

## Grid 環境における負荷分散機能の実装

東原 大記<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

広域ネットワーク上に配置された計算資源を、一つの計算機環境として提供可能とする、Grid 環境の研究が盛んに行われている。

Grid 環境とは、計算機構成やネットワーク構成が異なる、ヘテロな計算機環境である。多数の処理が投入される場合があるため、処理が特定の計算機に集中する可能性がある。処理の集中は、応答時間の増加につながるため、適切な計算機に処理を分散する負荷分散が必要である。

本研究では、Grid 環境における標準的なツールキットである、Globus Toolkit[1] を用いて、負荷分散機能を実装し、ゲノムアプリケーションへ応用する。

### 2 負荷分散

負荷分散 [2] とは、計算機の負荷を分散することにより計算資源を効率良く利用することである。Grid 環境において、応答時間の短縮や資源の利用効率の向上などが期待できる。

負荷分散では、処理を投入する時の計算機の負荷が重要である。しかし、Grid 環境においては、動的に変化する全ての計算機の負荷を、同時に集めることは困難である。また、全ての計算機の負荷情報を同時に集めようとする、情報収集のオーバーヘッドが非常に大きくなる。

Grid 環境における負荷分散では、静的な情報による計算機選択が重要である。静的な情報を用いて処理能力が高い計算機を選択することは、応答時間の短縮へつながらせる。静的な情報とは、CPU、メモリ、ネットワークなどが考えられる。

Grid 環境での負荷分散においては、どのような静的な情報を用いることが有効であるかを検討する必要性がある。また、動的な情報である負荷を考慮することが有効であるかを検討することも重要と考えられる。

### 3 Globus Toolkit

Globus Toolkit とは、Grid 環境を構築するために必要なサービスを提供するツールキットである。Grid 環境にある計算機を利用する場合、どのような計算機が存在するのかを知る必要がある。Globus Toolkit は、Grid 環境に登録された計算機の情報を提供する、MDS(Metacomputing Directory Service) がある。MDS が提供する情報は、計算機名、CPU 処理速度、ノード数、メモリ量などがある。MDS は、GIIS(Grid Index Information Service) と、GRIS(Grid Resource Information Service) からなる。GIIS は、GIIS サーバを用いて情報を集中的に管理するサービスである。一方、GRIS は各計算機ごとの計算機情報を管理する。GIIS の情報更新時間の間隔は GRIS と比べると間隔が大きい。そのため、GIIS は、Grid 環境全体の情報を得ることが出来る一方で、必ず最新の情報であるという保証は無い。負荷分散では、これらの更新時間についても考慮する必要性がある。

Grid 環境において処理の投入には、計算資源割り当て、プロセス生成、状態監視などの機能が必要となる。Globus Toolkit では、GRAM(Globus Resource Allocation Manager) を用いる事により、処理の投入に必要なサービスを提供する。

### 4 負荷分散機能

本研究で実装した、負荷分散の処理手順を以下に示す。

1. 情報収集
2. 計算機選択
3. 条件判断
4. 処理投入

#### 4.1 情報収集

GIIS を用いて、負荷分散に必要な計算機の選択基準となる静的な情報を収集する。収集する情報は、静的な情報である、計算機名、CPU 処理速度、ノード数、実メモリ量である、集めた情報を元に計算機リストを作成する。

#### 4.2 計算機選択

処理を投入する計算機の選択によって、応答時間は大きく変化する。静的な情報での処理能力の高い計算

Implementation of Load Balancing Method on the Grid

<sup>†</sup> Daiki Higashihara, Takahiro Koita

<sup>††</sup> Akira Fukuda

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

Kyushu University (<sup>††</sup>)

機を選択することが重要である。静的な情報を用いた計算機選択方式として、CPU 処理速度を選択基準とした CPU 方式と、実メモリ量を選択基準とした Memory 方式を実装した。計算機リストを、選択基準にあわせてソートする。基準の値が同じであった場合には、他の基準により順位を決定する。例えば、CPU 方式の場合、CPU 処理速度の高いものから順の計算機リストを作成される。

#### 4.3 条件判断

計算機は、情報収集後に作成された計算機リストの上位にある計算機から順に選択される。選択された計算機は、応答条件と動的情報による負荷条件を評価し、条件を満たしたものが処理投入対象となる。選択された計算機が条件を満たさない場合は、計算機リストの次の計算機の条件が評価される。

応答条件とは、計算機が応答しているかどうかを判断する条件である。具体的には、ping コマンドの応答の有無により判断する。情報収集で得られる情報は MDS を利用しているが、MDS が更新されるまでには時間がかかるため、更新以降に応答しなくなる計算機が存在することも考えられる。ping コマンドを用いて、選択される計算機が利用可能であるかどうかを確認する。

動的な情報による負荷判断とは、計算機の負荷がある一定のしきい値を超えているかどうかを判断する。負荷の集中を分散するためには、動的な負荷情報を用いることが重要である。しかし、Grid 環境で負荷情報を収集しようとする場合、情報収集オーバーヘッドが大きいことと、静的情報が応答時間に与える影響も大きいと考えられる。そこで、動的な情報である負荷を利用する方式としない方式を実装した。具体的には、前述の CPU 方式と Memory 方式に対して、動的な負荷情報を考慮した CPU-Load 方式と Memory-Load 方式を実装した。動的な情報を考慮しないものは、CPU-noLoad 方式と Memory-noLoad 方式とした。静的な情報と動的な情報を考慮した負荷分散方式である、CPU-Load 方式と Memory-Load 方式の方が、適切な負荷分散が行えると考えられるが、負荷情報収集のオーバーヘッドの影響は大きくなる。比較のため、CPU-noLoad 方式と Memory-noLoad 方式についても実装した。これらの noLoad 方式は、逐一負荷情報の問い合わせを行わないため負荷情報収集のためのオーバーヘッドはない。静的な情報と動的な情報の基準の組み合わせを表 1 に示す。

表 1: 計算機選択基準の組み合わせ

	静的情報	動的情報
CPU-noLoad	CPU	無し
CPU-Load	CPU	有り
Memory-noLoad	Memory	無し
Memory-Load	Memory	有り

#### 4.4 処理投入

選択した計算機への処理の投入は、GRAM を用いる。投入する処理がオプションなどの複数の引数を持つ場合には、処理の流れを全て、シェルスクリプトなどに記述しスクリプトを投入する。また、全ての計算機のホームディレクトリの同じ環境に、投入する処理が必要とするファイルが置いているものとする。処理が投入された計算機は、処理が終了するまでは、次の処理は投入されない。

#### 5 ゲノムアプリケーションへの応用

Grid 環境へ投入する処理として、ゲノムアプリケーションなど、応答時間に非常に時間のかかるものが挙げられる。本研究では、ゲノムアプリケーションである HMMER[3] に用意されている hmmsearch を対象とした。

#### 6 まとめと今後の課題

静的な情報と動的な情報を利用した負荷分散機能を、Grid 環境を対象として実装した。動的な負荷情報の影響を検討するため、各情報の利用の有無による複数の負荷分散方式を検討した。実装した負荷分散方式をゲノムアプリケーション HMMER に応用し、有効性を検証した。

今後の課題としては、実際の Grid 環境における性能評価と、より少ないオーバーヘッドで効率の良い負荷分散機能の実現である。

#### 参考文献

- [1] The Globus Project, <http://www.globus.org/>.
- [2] 山井 成良, 若林 進, 下篠 真司, 宮原 秀夫, UNIX におけるコマンド単位の負荷分散機能の設計と実装, 電子情報通信学会論文誌 D-I, Vol, J77-D-I, No.7, pp.483-492, (1994).
- [3] HMMER, Washington University in St.Louis, <http://hmmer.wustl.edu/>.