JCTC Journal of Chemical Theory and Computation

Parallel Calculation of CCSDT and Mk-MRCCSDT Energies

Eric Prochnow,* Michael E. Harding,† and Jürgen Gauss

Institut für Physikalische Chemie, Universität Mainz, D-55099 Mainz, Germany

Received April 14, 2010

Abstract: A scheme for the parallel calculation of energies at the coupled-cluster singles, doubles, and triples (CCSDT) level of theory, several approximate iterative CCSDT schemes (CCSDT-1a, CCSDT-1b, CCSDT-2, CCSDT-3, and CC3), and for the state-specific multireference coupled-cluster ansatz suggested by Mukherjee with a full treatment of triple excitations (Mk-MRCCSDT) is presented. The proposed scheme is based on the adaptation of a highly efficient serial coupled-cluster code leading to a communication-minimized implementation by parallelizing the time-determining steps. The parallel algorithm is tailored for affordable cluster architectures connected by standard communication networks such as Gigabit Ethernet. In this way, CCSDT and Mk-MRCCSDT computations become feasible even for larger molecular systems and basis sets. An analysis of the time-determining steps for CCSDT and Mk-MRCCSDT, namely the computation of the triple-excitation amplitudes and their individual contributions, is carried out. Benchmark calculations are presented for the N_2O, ozone, and benzene molecules, proving that the parallelization of these steps is sufficient to obtain an efficient parallel scheme. A first application to the case of 2,6-pyridyne using a triple-ζ quality basis (222 basis functions) is presented demonstrating the efficiency of the current implementation.

I. Introduction

Coupled-cluster (CC) theory represents one of the most powerful approaches to solve the electronic Schrödinger equation for atoms and molecules.[1−4] Based on the CC ansatz for the wave function a hierarchy of methods can be derived which provides increasing accuracy as well as a fast and systematic convergence toward the full configuration-interaction (FCI) limit. Despite the success of these CC methods, there are a large number of problems such as, for example biradicals, transition-metal compounds, and bond breaking processes, for which the underlying assumption that the wave function is dominated by one reference determinant breaks down. To overcome these limitations, several attempts to generalize single-reference CC theory to a multireference CC (MRCC) ansatz have been reported in the literature (for some examples of genuine MRCC approaches, see refs 5−10). At the moment, the most widely discussed MRCC approaches are based on a state-specific ansatz. Such a strategy enhances the reliability and naturally avoids convergence problems due to intruder states by focusing on one state at a time.[3] However, to obtain a unique state-specific theory, supplementary specifications, i.e., the so-called sufficiency conditions, need to be imposed. Accordingly, different formulations of state-specific MRCC theory are possible.[11] A size-extensive ansatz is the state-specific MRCC method suggested by Mukherjee and co-workers (Mk-MRCC).[5] Mk-MRCC shows faster convergence toward the FCI limit and also yields more accurate energies than other MRCC approaches.[12] In light of these factors efforts in our group toward the development of a state-specific MRCC theory have recently focused on the Mk-MRCC approach.[12−15]

When aiming for quantitative accuracy it is essential to consider in the CC treatment triple-excitation effects. While in most calculations triple excitations are often treated using approximate schemes such as the CC singles and doubles (CCSD) approach with a perturbative treatment of triples (CCSD(T)),[16] full inclusion of triple excitations within the

* Corresponding author e-mail: Eric.Prochnow@uni-mainz.de.

† Present address: Department of Chemistry and Biochemistry, The University of Texas at Austin, Austin, TX 78712.

10.1021/ct1002016 © 2010 American Chemical Society
Published on Web 07/08/2010

CC singles, doubles, and triples (CCSDT)[17,18] model has turned out to be of importance in areas where quantitative accuracy is needed.[19−25] Furthermore, explicit expressions for the implementation of full triples have not only been presented for the single-reference case[26] but also within the Mk-MRCC framework in order to tackle multireference cases.[14] However, the application of CC methods including a full treatment of triple excitations to larger chemical problems is hampered by the considerable increase of computational effort with increasing number of electrons and basis functions. If N represents a measure of the system size the operation count and therefore the execution time scales as N^8 for CCSDT and $d\cdot N^8$ for Mk-MRCCSDT with d denoting the number of chosen reference determinants. In comparison to that the storage requirements for intermediates and CC amplitudes only scales as N^6 and $d\cdot N^6$, respectively. Therefore, the limiting factor of CCSDT calculations is the execution time and not the storage requirement.

To circumvent limitations due to the execution time and motivated by the cheap computing power available through computer clusters, several parallel implementations of single-reference CC methods have been recently presented.[27−31] For most of these parallelization schemes additional tools have been developed. The parallelization in ref 28, for example, is based on the use of Global Arrays (GA)[32] as a parallelization tool. GA simulates shared memory programming on distributed memory computer clusters by striping large arrays across nodes. In view of the large memory and disk space available nowadays, the striping of individual arrays may no longer be necessary. Therefore, an alternative approach, namely Array Files (AF),[33] has been developed where the whole array is stored on a given node. The concept of AF has been used for a parallel implementation of CCSD energy calculations by Janowski et al.[31] A highly sophisticated parallel algorithm for the computation of CCSD(T) energies has been furthermore reported by Olson et al. thereby combining distributed and shared memory techniques by using the Distributed Data Interface (DDI/3).[34,35] This implementation is tailored to multiprocessor and multicore nodes connected via a dedicated communication network. An ansatz which works without an additional layer of complexity provided by libraries such as GA, AF, or DDI/3 has been reported recently for the parallel calculation of CCSD and CCSD(T) energies as well as analytic first and second derivatives.[29] This scheme is based on the message passing interface (MPI)[36] and the adaptation of an efficient serial algorithm to parallel environments where all nonparallel steps run redundantly on all available processors at the same time.

In the field of MRCC theory, parallel algorithms are hardly found as MRCC calculations are so far not considered a routine application. Only Piecuch and Landman reported a parallelization scheme for state-universal CC calculations with single and double excitations.[37] In this scheme the underlying serial algorithm has been modified by parallelizing the most time-determining steps using OpenMP.[38]

To the best of our knowledge, parallel CC algorithms are so far only capable to compute perturbative treatments of triple excitations such as in CCSD(T), but no parallel implementation for CCSDT or a corresponding multireference ansatz has been reported, with the sole exception of the parallel implementation of CCSDT within the general CC programs of Kállay[39] and the NWChem program package.[40] In this paper we describe an approach based on the adaptation of an efficient serial algorithm and a detailed analysis of the time-determining steps for a full treatment of triple excitations as well as the resulting parallel algorithm in the quantum-chemical program package CFOUR[41] is presented. The applicability of our parallelization strategy is demonstrated by computations for N_2O, ozone (O_3), benzene (C_6H_6), and 2,6-pyridyne (C_5NH_3).

II. Parallelization of CCSDT and Mk-MRCCSDT Energy Computations

In our efforts to parallelize CCSDT and Mk-MRCCSDT energy calculations, we follow the strategy already described in ref 29 for the parallel computation of CCSD and CCSD(T) energies as well as analytic first and second derivatives. To avoid the communication of intermediate quantities, most of the quantities needed within the CC iterations are stored completely on every node. Thus, the algorithm presented here is tailored to cluster architectures with moderate hardware specifications as well as inexpensive and rather slow standard interconnect structures such as, for example, Gigabit Ethernet. Furthermore, it is assumed that enough fast memory and disk space are available locally on every node to store the full set of t-amplitudes. In this way communication is minimized as only the CC amplitudes need to be communicated.

In the actual algorithm, parts of the intermediates are contracted with the proper t_3-amplitudes to yield parts of the resulting quantities which, at the end of the parallel operation, are broadcasted to all other nodes. The costs associated with the communication of the triples amplitudes thus scales as occ^3vrt^3 for CCSDT and $d\cdot occ^3vrt^3$ for Mk-MRCCSDT, where occ (vrt) denotes the number of occupied (virtual) orbitals. This is two orders of magnitudes smaller than the required time for the computation of the amplitudes which has a scaling of occ^3vrt^5 and $d\cdot occ^3vrt^5$, respectively. Disregarding communication latencies, the dominating steps have been implemented therefore in a way that the total amount of communication between the parallel processes does not depend on the number of involved processors due to the MPI routine used. Thus, the distribution of the time-determining steps to different nodes reduces the overall walltime significantly, particularly when larger examples are considered.

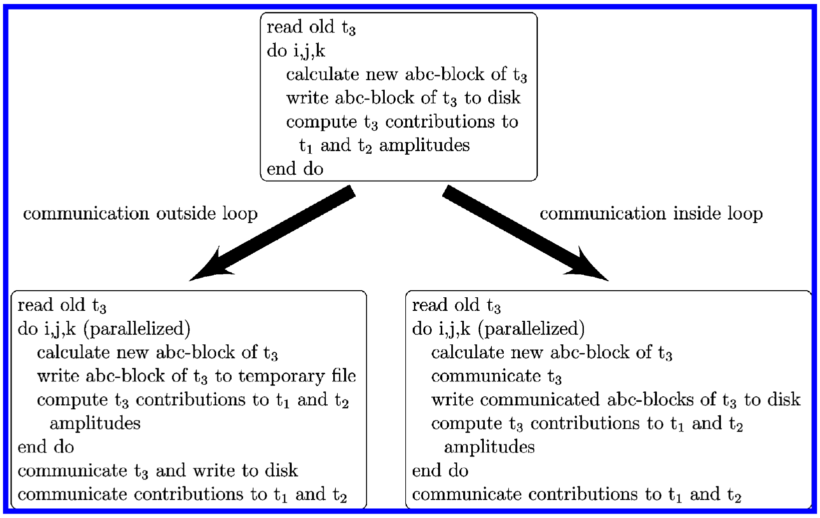
As the underlying theory and the general implementation of CCSDT and Mk-MRCCSDT have been described in the literature[13,14,26,42] we will report here only the essential equations and in this way avoid a reiteration of the detailed theory.

As for the parallelization of CCSD and CCSD(T) calculations,[29] it is important to identify the time-determining steps. CCSDT energy calculations are dominated by the computation of the t_3-amplitudes yielding a scaling of occ^3vrt^5. Furthermore, contributions of the t_3-amplitudes to the t_2-amplitude equations having a formal scaling of N^7 need to be considered. Thus, the time spent for the calculation of the triples amplitudes and

Table 1. Timings (Walltime in s) for the N_2O Molecule for One CC Iteration and the Triples Part within This Iteration[a]

no. of nodes	CC iter., outside	CC iter., inside	triples part, outside	triples part, inside
1	1065	1062	1046 (98%)	1043
2	547	545	528 (97%)	526
4	288	289	269 (93%)	269
8	158	157	139 (88%)	138
16	93	94	74 (80%)	75

[a] The code has been parallelized outside or inside the loop over occupied orbitals. The computation has been carried out at the CCSDT/cc-pCVTZ level (129 basis functions). For the communication outside the loop the ratio of the triples part to the whole CC iteration is given in parentheses. On each node 1 core has been used.



```
read old $t_3$
do i,j,k
  calculate new abc-block of $t_3$
  write abc-block of $t_3$ to disk
  compute $t_3$ contributions to
    $t_1$ and $t_2$ amplitudes
end do
```

communication outside loop

```
read old $t_3$
do i,j,k (parallelized)
  calculate new abc-block of $t_3$
  write abc-block of $t_3$ to temporary file
  compute $t_3$ contributions to $t_1$ and $t_2$
    amplitudes
end do
communicate $t_3$ and write to disk
communicate contributions to $t_1$ and $t_2$
```

communication inside loop

```
read old $t_3$
do i,j,k (parallelized)
  calculate new abc-block of $t_3$
  communicate $t_3$
  write communicated abc-blocks of $t_3$ to disk
  compute $t_3$ contributions to $t_1$ and $t_2$
    amplitudes
end do
communicate contributions to $t_1$ and $t_2$
```

Figure 1. Schematic representation of the two different parallelization strategies. The serial algorithm (top) has been parallelized by communicating the triples amplitudes outside (left) or inside (right) the loop over occupied orbitals i,j,k.

the resulting contributions to the t_2-amplitude equations increases more rapidly with the number of basis functions than in the pure CCSD part (the scaling of CCSD is $occ^2 vrt^4$) which renders the triples part by far the time-determining step in CCSDT calculations. This becomes obvious when considering the timings for a CCSDT energy calculation, e.g., for the N_2O molecule employing the cc-pCVTZ basis set[43] (Table 1). For this example the time spent in the triples part for a serial calculation is about 98% of a CC iteration.

Our approach to parallelize the computation of the triple amplitudes starts with the t_3-amplitude equations given as

$$t_{ijk}^{abc} D_{ijk}^{abc} = \langle \Phi_{ijk}^{abc} | e^{-\hat{T}} (\hat{H}_N - \hat{H}_0) e^{\hat{T}} | 0 \rangle \qquad (1)$$

with

$$\hat{H}_0 = \sum_p f_{pp} \{\hat{a}_p^\dagger \hat{a}_p\} \qquad (2)$$

The denominator array D_{ijk}^{abc} is defined as

$$D_{ijk}^{abc} = f_{ii} + f_{jj} + f_{kk} - f_{aa} - f_{bb} - f_{cc} \qquad (3)$$

where i, j, k, ... denotes occupied and a, b, c, ... virtual spin orbitals. The normal-ordered Hamiltonian $\hat{H}_N$ is given as

$$\hat{H}_N = \sum_{pq} f_{pq} \{\hat{a}_p^\dagger \hat{a}_q\} + \frac{1}{4} \sum_{pqrs} \langle pq || rs \rangle \{\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r\} \qquad (4)$$

with the Fock-matrix elements

$$f_{pq} = h_{pq} + \sum_{k}^{occ} \langle pk || qk \rangle \qquad (5)$$

as well as the one-electron integrals h_{pq} and the antisymmetrized two-electron integrals $\langle pk || qk \rangle$. The strings $\{\hat{a}_p^\dagger \hat{a}_p\}$, $\{\hat{a}_p^\dagger \hat{a}_q\}$, and $\{\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r\}$ denote normal-ordered sequences of creation ($\hat{a}_p^\dagger$) and annihilation ($\hat{a}_q$) operators.

The basic feature in the serial CFOUR algorithm for the formation of the t_3-amplitudes in a CCSDT energy calculation is an outer loop over an index triple i, j, k of the t_{ijk}^{abc} amplitudes (for a detailed description of the implementation see refs 26 and 42). Blocks of a, b, c index triples are computed within the loop one at a time and stored on disk for the next iteration. In addition, these blocks are used immediately to calculate the contributions to the singles and doubles amplitude equations. A schematic representation of this algorithm is depicted in the upper box of Figure 1.

The Mk-MRCCSDT ansatz is implemented using the same loop structure over indices i, j, k. The triples-amplitude equations of the Mk-MRCC ansatz are given as

$$\langle \Phi_{ijk}^{abc}(\mu) | e^{-\hat{T}_\mu} \hat{H} e^{\hat{T}_\mu} | \Phi_\mu \rangle c_\mu^\alpha + \sum_{\nu(\neq\mu)} \langle \Phi_{ijk}^{abc}(\mu) | e^{-\hat{T}_\mu} e^{\hat{T}_\nu} | \Phi_\mu \rangle H_{\mu\nu}^{\mathrm{eff}} c_\nu^\alpha = 0 \qquad (6)$$

where c_μ^α denotes the weighting coefficients of the reference determinants in the wave function. These are obtained by diagonalizing the effective Hamiltonian $H_{\mu\nu}^{\mathrm{eff}}$ with its matrix elements defined as

$$H^{\mathrm{eff}}_{\mu\nu} = \langle \Phi_\mu | \hat{H} e^{\hat{T}_\nu} | \Phi_\nu \rangle \overset{\mathrm{CMS}}{=} \langle \Phi_\mu | e^{-\hat{T}_\nu} \hat{H} e^{\hat{T}_\nu} | \Phi_\nu \rangle \qquad (7)$$

The second equality in eq 7 holds for a complete model space (CMS), i.e., a model space that contains all determinants generated by distributing m electrons in n_{act} active orbitals (for a detailed description of the Mk-MRCC ansatz, see refs 5 and 15). The advantage of the Mk-MRCC method is the fact that the amplitude equations (eq 6) can be separated into a single-reference and a coupling part.[44] The first part is very similar to eq 1 with the only difference that all quantities are now specific for one reference determinant μ. Consequently, the Mk-MRCC ansatz can be implemented within a single-reference CC (SRCC) code as, for example, provided by the CFOUR package by looping over the reference determinants and adding the coupling terms.[13] This does not affect the general structure of the algorithm for the calculation of the t_3-amplitudes as mentioned before. The only difference is an extra loop over reference determinants around the loop over the index triples i, j, k and the additional coupling terms which are added inside the i, j, k loop.

For a parallelization of the loop over the index triples i, j, k the t_3-amplitudes and the triples contributions to the t_1- and t_2-amplitudes computed on separate nodes need to be communicated. For the communication of the triples amplitudes two different strategies are possible as shown in Figure 1. The t_3-amplitudes are either written to a temporary file and communicated outside the loop or communicated directly after their computation inside the loop. The contributions to the t_1- and t_2-amplitudes are in both cases computed within the same i, j, k loop and communicated outside the loop. Using the strategy described above all triple terms and contributions are parallelized simultaneously. Each parallel process has to calculate a similar amount of i, j, k index triples which is then broadcasted to all other processors.

The performance of the two possible schemes is demonstrated via computations for the N_2O molecule[45] at the CCSDT level of theory using the cc-pCVTZ basis (129 basis functions).[46] The timings for one CC iteration and the respective triples part are given in Table 1. It can be seen that the walltime for one CC iteration is reduced considerably with increasing number of processors. Thus, our implementation will make calculations at the CCSDT level feasible within days or weeks that would take otherwise months to calculate, provided that an appropriate number of nodes is used. In addition, with a larger number of nodes the triples part becomes less dominant in a CCSDT calculation. While for two nodes 97% of the time are spent computing triples contributions, it is reduced to 80% on 16 nodes. The corresponding speedup[47] of the parallel algorithm is depicted in Figure 2. The speedup observed with increasing number of nodes is quite close to the optimal one. With 16 nodes the speedup of one CCSDT iteration reaches values between 11 and 12, while the triples part alone yields a speedup of 14. It is interesting to note that the two outlined communication strategies for the t_3-amplitudes (outside and inside the i, j, k loop) show very similar performance. However, a careful investigation of Figure 2 indicates some preferences to the "outside communication". Therefore, this communication strategy is used in the following.

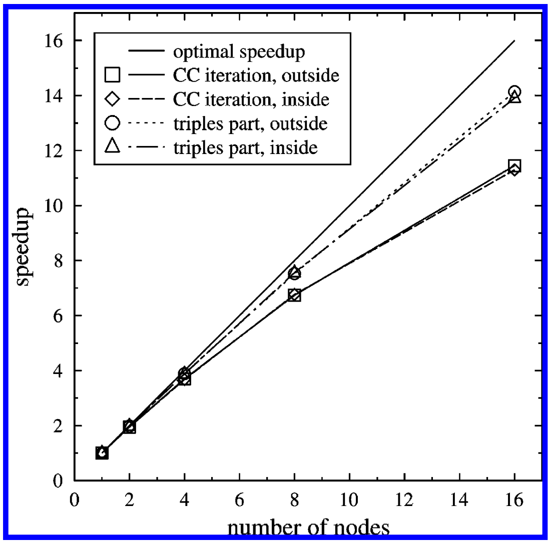


Figure 2. Parallel scaling of energy calculation at the CCSDT/cc-pCVTZ level of theory for N_2O. Depicted is the speedup for one CCSDT iteration and for the triples part within this iteration. The t_3-amplitudes are communicated either outside or inside the loop over i, j, k indices. On each node 1 core has been used.

Table 2. Timings (Walltime in s) for One CC Iteration of the Computation for the N_2O Molecule Using Approximate Schemes of CCSDT[a]

no. of nodes	CCSDT-1a	CCSDT-1b	CCSDT-2	CCSDT-3	CC3
1	596	605	650	660	667
2	311 (1.9)	318 (1.9)	356 (1.8)	373 (1.8)	373 (1.8)
4	170 (3.5)	175 (3.5)	214 (3.0)	231 (2.9)	229 (2.9)
8	99 (6.0)	102 (5.9)	142 (4.6)	158 (4.2)	157 (4.2)
16	64 (9.3)	67 (9.0)	106 (6.1)	123 (5.4)	123 (5.4)

[a] The computations have been carried out using the cc-pV5Z basis set (273 basis functions). The speedups for the corresponding computations are given in parentheses. On each node 1 cpu has been used.

As the full inclusion of triple excitations in CC calculation is computational demanding due to its scaling behavior (N^8), several approximate iterative schemes of CCSDT have been developed, namely CCSDT-n (n = 1a, 1b, 2, 3)[48,49] and CC3.[50] In these schemes certain terms of the CCSDT amplitude equations are skipped so that the computational scaling is reduced to N^7. As these methods are implemented within the same i, j, k loop in CFOUR their parallelization is straightforward but should be mentioned at this point. The timings for these methods obtained for the computation on N_2O using the cc-pV5Z basis[51] (273 basis functions)[52] are reported in Table 2. As for CCSDT the walltime is reduced considerably when increasing the number of nodes used, e.g., the calculation time for one iteration of the CCSDT-1a method is about 10 min on 1 node in comparison to about 1 min on 16 nodes yielding a speedup of 9.3. The speedup for 16 nodes observed here is smaller than for CCSDT which is explained by the fact that these computations are less dominated by the triples part. While the time spend calculating the triples contributions for the CCSDT-1a method is 95% of the whole CC iteration on 1 node (compared to 98%

Table 3. Timings (Walltime in s) for the O_3 Molecule for One CC Iteration and the Triples Part within This Iteration[a]

no. of nodes	CC iter.	triples part
1	5864	5811 (99%)
2	3065	3013 (98%)
4	1597	1544 (97%)
8	830	778 (94%)
16	436	384 (88%)

[a] The communication has been performed outside the loop. The computation has been carried out at the Mk-MRCCSDT/cc-pCVTZ level (129 basis functions). The ratio of the triples part to the whole CC iteration is given in parentheses. On each node 1 core has been used.

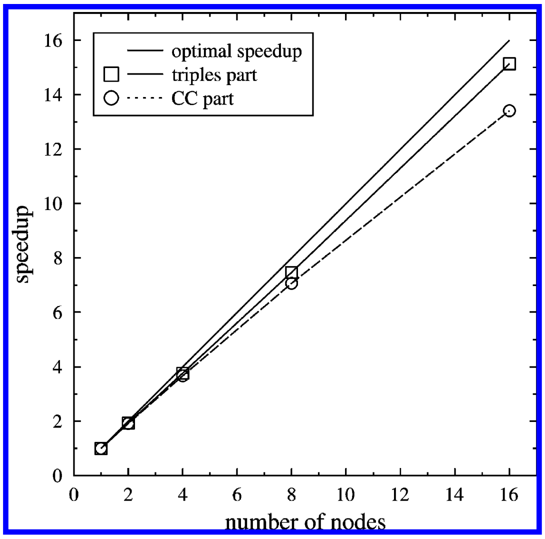


Figure 3. Parallel scaling of energy calculation at the Mk-MRCCSDT/cc-pCVTZ level of theory for O_3. Depicted is the speedup for one Mk-MRCCSDT iteration and for the triples part within this iteration. The t_3-amplitudes are communicated outside the loop over *i*, *j*, *k* indices. On each node 1 core has been used.

for CCSDT using a smaller basis) it is reduced to 58% on 16 nodes (80% for CCSDT).

Table 4. Timings (Walltime in s) for One CC Iteration for the Benzene Molecule and the Triples Part within This Iteration[a]

no. of cores	CC iteration	triples part
1	21669	21494
2	11036	10861
4	5866	5685
8	2986	2805
16	1542	1371
20	1302	1131
22	1209	1037
24	1327	1126
28	1160	963
32	1081	886
64	809	594

[a] The computation has been carried out at the fc-CCSDT/cc-pVTZ level (264 basis functions). The parallel processes have been distributed in a round-robin fashion.

The performance of the parallel computation of Mk-MRCCSDT energies is shown for the ozone molecule. The computation[53] has been performed using the cc-pCVTZ basis set and two-configurational SCF orbitals (with the active-space orbitals having b_1 and a_2 symmetry) using the experimental geometry ($R = 1.2717$ Å, $\angle = 116.8°$).[54] The timings for one CC iteration and the corresponding triples part are given in Table 3. The time for one CC iteration is considerably reduced with an increasing number of processors yielding a computation time of about 7 min on 16 nodes in comparison to 98 min on 1 node. The triples part of this calculation is even more dominant than for a single-reference CCSDT case. For Mk-MRCCSDT the time spent in the triples part is 99% on 1 node and 88% on 16 nodes (compared to 98% and 80% for CCSDT). The speedup obtained for the Mk-MRCCSDT computations is displayed in Figure 3. With 16 nodes the speedup observed lies between 13 and 14, while the triples part alone yields a speedup of over 15. These speedups are even closer to the optimal speedup as the ones obtained for the CCSDT calculation of N_2O, thus, demonstrating the efficiency of the parallelization scheme presented here for Mk-MRCCSDT computations.

III. Results and Discussion

In this section we focus on the overall performance and applicability of our parallelization scheme. Results of two applications are presented that involve typical problems in quantum chemistry for which the inclusion of triple excitation is desirable but so far extremely time-consuming.

A. The CCSDT Energy of Benzene. For the investigation of the energetics of molecules, such as the atomization energy, several schemes have been reported in the literature to achieve high-accuracy results.[22−24] From these studies it is obvious that triple excitations are of importance and that contributions beyond CCSD(T) due to a full treatment of triple excitations at the CCSDT level of theory are non-negligible: the contribution from nonperturbative triple excitations, approximated by the difference of CCSDT and CCSD(T) extrapolated from triple- and quadruple-ζ quality basis sets is found for example to be crucial for obtaining chemical accuracy (about 1 kcal/mol) for the molecules N_2, C_2H_2, CO_2, HCN, and O_2 (see ref 22) as well as for vinyl chloride,[55] cyclopropenylidene, propadienylidene,[56] and many others. However, these computations are often very time-consuming, and the applicability of these schemes thus is limited to rather small molecules. Using the parallelization scheme presented here, high-level computations on larger molecules as benzene or substituted benzenes become feasible.

To demonstrate the applicability of our parallelization scheme for medium-sized molecules we have chosen benzene (with geometrical parameters R(CC) = 1.3911 Å, R(CH) = 1.0800 Å)[57] as a typical example. The timings for the frozen-core (fc) energy calculation at the CCSDT level of theory using the cc-pVTZ basis set[51] are given in Table 4. Again, as for N_2O, the walltime is reduced considerably with increasing number of nodes. The walltime for one CC iteration on one node is about six hours, whereas the same computation on 16 nodes only requires about 25 min. Thus, the whole calculation is carried out within hours instead of days. In Figure 4 the corresponding speedup is visualized. In comparison to the results obtained for N_2O (Figure 2) the

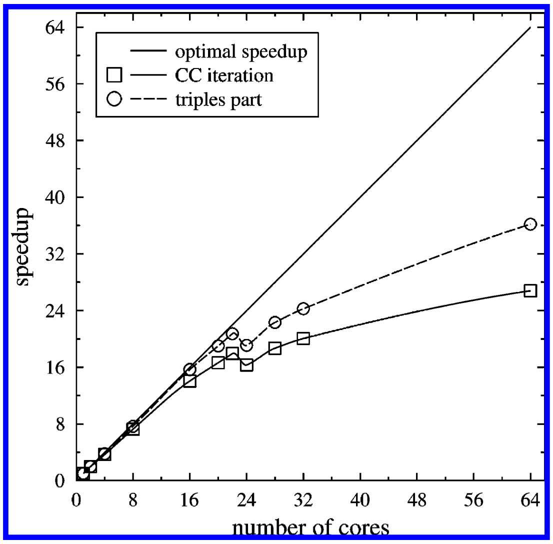


Figure 4. Parallel scaling of frozen-core energy calculation at the CCSDT/cc-pVTZ level for benzene. Depicted is the speedup for one CCSDT iteration and for the triples part within this iteration. The parallel processes have been distributed in a round-robin fashion.

Table 5. Timings (Walltime in s) and Corresponding Speedup for One Iteration in the Calculation of Benzene at the fc-CCSDT/cc-pVTZ Level (264 Basis Functions) with Increasing Number of cores per Node[a]

no. of cores per node	timing	speedup
1	1542	14
2	1812	12
3	2360	9

[a] The computation has been performed using 16 cores.

speedup for benzene is closer to the optimal speedup. Using 16 nodes the speedup observed is 14 for one CC iteration and larger than 15 for the triples part (compare to 11−12 and 14 in the case of N_2O). Thus, the scaling of the total time with the number of processors is improving for increasing system size (129 basis functions and 16 correlated electrons for N_2O, 264 basis functions and 30 correlated electrons for benzene) as the importance of the parallelized, time-determining steps is even larger. It is worthwhile to note the drop in the speedup when using 24 instead of 22 cores. All the computations presented in this section have been carried out on 22 nodes with four cores each. The processes are distributed in a round-robin fashion, i.e., every node first obtains only one job. After that all remaining jobs are distributed in the same way. That means that for computations with up to 22 parallel processes only one core per node has been used. After that point at least some of the nodes need to handle more than one process at a time. The influence of this on the computation time is shown in Table 5 for a computation using 16 cores. The same calculation has been carried out using a different number of cores per node. While using one core per node the walltime observed is 25 min, the computation of one iteration takes about five minutes longer when the number of cores is increased from one to two yielding a speedup of 12 instead of 14. Increasing the number of cores per node again by one, the computational effort grows by additional 9 min, lowering the speedup to nine. Thus, when increasing the number of cores per node used in a calculation a loss of hardware efficiency is observed which worsens with a higher number of cores per node.

B. The 2,6-Pyridyne (C_5NH_3) Molecule. We further demonstrate the applicability of our parallelization scheme by investigating the 2,6-isomer of the didehydropyridine (pyridyne). This choice of example has been motivated by the fact that the chemically interesting didehydroarenes (arynes)[58−61] exhibit a strong biradical character and that their theoretical treatment remains a challenge until today. Multireference approaches are the natural choice for the theoretical treatment of arynes. The focus in the following will be on biradicals derived from pyridine, namely pyridyne. These systems have been recently investigated using density-functional theory (DFT), multiconfigurational SCF (MCSCF), the reduced multireference CC approach, and Mk-MRCC.[13,62−66]

A particularly intriguing aspect of *meta*-arynes is the possibility that they may exist as monocyclic or bicyclic compounds. While for *meta*-benzyne the bicyclic structure is understood to be an artifact displayed at various levels of theory,[67] the issue has not been fully settled in the case of the didehydro compounds of heteroarenes. An earlier study of the geometry of 2,6-pyridyne at the CCSD, CCSD(T), and Mk-MRCCSD level did not provide a final answer for this class of molecules.[13] While CCSD and Mk-MRCCSD predicts both forms to be a minimum (with the monocyclic form to be more stable at the Mk-MRCCSD level and the bicyclic form to be more stable for CCSD) CCSD(T) only yields a minimum for the monocyclic structure. In order to narrow down the problem high-level multireference ab initio calculations are necessary. The recent development of the Mk-MRCCSDT approach[14] allows the inclusion of full triple excitations. However, calculations at this level of theory are demanding and time-consuming when proper basis sets are to be used.

To extend the previous study of 2,6-pyridyne single-point energy calculations at the Mk-MRCCSDT/cc-pVTZ level have been carried out. The geometries of the bicyclic and the monocyclic form as well as the transition state are obtained at the Mk-MRCCSD/cc-pCVTZ level (see ref 13). Two closed-shell determinants are used as reference with the orbitals taken from restricted-Hartree-Fock (RHF) calculations. The first, Φ_1, corresponds to the HF solution, while the second, Φ_2, is obtained by replacing the highest-occupied MO in the monocyclic form (a_1 symmetry) by the lowest unoccupied MO (b_2 symmetry).

The resulting energies are depicted in Figure 5. For comparison, the single-point energies computed at the fc-CCSD and fc-Mk-MRCCSD levels have been added to the figure. The CCSD energy for the bicyclic form is lower (2.9 kcal mol^{-1}) than that for the monocyclic form. The energy of the transition state lies in between. Mk-MRCCSD yields a lower energy for the monocyclic isomer (3.9 kcal mol^{-1}) and a barrier of 0.2 kcal mol^{-1} from the bicyclic form. When the effect of triple excitations is included at the CCSDT and Mk-MRCCSDT levels of theory the monocyclic form is found to be lower in energy than the bicyclic one. The energy computed at the Mk-MRCCSD/cc-pCVTZ transition-state geometry lies in between. The inclusion of triple excitations

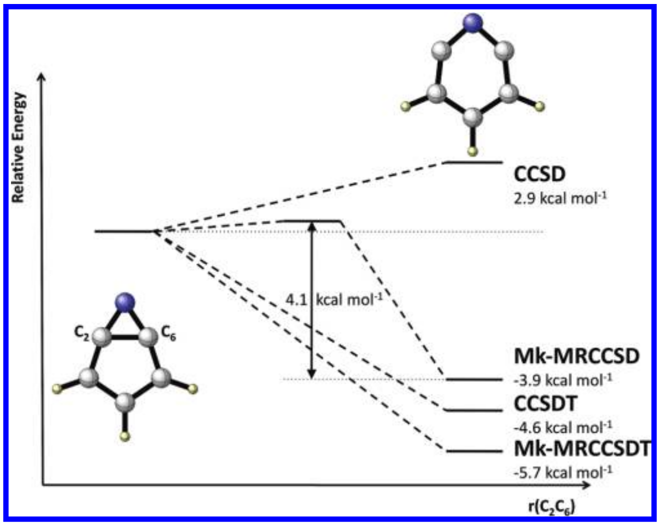


Figure 5. Energetic ordering of the monocyclic and bicyclic forms of 2,6-pyridyne as obtained in frozen-core computations at the CCSD, Mk-MRCCSD, CCSDT, and Mk-MRCCSDT levels of theory using the cc-pVTZ basis set and the RHF-Mk-MRCCSD/cc-pCVTZ geometries from ref 13. The energies are given relative to the bicyclic form.

Table 6. Timings (Walltime in s) for One CC Iteration for the Energy Computations of 2,6-Pyridyne at Geometries for the Bicyclic and Monocyclic Form As Well As the Transition State[a]

	bicyclic	transition state	monocyclic
Mk-MRCCSDT	10707	10718	10699
CCSDT	2529	2548	2526

[a] The frozen-core calculations have been carried out on 8 nodes (using 1 CPU each) employing the cc-pVTZ basis set (222 basis functions) and the Mk-MRCCSD/cc-pCVTZ geometry.

at the CCSDT level leads to a considerable stabilization of the monocyclic isomer (7.5 kcal mol^{-1}). The additional stabilization at the Mk-MRCCSDT level is significantly smaller (1.8 kcal mol^{-1}).

These results indicate that the energy minimum of the bicyclic isomer of 2,6-pyridyne may be an artifact of theory due to the lack of triple excitations. Interestingly, computations at the Mk-MRCCSD(T) level, where triple excitations are treated in an approximate manner, yield comparable findings supporting our conclusion.[64]

As the calculations at the Mk-MRCCSDT level are computationally demanding they have only been carried out using 8 and 16 nodes. The walltimes observed for the calculations are given in Table 6. The computation time for the monocyclic isomer is about 178 min on 8 nodes and about 102 min on 16 nodes yielding a speedup of 1.8. For the corresponding computation at the CCSDT level a speedup of 1.6 is observed.

IV. Conclusions

A detailed analysis of CCSDT and Mk-MRCCSDT energy computations indicates that these calculations are completely dominated by the determination of the triples amplitudes and their contributions to the singles and doubles amplitude equations. Therefore, a parallelization scheme is presented that starts from the efficient serial algorithm available within the CFOUR program package and parallelizes the most time-consuming parts in the calculation, namely steps related to the triple amplitudes. The central aspect of our parallel implementation is the distributed computation of the triple amplitudes as well as their contributions to the single and double amplitudes. In this way an algorithm is obtained for which sufficient local memory and disk space are needed but which does not depend on high-speed network connections. The parallel implementation presented here will be available in a future public release of the CFOUR program package.

Benchmark calculations demonstrate the applicability of our parallel algorithm. Computations at the CCSDT/cc-pCVTZ level of theory for the N_2O molecule (129 basis functions) exhibit a good scaling with the number of processors. In this case a speedup of 11 to 12 is observed on 16 processors. For larger systems the behavior is even better. For a frozen-core calculation on benzene at the CCSDT/cc-pVTZ level (264 basis functions) a speedup of 14 is obtained when 16 nodes are used. The excellent scaling behavior may be observed as long as the time-determining steps in the serial calculation dominate the overall time in the parallel computation. However, this might only be a problem when a larger number of nodes is used, as the computation on benzene still spends about 73% of the whole time in the triples part even when 64 processors are used.

We further present parallel Mk-MRCCSDT computations for ozone for which benchmark calculations at the Mk-MRCCSDT/cc-pCVTZ level (129 basis functions) have been carried out yielding speedups even closer to the optimal speedup than single-reference CCSDT. The speedup observed is 13 to 14 when 16 nodes are used. The applicability of the parallelization of the multireference algorithm is demonstrated by investigating the 2,6-pyridyne molecule. This compound requires a multireference treatment due to a strong biradical character and the inclusion of at least triple excitations for a qualitative correct description of the potential energy surface. Frozen-core computations at the Mk-MRCCSDT/cc-pVTZ level (222 basis functions) in combination with the parallelization scheme presented here become feasible in a resonable amount of time.

In this work we have demonstrated that by adapting an efficient serial algorithm to a parallel environment, calculations at the CCSDT and Mk-MRCCSDT level of theory are feasible within days or weeks instead of months or years even when larger basis sets are used or larger systems are investigated. Thus, this kind of implementation opens the field of application for CCSDT and Mk-MRCCSDT where high-accuracy is decisive.

V. Technical Details

All calculations were carried out on a 22 node cluster. The cluster nodes were equipped with two dual core Intel Xeon 5160 processors running at 2.93 GHz, 32 GB FB-DIMM RAM, and 8 striped SATA disks. For the network communication the onboard Gigabit Ethernet controler was used.

For the parallel implementation the message passing interface (MPI)[36] is used. The results presented here are obtained by using LAM/MPI.[68,69] The communication of the t_3-amplitudes is done by the MPI_BCAST subroutine. The triples contributions to the t_1- and t_2-amplitudes are communicated using the MPI_ALLREDUCE routine.

Note Added after ASAP Publication. This article was published ASAP on July 8, 2010. Equation 1 has been modified. The correct version was published on July 14, 2010.

Acknowledgment. This work has been supported by the Deutsche Forschungsgemeinschaft and the Fonds der Chemischen Industrie.

References

(1) Gauss, J. In *The Encyclopedia of Computational Chemistry*; Schleyer, P. v. R., Allinger, N. L., Clark, T., Gasteiger, J., Kollman, P. A., Schaefer, H. F., Schreiner, P. R., Eds.; Wiley: Chichester, 1998; p 615.

(2) Crawford, T. D.; Schaefer, H. F. *Rev. Comput. Chem.* **2000**, *14*, 33.

(3) Bartlett, R. J.; Musiał, M. *Rev. Mod. Phys.* **2007**, *79*, 291.

(4) Shavitt, I.; Bartlett, R. J. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*; Cambridge University Press: Cambridge, 2009; p 1.

(5) Mahapatra, U. S.; Datta, B.; Mukherjee, D. *Mol. Phys.* **1998**, *94*, 157.

(6) Mukherjee, D.; Moitra, R. K.; Mukhopadhyay, A. *Mol. Phys.* **1977**, *33*, 955.

(7) Jeziorski, B.; Monkhorst, H. J. *Phys. Rev. A* **1981**, *24*, 1668.

(8) Paldus, J.; Piecuch, P.; Pylypow, L.; Jeziorski, B. *Phys. Rev. A* **1993**, *47*, 2738.

(9) Hubač, I.; Pittner, J.; Čársky, P. *J. Chem. Phys.* **2000**, *112*, 8779.

(10) Hanrath, M. *J. Chem. Phys.* **2005**, *123*, 084102.

(11) Kong, L. *Int. J. Quantum Chem.* **2009**, *109*, 441.

(12) Evangelista, F. A.; Allen, W. D.; Schaefer, H. F. *J. Chem. Phys.* **2006**, *125*, 154113.

(13) Prochnow, E.; Evangelista, F. A.; Schaefer, H. F.; Allen, W. D.; Gauss, J. *J. Chem. Phys.* **2009**, *131*, 064109.

(14) Evangelista, F. A.; Simmonett, A. C.; Allen, W. D.; Schaefer, H. F.; Gauss, J. *J. Chem. Phys.* **2008**, *128*, 124104.

(15) Evangelista, F. A.; Allen, W. D.; Schaefer, H. F. *J. Chem. Phys.* **2007**, *127*, 024102.

(16) Raghavachari, K.; Trucks, G. W.; Pople, J. A.; Head-Gordon, M. *Chem. Phys. Lett.* **1989**, *157*, 479.

(17) Noga, J.; Bartlett, R. J. *J. Chem. Phys.* **1987**, *86*, 7041.

(18) Scuseria, G. E.; Schaefer, H. F. *Chem. Phys. Lett.* **1988**, *152*, 382.

(19) Helgaker, T.; Jørgensen, P.; Olsen, J. *Molecular Electronic Structure Theory*; Wiley: Chichester, 2001.

(20) Halkier, A.; Larsen, H.; Olsen, J.; Jørgensen, P.; Gauss, J. *J. Chem. Phys.* **1999**, *110*, 734.

(21) Sordo, J. A. *J. Chem. Phys.* **2001**, *114*, 1974–1980.

(22) Harding, M. E.; Vázquez, J.; Ruscic, B.; Wilson, A. K.; Gauss, J.; Stanton, J. F. *J. Chem. Phys.* **2008**, *128*, 114111.

(23) Császár, A. G.; Allen, W. D.; Schaefer, H. F. *J. Chem. Phys.* **1998**, *108*, 9751.

(24) Karton, A.; Rabinovich, E.; Martin, J. M. L.; Ruscic, B. *J. Chem. Phys.* **2006**, *125*, 144108.

(25) Feller, D.; Peterson, K. A. *J. Chem. Phys.* **2009**, *131*, 154306.

(26) Gauss, J.; Stanton, J. F. *J. Chem. Phys.* **2002**, *116*, 1773.

(27) Rendell, A. P.; Lee, T. J.; Lindh, R. *Chem. Phys. Lett.* **1992**, *194*, 84.

(28) Hirata, S. *J. Phys. Chem. A* **2003**, *107*, 9887.

(29) Harding, M. E.; Metzroth, T.; Gauss, J.; Auer, A. A. *J. Chem. Theory. Comput.* **2008**, *4*, 64.

(30) Lotrich, V.; Flocke, N.; Ponton, M.; Yau, A. D.; Perera, A.; Deumens, E.; Bartlett, R. J. *J. Chem. Phys.* **2008**, *128*, 194104.

(31) Janowski, T.; Ford, A. R.; Pulay, P. *J. Chem. Theory. Comput.* **2007**, *3*, 1368.

(32) Nieplocha, J.; Harrison, R. J.; Littlefield, R. In *Proceedings of Supercomputing*; IEEE Computer Society Press: Washington, DC, 1994; p 340.

(33) Ford, A. R.; Janowski, T.; Pulay, P. *J. Comput. Chem.* **2007**, *28*, 1215.

(34) Olson, R. M.; Bentz, J. L.; Kendall, R. A.; Schmidt, M. W.; Gordon, M. S. *J. Chem. Theory. Comput.* **2007**, *3*, 1312.

(35) Olson, R. M.; Schmidt, M. W.; Gordon, M. S.; Rendell, A. P. *Enabling the Efficient Use of SMP Clusters: The GAMESS/DDI Approach; In Supercomputing*; ACM/IEEE Conference: Phoenix, AZ, 2003; p 41.

(36) The MPIForum. MPI: a message passing interface. In *Proceedings of the 1993 ACM/IEEE Conference on Supercomputing*; ACM Press: Portland, OR, U.S.A., 1993.

(37) Piecuch, P.; Landman, J. I. *Parallel Comput.* **2000**, *26*, 913.

(38) OpenMP Fortran Application Programming Interface. http://www.openmp.org. Accessed March 12, 2010.

(39) Kállay, M.; Harding, M. E. Parallel version of the string-based general coupled-cluster program MRCC, 2006. See http://www.mrcc.hu. Accessed March 18, 2010.

(40) Bylaska, E. J. *NWChem, A Computational Chemistry Package for Parallel Computers, Version 5.1*; Pacific Northwest National Laboratory: Richland, Washington 99352-0999, U.S.A., 2007.

(41) CFOUR, a quantum chemical program package written by Stanton, J. F.; Gauss, J.; Harding, M. E. Szalay, P. G. with contributions from Auer, A. A.; Bartlett, R. J.; Benedikt, U.; Berger, C.; Bernholdt, D. E.; Bomble, Y. J.; Christiansen, O.; Heckert, M.; Heun, O.; Huber, C.; Jagau, T.-C.; Jonsson, D.; Jusélius, J.; Klein, K.; Lauderdale, W. J.; Matthews, D. A.; Metzroth, T.; O'Neill, D. P.; Price, D. R.; Prochnow, E.; Ruud, K.; Schiffmann, F.; Stopkowicz, S.; Vázquez, J.; Wang, F.; Watts, J. D. and the integral packages MOLECULE (Almlöf, J.; Taylor, P. R.),PROPS (Taylor, P. R.), ABACUS (Helgaker, T.; Jensen, H. J. Aa.; Jørgensen, P.; Olsen, J.),and ECP routines by Mitin, A. V.; van Wüllen, C. For the current version, see http://www.cfour.de. Accessed February 18, 2010.

(42) Gauss, J.; Stanton, J. F. *Phys. Chem. Chem. Phys.* **2000**, *2*, 2047.

(43) Woon, D. E.; Dunning, Jr, T. H. *J. Chem. Phys.* **1995**, *103*, 4572.

(44) Mahapatra, U. S.; Datta, B.; Mukherjee, D. *J. Chem. Phys.* **1999**, *110*, 6171.

(45) R(NN) = 1.1290 Å, R(NO) = 1.1849 Å, ∠(NNO) = 180.0°.

(46) The absolute energy obtained is−184.572494 hartree.

(47) The speedup is defined as the ratio of the serial execution time and the execution time on *n* cores.

(48) Urban, M.; Noga, J.; Cole, S. J.; Bartlett, R. J. *J. Chem. Phys.* **1985**, *83*, 4041.

(49) Noga, J.; Bartlett, R. J.; Urban, M. *Chem. Phys. Lett.* **1987**, *134*, 126.

(50) Koch, H.; Christiansen, O.; Jørgensen, P.; Sanchez de Merás, A. M.; Helgaker, T. *J. Chem. Phys.* **1997**, *106*, 1808.

(51) Dunning, Jr, T. H. *J. Chem. Phys.* **1989**, *90*, 1007.

(52) The absolute energies obtained are (in hartree): CC3:−184.595095, CCSDT-1a:−184.594467, CCSDT-1b:−184.594525, CCSDT-2:−184.588903, CCSDT-3:−184.589449.

(53) The absolute energy obtained is−225.306818 hartree.

(54) Tanaka, T.; Morino, Y. *J. Mol. Spectrosc.* **1970**, *33*, 538.

(55) Harding, M. E.; Gauss, J.; Pflüger, K.; Werner, H.-J. *J. Phys. Chem. A* **2007**, *111*, 13623.

(56) Vázquez, J.; Harding, M. E.; Gauss, J.; Stanton, J. F. *J. Phys. Chem. A* **2009**, *113*, 12447.

(57) Gauss, J.; Stanton, J. F. *J. Phys. Chem. A* **2000**, *104*, 2865.

(58) Myers, A. G.; Cohen, S. B.; Kwon, B. M. *J. Am. Chem. Soc.* **1994**, *116*, 1670.

(59) Schottelius, M. J.; Chen, P. *J. Am. Chem. Soc.* **1996**, *118*, 4896.

(60) Hoffner, J.; Schottelius, M. J.; Feichtinger, D.; Chen, P. *J. Am. Chem. Soc.* **1998**, *120*, 376.

(61) Winkler, M.; Cakir, B.; Sander, W. *J. Am. Chem. Soc.* **2004**, *126*, 6135.

(62) Debbert, S. L.; Cramer, C. J. *Int. J. Mass. Spectrom.* **2000**, *201*, 1.

(63) Li, X.; Paldus, J. *J. Chem. Phys.* **2008**, *129*, 174101.

(64) Evangelista, F. A.; Prochnow, E.; Gauss, J.; Schaefer, H. F. *J. Chem. Phys.* **2010**, *132*, 074107.

(65) Jagau, T.-C.; Prochnow, E.; Evangelista, F. A.; Gauss, J. *J. Chem. Phys.* **2010**, *132*, 144110.

(66) Manohar, P. U.; Koziol, L.; Krylov, A. I. *J. Phys. Chem. A* **2009**, *113*, 2591.

(67) Smith, C. E.; Crawford, T. D.; Cremer, D. *J. Chem. Phys.* **2005**, *122*, 174309.

(68) Burns, G.; Daoud, R.; Vaigl, J. LAM: An Open Cluster Environment for MPI. *Proceedings of Supercomputing Symposium*; 1994; p 379.

(69) Squyres, J. M.; Lumsdaine, A. A Component Architecture for LAM/MPI. *Proceedings, 10th European PVM/MPI Users' Group Meeting*, Venice, Italy, 2003; p 379.

CT1002016