

GAMESSのMCQDPT2計算についての並列化ベンチマーク

梅田 宏明^{†1,†2} 小関 史朗^{†1}
長嶋 雲兵^{†3} マイケル W. シュミット^{†4}

Distributed Data Interface (DDI) ライブラリを用いて GAMESS の MCQDPT2 (multiconfiguration quasi-degenerate second-order perturbation theory) ルーチンを並列化した。特に大きな分子を計算する際に問題となる二電子積分の積分変換部分についても、リモートメモリアクセスを利用することにより、高い並列化性能を得ることができた。ベンチマーク計算として比較的大きな有機分子についての MCQDPT2 計算を行ったところ、特に PC クラスタにおいて高い並列化性能を示している。

Benchmark Tests on Parallelized MCQDPT2 Calculations with GAMESS

HIROAKI UMEDA,^{†1,†2} SHIRO KOSEKI,^{†1} UMPEI NAGASHIMA^{†3}
and MICHAEL W. SCHMIDT^{†4}

The multiconfiguration quasi-degenerate second-order perturbation theory (MCQDPT2) routines in GAMESS suite of program codes have been parallelized using a distributed data interface (DDI). When remote memory access used, high parallelization speedups are achieved even in two electron integral transformation. For benchmark calculations, we apply our parallelization code to relatively large organic molecules. It shows high efficiency especially for PC clusters.

1. はじめに

近年の計算機システムの著しい発達によってもたらされた科学の様々な分野における研究の発展は驚くべきものがある。特に Gaussian, GAMESS, MOLCAS, MOLPRO, NWChem に代表される量子化学分野における様々なプログラムは、多くの化学者や生物学者にとって欠かすことができないツールとなった。実際、分子構造の決定や反応機構の解明等の特に有機化学の分野では、これらのプログラムはすでに実用的なツールとして、実験家にも広く使われている。

計算化学の分野においては、並列計算に適したアルゴリズムを利用することで、より大きく、より高度

なレベルの計算を実現するための努力が進められている。たとえば化学の非経験的分子軌道計算の分野で最も広く有効に使われている高精度な計算法の 1 つに MRMP2 (multireference second-order Møller-Plesset perturbation theory) 法がある。代表的な非経験的分子軌道計算プログラムの 1 つである GAMESS¹⁾ で実装されている MRMP2 計算ルーチンは、Nakano らが開発した MCQDPT2 (multiconfiguration quasi-degenerate second-order perturbation theory) 法²⁾ と呼ばれる複数の状態を同時に扱うことのできる MRMP2 法を実装したものであるが、比較的小さな分子について計算するのが限度であった。そこで我々はこれまで、この MCQDPT2 ルーチンの並列化に取り組んできた³⁾。

GAMESS の並列化には、専用の通信ライブラリである distributed data interface (DDI)⁴⁾ が利用されている。DDI に実装されているリモートメモリアクセスにより、中程度の大きさの分子に対する MCQDPT2 計算について 8 ノードの PC クラスタで 6.8 から 7.2 という高い並列化速度向上を記録した³⁾。しかしながら実際の有機化学、生物化学といった分野の研究では、

†1 大阪府立大学

Osaka Prefecture University

†2 富士総合研究所

Fuji Research Institute Corporation

†3 産業技術総合研究所

National Institute of Advanced Industrial Science and Technology

†4 アイオワ州立大学

Iowa State University

より大きな分子に対する MCQDPT2 計算が望まれており、これらの分子についての計算可能性を示すことが必要とされている。

この論文では比較的大きな分子に対する MCQDPT2 計算を行ったのでそれを報告する。この大きさの分子についての計算では、コアメモリ上のみすべてのデータを保存することは困難であり、ディスクアクセスを効果的に実行することも重要になってくる。2章では大規模分子の MCQDPT2 計算で特にボトルネックとなる二電子積分の積分変換部分の並列化についてそのアルゴリズムを解説し、3章で実際のベンチマーク結果を報告する。

2. 積分変換の並列化

二電子積分の積分変換は

$$(ij|kl) = \sum_{p,q,r,s} C_{pi} C_{qj} C_{rk} C_{sl} (pq|rs) \quad (1)$$

で示される4個のインデックスについての係数変換として構成されており、ハートリー・フォックレベル以上の高精度な計算を行う場合には不可欠なものである。ここで (...) は二電子積分を示し、 p, q, r, s および i, j, k, l はそれぞれ原子軌道と分子軌道についてのインデックスであり、 $C_{pi}, C_{qj}, C_{rk}, C_{sl}$ はそれらの間の変換係数である。たとえば C_{sl} を用いて二電子積分 $(pq|rs)$ を以下のように $(pq|rl)$ に変換することができる。

$$(pq|rl) = \sum_s^{N_{AO}} C_{sl} (pq|rs) \quad (2)$$

ここで N_{AO} は原子軌道の数である。変換後の分子軌道の数を N_{MO} とした場合、この積分変換の第1段階である式(2)をすべての p, q, r, l について実行するのに必要な演算量は、 $N_{AO}^4 N_{MO}$ となる。得られた $(pq|rl)$ について第2段階の変換 ($r \rightarrow k$) を行うためには $N_{AO}^3 N_{MO}^2$ の演算が必要である。以下、第3、第4段階の変換についてもそれぞれ $N_{AO}^2 N_{MO}^3$ 、 $N_{AO} N_{MO}^4$ の演算が必要となる。一般にはすべての分子軌道についての二電子積分を要求されることはあまりないが、形式的に $N_{MO} \approx N_{AO}$ と考えると、この積分変換全体の演算量は $O(N_{AO}^5)$ と評価される。このため、分子が大きくなる (= 原子軌道の数 N_{AO} の増加) による二電子積分の数および演算量の増大は深刻であり、大きな分子を扱ううえでの障害となっていた。

この積分変換を並列化する場合、全対全の通信が多数発生するため並列化性能は非常に悪くなることが予想される。一例として最初の2段階の積分変換の後に、

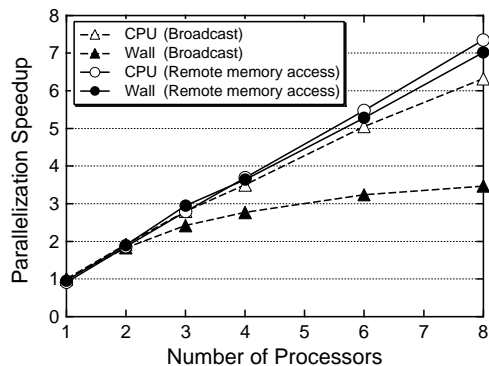


図1 積分変換部分の並列化速度向上

Fig. 1 Parallelization speedups of integral transformation routines.

ブロードキャスト型の通信によって他のノードと必要なデータを交換し、残りの2段階の積分変換を行うアルゴリズムにおける並列化効率を測定してみた。図1は中程度の大きさの有機分子 (4a,4b,8a,9a-tetrahydropyridino[1',2'-4,3]imidazolidino[1,5-a]pyridine)³⁾ の積分変換部分のみについての並列化効率を示した図である。この分子の MCQDPT2 計算には 219 個の原子軌道が関与し、生成される分子軌道 (frozen core, doubly occupied, active および external 軌道) の数はそれぞれ 13, 28, 8, 170 個となっている。また計算は Pentium3 866 MHz (cache 256 KB), 512 MB memory, 15 GB IDE HDD の計算機を 100 M ethernet で接続した Linux (kernel 2.4.2) クラスタ上で実行している。GAMESS のコンパイルは g77 2.95.2 (-O2 -malign-double) で行い、行列積ライブラリとして ASCI の Pentium2 用の BLAS ライブラリを利用した。

図1の破線がブロードキャストを用いた積分変換による並列化効率を示している。8ノードの場合の速度向上は Wall time で 3.47 にしかなっていない。これは CPU time についての速度向上が 6.32 であることと比較しても非常に悪い値であり、通信に関する部分のオーバーヘッドが大きいことが推測される。

このオーバーヘッドを最小限に抑え計算機資源を有効に活用するためには、リモートメモリアクセスの利用が有効である。我々が利用した DDI ライブラリには “SEND” や “RECIEVE” 等の通常のメッセージパッシング用の関数だけでなく、すべてのノードにわたる巨大な分散データ領域を利用するリモートメモリアクセスの機能が用意されており⁴⁾、これを利用した並列化が容易に実現できる。この分散データ領域からデータを取り出す “GET” 関数や分散データ領域に

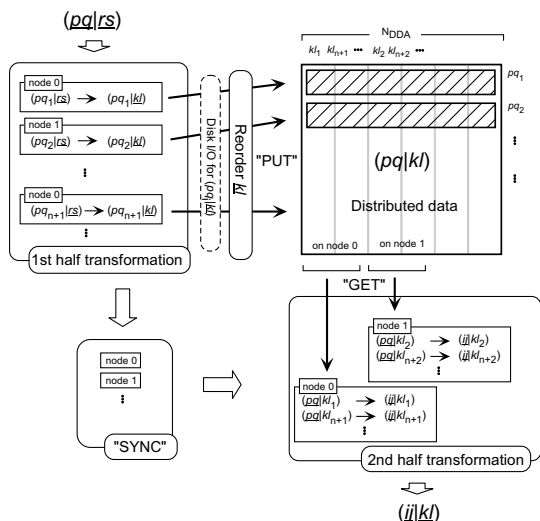


図 2 リモートメモリアクセスを利用した積分変換の並列化
Fig. 2 Parallel integral transformation with remote memory access.

データを代入する“PUT”関数等の通信は一方非同期通信であるため、他のノードの影響を受けることなく実行することができ、通信に関するオーバーヘッドを抑えることが可能となる。実際、文献 5) では RHF (restricted Hartree Fock) +MP2 法のためにリモートメモリアクセスを利用した積分変換を実装し、高い並列化効率を記録している。しかしながら文献 5) の積分変換ルーチンは、オンメモリでの計算を前提条件としているため、必要とするメモリ量の制約から大きい分子についての計算に利用することが現実問題として不可能であった。

そこで我々はリモートメモリアクセスを利用した、大規模な MCQDPT2 計算にも適用可能な新しい積分変換アルゴリズムを提案し、並列化性能の向上を図った。

図 2 にそのアルゴリズムの概略を示した。DDI における分散データ領域は二次元の配列として表現され、二次元配列の列を各ノードに分散させて確保することにより実現されている(図 2 右上の領域)。アルゴリズムの簡単化のため、分散データ配列として確保される二次元配列 DDA の列の数 N_{DDA} をノード数 N_{node} の整数倍とした。すなわち配列の $1 \sim N_{DDA}/N_{node}$ 列はノード 0 上、 $N_{DDA}/N_{node} + 1 \sim 2N_{DDA}/N_{node}$ 列はノード 1 上に確保される。ただし図 2 ではノード数 N_{node} を n で示している。

積分変換は二電子積分の持つ対称性から 2 つのインデックスを対にして扱うことが多いが、ここでも p, q および r, s を pq および rs の 2 つのインデックス

にまとめて考えることとする。

図 3 にコードの概略を示した。まず pq インデックスを各ノードに分配する。図 3 中 $inode$ は各ノードに付けられた番号 ($0 \sim N_{node}-1$) である。分配された pq に対し $(pq|rs) \rightarrow (pq|kl)$ の変換が実行される。ここで rs のように下線で示したインデックスは、そのすべての要素について取り扱うことを意味するものとする。変換された二電子積分 $(pq|kl)$ はただちに分散データ配列に転送 (“PUT”) される。この転送はリモートメモリアクセスによる一方通信であるため、ブロッキングによるオーバーヘッドは発生しない。

ここで注意しておくべきことは、二電子積分 $(pq|kl)$ を分散データ配列に転送する前に、インデックス kl に関してデータの並びを変更しておくことである。DDI におけるリモートメモリアクセスの実装では、図 2 に見られるように分散データ領域にとられた二次元配列の各列をそれぞれのノードに分配するものとなっている。これを考慮して、

$$kl' = (kl - 1)/N_{node} + \text{mod}(kl - 1, N_{node}) * (N_{DDA}/N_{node}) + 1 \quad (3)$$

のような kl から kl' への並べ換えを行う。並べ換えられた二電子積分 $(pq|kl')$ を分散データ配列上の pq 行に転送した場合、ノード 0 上に確保された分散データ配列には $kl' = 1, 2, \dots$ すなわち $kl = 1, N_{node} + 1, 2N_{node} + 1, \dots$ が配置されることになるが、この $kl = 1, N_{node} + 1, 2N_{node} + 1, \dots$ の二電子積分は、ノード 0 で実行される後半の積分変換 ($(pq|kl) \rightarrow (ij|kl)$) に必要なデータである。このように kl から kl' へとデータの並びを変更したうえで分散データ配列に二電子積分を転送することにより、後半の積分変換プロセスに必要なデータ転送をノード内のローカルな転送のみに制限することができ、ノードをまたいだ遅いデータ通信を削除することが可能になるのである。

前半の積分変換がすべて終了した時点でノード間の同期をとる。これにより分散データ配列の内容を確定し、後半の $(pq|kl) \rightarrow (ij|kl)$ の積分変換を行う。

各ノードには kl のインデックスが割り振られ、二電子積分 $(ij|kl)$ の作成に必要なデータ $(pq|kl)$ が分散データ配列から各ノード上のローカルな配列へとコピー (“GET”) される。この分散データ領域からの転送は、先に記述したようにノード内のデータ転送だけで実行できる。 $(pq|kl) \rightarrow (ij|kl)$ 変換後の二電子積分は、以降の MCQDPT2 計算における並列計算処理に利用するため、各ノードに分散した状態のままディスク装置等に保存される。

```

call DDI_CREATE(DDA,Npq,NDDA)  c 分散データ配列 DDA(Npq,NDDA) の確保

do pq=1+inode,Npq,Nnode
  read (pq|rs)
  (pq|rs) -> (pq|kl)          c 積分変換
  do kl=1,Nkl                c 順序変更
    kl'=(kl-1)/Nnode + mod(kl-1,Nnode)*(NDDA/Nnode)+1
    (pq|kl')' = (pq|kl)
  end do
  call DDI_PUT (pq|kl')' -> DDA(pq:pq,1:NDDA)  c 分散データ配列への転送
end do

call DDI_SYNC                c 同期

do kl=1+inode,Nkl,Nnode
  kl'=(kl-1)/Nnode + mod(kl-1,Nnode)*(NDDA/Nnode)+1
  call DDI_GET (pq|kl) <- DDA(1:Npq,kl':kl')  c 分散データ配列からの転送
  (pq|kl) -> (ij|kl)          c 積分変換
  write (ij|kl)
end do

```

図 3 リモートメモリアクセスを利用した積分変換の並列化コード

Fig. 3 Parallel integral transformation code with remote memory access.

実際の研究の現場で利用される計算機環境のもとではさらに別の問題も発生する。大きな分子の MC-QDPT2 計算では二電子積分の数が非常に多くなってしまいうため、すべての二電子積分を分散データ領域に格納することが困難になるのである。特に多くの研究者の利用する比較的小さな PC クラスタ環境ではこのような状況が頻繁に発生すると考えられ、十分なメモリを確保できない状況でも MCQDPT2 計算を可能にするために、中間ファイルに二電子積分の値を保存する機構を用意しなければならない。

今回の並列化では積分変換の前半部分 ($(pq|rs) \rightarrow (pq|kl)$) の変換後、分散データ領域に転送する前の段階で、二電子積分をディスクに保存することができる仕組みを用意した。ディスクファイルへの保存は自ノードに割り当てられたインデックス pq についての二電子積分のみを保存すればよいため、遅いディスクアクセスが他のノードの計算に影響を与えることはない。その後のプロセスは、積分変換の前半部を行う代わりにディスクファイルから二電子積分を読み込むよう変更するだけですむ。

ここでディスクファイルからの読み込み部分を単純に積分変換前半部分の再計算に変更すれば遅いディスクアクセスを避けながら大規模計算をすることも可能になる。しかしながら今回はノードあたりのディスクアクセスの速度が比較的速い PC クラスタタイプの並列計算機を主なターゲットとしてしており、再計算のコストは相対的に大きくなると思われる。そのため今回はこの方法を実装していないが、その変更自体は容易である。

図 2 で示されたりモートメモリアクセスを利用する並列化により、積分変換ルーチンの並列化効率は著し

く改善された(図 1 実線)。8 ノードの PC cluster の場合の並列化による速度向上は CPU time で 7.32 であり、WALL time についてさえ 7.02 と非常に効果的に並列化が実現されていることが分かる。この計算ではノード数が 8 以外の場合には積分変換の途中にディスクファイルへ二電子積分を保存しているが、ディスクアクセス部分も含めて並列化しているため高い並列化効率が保たれている。

中程度の大きさである分子についての MCQDPT2 計算ですら、二電子積分の積分変換部分が全体の計算時間の 8 割程度を占めることも多く、分子が大きくなれば積分変換部分に費やされる時間がさらに増大することも予想される。つまり、この部分をきわめて効果的に並列化することがより大きな分子についての MCQDPT2 計算の実現に直結するのである。

3. ベンチマーク計算

今回のコードの実用性をテストするために、いくつかのマシンの上で比較的大きな分子に対するベンチマーク計算を行った。ベンチマーク計算のためのサンプル分子としてここでは 3 つの有機分子をとりあげた。またベンチマーク計算に利用したマシンは

- Pentium3 866 MHz (cache 256 KB), 512 MB memory, 60 GB IDE HDD, 100 M ethernet × 8 ノード, Linux (kernel 2.4.2), g77 2.95.3 (-O2 -malign-double), ATLAS 3.2.0

ブロードキャストを利用した場合の結果(破線)も、ディスクアクセス部分を並列化したルーチンを利用している。ブロードキャスト型の並列化とは図 2 で分散データ配列を利用して行っている行列の転置操作をブロードキャストを利用して行ったものを指す。

表 1 Test 1 についてのベンチマーク結果 (図 4)
Table 1 Benchmark results for Test 1 shown in Fig. 4.

CPU		Pentium3	Pentium3	Alpha	Alpha	Alpha	Pentium4	Pentium4
N_{node}		8 nodes	8 nodes	2 nodes	2 nodes	2 nodes	4 nodes	4 nodes
Type	Time	cluster	cluster	SMP	cluster	cluster	cluster	cluster
AOINTS	(秒)	DIST	DUP	DIST	DIST	DUP	DIST	DUP
Total	(CPU)	11781.4	13171.4	19638.3	19584.9	21016.0	7398.2	8538.7
	(Wall)	29983.6	24233.6	34660.0	32970.0	32230.0	21221.7	23498.2
積分変換	(CPU)	10212.8	9620.5	15547.4	16039.3	16239.0	5853.8	5821.9
	(Wall)	28190.6	19675.4	29347.0	29028.0	26647.0	19385.8	19093.4
有効ハミルトニアンの 評価	(CPU)	923.6	923.1	3199.2	2671.7	2716.6	1101.3	1104.4
	(Wall)	958.2	957.5	3399.0	2684.0	2728.0	1111.3	1117.5

- Alpha 21264 667MHz dual CPU (cache 4 + 4 MB), 4 GB memory , 73 GB UltraWide SCSI HDD , 100 M ethernet × 2 ノード Tru64 5.0 A , Compaq Fortran 5.2 (-O4) , CXML 4.1
- Pentium4 1.7 GHz(cache 256 KB) , 1 GB memory , 30 GB IDE HDD , 1000 M ethernet × 4 ノード , Linux (kernel 2.4.4) , g77 2.95.3 (-O2 -malign-double) , ATLAS 3.3.0

の 3 種類である . 特に Alpha CPU を用いた計算機については共有メモリ型の並列計算機 (SMP) として利用できるため , これも利用した . SMP タイプの並列計算機では他の分散メモリ型の並列計算機 (クラスタ) と比較して , 通信が非常に速いがディスクへのアクセスが遅いという特徴を持つため , SMP タイプとクラスタタイプの並列計算機を比較することにより , 本並列アルゴリズムの特徴を知ることができる .

3.1 Test 1: 6, 7, 8, 13a, 13b-pentahydro-dipyrido-[1,2-a:2',1'-c][1,4]diazepine

この分子は図 4 で示される構造を持ち , 異性化の途中でラジカル中間体を生じる有機分子である . ラジカル化合物の電子状態は多くの安定な分子が持つ閉殻電子構造ではなく , 不対電子の存在する開殻電子構造を持つ . このような電子状態を持つ分子の分子軌道計算には , 動的な電子相関を十分に取り入れた MCQDPT2 のような高レベルの計算法が不可欠である .

この分子は基底関数として TZV(d,p) を利用した場合 , 396 個の原子軌道を持つ . また MCSCF (もしくは CASSCF) の active space を 8 個の軌道と 10 個の電子とした場合に MCQDPT2 計算で生成される reference CSF (configuration state function) は 600 個であり , frozen core , doubly occupied , およ

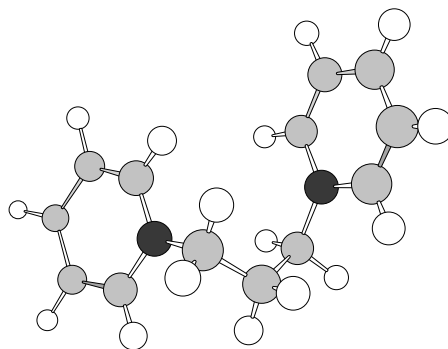


図 4 Test 1: 6, 7, 8, 13a, 13b-pentahydro-dipyrido-[1,2-a:2',1'-c][1,4]diazepine
Fig. 4 Test 1: 6, 7, 8, 13a, 13b-pentahydro-dipyrido-[1,2-a:2',1'-c][1,4]diazepine.

び external 軌道はそれぞれ 15 , 34 , 339 個である . どのマシンについても , クラスタ全体で 1.9 GBytes の分散データ領域を確保するように設定してベンチマークテストを行った (表 1) .

表 1 に MCQDPT2 計算の主要な部分についての CPU time と Wall time を示した . 全体の計算時間のほとんどが二電子積分の変換に費やされていることが分かる .

この分子の MCQDPT2 計算では GAMESS の生成する原子軌道ベースの二電子積分 , 積分変換ルーチンのために並べ換えを行った原子軌道ベースの二電子積分 , 積分変換前半終了時の二電子積分を保存するためにそれぞれ 10 ~ 15 GBytes のファイルを生成する . 原子軌道ベースの二電子積分ファイルが複数存在しているのは , GAMESS の二電子積分生成ルーチンが生成する二電子積分を , MCQDPT2 ルーチンで利用する順序に並べ換える必要があるからである . 表 1 中の AOINTS はこの GAMESS が最初に生成する二電子積分の保存法に関するオプションであり , “DUP” はすべての二電子積分を含むディスクファイルをすべて

のノードに作成することを意味し, “DIST” では二電子積分を各ノードに均等に分割して保存することを意味する. AOINTS オプションを “DUP” にした場合は “DIST” である場合に比較し, ディスクファイルのサイズが増加してしまうデメリットがあるものの, すべての二電子積分をローカルに保持しているためネットワークを利用した通信を大幅に減少させることが可能になる. AOINTS オプションを “DIST” にした場合, 各ノードに作られるファイルサイズは減少するものの, 積分変換に必要な二電子積分をノード間で転送する必要があるため高速なネットワークが要求される. 特に GAMESS の MCQDPT2 ルーチンでは, 積分変換を 2 度実行することもあり, 比較的遅い 100 M ethernet を利用している PC クラスタでは AOINTS オプションを “DUP” に設定するのが妥当であると考えられる. 実際, 表 1 の結果では, Pentium3, Alpha を 100 M ethernet で接続したクラスタシステムについて AOINTS オプションが “DUP” である場合により良いパフォーマンスをみせている. 一方で高速なデータ転送が可能な 1000 M ethernet で接続したマシンでは AOINTS オプションが “DIST” であっても “DUP” である場合に比べて大きな差がみられない. 全体の時間でみた場合には二電子積分の作成が並列化されていない分, むしろ “DIST” が有利になっている.

積分変換に際し生成される二電子積分のディスクファイル(並べ換えた原子軌道ベースの二電子積分, 積分変換前半終了時の二電子積分)は各ノードに分散して保存されるために, ノード数が多いほどそれぞれのファイルサイズが小さくなるようになっている. このためディスクシステムがノード数に比べて少ない SMP タイプのシステムでは高速な通信が可能であるにもかかわらず, Wall time ではクラスタシステムよりも遅くなってしまっている.

ディスクアクセスが問題となる今回のケースにおいては, 一般にはノード数の多い(ハードディスク装置

の多い)クラスタが有利になると考えられるが, このテストにおいて最も良いパフォーマンスをみせたのは Pentium4 CPU 4 ノードのクラスタであった. これは最も時間を費やしている積分変換部分で利用する行列積ルーチンが Pentium4 CPU において非常に高速に動作することに起因している.

3.2 Test 2: difluorobenzo[c]phenanthrene
difluorobenzo[c]phenanthrene (図 5) は, その安定構造が光学異性体を持ち, その反応過程が比較的少ない振動運動で記述できることから, 光学異性体の光反応制御に関する理論的研究についての重要なターゲット分子の 1 つと考えられている.

TZV(d,p) 基底関数のもとでは, 480 個の原子軌道が利用される. また frozen core, double occupied, active および external 軌道はそれぞれ 20, 43, 9, 408 個である. このとき 12 個の active electron が存在し, 2520 個の reference CSF が形成されることになっている. また計算では, それぞれのシステムで利用可能なメモリ量を考慮して Pentium3 クラスタに対しては 0.8 GBytes, Alpha マシンについては 1.6 GBytes, Pentium4 クラスタに対しては 2.5 GBytes の分散データ領域を利用した.

表 2 にベンチマーク計算の結果を示す. このベンチマーク計算で特徴的なのは, その軌道数の多さである. 二電子積分ファイルの大きさは最も大きいもので

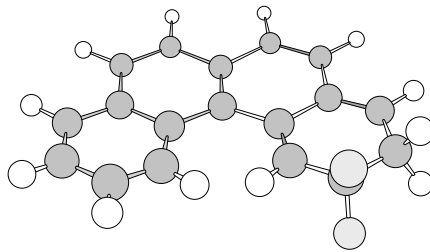


図 5 Test 2: difluorobenzo[c]phenanthrene
Fig. 5 Test 2: difluorobenzo[c]phenanthrene.

表 2 Test 2 についてのベンチマーク結果 (図 5)
Table 2 Benchmark results for Test 2 shown in Fig. 5.

CPU		Pentium3	Pentium3	Alpha	Alpha	Pentium4
N_{node}		8 nodes	8 nodes	2 nodes	2 nodes	4 nodes
TYPE	Time	cluster	cluster	cluster	cluster	cluster
AOINTS	(秒)	DIST	DUP	DIST	DUP	DIST
Total	(CPU)	38287.2	41888.2	71070.2	77653.1	30207.9
	(Wall)	116208.0	74456.6	119195.0	107478.0	84493.7
積分変換	(CPU)	27810.9	25945.2	40601.7	43934.4	16591.9
	(Wall)	104731.8	56145.7	87132.0	71704.0	70023.8
有効ハミルトニアン の評価	(CPU)	8940.0	8942.9	27962.2	27888.2	12546.6
	(Wall)	9116.6	9119.8	28175.0	28128.0	12657.5

30 GBytes にもなってしまう。このような場合 AOINTS オプションの設定によるパフォーマンスの差がはっきりと表れる。特に Pentium3 クラスタについては、AOINTS オプションによって 2 倍に近い性能差が積分変換部分に表れてきている。

2 ノード分のディスクファイルを作成するのに十分なハードディスク領域が確保できないため、今回利用した SMP タイプの計算機では計算自体が実行不可能であった。しかしながら相対的にディスクへのアクセス速度が遅くなる SMP タイプの計算機では、このようにディスクアクセスが多くなる計算に対して十分なパフォーマンスが出ないことは容易に想像される。

今回のベンチマーク計算では Test 1 のケースと比較してもさらにディスクファイルへのアクセスが多くなるため、ノード数の多い(ハードディスク装置の多い) Pentium3 8 ノードの PC クラスタが最も良いパフォーマンスを記録した。

3.3 Test 3: 7-aza-indole dimer

最後のベンチマーク計算としてとりあげるのは 7-aza indole の二量体である(図 6)。この分子に対しては 6-31G(d,p) 基底を利用した。原子軌道数は 330 であり、frozen core, doubly occupied, active, ex-

ternal orbital の数はそれぞれ 18, 38, 12, 262 個であり、active 電子は 12 個となる。その結果、生成される reference CSF は 113456 個にのぼり、このベンチマーク計算を特徴付けている。

表 3 に、Test 3 のベンチマーク結果を示す。reference CSF の多さは MCQDPT2 計算本体の計算時間の増加に表れる。このため表 3 では積分変換より MCQDPT2 計算に費やされる時間がはるかに大きくなっている。

MCQDPT2 計算(特に有効ハミルトニアン of 行列要素の評価)は reference CSF のインデックスに関するループから構成されているため容易に並列化が可能であった³⁾。有効ハミルトニアン of 評価で問題になるのは、行列要素の評価に必要な 2 体の結合定数を保存するディスクファイルの大きさが 6 GBytes 程度になってしまうことである。このファイルは各ノードに分割されないため、ディスクアクセスが 1 台のハードディスク装置に集中する SMP マシンではディスクアクセスの時間が全体のパフォーマンスを低下させてしまっている。

このベンチマークテストでは SMP タイプの計算機以外については比較的 CPU 速度そのもののパフォーマンスを反映した結果となった。

3.4 検 討

今回行ったベンチマーク計算では、分子の軌道数が多い Test1 および Test2 の場合には積分変換が、CSF の数が多い Test3 の場合には MCQDPT2 計算の有効ハミルトニアン of 評価が、それぞれ多くの計算時間を費やしている。

Test1 および Test2 では積分変換のための演算そのものは並列化されて高速に実行できているものの、ディスクアクセスの多さが全体のパフォーマンスを下げている。この部分での改良が求められる。Test2 に比べると小さな分子を扱った Test1 では、ディスク

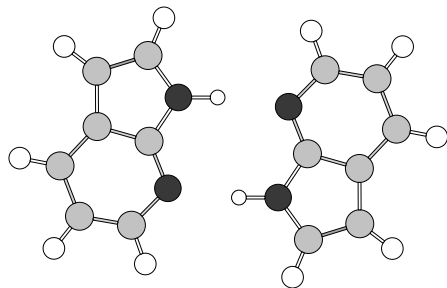


図 6 Test 3: 7-aza-indole dimer
Fig. 6 Test 3: 7-aza-indole dimer.

表 3 Test 3 についてのベンチマーク結果(図 6)
Table 3 Benchmark results for Test 3 shown in Fig. 6.

CPU		Pentium3	Pentium3	Alpha	Alpha	Alpha	Pentium4	Pentium4
N_{node}		8 nodes	8 nodes	2 nodes	2 nodes	2 nodes	4 nodes	4 nodes
Type	Time	cluster	cluster	SMP	cluster	cluster	cluster	cluster
AOINTS	(秒)	DIST	DUP	DIST	DIST	DUP	DIST	DUP
Total	(CPU)	54621.2	55248.5	177390.5	159365.8	160710.7	70934.2	71411.4
	(Wall)	59198.4	58171.0	182704.0	163218.0	162134.0	73911.7	73702.8
積分変換	(CPU)	2947.3	2963.0	6033.7	6096.6	5937.3	1800.5	1715.2
	(Wall)	6456.2	4655.7	8706.0	9301.0	6596.0	3839.5	2867.5
CAS CI	(CPU)	2947.1	2947.2	8215.2	8135.1	8157.2	3481.7	3497.7
計算	(Wall)	3092.4	3082.3	8489.0	8191.0	8226.0	3600.1	3598.0
有効ハミルトニアン of	(CPU)	48172.9	48184.5	158918.6	141495.7	142605.6	65289.0	65474.0
評価	(Wall)	48782.7	48793.9	160967.0	141929.0	143054.0	65925.7	66217.5

アクセスの影響が比較的小さいために行列積ルーチン
の高速性から Pentium4 クラスタが最も良いパフォー
マンスを見せた。しかし、さらに大きな分子である
Test2 では二電子積分の数が非常に多く、これにつ
いてのディスクアクセスが律速になってしまっている。
そのためノード数が多く相対的にディスクアクセスが
高速になる Pentium3 クラスタが最も良いパフォー
マンスをみせた。

一方、分子軌道の数が少なく CSF の数が多い Test3
ではディスクアクセス速度の影響はほとんどみられず、
ほぼ CPU の演算速度に比例したものとなっている。
これは有効ハミルトニアンを評価するルーチンが非常
に独立性の高いループから構成されており、通信量も
少なく、並列システムの構成にパフォーマンスがほと
んど影響されないことを示している。

以上のように、今回の並列化では積分変換における
ディスクアクセス時間が律速になってしまっているこ
とが判明した。この問題を解決するためには、中間フ
ァイルを作らずそのつど二電子積分の計算を行う積分
変換の「ダイレクト化」が必要であると考えられる。特
により大規模な分子についての分子軌道計算を行う際
にはディスクシステムのサイズの限界も考えられるた
め、この「ダイレクト化」が強く望まれる。二電子積
分の生成ルーチンとの連携も含め、早急に検討する
べき課題である。

4. おわりに

GAMESS の MCQDPT2 ルーチンを並列化し、そ
のベンチマーク計算を行った。特に二電子積分の積分
変換部分については DDI ライブラリで実装されてい
るリモートメモリアccessによる一方向非同期通信
を用いることで、非常に効果的に並列化を実行する
ことを可能にしている。今回、大規模な分子に対する
MCQDPT2 計算のボトルネックとなっている二電子
積分の積分変換部分を効果的に並列化できたことで、
比較的大きな分子の MCQDPT2 計算のベンチマーク
を実行することが可能となった。現時点ではディス
クアクセスが全体のパフォーマンスを低下させている
傾向があるため、SMP タイプのマシンでは十分な性能
を発揮できているとはいえないが、広く使われている
Pentium3 PC クラスタは、ノード数を増加させるこ
とにより非常に高い性能を発揮することが確認できた。

謝辞 本研究は一部、文部科学省平成 13 年度科学
技術振興調整費(総合研究)科学技術計算専用ロジッ
ク組込み型プラットフォーム・アーキテクチャの開発」
によっている。

参考文献

- 1) Schmidt, M.W., Gordon, M.S., Koseki, S., et al.: General Atomic and Molecular Electronic Structure System (GAMESS), *J. Comp. Chem.*, Vol.14, pp.1347-1363 (1993).
- 2) Nakano, H.: Quasidegenerate perturbation theory with multiconfigurational selfconsistent-field reference functions, *J. Chem. Phys.*, Vol.99, pp.7983-7992 (1993).
- 3) Umeda, H., Koseki, S., Nagashima, U. and Schmidt, M.W.: Parallelization of Multireference Perturbation Calculations with GAMESS, *J. Comp. Chem.*, Vol.22, pp.1243-1251 (2001).
- 4) Fletcher, G.D., Schmidt, M.W., Bode, B.M. and Gordon, M.S.: The Distributed Data Interface in GAMESS, *Comp. Phys. Comm.*, Vol.128, pp.190-200 (2000).
- 5) Fletcher, G.D., Schmidt, M.W., Bode, B.M. and Gordon, M.S.: Developments in Parallel Electronic Structure Theory, *Adv. Chem. Phys.*, Vol.110, pp.267-294 (1999).

(平成 13 年 8 月 31 日受付)

(平成 14 年 2 月 13 日採録)



梅田 宏明

1970 年生。1997 年東北大学大学院理学研究科修了。理学博士。富士総合研究所(株)研究員。GAMESS の並列化および化学反応ダイナミクスの量子制御に関する研究に従事。



小関 史朗

1957 年生。1985 年東北大学大学院理学研究科修了。理学博士。2001 年より大阪府立大学教授。計算化学・理論化学。日本化学会、アメリカ化学会、応用物理学会、日本ケイ素光

化学会各正会員。



長嶋 雲兵 (正会員)

1955 年生 . 1983 年北海道大学大学院理学研究科修了 . 理学博士 . 独立行政法人産業技術総合研究所グリッド研究センター . 計算化学 , 情報化学 , 広域分散並列処理 . IEEE , 日

本化学会 , 日本コンピュータ化学会 , 応用数理学会各会員 .



Michael W. Schmidt

1954 年生 . Ph.D. in Physical Chemistry, Iowa State University, 1982. Professor, Tokyo Metropolitan University, 2000 ~ 2001. Associate Scientist, Ames Laboratory,

ISU.
