed at different processor and memory frequencies based only on a small number system parameters, (2) predicts the power consumption based on the instantaneous power consumption using the Intel RAPL technology, and (3) select the appropriate processor–memory frequency pairs that minimize total energy consumption while satisfying the performance-loss constraint. The runtime system operates on timeslices and uses history-window approach to predict application future behavior, and uses an adaptive mechanism to adjust dynamically the memory and compute-intensity of an application. The system was validated on the SPEC CPU TM 2006 and NAS parallel benchmarks as well as on the sparse linear algebra application pARMS, all of which contain both memory- and compute-intensive codes. It saved a maximum of 22% energy with a small average performance loss of 4.8%, see Fig. 1 (left) and (right), respectively.

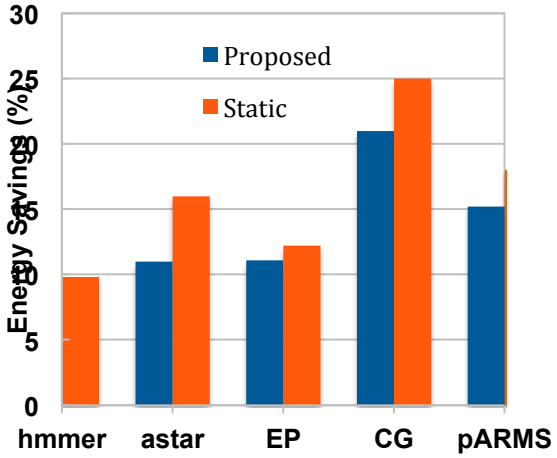


Figure 1: Performance loss (left) and energy savings (right) for the SPEC, NASA parallel benchmarks and pARMS benchmarks under the proposed runtime and and static strategies.

● Dynamic voltage and frequency scaling (DVFS) granularity has been determined for three recent Intel processor types and observed that the absence of the per-core granularity may hamper energy savings, as shown in the following Table 1.

Table 1. GAMESS energy saving as percentage of the maximum frequency for three frequency scaling mechanisms in a node: *native* (per-core frequency change) *allmean* (frequency average among all cores in a socket) and *emulated* as socket-level (Bolt) and as twin-core (Dynamo).

Platform	native	allmean	emulated
Bolt	9	-15	0
Dynamo	9	-19.2	9

● A novel energy-saving strategy, which takes into account the DVFS granularity of different processor types while choosing a ceiling frequency (or power limiting level) has been proposed and studied. The strategy was validated on two GAMESS inputs operating under different power budgets and it delivered performance within 2% of the best possible performance. Moreover, it was observed that the ratio of the performance loss to power reduction remained low in GAMESS: Power reduction of as much as 40% resulted in performance loss of about 9% (see Fig.2).

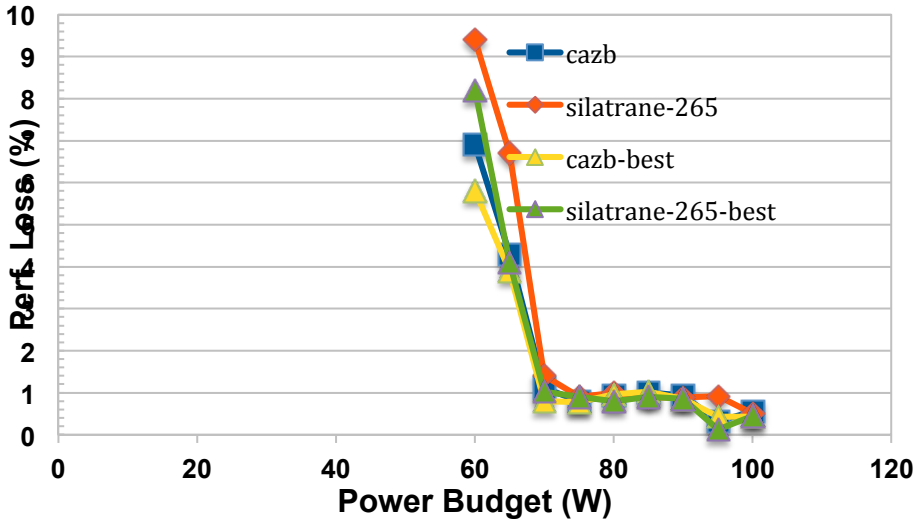


Figure 2: Performance loss for the cazb and Silatrane-265 GAMESS inputs with different power budgets. The data suffixed with '-best' shows the lowest possible performance for a power budget.

● The Empirical Mode Decomposition and Hilbert-Huang Transform (EMD/HHT) analysis method has been applied to the power traces collected on several hardware platforms, featuring KNC and KNL accelerators for GAMESS. Using EMD/HHT analysis, hardware utilization can be broadly classified based on the overall intensity of the resulting histogram. It was shown that varying clock-rate or the number of cores impacts the entire time-frequency domain of the EMD/HHT analysis (See Fig. 3.)

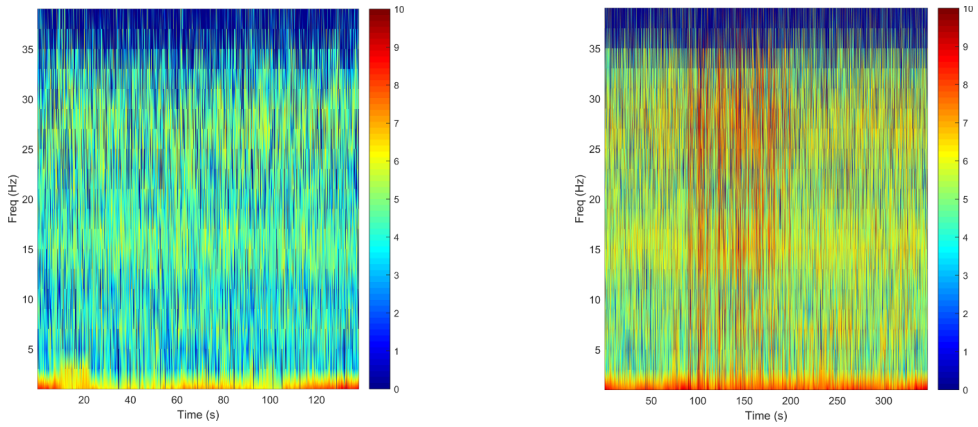


Figure 3: Comparison of EMD/HHT histograms generated for power traces collected for GAMESS on a KNL system while varying the number of cores or clock-rate: (left) 32 cores, (right) 63 cores.

We formulated a new metric called iso-power-efficiency that describes scaling under a power budget. The idea is to scale up the problem size with increasing power budget, rather than just with increasing numbers of processors. For many applications, speedup saturates and parallel efficiency decreases if the problem size is held fixed while increasing the number of processors (the form of scaling known as strong scaling). For some problems, it is possible to maintain a fixed parallel efficiency by increasing both the problem size and the number of processing elements. The rate at which the problem size must increase to maintain constant efficiency for a given rate of increase of the number of processors is given by the iso-efficiency function. We have developed a new scalability function called iso-power-efficiency that determines the rate at which the problem size must increase to maintain constant efficiency for a given rate of increase of the application's power budget. For a given power budget, an application can choose to use a larger number of processors running at lower power. Speedup can often be obtained within a given power budget by such overprovisioning. Deriving the iso-power-efficiency function for a given problem involves 1) determining optimal configurations for problem instance/power budget pairs, and 2) expressing the parallel overhead as a function of problem size and power budget. We have shown that the rate of growth required for problem size can be lower with iso-power-efficiency than with iso-efficiency, thus yielding better scalability, and we have developed a regression modeling methodology for fitting observed execution data to an iso-power-efficiency function. This work has led to a publication and presentation at the Workshop on High-Performance, Power-Aware Computing (HPPAC 2015).

Soft-Fault Models. Investigations of soft-fault models in hybrid parallel asynchronous iterative methods Specifically, we investigated impacts of undetected soft faults on an

asynchronous iterative method, and to compare and contrast several techniques for simulating the occurrence of and recovery from a fault. The data shows that the two numerical soft-fault models (shuffle-based and perturbation-based, respectively, SBSFM and PBSFM), considered in this research, produce more consistently than a "bit-flip" model bad enough behavior to test a variety of recovery strategies, such as those based on partial checkpointing (cf. Fig. 4 left and right). Results were presented for asynchronous iterative methods, implemented in a hybrid parallel fashion, using OpenMP and MP.

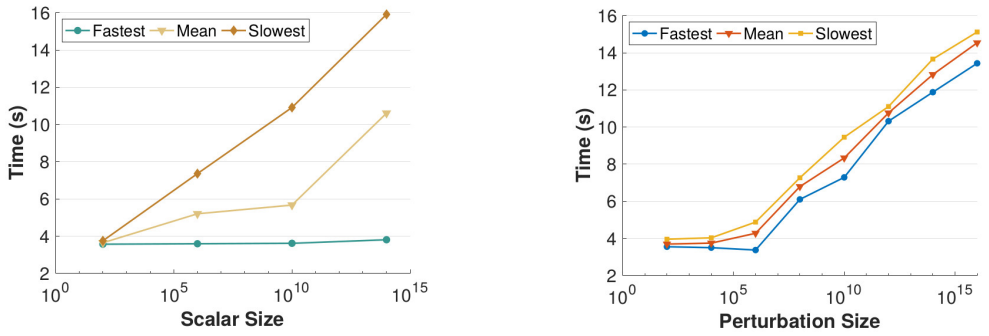


Figure 4: Effect of fault recovery with the SBSFM (left) and PBSFM (right) using partial checkpointing. All the runs in PBSFM are closer to the mean than those in SBSFM are.

Emerging HPC Architectures. Many in the HPC community are convinced that the 64-bit ARMv8 platform specifically due to its improved double-precision and SIMD support over previous ARM generations, will figure prominently into the set of solutions that allow continued progress in the scientific and engineering endeavors that rely on the massive scale of computation offered by large HPC systems. To that end, it is critical to understand the performance and energy efficiency of the ARM architecture in the context of well-established architectures deployed in HPC. Under the auspices of the CoDAASH grant, EP Analytics (in close collaboration with Ames Laboratory) conducted multiple studies that explore the viability of multiple ARMv8 processor architectures in terms of performance and energy efficiency. The highlights of previous studies are provided below:

1. We presented an analysis of the performance, parallel scalability and energy efficiency of GAMESS on a commercially available HP Moonshot 64-bit ARM cluster that consisted of Applied Micro XGene 1 processors. The performance and energy efficiency metrics were also compared to a conventional x86 Intel Ivy Bridge system. A 2:1 Moonshot core to Ivy Bridge core performance ratio was observed for GAMESS calculations. Doubling the number of cores to complete the execution faster on the 64-bit ARM cluster led to better energy efficiency compared to the Ivy Bridge system.
2. We presented a comprehensive study of the performance, power and energy consumption of the Applied Micro XGene 1, the first commercially available 64-bit ARMv8 platform, for HPC workloads. The study included a detailed comparison of the X-Gene to three other architectural design points common in HPC systems. Across these platforms, careful measurements across 400+ workloads were performed, covering different application domains, parallelization models, floating-point precision models and memory intensities.

The study found that XGene has an average of 1.2× better energy consumption than an Intel Sandy Bridge, while the Sandy Bridge is an average of 2.3× faster than XGene.

Building on these previous studies, evaluation of the emerging many-core architectures – Cavium ARMv8 ThunderX, and Intel's Knights Landing (KNL) processors – was conducted. To continue to evaluate performance and energy efficiency of different ARMv8 offerings, XGene 1 was also included in the study. Since many-core architectural design mandate efficient usage of the available cores via thread-level parallelism, GAMESS calculations that use hybrid MPI+OpenMP parallel model were included in the study. Optimal build-time (e.g., compilers and math libraries) and run-time configurations determined empirically through parameter sweep were used for final power and performance measurements. For example, run-time configurations for threaded GAMESS calculations consist of number of ranks per node and number of OpenMP threads per rank, both of which were determined empirically to select the best performing configuration. Figure 5 presents the results of this study. In the figure, performance (time-to-solution), power (averaged over entire run) and energy (product of average power and execution time) metrics for KNL and XGene 1 systems are normalized using the respective measurements for ThunderX system. The results demonstrate that ARMv8 systems are better in terms of power-draw and energy efficiency perspectives. ThunderX outperforms KNL in half the benchmarking calculations, while maintaining leads in power draw and energy on all the calculations.

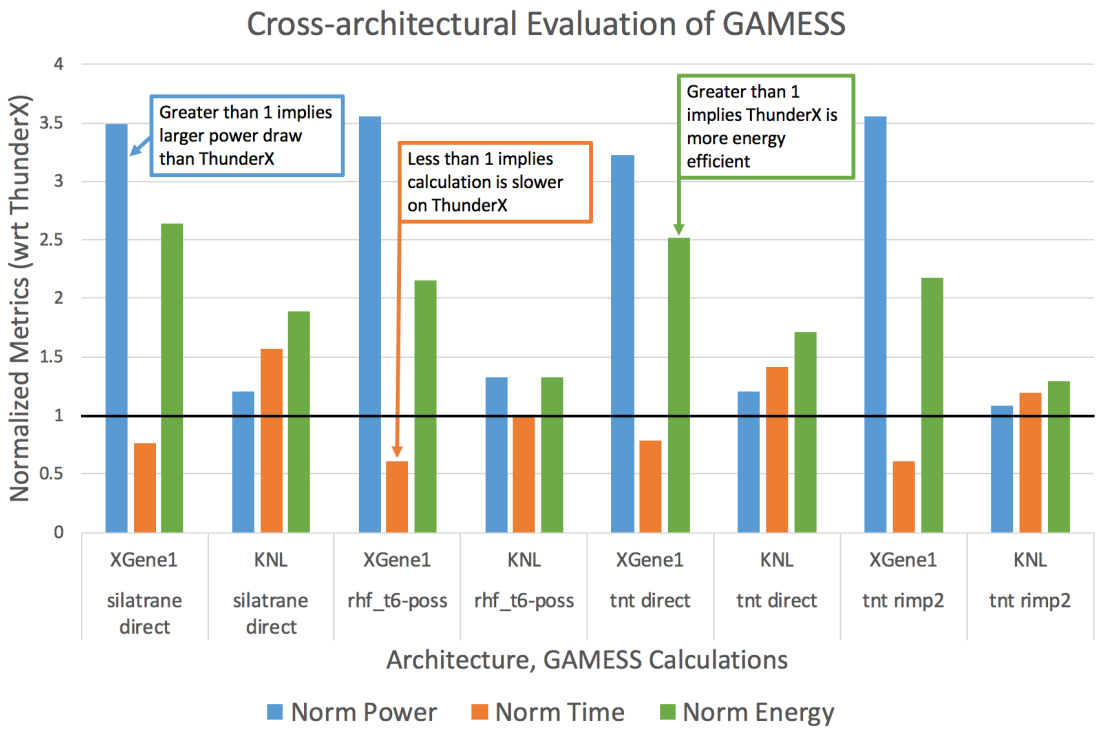


Figure 5: Cross-architectural comparisons; performance, power and energy metrics are normalized with respect to Cavium ThunderX

The study described above was expanded to include a large set of GAMESS benchmarking calculations to focus on just the many-core architectures – ThunderX and KNL. Figure 6 shows the results. Similar to Figure 1, performance and energy metrics for KNL are normalized using the respective measurements for ThunderX system. For approximately half the calculations, ThunderX shows better time-to-solution; for 60% of the benchmarking inputs, ThunderX delivers better energy efficiency. Some calculations, however, show significantly lower performance on ThunderX (as much as 2.4X slower on ThunderX).

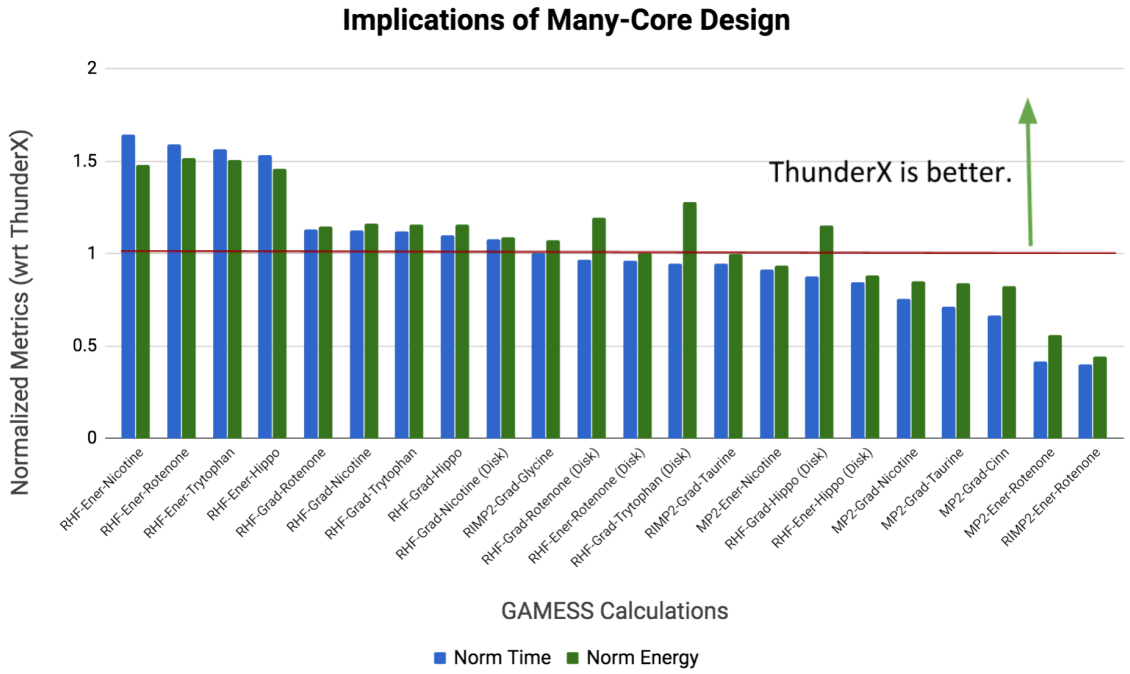


Figure 6: Comparing many-core architectures in terms of their suitability to GAMESS calculations.

Another emerging HPC architectural trend is the use of Field Programmable Gate Arrays (FPGAs) as accelerators. FPGAs are expected to consume significantly less energy per flop than CPU and GPU architectures.

We ported a portion of the GAMESS-SIMGMS computational chemistry kernel containing the Hartree-Fock procedure to FPGA enabled machines using the OpenARC compiler from Oak Ridge National Laboratory and evaluated the performance results. The GAMESS-SIMINT Hartree-Fock quantum chemistry method is used both to compute molecular properties and as a starting point for higher accuracy, more computationally demanding methods. The computational bottleneck of the Hartree-Fock procedure is construction of the Fock matrix, which requires computation of many electron repulsion integrals (ERIs). Since OpenARC takes only C code as input, we translated the GAMESS-SIMGMS kernel to pure C code by hand. We then inserted OpenACC directives to parallelize the code. We used OpenARC to transform the code into OpenCL optimized for FPGAs. The final step was to use the Intel SDK for OpenCL to compile the code to an FPGA executable. We verified that the FPGA executable produced the same results as the C++ version of the code running on a CPU. We

achieved up to 9.5X speedup on a Stratix V FPGA, compared to an Intel Xeon E5520 CPU, using double precision.

We continued our exploration of FPGAs by using the OpenARC compiler to generate optimized OpenCL code for the Arria 10 FPGA. We also generated code for the NVIDIA P100 GPU. We achieved up to 64 times speedup on the Arria 10, compared to an Intel Xeon E5520 CPU, using double precision. This speedup was more than ten times what we previously achieved on the Stratix V FPGA. We achieved better speedup (up to 160x) and better scalability on the P100 GPU than on the FPGAs. Performance comparisons are shown in Figures 7-9 below.

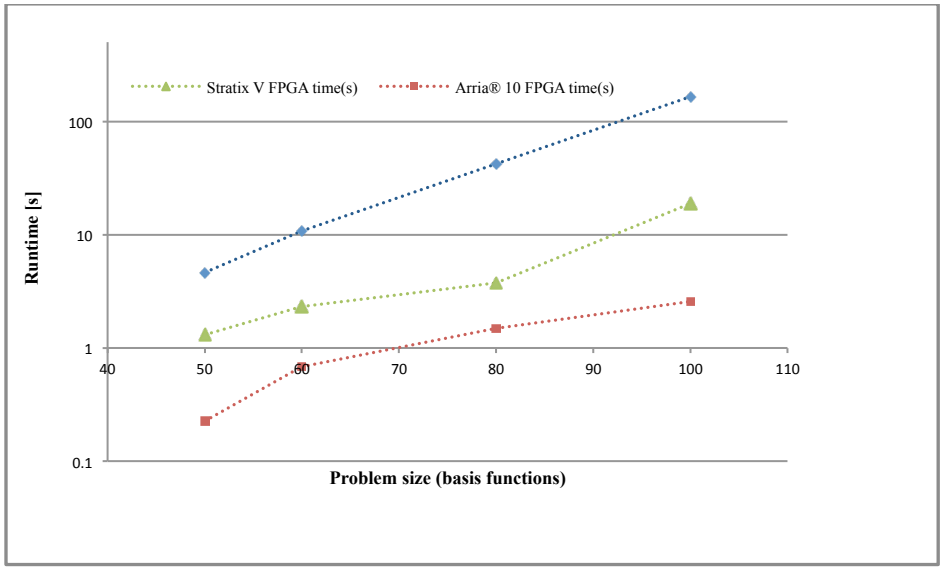


Figure 7: Runtime of SIMGMS kernel on FPGAs (Stratix V and Arria 10) vs. CPU (log scale)

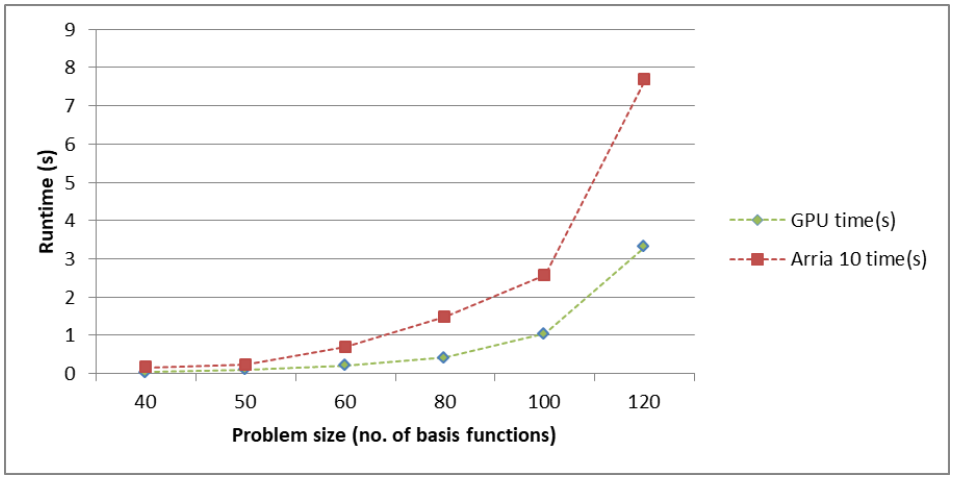


Figure 8: Runtime of SIMGMS kernel on NVIDIA GPU vs. Arria 10 FPGAs

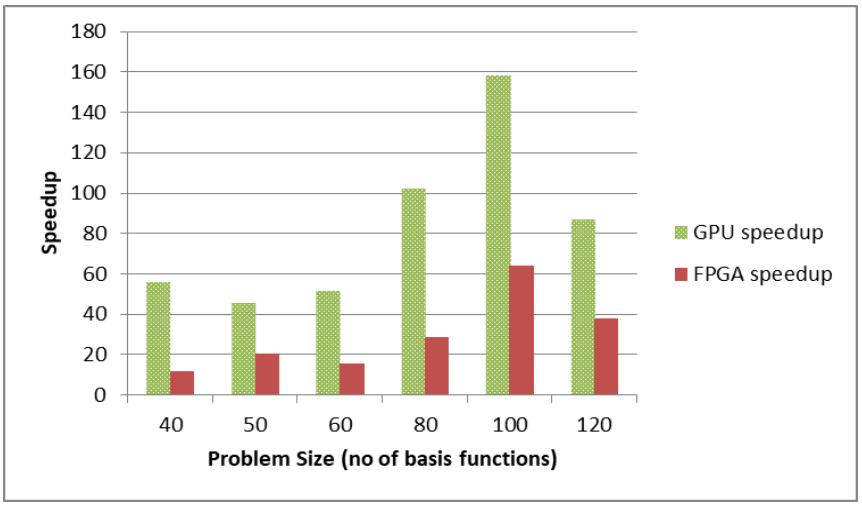


Figure 9: Speedup on NVIDIA P100 GPU and Arria 10 FPGA compared to Intel Xeon E5520 CPU

Oversubscription. We have completed the work evaluating the effects of oversubscription on semi-direct MP2 algorithms on multiple nodes. As demonstrated in our 2017 paper, there are specific subroutines within the semi-direct MP2 energy and semi-direct MP2 gradient algorithms in NWChem where performance can be significantly improved by increasing the parallelism. Provided in Figure 10 are the normalized semi-direct MP2 energy wall times using both two and three nodes executing with 1*n*, 2*n* and 3*n* processes, where *n* is the number of physical processes.

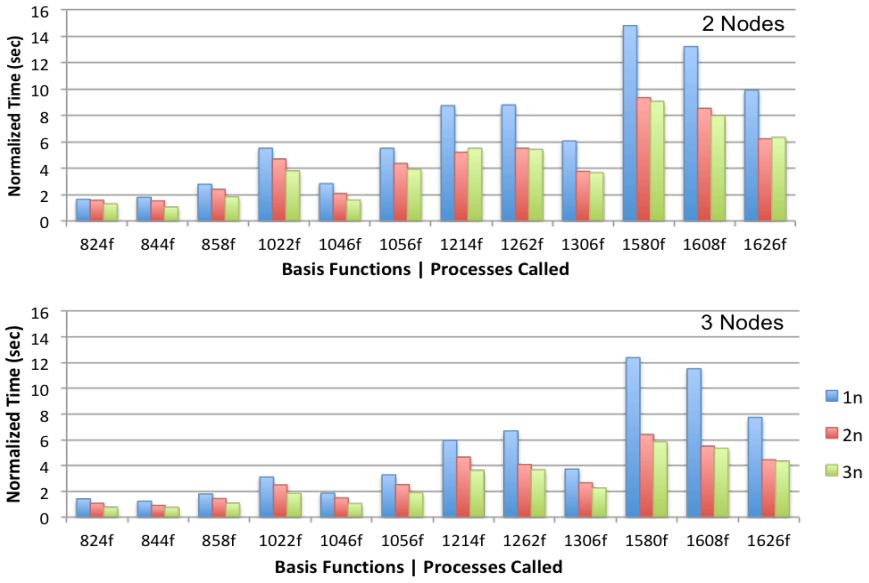


Figure 10. Semi-direct MP2 energy wall times at 1*n,* 2*n*, and 3*n* processes, normalized by the number of basis functions, for (top) two nodes and (bottom) three nodes.

Oversubscribing when using two nodes improved MP2 energy wall times 27-36%, with the largest improvements happening at the 3*n* level. When executing on three nodes, oversubscription improved MP2 energy wall times 30-44%, with the largest improvements again taking place at the 3*n* level. All of the time saved occurs within the *moin* subroutine, which is I/O intensive. The *make* subroutine, which is much more computationally intensive than the *moin*, is adversely affected by oversubscription, but only accounts for a small percentage of the total MP2 calculation.

Using the Performance Counter Monitor (PCM) API, we were able to monitor CPU and DRAM power, cache access, C-state residency, and average socket frequency. Figure 11 provides the percent energy used relative to 1*n* for each level of oversubscription for both two and three nodes. We found that through oversubscription total combined CPU and DRAM energy dropped by an average 5-10% for both two and three nodes. More total energy was saved by the DRAM than by the CPU when oversubscribing, most likely due to more efficient use of the L2 and L3 cache.

C-state residency data, which displays the percentage of time the cores were in a particular power-saving C-state, show a strong correlation between the time saved on the calculations and the time less spent in the idle C7 state. The plots in Figure 12 illustrate this relationship and provide the R^2 values, as well. Average socket frequency data corroborates this trend showing the average frequency increases significantly as the level of oversubscription increases. The drop in C7 residency along with the socket frequency data indicate an overall increase in CPU efficiency through oversubscription.

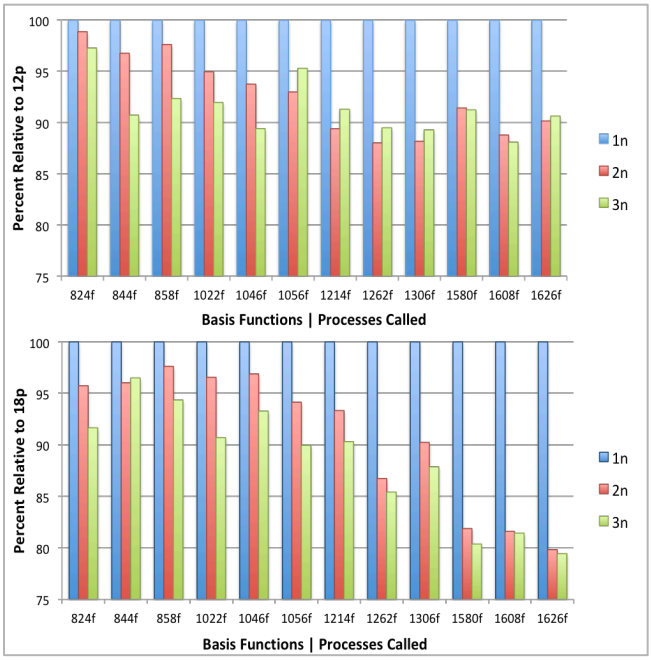


Figure 11. Percent energy used for all twelve chemical systems at each level of oversubscription relative to 1*n* processes for

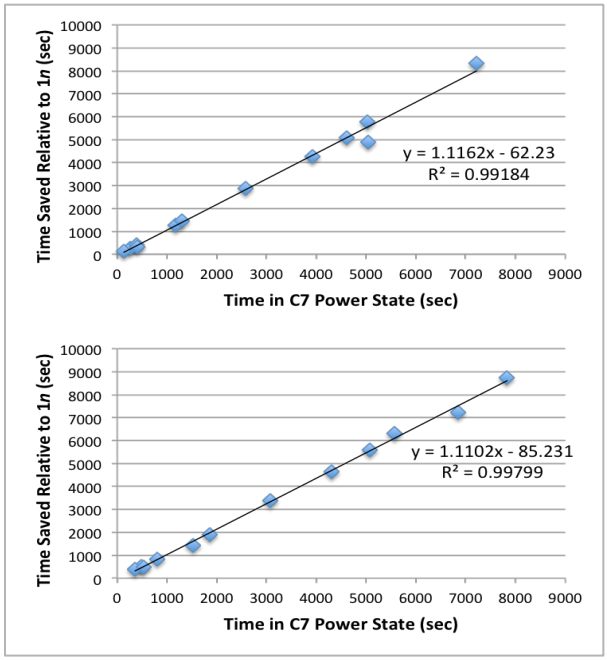


Figure 12. Total wall time saved versus time difference in C7 residency compared to 1*n* for (top) 2*n* processes and (bottom)

Similar results were found for the semi-direct MP2 gradient algorithm with time savings of 8-15% compared to no oversubscription. Only two of the eleven subroutines, *moin* and *back*, benefit from oversubscription. The remaining nine subroutines are either minimally or adversely affected. Unlike the MP2 energy calculations, the differences in total energy consumed by the CPU and DRAM at the various levels of oversubscription are negligible.

The work performed using a single node and multiple nodes indicate that with a more careful application of these methods, even greater time and energy savings are possible, and could most likely be applied to similarly structured calculations.

Data Flow Algorithms. Dataflow programming models have been growing in popularity as a means to deliver a good balance between performance and portability in the post-petascale era. Over the past 12 months, we finalized the performance tuning of our dataflow programming models for a state-of-the-art electronic structure theory application that we developed over the time span of this grant. The main objective of this work is to move away from traditional programming models that force scientific applications to be developed for specific architectures or platforms. Instead, we use dataflow programming models to represent the algorithms in a way that enables us to observe and capture data dependencies, which is the most essential property of an algorithm. We evaluated different dataflow executions: (1) explicit dataflow, where the dataflow is specified explicitly by the developer; and (2) implicit dataflow, where a task scheduling runtime derives the dataflow using per-task, data-access information embedded in a serial program. We used methods from the NWChem Quantum Chemistry application as our science driver, and we present our findings using three different task-based runtimes PaRSEC, StarPU, and OpenMP, which enable the different forms of dataflow execution.

"Implicit Dataflow" Performance: 32, 64, 128 nodes:
Figure 13 shows that performance of implicit dataflow Coupled Cluster Single Double (CCSD) is on par for all three runtimes and exhibits a speedup over the original CCSD of a factor of 1.5 when running on 128 nodes. This effort substantiates that "*implicit dataflow-based execution at the node level*" reveals notable performance benefits and enables more efficient and scalable computation.

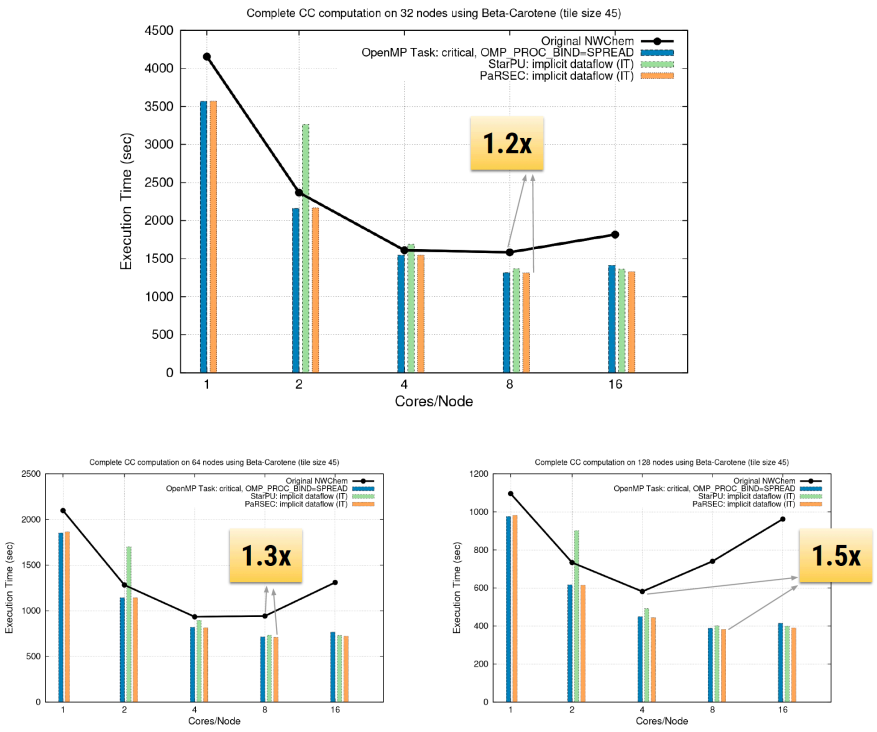


Figure 13: Implicit Dataflow computation for NWChem CCSD on 32, 64, 128 nodes

"Explicit Dataflow" Performance: 32, 64, 128 nodes. On the other hand, the explicit dataflow model demands a much bigger engineering job compared to the implicit dataflow models. The Parameterized Task Graph (PTG) programming paradigm proposes a completely different path from the way parallel applications have been designed and developed up to the present. The PTG decouples the expression of parallelism in the algorithm from the control flow ordering, data distribution, and load balance.

Despite the lower startup overhead of implicit dataflow paradigms in terms of development effort, (i.e., simply submitting tasks in the sequential flow of the original code), the significance of the increased implementation effort of the PTG becomes visible when comparing the superior performance of the explicit dataflow version of CCSD with the implicit dataflow and traditional CCSD computation. Figure 14 shows that PTG version of CCSD outperforms the original CCSD version by a significant margin---to be precise, by a factor of 2.6 on 32 nodes.

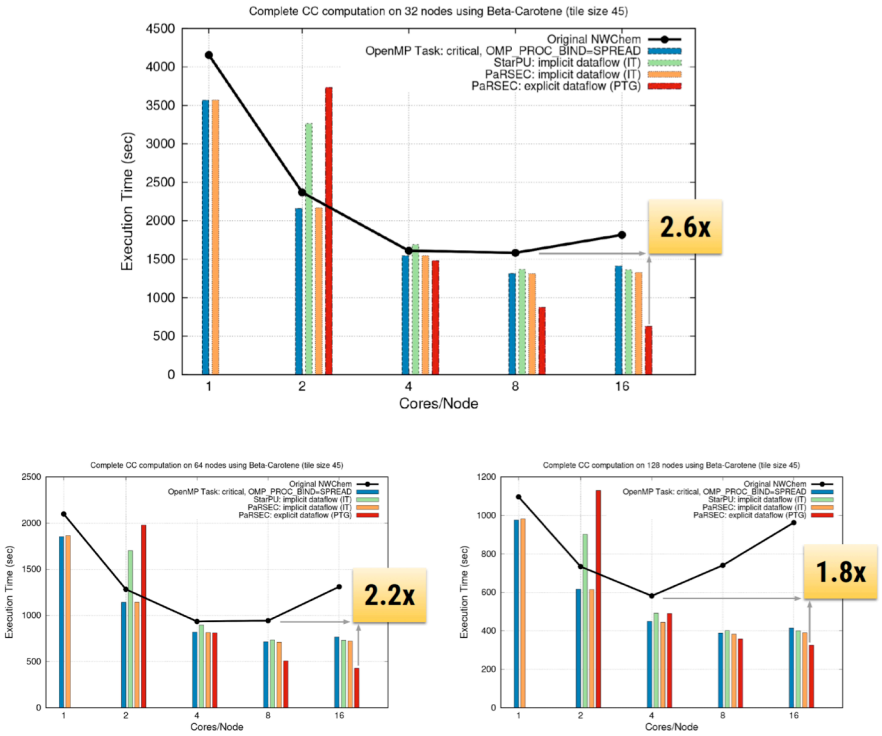


Figure 14: Implicit and Explicit (red bars) Dataflow computation for NWChem CCSD on 32, 64, 128 nodes

Additionally, the explicit dataflow version manages to use an increasing number of cores to improve the performance---all the way up to 2,048 cores when running on 128 nodes (with 16 cores/node)---demonstrating not only a significant performance boost but also better scaling and greater utilization of compute resources due to the ability to fully overlap computation with communication. On the contrary, the original and the implicit dataflow CC code perform best on 8 cores/node and are not able to take full advantage of the 16 available cores/node because both versions are tied to the limitations of the original control flow, such as blocking communication, shared variables that are atomically updated, which is at the heart of the original load balancing solution, and a significant amount of synchronizations that limit the overall scaling on much larger computational resources. In contrast, the PTG CC version distributes the work in a round-robin fashion and avoids any kind of global agreement in the critical path of the DAG execution.

Continued collaboration with Heike Jagode and Anthony Danalis to implement the Hartree-Fock self-consistent field algorithm (SCF) (15) using a dataflow-based process that will be scheduled using PaRSEC. This year, all of the individual steps for a single iteration of an SCF calculation were implemented as sequentially executed directed acyclic graphs (DAGs) within the PaRSEC framework. This has facilitated fine-grained unit testing of all the components. Currently, all the components are being combined into a single DAG that will perform an entire iteration of the SCF loop. Once that is complete, the SCF convergence check will be moved into the framework and benchmarking will begin.

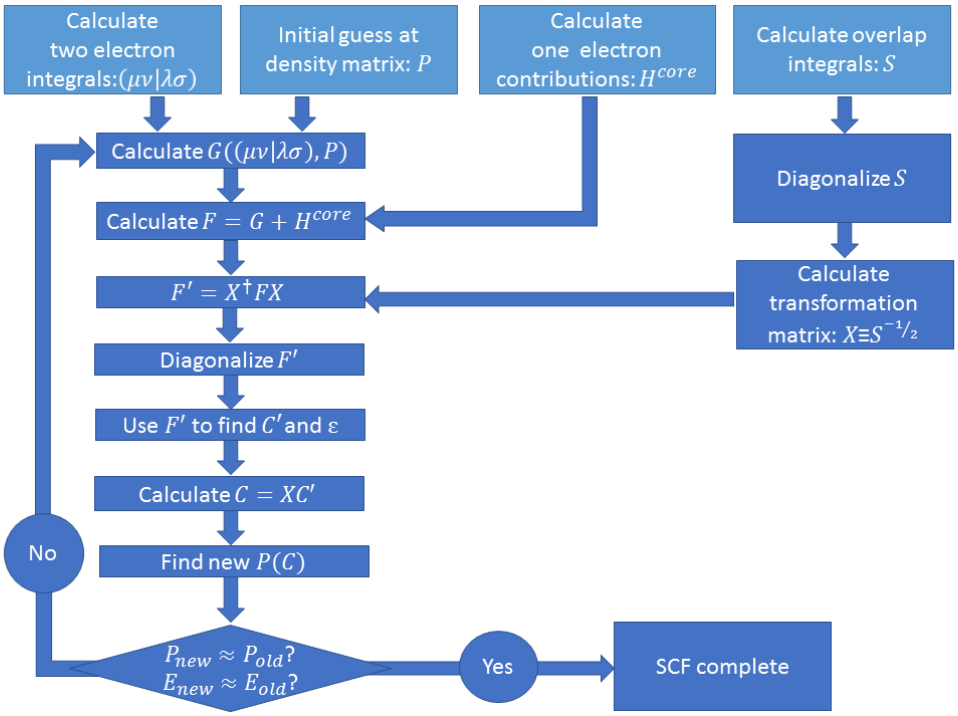
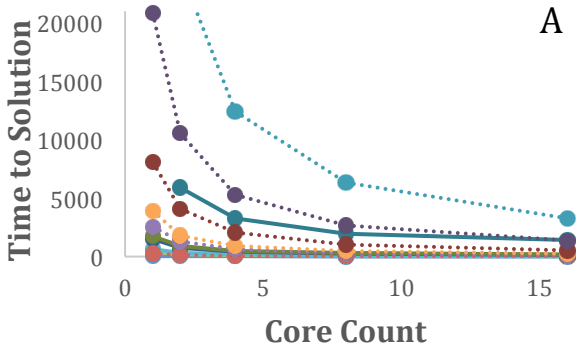


Figure 15: Overall structure of the self-consistent field (SCF) calculation. Currently, each rectangle represents a distinct kernel that must finish execution and whose results must be synced before the next step can begin. Once this work is complete, intermediate syncing operations will be unnecessary and tasks can begin as soon as the necessary chunk of data from the previous task becomes available.

KNL vs. Haswell Performance Analysis. In the pursuit of improving the field of high performance computing, novel computer architectures have the capability to make drastic improvements and help to push the cutting edge of scientific research. One of the ever-steadfast roadblocks to many computational tasks is the run time of complex applications. The design of faster and more parallel oriented architectures continues to address this roadblock. In this work, the time to solution of common quantum chemistry calculations were tested on the second-generation Xeon Phi and the Haswell architectures. The Haswell had a faster time to solution for every given system or core count tested, even when compared to the full thread level parallelized OpenMP RHF on the KNL. The more robust cores on the Haswell, along with a much higher clock speed, makes the architecture much faster than the KNL when compared core to core. One example is provided in the following graph for the MP2 method and the 6-311G(d) basis set.



Effective Fragment Potential. The effective fragment potential (EFP) is a highly accurate method for capturing intermolecular interactions. In the effective fragment potential method the Coulomb potential is represented using a set of multipole moments generated by the distributed multipole analysis (DMA) method. Misquitta, Stone, and Fazeli recently developed a basis space-iterated stockholder atom (BS-ISA) method to generate multipole moments. Our recent study assesses the accuracy of the EFP interaction energies using sets of multipole moments generated from the BS-ISA method, and from several versions of the DMA method (such as analytic and numeric grid-based), with varying basis sets. Both methods lead to reasonable results, although using certain implementations of the DMA method can result in large errors. With respect to the CCSD(T)/CBS interaction energies, the mean unsigned error (MUE) of the EFP method for the S22 data set using BS-ISA–generated multipole moments and DMA-generated multipole moments (using a small basis set and the analytic DMA procedure) are 0.78 and 0.72 kcal/mol, respectively. The MUE accuracy is on the same order as MP2 and SCS-MP2. The MUEs are lower than in a previous study benchmarking the EFP method without the EFP charge transfer term, demonstrating that the charge transfer term increases the accuracy of the EFP method. Regardless of the multipole moment method used, it is likely that much of the error is due to an insufficient short-range electrostatic term (i.e. charge penetration term), as shown by comparisons with symmetry-adapted perturbation theory. The conclusion, as illustrated in the Table 2, is that there is no compelling reason to switch from the DMA method to the ISA method.

Table 2: MUE for the EFP Coulomb term (kcal/mol)

	EFP/ISA	EFP/DMA0-small	EFP/DMA0-small-atoms	EFP/DMA0	EFP/DMA4	EFP/DMA-mixed
MUE(HB)	2.485	1.631	1.806	0.863	5.453	3.596
MUE(DISP)	2.560	2.475	2.431	3.105	1.514	1.487
MUE(MIXED)	0.960	0.553	0.614	0.806	0.816	0.897
MUE(overall)	2.027	1.595	1.654	1.677	2.545	1.970

HB=hydrogen bonded; DISP=dispersion dominant.

Journal publications:

K. Kadau, F. J. Cherne, R. Ravelo, T. C. Germann, "Shock-induced phase transformations in gallium single crystals by atomistic methods", Physical Review B 88:144108, 10/2013.

R. Ravelo, T. C. Germann, O. Guerrero, Q. An, B. L. Holian, "Shock-induced plasticity in tantalum single crystals: Interatomic potentials and large-scale molecular-dynamics simulations", Physical Review B 88: 134101, 10/2013.

S.S. Leang, A. P. Rendell, and M.S. Gordon, "Quantum Chemical Calculations using Accelerators: Migrating Matrix Operations to the NVIDIA Kepler GPU and the Intel Xeon Phi", J. Chem. Theory Comp., 10, 908 (2014).

Xingfu Wu, Valerie Taylor, Charles Lively, Hung-Ching Chang, Bo Li, Kirk Cameron, Dan Terpstra and Shirley Moore, MuMMI: Multiple Metrics Modeling Infrastructure (Book Chapter), Tools for High Performance Computing, (Eds: Knupfer, A, Gracia, J., Nagel, W.E, Resch,M.M), Springer, 2014.

C. Lively, V. Taylor, X. Wu, H.-C. Chang, C.-Y. Su, K. Cameron, S. Moore, and D. Terpstra, E-AMOM: An Energy-Aware Modeling and Optimization Methodology for Scientific Applications on Multicore Systems, Computer Science - Research and Development, Volume 29, Issue 3 (2014), Page 197-210.

D.R. Tramontina, P. Erhart, T.C. Germann, J.A. Hawreliak, A. Higginbotham, N. Park, R. Ravelo, A. Stukowski, M.J. Suggit, Y. Tang, J.S. Wark and E.M. Bringa, "Molecular dynamics simulations of shock-induced plasticity in tantalum", Journal of High Energy Density Physics, 10, 9. (2014).

M. Sosonkina, L.T. Watson, J. He "Remark on on Algorithm 897: VTDIRECT95", ACM TOMS 41(3), Article No. 22, 2015.

K. Keipert, G. Mitra, V. Sunriyal, S.S. Leang, M. Sosonkina, A. Rendell, and M.S. Gordon, "Energy Efficient Computational Chemistry: Comparison of X86 and ARM Systems", J. Chem. Theory Comp., 11, 5055 (2015)

D.G. Tomlinson, A. Asadchev, and M.S. Gordon, "A New Approach to Second Order Perturbation Theory", J. Chem. Theory Comp., 37, 1274 (2016).

Jagode, H., Danalis, A., Dongarra, J. "Accelerating NWChem Coupled Cluster through dataflow-based Execution," *International Journal of High Performance Computing Applications (IJHPCA)*, pp. 1-13, Jan. 2017

V. Sundriyal and M. Sosonkina, "Modeling of the CPU Frequency to Minimize Energy Consumption in Parallel Applications", Sustainable Computing, 17 (2018), pp. 1–8, DOI: https://doi.org/10.1016/j.suscom.2017.12.002

E. Fought, V. Sundriyal, M. Sosonkina, and T.L. Windus, "Saving time and energy with oversubscription and semi-direct Møller-Plesset second order perturbation methods", *J. Chem. Theory and Comp.*, **2017**, *38*, 830-841. DOI: 10.1002/jcc.24756

V. Sundriyal, K. Kristopher, M. Sosonkina, M.S. Gordon, "Effect of Frequency Scaling Granularity on Energy-Saving Strategies", Int'l Journal on High-Performance Applications, First published online May 2018, DOI:https://doi.org/10.1177/1094342018774405

V. Sundriyal and M. Sosonkina, "Runtime power-aware energy-saving scheme for parallel applications", Int. J. of High Performance Systems Architecture, 2017 Vol.7, No.3, pp.129–139, First published online April 2018, DOI: https://doi.org/10.1504/IJHPSA.2017.10012609

E. Coleman and M. Sosonkina, "Self-Stabilizing Fine-Grained Parallel Incomplete LUFactorization", Sustainable Computing, First published online Feb. 2018, DOI: https://doi.org/10.1016/j.suscom.2018.01.003

Jagode, H., Danalis, A., Hoque, R., Faverge, M., Dongarra, J. "Evaluation of Dataflow Programming Models for Electronic Structure Theory" Concurrency and Computation: Practice and Experience, Special Issue on Parallel and Distributed Algorithms, issue e4490, pp. 1-20, 2018

Jagode, H., Danalis, A., Hoque, R., Faverge, M., Dongarra, J. "Evaluation of Dataflow Programming Models for Electronic Structure Theory" Concurrency and Computation: Practice and Experience, Special Issue on Parallel and Distributed Algorithms, issue e4490, pp. 1-20, 2018

Haidar, A., Jagode, H., Vaccaro, P., YarKhan, A., Tomov, S., Dongarra, J. "Investigating Power Capping toward Energy-Efficient Scientific Applications," *Concurrency and Computation: Practice and Experience (CCPE): Special Issue on Power-Aware Computing*, vol. 2018, issue e4485, pp. 1-14, 2018.

E.L. Fought, V. Sundriyal, M. Sosonkina, and T.L. Windus, "Improving Efficiency of Semi-Direct Moller-Plesset Second Order Perturbation Methods Through Oversubscription on Multiple Nodes", J. Comp. Chem., submitted

V. Sundriyal, K. Keipert, M.Sosonkina, and M.S. Gordon, "Effect of Frequency Scaling Granularity on Energy-Saving Strategies", Int. J. High Performance Computing, accepted.

Conference papers:

Peraza, J., Tiwari, A., Laurenzano, M., Carrington, L., Ward, W., and Campbell, R. (2013) Understanding the Performance of Stencil Computations on Intel's Xeon Phi. In Cluster Computing 2013.

Jagode, H., Danalis, A., Herault, T., Bosilca G., Dongarra, J., Kowalski, K., Windus, T.L. "Utilizing Dataflow-based Execution for Coupled Cluster Methods" IEEE Cluster 2014, Madrid, Spain, September 22-26, 2014.

Tiwari, A., Laurenzano, M., Jundt, A., Ward, W., Campbell, R., and Carrington, L. (2014) Adaptive Model-driven Facility-wide Management of Energy Efficiency and Reliability. in Workshop on Modeling and Simulation of Exascale Systems and Applications (MODSIM'14), Seattle, WA

Laurenzano, M., Tiwari, A., Jundt, A., Peraza, J., Carrington, L., Ward, W., and Campbell, R. (2014) Characterizing the Performance-Energy Tradeoff of Small ARM Cores in HPC Computation. in EuroPar 2014. August 2014.

Cicotti, P., Tiwari, A., and Carrington, L. (2014) Efficient Speed (ES): Adaptive DVFS and Clock Modulation for Energy Efficiency. In CLUSTER 2014. September 2014.

Jagode, H., Danalis, A., Bosilca G., Dongarra, J. "Accelerating NWChem Coupled Cluster through dataflow-based Execution" 11th International Conference on Parallel Processing and Applied Mathematics (PPAM 2015), Krakow, Poland, September 6-9, 2015.

Danalis, A., Jagode, H., Bosilca G., Dongarra, J. "PaRSEC in Practice: Optimizing a legacy Chemistry application through distributed task-based execution" IEEE Cluster

2015, Chicago, Illinois, USA, September 8-11, 2015.

Rogelio Long, Shirley Moore, and Barry Rountree:
Iso-power-efficiency: an approach to scaling application codes with a power budget.
Eleventh Workshop on High-Performance Power-Aware Computing (HPPAC 2015),
Hyderabad, India, May 2015.

Rogelio Long, Shirley Moore, and Barry Rountree:
Iso-power-efficiency: an approach to scaling application codes with a power budget.
Eleventh Workshop on High-Performance Power-Aware Computing (HPPAC 2015),
Hyderabad, India, May 2015.

Porter, L., Laurenzano, M., Tiwari, A., Jundt, A., Ward, W., Campbell, R., and Carrington, L. (2015) To SMT or not to SMT: Understanding the Impact of Simultaneous Multithreading in HPC. In ACM Transactions on Architecture and Code Optimization (ACM TACO), January 2015.

Jundt, A., Tiwari, A., Ward, W., Campbell, R., Carrinton, L. (2015) Optimizing Codes on the Xeon Phi: A Case-study with LAMMPS. In XSEDE 2015.

Tiwari, A., Jundt, A., Ward, W., Campbell, R., Carrington, L. (2015) Building Blocks for a System-wide Power and Thermal Management Framework. To Appear in ICPADS 2015.

Jundt, A., Cauble-Chantrenne, A., Tiwari, A., Peraza, J., Laurenzano, M., Carrington, L. (2015) Compute Bottlenecks on the New 64-bit ARM. In E2SC 2015.

Tiwari, A., Keipert, K., Jundt, A., Peraza, J., Leang, S., Laurenzano, M., Gordon, M., Carrington, C. (2015) Performance and Energy Efficiency Analysis of 64-bit ARM Using GAMESS. In Co-HPC 2015.

Rogelio Long and Shirley Moore. Countering the noise-induced critical path problem. 12th Workshop on High-Performance Power-Aware Computing (HPPAC 2016), Chicago, IL, May 2016.

Laurenzano, M., Tiwari, A., Cauble-Chantrenne, A., Jundt, A., Ward, W., Campbell, R., Carrington, L. (2016) Characterization and bottleneck analysis of a 64-bit ARMv8 platform. In ISPASS 2016.

Shirley Moore. Achieving safety for power shifting in over-provisioned high performance computing systems. 12th Workshop on High-Performance Power-Aware Computing (HPPAC 2016), Chicago IL, May 2016.

W. K. Umayanganie Munipala and Shirley Moore. An evaluation framework for scientific programming productivity. International Workshop on Software Engineering for Science (SE4Science 2016), Austin, Texas, May 2016.

W. K. Umayanganie Munipala and Shirley Moore. Code complexity versus performance for GPU-accelerated applications. 4th International Workshop on Software Engineering for Software Engineering in Computational Science and Engineering (SEHPCCSE'16), Salt Lake City, Utah, November 2016.

Laurenzano, M., Tiwari, A., Cauble-Chantrenne, A., Jundt, A., Peraza, J., Ward, W., Campbell, R, and Carrington, L. (2016): Characterization and Bottleneck Analysis of a 64-bit ARMv8 Platform. *In Intl. Symposium on Performance Analysis of Systems & Software (ISPASS)*.

David Pruitt and Eric Freudenthal. Preliminary investigation of mobile system features potentially relevant to HPC. Fourth Workshop on Energy Efficient Supercomputing (E2SC'16), Salt Lake City, Utah, November 2016.

E.N. Hahn, T.C. Germann, R. Ravelo, J.E. Hammerberg and M.A. Meyers, "Equilibrium Molecular Dynamics Simulations of Spall in Single Crystal Tantalum", AIP Proceedings of the 19th Biennial Conference on Shock Compression of Condensed Matter, 2016.

V. Sundriyal, E. Fought, T.L. Windus, and M. Sosonkina, "Power Profiling and Evaluating the Effect of Frequency Scaling on NWChem", *Proc. of 24th HPC Symp HPC '16*, **2016**, 19:1-19:8, pub. Society for Computer Simulation International, San Diego, CA, USA, ISBN: 978-1-5108-2318-1.

Rogelio Long and Shirley Moore. Countering the noise-induced critical path problem. 12th Workshop on High-Performance Power-Aware Computing (HPPAC 2016), Chicago, IL, May 2016.

David Pruitt and Eric Freudenthal. Preliminary investigation of mobile system features potentially relevant to HPC. Fourth Workshop on Energy Efficient Supercomputing (E2SC'16), Salt Lake City, Utah, November 2016.

E.N. Hahn, T.C. Germann, R. Ravelo, J.E. Hammerberg and M.A. Meyers, "Equilibrium Molecular Dynamics Simulations of Spall in Single Crystal Tantalum", AIP Proceedings of the 19th Biennial Conference on Shock Compression of Condensed Matter, 2016.

Shirley Moore. Achieving safety for power shifting in over-provisioned high performance computing systems. 12th Workshop on High-Performance Power-Aware Computing (HPPAC 2016), Chicago IL, May 2016.

W. K. Umayanganie Munipala and Shirley Moore. An evaluation framework for scientific programming productivity. International Workshop on Software Engineering for Science (SE4Science 2016), Austin, Texas, May 2016.

W. K. Umayanganie Munipala and Shirley Moore. Code complexity versus performance for GPU-accelerated applications. 4th International Workshop on Software Engineering for Software Engineering in Computational Science and Engineering (SEHPCCSE'16),

Salt Lake City, Utah, November 2016.

Umayanganie Klaassen, Shirley V. Moore, Kristopher Keipert, Mark S. Gordon, Jeffrey S. Vetter and Seyong Lee. Porting a GAMESS Computational Chemistry Kernel to FPGAs. H2RC'17, Denver, CO, November 2017.

V. Sundriyal, E. Fought, M. Sosonkina, and T.L. Windus, "Evaluating Effects of Application Based and Automatic Energy Saving Strategies on NWChem", *Proc. 25th HPC Symposium, HPC '17*, **2017**, *16*, 1-12.

Umayanganie Klaassen, Shirley V. Moore, Kristopher Keipert, Mark S. Gordon, Jeffrey S. Vetter and Seyong Lee. Porting a GAMESS Computational Chemistry Kernel to FPGAs. H2RC'17, Denver, CO, November 2017.

E. Coleman, E. Jensen, and Masha Sosonkina, "Impacts of Three Soft-Fault Models on Hybrid Parallel Asynchronous Iterative Methods", In *Proc. 7th Workshop on Applications for Multi-Core Architectures (WAMCA)*, Lyon, France, Sep. 2018, 8 pages.

E. Jensen, E. Coleman, and Masha Sosonkina, "Using Modeing to Improve the Performance of Asynchronous Jacobi", In *Proc. 2018 Int'l Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, Las Vegas, NV, Jul. 2018.

E. Coleman and M. Sosonkina, "Convergence and resilience of the fine-grained parallel incomplete LU factorization for non-symmetric problems", In *Proc. of the 26th High Performance Computing Symposium* (HPC '18), Society for Computer Simulation International, San Diego, CA, USA, 2018, Article 7, 12 pages.

E. Jensen and M. Sosonkina, "Modeling a task-based matrix-matrix multiplication application for resilience decision making", In *Proc. of the 26th High Performance Computing Symposium* (HPC '18), Society for Computer Simulation International, San Diego, CA, USA, 2018, Article 10, 11 pages.

V. Sundriyal, M. Sosonkina, B. M. Westheimer, and M. Gordon, "Comparisons of core and uncore frequency scaling modes in quantum chemistry application GAMESS", In *Proc. of the 26th High Performance Computing Symposium* (HPC '18), Society for Computer Simulation International, San Diego, CA, USA, 2018, Article 13, 11 pages.

Tiwari, A., Keipert, K., Jundt, A., Peraza, J., Leang, S., Laurenzano, M., Gordon, M., and Carrington, L. (2015): Performance and Energy Efficiency Analysis of 64-bit ARM using GAMESS. *In 2nd Intl. Workshop on Hardware-Software Co-Design for High Performance Computing (Co-HPC)*.

Presentations:

M. Sosonkina, “Performance and Energy Modeling of CoMD offloaded to Intel Xeon Phi”, Seminar at Old Dominion University, ECE Department.

G. Lawson "Thread Affinity, Power, Energy, and Performance on the Intel Xeon Phi", Modeling, Simulation, and Visualization Student Capstone Conference, April 17, 2014 Suffolk, VA.

"Utilizing Dataflow-based Execution for Coupled Cluster Methods", *IEEE Cluster 2014*, Madrid, Spain, September 22-26, 2014.

"Power Monitoring with PAPI for Extreme Scale Architectures and Dataflow-based Programming Models", *Workshop on Monitoring and Analysis for High Performance Computing Systems Plus Applications (HPCMASPA 2014), IEEE Cluster 2014*, Madrid, Spain, September 26, 2014.

G. Lawson, "Towards Modeling Energy Consumption on Intel Xeon Phi", Modeling, Simulation, and Visualization Student Capstone Conference, April 16, 2015 Suffolk, VA.

I. Ngatang, "Initial Investigation of CoMD on Tightly-Coupled Heterogeneous Platforms", Modeling, Simulation, and Visualization Student Capstone Conference, April 16, 2015 Suffolk, VA.

"Massively Scalable Coupled Cluster Codes using Dataflow-based Execution" T.L. Windus, K. Kowalski, A. Danalis, H. Jagode, 251st American Chemical Society National Meeting & Exposition, Boston, MA August 2015

"Power Management and Event Verification in PAPI", *9th Parallel Tools Workshop*, Dresden, Germany, September 2-3, 2015.

"Accelerating NWChem Coupled Cluster through dataflow-based Execution", *Parallel Processing and Applied Mathematics: 11th International Conference, PPAM 2015,* Krakow, Poland, September 6-9, 2015.

"PaRSEC in Practice: Optimizing a legacy Chemistry application through distributed task-based execution", *IEEE Cluster 2015*, Chicago, Illinois, September 8-11, 2015.

M. Sosonkina, “Energy Modeling in Applications offloaded to Intel Xeon Phi”, Invited talk at the Int'l Simulation Conference SIMULATION 2016, Kyiv, Ukraine May 2016.

“Dataflow-based Execution of Coupled Cluster Codes”, T.L. Windus, "Emerging Methods for Quantum N-Body Problem" Symposium, 9th Congress of the International Society for Theoretical Chemical Physics, July 2016

"Recent Advances in the Performance API (PAPI)", *10th Scalable Tools Workshop*, Lake Tahoe, CA, USA, August 1-4, 2016.

"PAPI for Intel Xeon Phi Knights Landing", *International SuperComputing 2016 (ISC High Performance 2016)*, Frankfurt, Germany, June 19-23, 2016.

Shirley Moore, "Exploring New Architectures for Computational Chemistry", presentation at the 255th American Chemical Society National Meeting & Exposition, March 18-22, 2018, New Orleans, Louisiana.