

OBIGrid における HMMER 高速化の検討

小舟 康予[†], 小坂 隆浩[†], 福田 晃^{††}

[†] 大阪産業大学 ^{††} 九州大学

OBIGrid において, 分散実行による HMMER 高速化方式について検討する. Grid 環境において HMMER を実行するために, MPI を用いて並列化した MPI-HMMER を実装した. 通信遅延が大きい Grid 環境において, MPI-HMMER 高速化の可能性を検討するために, OBIGrid 内の複数のサイトの計算機を使用し, MPI-HMMER の処理の分割, 各計算機への処理の割り当てを変化させ, 実行した. 各サイト内における MPI-HMMER の実行では, 実行時間が短縮できることを示したが, 複数のサイトの計算機を用いての実行では, 通信オーバーヘッドが大きく実行時間が増加した. サイト間での通信頻度を低くするために, サイトごとに処理を行うシーケンスを割り当てて実行した. この入力シーケンス分割による実行方法では, 複数のサイトの計算機を用いることで, 単一のサイトのみでの実行時間よりも短縮できることを示した.

Faster Execution of Parallelized HMMER on the OBIGrid

Yasuyo Kofune[†], Takahiro Koita[†], and Akira Fukuda^{††}

[†] Osaka Sangyo University

^{††} Kyushu University

HMMER is a useful genomic application program widely used by genomic researchers. However, HMMER takes much execution time to analyze many genomic sequences. To reduce the execution time, MPI-HMMER is implemented and evaluated on the OBIGrid. The evaluation results show that the execution time of MPI-HMMER can be reduced by parallel execution in each site. To improve the execution time, MPI-HMMER is modified to reduce the communication frequency. For the modified MPI-HMMER, the sequences are divided and the divided sequence is performed in each site. The modified MPI-HMMER can reduce the execution time and it can process the sequences efficiently than the original MPI-HMMER.

1 はじめに

HMMER¹⁾ は, 蛋白質シーケンスの分類・解析を行うゲノムアプリケーションである. HMMER は, Sean Eddy(Washington University) によって, 1998 年より開発が開始され, 2004 年現在, hmmer-2.3 がリリースされている. 有用なゲノムアプリケーションとして, 多くの研究者に利用されている. しかし, HMMER を用いた分類・解析処理は, ひとつの蛋白質シーケンスに対して, 多数の既存の蛋白質シーケンスの共通点をモデル化した Profile-HMM とのマッチング処理を繰り返す. そのため, 長い実行時

間が必要となる. HMMER の実行時間の短縮は重要な課題である.

HMMER の実行時間短縮が実現可能な計算環境として, Grid 環境があげられる. Grid 環境とは, 広域ネットワーク上に分散配置された計算資源を統合的に扱うことで, PC クラスタ以上の処理能力が実現可能な計算環境である. Grid 環境は, 処理能力が異なる多種多様な計算機で構成され, 各計算機の負荷や通信遅延など動的に変化する要素が多い計算環境である.

HMMER には, PVM による単純な並列実行方式が実装されている. しかし, 実装されている PVM による並列実行方式は, 処理を小さく分割して, 各計算機に割り当てるため, 通信頻度が高くなる. Grid 環境のように通信遅延が大きい環境では, 通信オー

[†] 大阪産業大学
Osaka Sangyo University
^{††} 九州大学
Kyushu University

バヘッドが大きくなり実行時間の短縮は期待できない。HMMER の実行時間の短縮のためには、Grid 環境の特徴を考慮した並列実行方式を実装する必要がある。本研究では、Grid 環境で容易に実行可能な MPI を用いて並列化した MPI-HMMER を実装する。さらに、Grid 環境での実行時間の短縮を目的とし、OBIGrid(Open BioInformatics Grid)²⁾ 内の複数のサイトの計算機を用いて、実装した MPI-HMMER を評価する。

2 HMMER の並列処理

HMMER には実行時間のかかるマッチング処理に関して、PVM による並列実行方式が実装されている。HMMER に実装されている PVM による並列実行方式は、処理形態はマスタースレーブ方式をとり、依存関係のない多数のジョブを並列に実行するという単純な方式である。マスタースレーブ計算機が処理をマッチング要素ごとの小さな処理単位に分割し、分割した小さな処理単位で逐一各スレーブ計算機に処理を割り当てる。本研究では、マスタースレーブ計算機がスレーブ計算機に割り当てる処理単位をブロックと呼ぶこととする。また、PVM による並列実行方式において、マスタースレーブ計算機に 1 回の通信で割り当てるブロックをブロックサイズ 1 とする。

小さなブロックに分割する並列実行方式は、通信遅延の小さい PC クラスタや 1 台の並列計算機では、比較的小さい通信オーバーヘッドで実行可能であり、実行時間の短縮も可能である。しかし、Grid 環境のように通信遅延の大きい環境では、小さなブロックに分割する並列処理は、通信オーバーヘッドが非常に大きくなり、実行時間が増加することが考えられる。各スレーブ計算機に割り当てるブロックを大きくすることで、通信オーバーヘッドを抑えられる可能性はあるが、実装されている PVM による並列実行方式ではブロックサイズ 1 でしか並列処理を行うことはできない。また、Grid 環境のようなヘテロな環境では、単純にブロックサイズを大きくした場合、各計

算機に割り当てられる処理のバランスが悪くなり、逆に実行時間が増加することも考えられる。

3 MPI-HMMER

既存の HMMER は、PVM により並列実行方式を実現している。しかし、Grid 環境では GlobusToolkit³⁾ が事実上標準として用いられており、GlobusToolkit によりサポートされていない PVM は Grid 環境での実行に適さない。本研究では、既存の HMMER を MPI を用いて並列化した MPI-HMMER を実装した。Grid 環境においては、GlobusToolkit 上で利用可能な MPICH-G2⁴⁾ が実装されているため、MPI を用いて並列化することにより、Grid 環境で容易に実行可能となる。

MPI-HMMER の処理形態は、既存の PVM による並列実行方式と大きくは変わらず、マスタースレーブ方式である。異なる点は、ブロックサイズが変更可能な点である。ブロックサイズは、2 種類の方法で指定可能とした。1 つは、すべてのスレーブ計算機に対して、すべて同じブロックサイズを指定する方法である。もう 1 つは、スレーブ計算機ごとに異なるブロックサイズを指定する方法である。すべてが同じ処理能力の計算機で構成される計算環境においては、すべての計算機に同じブロックサイズを指定する方法で、実行時間の短縮は可能である。しかし、Grid 環境のような異なる処理能力をもつ多数の計算機からなる計算環境では、計算機ごとにブロックサイズを指定することが重要となる。計算機の処理能力や通信遅延などを考慮してブロックサイズを指定することで、Grid 環境に適した並列実行方式を実現できる可能性があるが、それぞれの計算機に対するブロックサイズについては十分な検討が必要である。

4 評価

MPI-HMMER を、実際の Grid 環境で実行し、実行時間の比較・検討を行った。マッチングに用いた

表 1: 評価環境

サイト	CPU	クロック数 (MHz)	メモリ (MB)	台数 (台)
理化学研究所	Celeron	1300	900	9
同志社大学	Pentium	600	128	8
大阪産業大学	AthlonXP	1700	256	5
	Celeron	2000	256	5

データベースは Pfam7.2 で、入力シーケンス数は 15 本とした。

4.1 評価環境

評価は、OBIGrid(Open BioInformatics Grid) に登録されている理化学研究所・同志社大学・大阪産業大学の計算機 27 台を用いて行った。OBIGrid とはバイオインフォマティクスの研究の推進を目的に立ち上げられ、大学・企業・研究所が計算機を提供して構築された Grid 環境である。OBIGrid は、各サイト間を VPN(Virtual Private Network) 接続し、GlobusToolkit を用いて Grid 環境を構築している。2004 年 1 月現在、27 サイトが参加し、約 360 台の計算機が登録されている。使用した計算機の概要を表 1 に示す。

各サイト内では NFS によりファイル共有されているが、サイト間でのファイル共有はない。あらかじめデータベース、シーケンスファイルなど実行に必要なファイルはすべて、MPI-HMMER 実行前にコピーしておく。実際には、これらのファイルは実行時に必要な計算機にオンデマンドにコピーされることが望ましいが、ファイルコピーのオーバーヘッドが大きいため、実行前のコピーとした。

4.2 評価方法

はじめに、MPI-HMMER の実行時間と比較し、実行時間の短縮を検討するために必要となる各サイトの計算機 1 台での HMMER の実行時間を計測する。

次に、MPI-HMMER を以下の 2 通りの方法で実

表 2: 計算機 1 台での実行時間

計算機	実行時間 (秒)
理化学研究所サイト	1498
同志社大学サイト	2336
大阪産業大学サイト (AthlonXP)	1278
大阪産業大学サイト (Celeron)	1021

行する。

1. 各サイト内でブロックサイズを変化させ実行する。
2. 複数のサイトの計算機を組み合わせ、ブロックサイズを変化させ実行する。

MPI-HMMER の実行時間と計算機 1 台での HMMER の実行時間を比較し、Grid 環境における MPI-HMMER の実行時間の短縮を検討する。

4.3 評価結果

表 2 に、各サイトの計算機 1 台で HMMER を実行した場合の実行時間を示す。計算機の処理能力により、実行時間に大きな差がでる。計算機 1 台での HMMER の実行時間より、MPI-HMMER の実行時間の短縮のためには、計算機の処理能力を考慮してブロックサイズを指定することが重要であると考えられる。

各サイト内で MPI-HMMER を実行した場合の実

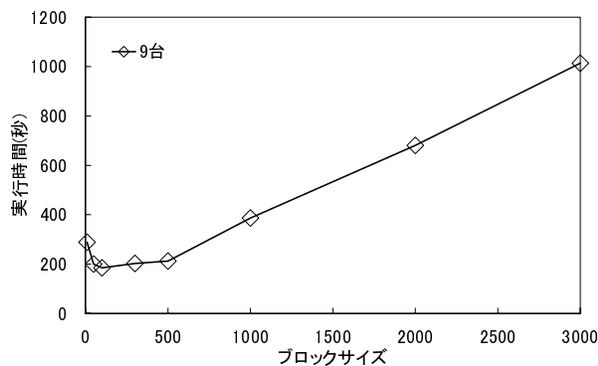


図 1: 理化学研究所サイトでの実行結果

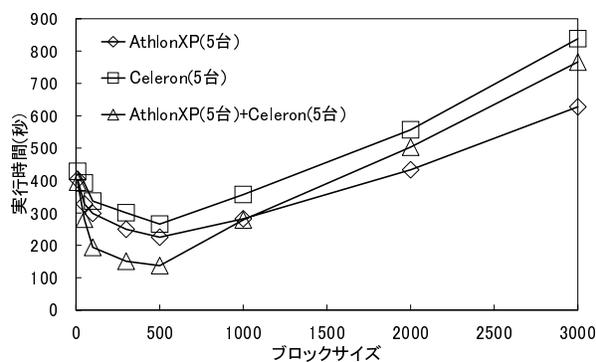


図 3: 大阪産業大学サイトでの実行結果

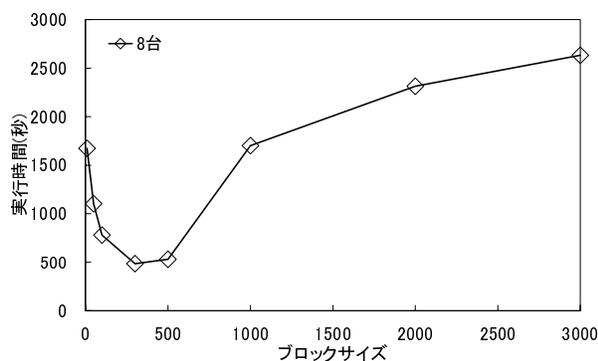


図 2: 同志社大学サイトでの実行結果

行結果を示す．図 1 に理化学研究所サイトでの実行時間を，図 2 に同志社大学サイトでの実行時間を示す．理化学研究所サイトでは計算機 9 台を使用し，ブロックサイズは，10 から 3000 まで変化させた．同志社大学サイトでは計算機 8 台を使用し，ブロックサイズについては，理化学研究所サイトと同様に 10 から 3000 まで変化させた．

図 3 に大阪産業大学サイトでの実行時間を示す．大阪産業大学サイトは，処理能力の異なる計算機が複数あるので，処理能力ごとに計算機をグループ分けし，グループごとに MPI-HMMER を実行した場合と，複数のグループを組み合わせる場合の実行時間を計測した．ブロックサイズは，理化学研究所サイト，同志社大学サイトと同様に，10 から

3000 まで変化させた．

理化学研究所，同志社大学，大阪産業大学いずれのサイトにおいても，各サイトの 1 台の計算機での実行時間と比べると，いずれのブロックサイズを指定した場合も実行時間は短縮されている．しかし，ブロックサイズ 10 という小さなブロックサイズでの実行やブロックサイズ 1000 以上という大きなブロックサイズでの実行では，他のブロックサイズでの実行時間に比べて実行時間が増加している．

ブロックサイズ 10 という小さなブロックサイズでの実行は，大きなブロックサイズでの実行に比べて通信頻度が高くなる．ブロックサイズ 10 で実行時間が大きくなっているのは，実行時間に通信オーバーヘッドが大きく占めているためだと考えられる．ブロックサイズ 1000 以上の大きなブロックサイズでの実行は，通信オーバーヘッドは抑えられるが，1 台の計算機に一度に割り当てられる処理量が大きくなる．そのため，各計算機に割り当てられる処理のバランスが悪くなり，実行時間が大きく増加したと考えられる．また，いずれのサイトにおいても，計算機の処理能力により実行時間にばらつきはあるが，ブロックサイズ 100 から 500 での実行では，実行時間が短縮されている．理化学研究所サイトではブロックサイズ 100，同志社大学サイトではブロックサイズ 300，大阪産業大学サイトではブロックサイ

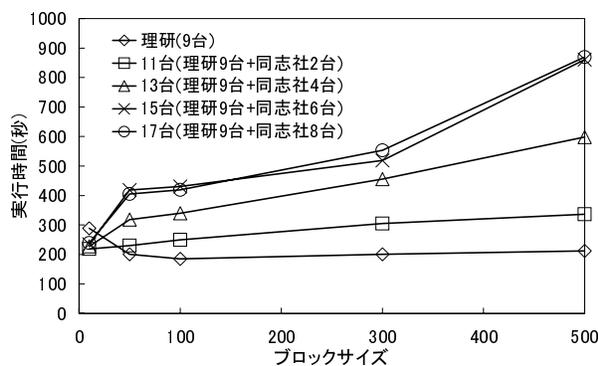


図 4: 理研-同志社大での実行結果

ズ 500 での実行が最も実行時間が短縮されている。単一のサイト内では、ブロックサイズ 100 から 500 での実行が、通信オーバーヘッドが実行時間に与える影響は少なく、各計算機の処理のバランスも一定程度保たれていると考えられる。

次に、複数のサイトの計算機を組み合わせ、MPI-HMMER を実行した場合の実行時間を示す。図 4 に理化学研究所サイトと同志社大学サイトの 2 つのサイトにおいて計算機の組み合わせを変化させ、MPI-HMMER を実行した場合の実行時間を示す。各計算機には、同じブロックサイズを指定した。大阪産業大学サイトの計算機を組み合わせ、同様に実行を試みたが、大阪産業大学サイトの計算機を加えると、通信遅延が非常に大きいため実行できなかった。ここでは、理化学研究所サイトと同志社大学サイトの 2 つのサイトでの実行結果のみを示す。

実行結果を見ると、いずれの計算機の組み合わせにおいても、理化学研究所サイトの実行時間よりも大きな実行時間となっている。複数のサイトを用いることにより、使用できる計算機台数は増え、実行時間の短縮が期待できる。しかし、実際は、サイト間の通信遅延が非常に大きく、実行時間に通信オーバーヘッドが大きく影響するため、単一のサイトでの実行よりも、実行時間が増加した。

ブロックサイズを大きくすることで、通信頻度を

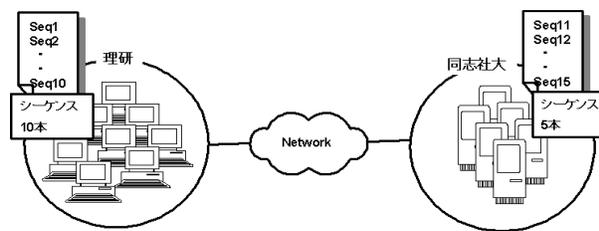


図 5: シーケンス分割実行方法の概要

抑え、実行時間を短縮することを検討した。しかし、複数のサイトを組み合わせでの実行では、予想していた以上に通信遅延が大きかったために、実行時間の短縮は困難であった。

次に、入力シーケンスを分割することにより、通信頻度を低くする方式について検討する。複数のサイトを使用しての MPI-HMMER の実行時間の短縮には、サイト間での通信頻度を低くする必要がある。15 本のシーケンスが登録されているシーケンスファイルを指定し、MPI-HMMER を実行すると、まず 1 本目のシーケンスを実行時に指定したすべての計算機を使用し処理する。1 本目の処理が終了すると 2 本目のシーケンスの処理をすべての計算機で行う。2 本目が終了すると 3 本目というように、実行時に指定したすべての計算機で、1 本ずつ処理を行う。1 本のシーケンスを処理するのに、サイト間で数回から数十回の通信を行う必要がある。

サイト間での通信頻度を抑えるために、シーケンスファイルを分割し、サイトごとに処理を行うシーケンスを割り当て、各サイト内で MPI-HMMER を実行するようにした。各サイト内での処理結果は、指定した計算機にファイルで送信されるようにした。図 5 に、シーケンスファイルを分割して実行する場合の概要を示す。

図 6 に、理化学研究所サイトにシーケンス 10 本、同志社大学サイトにシーケンス 5 本の処理を割り当てた場合の実行結果を示す。

ブロックサイズ 10 では、理化学研究所サイトの

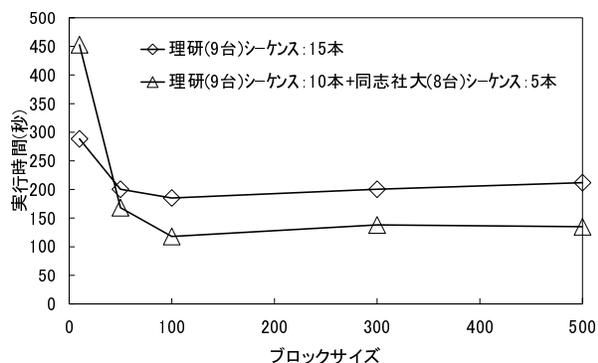


図 6: 理研-同志社大での実行結果 (シーケンス分割)

みで実行した場合に比べて、理化学研究所サイトと同志社大学サイトの2つのサイトを用いると大きく実行時間が増加している。しかし、ブロックサイズ50以上では、2つのサイトの計算機を使用して実行することで、理化学研究所サイトのみでの実行時間の約40%の実行時間の短縮が実現できた。複数のサイト間で実行することで、通信遅延は大きくなる。しかし、サイト間での通信頻度をできるだけ低くし、各サイト内のすべての計算機を有効に利用することで、実行時間の短縮は可能である。

今回は、各サイト内での実行結果より、理化学研究所サイトに10本のシーケンスを、同志社大学サイトに5本のシーケンスを割り当てた。多くのサイトを用いて、サイト内の計算機台数、計算機の処理能力などを考慮して、指定するブロックサイズ、割り当てるシーケンス数を決定することで、さらに実行時間の短縮は期待できると考えられる。

5 まとめと今後の課題

HMMER を MPI を用いて並列化した MPI-HMMER を実装し、OBIGrid 内での計算機で実行・評価した。

各サイト内の評価結果より、MPI-HMMER の実行時間の短縮に有効なブロックサイズについて検討した。小さなブロックサイズでは、各計算機の処理

のバランスはよいが、実行時間に通信オーバーヘッドが大きく占めるため十分な実行時間の短縮が望めない。また、単純にブロックサイズを大きくして実行すると、通信オーバーヘッドは抑えられるが、特定の計算機のみ処理が割り当てられ、すべての計算機が有効に利用されず逆に実行時間が増加する。

サイト間での実行は、通信オーバーヘッドが非常に大きくなるために、実行時間の短縮が困難であった。サイト間の通信頻度を低くするために、シーケンスファイルを分割し、サイトごとにシーケンスを割り当て、各サイト内で実行した。シーケンスファイルを分割して、2つのサイトを用いて実行することにより、単一のサイトのみを用いた場合の実行時間よりも、実行時間が短縮できた。

今回、評価に使用した計算機環境は、限られた条件の Grid 環境である。異なる条件の Grid 環境での MPI-HMMER の評価は必要である。また、異なるデータベースやシーケンスでの評価も必要である。

参考文献

- 1) HMMER, <http://hmmer.wustl.edu/>.
- 2) OBIGrid, <http://www.obigrid.org/>.
- 3) Globus Project, <http://www.globus.org/>.
- 4) MPICH-G2, <http://www3.niu.edu/mpi/>.