

## 通信量を考慮したタスクスケジューリング法の提案と評価

4T-5

小林 真也 木村 春彦 武部 幹

金沢大学工学部電気・情報工学科

## 1. まえがき

マルチプロセッサシステムにおけるタスク割当てに対する戦略に、スケジューリング問題の応用として取り扱う方法がある。しかしながら、CP法<sup>(1)</sup>、CP/MISF法<sup>(2)</sup>、DF/IHS法<sup>(3)</sup>などのアルゴリズムではプロセッサ間の通信時間を取り扱うことができない。

本論文では、プロセッサ間の通信時間を考慮したタスク割当て法を提案し、従来方式との比較を行いその有効性を示す。

## 2. 問題と定義

マルチプロセッサシステムにおけるタスク割当て問題は、処理能力の等しい $m$ 台のプロセッサで、 $n$ 個のタスクの処理を行い、タスク全ての処理が完了するまでの時間を最小とする、タスクの処理順序と実行プロセッサを決定する問題である。

各タスクはそれぞれ固有の処理時間 $P_k$  ( $1 \leq k \leq n$ )を持つ。また、タスク集合は半順序集合をなす。タスク $T_j$ とタスク $T_k$ が半順序 $T_j \ll T_k$ を満たすとき、 $T_j$ を $T_k$ の先行タスク、 $T_k$ を $T_j$ の後続タスクと呼ぶ。 $T_j \ll T_k$ であれば、 $T_j$ の終了時刻 $E_j$ と $T_k$ の開始時刻 $S_k$ は $E_j \leq S_k$ でなければならない。さらに、 $T_j \ll T_i \ll T_k$ となるタスク $T_i$ が存在しないとき、 $T_j$ と $T_k$ は依存関係にあるといい、 $T_j$ を $T_k$ の直前タスク、 $T_k$ を $T_j$ の直後タスクと呼ぶ。

タスクを頂点とし、依存関係を有向の弧で表すと、タスク間の依存関係は無サイクル有向グラフで表わせる。ここで、弧の始点に対するタスクが、終点に対するタスクの直前タスクとなる。この有向グラフをタスクグラフと呼ぶ。また、直前タスクの存在しないタスクを開始タスク、直後タスクの存在しないタスクを終了タスクと呼ぶ。直前タスクが他のプロセッサで実行されると、プロセッサ間で通信が行われる。送受信されるデータ量はタスクの組毎に固有である。

## 3. 提案方式

提案方式の基本戦略は通信を行わなくても済むタスクの実行を、他のタスクに優先して行うというものである。提案方式の割当て手順を図1に示す。以下に提案方式の各手続きの詳細を述べる。

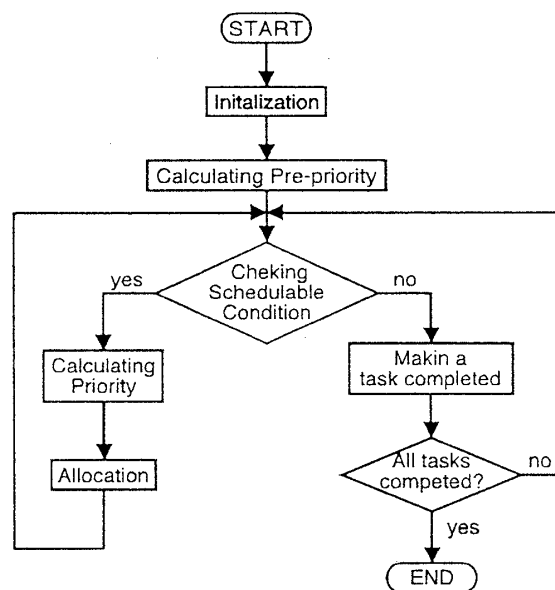


図1 割当てアルゴリズム

## (1)初期化(Initialization)

現在時刻 $C$ を0にする。

## (2)プレ・プライオリティ決定(Calculating Pre-priority)

各タスクから終了タスクまでの最長パス長をそのタスクのプレ・プライオリティ $pp(i)$ とする。

## (3)割当て可能条件の判定(Cheking Schedulable Condition)

直前タスクが全て完了したタスク(割当て可能タスク)が存在し、かつどのタスクも割当てされていない空きプロセッサが存在する場合に割当て可能条件が成立している。

## (4)プライオリティの決定(Calculating Priority)

割当て可能タスクと空きプロセッサの全ての組に対し次式で定義されるプライオリティを求める。

$$p(i,j) = pp(i) + co(i,j) \quad (1)$$

$co(i,j)$ : タスク $i$ の先行タスクの1つ以上がプロセッサ $j$ で実行され、タスク $i$ をプロセッサ $j$ で実行する場合には必要ないが、タスク $i$ をプロセッサ

Proposal and Evaluation of Scheduling Method Considering Communication Delay

Shin-ya Kobayashi, Haruhiko Kimura and Tsuyoshi Takebe  
Kanazawa University

2-40-20 Kodatsuno, Kanazawa, Ishikawa 920, Japan

j 以外に割り当てたときに必要となる通信の時間。

例えばタスク4の直前タスクがタスク1, 2, 3の3つであり, 依存関係のあるタスクが異なるプロセッサに割り当てられたとき必要な通信時間がタスク1と4では10, タスク2と4では5, タスク3と4では7であれば, タスク1がプロセッサb, タスク2と3がプロセッサcに割り当てられているならば,  $co(4, a) = 0$ ,  $co(4, b) = 10$ ,  $co(4, c) = 12$ となる。(5)割り当ての決定(Allocation)

最も大きいプライオリティの値となるタスクqとプロセッサrの組を求め, タスクqをプロセッサrに割り当てる。最も大きいプライオリティが2つ以上ある場合は, 以下の規則に従って決定する。

[規則1] 同一タスクに対する競合の場合

プライオリティ  $p(a, x)$  と, プライオリティ  $p(a, y)$  が共に等しく, その値がAである場合には, プロセッサxに対してAの次に大きい値であるプライオリティ  $p(b, x)$  とプロセッサyに対してAの次に大きい値である  $p(c, y)$  を求め,  $A - p(b, x)$  と  $A - p(c, y)$  を比較し, その差が大きい方のプロセッサにタスクaを割り当てる。

[規則2] 同一プロセッサに対する競合の場合

プライオリティ  $p(a, x)$  とプライオリティ  $p(b, x)$  が共に等しく, 最大値Bである場合には, タスクaに対してBの次に大きい値であるプライオリティ  $p(a, y)$  とタスクbに対してBの次に大きい値であるプライオリティ  $p(b, z)$  を求め,  $B - p(a, y)$  と  $B - p(b, z)$  を比較し, その差が大きいタスクをプロセッサxに割り当てる。

尚, 3つ以上のタスクやプロセッサでの競合解決も上記の規則同様に行う。上記の規則で決定できない場合には, 任意の割り当てを行う。

割り当てが決定すれば, プロセッサnがタスクmを完了する時刻(完了予定時刻)を現在の時刻Cにタスクmを実行するために必要な通信時間とタスクmの処理時間を加えた値とする。

(6)タスクの完了(Making a task completed)

現在タスクが割り当てられているプロセッサの内, 完了予定時刻が最も小さいプロセッサを選び, タスクを完了状態, プロセッサを空き状態とする。また, 現在の時刻Cを完了予定時刻に更新する。□

以上の手続きによって全てのタスクを順次プロセッサに割り当てる。

#### 4. 評価

通信時間を考慮したタスク割当法として, 提案方

式以外には文献(3)で提案された方式がある。

常微分方程式の数値解法の1つであるRunge-Kutta法のプログラム $\omega$ を対象に, 文献(3)の方法と比較評価を行った結果を表1に示す。尚, プログラムはループ展開後各行毎に1つのタスクとした。また, 各タスクの処理時間はSUN SS-ELCにより計測した(タスク総数2969個, 平均タスクサイズ1.942  $\mu$ s, 総処理時間5765.2  $\mu$ s)

表1 タスク処理時間

プロセッサ 台数	通信時間=0.32 $\mu$ s		通信時間=0.64 $\mu$ s	
	提案方式	従来方式	提案方式	従来方式
2	3439.829	3748.291	3947.365	4226.669
3	3021.440	3127.421	3529.895	3750.845
4	2806.576	2873.556	3295.882	3473.109
5	2828.671	2893.765	3299.028	3473.109
6	2815.655	2893.765	3357.405	3473.109
7	2848.390	2893.765	3375.855	3473.109
8	2843.312	2893.765	3341.691	3473.109
9	2852.817	2893.765	3347.206	3473.109
10	2846.162	2893.765	3354.129	3473.109
11	2861.053	2893.765	3389.092	3473.109
12	2867.664	2893.765	3416.398	3473.109
13	2832.282	2893.765	3406.604	3473.109
14	2836.471	2893.765	3365.827	3473.109
15	2840.237	2893.765	3417.624	3473.109
16	2838.222	2893.765	3369.732	3473.109
17	2875.828	2893.765	3426.033	3473.109
18	2839.951	2893.765	3423.913	3473.109
19	2874.044	2893.765	3434.301	3473.109
20	2861.411	2893.765	3423.196	3473.109
21	2859.438	2893.765	3373.791	3473.109
22	2875.185	2893.765	3448.899	3473.109
23	2842.133	2893.765	3421.259	3473.109
24	2857.039	2893.765	3377.251	3473.109
25	2844.993	2893.765	3439.951	3473.109
26	2882.365	2893.765	3375.990	3473.109
27	2848.158	2893.765	3379.692	3473.109
28	2847.165	2893.765	3445.071	3473.109
29	2845.666	2893.765	3381.173	3473.109
30	2846.477	2893.765	3446.317	3473.109

#### 5. あとがき

本研究では, マルチプロセッサシステムのタスク割当法として, 通信時間を考慮したスケジューリング法の提案を行い, 従来方式との比較を行い, より短い処理時間となる割り当てが行えることを示した。

#### 参考文献

- (1) Adam, Chandy and Dickson: "A Comparison of List Schedules for Parallel Processing Systems", Commun. ACM, 17, 12, pp. 685-690 (1974-12).
- (2) 笠原博徳, 成田誠之助: "マルチプロセッサ・スケジューリング問題に対する実用的な最適及び近似アルゴリズム", 信学論(D), J67-D, 7, pp. 792-799 (1984-07).
- (3) 北川秀夫, 松田文夫, 内川嘉樹, 服部秀三: "通信時間を考慮に入れたマルチプロセッサスケジューリングアルゴリズム", 信学論(D-1), J73-D-1, 10, pp. 812-817 (1990-10).
- (4) 戸川隼人: "科学技術計算ハンドブック 基礎篇C言語版", サイエンス社 (1992).