# Performance Study of Popular Computational Chemistry Software Packages on Cray HPC Systems

Junjie Li (lijunj@iu.edu)
Shijie Sheng (shengs@iu.edu)
Raymond Sheppard (rsheppar@iu.edu)
Pervasive Technology Institute, Indiana University
535 W Michigan St, Indianapolis, IN, USA

## Abstract

Two Cray HPC systems, XE6/XK7 and XC30, are deployed at Indiana University serving scientific research across all the eight campuses. Across over 130 different scientific disciplines that are using HPC, the major source of workload comes from chemistry. Here, the performance of quantum chemistry package NWChem and molecular dynamic software GROMACS and LAMMPS are investigated in in terms of parallel scalability and different parallelization paradigms. Parallel performance up to 12,288 MPI ranks are studied, and the results from the two different Cray HPC systems are compared.

## I. Introduction

Indiana University currently focuses on the physical sciences, liberal arts and medical sciences with very limited engineering. Research is supported by a long-serving 1,020-node XE6/XK7 [1] and a recently installed 552-node XC30 [2]. Chemistry and physics are the two primary consumers of high performance computing (HPC). The chemistry department burns over 40% of the total workload. Chemistry has long been a scientific field that relies heavily on simulations. Computational chemistry software are getting increasingly complicated. More advanced algorithms are being modelled, and multiple parallel programming paradigms are often available for each model. Therefore, as an HPC service provider, it is crucial to understand the performance and limitation of these applications so that we can improve our service and help scientists generate the best throughput for their scientific research.

In this work, we evaluate the performance of three popular computational chemistry packages, NWChem [3], Gromacs [4] and Lammps [5] and present the results of multiple configurations on each of our Cray machines. These software packages support both quantum chemical methods and classical molecular dynamics (MD). Various theories are benchmarked, from the popular fast methods to the highly expensive accurate methods.

This paper is organized as follows: Section II introduces Indiana University's two Cray HPC systems, Section III outlines the functionality and features of the different computational chemistry software packages, Section IV discusses our benchmark methods, Section V presents the results, and the paper is then concluded in Section VI.

## II. Cray HPCs at Indiana University

### A. BigRed II (XE6/XK7)

Big Red II is Indiana University's primary system for high-performance parallel computing. With a theoretical peak performance (Rpeak) of one thousand trillion floating-point operations per second (1 petaFLOP) and a maximal achieved performance (Rmax) of 596.4 teraFLOPS, Big Red II is among the world's fastest research supercomputers. Owned and operated solely by IU, Big Red II is designed to accelerate discovery both through computational experimentation and effective analysis of big data in a wide variety of fields, including medicine, physics, fine arts, and global climate research.

Big Red II features a hybrid architecture based on two Cray's supercomputer platforms. As configured upon entering production in August 2013, Big Red II comprised 344 XE6 (CPU-only) compute nodes and 676 XK7 "GPU-accelerated" compute nodes, all connected through Cray's Gemini scalable interconnect, providing a total of 1,020 compute nodes, 21,824 processor cores, and 43,648 GB of RAM. Each XE6 node has two AMD Opteron 16-core Abu Dhabi x86_64 CPUs and 64 GB of RAM; each XK7 node has one AMD Opteron 16-core Interlagos x86_64 CPU, 32 GB of RAM, and one NVIDIA Tesla K20 GPU accelerator.

### B. BigRed II+ (XC30)

Big Red II+ is a supercomputer that complements Indiana University's Big Red II by providing an environment dedicated to large-scale, compute-intensive research. Researchers, scholars, and artists with large-scale research needs have benefited from Big Red II; these users can now take advantage of faster processing capability and networking provided by Big Red II+. The system will help support programs at the highest level of the university, such as the Grand Challenges Program.

Big Red II+ is a Cray XC30 supercomputer providing 552 compute nodes, each containing two Intel Xeon E5 12-Core x86_64 CPUs and 64 GB of DDR3 RAM. Big Red II+ has a theoretical peak performance (Rpeak) of 286 trillion floating-point operations per second (286 teraFLOPS). All compute nodes are connected through the Cray Aries interconnect.

## III. Computational Chemistry Software Packages

### A. NWChem

NWChem is an ab initio computational chemistry software package which also includes quantum chemical and molecular dynamics functionality. It was designed to run on high-performance, parallel supercomputers as well as conventional workstation clusters. It aims to be scalable, both in its ability to treat large problems efficiently, and in its usage of available parallel computing resources. NWChem has been developed by the Molecular Sciences Software group of the Theory, Modeling & Simulation program in the Environmental Molecular Sciences Laboratory (EMSL) at the Pacific Northwest National Laboratory (PNNL). It was first was funded by the EMSL Construction Project.

### B. GROMACS

GROningen MAchine for Chemical Simulations (GROMACS) was initially released in 1991 by the University of Groningen, in the Netherlands and has been developed at the Royal Institute of Technology and Uppsala University in Sweden since 2001. It has gained popularity not only because it is easy to use and has a rich tool set for post-analysis, but also because its algorithm has good performance and optimizes well to different parallel paradigms.

### C. LAMMPS

Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) was developed at Sandia National Laboratory. It utilizes MPI for parallel communications and contains extensive packages (GPU, USER-OMP) to enable other parallel models. Many of its models are optimized for CPUs and GPUs. It also contains some features that other MD packages do not provide, for example, reactive force field simulations.

## IV. Benchmark Methods

NWChem represents the state-of-the-art software for quantum chemistry, its parallel performance outperforms other programs like GAMMES. Here we choose two representative types of calculations for benchmarking. The most popular in general use due to its relative speed to completion is the Density Functional Theory (DFT) energy calculation. It scales as $O(N^3 \sim N^4)$. The other is the so called "golden standard" method for quantum chemistry – Coupled Cluster

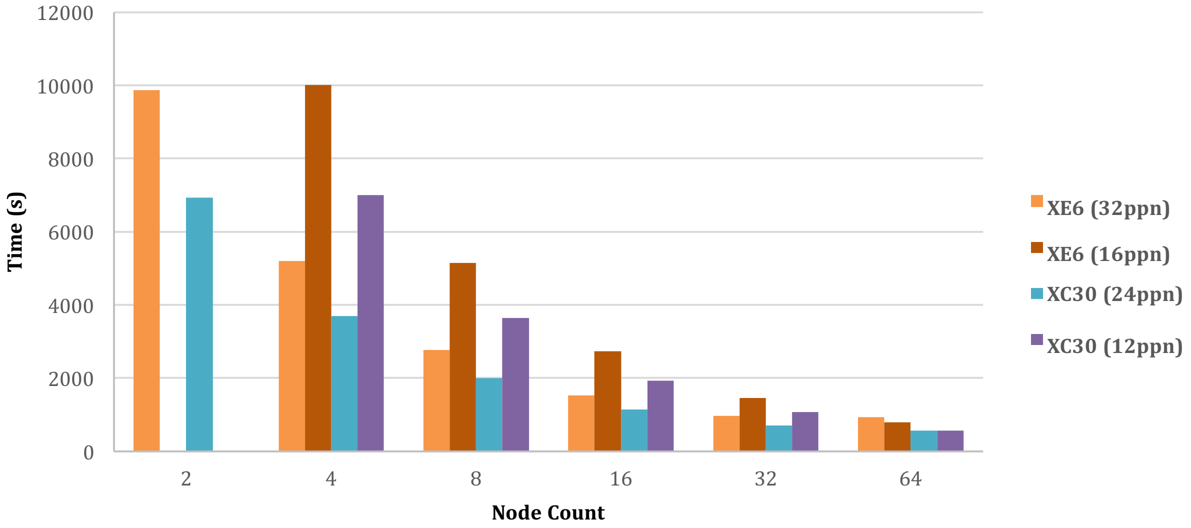


**Figure 1.** Performance of DFT calculations in NWChem on Cray XE6/XK7 and XC30

using Single, Double and perturbative Triple (CCSD(T)). This method has O(N$^7$) scaling. A 240-atom Carbon nanotube was used for the DFT calculation with a PBE/6-31G* level of theory. In total, this creates 3600 basis functions. A smaller system, Pentacene, with 378 basis functions, was used for testing CCSD(T) calculations.

For the molecular dynamics simulations using LAMMPS and GROMACS, we prepared a box of SPC/E water molecules (~100k atoms) in a 10 nm cubic box at room temperature and 1 atmosphere pressure. The long-range Coulomb forces are treated with the particle mesh Ewald (PME) method. The system is pre-equilibrated and each performance test takes 5 min with 2 fs time step in the simulation.

All programs are compiled using the GNU 4.9.3 compiler and its corresponding math library. Math library computations are proven to the main factor for performance [6]. While the XE6/XK7 uses AMD processors, the XC30 uses Intel, processors. For each machine, we chose the most recent Cray libsci math library available (17.09.1 on XC30 and 16.03.1 on XE6/XK7). Cray MPI was used for all applications and all tests. The XE6/XK7 was run exclusively in Extreme Scalability Mode (ESM). This distinction is not applicable on the XC30.

## V. Benchmark Results

### A. NWChem

1. DFT calculations

Using 3600 localized basis functions is about the upper limit that can be routinely used to perform energy minimization or ab initio dynamics using DFT. Thus, our test system well represents actual research usage. Using only half of the cores on each node for large MPI jobs may improve performance by distributing the MPI communication across a larger section of the network and reducing the communication pressure from each individual node. In the case of the XE6, this also allows tying each core to a single floating-point unit. Both full node (using all cores on each node) and half node jobs were tested. The results are summarized in Figure 1).

When comparing each test relative to others within the same machine, the calculation's timing continues to improve up to 64 nodes on both machines. The speedup for both machines in the 1 to 16 node range is about 80~90% every time the node count is doubled, and about 60% in the range from 16 to 32 nodes. At 64 nodes, XE6 only gained ~4% extra speed with respect to the 32-node test. The XC30's improvement at this range was still measured at 25%. The MPI libraries on both machines errored at 128 nodes. When comparing the machines against each other with the same node counts, the XC30 was a clear winner over the XE6. For node counts less than 32, the new XC30 outperforms the old XE6 by about 40%. By the time the node count reaches 64, this disparity has increased to 59%. The timing results indicate both machines are well suited for DFT calculations. Lastly, using half of each node doesn't improve performance in this case.

2. CCSD(T)

CCSD(T) is a computationally intensive calculation. This method is difficult to implement well and very computationally expensive. The 368-basis function calculation we do here is a large calculation and is a good representation of how this "golden standard" method is most commonly used in reality which is a theoretical best value to compare against less expensive methods during method benchmarks.

Do to the limitation of the BigRed2's configuration, our largest test used 128 nodes. As shown in Figure 2, the calculation again scales relatively well on both machines even up to 512 nodes. On 32 nodes, XC30 outperforms XE6 by 32%. Using 64 nodes, the XC30 ran 60% faster. This trend continued, but was attenuated at 128 nodes where the XC30 was 65% faster than the XE6. These timings are a result of not only the stronger processing power of the newer CPU but also the faster interconnect. This was proven by rerunning the jobs using only half of the cores on each node (every two cores on the XE6 share a single floating-point unit). The relative performance was not changed much at larger number of nodes.

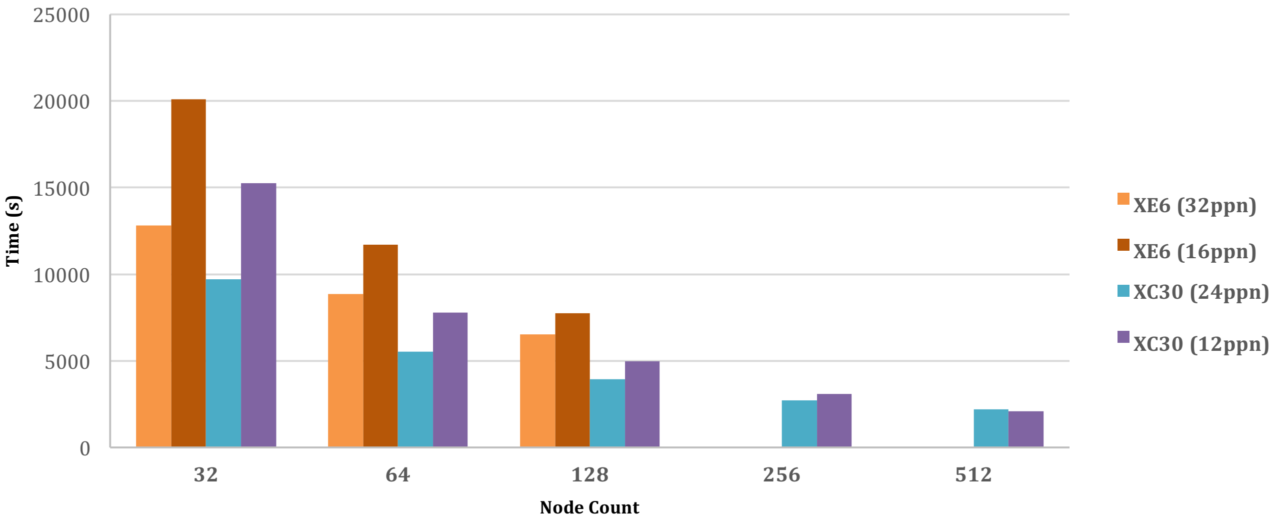


**Figure 2.** Performance of CCSD(T) calculations in NWChem on Cray XE6/XK7 and XC30

## B. GROMACS

1. BigRed2+ (XC30)

In Figure 3(A), tests of pure MPI jobs are presented. In all cases, performance scales well with the increase of processes per node (ppn). The implementation of hyper-threading helps boost the performance under 16 nodes. When the number of nodes increases beyond 16, it starts to hamper the performance. The same conclusion can be drawn from the pure OpenMP model in Figure 3(B). The scalability of pure OpenMP is worse than the pure MPI model. In fact, it reaches a bottleneck for nodes larger than 32. For hybrid MPI-OpenMP situation from Figure 3(C), the performance is much better than pure OpenMP model and is almost comparable to pure MPI model. However, for node numbers larger than 32, pure MPI is faster. For example, a 128-node job using pure MPI (ppn=24) is 82% faster than best-performed pure OpenMP job and 97% faster than hybrid MPI+OpenMP tasks.

Part of the profiling results from the above tests are presented in Figure 4. As the number of nodes increases, MPI communication takes more time and ultimately occupies the majority of the run time. In fact, for 128-node jobs, MPI takes ~92% of the run time for all parallel models. In that case, hyper-threading, significantly hinders the performance, resulting in a 64% reduction in performance when ppn=24.

As shown in Figure 4(C), even though hybrid MPI+OpenMP scheme reduces the total number of MPI ranks, it does not help boost the performance. The reason is two fold. First, the implementation of OpenMP results in run-time overhead. Second, atomic operations in the algorithm can pause other running threads. For the 128-node case, ETC (all the other contributions except for MPI communications and user programs) takes 11% of the run time, much higher than the 1% cost in a pure MPI (ppn=24) job. On the other hand, the reduction of the number of MPI ranks does not actually reduce the communication cost. Using ppn=6 and omp thread=4 only reduces the MPI task cost from 92% to 81%.

**Figure 3.** GROMACS Performance (Cray XC30)

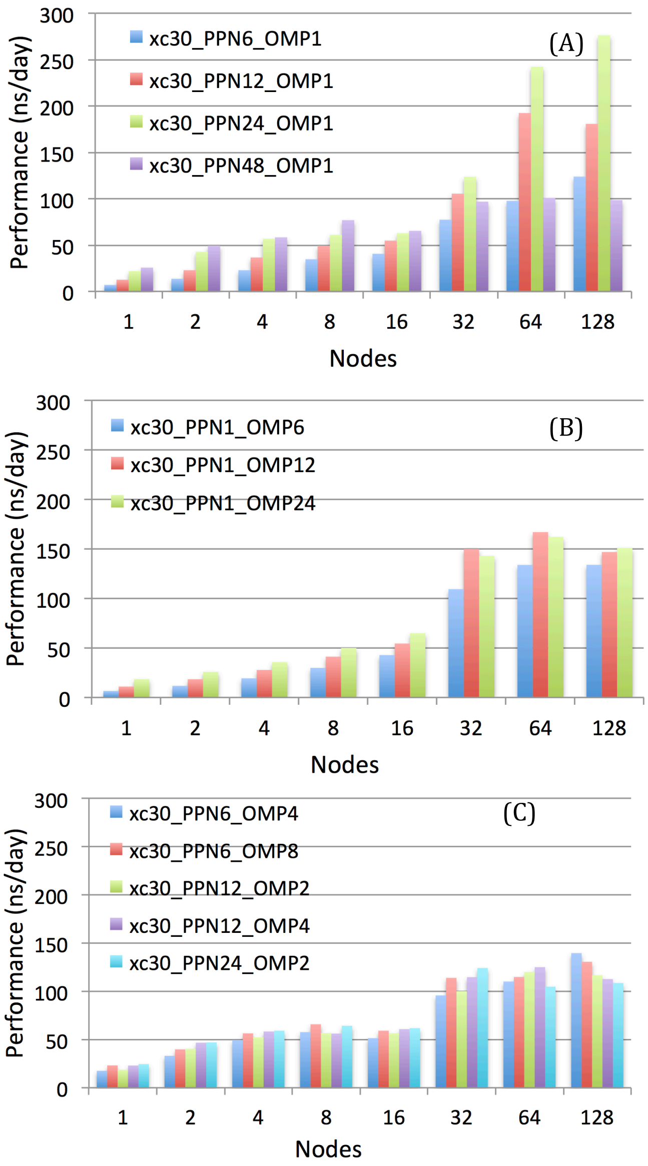
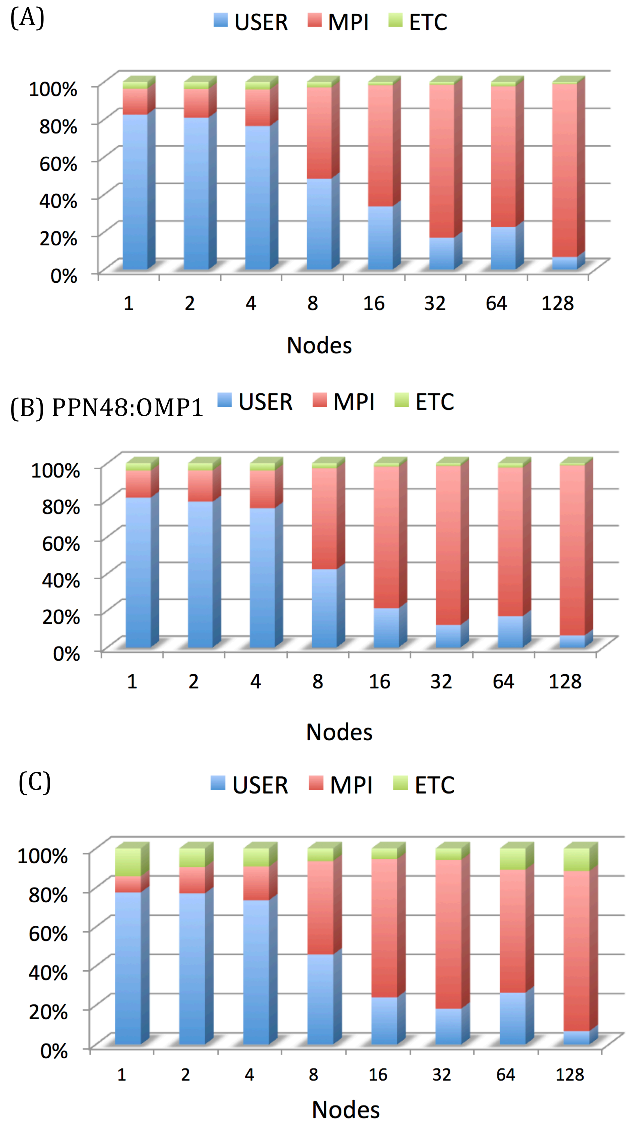


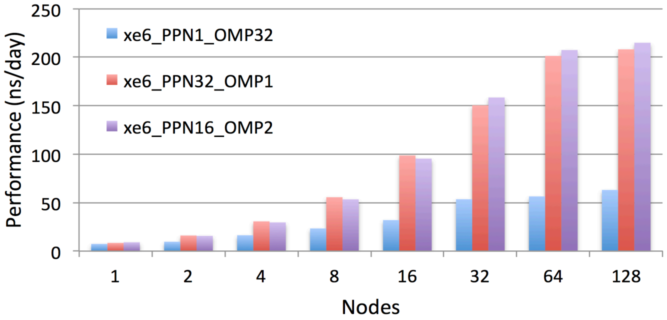
**Figure 4.** GROMACS Profiling (Cray XC30)



2. BigRed2 (Cray XE6/XK7)

Similar to BigRed2+, pure OpenMP jobs perform much worse than pure MPI jobs (Figure 5). Even when using 1 node, it is recommended to run GROMACS in pure MPI ranks. The inner bandwidth speed in BigRed2 is slower than BigRed2+. As a result, it is observed that when using a large number of nodes, the hybrid MPI+OpenMP model is slightly faster than the pure MPI model (~3% increase in 128 nodes).

**Figure 5.** GROMACS Performance (Cray XE6)



On BigRed2 XK7 nodes, the compute mode of the Tesla K20 GPUs is set as Exclusive Processing, so multiple MPI accesses to single GPU context are not allowed. Shown in Figure 6(A), when running on a small number of nodes (less than 16), jobs have better performance with more OpenMP threads. However, when the node number is large, too many threads slow down the performance.

On XK7 nodes, CUDA acceleration greatly boosts performance. Figure 4(B) shows a 4x speed up when CUDA is enabled in 1 node. This gain is reduced as more nodes are used. However, on 128 nodes, CUDA still performs 31% better than the same run with the GPU not utilized.

3. Comparison

Compared with XC30 nodes on BigRed2+, BigRed2 is slower in the benchmarking test even though it has GPU nodes and more CPU cores. On 128 nodes, the best performance of XC30 reaches 276 ns/day. Meanwhile, the XE6/XK7's best results are 214 ns/day and 176 ns/day, respectively.

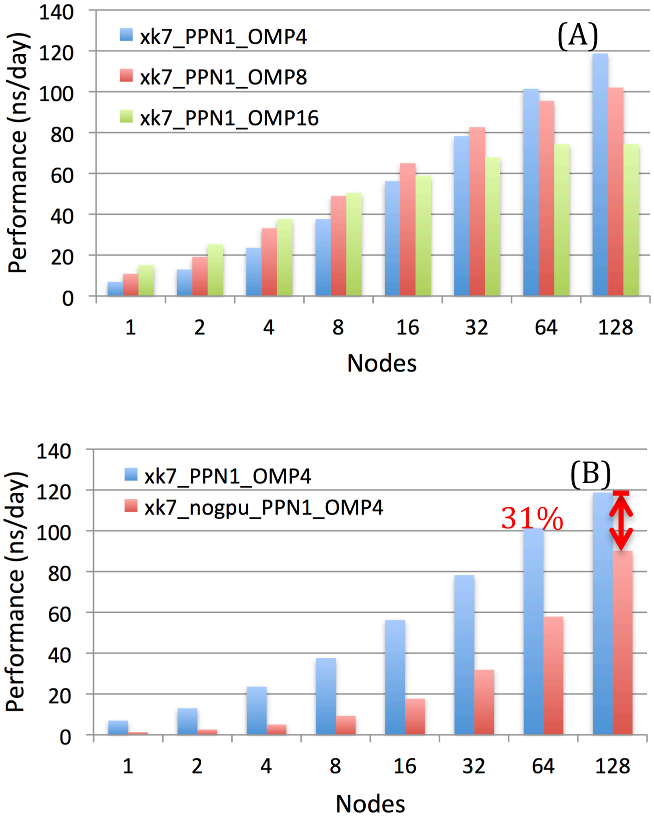
## C. LAMMPS

1. BigRed2+(XC30)

Generally speaking, LAMMPS runs slower than GROMACS for pure water simulations due to the fast SETTLE algorithm used in GROMACS which are specifically targeted at 3-site water molecules. As shown in Figure 7(A), when ppn increases to 24 for 128 node jobs, it starts to lower overall speed. It seems that MPI communication has become the major issue. It is also validated in Figure 7(C) that hybrid MPI+OpenMP runs faster that pure MPI jobs.

Similar in GROMACS, hyper-threading is useful to slightly increase the speed (5%-10%) only when a small number of nodes are used (less than 16). When the number of nodes is large, it is unfavorable and should be avoided.

**Figure 6.** GROMACS Performance (Cray XK7)



2. BigRed2 (XE6/XK7)

Figure 8 shows the performance on the XE6 (BigRed2's pure CPU nodes) using different parallel models. Pure OpenMP jobs are the slowest. Pure MPI jobs are only fast on small numbers of nodes. Hybrid MPI+OpenMP jobs run the fastest for large numbers of nodes since it reduces the MPI communication time. It has a 43% performance boost on 128 nodes compared with a pure MPI job.

To utilize CUDA on the XK7 BigRed2's GPU enabled nodes), only a single MPI process is allowed to access one GPU. However, LAMMPS does not support CUDA+OpenMP model simultaneously with GPU and USER-OMP packages. So only 1 MPI x 1 OpenMP thread is used on each XK7 node for the benchmark test. The results are shown in Figure 9. Enabling CUDA will result in 2x ~ 3x increase in speed. On 128 nodes, the performance reaches 42 ns/day, which is comparable with the peak value of 51 ns/day on XE6 nodes.

**Figure 7.** LAMMPS Performance (Cray XC30)

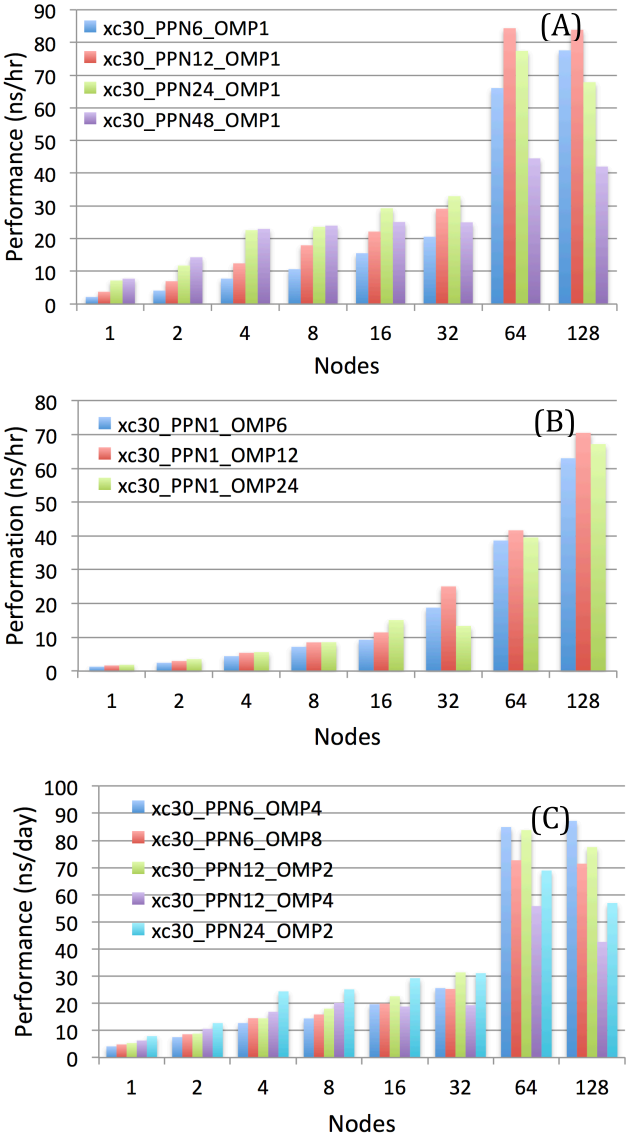


**Figure 8.** LAMMPS Performance (Cray XE6)

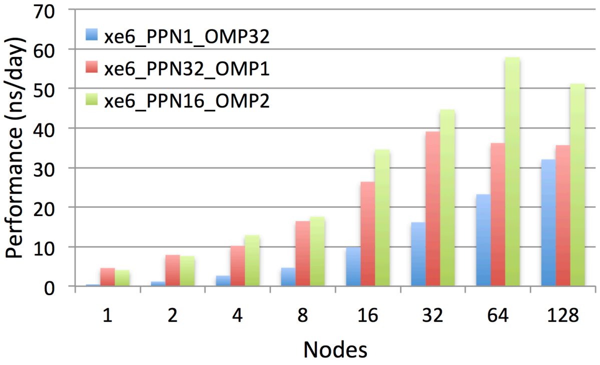
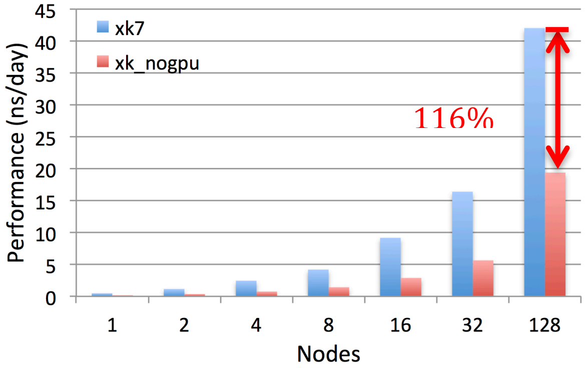


**Figure 9.** LAMMPS Performance (Cray XK7)



3. Comparison

In the benchmarking test of LAMMPS, the peak performance (87ns/day) on BigRed2+ is about 70% faster than BigRed2's best (51ns/day on the XE6). When the MPI communications task is heavier, BigRed2+ has better performance.

## VI. Conclusion

Over the intensive tests of different computational chemistry methods, we conclude these state-of-the-art applications can perform very well using massive parallelism on Cray's HPC systems. The newer XC30 pure CPU machine outperforms the old XE6 by a large margin and even beats GPU equipped XK7 at large node counts. For the molecular dynamics programs where hybrid MPI+OpenMP is available, the hybrid scheme provided slightly better performance.

## Acknowledgement

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute, and in part by the Indiana METACyt Initiative. The Indiana METACyt Initiative at IU was also supported in part by Lilly Endowment, Inc.

## References

[1] https://kb.iu.edu/d/bcqt

[2] https://kb.iu.edu/d/aoku

[3] M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. J. van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, W. A. de Jong, "NWChem: a comprehensive and scalable open-source solution for large scale molecular simulations". Comput. Phys. Commun. 181, 1477 (2010)
https://www.nwchem-sw.org

[4] Hess, B., Kutzner, C., van der Spoel, D. and Lindahl, E. "GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation".
J. Chem. Theory Comput., 4, 435-447 (2008)
https://www.gromacs.org

[5] S. Plimpton, Fast Parallel. Algorithms for Short-Range Molecular Dynamics. J. Comp. Phys., 117, 1-19 (1995)
https://lammps.sandia.gov

[6] J. Deslippe and Z. Zhao. "Comparing Compiler and Library Performance in Material Science Applications on Edison." Cray User Group 2013
https://cug.org/proceedings/cug2013_proceedings/includes/files/pap135.pdf