通信時間を削減するためのタスク複製の手法の提案

6U−5

高木 秀樹† 李 鼎超‡ 石井 直宏†

†名古屋工業大学知能情報システム学科 ‡名古屋工業大学情報処理教育センター

1 はじめに

分散メモリ並列 アーキテクチャ での並列プログラムの実行効率は，タスクスケジューリング時に使用される戦略に大きく依存する．

タスクスケジューリングの問題にはタスクの複製を許すか，許さないかの 2 つの見解が存在する．タスクの複製を行う場合，複製を行うタスクの選択，複製されるタスクを割り付けるための適切なスケジューリングホールの発見などを行う必要がある．そのため，タスクの複製を行うことによりスケジュールの問題はより複雑になる．しかしながら，タスクの複製のあるなしに関わらず，タスクスケジューリングの問題は NP 困難な問題である．したがって，効率的な方法は積極的に取り入れるべきである．

タスク複製の方針では，次の 3 つの問題を考慮する必要がある：1) いつ複製を行なうのか，2) どのタスクを複製するのか，3) どこに複製するのか．以上の 3 つの問題の解決により新たな複製の方針が決定される．

本稿では，いくつかのよく知られた優先度ベースのスケジューリングアルゴリズムの性能を時間複雑性の増加を伴うことなく改良できる複製の手法の提案を行う．

2 問題の定式化

2.1 ターゲットマシン

本稿では，任意の方法で相互結合された均一のプロセッサにより構成されるアーキテクチャをターゲットアーキテクチャとする．プロセッサはそれ自身のプライベートメモリを持ち，共有の広域メモリを持たない．プロセッサ間の通信は明確に直接メッセージを送ることで行われる．プロセッサがお互いに通信可能であれば，2 つのプロセッサ間には重みを持つエッジが存在する．タスク間の通信遅延は，送られるメッセージの数と，それぞれのタスクを実行するプロセッサ間の転送率によって決定される．また，同じプロセッサ上で実行されるタスク間でデータが必要とされる時には，データはローカルメモリから読み込まれると仮定する．そのため，同じプロセッサ間に割り付けられたタスク間の通信コストは 0 である．

"Using Task Duplication to Reduce Communication Delay"
Hideki Takagi, Dingchao Li,Naohiro Ishii
Nagoya Institute of Technology, Gokiso-cho, Syowa-k,
Nagoya 466-8555, Japan

2.2 タスクグラフ

本稿では，タスクグラフと呼ばれる有向無サイクルグラフを扱い，これを用いて並列プログラムを表現する．タスクグラフはノードとエッジによってタスクとタスク間の先行制約および順序関係を表現する．

$G = (\Gamma, A, \mu, \eta)$ はタスクグラフを表す．Γ はタスク $T_i (i = 1, \cdots, n)$ の有限の集合を，A はアーク $arc(T_i, T_j)(i, j = 1, \cdots, n, i \neq j)$ の有限の集合を表す．関数 $\mu(T_i)$ は T_i の実行時間を，関数 $\eta(T_i, T_j)$ はタスク T_i の実行終了時に T_i から T_j に送られるメッセージの数を返す関数である．$arc(T_i, T_j)$ は，T_i の実行が完了し T_i からのメッセージが T_j に到着するまで T_j を実行開始できないことを意味する．この時，T_i を T_j の直接先行タスクと呼び，T_j を T_i の直接後続タスクと呼ぶ．プロセッサ間の通信は，直接先行タスク T_i のメッセージを直接後続タスク T_j が必要とし，かつ T_i, T_j がそれぞれ異なるプロセッサに割り付けられた場合に発生する．

3 タスクの複製

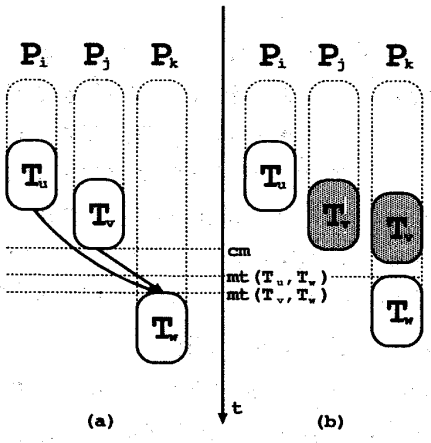

図 1(a) タスク複製前．(b) タスク複製後

本節で使用する表記を説明する．$P(T)$ は，タスク T を実行するプロセッサを，$\lambda(P_i, P_j)$ はプロセッサ P_i から P_j にメッセージを送る時間を表す．$st(P, T), ct(P, T)$ はそれぞれ P 上で T を実行した時の T の実行開始時間と実行完了時刻を表す．cm はイベントクロックのカレント時間を，T_{ready}, P_{idle} はそれぞれ cm 時に利用可能なプロセッサとタスクの集合を表す．また，$mt(T_i, T_j)$ をプロセッサ $P(T_j)$ にプロセッサ $P(T_i)$ からメッセージが到着する時間とし，以下の式で計算する．

$$mt(T_i, T_j) = ct(T_i) + \lambda(P(T_i), P(T_j))$$

タスクの複製は次の手順で行なう．

1. 優先度ベースのスケジュールアルゴリズムに従い，cm 時に最も優先度の高いタスクとプロセッサの組 (T, P) を選択する．
2. T の直接先行タスク T_{pre} に対して $mt(T_i, T)(T_i \in T_{pre})$ を計算し，T に最も遅く到着するメッセージを送るタスク T' を求める．
3. $P(T) = P(T')$ の場合，複製する余地がないため複製は行なわない．$P(T) \neq P(T')$ の時，T' の P 上での実行終了時刻 $ct(T', P)$ を求める．T に2番目に遅く到着するメッセージを送るタスクを T'' とすると，T' を P に複製した場合のタスク T の実行開始時間 $st\_dup(T, P)$ は以下の式で与えられる．

$$st_{dup}(T, P) = \max\{mt(T'', T), ct(T', P(T')), cm\}$$

4. 複製を行なわない場合の実行開始時間 $st(T, P)$ と $st\_dup(T, P)$ を比較する．$st(T, P)$ の方が小さい場合には，複製は行なわない．$st\_dup(T, P)$ の方が小さい場合は，タスクの複製を行ない，$st_{dup}(T, P)$ を実行開始時刻とする．

図1にタスクの複製の例を示す．タスク T_w はタスク T_u と T_v からメッセージを受け取る．T_v からメッセージが到着する時間 $mt(T_v, T_w)$ を T_u からメッセージが到着する時間 $mt(T_u, T_w)$ と比較する．この場合 $mt(T_v, T_w)$ の方が遅いため，T_v を複製する．この結果，T_w の新たな実行開始時間は $mt(T_u, T_w)$ となり，T_w の実行開始時間はタスク複製を行なわない場合よりも早くなる．

4　評価

本節では，逐次計算機上で実験を行なった結果を示す．本論文で提案したタスク複製の手法をETF法[1]及びDL法[1]に対して適応した結果を示し，タスク複製の手法の有効性を示す．

DAGの粒度はアルゴリズムの性能を評価する場合に重要な要素である．粒度の計算は[2]で与えられた粒度の定義を採用する．粒度は通信時間と実行時間の比率を表す尺度であり，小さな数字を取るときほど，通信時間が実行時間に比べ大きいことを表している．

本実験では，ターゲットマシンとして均質の8個のプロセッサにより構成されるハイパーキューブを採用した．対象となる並列プログラムとしてタスク数が50個のランダムに生成された4000個のタスクグラフを用いた．

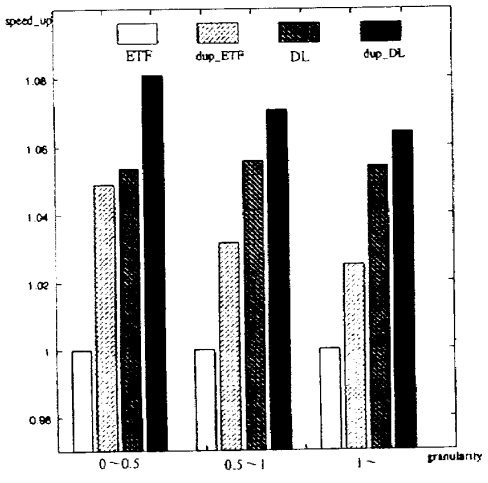

図2 評価結果

図2にETF法を1とした場合の，複製を伴ったETF法，DL法，複製を伴ったDL法のそれぞれの速度向上率を示した．左からそれぞれ，粒度が0～0.5の場合，0.5～1の場合，そして1以上の場合に対する結果を示した．この結果より，粒度が小さい場合，つまり通信時間が実行時間に比べて大きい場合に本手法の効果があることがわかる．これは，通信時間が大きい場合に，複製するタスクを埋めることができるホールが多いことによると考えられる．

また，DL法においてETF法より効果が小さいのは，DL法がタスクを前詰めにスケジューリングする手法であるため，タスクの複製を行なう余地が少ないことが理由であると考えられる．

5　まとめ

本稿では，優先度ベースのスケジューリングアルゴリズムに対して適応できるタスク複製の手法を提案し，その評価を行なった．ETF法に対しては4%，DL法に関しては2%程度性能が向上した．また，複製を伴うDL法はETF法にたいして7～8%程度速度の向上が得られた．今後の課題としては，他の優先度ベースのアルゴリズムに対しても実験を行なうことなどがある．

参考文献

[1] JING-JANG HWANG,YUAN-CHIEH CHOW, "SCHEDULING PRECEDENCE GRAPHS IN SYSTEMS WITH INTERPROCESSOR COMMUNICATION TIMES" SIAM J COMPUT.vol 18,No.2,pp.244-257

[2] A-A-Khan,C-L-McCreary and M-S-Jones "A Comparison of Multiprocessor Scheduling Heuristics" *International Conference on Parallel Processing* ,II-243-250(1994)