A-017

グリッド環境における完全交換に対するスケジューリングアルゴリズム
Scheduling Algorithms for the Total Exchange on the Grid Environment

中平 健太 † Kenta Nakahira
藤原 暁宏 † Akihiro Fujiwara

1. はじめに

近年，地理的に分散した計算資源を利用した大規模広域分散処理 (グリッド) 環境が注目を集めている．本研究では，このグリッド環境において，基本的通信処理の 1 つである完全交換について，タスク分割を行うことによって効率的に完全交換を行なうスケジューリングアルゴリズムの提案を行った．また，提案したアルゴリズムと既知の完全交換アルゴリズムをグリッド環境をシミュレートするソフトウェアである SimGrid[2] 上に実装し，アルゴリズムの効率の比較を行なった．この実験により，提案したアルゴリズムによって従来のスケジューリングアルゴリズムが苦手とする入力に対して優れたスケジューリングが得られることを示した．

2. 完全交換と通信モデル

完全交換とは，対象となる全てのプロセッサが自分以外の全プロセッサへの送信処理を行う通信操作である．

本研究で用いる通信モデルにおいては，プロセッサ P_i からプロセッサ P_j への m バイトのメッセージの通信を通信イベントと呼び，スタートアップコスト T_{ij} とデータ通信速度 B_{ij} という 2 つの変数を用いて，この通信イベントのコスト C_{ij} を以下のように示す．

$$C_{ij} = T_{ij} + \frac{m}{B_{ij}}$$

また，各プロセッサは同時に 1 つの送信処理と 1 つの受信処理を行うことができるものと仮定する．

上式で与えられる各プロセッサ間の全通信コストを表した行列を通信行列 $\boldsymbol{C}$ とする．ここで $\boldsymbol{C}$ の各行及び各列はそれぞれ対応するプロセッサの送信及び受信コストを表す．よって各行及び各列の和のうち最大のものが全通信処理を実行するために最低限必要な時間であり，これを最短通信時間 t_{lb} と呼ぶ．

3. 既知のスケジューリングアルゴリズム

本研究では，既知の完全交換スケジューリングアルゴリズムと提案する完全交換スケジューリングアルゴリズムの比較を行う．以下では既知の完全交換スケジューリングアルゴリズムについて簡単に説明を行う．

キャタピラ法 [1]: 各 $j(1 \leq j \leq p-1)$ ステップ目において，各プロセッサ $P_i(0 \leq i \leq p-1)$ はプロセッサ $P_{(i+j) \bmod p}$ に送信を行う．

オープンショップ法 [1]: 最も早く送信イベントが開始できるプロセッサから，最も早く受信イベントが実行できるプロセッサに送信を行うようにスケジューリングを行う．全てのイベントが決定するまでこの操作を繰り返す．

静的分割法 [3]: いくつかの通信イベントを予め分割した後にオープンショップ法を適用する．

動的分割法 [3]: オープンショップ法の実行途中に通信待ちのアイドルな時間が発生した場合，その要因となる通信イベントをアイドルな時間を解消するように分割する．

4. 致命的な通信行列

以下のような通信行列を上記の既知の完全交換スケジューリングアルゴリズムの入力とした場合，いずれのアルゴリズムを用いても完了時間が約 $2h$ のスケジューリングしか得られないことを証明した．また，この通信行列に対する最適な完了時間は $\frac{3}{2}h$ であることも示した．本研究では，この通信行列を致命的な通信行列と呼ぶ．

$$\begin{bmatrix} * & h & \delta & \cdots & \delta \\ \delta & * & h & \cdots & \delta \\ \delta & \delta & * & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & h \\ \delta & x & \cdots & x & * \end{bmatrix}, \quad \begin{array}{l} h = (p-2)x \\ \delta \ll h, x \end{array}$$

5. 改良分割法

ここでは，改良分割法という新しいスケジューリングアルゴリズムの提案を行う．改良分割法は，致命的な通信行列に対して，効率のよいスケジューリングの妨げとなる通信処理をあらかじめ細かく分割しておき，動的分割法の実行途中に，可能であればそれらを再結合するという手法を取り入れたアルゴリズムである．

まず最初に，提案アルゴリズムでは，致命的な通信行列の各イベントについて，その大きさが δ である通信イベントの集合 C_{small} と，その大きさが h もしくは x である通信イベントの集合 C_{large} に分割する．本アルゴリズムでは，最初に C_{large} 含まれるイベントについてすべてスケジューリングを行った後に，C_{small} に含まれるイベントについてスケジューリングを行う．これは，小さいイベントは後からまとめてスケジュールした方が全体的な効率の向上に繋がるためである．

次に，C_{large} に属する通信イベントの分割を行うが，コストが h である通信イベントのみを分割するために，X, Y という 2 つのパラメタを用いて分割の閾値を設定する．まず，通信イベントのコストが，

$$Y \times (\text{通信イベントのコストの最大値})$$

より大きな通信イベントを分割候補とする．次に，この分割候補のうち，分割により増加する各行，及び，各列の通信コストの総和が，

$$X \times t_{lb}$$

† 九州工業大学 情報工学部

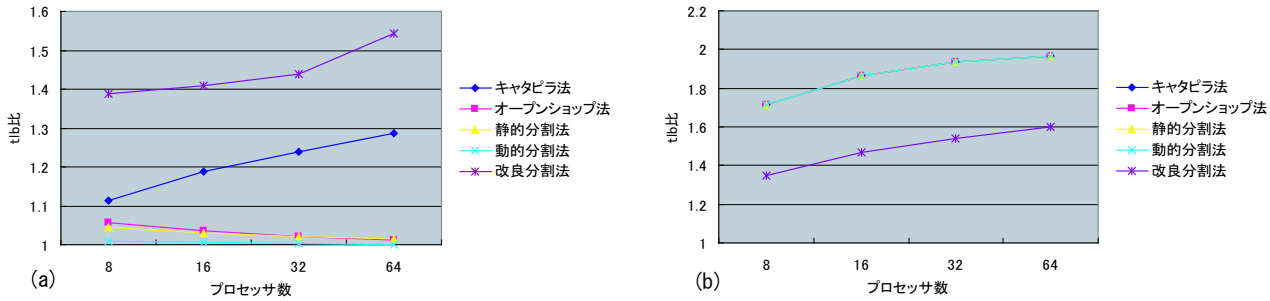

図 1: アルゴリズムの完了時間の比較．(a) ランダムな通信行列，(b) 致命的な通信行列

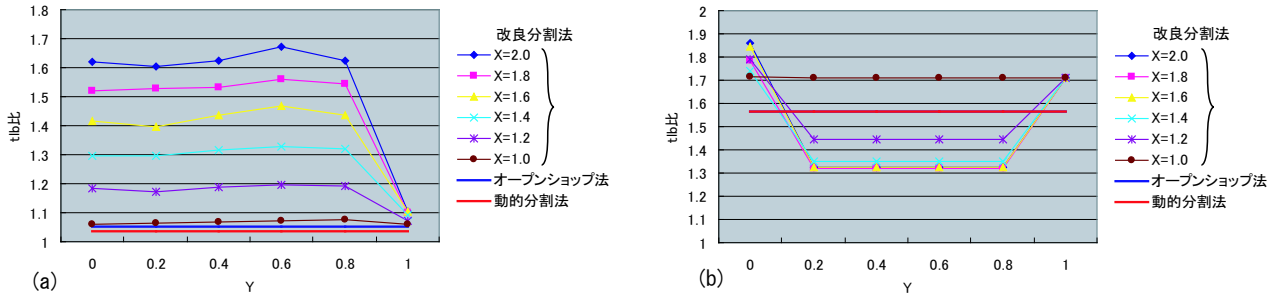

図 2: X,Y の値による完了時間の変化．(a) ランダムな通信行列．(b) 致命的な通信行列

を超えない場合，この分割候補を実際に分割する．この操作を全ての C_{large} に属する通信イベントについて実行する．

次に，この通信イベントを分割した通信行列に対して動的分割法 [3] を適用する．このとき，各プロセッサにおいて同じ宛先プロセッサを持つ通信イベントが連続してスケジューリングされる場合がある．その場合には，それらのメッセージを結合することにより，それらを分割するときに生じた分割オーバヘッドを打ち消すことができる．

最後に C_{small} に含まれるイベントに動的分割方を適用する．

6. 検証実験

本研究ではグリッド環境をシミュレートするソフトウェアである SimGrid[2] を用いて各アルゴリズムを実装し動作速度に関する実験を行った．まず最初に，プロセッサ数を 8 から 64 まで変化させ，ランダムな通信行列と致命的な通信行列に対するスケジューリングアルゴリズムの完了時間の比較を行った．この時，スタートアップコストの割合は 0.1 ～ 50(%) の乱数によって生成した．ランダムな通信行列と致命的な通信行列に対する結果を図 1 に示す．

ランダムな通信行列に対しては動的分割法により一番効率のよいスケジューリングが得られているが，一方，致命的な通信行列を効率的にスケジューリングする改良分割法はランダムな通信行列を苦手とすることが分かった．

同じ条件で，オープンショップ法，動的分割法，改良分割法で完了時間の比較を行った．この際，動的分割法の X を 1.0～2.0,Y を 0～1.0 で変化させた．ランダムな通信行列と致命的な通信行列に対する結果を図 2 示す．

致命的な通信行列に対しては，Y が 0.2～0.8 の範囲の場合にはコスト h の通信イベントのみを分割するので最も効率的となり，X の大部分の範囲において著しい効率の向上がみられる．一方，ランダムな通信行列に対しては，X の値が小さくなるにつれスケジューリング効率は向上し，Y は 1 の場合に既知のアルゴリズムに近いスケジューリング効率を得ている．しかし，この X と Y の組み合わせは致命的な通信行列と相反してしまうので，致命的とランダムの通信行列の両方を効率良くスケジュールできる X と Y の組合せが存在しないことがわかる．

7. まとめ

本研究では，グリッド環境に対する完全交換におけるスケジューリングアルゴリズムを，SimGrid というグリッドシミュレータ上に実装し，各アルゴリズムの比較，及び，検証を行った．一般的な通信行列に対して最も効率的にスケジューリングを行うアルゴリズムは動的分割法であった．しかし，致命的な通信行列が存在し，この致命的な通信行列に対しては，改良分割法が著しく有効であった．しかし，通常のランダムな通信行列に対しては，改良動的分割法は他のアルゴリズムよりもスケジューリング性能は悪くなっている．したがって，致命的な通信行列とランダムな通信行列の両方を効率よくスケジューリングできるアルゴリズムの提案が今後の課題である．

参考文献

[1] P.B.Bhat, V.K.Prasanna, C.S.Raghavendra, *Adaptive Communication Algorithms for Distributed Heterogeneous Systems*, HPDC-7, 1998.

[2] H. Casanova. *Simgrid: a Toolkit the Simulation of Application Scheduling*, CCGrid, 2001.

[3] Y.Jinno, M.Ito and A.Fujiwara, Efficient scheduling algorithms for total exchange on GRID environment, PDPTA, Vol.1, 2003.