

グリッドスケジューラのための遺伝的アルゴリズムの検討
 斉藤 宏樹

1 はじめに

現在、グリッドスケジューリングに関する研究が盛んに行われている¹⁾。グリッドとは、遠隔地に配置された複数の計算資源を結びつける基盤であり、グリッドスケジューリングとはグリッド上で、複数の資源を有効利用して大規模計算あるいは遠隔協調計算が最小の時間で実行できるように、ジョブやタスクを適切な計算資源へ割り当てることである。スケジューリング問題は、多変数の最適化問題と定義される。本研究では最適化手法の一つである遺伝的アルゴリズム (Genetic Algorithm : GA) および分散遺伝的アルゴリズム (Distributed Genetic Algorithm : DGA) を、作成したグリッドスケジューリング問題に適用し、検討を行う。

2 グリッドにおけるスケジューリング問題

2.1 グリッドにおけるスケジューリング問題の定義

グリッドにおけるスケジューリング問題とは、利用できる計算機の数や性能、負荷状況が動的に変化する環境で、対象問題 (分散アプリケーション) の実行時間を最小にする問題である。ネットワーク情報や計算機の性能、アーキテクチャの情報など、グリッドにおける計算資源の静的な情報や、計算機の稼働台数に関する情報やそれら計算機のバンド幅、CPU やメモリなどの負荷情報など、動的な情報を利用することで、分散アプリケーションの解候補が生成される。生成された解候補は、分散アプリケーションの実行時間を見積もるために、実行モデルにあてはめられる。そのためスケジューリング問題は、スケジューラと分散アプリケーション、そして実行モデルがそれぞれ必要となる。

2.2 分散アプリケーション

本研究で定義した分散アプリケーションは、ジョブ分配問題である。ジョブ分配問題とは、あるジョブを、グリッドの計算資源の性能に合わせて複数のタスクに分割し、グリッド計算資源に分散させた場合のジョブの最短実行時間を求める問題である。問題では、一定サイズのタスクを処理すると、同期をとって通信を行うものとする。Fig. 1 に分散実行の概要を示す。なお、タスクを割り当てる計算機も自由に選択できるものとする。

2.3 実行モデルの作成

グリッドは従来の並列計算機によるホモな環境と異なり、ヘテロな環境であるため、分散実行時の実行時間を

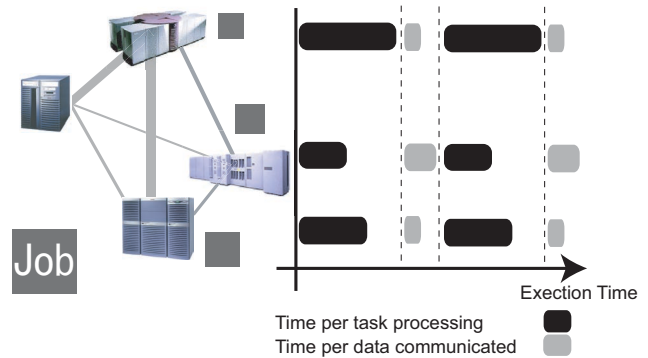


Fig. 1 Scheduling Problem

見積もることが極めて難しく、現在も正確な実行モデルが作成されていない。本研究ではホモな環境における ScaLAPACK の計算ルーチンの実行モデル²⁾を参考にして、対象とする分散アプリケーションの計算時間や通信時間、分配によるオーバーヘッドの時間を考慮した実行モデルを作成した。式 (1) にその式を定義する。なお、 N はジョブのサイズであり、 i は計算機識別の ID、 N_i は計算機 i に割り当てられたタスクのサイズ、 M は使用計算機数である。

$$\begin{aligned}
 T(N, machine_list) = & \max_{i=1}^M \{ C_t(N_i) t_t(perf.cpu_i) \} \\
 & + \max_{i=1}^M \{ C_c(N_i) t_c(perf.network_i) \} \\
 & + T_o(M) \tag{1}
 \end{aligned}$$

3 グリッドスケジューラのための遺伝的アルゴリズムの検討

3.1 遺伝的アルゴリズムの適用

近年、シミュレーテッドアニーリング (Simulated Annealing : SA) や欲張り法によるスケジューラの研究が行われている。しかし、SA ではスケジューラとしてのオーバーヘッドが発生し、欲張り法では得られる解の精度に問題がある。そのため、本研究では、スケジューラに GA を適用する方法について検討を行う。

GA では、アプリケーションの実行時間を最小にするような計算機の組み合わせと、割り当てるタスクのサイズを個体で表現する。グリッドで使用できる計算機 1 台を 1 設計変数とし、1 設計変数を 5 ビットの遺伝子で表現する。5 ビットのうち、先頭 1 ビットを 0, 1 のフラグとし、残り 4 ビットを 0, 1 のグレイコードとする。フラグが 1 であればその計算機を使用することを示し、0

であればその計算機を使用しないことを意味する．そして4ビットのグレイコードは，計算機に割り当てられたタスクのサイズを示す．Fig. 2 にその例を示す．なお評価関数は，作成した式 (1) の実行モデルに，見積もったコストを代入した式を用いる．

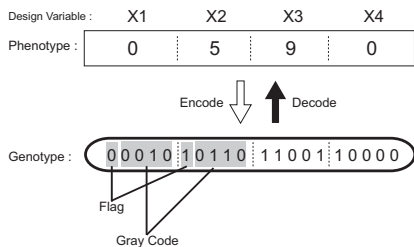


Fig. 2 Expression and Cording of Individual

3.2 ローカルサーチ手法の組み込み

グリッドにおけるスケジューリング問題は，大域的最適化問題であり探索領域が広いため，最適解を得ることが難しい．また動的に資源情報が変化するため，時間的制約も強い．GA は局所的探索能力が弱く，膨大な評価計算を繰り返すため最適解を得るのに時間を要する．この問題を解決する手段として，ローカルサーチ (Local Search) を GA に適用した方法が挙げられる．ローカルサーチとは，与えられた解をより良好な解に改善する処理である．本研究では SA によるローカルサーチを GA に組み込み，より少ない評価計算回数で最適解を得ることを目標とする．

4 数値実験

4.1 スケジューリング問題の設定

想定した利用可能なグリッドの計算資源を，Fig. 3 に示す．Fig. 3 に示す CPU 性能や計算機間のネットワーク性能，利用可能な計算機台数などの静的な情報がスケジューラに与えられるものとする．また計算コストが $O(N^2)$ であるのに対して，通信コストが $O(N)$ となるアプリケーションを対象とし，分散によるオーバーヘッドのコストを $O(N^2)$ と設定した．

4.2 グリッド・スケジューリング問題を用いた評価

作成したグリッドスケジューラ問題に，ローカルサーチを組み込んだ SGA, DGA を 2000 世代，20 回試行をする．その評価関数値の中央値を，Fig. 4(a) に，得られた最良の解のスケジュールを Fig. 4(a) に示す．GA, DGA のパラメータは，交叉には一点交叉，選択にはトーナメント選択を適用し，総個体数 160，遺伝子長 200，交叉率 0.6，トーナメントサイズ 4，突然変異率 0.005，移住率 0.5，移住間隔 5，島数 4 である．SA のパラメータは Table 1 に示す．Fig. 4(a) の実験結果から，ローカ

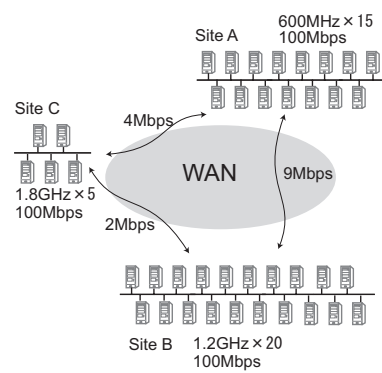
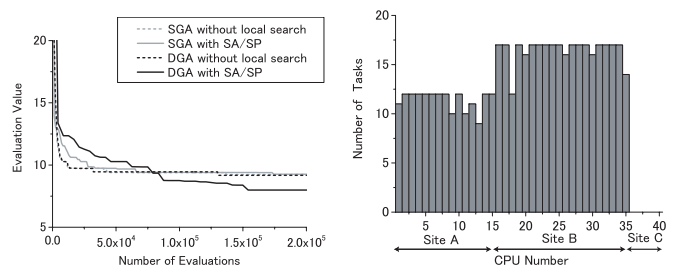


Fig. 3 Image of Testbed Resources

ルサーチに SA を用いた DGA が最も解探索が良好であることがわかる．

Table 1 SA parameters

最高温度	15.22	アニーリング数	1000
最低温度	0.002	クーリング関数	$a(T) = aT$
クーリング率	0.914	クーリング間隔	10



(a) History of the evaluation value

(b) Best Schedule of experiment

Fig. 4 Result of Experiment

5 まとめ

本研究では，グリッドにおけるスケジューリング問題を作成し，GA による解探索の検討を行った．その結果，GA のみでは局所解へ陥る可能性があることがわかった．そして，ローカルサーチを GA に組み込むことで，局所解への収束を防げることが本研究により確認できた．今後は，SA を組み込んだ GA によるスケジューラで，動的な情報を考慮した実問題を解くことを目標とする．

参考文献

- Asim YarKhan, Jack J. Dongarra. *Experiments with Scheduling Using Simulated Annealing in Grid Environment*. Workshop on Grid Computing (submitted), June 7, 2002. 2
- Antoine Petit, Susan Blackford, Jack Dongarra. *Numerical libraries and the Grid*. The International Journal of High Performance Computing Applications, 2001.