

## Grid 環境における HMMER 実行方式の検討

小舟 康予<sup>†</sup>, 小坂 隆浩<sup>†</sup>, 福田 晃<sup>††</sup>

Grid 環境において、ゲノムアプリケーション HMMER の実行時間を短縮できる実行方式を検討する。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、OBIGrid に登録されている大阪産業大学・同志社大学・理化学研究所の計算機で実行・評価を行う。各サイトにおいて、ブロックサイズと使用する計算機台数を変化させて MPI-HMMER を実行し、実行時間短縮に有効なブロックサイズを検討する。

### Performance Evaluation of MPI-HMMER on the Grid

Yasuyo Kofune<sup>†</sup>, Takahiro Koita<sup>†</sup>, and Akira Fukuda<sup>††</sup>

HMMER is a useful genomic application program widely used by genomic researchers. However, HMMER takes much time to analyzing many genomic sequences. To reduce the execution time, MPI-HMMER is implemented and evaluated on the OBIGrid.

## 1 はじめに

HMMER<sup>1)</sup> は、蛋白質データを分類・解析するアプリケーションである。有用な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 は、ひとつの蛋白質データに対して、繰り返し、多数の既存の蛋白質データとの比較を行うため実行時間がかかる。また、蛋白質データの分類・解析は、繰り返し、何度も処理を行う必要がある。よって、HMMER の実行時間短縮は重要な課題である。

HMMER の実行時間の短縮が可能な環境として、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環境である。Grid 環境は、処理能力が一樣ではない計算機で構成され、通信遅延など動的に変化する要素が多い環境である。

HMMER は、処理を複数のブロックに分割可能であり、それぞれのブロックは独立して処理が可能であるため並列処理に適しており、PVM による並列実行方式が実装されている。実装されている実行方式は、処理を最小のブロックサイズに分割し、最小のブロックを逐一計算機に割り当てるセルフスケ

ジューリング方式である。しかし、Grid 環境において HMMER を実行する場合、最小のブロックを広域ネットワーク上で割り当てると、非常に通信オーバーヘッドが大きくなるため、実行効率が悪く、大きな速度向上も望めない。

本研究では、Grid 環境で実行可能な MPI による HMMER の並列実行方式を実装する。さらに、Grid 環境での実行時間の短縮を目的とし、OBIGrid 内の複数のクラスタにて、HMMER の並列実行方式を評価する。

## 2 HMMER

HMMER は、Sean Eddy(Washington University) によって開発されたゲノムアプリケーションである。HMMER は、シーケンス群から profile-HMM を作成し、profile-HMM を基準にデータベースとのマッチング処理を行う。データベースとのマッチング処理を繰り返し、蛋白質データを分類・解析するアプリケーションである。

HMMER は、処理を複数のブロックに分割可能であるため、並列処理に適したアプリケーションである。HMMER の実行時間の大部分は、データベースとのマッチング処理である。そのため、HMMER には、PVM を用いた並列実行方式が実装されている。実装されている並列実行方式は、マッチング要素を小さいブロックに分割し、分割した多数のブロックを並列に処理するという単純なセルフスケジューリ

<sup>†</sup> 大阪産業大学  
Osaka Sangyo University

<sup>††</sup> 九州大学  
Kyushu University

ング方式である。この小さいブロック単位によるセルフスケジューリングは、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るが、Grid 環境のような通信遅延の大きい環境では通信オーバーヘッドが非常に大きくなる可能性がある。ブロックサイズを変更することにより、より小さい通信オーバーヘッドでの実行は可能となるが、PVM による並列実行方式では最小のブロックサイズでしか並列処理をおこなうことはできない。また、ブロックサイズを大きくした場合、各計算機での処理のバランスが悪くなり、実行時間が増加することも考えられる。Grid 環境において、HMMER を短時間で実行するためには、通信オーバーヘッドと処理のバランスの基本的なトレードオフを評価し、評価結果をもとに、新たな並列実行方式を検討する必要がある。本研究では、特に通信オーバーヘッドと処理のバランスに着目し、PC クラスタにて評価する。

### 3 実装

従来の 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 GlobusToolkit<sup>2)</sup> 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 環境に適さない。そこで、HMMER を MPI を用いて並列化し、MPI-HMMER を実装した。Grid 環境においては、GlobusToolkit 上で利用可能な MPICH-G 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 環境で容易に利用可能となる。

MPI-HMMER の処理は、マスタースレーブ方式である。マスターが処理をブロックに分割し、分割したブロックを各スレーブに割り当てる。スレーブが割り当てたブロックを処理し、処理結果をマスターに返すと、マスターより新たなブロックが割り当てられる。すべての処理が終了すると、マスターは処理結果をまとめる。このときブロックサイズは、2 種類の方法で指定可能とした。1 つは、すべてのスレーブに対して、すべて同じブロックサイズとする

表 1: 評価環境

サイト	CPU[MHz]	メモリ [MB]	台数 [台]
osu 大産大	PentiumII 350	128	4
oasis 同志社	PentiumIII 600	256	6
le 理研	Celeron 1300	900	5

方法である。もう 1 つは、各計算機毎にブロックサイズを指定し、計算機毎に異なるブロックサイズとする方法である。すべてが同じ計算機環境においては、すべて同じブロックサイズとする方法で速度向上が望める。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環境では、計算機毎にブロックサイズを指定することにより、Grid 環境に適した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と考えられる。

### 4 評価

前章で述べた MPI-HMMER を、実際の Grid 環境で実行し、実行時間の比較・検討をおこなった。計算機台数とブロックサイズを変化させ、実行時間を測定した。ブロックサイズの指定は、評価を単純にするため、すべての計算機に対してすべて同じブロックサイズとした。

#### 4.1 評価環境

評価は、OBIGrid(Open BioInformatics Grid)<sup>3)</sup> に登録されている大阪産業大学・同志社大学・理化学研究所の計算機 15 台を用いて行った。OBIGrid とはバイオインフォマティクスの推進を目的に立ち上げられた Grid 環境である。OBIGrid は、各サイト間を VPN(Virtual Private Network) 接続し、Globus Toolkit を用いて Grid 環境を構築している。2003 年 7 月現在、26 サイトが参加し、約 300 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

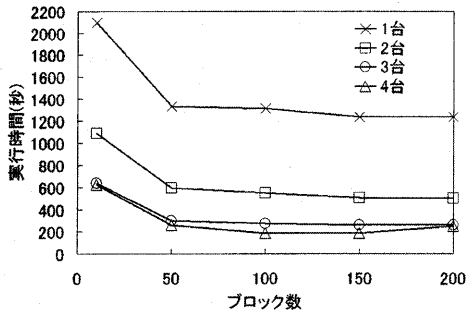


図 1: 大阪産業大学での実行時間

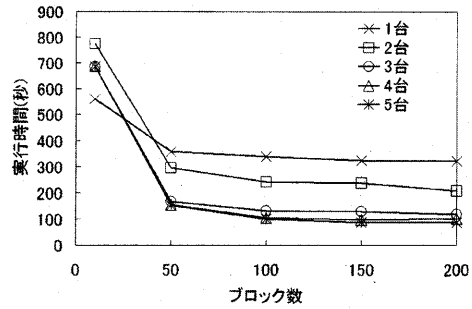


図 2: 理化学研究所での実行時間

す。いずれのサイトも、サイト内の計算機はすべて同じ性能である。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、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適宜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 評価結果

まず各サイトごとに、使用する計算機台数とブロックサイズを変化させて、MPI-HMMER を実行した。マッチングのためのデータベース TIGRFAMs を用い、入力シーケンス数は 5 とした。

ブロックサイズを 10 から 200 まで変化させ、計算機台数を 1 台から 4 台まで変化させた場合の、大阪産業大学サイトでの実行時間を図 1 示す。同様に、理化学研究所サイトの実行時間を図 2 に示す。同志社大学サイトにおいても、同様の評価をおこなったが、OBIGrid は単一ユーザが占有できる環境でないため、他プロセスとの干渉と VPN 性能のばらつきにより、安定した実行時間が得られなかった。そのため、ここでは大阪産業大学と理化学研究所のみの結果を示す。

大阪産業大学、理化学研究所サイトとも、ブロックサイズを増加させると、ブロックサイズ 50 まで

は実行時間が大きく短縮されている。しかし、その後、100、150、200 とブロックサイズを増加させても、大きな実行時間の短縮はみられない。ブロックサイズが 10 の場合、より大きなブロックサイズに比べ、多数の通信が発生する。通信オーバーヘッドが実行時間に占める割合が大きくなっているため、ブロックサイズ 10 の実行時間が大きくなっていると考えられる。しかし、ブロックサイズが 50 より大きい場合、クラスタ内の通信では、通信オーバーヘッドが実行時間に与える影響は少ないとみられる。また、ブロックサイズが大きくなっても、実行時間の増加がほとんどないことから、処理のバランスもある程度保たれていると考えられる。さらに大きなブロックサイズの場合、実行時間が伸びる可能性が考えられるが、実装上の問題があり本研究では評価できなかった。Grid 環境では、もっと通信遅延の大きい場合も考えられるため、大きなブロックサイズによる評価も必要である。

使用する計算機台数に対しては、大阪産業大学サイトではいずれのブロックサイズにおいても、計算機台数が増えるにしたがって実行時間は短縮されているが、理化学研究所では、ブロックサイズ 10 の場合は計算機台数が増えると実行時間が増加している。ブロックサイズが 50 以上の場合は、大阪産業大学サイトと同様に実行時間は短縮されている。

次に、理化学研究所サイトにおいて、それぞれの

3) obigrid, <http://www.obigrid.org/>.

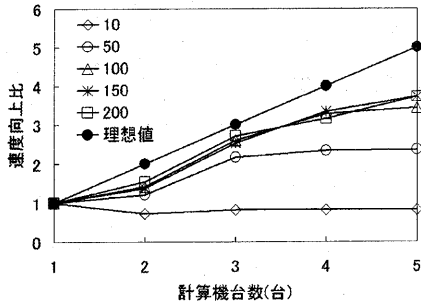


図 3: 理化学研究所での速度向上比

ブロックサイズに対して、使用する計算機台数を増やした場合の速度向上比を図 3 示す。

ブロックサイズ 10 では、台数増加による通信オーバーヘッドの方が大きいため速度向上比は 1 以下の値となっている。しかし、ブロックサイズを 100, 150, 200 と増加させていくと理想値に近づいていくことが分かる。ブロックサイズ 100 以上では、5 台までの計算機台数に対して、ある程度実行時間の短縮が期待できる速度向上比が得られている。

## 5 まとめと今後の課題

HMMER を MPI を用いて並列化し、実行時間短縮のための実行方式を検討した。ブロックサイズ 10 以下の小さいブロックサイズは、処理のバランスは良いが、通信オーバーヘッドが大きくなるため十分な実行時間短縮が望めない。しかし、ブロックサイズ 100 以上では、比較的少ない通信オーバーヘッドで、適当な処理のバランスを実現し、ある程度の速度向上が得られることが分かった。今後の課題としては、複数のサイト間で MPI-HMMER を実行し、複数のサイト間での実行時間短縮を検討する。また、各計算機の負荷なども考慮したブロックサイズの割り当て方法を検討する。

## 参考文献

- 1) HMMER, <http://hmmer.wustl.edu/>.
- 2) Globus Project, <http://www.globus.org/>.