

MPI-HMMERのグリッド向けスケジューリング Hueristic Scheduling Scheme for MPI-HMMER

松井 洋介[†]
Yosuke Matsui

小舟 康予[†]
Yasuyo Kofune

小坂 隆浩[†]
Takahiro Koita

福田 晃[‡]
Akira Fukuda

1. はじめに

生物学の分野において、蛋白質データを分類・整理するゲノムアプリケーションとしてHMMER^[1]が利用される。HMMERは、大量の蛋白質データを処理する場合、膨大な実行時間を要するため、実行時間の短縮が重要な課題である。本研究では、Grid環境におけるHMMERの実行時間短縮を目的とする。MPIを用いて並列化したMPI-HMMERを対象に、複数のMPI-HMMERが同時に実行される場合のスケジューリング方法を検討する。Grid環境では、様々な状態の計算機が混在するため、各計算機を有効に利用するスケジューリング方法が重要である。計算機を有効に利用するために、各計算機の負荷や通信状況を考慮したスケジューリング方法の実現方策について検討する。

2. HMMER

HMMERは、蛋白質データベース(HMMデータベース)の検索に広く利用されるゲノムアプリケーションである。HMMERは、プロフィールとして、蛋白質データ配列中の各位置の状態を、20種類のアミノ酸の出現確率として表した隠れマルコフモデル(HMM)を使用する。HMMERを利用することにより、未知の蛋白質シーケンスを入力とし、既存の蛋白質データベースとの比較が可能となり、未知の蛋白質シーケンスの機能を予測することも可能となる。

2.1 HMMERの動作

HMMERの基本的な動作を以下に示す。また、HMMERの動作概念を図1に示す。

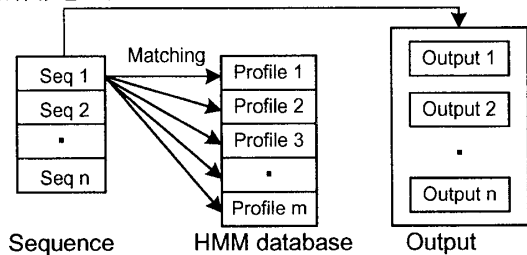


図1: HMMERの動作概念

- (1) 入力シーケンスとHMMデータベースを指定する。
- (2) シーケンス毎にHMMデータベースとのマッチング処理を行う。
- (3) 全てのシーケンスとのマッチング処理を終えたら、シーケンスごとに得られた処理結果を一つの処理結果にまとめる。

2.2 MPI-HMMER

HMMERは、膨大なデータを扱うためマッチング処理の実行時間が大変長いという問題がある。本研究では、HMMERの実行時間短縮を実現する環境としてGrid環境を利用する。Grid環境とは、地理的に分散した計算機を仮想的な一つの計算機とみなし、分散並列処理を行う環境である。Gridを利用することにより、PCクラスタ以上の処理能力を得ることが可能となり、膨大なデータを扱う処理に対しても、実用的な実行時間で処理を終えることが可能となる。しかし、既存のHMMERはGrid環境での実行を前提としていないため、我々はGrid環境で実行可能なMPI-HMMERを実装した。MPI-HMMERの処理形態は、処理の指示を出すマスターと、マスターからの指示を受け取り、実際にマッチング処理を行うスレーブによる、マスター・スレーブ方式を採用している。MPI-HMMERにおける処理の流れを以下に示す。また、図2にMPI-HMMERの処理概要を示す。

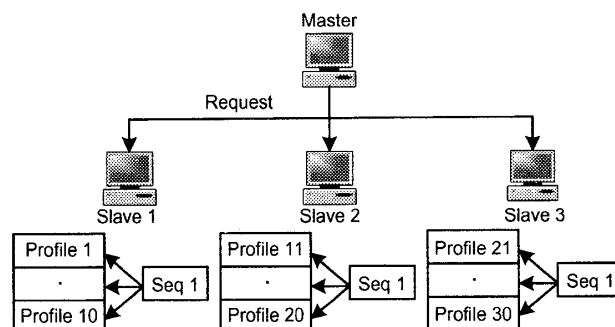


図2: MPI-HMMERの処理

- (1) 入力シーケンスとHMMデータベースを指定する。
- (2) マスターは、処理を複数のブロックに分割する。
- (3) マスターは、分割したブロックを各スレーブに割り当てる。
- (4) 各スレーブは、マスターから送られたブロックに対しマッチング処理を行う。
- (5) 各スレーブは、処理結果をマスターに返す。
- (6) マスターは、処理を終えたスレーブに新たなブロックを割り当てる。
- (7) 全てのマッチング処理が終了するまで、以上の処理を繰り返す。
- (8) マスターは、各スレーブから返された処理結果を一つにまとめる。

MPI-HMMERで実行時間短縮のためには、各スレーブに処理を割り当てるスケジューリング方法を考えなければなら

[†] 同志社大学工学部情報システムデザイン学科

[‡] 九州大学大学院システム情報科学研究科

らない。ここでの、スケジューリングとは、各計算機に処理を割り当てる場合に、どのような処理をどの計算機に割り当てるかを決定することである。Grid環境において効率良く MPI-HMMER を実行するためには、計算機の優先順位や台数を決定する必要がある。また、マスター・スレーブ間の通信オーバーヘッドを考慮し、適切な計算機台数と通信オーバーヘッドのトレードオフを実現するために、計算機の選択対象についても検討する必要がある。Grid環境では、様々な状態の計算機が存在するため、各計算機の負荷や通信などの要因を考慮したスケジューリングを行わなければ実行時間短縮にはつながらない。次章で HMMER の実行時間短縮のためのスケジューリング方法を検討する。

3. スケジューリング方法の検討

3.1 スケジューリングの要件

スケジューリング方法を考える場合に、考慮しなければならないことは、計算機の処理能力と通信オーバーヘッドである。計算機の処理能力を考慮したスケジューリングは、処理能力の高い計算機を優先的に選択することである。一方、通信オーバーヘッドを考慮したスケジューリングは、通信遅延が少ないローカルの計算機など、近くの計算機を優先的に選択する方法である。処理能力を考慮してスケジューリングを行うと、Grid環境すべての計算機が選択対象となるので、処理能力の高い計算機が多数利用可能となるが、通信オーバーヘッドが大きくなることが考えられる。逆に、通信オーバーヘッドを考慮してスケジューリングを行うと、ネットワーク的に近くにある計算機だけが選択対象となるため、通信オーバーヘッドは抑制されるが、他に処理能力の高い計算機があっても選択されない場合がある。処理能力と通信オーバーヘッドの両方を考慮した有効なスケジューリングを実現するには、処理能力と通信オーバーヘッドのトレードオフを考慮し、それぞれの短所をできる限り軽減する必要がある。

3.2 スケジューリング方法

これまで述べたように、MPI-HMMER のスケジューリング方法を実現するためには、適切に計算機を選択する必要がある。MPI-HMMER において、適切な計算機を選択するスケジューリング方法としては、計算機の優先順位・計算機台数・計算機対象についていくつかの実現選択肢が考えられる。次節以降、各選択肢とその得失について述べ、本研究のスケジューリングの実現方策について検討する。

3.2.1 計算機優先順位

計算機を選択をする場合に、各計算機に優先順位をつける。ある特定の計算機をマスターとした場合に、スレーブの処理を短時間で処理可能な計算機程高い優先順位となるように、優先順位を算出する。優先順位は、各計算機の予想実行時間 T に基づいて決定する。各計算機の予想実行時間が短い計算機から順に高い優先順位をつけ、優先順位の高い計算機から選択する。各計算機の予想実行時間 T は、マスターと1つのスレーブ間の通信回数 N 、通信遅延 L 、処理能力 P より算出される。通信遅延 L を決定(予測)するために、NWS^[1] (Network Weather Service) を利用する。また、通信回数 N や処理能力 P はGrid環境内の計算機について予め得られているものとする。各計算機の実行時間は

式(1)より与えられる。

$$T = (L * N) + P \quad (1)$$

3.2.2 計算機台数

HMMERの実行時間短縮のためには、使用する計算機の台数を考える必要がある。使用計算機台数を単に増やすと逆に実行時間が長くなってしまう場合もあり得る。適切な使用計算機台数を決定するには、HMMERのブロックサイズを考慮する必要がある。入力シーケンスやブロックサイズと計算機台数の関連性については、これまで評価結果が得られており、相関性があることが分かっている^[1]。しかし、実際にスケジューリングで計算機台数を決定するためには、さらに実環境で評価を行い、検討する必要がある。

3.2.3 選択対象

計算機を選択対象として、4つの選択方式が考えられる。選択方式として、全ての計算機を選択対象とするA方式、ローカルドメインの計算機のみを選択対象とするB方式、優先順位の平均が最も高いドメインの計算機のみを選択対象とするC方式、クラスタのみを選択対象とするD方式を実装する。A方式は、通信遅延が大きくなる場合があるため、通信オーバーヘッドが増加してしまう。B方式は、通信は抑えられるが、利用できる計算機は少なく、また、ローカルに優先順位の低い計算機が多ければ、実行時間は長くなってしまふ。C方式は、通信オーバーヘッドを抑えられ、優先順位が高いため実行時間の短縮が見込まれるが、利用できる計算機が少ない。D方式は、階層的なマスター・スレーブを採用した場合、有効と考えられる。表1に選択対象の4方式をまとめた。

表1: 選択対象方式

方式	対象計算機
A	全ての計算機
B	ローカルドメインの計算機
C	優先順位の平均が最も高いドメインの計算機
D	クラスタ

4. まとめと今後の課題

Grid環境におけるHMMERのスケジューリング方法を検討した。また、スケジューリング実現方法の選択肢の得失について考察した。

今後の課題は、これらの実現方法をもとにスケジューリングを実装し、妥当性について検討することである。また、HMMERの処理の大きさに応じて、使用する計算機台数を決定する方法に関してもさらに調査する予定である。

参考文献

- [1] HMMER, <http://hmmmer.wustl.edu/>.
- [2] NWS, <http://nws.cs.ucsb.edu/>.
- [3] Yasuyo Kofune, Takahiro Koita and Akira Fukuda, "Performance Evaluation of MPI-HMMER on the OBIGrid", Proc. of the Parallel and Distributed Processing Techniques and Applications (PDPTA2004), pp.476-480, 2004.