2008－HPC－116 (22)
2008／8／6

分子軌道計算におけるヘテロクラスタ上での性能評価 —ネットワークおよびタスク分散方式の改良—

早川 潔* 佐々木 徹** 梅田 宏明***, † 長嶋 雲兵***, †

近年，ヘテロクラスタ上で効率よく数値計算することが盛んに研究されている．ヘテロクラスタ上での数値計算で問題となるのが，負荷バランス問題である．負荷分散の方法としては，動的負荷分散方式および静的負荷分散方式が挙げられる．これらの方式の実行効率を上げるため，2つの改善を行った．1つは計算プロセスの割当改善，もう1つはネットワーク改善である．計算プロセスの割当改善では，静的負荷分散方式において，最大で28.9%の大幅な性能改善が得られ，ネットワーク改善では，動的負荷分散方式において約10%の性能向上が得られた．

Evaluations of Molecular Orbital Calculations on a Hetero-Cluster –Improvements of network and task distribution method–

Kiyoshi Hayakawa, *Tohru Sasaki***, *Hiroaki Umeda****, †, and *Umpei Nagashima****, †

The heterogeneous cluster systems allow us to get high performance and low cost using new CPUs and existing CPUs. The purpose of our work is to propose the best load balancing method(including the scheduler) of Molecular Orbital Calculations on the heterogeneous cluster system. In the task scheduler, we have implemented static load balancing and dynamic load balancing methods. This paper shows 2 improvements to bump up parallelization efficiency on the load balancing methods. One is scheduling multiple FMC processes on static load balancing method, the other is improvement of network. As the result, EMDC achieved max 28.9% improvement of it in the multiple FMC processes and also achieved 10% improvement of the parallelization efficiency in the network improvement.

1 はじめに

CPUの低消費電力化にともない，クラスタの低消費電力化およびコンパクト化が重要になりつつある[1]．汎用部品（パソコンのマザーボードやインテルのCPUなど）で構成されたPCクラスタシステムは，そのシステム構築が比較的安価でかつ容易なため，数十～数百台規模のシステムに膨らんできている．また，市販マイクロプロセッサの性能が急激に向上しており，そのプロセッサを使用するPCクラスタシステムは，より高速なシステムのため，小規模な企業や研究所でも導入されている．

クラスタを導入した後の問題点として，故障後の部品調達が難しいことが挙げられる．一般市場におけるCPUのライフサイクルは2～3年であり，故障した時期が遅れるほど，市場で入手できにくくなる．入手できたとしても，性能の高いCPUよりも高価になっている場合が多く，その場合，性能の高いCPUに買い換えたほうが得策である．また，新たにノード台数を増やす場合にも，性能の高いCPUを増設するほうが安価になる場合が多い．

そのようなCPU交換またはノード増設方法を行った場合に問題となるのが，ノードの性能にアンバランスが生じ，ノードの利用効率が悪化することである．この問題に対処するため，様々な負荷分散方式が提案されている[2][3]．ヘテロクラスタにおける負荷分散方法として，大別してHoHeおよびHeHoの2種類に分けられる[2]．HoHeは，ある計算を行う場合，計算プロセスを各ノードに均等に割り当て，計算プロセスが処理するデータ量を各ノードの性能に応じて割り当てる方式である．一方，HeHoは，各計算プロセスに割り当てるデータ量を均一にして，各ノードに割り当てる計算プロセスを各ノードの性能に応じて割り当てる方式である．

*:大阪府立工業高等専門学校 Osaka Prefectural College of Technology
**: (株) アプリオリ・マイクロシステムズ A-Priori Microsystems Ltd
***:産業技術総合研究所 National Institute of Advanced Industrial Science and Technology
†:CREST 科学技術振興機構 CREST Japan Science and Technology Agency

ヘテロクラスタにおける負荷分散方式によるノードの利用効率を調べるため，本研究室では，EMDCというヘテロクラスタのテストベットを開発した．このEMDC上で，分子軌道計算アプリケーションにおける2つの性能改善を行った．1つは，計算プロセスの負荷バランスの改善であり，もう1つは，ネットワークの改善である．

計算プロセスの負荷バランスの改善では，静的負荷分散上でHeHoで行う．計算プロセスに割り当てる計算量は一定のまま，1計算ノードに割り当てる計算プロセスの数を増やすことによって，負荷バランスの均等を計る．

ネットワークの改善では，新たにInter-Chassis Networkを追加し，ネットワークの衝突回避を計る．

2 EMDCシステム

分子軌道計算に用いたEMDCのシステムを図1に示す．本システムは，ヘテロジニアスな実行環境を提供し，その環境下で効率よいアプリケーションを開発するためのテストベットとして開発した．種々の計算資源をクラスタ化し，その上で効率よいアプリケーションを開発することにより，処理速度が劣るノードを混在させても，それらをクラスタ化し，高速に処理できるシステムを目指す．

本システムでは，PentiumM(2.0GHz)搭載のノードが9台，PentiumIII搭載のノード30台，およびホストコンピュータで構成されている．PentiumIII搭載のノードでは，3ノードのうち1ノードが動作周波数600MHzのCPUで残りの2ノードが700MHzのCPUである．PentiumIIIやPentiumMなどの比較的低動作周波数ではあるが低消費電力であるCPUを利用して，低消費電力，且つコンパクトなクラスタを目指している．PentiumMノードには1Gbyte(デュアルチャネル)，PentiumIIIには256Mbyteのメモリが搭載されている．HDDは，コンパクトフラッシュメモリ（PentiumM・PentiumIIIノードともに1Gbyte）を採用している．このことにより，機械稼働部品が減り，より長期運用・低消費電力なシステムになる．

ネットワーク構成は，筐体内ネットワークのIntra-Cahssis Networkおよび筐体外ネットワークのInter-Chassis Networkで構成されている．Intra-Chassis Networkは，Gigabit Ethernetで構築した．PentiumIIIノードのInter-Chassis Networkは，100Base-TXのEthernetで構築し，PentiumMノードのInter-Chassis Networkは，Gigabit Ethernetで構築した．

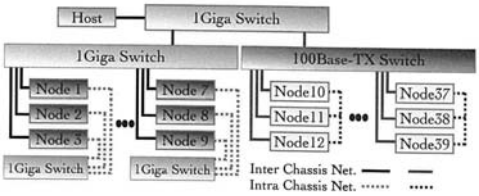

図1: EMDCシステム

2.1 PentiumIIIシャーシのネットワーク構成

PentiumIIIのネットワーク構成を図2に示す．マザーボード内蔵のネットワークインターフェースカード（以下NICと略す）の他に，PCIスロットにNICを挿入する．筐体の真ん中に実装されているノード（Centerノード）は，PCIに2つのNICを搭載し，筐体の左側のノード（Leftノード）および筐体の右側のノード（Rightノード）と直接接続されている．従って，LeftおよびRightノードは，PCIスロットに1つのNICを挿入し，Centerノードと接続されていることになる．

筐体内外の通信は，3種類のネットワークを用いて行われる．Inter-Chassis Networkで1つ使用され，残り2つはIntra-Chassis Networkで使用される．この場合，Leftノードと Rightノードは直接通信できない．Left-Right間通信は，Centerノードを介して行われる．

2.2 PentiumMシャーシのネットワーク構成

PentiumMのネットワーク構成を図3に示す．マザーボード内蔵の2つのNICを利用して，筐体内外のネットワークを構築する．そのため，各筐体に対して，ギガビットのスイッチを用意した．

PentiumMのネットワークの場合，PentiumIIIのそれと異なり，2つのネットワークを利用して筐体内外の通信を行う．

3 分子軌道計算プログラム

EMDCシステムで実行する分子軌道計算プログラムは，EHPCプロジェクト[5]で作成されたプロ

グラムを EMDC 用に移植したものである．分子軌道計算では，Fock 行列生成に多くの時間を費やす．従って，この Fock 行列生成を並列化する．

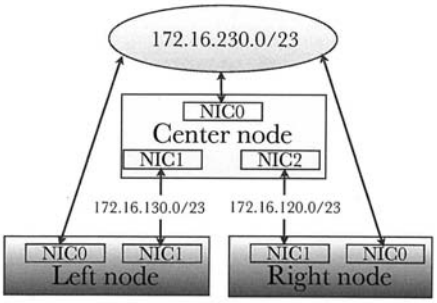

図 2: PentiumIII シャーシのネットワーク構成

図 4 に Fock 行列生成の処理シーケンスを示す．初期化から始まり，Fock 行列を初期化し，Fock 行列生成を行う計算プロセス（以後 FMC プロセスと略す）に分散させる．バリアを行い，全 FMC プロセスの終了後，データを集める．条件に合わなかったら，再度 Fock 行列生成を行う．この Fock 行列生成のサイクルを SCF サイクルと呼ばれる．

4 負荷分散方式

本節では，Intra・Inter Chassis Network を利用した動的負荷分散方式，および静的負荷分散方式での計算プロセス複数実行方式について述べる．

4.1 Intra・Inter Chassis Network を利用した動的負荷分散方式

動的負荷分散方式では，ホストコンピュータが，実行時に，各ノードへ計算タスクを割り当てる．実装した分子軌道計算アプリケーションは，計算コントロールプロセス (図中 Appli_Control)，通信ブリッジプロセス (図中 Communication Bridge) および計算（FMC）プロセス（図中 FMC）の 3 つのプロセスで構成される（図 5 参照）．各プロセスの通信は，H_comm および L_comm と呼ばれる通信ライブラリを介して行われる．ホストコンピュータが計算コントロールプロセスを担当し，Center ノードが通信ブリッジおよび FMC プロセスを担当し，Right および Left ノードが FMC プロセスを担当する．計算コントロールプロセスが，計算タスクを，通信ブリッジを介して，空いている FMC プロセスに割り当てる形で負荷を分散する．この方式を用いれば，早くおわる FMC プロセスに計算タスクが多く割り当てられ，自ずと，HoHe が実現される．

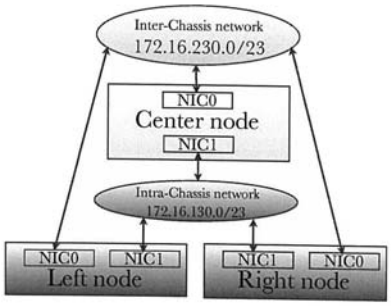

図 3: PentiumM シャーシのネットワーク構成

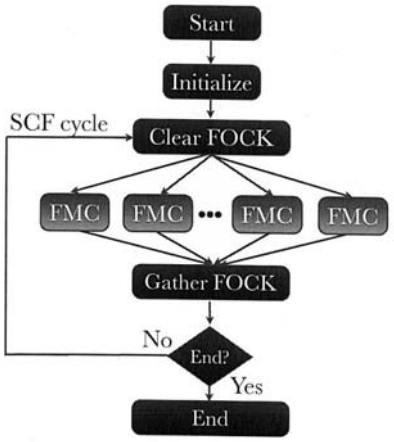

図 4: Fock 行列生成プロセス（FMC プロセス）の処理シーケンス

動的負荷分散方式では，空き FMC プロセスを検索する通信，計算タスクを割り当てるために必要な通信が，頻繁に行われる [4]．そこで，この通信を筐体内と筐体外で分けて行う事により，通信の衝突を回避し，スループットを向上させる．通信コントロールと通信ブリッジとの通信は，Inter-Chassis Network を用いて行い，通信ブリッジと計算ノードとの通信は，Intra-Chassis Network を用いて行う．筐体内のノードにある空き FMC プロセスの検索は，センターノードの通信ブリッジが行っているので，Intra-Chassis Network を使ったほうが効率よく空き FMC プロセスを検出できる．

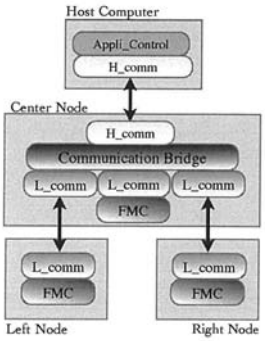

図 5: 分子軌道計算における処理プロセス構成

4.2 静的負荷分散方式での FMC プロセス複数実行

静的負荷分散方式の場合，1) 実行前に計算タスクの量を決め，2) 同じ計算量の計算タスクを各の FMC プロセスに割り当て，3) 計算結果を計算コントロールプロセスに集め，4) 次の計算タスクを新たに割り当てる，という処理を繰り返す．

この場合，1 ノードあたり 1 FMC プロセスを割り当てると，性能がよい PetiumM ノードの FMC プロセスが早く終わり，待ち時間が生じる．そこで，PentiumM ノードに複数の FMC プロセスを割り当てることにより，待ち時間が少なくなるように調整する．今回の実装では，PentiumM ノードに計算ノードを追加する際に，計算ノードと計算コントロールとの通信を行うために，通信ブリッジも追加した．

5 性能評価

性能評価として，負荷バランス改善およびネットワーク改善について評価する．

5.1 評価環境

分子軌道計算の性能評価として，C 端末を OH キャップしたグリシンの 5 量体 $(Glycine)_5$ の HF/6-31G(d,p) (ガウス型基底の数は 400) を計算した．文献 [5] に示されているスクリーニングを行い，動的負荷分散方式で計算した場合と静的負荷分散方式で計算した場合を比較した．分子軌道プログラムには GAMESS[6] を Appli_Control 処理部として使用した．

5.2 負荷バランス改善に関する評価

PentiumM ノードで複数の FMC プロセスを並列実行させた場合の並列化効率を図 6 および図 7 に示す．図 6 は，分子軌道計算の中の Fock 行列生成に費やした時間（つまり，各 FMC プロセスの SCF をさせてから Barrier が終わるまで）である．一方，図 7 は，Fock 行列生成を行うための準備や Fock 行列生成処理の後処理など，並列処理していない部分も含めた全体の時間である．

図 6 において，30 台，33 台において，4 プロセス実行の時に，並列化効率のピークパフォーマンスが得られ，36 台では，3 プロセス実行の時に，ピークパフォーマンスが得られた．PentiumM は PentiumIII に比べて，静的負荷分散方式では，2.83 倍速い．単純に考えれば,FMC プロセスを 3 に増やしたときに，並列化効率は低下するはずである．PentiumM ノードの FMC プロセス数を $n(n \geq 2)$ としたときの PentiumIII の計算ノードの実行時間を $T_{III}(n)$ とし，PentiumM の計算ノードの実行時間を $T_m(n)$ とする．PentiumIII ノード i 台，PentiumM ノード j 台の場合，$T_{III}(1)$ が既知のとき，PentiumIII の FMC プロセスの実行時間が

$$T_{III}(n) = \frac{(i+j)T_{III}(1)}{i+nj} \tag{1}$$

と見積もることができる．一方，PentiumM ノードの FMC プロセスは，単純計算で

$$T_M(n) = \frac{n(i+j)T_M(1)}{i+nj} \tag{2}$$

と見積もることができる．仮に $T_M(1) = 2.83T_{III}(1)$ とすると，(2) 式は (3) 式になる．

$$T_M(n) = \frac{n(i+j)T_{III}(1)}{2.83(i+nj)} \tag{3}$$

この式を適用すると，2 プロセスと 3 プロセスでほぼ同じ並列化効率を示し，4 プロセスでは，低下することになる．

しかし，FMC プロセスを 4 にしても，並列化効率が低下しなかった．これは，PentiumM の 1 サイクルあたりの発行命令数にあると考えられる．PentiumM は 1 サイクルあたり最大 3 命令発行可能である．1 プロセスでは，命令発行スロットに空きがあり，最大命令発行ができず，FMC プロセスを増やすことによって，命令発行スロットが埋まり，最大発行が可能となり，FMC プロセスを 2 倍に増やしても，計算時間は，2 倍より小さい数値になったのではないかと推測される．

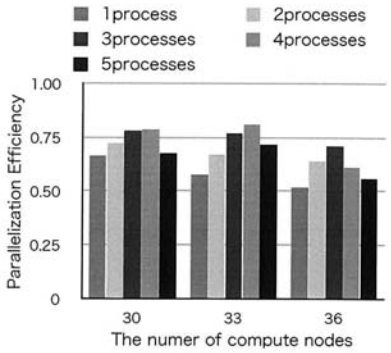

図 6: FMC プロセス数と並列化効率の関係（Fock 行列生成処理時間のみ）

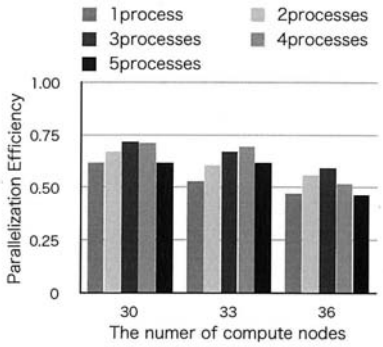

図 7: FMC プロセス数と並列化効率の関係（全体の処理時間）

36 台の場合は，3FMC プロセスまでは，並列化効率が向上していったが，4FMC プロセスのときに，並列化効率が低下した．この原因としては，計算タスクの不均一さが原因である．今回の実装では，FMC プロセスの処理として，分子軌道に影響を与えない計算は省いている．したがって，厳密には，同じ計算量のタスクでも計算時間に多少の差が生じる．FMC プロセスが少ない場合，計算タスクが大きいので，計算時間の差は現れにくいが，FMC プロセスが増加するにつれて，計算時間の差が現れて，並列化効率の低下をもたらしたと考えられる．

図 7 において，30 台および 36 台の時に，4FMC プロセスを実行した場合，並列化効率の低下がみられた．FMC プロセスを増加させることにより，計算タスク分割やその転送などの時間がかかるため，トータルな計算時間では，30 台でも微小であるが，並列化効率の低下がみられ，36 台では，Fock 行列生成よりも急激に並列化効率が低下した．

5.3 ネットワーク改善に関する評価

Intra-Chassis network のみを使用した場合（以後，1net と略す）と Intra および Inter-Chassis Network の両ネットワークを使用した場合（以後，2net と略す）の並列化効率を比較した．静的負荷分散方式における 1net と 2net の比較を図 8 に示し，動的負荷分散方式のそれを図 9 に示す．図において，1 から 27[1] ノードまでは，pentiumIII ノードのみでクラスタを構成し，30 ノード以降は，PentiumIII および PentiumM ノードで（ヘテロ）クラスタを構成した．なお，30 ノード以降の並列化効率は，PentiumM ノードの性能を PentiumIII ノードの性能に変換して計算した（文献 [4] において，PentiuM I ノードを静的負荷分散方式で 2.83 ノード，動的負荷分散方式で 3.05 ノードと換算して並列化効率を算出した）．

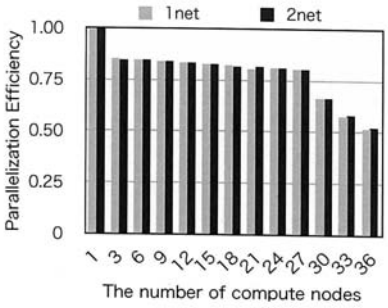

図 8: ネットワーク改善に伴う並列化効率の比較 (静的負荷分散方式)

図 8 からわかるように，静的負荷分散方式では，1net と 2net とではほとんど変化が見られなかった．これは，静的負荷分散方式では，ほとんどの時間を計算時間に費やし，通信はほとんどしていないからである．今回計算した分子では，36 ノードの場合，少なくとも 150 秒程度の計算と 1 秒も満たない通信を 13 回繰り返しただけだったので，2net 化にしたメリットが活かしきれなかった．

動的負荷分散方式の場合，計算中，頻繁に通信を行うので，6 計算ノードあたりから 2net のほうが有利になった．また，1 から 27 台までの PentiumIII の時より，30 台から 36 台までの PentiumM が加わっ

[1] 3 ノードの PentiumIII ノードが故障中のため，PentiumIII ノードは最大 27 ノード

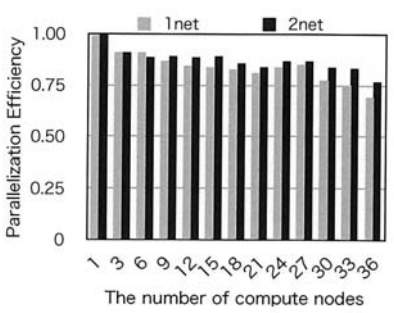

図 9: ネットワーク改善に伴う並列化効率の比較 (動的負荷分散方式)

てからのほうが，2net による並列化効率の向上が顕著になった．これは，PentimM の場合，NIC を内蔵しているということおよび 2net 化による衝突回避が効いたためである．PentiumM の処理が早いため，PentiumIII より多くの計算タスクが割当られた．それらの通信の衝突回避が 1net に比べて多くなった．

6 おわりに

分子軌道計算において，静的負荷分散方式における負荷バランスの改善およびネットワークの改善を行った．これらの改善を評価するため，ヘテロクラスタのテストベッドである EMDC を利用した．EMDC の PentiumIII27 台，PentiumM9 台を利用して，並列化効率を測定した．

性能評価では，静的負荷分散方式において，PentiumM ノードの FMC プロセス数を増加させて，並列化効率を測定した．その結果，36 台で PentiumM の FMC プロセスを 3 つに増やすことにより，並列化効率が，1FMC プロセスより，28.9%向上した．このときの実行時間は 1409.9 [sec] で，動的負荷分散方式を 36 台で実行したときの実行時間 (1368.4[sec]) には達しなかった．また，Inter-Chassis Network のみで分子軌道計算をさせた場合と，Intra-Chassis および Inter-Chassis Network を利用して分子軌道計算をさせた場合とを比較した．Intra-Chassis および Inter-Chassis Network を利用したネットワークは，Inter-Chassis Network のみと比べて，最大で約 10%の性能改善が得られた．

今後の課題としては，HoHe 方式と HeHo 方式をハイブリッドした負荷分散方式を検討する．また，そのハイブリッド方式に合ったネットワークフレームワームも検討する予定である．

謝辞

この研究の一部は，平成 19 年度科学研究補助金 (基盤研究 C:課題番号 18500044)「低消費電力コンパクトクラスタの研究」によって行われた．

参考文献

[1] 中島 浩, 中村 宏, 佐藤 三久, 朴 泰祐, 松岡 聡, 高橋 大介, 堀田 義彦, "高性能計算のための低電力・高密度クラスタ MegaProto", 情報処理学会論文誌コンピューティングシステム, Vol.46, No. SIG12 (ACS 11), pp. 46–61(2005).

[2] Kalinov,A.andKlimov,S.:Optimal mapping of a parallel application processes onto heterogeneous platform. Proc. 19th IEEE International Parallel and Distributed Processing Symposium(IPDPS2005), IEEEComputer Society, CD-ROM(2005).

[3] 高橋 翔, 市川 周一, "不均一クラスタの最適構成予測モデルの各応用への適用と評価", 報処理学会研究報告, 2006-ARC-167, pp. 97-102(2006).

[4] K.Hayakawa, T.Sasaki, H.Umeda, and U.Nagashima "Molecular Orbital Calculations on Embedded Middle Density Cluster System",Int'l Conf. on Parallel and Distributed Computing and Systems(PDCS2007), pp.1-6(2007).

[5] 梅田 宏明, 稲富 雄一, 本田 宏明, 長嶋 雲兵：分子軌道計算専用計算機のためのフォック行列並列計算アルゴリズムの開発, 日本コンピュータ化学会, Vol. 4 No. 4 ,pp. 179–187(2005).

[6] Schmidt, M., Baldridge, K., Boatz, J., Elbert, S., Gordon, M., Jensen, J., Koseki, S., Matsunaga, N., Nguyen, A., Su, S., Windus, T., Dupuis, M., Montgomery, J. :General atomic and molecular electronic structure system. Journal of Computational Chemistry **Vol. 14, Issue 11** , pp.1347-1363 (1993) .

[7] 須田礼仁 "ヘテロ並列計算環境における性能評価", 報処理学会研究報告, 2006-HPC-115, pp. 25-30(2008).