

## Grid 環境におけるスケジューリング方式の検討

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

<sup>†</sup>大阪産業大学 <sup>‡</sup>九州大学

### 1 はじめに

ゲノムアプリケーションのひとつである HMMER[1] を対象とした, Grid 環境におけるスケジューリング方式を検討する.

HMMER は, 蛋白質データを分類, 解析するアプリケーションであり, BLAST などより複雑な処理を行うため, 実行時間がかかる. そのため, 実行時間の短縮は重要な課題である. HMMER の実行時間の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 環境は, 処理能力が一樣ではない計算機で構成され, 動的に変化する要素が多い. HMMER の実行時間短縮には, これらの要素を考慮したスケジューリング方式が重要となる.

### 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. 現在, hmmer2.2g がリリースされている. 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 には, PVM を用いた並列処理が実装されている. しかし, 実装されているスケジューリング方式は, マッチング要素ごとに処理を分割し, 分割した多数の処理を並列に実行するという単純なセルフスケジューリングである. そのため, Grid 環境のようにヘテロな計算環境では, 処理が特定の計算機に集中したり, ネットワークの状態によっては, 処理結果の通信遅延が大きくなり, 実行時間が増加する可能性がある.

### 3 Grid 環境とスケジューリング

1 台の処理能力の高い計算機やクラスタのみを使用することによる, HMMER の実行時間の短縮には限界がある. Grid 環境は, Grid 環境の特性を考慮した

スケジューリングを行うことで, PC クラスタ以上の処理能力が実現できる環境である.

HMMER を PVM により並列処理した場合の実行時間の例を表 1 に示す. PCX は, Pentium4 1700(MHz) の計算機を表し, PCY は, Pentium2 348(MHz) の計算機を表す.

表 1: 並列処理した場合の実行時間

計算機	実行時間 (sec.)
PCX(1 台)	1520
PCX(9 台)	660
PCX(9 台)+PCY(8 台)	811

PCX1 台で実行すると, 実行時間は 1520 秒かかる. PCX9 台で実行することで, 実行時間は 1 台で実行した場合の約 43% になる. しかし, PCX(9 台) に PCY(8 台) を加えると, 逆に実行時間が増加する. 原因としては, PCX と PCY の処理能力が大きく異なるため, PCX が効率良く利用されていない可能性が考えられる. また, 9 台から 17 台と台数が増加したことによる, 通信オーバーヘッドの増加が考えられる. 単純な台数の増加が実行時間の短縮につながるとは限らず, 加える計算機によっては, 逆に実行時間が増加してしまう可能性がある.

以上より, Grid 環境において HMMER の実行時間を短縮するためには, 処理能力を考慮し, 適切な計算機を, 適切な台数を使用することが重要である.

### 4 実装

Grid 環境の特性を考慮したスケジューリングを実現するためには, いくつかの機能を実装する必要がある. 必要な機能は, GlobusToolkit2.0[2] により実装した. また, HMMER は, PVM を用いた並列処理が可能となっているが, Grid 環境上では, PVM はサポートされていない. 並列処理に関しては, MPICH-G2[3] が提供されている. HMMER を MPI を用いて並列化し, 実装した [4].

Grid 環境における HMMER のスケジューリング方式は以下のような流れで行われる.

A Study of Scheduling Schemes on the Grid

<sup>†</sup> Yasuyo Kofune, Takahiro Koita

<sup>‡</sup> Akira Fukuda

Osaka Sangyo University (<sup>†</sup>)

Kyushu University (<sup>‡</sup>)

1. 情報収集
2. 使用計算機の選択
3. HMMER の実行
4. 使用計算機の解放

#### 4.1 情報収集

Grid 環境を構成している各計算機の情報は、GlobusToolkit2.0 の機能のひとつである MDS(Metacomputing Directory Service) より得る。MDS は、Grid 環境を構成している各計算機の CPU アーキテクチャ、メモリサイズといった静的な情報と、計算機の負荷やメモリ使用状況といった動的な情報を管理し、ユーザに提供するサービスである。ここでは、使用計算機を選択に必要な情報として、計算機名、CPU クロック値、ドメイン名を収集する。

#### 4.2 使用計算機を選択

MDS より収集した情報をもとに、使用計算機を選択する。選択対象とする計算機と処理能力の低い計算機の利用有無により、複数の計算機選択方式を実装した。選択対象とする計算機を、全計算機とする (ALL) 方式、同一ドメイン内のみとする (DOMAIN) 方式、クラスタのみとする (CLUSTER) 方式の 3 方式とする。ALL 方式は、利用可能な計算機数は多いが、通信オーバーヘッドが増加する可能性がある。逆に、CLUSTER 方式は、利用可能な計算機は少ないが、通信オーバーヘッドを抑えられる。DOMAIN 方式は、ALL 方式と CLUSTER 方式の中間の性質を持った方式である。

さらに、処理能力の低い (LOW) 計算機を選択対象から除外する方式についても実装した。処理能力の低い計算機は、利用対象として選択しても、大きな処理能力の向上は期待できず、かえって処理能力を低下させる可能性もある。LOW 計算機を対象とする方式と除外する方式を実装した。表 2 に実装した計算機選択方式をまとめる。

表 2: 計算機選択方式

方式	対象	LOW 計算機
ALL	全計算機	対象
DOMAIN	同一ドメイン内	対象
CLUSTER	クラスタ	対象
ALL-HIGH	全計算機	除外
DOMAIN-HIGH	同一ドメイン内	除外
CLUSTER-HIGH	クラスタ	除外

使用計算機を決定する際、既に、HMMER の実行に使用されている計算機は選択対象から除外することと

する。但し、選択対象の計算機が全て使用されている場合は、既に使用されている計算機も選択対象となる。

#### 4.3 HMMER の実行

使用計算機が決定すると、mpirun コマンドより、HMMER を実行する。

#### 4.4 使用計算機の解放

HMMER の実行が終了すると、使用計算機は解放され、再び、使用可能となる。但し、重複して使用されている場合は、全ての実行が終了するまで解放されない。

### 5 性能評価

前述のスケジューリング方式を実際の Grid 環境で評価予定である。評価環境は、大阪産業大学・九州大学から構成される Grid 環境を用いる。HMMER の実行時間を短縮するためには、使用計算機を選択と共に、使用台数が重要となる。4.2 節で述べたように、選択対象とする計算機を 6 つの方式に分け、それぞれの方式が対象とする計算機の中から、使用計算機を決定する。使用台数の決定方式については、現在検討中である。それぞれの方式で決定された計算機を用い、HMMER の実行時間を計測する。それぞれの方式で計測した実行時間を比較し、HMMER の実行時間をより短縮できるスケジューリング方式を検討する。

### 6 まとめと今後の課題

HMMER を MPI を用いて並列化し、Grid 環境での HMMER のスケジューリング方式を実装した。

今後の課題は、計算機の分類基準や選択対象の妥当性について検討する。Grid 環境上で実装したスケジューリング方式を用い、HMMER を実行し、実行時間短縮のための適切なスケジューリング方式を検討する。さらに、計算機の負荷、通信量、通信回数を考慮したスケジューリング方式についても検討する。

#### 参考文献

- [1] HMMER, <http://hmmer.wustl.edu/>.
- [2] The Globus Project, <http://www.globus.org/>.
- [3] MPICH-G2, <http://www3.niu.edu/mpi/>.
- [4] 池辺 貞郎, メタコンピューティング環境における対話型アプリケーション HMMer のスケジューリング, 奈良先端科学技術大学院大学修士論文, 2002.