

ゲノムアプリケーション InterProScan のスケジューリング機能の検討

井上 悠佑[†], 小坂 隆浩[†], 福田 晃^{††}

InterProScan は有用なゲノムアプリケーションとして広く利用されているが, 非常に多くの実行時間を要する. InterProScan の実行時間を短縮するために, Grid 環境のような大規模な計算機環境の利用が考えられる. しかし, InterProScan には, 単純なスケジューリング機能しか実装されておらず, そのままでは Grid 環境の大量の計算機を有効に利用することができない. InterProScan を Grid 環境で実行するには新たなスケジューリング機能が必要となる. 本研究では, Grid 環境を対象に InterProScan スケジューリング機能を実装し, 性能評価を行う.

Performance Evaluation of Scheduling Schemes for InterProScan

Yusuke Inoue[†], Takahiro Koita[†], and Akira Fukuda^{††}

This paper presents the implementation details of the scheduling schemes for genomic application InterProScan. Current scheduling scheme of the InterProScan allocates InterProcess to fixed computers on the Grid. However, this allocation cannot use the resources on the Grid. Several scheduling schemes are implemented and evaluated on OBIGrid. From the results, the best scheduling scheme can execute the InterProScan efficiently on the Grid environment.

1 はじめに

ゲノムアプリケーション InterProScan¹⁾ は, EBI(European Bioinformatics Institute)によって開発されたゲノムアプリケーションである. InterProScan は, 蛋白質シーケンスを複数のデータベースに対し, 複数の解析ツールを実行することにより解析処理を行う. 複数の解析処理を同時に行うので, 単一の解析処理では得られない解析結果が得られることも多く, 有用なゲノムアプリケーションとして広く利用されている. しかし, 複数の解析処理が同時に実行されるため, 単一の解析処理に比べ非常に多くの実行時間を要する. InterProScan には単純な分散実行方式が実装されている. InterProScan の分散実行方式で用いられているスケジューリング機能は, インストール時に指定した計算機のみ, インストール時に指定した解析処理を, 固定的に割り当てる. このスケジューリング機能は, 複数の InterProScan を同時に実行する場合や, 多くの計算機が利用可能な場合には, 計算機を有効に活用できない. しかし, InterProScan により新たなデータベースを構築する場合や複数の利用者がある場合も多く, 従来以上の処理能力をもつ計算

機環境と共に, 多数の計算機を有効に活用するスケジューリング機能が必要とされる.

従来以上の処理能力を提供可能な計算機環境として Grid 環境があげられる. Grid とは, ネットワークに接続された様々な計算機資源を利用可能にするインフラストラクチャである. Grid 環境とは, 広域ネットワーク上の計算機資源を多数利用し, PC クラスタ以上の計算能力や大きな記憶領域を有する大規模な計算機環境である. Grid 環境の特徴として, 多数のユーザが多数の計算機を利用することがあげられる. ユーザは多数の計算機を利用できるが, 他のユーザの処理が実行される可能性がある. 他のユーザの処理が実行される場合, 計算機の負荷や処理能力などの情報を考慮しないと, 特定の計算機への処理の集中により応答時間の増加につながる. また, 実行時間のかかる処理を処理能力の高い計算機で処理するためにあらかじめアプリケーションに関する情報についても考慮する必要がある. Grid 環境で InterProScan を実行するには, アプリケーションや計算機の情報 considering, 適切に解析処理を割り当てるスケジューリング機能が必要である.

本研究では, Grid 環境で InterProScan を実行するためのスケジューリング機能を実装し, 有効性を評価する. 評価環境としては, バイオインフォマティ

[†] 大阪産業大学
Osaka Sangyo University

^{††} 九州大学
Kyushu University

クスのための実験的 Grid 環境である OBIGrid³⁾ (Open BioInformatics Grid) を用いる。

2 InterProScan

InterProScan は EBI によって開発されたゲノムアプリケーションである。InterProScan は、既存の解析ツールやデータベースを複数組み合わせることで解析処理を行う統合ゲノムアプリケーションである。InterProScan は、複数の解析ツールと複数のデータベースを組み合わせて蛋白質シーケンスの解析処理する点に特徴がある。表 1 に InterProScan の解析ツールとデータベースの組み合わせを示す。

InterProScan は、解析ツールとデータベースの組み合わせにより、7つの解析処理から構成されている。複数の解析処理を行うことで、解析結果を比較したり解析漏れを防ぐことができる。InterProScan は蛋白質シーケンスの解析を、解析ツールとデータベース・解析命令を記述したスクリプトなどで実行する。図 2 に InterProScan の構成を示す。図 2 に示すように InterProScan は蛋白質シーケンスを複数の方法で同時に解析処理することが可能である。各解析処理は独立しており、全ての解析処理が終了すると解析結果は 1 つにまとめられ TXT 形式・XML 形式・HTML 形式などで出力される。

既存の InterProScan には各解析処理を同時に実行するためのスケジューリング機能が実装されている。このスケジューリング機能による解析処理の割り当ては、インストール時に決定され、以後変化しない。インストール時、1 つ解析処理に対して 1 台の計算機を指定し、固定的に割り当てが行われる。例えば、解析ツール hmmpfam とデータベース PFAM による解析処理をある計算機 X でおこなうように指定した場合、解析ツール hmmpfam とデータベース PFAM による解析処理はすべて計算機 X で行われる。既に計算機 X で解析処理が行われている場合や、他により高い処理能力をもつ計算機がある場合にも、解析ツール hmmpfam とデータベース PFAM による解析処理は計算機 X に割り当てられる。つまり、InterProScan は 7 つの解析処理から構成され、最大でも 7 台の計算機しか利用されることはないため、多数の計算機が

表 1: InterProScan の解析ツールとデータベース

解析ツール	データベース
ScanRegExp	PROSITE patterns
pfscan	PROSITE profile
FingerPRINTSscan	PRINTS
hmmpfan	PFAM
Blast package	PRODOM
hmmpfam	SMART
hmmpfam	TIGFAMs

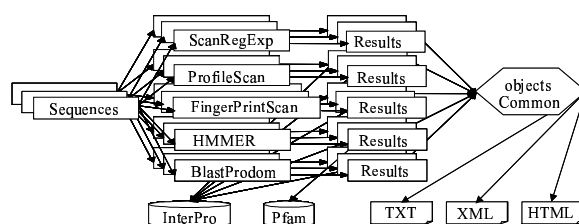


図 1: InterProScan の構成

あっても有効に活用できないという問題点がある。

また、スケジューリング機能以外にも、実行方式とファイル管理に関する問題点がある。InterProScan では、各解析処理の遠隔実行を rsh コマンドによりおこなっているため、Grid 環境のような多数の計算機からなる計算機環境には適さない。また、ファイルはすべて NFS により共有されていることを前提としているため、Grid 環境ではそのまま利用できない。我々は、これまで InterProScan の実行支援機能として、実行方式とファイル管理に関する機能を実装し、InterProScan の Grid 環境での実行を実現した⁴⁾。しかし、Grid 環境の計算機を有効に活用し、InterProScan を効率良く実行するためにはさらにスケジューリング機能の実装が重要である。

3 予備評価

InterProScan のスケジューリング機能を実装するために、まず InterProScan の各解析処理の基本的な特徴を評価する。ここでは、各解析処理の実行時間を計測し、全実行時間中に占める割合を調

表 2: 各解析処理の実行時間と割合

解析ツール	データベース	実行時間 (秒)	割合 (%)
ScanRegExp	PROSITE patterns	34.2	8.6
pfscan	profile	29.2	7.3
FingerPRINTScan	PRINTS	29.8	7.5
hmmpfam	PFAM	228.2	57.4
Blast package	PRODOM	29.8	7.5
hmmpfam	SMART	29.8	7.5
hmmpfam	TIGFAMs	16.6	4.1

べる。そして、評価に用いる OBIGrid での各計算機の InterProScan 実行時間を調べる。

3.1 実行時間の割合

InterProScan は、複数の独立実行可能な解析処理からなる。これらの解析処理の実行時間が大きく異なる場合、アプリケーションの情報として利用することにより効率良く InterProScan を実行できる可能性がある。ここでは、各解析処理の計算時間と InterProScan 全実行時間に占める割合を調べた。評価には、1 台の計算機 (Celeron, 1.3GHz) を用いた。各解析処理の実行時間と全体に占める割合を表 2 に示す。

表 2 より、解析ツール hmmpfam とデータベース PFAM の組み合わせによる解析処理が、InterProScan 全体の実行時間の 50%以上を占めた。それに対して、他の解析処理はいずれも 10%以下である。すべての解析処理を同時に同じ性能の計算機で実行した場合、InterProScan の実行時間は、解析ツール hmmpfam とデータベース PFAM の組み合わせによる解析処理の実行時間となる。解析ツール hmmpfam とデータベース PFAM の組み合わせによる解析処理を、処理能力の高い計算機で実行することにより、実行時間を短縮できる可能性があることがわかる。

3.2 計算機の性能

次に、計算機の処理能力に関して予備評価をおこなった。OBIGrid のうち大阪産業大学・理化学研究所・同志社大学・東京工業大学の 4 サイトの計算機を対象とし評価をおこなった。OBIGrid と

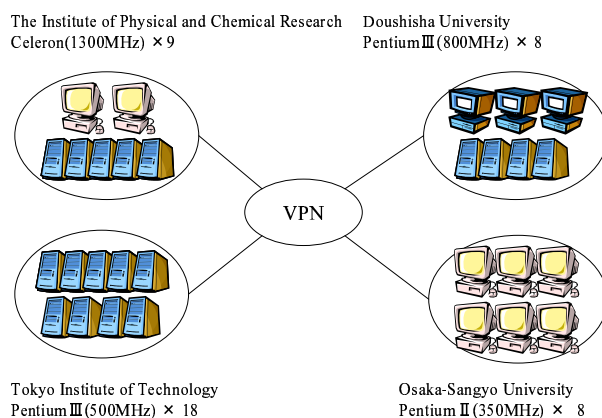


図 2: OBIGrid(評価環境)

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、各サイトが計算機を提供して構築された Grid 環境である。2003 年 7 月現在、OBIGrid の参加サイト数は 26、接続計算機数は約 300 台で各サイトを VPN で接続し Grid 環境を構築している。評価環境を図 2 に示す。各サイトには、それぞれ同じ処理能力をもつ計算機が複数台ある。各サイトの計算機 1 台による InterProScan の実行時間を表 3 に示す。

表 3 より、理化学研究所の計算機の InterProScan 実行時間が他の計算機の実行時間より短く、高い処理能力をもつことが分かる。前節の予備評価結果を考慮すると、解析ツール hmmpfam とデータベース PFAM の組み合わせによる解析処理をこれらの

表 3: 各サイトの平均実行時間

サイト	平均実行時間 (秒)
Osaka Sangyo Univ.	911.6
Doshisha Univ.	471
Institute of Physic.	415
Tokyo Institute of Tech.	503.4

計算機により処理させることにより, InterProScan の実行時間は短くなると考えられる. 逆に, 大阪産業大学の計算機で, 解析ツール hmmpfam とデータベース PFAM の組み合わせによる解析処理をおこなうと, 実行時間は大幅に増加すると考えられる.

4 実装

InterProScan を Grid 環境で実行するためのスケジューリング機能の概要について述べる. ここでは, Fixed, Random, Load, Adaptive 方式の 4 つのスケジューリング方式を実装した.

4.1 Fixed 方式

Fixed 方式は InterProScan に実装されている既存のスケジューリング方式である. Fixed 方式では, インストール時に解析処理とそれを実行する計算機を指定する. 指定された計算機は, 固定的に同じ解析処理が割り当てられる. ここでは, 1 番平均実行時間が短かった東京工業大学の計算機 7 台をインストール時に指定した. Grid 環境内で 1 つだけしか InterProScan が実行されない場合, この方式の応答時間は最も短くなると考えられるが, 複数の InterProScan が実行される場合, 同じ解析処理が 1 つの計算機に集中するために大幅に応答時間が増加することが考えられる. また, 他に利用可能計算機があっても, Fixed 方式では解析処理は割り当てられない.

4.2 Random 方式

Random 方式は, 全ての計算機を対象に解析処理をランダムで割り当てる. 全ての計算機が割り当て対象となるため, Fixed 方式よりも利用可能計算機数は増える. また, 解析処理は全ての計算

機に割り当てられるため, Fixed 方式より処理が集中する可能性は低くなる. しかし, アプリケーションの情報や計算機情報を全く考慮しないため, 時間のかかる解析処理を処理能力の低い計算機に割り当てる場合もおこるため, 応答時間は長くなる可能性がある.

4.3 Load 方式

Load 方式は, 3.2 節で得られた各計算機の平均実行時間 (T_{exec}) と CPU 使用率 (U_{cpu}) を元に, 各計算機の予想実行時間 ($T_{estimate}$) を算出しスケジューリングを行う. 各計算機の予想実行時間 ($T_{estimate}$) は次の式で与えられる.

$$T_{estimate} = T_{exec} \times \left(\frac{1}{1 - U_{cpu}} \right) \quad (\text{但し, } U_{cpu} < 1)$$

予想実行時間は, 計算機の処理能力と, 新たに投入される解析処理が利用できると期待される CPU 利用率より与えられる. 例えば, CPU 利用率が 0 で, 既に得られている平均実行時間が 100 の場合, 予想実行時間は 100 となる. また, CPU 利用率が 0.99 の場合, 予想実行時間は 10000 となる. Load 方式は, 計算機の処理能力と, CPU の利用率という, 計算機情報を考慮した方式となる. CPU 利用率が低く, 処理能力の高い計算機から割り当てが行われるため, Fixed 方式や Random 方式より適切計算機が選択される.

4.4 Adaptive 方式

Adaptive 方式は, 基本的には Load 方式と同じであるが, さらにアプリケーションの情報も考慮して割り当てをおこなう. Adaptive 方式は, 各計算機の平均計算時間と CPU 使用率を元に各計算機の予想実行時間を算出しスケジューリングを行う点は, Load 方式と同じである. Load 方式と異なる点は, Adaptive 方式では, 最も実行時間のかかる解析ツール hmmpfam とデータベース PFAM の組み合わせの解析処理を, 処理能力の高い計算機に優先的に割り当てる. それ以外の解析処理は処理能力の高くない計算機に割り当てる. すなわち, 解析処理を実行時間のかかる処理とかけられない処理に分類し, 計算機も実行時間のかかる処理用と

実行時間のかからない処理用に分類し、それぞれの処理において予想実行時間を算出し、割り当てをおこなう。Adaptive 方式では、アプリケーションの情報をもとに、処理能力の高い計算機を時間のかかる処理が優先的に利用することにより、応答時間の短縮を図る。ここでは、InterProScan 実行時間の最も短かった理化学研究所の計算機 9 台を処理能力の高い計算機とした。

5 性能評価

前章で述べた 4 つのスケジューリング方式を実装し、OBIGrid 環境で評価した。評価環境は、3.2 節と同じである。ランダムに InterProScan を生成し、Grid 環境に投入した。平均投入間隔は指数分布に従う。負荷率を変化させ、InterProScan を 100 個投入したときの平均応答時間を求めた。ここで負荷率とは、平均投入間隔により与えられ、計算機環境内の InterProScan の混雑度を表す指標であり、負荷率 100% は、もうそれ以上 InterProScan を処理することができない状態を表す。例えば、InterProScan を 100 秒で終了できる 2 台の計算機があり、平均投入間隔が 50 秒である場合、負荷率は 100% となる。ここでの負荷率は、3.2 節の結果より算出した。詳細については、文献 4) を参照にされたい。

図 3 に、4 つのスケジューリング方式の負荷率と平均応答時間を示す。既存の InterProScan のスケジューリング機能である Fixed 方式は、負荷率が 30% 以下と低い場合でも応答時間が大きい。これは評価環境の全体の計算機を有効に利用できず、各解析処理を常に同じ計算機で実行したためである。特に、解析ツール hmmpfam とデータベース PFAM の組み合わせの解析処理が 1 台の計算機に集中するため、負荷が増加するに従い応答時間が急激に増加している。

Random 方式は、10% より大きい負荷率では、Fixed 方式よりも応答時間は短い。しかし、負荷率 10% では、わずかではあるが Fixed 方式より応答時間は長い。Random 方式では負荷率が低くても、平均実行時間が遅い計算機に解析ツール hmmpfam とデータベース PFAM の組み合わせの解析処理が割り当てられたり、特定の計算機に処理が集中す

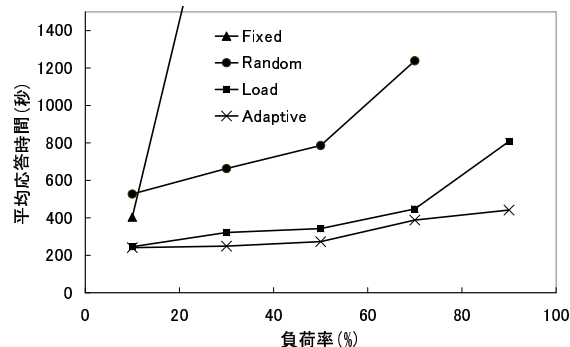


図 3: 負荷率と平均応答時間

る可能性がある。特に実行時間の遅い大阪産業大学の計算機に解析ツール hmmpfam とデータベース PFAM の組み合わせの解析処理が割り当てられると大幅に応答時間が増加することが考えられる。

Load 方式では、すべての負荷率において、Random 方式と Fixed 方式よりも応答時間が短い。また、負荷が高くなった状態でも実行時間の増加が抑えられた。Load 方式は、常に計算機情報を利用することにより、解析処理の割り当てをおこなっているため、すべての計算機が無駄なく有効に利用されているためであると考えられる。

Adaptive 方式は、負荷率が低い場合、応答時間は Load 方式とほとんど変わらなかった。しかし、負荷率が高い場合は Load 方式に比べ応答時間が短い。負荷率が低いときは全ての計算機で解析が行われることは少ないため、Load 方式と応答時間に大きな差は出なかった。負荷率が高い状態では解析ツール hmmpfam とデータベース PFAM の組み合わせの解析を優先的に処理能力の高い計算機に割り当てることにより、応答時間が短くなっている。

次に、負荷率 30% での応答時間の最大・最小・平均と標準偏差を表 4 に示す。Fixed 方式と Random 方式が応答時間のばらつきが大きいのに対し、Load 方式と Adaptive 方式の応答時間のばらつきが小さいことが分かる。特に Adaptive 方式では、標準偏差も平均も最も小さく、アプリケーションの情報と計算機の情報を利用することにより、安定した

表 4: 各方式の平均/最大/最小応答時間と標準偏差 (負荷率 30%)

方式	平均 (秒)	最大 (秒)	最小 (秒)	標準偏差
Fixed	2167	3574	371	1337
Random	586	2575	220	549
Load	322	614	208	125
Adaptive	225	243	207	12

応答時間を実現していることがわかる。

6 まとめと今後の課題

InterProScan のスケジューリング機能を実装し、OBIGrid にて評価した。Fixed・Random・Load・Adaptive の 4 方式を評価したところ、すべての負荷率に対して Adaptive 方式が最も応答時間が短かった。Adaptive 方式は、アプリケーションの情報と計算機の情報を使うことにより、適切な割り当てを実現することにより、Grid 環境の計算機を有効利用する方式であると考えられる。

今後の課題として、InterProScan の実行時間をより短縮するために各解析処理のさらなる実行時間短縮があげられる。特に hmmpfam の並列化による実行時間の短縮を検討中である。また、Adaptive 方式は、計算機情報として CPU 使用率を取り上げた。さらに他の指標を用いることも考えられる。

参考文献

- 1) EBI, <http://www.ebi.ac.uk/>.
- 2) Globus Project, <http://www.globus.org/>.
- 3) OBIGrid, <http://www.obigrid.org/>.
- 4) Takahiro Koita, Yusuke Inoue and Akira Fukuda, “ Implementation and Evaluation of Resource Allocation for a Genomic Application Program on the Grid,” Proc. of the IASTED Int’l Conf. on Parallel and Distributed Computing and Networks(AI2003), pp.524-528, 2003.