

レプリカ交換分子動力学シミュレータ REMD Toolkit のグリッド上での実行

佐藤 仁 † 伊藤正勝 ††
中田秀基 †† 松岡 聡 †,†††

レプリカ交換法は、問題の粒度が大きく、計算機間のデータの通信量が少ないため、グリッド上での計算に向いていると考えられている。我々は、今まで実際にレプリカ交換法のアルゴリズムを実装した REMD Toolkit をグリッド上で実行し、その性能について検証を行ってきたが、REMD Toolkit は、全ワークカに対するバリア同期を含むため、潜在的に性能ヘテロなグリッド上においては著しく実行時間が低下する問題がある。そこで、本稿では REMD Toolkit に性能ヘテロな環境に対応するための改良を行い、その有効性を確認した。

Execution of the replica exchange molecular dynamics simulator on the Grid

HITOSHI SATO †, MASAKATSU ITO ††, HIDEMOTO NAKADA ††
and SATOSHI MATSUOKA †,†††

Replica-exchange method is considered to be suitable for execution on the Grid environment because of its large granularity and small size data transfer. To confirm the suitability, we have performed several experiments on various environment, using an application program called REMD Toolkit that implements Replica-exchange method. But, REMD Toolkit has a problem to which the performance falls remarkably on the Grid environment, because it includes the barrier synchronization to all workers. We improve the REMD Toolkit to cope with performance-heterogeneous environment. The results showed that the improved version is faster than original version in the performance-heterogeneous environment.

1. はじめに

複数の管理主体に属する計算機資源を集散的に活用して大規模な計算を行うグリッドと呼ばれる計算環境が現実的になりつつある。このようなグリッド上のアプリケーションとして、産業技術総合研究所（産総研）グリッド研究センターでは、REMD (Replica Exchange Molecular Dynamics) Toolkit⁽⁸⁾⁷⁾ が開発されている。REMD Toolkit はタンパク質の立体構造に基づく薬物分子設計を目的としたアプリケーションであり、Simulated Annealing に似たレプリカ交換法⁽³⁾⁵⁾ と呼ばれるアルゴリズムを用いている。一般に、レプリカ交換法による分子シミュレーションは、問題の粒度が大きく、計算機間の情報通信量が少なくできるので、並列計算に向いていると考えられている。ま

た、REMD Toolkit がシミュレーションの対象とする分子のサイズは非常に大きく、莫大な計算時間が必要であるため、求解時間の短縮が切望されている。このため、REMD Toolkit はグリッド上での実行に適していると考えられる。

しかしながら、REMD Toolkit では、全ワークカに対するバリア同期を多数含むため、潜在的に性能ヘテロなグリッド上においては低速な PE に律速されるため、著しく実行時間が低下する問題がある。そこで、本稿では REMD Toolkit に性能ヘテロな環境に対応するための改良を行い、その有効性を確認した。

2. レプリカ交換法と REMD Toolkit

2.1 レプリカ交換法

タンパク質の分子などの多自由度の複雑系では、エネルギー極小状態が無数に存在するため、分子シミュレーションを実行しても、構造がエネルギー極小状態に留まってしまい、信頼できる正確な物理量が得られないという問題がある。⁽⁶⁾ この問題を解決する手法と

† 東京工業大学 Tokyo Institute of Technology

†† 産業技術総合研究所 National Institute of Advanced Industrial Science and Technology

††† 国立情報学研究所 National Institute of Informatics

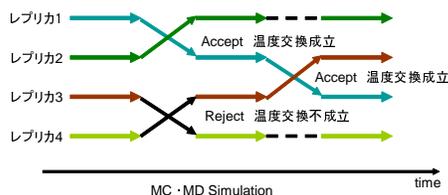


図 1 レプリカ交換法

して、レプリカ交換法というアルゴリズムが知られている。レプリカ交換法による分子シミュレーションでは、分子の構造の複数のレプリカを用意し、それぞれに異なる温度を割り振り、構造サンプリングを行う。サンプリングのためのシミュレーションには、分子動力学法 (MD) あるいはモンテカルロ法 (MC)⁹⁾ を用いるが、分子レプリカの間には相互作用がないので、サンプリングステップは独立かつ並列に計算することができる。低温でサンプル構造が準安定状態に囚われるという MC/MD の欠点は、レプリカ間で温度交換を行うことで解消されるので、この手法では信頼性の高い物理量が得られる。

2.2 REMD Toolkit

REMD Toolkit は、産総研グリッド研究センターで開発されているタンパク質の立体構造に基づく薬物分子設計を目的としたアプリケーションである。

薬物分子の設計において、タンパク質分子の 3 次元構造がどのようにになっているか、また、タンパク質分子がどのように空間中を揺らぐのかというような情報が非常に有用であるとされる。タンパク質の構造に関する情報は、ターゲットとなるタンパク質に特異的に結合するリガンドと呼ばれるタンパク質を予測する際に必要とされるのに対し、タンパク質構造の揺らぎに関する情報は、ターゲットとなるタンパク質中の問題となっている部分ヘリガンドをドッキングさせる際に必要な情報となる。

ドッキングの際の結合親和力を正確に評価するためには、リガンドとなるタンパク質構造の揺らぎの複雑さがその構造の部分状態の統計集団として表現されるように統計力学を厳密に適用させることが必要とされる。また、このような統計力学を用いた評価は、タンパク質の様々な構造の変化と関連するタンパク質の機能を理解するために本質的でもある。

しかしながら、2.1 節で述べたように、タンパク質は、エネルギー極小状態が無数に存在するため、従来のモンテカルロ法や分子動力学法による分子シミュレーションでは、それらのエネルギー極小状態に留まってしま

うため、信頼できる定量的な予測を行うことが絶望的に難しいため、このようなシミュレーションを行う際は、レプリカ交換法を用いることが必要とされる。

REMD Toolkit は、レプリカ交換法を用いてタンパク質構造を効率よくサンプリングを行う。また、オブジェクト指向言語である C++ により記述されているため、プログラムのモジュールを変更することにより、様々な統計集団の分子シミュレーションに対応することができ、また、様々な並列化を行うことができる。現状では、REMD Toolkit の並列化は MPI を用いている。

2.3 REMD Toolkit の並列化手法 (割り当て固定法)

REMD Toolkit は、マスタとワーカ以外に通信を行わないという点において純粋なマスタ・ワーカ方式である。図 2 のようにワーカがそれぞれにひとつの分子の構造座標 (レプリカ) の情報を管理しているのに対し、マスタはすべてのレプリカのエネルギー分布と温度分布を管理している。

ワーカはそれぞれ独立に分子動力学法やモンテカルロ法による構造サンプリングを行うが、定まった間隔においてマスタにエネルギー、反応座標 ($double \times 2$ の情報量) をバッファして送信する。反応座標とは、シミュレーションによりどの程度構造のサンプリングが進んだかを表す指標である。

マスタはいったん全てのワーカから情報を受け取り、それらの情報を集計し、このうちのエネルギーをもとにレプリカの温度を決め直し (交換し)、新しい温度をワーカに送信する。この際のマスタからワーカへの通信量は $double \times 3 + bool$ ($bool$ は温度更新のフラグ) となる。温度の他にも 2 つの熱力学的パラメータがワーカに送信されるので、REMD Toolkit では、通常のカノニカル統計集団の他にも一般化された統計集団を扱うことができる。

温度交換が発生した場合 (温度更新のフラグが $true$ の場合) は、マスタから送られてきた温度情報を元に、ワーカの温度の情報の再設定を行う。温度交換が発生しない場合 (温度更新のフラグが $false$ の場合) は、何の処理も行わず、そのまま次のシミュレーションを行う。この手法を以下、割り当て固定法と呼ぶ。

2.4 割り当て固定法の問題点

2.3 節で述べた割り当て固定法では、温度情報を交換する周期ごとにすべてのワーカが同期する。この構造ではワーカ間に性能差があった場合、最も進捗の遅いワーカがボトルネックとなり、進捗の速いワーカは遅いワーカの終了を待つことになる。このため、本質的に性能ヘテロなグリッド環境では有効に運用することが難しいことが予想される。

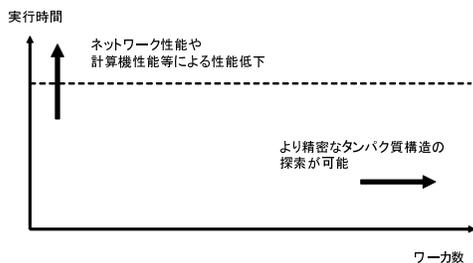


図 5 割り当て交換法のワーカ数と実行時間の関係

4. 実験

4.1 実験の概要

割り当て交換法を用いて実装した REMD Toolkit の性能を検証するために、下記の 3 つの環境で実験を行った。

- (1) 性能ホモな環境
- (2) 性能ヘテロな環境
- (3) 広域ネットワーク環境

実験に用いたタンパク質は、オリゴペプチド (Ala)₁₀ (原子数 66) であり、温度は、下限を $T_L = 168.2K$ 、上限を $T_H = 2262.7K$ に固定し、指数関数的にワーカ数に応じて変化させた。例えば、 i 番目のワーカの担当する温度は以下ようになる。ここで、 N は、ワーカ数である。

$$T_i = T_L \times \left(\frac{T_H}{T_L} \right)^{\frac{i}{N-1}}$$

各実験は、1 回の温度交換を行うまでの区間の分子シミュレーションを 200 ステップ、温度交換の回数を 1000 回と固定して行った。従って、図 5 のようにワーカ数の増加の割合に対し、実行時間が一定に保たれていれば、スケーラブルであると判断した。

また、割り当て交換法では、以下のようなパラメタを各ワーカの進捗状況の把握の際や座標交換の決定の際に用いている。

- 交換決定パラメタ
- 交換非対象パラメタ
- マスタ報告パラメタ

交換決定パラメタは、ワーカ間の割り当て交換の決定の際に用いられ、マスタ報告パラメタは、1 回の温度交換シミュレーション区間の中でのワーカがマスタへ進捗状況を報告する回数を定める。各ワーカの進捗状況は以下のように定める。ここで、 $progress$ は、各ワーカの進捗状況を表す値、進捗状況を $eTerm$ は、温度交換シミュレーション交換を実行した回数、 $rTerm$ は、1 回の温度交換シミュレーション区間においてワーカが

マスタへ行った報告の回数、 $nRTerms$ は、マスタ報告パラメタ、すなわち、1 回の温度交換シミュレーション区間でワーカがマスタへ報告する総回数である。

$$progress = eTerm + \frac{rTerm}{nRTerms}$$

交換非対象パラメタは、極端に進捗が遅れたワーカが存在するような場合にそのワーカに対して連続して割り当て交換が発生することを避けるため、割り当て交換の制約を加えるパラメタである。上述の進捗状況を表す値を元に、交換非対象パラメタで設定した値の間は、割り当て交換が発生しないようにする。

マスタ報告パラメタが n の場合は、1 回の温度交換シミュレーション区間でワーカは n 回マスタと通信を行うことになるので、全体のシミュレーションでは、ワーカは、温度交換数 $\times n$ 回マスタと通信を行うことになる。

4.2 実験環境

本実験で用いる実験環境は、東京工業大学松岡研究室 PRESTO III クラスタ、産業技術総合研究所 UME クラスタ、東京工業大学学術国際情報センターの GSIC-PRESTO クラスタである。表 1 に各計算機の仕様を示す。ただし、PRESTO III 以外のクラスタは、占有可能ではないため、他のユーザも頻繁に利用しているような状況下における実行結果である。(1) 性能ホモ環境、(2) 性能ヘテロ環境での実験には、gcc3.1.1、MPI は mpich 1.2.5 を用い、(3) Globus を用いたグリッド環境での実験には、gcc3.1.1、MPI は MPICH-G2 1.2.5 を用いている。

4.3 性能ホモな環境での実験

まずはじめに、割り当て固定法、割り当て交換法で実装されたプログラムの挙動を確認するために、東京工業大学松岡研究室 PRESTO III クラスタで次のような実験を行った。割り当て交換法、割り当て固定法の両プログラムをワーカ数を変化させて実行した。割り当て交換法のパラメタは、マスタ報告パラメタを 5、交換決定パラメタを 0.3、交換非対象パラメタを 3 とした。実行結果の結果を図 6 に示す。割り当て固定法では、ワーカ数が増加するにつれ実行時間が増加しているが増加の割合は低く抑えられているのに対し、割り当て交換法では、ワーカ数が増加に応じて実行時間の大幅な増加が観測された。これは、ワーカがマスタへ進捗状況の報告を行う際に発生する通信がボトルネックになるためであると考えられる。実際、図 7 に示されたワーカ数とマスタ、ワーカ間の MPI Recv にかかる時間の関係を見ると、割り当て固定法においては、ワーカ数の増加に対し MPI Recv にかかる時間が低く抑えられているのに対し、割り当て交換法では、ワーカの台数の増加に応じて MPI Recv にかかる時間が増加することが観測された。また、図 6 と図 7 より、MPI Recv にかかる時間の増加が直接全体の実行時間に影響していることが確認できた。以上から、割り当て交

表 1 使用計算機システムの仕様

名称	PRESTO III	UME	GSIC-PRESTO
設置場所	東工大大岡山キャンパス	産総研つくば	東工大大岡山キャンパス
PE	Athlon MP 1900+ × 2	Pentium III 1.4GHz × 2	Pentium III 800MHz
Mem	768MB	2.4GB	512MB
ノード数	256	32	1
CPU 数	512	64	1
NIC	100BASE-T	1000BASE-T	100BASE-TX
OS	Linux 2.4.19	Linux 2.4.20	Linux 2.4.20

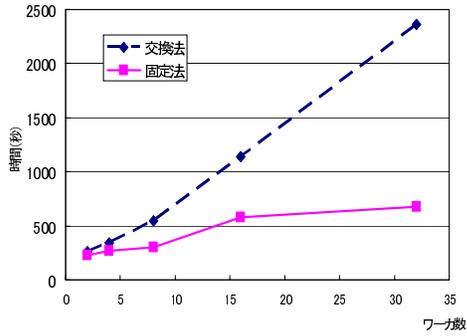


図 6 性能ホモ環境での実行時間の比較

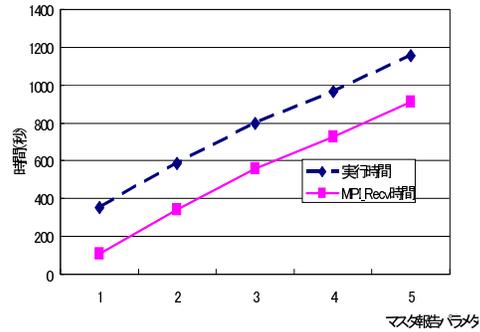


図 8 割り当て交換法のマスタ報告パラメタと実行時間, 及び, マスタ, ワーカー間の通信時の MPI Recv 時間

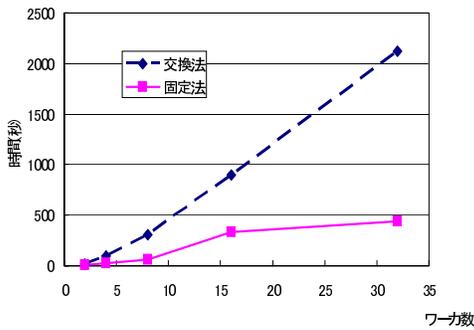


図 7 割り当て固定法, 割り当て交換法でのワーカー数とマスタ, ワーカー間の通信時の MPI Recv 時間

換法では, ワーカーのマスタへの進捗状況の報告のための通信が全体の実行時間に大きく影響を与えることが伺える。

次に, 割り当て交換法においてワーカーがマスタへ行く進捗状況の報告の際の通信が報告数に応じてどのように全体の実行時間に影響するのかを確認するために, マスタ報告パラメタを変化させて割り当て交換法を実行した。図 8 にマスタ報告パラメタと実行時間, 及び, マスタ, ワーカー間の通信時の MPI Recv 時間の関係を示す。マスタ報告パラメタの増加に応じて, 実行時間が線形に増加していくことを確認した。また, この原因は, MPI Recv に費やされた時間が直接影響しているためであると考えられる。

また, 割り当て交換法においてワーカー間の割り当て交換の際の通信がどのように全体の実行時間に影響するのかを確認するために交換決定パラメタ, 交換非対象パラメタを変化させて実験を行った。図 9 は, 交換決定パラメタと割り当て交換発生数の関係を示した図であり, 図 10 は, 交換決定パラメタと割り当て交換の際の MPI Recv に費やされた時間の関係を示した図である。図 9 より, 交換決定パラメタの値を小さく設定した場合は, 頻りにワーカー間で割り当て交換が発生するのに対し, 交換決定パラメタの値を大きく設定した場合は, ワーカー間の割り当て交換の発生が低く抑えられていることを確認した。また, 図 10 より, 割り当て交換が多数発生した場合は, MPI Recv に費やされる時間が増加し, 割り当て交換の発生が少数である場合は, MPI Recv に費やされる時間が減少することを確認した。しかしながら, 図 11 から, 交換決定パラメタの値をどのように変化させていても, 実行時間に影響しないことが伺える。これは, 図 10, 図 11 よりワーカー間のレプリカ交換が全体の実行時間と比較して極めて小さいためであると考えられる。

以上の実験から, 性能ホモな環境においては, 割り当て固定法は, ワーカー数の増加に対して, 実行時間の増加が抑えられている。これに対し, 割り当て交換法は, ワーカーのマスタ報告パラメタを増加させた場合, 報告数の増加に応じて, 実行時間の増加が観測された。この性能低下の原因は, ワーカーのマスタへの報告数の増加によりマスタ, ワーカー間の通信数が増加したため

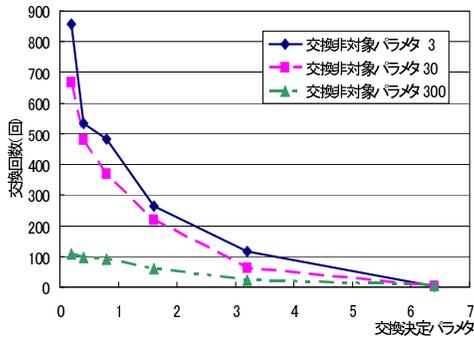


図 9 割り当て交換法の座標交換発生数と交換決定パラメタの関係

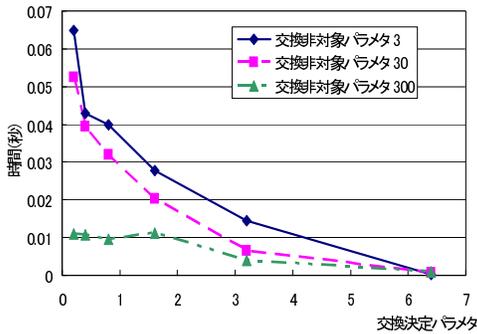


図 10 割り当て交換の際の MPI Recv 時間と交換決定パラメタの関係

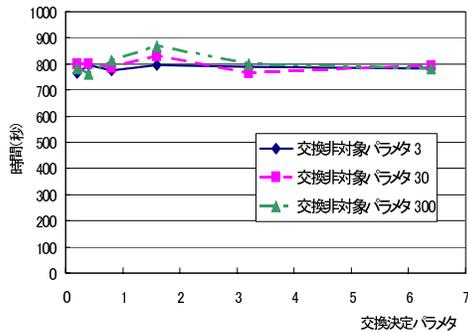


図 11 割り当て交換法の実行時間と交換決定パラメタの関係

あると考えられる。一方で、全体の実行時間へのワーカ間の割り当て交換の影響はほとんどないことを確認した。

4.4 性能ヘテロである環境での実験

性能ヘテロである環境での REMD Toolkit の実行を評価するために、東京工業大学松岡研究室 PRESTO III クラスタで次のような実験を行った。性能ヘテロな環境を実現するために、一部の計算機に複数のワー

表 2 性能ヘテロ環境での負荷

	高速ワーカ数	低速ワーカ数	低速ワーカ相対速度
負荷なし	16	0	1
負荷 1	12	4	1/2
負荷 2	10	6	1/3
負荷 3	8	8	1/4

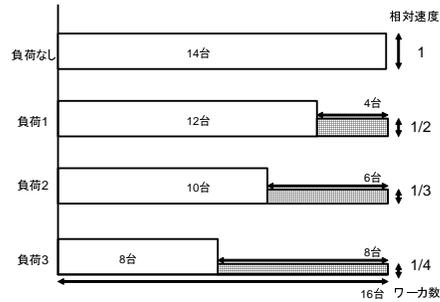


図 12 性能ヘテロ環境での負荷

カを割り当て擬似的に計算機性能が低下したのと同様な環境を設定した。この実験では、ワーカ数は、16 に固定した。また、交換決定パラメタは 0.3、交換非対称パラメタは 3 に設定し、マスタ報告パラメタは、1 から 5 まで変化させて実行した。負荷のかけ方は表 2 及び 12 に示す 4 通りを用意した。

結果を図 13 に示す。負荷が増加するにつれ、割り当て固定法ではもっとも低速なプロセッサの性能に応じて性能が低下している。これに対して、割り当て交換法では、マスタ報告パラメタが小さい場合は、割り当て交換法と同様に、実行時間が増加するのに対し、マスタ報告パラメタが大きい場合は、それほど大きな影響を受けていないことが伺える。また、割り当て交換法でマスタ報告パラメタが 1 の場合は、割り当て固定法よりも高速に動作した。これは、割り当て交換法ではバリア同期を取っているのに対し、割り当て交換法では、バリア同期が解消されているため、高速に動作したと考えられる。また、図 14 に示す割り当て交換法のマスタ、ワーカ間の通信時の MPI Recv 時間をみると、割り当て固定法では、線形に増加するのに対し、割り当て交換法では、MPI Recv 時間の増加の割合が抑えられているのが伺える。

次に、割り当て交換法での負荷と割り当て交換発生数の関係をマスタ報告パラメタ別に表した図を図 15 に示す。マスタ報告パラメタ 1 の場合は、割り当て交換発生数が 0 であるため、図 15 から除いてある。負荷が増加するにつれ、交換回数が増加することが伺える。また、マスタ報告パラメタが大きい場合は、負荷が増加するにつれて、割り当て交換発生回数の増加の割合が大きく、逆に、マスタ報告パラメタが小さい場合は、負荷が増加した場合でも、割り当て交換発生回数の増加

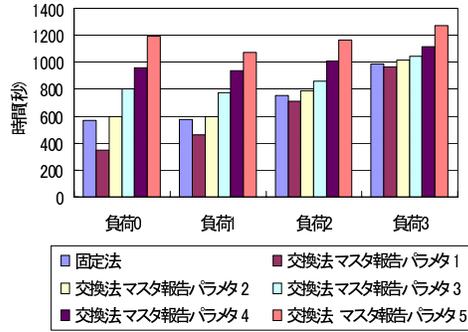


図 13 性能ヘテロ環境での実行時間の比較

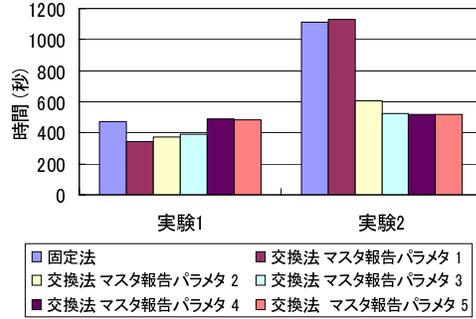


図 16 広域ネットワーク環境での実行時間の比較

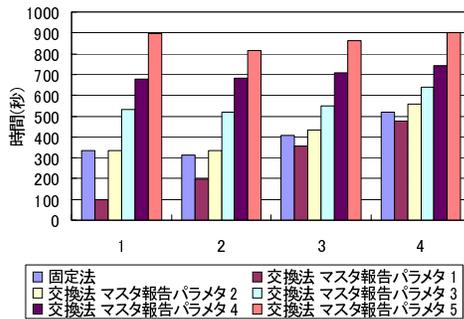


図 14 性能ヘテロ環境での割り当て交換法のマスタ, ワーカー間の通信時の MPI Recv 時間

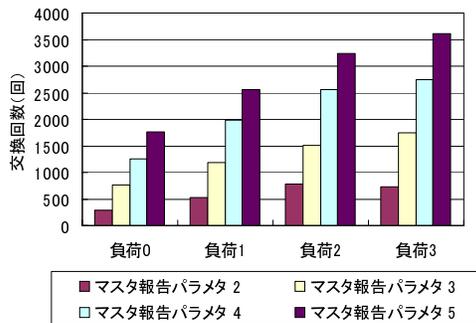


図 15 性能ヘテロ環境での割り当て交換の発生回数

の割合が低く抑えられていることを確認した。

4.5 広域ネットワーク環境での実験

地理的に分散したサイト間での REMD Toolkit の性能を検証するために, MPICH-G2⁽²⁾⁽⁴⁾ を用いて, 次の実験を行った。

- (1) マスタに GSIC-PRESTO を 1 台割り当て, ワーカーを UME に割り当て実行 (性能ホモ環境)
- (2) マスタに UME を 1 台割り当て, ワーカーに GSIC-

PRESTO1 台 (低速ワーカー) と UME15 台 (高速ワーカー) を割り当てて実行 (性能ヘテロ環境) MPICH-G2 は, グリッドの標準ミドルウェアである Globus¹⁾ のサービスを使用した MPI1.1 仕様の MPI である。GSIC-PRESTO, UME 間のスループットは 9.81Mbyte/s, RTT は 4.1ms 程度である。この実験では, ワーカーの数を 16 に固定し, 交換決定パラメタを 0.3, 交換非対象パラメタを 3 とし, マスタ報告パラメタを変化させて実験を行った。結果を, 図 16 に示す。図 16 の実験 1 は, 上述の (1) の性能ホモ環境に対応し, 実験 2 は, (2) の性能ヘテロ環境に対応する。

(1) の実験 (図 16 中実験 1) では, 実行時間に関して割り当て交換法の方が若干良好な結果を示すのが確認できる。この実験では, 各ワーカーの性能は等しいので, 前述の 4.3 節と同様にマスタ報告パラメタを増加させると実行時間が低下するのがみられるが, 実行時間の低下の割合が低く抑えられている。

(2) の実験 (図 16 中実験 2) では, 割り当て固定法と割り当て交換法のマスタ報告パラメタ 1 の場合に実行時間の低下が確認できる。これは, 低速なワーカーが存在するため, バリア同期の際に律速されるためであると考えられる。しかしながら, 割り当て交換法のマスタ報告パラメタ 2, 3, 4, 5 の場合は, 割り当て固定法よりも高速に動作することから割り当て交換法自体は有効であることが伺える。

また, 割り当て固定法と割り当て交換法でのマスタとワーカー間の通信の際にかかる MPI Recv 時間の関係 (17) をみると, 両実験において割り当て交換法のほうが割り当て固定法よりも良好な結果を示している。

4.6 議論

性能ホモ環境の実験から, 割り当て固定法は, ワーカー数の増加に対して, 実行時間の低下が抑えられているためスケーラブルであると考えられる。一方, 割り当て交換法では, ワーカーの台数の増加に応じて実行時間の低下することを確認した。この原因は, ワーカーがマスタへ行うシミュレーション進捗状況報告の通信にかかる時間が増加するためであると考えられる。この MPI

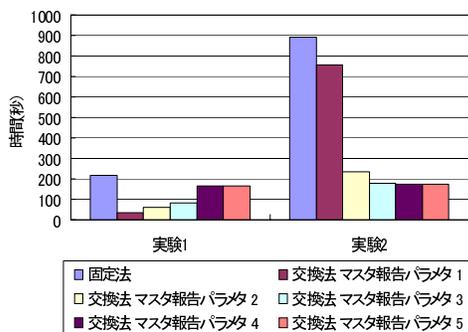


図 17 広域ネットワーク環境でのマスタとワーカ間の通信の際にかかる MPI Recv 時間の比較

Recv にかかる時間の増加は、マスタ、ワーカ間の通信量の問題の他に、MPI の実装の問題で通信がバッファリングされている可能性も考えられるので、今後、検証を行う予定である。また、ワーカの割り当て交換の際の通信にかかる時間は、ワーカがマスタへ行うシミュレーション進捗状況報告の通信にかかる時間と比較して、非常に小さいため、割り当て交換が大量に発生した場合でも、全体の実行時間には影響しないことを確認した。しかしながら、今回のシミュレーションに用いたタンパク質は、原子数が 66 と小さいサイズのものであるため、原子数の大きいタンパク質に対して割り当て交換法を実行した場合に、割り当て交換の際のワーカ間の通信が増大することが考えられるので、この点について検証する必要がある。

性能ヘテロ環境の実験からは、割り当て固定法は、負荷を増やすにつれ、低速なワーカに律速され、実行時間が線形に低下していくのに対し、割り当て交換法は、マスタ報告パラメタを大きくした場合に、実行時間の増加の割合が低く抑えられていることが確認できた。

また、広域ネットワーク環境での性能ホモ、性能ヘテロ環境での実験では、割り当て交換法が、割り当て固定法と比較して、良好に動作することを確認できた。これは、マスタとワーカ間の通信の際の MPI Recv 時間減少するためであると考えられる。

以上の点から、割り当て交換法自体は有効であることが伺える。しかしながら、具体的にどのような条件で割り当て交換法が有効に動作するのかは、明らかではない。この点について、今後検討を行い、交換決定のアルゴリズム、および、パラメタに関して改善を行っていく必要がある。

5. おわりに

本稿では、産総研で開発されているレプリカ交換法のアルゴリズムを実装した REMD Toolkit に対し、性能ヘテロな環境に対応させる割り当て交換法の提案、

及び、その実装を行い、以下の環境で実験評価を行った。

- 性能ホモ環境
- 性能ヘテロ環境
- 広域ネットワーク環境

実験の結果、割り当て交換法は、いくつか環境では性能が逆に低下したものの、性能ヘテロな環境では効果があることを確認した。今後の課題としては以下が挙げられる。

- 割り当て交換法のアルゴリズムおよびパラメータの改良
今回の実験では、具体的にどのような条件で割り当て交換法が有効に動作するのかは、明らかではない。この点について、今後検討を行い、交換決定のアルゴリズム、および、パラメタに関して改善を行っていく必要がある。
- 高レイテンシ環境での性能測定
本稿で用いたグリッド環境は地理的には隔離しているものの、ネットワーク的には非常に高速であり、レイテンシ耐性を検証するために良好な環境であったとはいえない。国際回線などを使用した高レイテンシ環境での評価が必要である。

参考文献

- 1) The globus project. <http://www.globus.org>.
- 2) Mpich-g2. <http://www3.niu.edu/mpich>.
- 3) Koji Hukushima and Koji Nemoto. Exchange monte carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan Vol 65 1604-1608page*, 1996.
- 4) N.Karonis, B.Toonen, and I.Foster. Mpich-g2: A grid-enabled implementation of the message passing interface. *Journal of Parallel and Distributed Computing (JPDC)*, 2003.
- 5) Yuji Sugita and Yuko Okamoto. Replica-exchange molecular dynamics methods for protein foldings. *Chemical Physics Letters Vol.314 page 141-151*, 1999.
- 6) 岡崎進, 岡本祐幸 (編). 生体系のコンピュータシミュレーション. 化学同人, 2002.
- 7) 佐藤仁, 伊藤正勝, 中田秀基, 松岡聡. レプリカ交換分子動力学シミュレータ remd toolkit のグリッド上での実行. 情報処理学会研究報告 HPC-95, 2003.
- 8) 伊藤正勝, 長嶋雲兵. レプリカ交換分子動力学法による分子シミュレーションの並列化 remd toolkit. 情報処理学会研究報告 HPC-95, 2003.
- 9) 片岡洋右. 分子動力学法とモンテカルロ法. 講談社, 1994.