JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

# Optimization of the Coupled Cluster Implementation in NWChem on Petascale Parallel Architectures

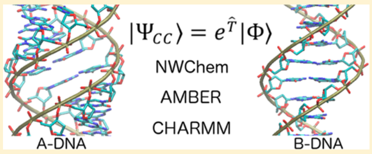
Victor M. Anisimov,*[†] Gregory H. Bauer,[†] Kalyana Chadalavada,[†] Ryan M. Olson,[‡] Joseph W. Glenski,[‡] William T. C. Kramer,[†] Edoardo Aprà,[§] and Karol Kowalski[§]

[†]National Center for Supercomputing Applications, University of Illinois at Urbana−Champaign, 1205 West Clark Street, MC-257, Urbana, Illinois 61801, United States

[‡]Cray, Inc., 380 Jackson Street, St. Paul, Minnesota 55101, United States

[§]Environmental Molecular Sciences Laboratory, Pacific Northwest National Laboratory, P.O. Box 999, K8-91, Richland, Washington 99352, United States

S Supporting Information



**ABSTRACT:** The coupled cluster singles and doubles (CCSD) algorithm in the NWChem software package has been optimized to alleviate the communication bottleneck. This optimization provided a 2-fold to 5-fold speedup in the CCSD iteration time depending on the problem size and available memory, and improved the CCSD scaling to 20 000 nodes of the NCSA Blue Waters supercomputer. On 20 000 XE6 nodes of Blue Waters, a complete conventional CCSD(T) calculation of a system encountering 1042 basis functions and 103 occupied correlated orbitals obtained a performance of 0.32 petaflop/s and took 5 h and 24 min to complete. The reported time and performance included all stages of the calculation from initialization to termination for iterative single and double excitations as well as perturbative triples correction. In perturbative triples alone, the computation sustained a rate of 1.18 petaflop/s. The CCSD and (T) phases took approximately $^3/_4$ and $^1/_4$ of the total time to solution, respectively, showing that CCSD is the most time-consuming part at the large scale. The MP2, CCSD, and CCSD(T) computations in 6-311++G** basis set performed on guanine−cytosine deoxydinucleotide monophosphate probed the conformational energy difference between the A- and B-conformations of single stranded DNA. Good agreement between MP2 and coupled cluster methods has been obtained, suggesting the utility of MP2 for conformational analysis in these systems. The study revealed a significant discrepancy between the quantum mechanical and classical force field predictions, suggesting a need to improve the dihedral parameters.

## 1. INTRODUCTION

The dynamic pace and competitiveness of the modern research environment fuel the need to seek understanding of material properties from the fundamental principles governing atomic and molecular interactions. Computer simulations capable of providing such understanding play an increasing role in the discovery process in modern chemical and biological sciences. With the launch of the Blue Waters supercomputer by the National Science Foundation and the University of Illinois, the academic community gained access to a machine with 22 640 XE6 and 4224 XE7 nodes and peak performance of 13.34 petaflop/s, where one petaflop (PF) is $10^{15}$ floating-point operations. Effective utilization of this computational power is an important challenge facing the research community.

In this work, we focus on performance optimization of the NWChem package on Blue Waters. NWChem is a popular open-source high-performance computing (HPC) computational chemistry package.[1] It receives hundreds of citations every year, according to ISI Web of Knowledge (Copyright 2014 Thomson Reuters). Since accurate quantum mechanical simulations of complex theoretical chemistry problems require petascale resources, even a fractional improvement at such scale can save many thousands of computing hours. The existence of a large NWChem user community further justifies the work on performance optimization.

Application performance is vital to computational study of large molecular systems. In theoretical chemistry, the study of DNA is of particular importance.[2] The seemingly trivial chemical structure of DNA hides a wealth of details that determines the subtle line between stability and evolution of the genetic code, repair and mutation, winding and unwinding of the helix; all of this culminates in the amazing fidelity of DNA replication. Understanding at the atomic level how DNA manages those various functions holds the key to future advancements in biotechnology.[3]

Electronic structure methods serve as the focal point in theoretical studies of DNA due to their ability to provide systematic improvement in the quality of theoretical predictions with increases in the level of theory.[4] Among the recent computational findings is the discovery that a sugar−phosphate backbone predetermines the 3D structure of right-handed Watson−Crick (WC) DNA in A- and B-conformations.[5] These computations revealed that the main characteristics of the WC

**Received:** May 8, 2014
**Published:** September 4, 2014

duplexes, such as the specific regions of sugar−phosphate torsions, sugar ring puckering, and nearly parallel base arrangements, are predefined in the local energy minima of deoxydinucleoside monophosphates (dDMP) representing the minimal fragments of single-stranded DNA.[6] According to those studies, the WC duplex is constructed in such way that the local energy minima of the free single strand match the preferable conformations of the duplex.[7] The studies also explain how the sequence dependence in WC DNA originates from the directionality and preferred values of sugar−phosphate torsions, combined with the difference of purines from pyrimidines in ring shape.[8,9] Since no such sequence dependence can be observed in dDMPs that correspond to other DNA conformations (e.g., Z-family and Hoogsteen duplexes) and because different DNA conformations have a very different biological function,[10] these facts reinforce the need for accurate description of the energy profile of single-stranded DNA.

Among electronic structure methods, the coupled cluster (CC) family of methods[11−13] is the most powerful and accurate level of theory because it efficiently accounts for the dominant portion of electron correlation effects. A thorough description of CC theory can be found in recent reviews.[14,15] CC methods are implemented in a number of popular HPC packages, including NWChem,[16] GAMESS US,[17] MOLPRO,[18] ACES III,[19] PSI,[20] Q-Chem,[21] and PQS,[22] to name a few. NWChem incorporates the work of Kobayashi and Rendell,[23] which will be referred hereinafter as Rendell's code. This method is based on the algorithm originally proposed by Scuseria.[24] Due to the use of spin-adapted formalism,[25] the method is only applicable to restricted Hartree−Fock (RHF) reference and needs a smaller number of cluster amplitudes. Because of that, Rendell's code exhibits superior performance and moderate memory requirements. Since computational chemistry encompasses a large number of ground-state applications, including binding problems,[4,26−28] geometry optimization,[6] potential energy scans,[29] etc., in which the RHF wave function provides an adequate reference function, having an economical CC method is extremely important so an efficient computational method may be used to pursue those numerous science problems.

Due to the high computational cost of CC methods, access to petascale resources is vital for the computation to be accomplished in a reasonable amount of time. Since nearly all interesting chemical and biological processes proceed on at least the mesoscopic scale, the size of predictive molecular models in terms of number of atoms must be large. Because of that, quantum mechanical computation of small systems gradually becomes of limited practical utility in chemistry. In the present work, we are interested in the application of conventional CC methods to systems encountering about 1000 basis functions.

One of the major uses of applying high-level methods to large systems is to validate the accuracy of computationally less demanding approximate methods. For instance, linear scaling CC methods can perform CC computations at significantly smaller computational cost than their conventional counterparts.[30−38] Since the linear scaling theories involve approximation, without validation the accuracy of the implemented methods would remain vastly unknown. Speeding up the conventional CC methods for validation of linear scaling theories is an important component in the development of approximate electronic structure methods for large molecular systems.

Among CC methods, CCSD(T) is a highly versatile method that involves iterative accounting for single and double excitations (CCSD)[39] and perturbative accounting for triple excitations (T) using the previously computed CCSD amplitudes.[40] The computational cost of CCSD and (T) methods in terms of floating point operations (FLOP) increases as $N^6$ and $N^7$, respectively, where $N$ designates system size.

In addition to its high computational cost, CCSD(T) requires terabyte-scale intermediate data storage for chemically interesting applications. An array of CCSD amplitudes in RHF formalism using double precision consumes $4N_{\mathrm{vir}}^2N_{\mathrm{occ}}(N_{\mathrm{occ}} + 1)$ bytes, where $N_{\mathrm{occ}}$ is the number of correlated occupied molecular orbitals and $N_{\mathrm{vir}}$ is the number of virtual molecular orbitals. Electron repulsion integrals also require large storage, and depending on implementation, the CC program may need several other large arrays. The traditional solution is to offload these arrays to a hard disk, as they altogether cannot fit in the available random access memory (RAM) on the compute node. The downside of this approach is increased input/output (I/O) traffic that should be addressed in order to make the computations practical.

With a favorable ratio of computation to communication, the (T) part in Rendell's code scales well to a very large number of nodes allowing perturbative triples to be efficiently computed. The CCSD part of Rendell's code has a complex communication pattern and limited scalability to only a fraction of the (T) scale. Additionally, it usually takes about 20 iterations for CCSD amplitudes to converge in NWChem with the required precision of $10^{-6}$ in the norm of the residual vector, which adds a factor of 20 to the standard cost of CCSD in terms of time to solution. This factor will further increase if a greater number of iterations will be required, particularly when requesting tighter convergence. Additional cost factors appear due to a communication bottleneck when using a large number of cores. The iterative nature of CCSD combined with limited scalability of the code actually makes the CCSD part equally if not more time-consuming than the (T) part. This is why the performance of CCSD(T) computation cannot be measured by the performance of the (T) part alone despite the dominant floating-point count of the latter.

The critical performance-limiting factor in most CC codes is I/O. This requires implementing efficient strategies to reduce or eliminate the volume of file reads and writes. In NWChem, this problem is addressed by adopting the Global Arrays (GA) parallelization model.[41,42] GA provides the mechanism for storing large data arrays in the distributed RAM of compute nodes. This often means that a large number of compute nodes is necessary to conduct any meaningful computation. Read and write operations on the remote memory storage are accomplished in GA via *ga_get*() and *ga_put*() functions, respectively. The data are accessed by array indices, so the application programmer does not need to worry about the physical location of data when writing the computer code.

Despite the undisputable utility provided by GA, the distributed memory paradigm of GA has a scalability limitation. GA uniformly distributes the global arrays across all compute nodes. Increasing the number of compute nodes by a factor of $M$ increases the intensity of transmitted messages by a factor of $M^2$ for the same problem size. Therefore, in large-scale computations employing the GA model, network communication soon becomes the performance-limiting factor. Support for a nearest-neighbor communication model that would improve scalability is presently not included in GA.[41,42] This illuminates the anticipated improvements in GA since the future of HPC is aligned with linear scaling algorithms.

In the present work, we investigate the bottleneck in Rendell's CCSD code[23] and offer a necessary solution.

## 2. THEORY

To give a brief overview, CC is based on the exponential ansatz to incorporate the effects of electron correlation into the wave function of the computed system,

$$|\Psi_{\mathrm{CC}}\rangle = \mathrm{e}^{\hat{T}}|\Phi\rangle$$

where $|\Psi_{\mathrm{CC}}\rangle$ is the target CC wave function of the system, $|\Phi\rangle$ is a reference function, typically chosen as a Hartree–Fock determinant, and $T = T_1 + T_2 + ...$ is the cluster operator expressed as a sum of its many-body singly- ($T_1$), doubly- ($T_2$), and so forth excited components defined as

$$T_1 = \sum_{i,a} t_i^a E_i^a$$

$$T_2 = \frac{1}{2}\sum_{ij,ab} t_{ij}^{ab} E_i^a E_j^b$$

$$\ldots$$

where $E_i^a$ are the unitary group generators and $t_i^a$, $t_{ij}^{ab}$ are singly- and doubly excited cluster amplitudes.[43] Throughout this work, we will use symbols $i$, $j$, $k$, $l$ to denote the indices of occupied molecular orbitals (MO), $a$, $b$, $c$, $d$ to label the indices of virtual MOs, and letters of Greek alphabet $\alpha$, $\beta$, $\gamma$, $\delta$ to denote the indices of atomic orbitals (AO).

Traditionally, CC computation is I/O bound due to massive manipulation with the large data set of 2-electron integrals in the MO basis. Out of those integrals, the largest set is represented by integrals with three or four virtual MO indices. In Rendell's code, 1- and 2-virtual orbital index integrals in the MO basis are stored in the distributed aggregate memory, whereas 3- and 4-virtual orbital index integrals are computed on the fly in "direct" mode in their AO-basis representation. The details of the AO-driven CCSD formalism are available in the original publication.[23] Here, we review only the most time-consuming contraction $B_{cd}^{ab}\tau_{ij}^{cd}$, labeled D3 according to Rendell's nomenclature, where

$$B_{cd}^{ab} = (ac|bd) - (ac|dk)t_k^b - (bd|ck)t_k^a$$

$$\tau_{ij}^{cd} = t_{ij}^{cd} + t_i^c t_j^d$$

The original D3 contraction needs integrals and amplitudes in the MO basis. In Rendell's code this contraction is reformulated in the AO basis. The resulting computational algorithm is presented in Scheme 1, as described in ref 23.

In the D3 contraction, computation and consumption of AO integrals are schematically arranged in four enclosed loops defined by shell indices. Integrals are computed in blocks of shell indices using the advantage of basis set contraction, which speeds up the calculation of 2-electron integrals belonging to the same shell. Partially AO-indexed amplitudes $[\tau_{ij}^{\gamma\delta} \pm \tau_{ij}^{\delta\gamma}]$ are stored in GA memory and retrieved in blocks determined by the shell indices. We will call this array ST2 based on the name assigned to it in the program code. The AO integral blocks are processed in parallel, and the result is accumulated.

Although in this formalism the same AO integrals have to be computed four times, this repetition does not limit performance. As mentioned by Kobayashi and Rendell, the cost of AO-integral calculation is minor.[23] Their assessment is even truer with today's powerful CPUs. Since CC computation is typically communication bound, it causes the CPU power to be underutilized. Therefore, recomputing some data locally instead of requesting them from a remote location is a useful strategy to reduce the communication traffic and improve the application performance.

**Scheme 1. Computation of D3 Term Using 2-Electron Integrals and Amplitudes in the AO Basis[a]**

```
do $\alpha_{sh} = 1, N_{sh}$
   do $\beta_{sh} = 1, \alpha_{sh}$
      do $\gamma_{sh} = 1, N_{sh}$
         do $\delta_{sh} = 1, \gamma_{sh}$

            compute shell block of AO integrals $(\alpha\beta|\gamma\delta)$, $(\alpha\gamma|\beta\delta)$
            get $[\tau_{ij}^{\gamma\delta} \pm \tau_{ij}^{\delta\gamma}]$ block from distributed GA memory
            accumulate $[(\alpha\gamma|\beta\delta) \pm (\alpha\delta|\beta\gamma)][\tau_{ij}^{\gamma\delta} \pm \tau_{ij}^{\delta\gamma}]$

         enddo
      enddo
   enddo
enddo
```

[a] $N_{\mathrm{sh}}$ is number of AO shells, and $\alpha_{\mathrm{sh}}$, $\beta_{\mathrm{sh}}$, $\gamma_{\mathrm{sh}}$, and $\delta_{\mathrm{sh}}$ are shell indices. The work in the innermost loop is executed in parallel.

Although CC is a very robust theory, its high computational cost fuels the search for more affordable alternatives. The highly popular and least expensive technique for treatment of electron correlation is second-order Møller–Plesset (MP2) perturbation theory.[44] For many equilibrium properties, where the Hartree–Fock determinant is a good approximation of exact wave function, MP2 provides a reliable solution comparable in quality to CC results.[45,46] The relationship between perturbation and CC theory is established by the linked-cluster theorem,[47−51] where CC theory can be viewed as an infinite summation of the perturbative expansion.[14] This relation can be understood from the analysis of correlation energy $E_{\mathrm{corr}}$ for MP2 and CCSD,[52]

$$E_{\mathrm{corr}}^{\mathrm{MP2}} = \frac{1}{4}\sum_{ijab}\frac{|\langle ij\,||\,ab\rangle|^2}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}$$

$$E_{\mathrm{corr}}^{\mathrm{CCSD}} = \frac{1}{4}\sum_{ijab}\langle ij\,||\,ab\rangle(t_{ij}^{ab} + 2t_i^a t_j^b)$$

where $\varepsilon_k$ is the energy of molecular orbital $k$ and $\langle ij||ab\rangle$ is the antisymmetrized two-electron integral in the MO basis. In CCSD energy expression, neglecting the single excitations and taking doubly excited amplitudes at their fixed first-order value,

$$t_{ij}^{ab(1)} = \frac{\langle ij\,||\,ab\rangle}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}$$

gives the MP2 correlation energy.[52,53]

The qualitative agreement between MP2 and CCSD is not always guaranteed and needs to be validated in each particular case since the overall theoretical differences in the methods are significant. Specifically, the MP2 level includes double excitations, CCSD additionally includes disconnected triple and quadruple excitations, and CCSD(T) incorporates connected triples, which add the well-known robustness to the CCSD(T) level of theory. The iterative character of singly and doubly excited CCSD amplitudes employed in the CCSD(T) approach is also a factor contributing to the well-known balance between the correlation effects stemming from various ranks of excitations in the energy expansion. In this work, the application of CCSD(T) to full-size molecular systems will clarify whether MP2 is a satisfactory level of theory for conformational studies of single-stranded DNA.

## 3. METHODS

In this work, all quantum mechanical computations are performed in the gas phase using the standard 6-311++G** basis set[54,55] in the NWChem package.[1] Floating-point operations are counted for performance analysis by using the PAPI 5.0.1 library.[56] Studied molecules are guanine (G), guanine−guanine stack (GG-stack), and guanine−cytosine deoxydinucleotide monophosphate in A- and B-conformations neutralized by a sodium cation, abbreviated GC-dDMP-A and GC-dDMP-B, respectively. The choice of samples is dictated by the need to provide a gradual increase in the problem size, and the choice of dDMPs in A- and B- conformations aims at evaluating the utility of various computational methods for conformational studies of DNA. Initial geometries of GC-dDMP in A- and B-conformations are obtained from the crystal structures with PDB id 348D[57] and 1CGC,[58] respectively. Prior to energy computations, the geometry of all molecular systems was subjected to MP2/6-311++G** optimization under the default convergence criterion. Optimized structures of the studied compounds are displayed in Figure 1.

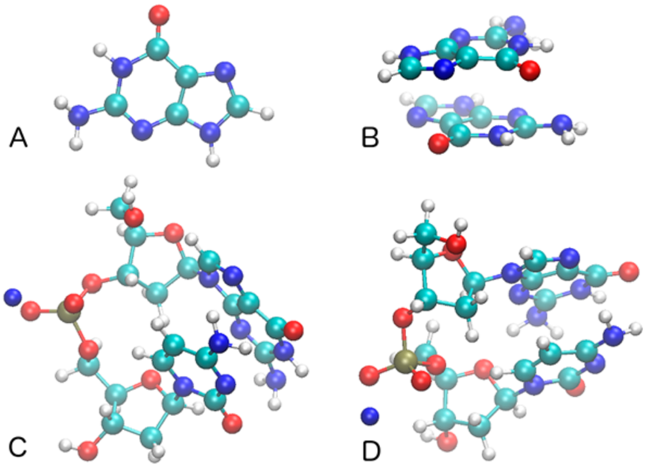


**Figure 1.** MP2/6-311++G** optimized geometries of guanine (panel A), guanine−guanine stack (panel B), GC-dDMP-A neutralized with a sodium cation (panel C), and GC-dDMP-B neutralized with a sodium cation (panel D). Images were created using VMD software.[62]

To manage the resource consumption, CCSD(T) computations were performed in two steps. First, a CCSD calculation is performed and the amplitudes are written to a file. Next, a (T) calculation is performed by restarting the job from the saved amplitudes. In the CCSD part, the convergence threshold was set to $10^{-6}$ in the norm of the residual vector. AO integrals were computed with the precision of $10^{-12}$ Hartree. The frozen core approximation was adopted in MP2 and CC computations. Classical force field single-point energy calculations in a vacuum were performed with Amber parm99[59] and CHARMM c36[60] force fields by using the NAMD program package.[61] The nonbonded cutoff was set to 20 Å in the empirical calculations.

## 4. PERFORMANCE OPTIMIZATION

Since CCSD is the least efficient part in the CCSD(T) calculation, we focused on improving the CCSD implementation. By measuring the time consumed by different parts of the code we confirmed the previous finding that CCSD calculation spends most of its time in D3 contraction.[23] Detailed analysis of the code revealed that the performance bottleneck of the D3 term comes from the copious amount of *ga_get*() communication requests made in this part of the code to retrieve the ST2 data from distributed memory. The number of such requests can be evaluated from the code snippet shown in Scheme 1. One can see that ST2 data are requested $[N_{sh}(N_{sh}+1)/2]^2$ times from GA memory per CCSD iteration. This is an $O(N^4)$ dependence on the problem size. With the guanine monomer, measurements indicate that the ST2 array is accessed 75 588 748 times during a single CCSD iteration. With the guanine dimer (GG-stack), the number of ST2 requests grew to 1 081 653 855. This represents a 14.3−fold increase in the communication volume. A small deviation from the theoretical factor of 16 comes from the code employing Schwarz inequality to skip the calculation of negligibly small AO-integrals.

The quartic growth in the number of communication requests with the increase in problem size is a serious issue for code scalability. Under such a condition, timely processing of the network operations would be challenging for any interconnect. Indeed, timers placed on *ga_get*() requests showed that most of the D3-term processing time is spent waiting for ST2 data to arrive. This delay happens from interconnect saturation, that is, the interconnect's inability to process a huge number of small data packets in a timely manner. This also explains why the use of nonblocking *ga_nbget*() instead of blocking *ga_get*() in that part of the code was ineffective, since there was not enough computational work

**Table 1. Timing of a Single CCSD/6-311++G** Iteration in NWChem 6.1**[a]

| molecule | $N_{occ}$ | $N_{vir}$ | ST2 size | GA ST2 calls | Ver | nodes/cores | ST2 get() | CCSD time |
|---|---|---|---|---|---|---|---|---|
| guanine | 39 | 238 | | 75 588 748 | old | 5/80 | $6 \times 10^{-5}$ | 130 |
| guanine | 39 | 238 | 232 | | new | 5/80 | | 68 |
| GG-stack | 78 | 476 | | 1 081 653 855 | old | 200/3200 | $1 \times 10^{-3}$ | 905 |
| GG-stack | 78 | 476 | | 1 081 653 855 | old | 200/2400 | $1 \times 10^{-3}$ | 898 |
| GG-stack | 78 | 476 | 3691 | | new | 200/1600 | | 171 |
| GC-dDMP-B | 150 | 892 | | 11 329 966 275 | old | 1100/13200 | $5 \times 10^{-3}$ | 8408 |
| GC-dDMP-B | 150 | 892 | 44060 | | new | 1100/1100 | | 4319 |

[a] "ST2 size" is memory in MB per parallel task for the ST2 array; "GA ST2 calls" is the number of ST2 requests made to GA distributed memory; "Ver" is version of NWChem 6.1, old−original, new−modified; "nodes/cores" is the total number of XE6 nodes/cores used in the job; "ST2 get()" is the average time to process a single *ga_get*() request to a distributed ST2 array by each parallel task in seconds; "CCSD time" is the CCSD single iteration time in seconds.

between *ga_get*() requests to overlap communication and computation and avoid hitting the interconnect latency limit.

To mitigate the above issue, we replicated the ST2 array on each compute process, thus completely eliminating the need to access this array through the network. The drawback of this approach is the increased memory requirement per compute process. On Blue Waters, Cray XE6 compute nodes are equipped with 64 GB RAM. This made it possible to perform conventional CCSD(T) computation of a GC-dDMP system containing 1042 basis functions, which is a large problem size by CC standards and the largest studied in this work. The ST2 array in Rendell's code has the size of $4N_{\mathrm{occ}}^2N_{\mathrm{bf}}(N_{\mathrm{bf}}+1)$ bytes, where $N_{\mathrm{occ}}$ and $N_{\mathrm{bf}}$ are the number of correlated occupied orbitals and basis functions, respectively. To give an example, guanine, guanine−guanine stack, and GC-dDMP need 232, 3691, and 44 060 MB of memory storage, respectively, for ST2 arrays, as shown in Table 1.

On Blue Waters, an XE6 node has 16 floating-point cores. Although the available RAM limits the number of cores per node that can be used in the new code, this limitation proved to be noncritical. The old CCSD code in NWChem cannot utilize all 16 cores per node when running at scale due to a drastic increase in communication. Therefore, before starting a production run at scale, it is always prudent to test how many cores (parallel tasks) per node would maximize the application performance. This situation is not unique to NWChem and could be seen with any communication intensive code. To keep the comparison fair, we ran the tests on the same number of nodes while giving the freedom to the codes to use whatever number of cores they can beneficially utilize.

From the tests on GG-stack, we found that the old code showed the best performance when using 12 cores per node, while the new code could utilize only 8 cores per node. Nevertheless, the new code gave a 5-fold speedup over the old code when running both jobs on 200 nodes (Table 1). This means that the new code is 5-fold more efficient than the old code on 200 nodes with the workload of 554 basis functions.

A 2-fold speedup is observed for a single guanine molecule (Table1), which is the smallest case (277 basis functions) considered in this work. Both the old and new codes beneficially utilize 16 cores per node. When executed on 5 nodes, the new code is twice as fast as the old one.

A nearly 2-fold speedup was also obtained for GC-dDMP with 1042 basis functions, the largest molecule that we studied (Table 1). In this case, the old code showed the best performance when using 12 cores per node. In the new code, we used a single core per node and got the speed up factor of 2 in CCSD iteration time when running both codes on 1100 nodes.

The varying degree of speedup from 2- to 5-fold depending on the problem size could be explained as follows. For the small system (guanine, 277 basis functions, Table 1) the number of requests to the ST2 array via GA calls is minimal and is within the throughput capacity of the interconnect. This is confirmed by the obtained near-zero processing time of a *ga_get*() request to the ST2 data array (Table 1). Therefore, the speedup in the new code is due to faster CPU-to-local-memory vs CPU-to-remote-memory communication.

In the intermediate-size test, a GG-stack with 554 basis functions, the number of *ga_get*() requests is substantially increased (see Table 1). This slowed down the processing time of the *ga_get*() function to $10^{-3}$ seconds (see Table 1) due to interconnect saturation, whereas the normal processing time should be on the order of microseconds. This communication bottleneck reduces the performance of the old code. Switching to the new code eliminates the communication issue and gives the remarkable 5-fold boost in performance.

In the largest case of GC-dDMP with 1042 basis functions, interconnect saturation severely impacts performance of the old code and the processing time of *ga_get*() function slowed down further to 5 × $10^{-3}$ seconds (Table 1). Switching to the new code, we could utilize only a single core per node due to the memory limit. Therefore, we see only a 2-fold speedup on 1100 nodes.

Improving code performance on small-to-medium node counts is only a part of the anticipated improvement. To improve time to solution, the code should perform well on large node counts as well. Therefore, it is interesting to compare how well the old and new codes behave on different node counts in a strong scaling test. Figure 2 shows a plot of the speedup factor as a function of number of nodes in a CCSD/6-311++G** calculation of GC-dDMP-B when measuring a single iteration time. For this problem size, the old NWChem CCSD code can maximally utilize 2000 XE6 nodes. After this limit is reached, adding more nodes to the job does not improve the calculation time. The old code running on 1100, 2000, and 7000 nodes completed a single CCSD iteration in 8408, 6618, and 8276 s, respectively. Still, the ability to utilize 2000 nodes is not a bad result considering the complexity of the computations.

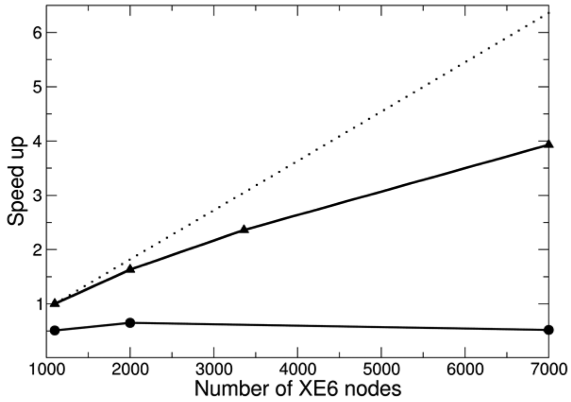


**Figure 2.** Scaling of the old (circles) and new (triangles) NWChem 6.1 codes on a single CCSD/6-311++G** iteration time for GC-dDMP-B. Dotted line represents ideal scaling.

The scaling limitation of the old code has been overcome in the new NWChem code. The new code executed on 1100, 2000, 3360, 7000, and 20 000 nodes finished a single CCSD iteration in 4319, 2649, 1829, 1099, and 776 s, respectively. Remarkably, the best result of 6618 s for a single CCSD iteration obtained by the old code on 2000 nodes has been improved to 776 s by using the new code on 20 000 nodes. This is an 8.5-fold speedup as a result of the improvement in code scalability.

## 5. SUSTAINED PETAFLOP PERFORMANCE OF NWCHEM ON BLUE WATERS

The complexity of modern HPC systems makes it hard to project how well the machine is suited for a particular application. Several popular benchmarks have been around for a while due to their various strengths and utilities. Among the best known are Linpack[63] and SPECfp.[64] Due to their multidecadal age, they hardly represent the breadth of modern science applications. To obtain the full picture, Blue Waters employs the SPP metrics[65] to measure the actual performance on real applications in cosmology, climate, seismology, material sciences, and elementary particle physics, among other fields. In addition to the requirement to use real science applications in the SPP test, the total time to solution has to include all steps involving I/O, initialization, production, and termination, and run on at least $^1/_5$ of the machine.[65,66]

To obtain a realistic assessment of performance of CCSD(T) computation in NWChem on Blue Waters, we took a single-point energy computation of GC-dDMP-B at the CCSD(T)/6-311++G** level of theory. Both (T) and CCSD portions of CCSD(T) calculation were executed on 20 000 XE6 nodes. This represents 88% of the total number of XE6 nodes on Blue Waters. To save on compute time in the dedicated mode, only a single CCSD iteration was performed at the full machine scale. This job took 1214 s to complete. The difference between the wall clock time (1214 s) and a single CCSD iteration (776 s) is taken as the cost of initialization, I/O, and termination (438 s). The amplitudes are written to disk only once at the end of the CCSD job. From a separate CCSD computation on the same input data in a nondedicated machine run, we found that the CCSD job needs 18 iterations to converge. From these data, we obtained the total CCSD time on a dedicated machine by multiplying the single CCSD iteration time by a factor of 18 and adding the previously identified cost of initialization, I/O, and termination (438 + 776 × 18 = 14 406 s). This synthetic time is presented in Table 2 as CCSD time to solution on 20 000 nodes.

**Table 2. Sustained Petaflop Performance of the CCSD(T)/6-311++G** computation of GC-dDMP-B performed on 20 000 XE6 nodes on Blue Waters**[a]

| method | time (s) (20 000 nodes) | GFLOP count | performance, PF/s |
|---|---|---|---|
| CCSD | 14 406 | 195 796 351 | 0.01 |
| (T) | 5024 | 5 948 249 197 | 1.18 |
| CCSD(T) | 19 430 | 6 144 045 548 | 0.32 |

[a] Reported time is the total time to solution, which includes all parts of the computation.

The (T) job was submitted after the CCSD part completed. Reported in Table 2 is the full time of the (T) job from initialization until termination and includes reading the CCSD amplitudes from disk, performing integral transformation on the MO basis, and computing triple excitations. The (T) computation utilized 8 cores per XE6 node, whereas the CCSD job used 1 core per node. The FLOP count of the CCSD and (T) jobs was obtained by calling the PAPI library from within the NWChem code. The performance of the whole CCSD(T) computation as well as that of its individual parts was obtained by dividing the FLOP count by the total time spent in the job, as reported in Table 2.

According to the recorded data, the new code conducts the complete conventional CCSD(T) computation at the average performance rate of 0.32 PF/s for the duration of 5 h and 24 min. This result is quite remarkable in the sense that there are no analogs to which we can compare it. In the (T) part, NWChem maintained 1.18 PF/s for 1 h and 24 min. This is the first time that the (T) computation surpassed the 1 PF/s mark, with NWChem systematically leading in this area.[27] The CCSD part performs significantly slower than the (T) part; however, including it in the whole picture is necessary in order to obtain a realistic performance figure for CCSD(T) computation.

When comparing the relative computational costs of CCSD and (T) parts, it is customary to refer to their $O(N^6)$ and $O(N^7)$ scaling in terms of FLOP count as a function of number of basis functions. This definition indeed works well for small to moderate problem sizes in small node-count simulations. However, the situation changes when venturing to large problem sizes and large-scale computations where additional factors start determining the application time to solution. At scale, computational cost cannot be used as a synonym of FLOP count because on modern hardware architectures the computational throughput only partially depends on the peak floating-point capacity of the processing unit. Limited memory bandwidth, cache misses, latency, and bandwidth of the interconnect, performance of file system operations, and efficiency of parallelization all affect the application performance in a nontrivial manner. Taking that into account, the only useful definition for "computational cost" is the price in terms of node-hours to be paid for the computation to be completed. Referring to node-hours vs core-hours reflects the common practice of applications using the entire node.

This definition highlights a few important limitations. FLOP count is a useful measure of computational cost when minor modifications are applied to the algorithm so the computational cost before and after modification can be quantitatively accessed. For algorithms like CCSD and (T) that are coded very differently, FLOP count as a measure of relative computational cost is impractical and not informative.

Speaking about computational cost in terms of FLOP count unnecessarily obscures the information. As an illustration, the reader can see from Table 2 that CCSD consumes $^3/_4$ of the total time, whereas the (T) part takes $^1/_4$ of the total time to solution on the same number of nodes. This highlights the issue that the cost of computation is measured not in FLOPs but in the node-hours necessary to complete the work and that these two are not equivalent. Referring to the multidimensional complexity of computational processes, CCSD should be viewed as the most computationally expensive part in CCSD(T) simulation based on its longer time to solution. This definition places an emphasis on the need to put further effort into performance optimization of the CCSD part, which is effectively the bottleneck of CCSD(T) calculations on large node counts. While the issue impacting performance is simple to state, there is no clear-cut solution to this problem. We hope that the current implementation will provide a starting point for the anticipated follow-up studies.

## 6. PROBE OF CONFORMATIONAL PROFINE OF GC-DDMP

Complex computational chemistry problems often require accurate description of electron correlation effects.[4] Among the important computational problems is the understanding of conformational variability of Watson−Crick DNA.[5] DNA can be found in two major A- and B-conformations in nature (Figure 3). These structures originate from the ability of the sugar backbone to assume two distinct conformations, as depicted in Figure 4.

In hydrophilic environments, DNA primarily exists in B-conformation, whereas it adopts A-conformation in hydrophobic environment.[67] The transition between these con-

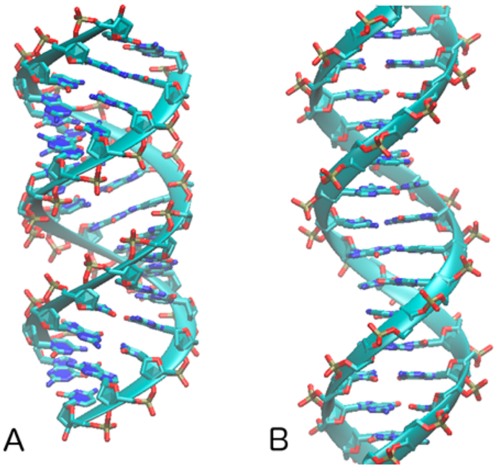


**Figure 3.** Watson−Crick DNA in A- (panel A), and B- (panel B) conformations.

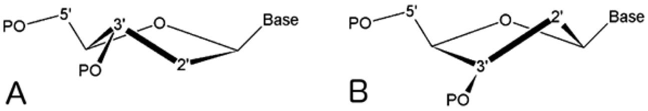


**Figure 4.** Sugar puckering in A- (panel A) and B- (panel B) conformations of DNA.

formations is extensively studied in the literature,[68] and the role of the sugar−phosphate backbone in the DNA structure is described in the Introduction. One of the purposes of this work is to find a reliable level of theory that would allow study of the conformational profile of DNA. Since A-DNA is more stable in hydrophobic environments than B-DNA, it is natural to expect that the gas phase will favor A-DNA more than B-DNA. Indeed, when extracting a GC-dDMP fragment in A- and B-conformations from experimental X-ray data and relaxing its geometry by the MP2/6-311++G** method in the gas phase we get exactly the anticipated result: that the A-DNA fragment is favored over the B-DNA fragment in the gas phase by −1.9 kcal/mol (see Table 3).

Using the MP2 level of theory for computing the potential energy surface is a typical strategy for optimization of torsion parameters in classical force fields. The high cost and large volume of the necessary computations primarily determine the level of theory that can be deemed practical. MacKerell and co-workers used HF/6-31+G* and MP2/6-31+G* theory levels in the optimization of torsion parameters in a CHARMM27 force field,[60] which is presumably the most accurate force field to date for DNA.[69,70] Similar optimization was undertaken in an AMBER98 force field for DNA and included MP2/6-31G* calculations.[71] The recent dihedral parameter fitting in CHARMM employed CCSD(T)/cc-pVDZ calculations on small molecules.[29]

An important question is whether we can trust the MP2 method in the energy analysis of A- and B-DNA conformations. This issue can only be resolved with the help of CCSD(T) computations. Despite limited facility of the MP2 method to describe electron correlation effects, its ability to provide molecular structure of reasonable quality and in good agreement with X-ray data is well-known.[72,73] Therefore, we can use MP2 optimized geometry and perform single-point energy computation by using the CCSD(T) method.

As shown in Table 3, the CCSD(T) computation predicted that the A-conformation of GC-dDMP is favored over the B-conformation by −1.2 kcal/mol, reasonably supporting the MP2 data. This suggests that, despite its limitations, MP2 is an acceptable method for conformational energy calculation of A- and B-DNA fragments. The close agreement obtained between the CCSD and CCSD(T) energies indicates that triple excitations, although important, do not drastically alter the energy value in this particular case. Of course, more computations are necessary in order to derive a foolproof conclusion. Nevertheless, the obtained data support the utility of the MP2 method in conformational analyses of DNA.

Having the ability to reliably compute the conformational energy of A- and B-DNA fragments is an important step toward improving classical force field potentials, which are extensively used in DNA simulations.[74] Using the CCSD(T) energies we can evaluate the ability of the popular CHARMM and AMBER force fields for DNA to correctly reproduce the relative energy between the GC-dDMP conformations.

In these tests, CHARMM showed the correct trend, where an A-DNA fragment is favored over a B-DNA fragment in gas phase. However, the CHARMM prediction of −5.6 kcal/mol (Table 1) is much larger than the CCSD(T) value. The AMBER force field predicted that the A-conformation is disfavored over the B-conformation by 4.1 kcal/mol, in disagreement with the CCSD(T) data. In summary, both force fields failed to show quantitative agreement with the CCSD(T) data, although the CHARMM force field performed slightly better than the AMBER force field. To correct the deficiencies, one needs to reoptimize these empirical potentials against the high-level ab initio data.

In a force field optimization study, which is beyond the scope of the present work, one should ideally use a larger basis set than 6-311++G**, if possible. Doing that at CCSD(T) level will be very computationally expensive. In any case, the currently employed 6-311++G** basis set is much better than the basis sets typically used in the optimization of popular biomolecular force fields.[29,60,71] Using a larger basis set would certainly be feasible with the MP2 method. Our results indicate that MP2 could be used to generate a reliable detailed conformational profile of DNA in the vicinity of energy minima for the purpose of refining the DNA force fields.

## ■ CONCLUSIONS

In summary, Rendell's CCSD code in NWChem has been optimized to overcome the communication bottleneck in the D3 contraction term. The optimization eliminated the quartic increase in communication intensity with the increase in the number of basis shells in the D3-section of the code. This modification provided a 2- to 5-fold performance increase with the actual number depending on the problem size and the

**Table 3. Single-Point Energy Difference between A- and B-Conformations of GC-dDMP[a]**

| CHARMM | AMBER | MP2/6-311++G** | CCSD/6-311++G** | CCSD(T)/6-311++G** |
|---|---|---|---|---|
| −5.6 | 4.1 | −1.9 | −1.3 | −1.2 |

[a] Energy difference is $E_{\text{A-conf}} - E_{\text{B-conf}}$, kcal/mol.

available memory. Optimization improved the scalability of the code and made possible running the entire conventional CCSD(T) computation on 20 000 nodes, which is 88% of the total number of XE6 nodes on Blue Waters. The $O(N^4)$ scaling of the ST2 data storage with the number of basis functions limited the size of the studied systems to about 1000 basis functions on compute nodes with 64 GB of RAM, and the present implementation could use only a single core per node in the computation of GC-dDMP. The work on performance optimization of the CCSD method illustrated that having large RAM on HPC machines and aggregating the data communication routes are vital for the quantum chemistry community to effectively utilize petascale resources. Application of the CCSD(T) method to the study of the conformational energy difference between A- and B-families of GC-dDMP revealed a deficiency in the classical force fields for DNA that should be addressed in future force field optimization efforts. Our computations showed a reasonable level of agreement between MP2 and CCSD(T) in the conformational study of DNA strands, endorsing the utility of the MP2 method for conformational studies of DNA.

## ■ ASSOCIATED CONTENT

### Ⓢ Supporting Information

Cartesian coordinates of the optimized structures and PDB stubs of GC-dDMP in AMBER and CHARMM conventions. This material is available free of charge via the Internet at http://pubs.acs.org.

## ■ AUTHOR INFORMATION

### Corresponding Author

*Email: anisimov@illinois.edu.

### Notes

The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENTS

This research is part of the Blue Waters sustained petascale computing project, which is supported by the National Science Foundation (award No. OCI 07-25070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana−Champaign and its National Center for Supercomputing Applications. Part of the test calculations has been performed using EMSL, a national scientific user facility sponsored by the Department of Energy's Office of Biological and Environmental Research and located at Pacific Northwest National Laboratory. The Pacific Northwest National Laboratory is operated for the US Department of Energy by the Battelle Memorial Institute under Contract DE-AC06.76RLO-1830.

## ■ ABBREVIATIONS

AO, atomic orbital; CC, coupled cluster; CCSD, coupled cluster including single and double excitations; DNA, deoxyribose nucleic acid; FLOP, floating-point operation; G, guanine; GA, global arrays; GC-dDMP, guanine−cytosine deoxydinucleotide monophosphate; GC-dDMP-A, GC-dDMP in A-conformation; GC-dDMP-B, GC-dDMP in B-conformation; GG-stack, guanine−guanine stack; HF, Hartee−Fock; HPC, high-performance computing; I/O, input-output; MO, molecular orbital; MPI, message passing interface; PDB, Protein Data Bank; PF, petaflop; RAM, random access memory; SPP, sustained petaflop performance; T, perturbative triple excitations; TCE, tensor contraction engine; RHF, restricted Hartree−Fock; ROHF, restricted open-shell Hartree−Fock; UHF, unrestricted Hartree−Fock

## ■ REFERENCES

(1) Valiev, M.; Bylaska, E. J.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Van Dam, H. J. J.; Wang, D.; Nieplocha, J.; Apra, E.; Windus, T. L.; de Jong, W. A. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Comput. Phys. Commun.* **2010**, *181*, 1477−1489.

(2) Cheatham, T. E., III Simulation and modeling of nucleic acid structure, dynamics, and interactions. *Curr. Opin. Struct. Biol.* **2004**, *14*, 360−367.

(3) Nielsen, P. E. Peptide nucleic acids (PNA) in chemical biology and drug discovery. *Chem. Biodiv.* **2010**, *7*, 786−804.

(4) Sedlak, R.; Janowski, T.; Pitonak, M.; Rezac, J.; Pulay, P.; Hobza, P. Accuracy of quantum chemical methods for large noncovalent complexes. *J. Chem. Theory Comput.* **2013**, *9*, 3364−3374.

(5) Poltev, V. I.; Anisimov, V. M.; Danilov, V. I.; Deriabina, A.; Gonzalez, E.; Jurkiewicz, A.; Les, A.; Polteva, N. DFT study of B-like conformations of deoxydinucleoside monophosphates containing Gua and/or Cyt and their complexes with Na+ cation. *J. Biomol. Struct. Dyn.* **2008**, *25*, 563−571.

(6) Poltev, V. I.; Anisimov, V. M.; Danilov, V. I.; Deriabina, A.; Gonzalez, E.; Garcia, D.; Rivas, F.; Jurkiewicz, A.; Les, A.; Polteva, N. DFT study of minimal fragments of nucleic acid single chain for explication of sequence dependence of DNA duplex conformation. *J. Mol. Struct: THEOCHEM* **2009**, *912*, 53−59.

(7) Poltev, V. I.; Anisimov, V. M.; Danilov, V. I.; van Mourik, T.; Deriabina, A.; González, E.; Padua, M.; Garcia, D.; Rivas, F.; Polteva, N. DFT study of polymorphism of the DNA double helix at the level of dinucleoside monophosphates. *Int. J. Quantum Chem.* **2010**, *110*, 2548−2559.

(8) Poltev, V. I.; Anisimov, V. M.; Danilov, V. I.; Garcia, D.; Deriabina, A.; Gonzalez, E.; Salazar, R.; Rivas, F.; Polteva, N. DFT study of DNA sequence dependence at the level of dinucleoside monophosphates. *Comput. Theor. Chem.* **2011**, *975*, 69−75.

(9) Poltev, V.; Anisimov, V. M.; Danilov, V. I.; Garcia, D.; Sanchez, C.; Deriabina, A.; Gonzalez, E.; Rivas, F.; Polteva, N. The role of molecular structure of sugar−phosphate backbone and nucleic acid bases in the formation of single-stranded and double-stranded DNA structures. *Biopolymers* **2014**, *101*, 640−650.

(10) Choi, J.; Majima, T. Conformational changes of non-B DNA. *Chem. Soc. Rev.* **2011**, *40*, 5893−5909.

(11) Coester, F. Bound states of a many-particle system. *Nucl. Phys.* **1958**, *7*, 421−424.

(12) Coester, F.; Kummel, H. Short-range correlations in nuclear wave functions. *Nucl. Phys.* **1960**, *17*, 477−485.

(13) Cizek, J. On the correlation problem in atomic and molecular systems. Calculation of wavefunction components in Ursell-type expansion using quantum-field theoretical methods. *J. Chem. Phys.* **1966**, *45*, 4256−4266.

(14) Kowalski, K.; Bhaskaran-Nair, K.; Brabec, J.; Pittner, J. Coupled cluster theories for strongly correlated molecular systems. In *Strongly Correlated Systems*; Avella, A., Mancini, F., Eds.; Springer Berlin Heidelberg: Berlin, Germany, 2013; Vol. *176*, pp 237−271.

(15) Bartlett, R. J.; Musial, M. Coupled-cluster theory in quantum chemistry. *Rev. Mod. Phys.* **2007**, *79*, 291−352.

(16) Kowalski, K.; Krishnamoorthy, S.; Olson, R. M.; Tipparaju, V.; Apra, E. Scalable implementations of accurate excited-state coupled cluster theories: application of high-level methods to porphyrin-based systems. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, Seattle, WA, Nov. 12−18, 2011; ACM: New York, 2011.

(17) Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S.; Windus, T. L.; Dupuis, M.; Montgomery, J. A., Jr. General

atomic and molecular electronic structure system. *J. Comput. Chem.* **1993**, *14*, 1347−1363.

(18) Werner, H.-J.; Knowles, P. J.; Knizia, G.; Manby, F. R.; Schütz, M. Molpro: A general-purpose quantum chemistry program package. *WIREs Comput. Mol. Sci.* **2012**, *2*, 242−253.

(19) Lotrich, V.; Flocke, N.; Ponton, M.; Yau, A. D.; Perera, A.; Deumens, E.; Bartlett, R. J. Parallel implementation of electronic structure energy, gradient, and Hessian calculations. *J. Chem. Phys.* **2008**, *128*, 194104.

(20) Turney, J. M.; Simmonett, A. C.; Parrish, R. M.; Hohenstein, E. G.; Evangelista, F. A.; Fermann, J. T.; Mintz, B. J.; Burns, L. A.; Wilke, J. J.; Abrams, M. L.; Russ, N. J.; Leininger, M. L.; Janssen, C. L.; Seidl, E. T.; Allen, W. D.; Schaefer, H. F.; King, R. A.; Valeev, E. F.; Sherrill, C. D.; Crawford, T. D. Psi4: An open-source ab initio electronic structure program. *WIREs: Comput. Mol. Sci.* **2012**, *2*, 556−565.

(21) Krylov, A. I.; Gill, P. M. W. Q-Chem: An engine for innovation. *WIREs: Comput. Mol. Sci.* **2013**, *3*, 317−326.

(22) Baker, J.; Janowski, T.; Wolinski, K.; Pulay, P. Recent developments in the PQS program. *WIREs: Comput. Mol. Sci.* **2012**, *2*, 63−72.

(23) Kobayashi, R.; Rendell, A. P. A direct coupled cluster algorithm for massively parallel computers. *Chem. Phys. Lett.* **1997**, *265*, 1−11.

(24) Scuseria, G. E.; Janssen, C. L.; Schaefer, H. F. An efficient reformulation of the closed-shell coupled cluster single and double excitation (CCSD) equations. *J. Chem. Phys.* **1988**, *89*, 7382−7387.

(25) Paldus, J.; Cizek, J. Time-independent diagrammatic approach to perturbation theory of fermion systems. In *Adv. Quantum Chem.*; Lowdin, P.-O., Ed.; Academic Press: New York, 1975; Vol. *9*, pp 105−197.

(26) Xantheas, S. S. Low-lying energy isomers and global minima of aqueous nanoclusters: Structures and spectroscopic features of the pentagonal dodecahedron $(H_2O)_{20}$ and $(H_3O)+(H_2O)_{20}$. *Can. J. Chem. Eng.* **2012**, *90*, 843−851.

(27) Apra, E.; Rendell, A. P.; Harrison, R. J.; Tipparaju, V.; deJong, W. A.; Xantheas, S. S. Liquid water: Obtaining the right answer for the right reasons. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, Portland, OR, Nov. 14−20, 2009; ACM: New York, 2009.

(28) Sinnokrot, M. O.; Sherrill, C. D. Highly accurate coupled cluster potential energy curves for the benzene dimer: Sandwich, T-shaped, and parallel-displaced configurations. *J. Phys. Chem. A* **2004**, *108*, 10200−10207.

(29) Klauda, J. B.; Venable, R. M.; Freites, J. A.; O'Connor, J. W.; Tobias, D. J.; Mondragon-Ramirez, C.; Vorobyov, I.; MacKerell, A. D.; Pastor, R. W. Update of the CHARMM all-atom additive force field for lipids: Validation on six lipid types. *J. Phys. Chem. B* **2010**, *114*, 7830−7843.

(30) Riplinger, C.; Sandhoefer, B.; Hansen, A.; Neese, F. Natural triple excitations in local coupled cluster calculations with pair natural orbitals. *J. Chem. Phys.* **2013**, *139*, 134101.

(31) Masur, O.; Usvyat, D.; Schuetz, M. Efficient and accurate treatment of weak pairs in local CCSD(T) calculations. *J. Chem. Phys.* **2013**, *139*, 164116.

(32) Scuseria, G. E.; Ayala, P. Y. Linear scaling coupled cluster and perturbation theories in the atomic orbital basis. *J. Chem. Phys.* **1999**, *111*, 8330−8343.

(33) Flocke, N.; Bartlett, R. J. A natural linear scaling coupled-cluster method. *J. Chem. Phys.* **2004**, *121*, 10935−10944.

(34) Ziolkowski, M.; Jansik, B.; Kjaergaard, T.; Jorgensen, P. Linear scaling coupled cluster method with correlation energy based error control. *J. Chem. Phys.* **2010**, *133*, 014107.

(35) Li, S.; Ma, J.; Jiang, Y. Linear scaling local correlation approach for solving the coupled cluster equations of large systems. *J. Comput. Chem.* **2002**, *23*, 237−244.

(36) Li, W.; Piecuch, P.; Gour, J. R.; Li, S. Local correlation calculations using standard and renormalized coupled-cluster approaches. *J. Chem. Phys.* **2009**, *131*, 114109.

(37) Kobayashi, M.; Nakai, H. Divide-and-conquer-based linear-scaling approach for traditional and renormalized coupled cluster methods with single, double, and noniterative triple excitations. *J. Chem. Phys.* **2009**, *131*, 114108.

(38) Friedrich, J.; Coriani, S.; Helgaker, T.; Dolg, M. Implementation of the incremental scheme for one-electron first-order properties in coupled-cluster theory. *J. Chem. Phys.* **2009**, *131*, 154102.

(39) Purvis, G. D.; Bartlett, R. J. A full coupled-cluster singles and doubles model: The inclusion of disconnected triples. *J. Chem. Phys.* **1982**, *76*, 1910−1918.

(40) Raghavachari, K.; Trucks, G. W.; Pople, J. A.; Head-Gordon, M. A fifth-order perturbation comparison of electron correlation theories. *Chem. Phys. Lett.* **1989**, *157*, 479−483.

(41) Nieplocha, J.; Palmer, B.; Tipparaju, V.; Krishnan, M.; Trease, H.; Apra, E. Advances, applications, and performance of the global arrays shared memory programming toolkit. *Int. J. High Perf. Comput. Appl.* **2006**, *20*, 203−231.

(42) Nieplocha, J.; Harrison, R. J.; Littlefield, R. J. Global arrays: A portable "shared-memory" programming model for distributed memory computers. In *Proceedings of the 1994 ACM/IEEE Conference on Supercomputing*, Washington, DC, Nov. 14−18, 1994; IEEE Computer Society Press: Los Alamitos, CA, 1994.

(43) Paldus, J.; Li, X. A critical assessment of coupled cluster method in quantum chemistry. In *Adv. Chem. Phys.*; John Wiley & Sons, Inc.: New York, 2007; pp 1−175.

(44) Møller, C.; Plesset, M. S. Note on an approximation treatment for many-electron systems. *Phys. Rev.* **1934**, *46*, 618−622.

(45) Dahlke, E. E.; Leverentz, H. R.; Truhlar, D. G. Evaluation of the electrostatically embedded many-body expansion and the electrostatically embedded many-body expansion of the correlation energy by application to low-lying water hexamers. *J. Chem. Theory Comput.* **2008**, *4*, 33−41.

(46) Olson, R. M.; Bentz, J. L.; Kendall, R. A.; Schmidt, M. W.; Gordon, M. S. A novel approach to parallel coupled cluster calculations: Combining distributed and shared memory techniques for modern cluster based systems. *J. Chem. Theory Comput.* **2007**, *3*, 1312−1328.

(47) Brueckner, K. A. Two-body forces and nuclear saturation. III. Details of the structure of the nucleus. *Phys. Rev.* **1955**, *97*, 1353−1366.

(48) Brueckner, K. A. Many-body problem for strongly interacting particles. II. Linked cluster expansion. *Phys. Rev.* **1955**, *100*, 36−45.

(49) Goldstone, J. Derivation of the Brueckner many-body theory. *Proc. R. Soc. London A* **1957**, *239*, 267−279.

(50) Lindgren, I.; Morrison, J. *Atomic Many-Body Theory*. Springer: Berlin, Germany, 1982; Vol. *13*, p 469.

(51) Kutzelnigg, W. How many-body perturbation theory (MBPT) has changed quantum chemistry. *Int. J. Quantum Chem.* **2009**, *109*, 3858−3884.

(52) Bochevarov, A. D.; Sherrill, C. D. Hybrid correlation models based on active-space partitioning: Correcting second-order Møller−Plesset perturbation theory for bond-breaking reactions. *J. Chem. Phys.* **2005**, *122*, 234110.

(53) Nooijen, M. Combining coupled cluster and perturbation theory. *J. Chem. Phys.* **1999**, *111*, 10815−10826.

(54) Krishnan, R.; Binkley, J. S.; Seeger, R.; Pople, J. A. Self-consistent molecular orbital methods. XX. A basis set for correlated wave functions. *J. Chem. Phys.* **1980**, *72*, 650−654.

(55) Clark, T.; Chandrasekhar, J.; Spitznagel, G. W.; Schleyer, P. V. R. Efficient diffuse function-augmented basis sets for anion calculations. III. The 3-21+G basis set for first-row elements, Li−F. *J. Comput. Chem.* **1983**, *4*, 294−301.

(56) Performance Application Programming Interface. http://icl.cs.utk.edu/papi/index.html (accessed June 27, 2014).

(57) Finley, J. B.; Luo, M. X-ray crystal structures of half the human papilloma virus E2 binding site: d(GACCGCGGTC). *Nucleic Acids Res.* **1998**, *26*, 5719−5727.

(58) Heinemann, U.; Alings, C.; Bansal, M. Double helix conformation, groove dimensions and ligand binding potential of a G/C stretch in B-DNA. *EMBO J.* **1992**, *11*, 1931−1939.

(59) Wang, J.; Cieplak, P.; Kollman, P. A. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? *J. Comput. Chem.* **2000**, *21*, 1049−1074.
(60) Foloppe, N.; MacKerell, A. D., Jr. All-atom empirical force field for nucleic acids: I. Parameter optimization based on small molecule and condensed phase macromolecular target data. *J. Comput. Chem.* **2000**, *21*, 86−104.
(61) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781−1802.
(62) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J. Mol. Graphics* **1996**, *14*, 33−38.
(63) Petitet, A.; Whaley, R. C.; Dongarra, J. HPL—A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. http://www.netlib.org/benchmark/hpl/ (accessed June 27, 2014).
(64) Standard Performance Evaluation Corporation. http://www.spec.org (accessed June 27, 2014).
(65) Kramer, W. Measuring sustained performance on blue waters with the SPP metric. In *Cray User Group Meeting (CUG2013)*, Napa, CA, May 6−9, 2013.
(66) Kramer, W. T. C. *PERCU: A Holistic Method for Evaluating High Performance Computing Systems*; EECS Department, University of California: Berkeley, CA, 2008.
(67) Bloomfield, V. A.; Crothers, D. M.; Tinoco, I. J. *Nucleic Acids: Structures, Properties and Functions*; University Science Books: Sausalito, CA, 2000.
(68) Arscott, P. G.; Ma, C.; Wenner, J. R.; Bloomfield, V. A. DNA condensation by cobalt hexaammine(III) in alcohol−water mixtures: Dielectric constant and other solvent effects. *Biopol.* **1995**, *36*, 345−364.
(69) Reddy, S. Y.; Leclerc, F.; Karplus, M. DNA polymorphism: A comparison of force fields for nucleic acids. *Biophys. J.* **2003**, *84*, 1421−1449.
(70) Wolf, M. G.; Groenhof, G. Evaluating nonpolarizable nucleic acid force fields: A systematic comparison of the nucleobases hydration free energies and chloroform-to-water partition coefficients. *J. Comput. Chem.* **2012**, *33*, 2225−2232.
(71) Cheatham, T. E.; Cieplak, P.; Kollman, P. A. A modified version of the Cornell et al. force field with improved sugar pucker phases and helical repeat. *J. Biomol. Struct. Dyn.* **1999**, *16*, 845−862.
(72) Sode, O.; Keçeli, M.; Hirata, S.; Yagi, K. Coupled-cluster and many-body perturbation study of energies, structures, and phonon dispersions of solid hydrogen fluoride. *Int. J. Quantum Chem.* **2009**, *109*, 1928−1939.
(73) Nanda, K. D.; Beran, G. J. O. Prediction of organic molecular crystal geometries from MP2-level fragment quantum mechanical/molecular mechanical calculations. *J. Chem. Phys.* **2012**, *137*, 174106.
(74) Perez, A.; Luque, F. J.; Orozco, M. Frontiers in molecular dynamics simulations of DNA. *Acc. Chem. Res.* **2012**, *45*, 196−205.