

タスク多重割当による処理時間短縮

小林 真也* 近藤 博房**
中西 暉*** 手塚 慶一****

* 金沢大学工学部 ** ミノルタカメラ
*** 大阪大学言語文化部 **** 大阪大学工学部

並列処理システムで処理されるジョブを分割して得られるタスクには、発生条件が排他的であるため、共起性のない互いに排反なタスクが存在する。このタスク間の排反性に着目し、互いに排反なタスクを同一のプロセッサに割り当てることにより処理時間の短縮を図る割当法として、タスク多重割当法がある。本稿では、タスク多重割当による処理速度の改善について、理論解析とシミュレーションにより定量的検討を行う。

Analysis on effect of task duplicate allocation

Shin-ya Kobayashi* Hirofusa Kondo**
Hikaru Nakanishi*** Yoshikazu Tezuka****

* Faculty of Technology, Kanazawa University
** Minolta Camera
*** Faculty of Language and Culture, Osaka University
**** Faculty of Engineering, Osaka University

Task duplicate allocation algorithm makes use of an exclusiveness between probability tasks. In this paper we analyse the effects of duplicate allocation quantitatively by statistic theory and simulation. We suggest a limit of the task duplicate allocation algorithm.

1 まえがき

並列処理システムは、ユーザから与えられたジョブをいくつかのタスクに分割して、各プロセッサを同期制御しながら処理を行うシステムである。このような並列処理システムにおいて高速なジョブの実行を行うためには、ジョブの持つ並列性を損なうことなく、プロセッサに割り当てるタスク割当法が重要である。

並列処理システムで処理されるジョブを構成するタスクには、処理の実行時に共起性のないタスクの組がある。我々の研究グループでは、この関係に注目することにより処理時間の短縮を目指した割当法として、タスク多重割当法の提案を行ってきた。しかしながら、処理速度の改善効果の定量的検討は不十分であった。

本稿では、シミュレーションと理論解析により処理速度の改善効果を示す。

2 タスク間関係

2.1 依存関係

ジョブを分割することにより得られたタスク相互間には、その発生の順序関係を規定する依存関係がある。この依存関係には、データの依存関係であるデータ依存関係と発生の依存関係である制御依存関係がある。以下各々について述べる。

・データ依存関係

データ依存関係にある2つのタスク間においては、ジョブの処理系列においてその2つのタスクが発生するならば、データ後続タスクは、必ずデータ先行タスクの終了以降にのみ発生できる。

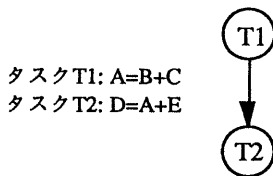


図1 データ依存関係

・制御依存関係

あるタスクが、他のタスクの処理結果によってその発生の有無が決定されるときに、それら2つのタスクは制御依存関係があるという。

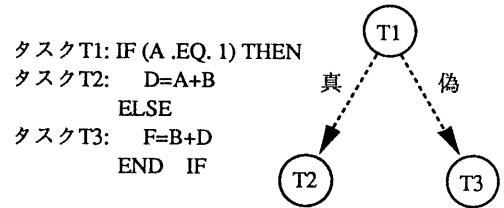


図2 制御依存関係

2.2 排反関係

ジョブを構成するタスクの中には、その発生が乱数や入力データによって決定されるため、ジョブの実行時に常に発生するとは限らないタスクが存在する。このようなタスクを確率タスクと呼ぶ。確率タスク間には、その発生条件が互いに排反的であるため、実際のジョブの処理系列において、ともに発生することのないタスクが存在する。これら2つのタスク間の関係をタスク間の排反関係と呼ぶ。例えば、図3において、タスク4と排反なタスクはタスク4と同じくタスク3を先行タスクとしており、その発生条件がタスク4とは異なるタスク5、6、およびタスク4の先行タスクであるタスク8と排反であるタスク8の3つである。

3 タスク多重割当法

3.1 多重割当

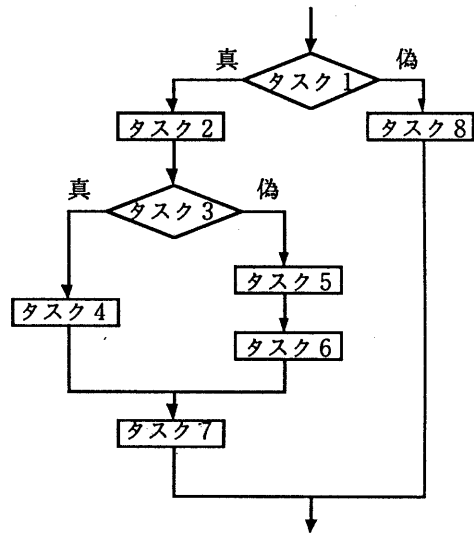
前節で述べてたように、確率タスクには発生条件が異なるため、ジョブの処理系列においてともに発生することのない互いに排反なタスクが存在する。このようなタスクを同一のプロセッサに割り当てても、実際の処理系列においてプロセッサをめぐる競争することはない。一方、このような互いに排反なタスクを同一のプロセッサに割り当てた場合と割り当てなかった場合を比較すると、図4に示すタスク集合を2台のプロセッサに割り

```

IF (タスク1) THEN
  タスク2
  IF (タスク3) THEN
    タスク4
  ELSE
    タスク5
    タスク6
  END IF
  タスク7
ELSE
  タスク8
END IF

```

・プログラムリスト



・フローチャート

図3 条件分岐を含むプログラム

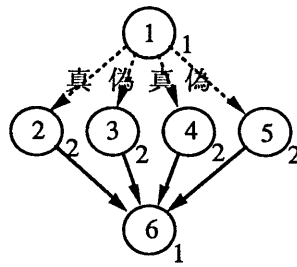


図4 タスク集合例

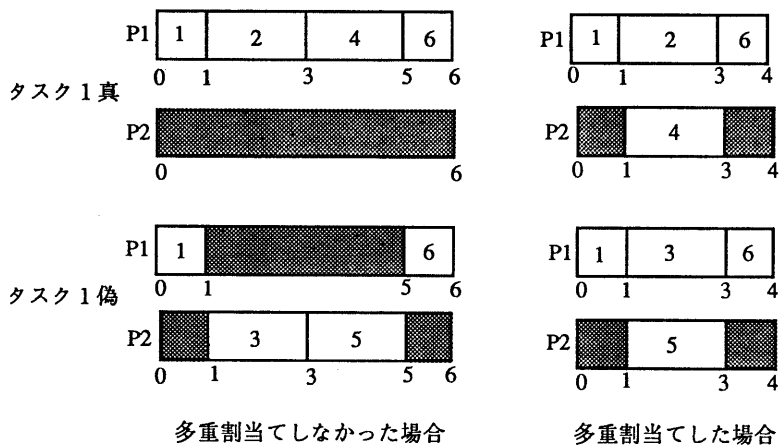


図5 多重割当て

当てるとすると、図5に示すように、互いに排反なタスクを同一のプロセッサに割り当てた方が処理時間が短くなる。このように、発生が確率的に決定するタスクを含むタスク集合をプロセッサに割り当てる際には、タスク間の排反関係に注目し、互いに排反なタスクを同一のプロセッサに割り当てた方が処理時間を短くすることができる。この、互いに排反なタスクを同一のプロセッサに割り当てることを多重割当という。

タスク多重割当法は、このタスクの多重割当を行うことができるタスク割当法である。タスク多重割当法は、CP/MISFと同様に、ある規則に従い各タスクの優先順位を決定し、割当可能となったタスクを優先順位の順に従って、プロセッサに割り当てるリストスケジューリングの一種である。以下にタスク多重割当法の割当アルゴリズムを説明する。

3. 2 優先順位決定

後続タスクのないタスクである終了タスクからの最長パス長を求め、最長パス長のもっとも長いタスクに最も高い優先順位を付ける。その際、最長パス長が等しいタスクが複数個あれば、直後の後続タスク数の多いタスクに高い優先順位を付ける。

本稿では、確率タスクの発生確率を考慮し、最長パス長を以下のように算出した。

(タスク T の最長パス長)

タスク T の直後タスクを、それらのタスクの発生する条件により以下の3つに分類し、各々の集合ごとの最長パス長を求める。

- (1) タスク T の結果に関わらず発生する全ての後続タスクの最長パス長の最大値 L_1 。
- (2) タスク T が真のときに発生する全ての後続タスクの最長パス長の最大値 L_2 。
- (3) タスク T が偽のときに発生する全ての後続タスクの最長パス長の最大値 L_3 。

次に、 L_1 、 L_2 、 L_3 の最大値を L_{max} 、中央値を L_{mid} とする。また、 L_{max} を与える直後タ

スクの発生確率を P_{max} 、タスク T の処理量を a とすると、タスク T の終了タスクへ至る最長パス長は、

$$a + L_{max} * P_{max} + L_{mid} * (1 - P_{max})$$

とする。最長パス長の算出例を図6に示す。

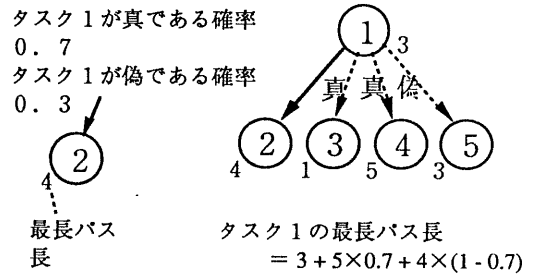


図6 最長パス長算出例

3. 3 タスクの処理量

タスク多重割当法では、タスクが確率タスクであれば、プロセッサを占有する時間を各タスクの処理量そのものを使わずに、その発生確率を考慮した値を用いる。現在、タスクの処理量は以下のように取り扱っている。

確率タスクの直後タスクをタイプ1とタイプ2の2つに分類する。確率タスク T の処理量をタイプ1、2それぞれに対して、以下のように異なる処理量 a_1 と a_2 を持つものとする。

ここで、

a : 確率タスク T の発生したときの処理量

p : 確率タスク T の処理量

とする。

(タイプ1)

確率タスク T が発生した場合にのみ発生する直後タスクであり、以下のようなタスクである。

- ・ 確率タスク T と制御依存関係があるタスク
- ・ 確率タスク T と制御依存関係のあるタスクと制御依存関係がある直後タスク

このタイプに属する直後タスクが発生するのは、確率タスク T が発生する場合に限られるので、これらの直後タスクが発生したときには、必

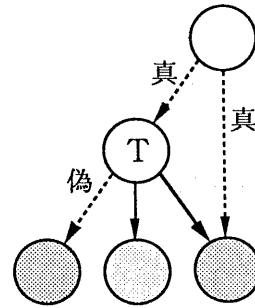
・確率タスクTの直後タスクに対する処理量

タイプ1（確率タスクTが発生した場合のみ発生）に対して

$$\text{処理量} = a$$

タイプ2（確率タスクTの発生とは独立に発生）に対して

$$\text{処理量} = a * p$$



タスクTの処理量：a タスクTの発生確率：p タイプ1 タイプ2 タイプ1

図7 確率タスクの処理量

ず確率タスクTは発生している。これより、直後タスクがタイプ1に属するならば、この直後タスクに対する確率タスクTの処理量a1を次式のように定める。

$$a1 = a \quad \text{式1}$$

(タイプ2)

タイプ1以外の直後タスクである。すなわち、確率タスクTとはデータ依存関係のみがあり、確率タスクTの発生とは独立に発生するタスクである。このタイプに属する直後タスクは確率タスクTが発生しない場合でも発生する。したがって、確率タスクTが発生しない場合、確率タスクTの処理量は0であり、確率タスクTが発生する場合には、その処理量はaであることから、タイプ2の直後タスクに対する確率タスクTの処理量を次式のような処理量の期待値とする。

$$a2 = a * p \quad \text{式2}$$

3.4 割当アルゴリズム

タスク多重割当法では以下のアルゴリズムに従って割当を行う。

- (1) 初期設定
- (2) 処理可能タスクの選出
- (3) プロセッサへのタスクの割当て
- (4) 現時刻の更新
- (5) プロセッサ状態の更新

次に、タスク多重割当法の特徴である、排反なタスクの多重割当てと、確率タスクの直後タスクの割当てについて、図8に示すタスク集合にタスク多重割当法を適用した場合を例にとって説明する。使用するプロセッサは2台とする。

・Time = 0

まず最初に割当可能となるタスク1をプロセッサ1に割り当てる。

・Time = 2

タスク1の割当終了後、後続タスクのタスク2, 3, 4が割当可能になる。このとき、タスク3およびタスク4は互いに排反なタスクであることより、同一のプロセッサ（プロセッサ1）に割り当てる。（図9）

・Time = 5

タスク3の割当終了後、後続タスクのタスク5が割当可能となるが、タスク5とタスク4は互いに排反なタスクであることより、タスク5はタスク4と同一のプロセッサ（プロセッサ1）に割り当てる。（図10）

・Time = 6

タスク4の後続タスクであるタスク6が割当可能となる。

・time = 7

タスク5のデータ後続タスクであるタスク7は、タイプ2に属する後続タスクである。よっ

確率タスクの発生確率：0.5

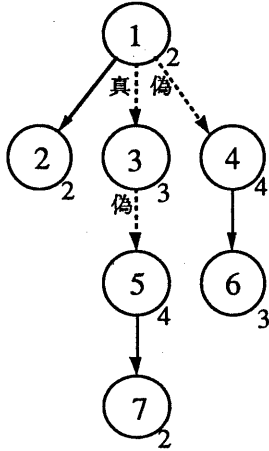


図8 割当対象

て、タスク5のタスク7に対する処理量は、式2より2となりタスク7はTime = 7で割当可能となる。このとき、タスク7は先行タスクであるタスク5と同一のプロセッサ（プロセッサ1）に割り当てる。（図11）

4 性能解析

タスク多重割当法の処理速度改善効果を理論解析とシミュレーションにより求めた。解析対象と

Time = 2

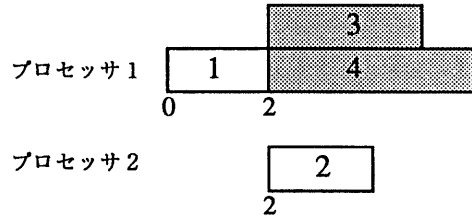


図9 Time = 2における割当状況

Time = 5

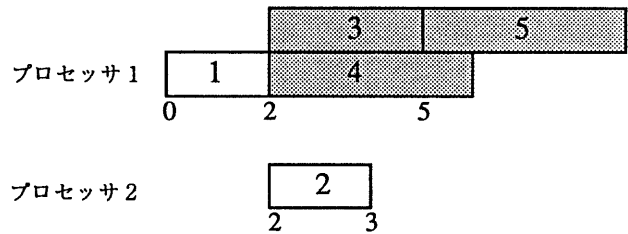


図10 Time = 5における割当状況

Time = 7

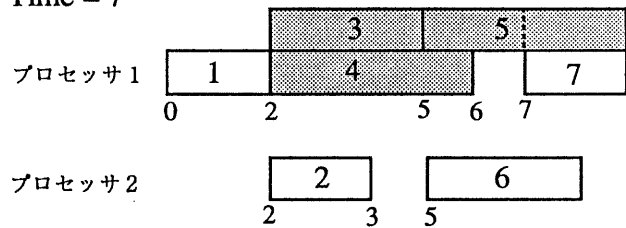


図11 Time = 7における割当状況

しては、以下に示す仮定を持つモデルを考えた。

・解析対象モデル

- (1) 全タスク数は100個である。
- (2) 条件タスクは1個である。
- (3) 発生条件が真であるタスクと偽であるタスクは同数個存在する。
- (4) タスク相互間にはデータ依存関係がない。

上記のモデルに対して、排反関係を考慮せずに、2台のプロセッサに対して全ての確率タスクが常に発生すると仮定し、制御依存関係をデータ依存関係とみなしてCP/MISF法を適用した場合とタスク多重割当法を用いて割当を行った場合と比較し、CP/MISF法を用いた場合の処理時間に対して、タスク多重割当法を用いて割当を行った場合の処理時間の比Rを求めた。

$$R = \frac{\text{(タスク多重割当法での処理時間)}}{\text{(CP/MISF法での処理時間)}}$$

4. 1 理論解析

以下の議論において用いる変数を以下の様に定義する。

S : 全タスクの処理量 (時間) の和

r : 全タスクに対する確率タスクの割合

・改善限界

CP/MISF法で最も処理時間がかかるのは、1つのプロセッサに割り当てられた確率タスクの発生条件が全て同じときである。従って、CP/MISF法による処理時間はプロセッサ台数が2台であるからS/2となる。一方、タスク多重割当法では、確率タスクの処理量の和はrSとなり、確率タスクは多重割当されるため確率タスクの処理に要する時間は

$$(rS/2) / 2 = rS/4$$

となる。従って、全タスクの処理に要する時間は

$$(S - rS) / 2 + rS / 4 = S/2 - rS/4 = (2 - r)S/4$$

となる。よって、処理時間比Rは、

$$R = \frac{(2 - r)S/4}{S/2} = 1 - r/2$$

となる。

・改善平均

タスク間の排反関係を考慮せずにCP/MISF法を適用した場合、これら互いに排反なタスクが多重割当されることがある。そこで、多重割当される確率を考慮し、CP/MISF法の処理時

間の期待値を求め、タスク多重割当法の処理時間比の期待値を以下のように求めた。

ここでは、条件タスクが真の時に発生するタスク、偽のときに発生するタスクともに5個存在する場合について説明する。

確率タスクのみに注目すると一方のプロセッサに割り当てられたタスクが決れば、もう一方のプロセッサに割り当てられたタスクも決定される。そこで一方のプロセッサに割り当てられたタスクについて、その組み合わせにより分類を行うと、次の6通りに分けられる。なお、真のときに発生するタスクを○、偽のときに発生するタスクを×で表す。

(○○○○○) (○○○○×) (○○○××)
(○○×××) (○××××) (×××××)

これら6つの場合各々について、確率タスクの処理に要する時間と、発生確率を求めると以下のようになる。なお、確率タスク1個の処理量は1と仮定する。

・ (○○○○○)

条件が真であるあるタスクが5個、偽であるタスクが0個であるから、場合の数は

$${}_5C_5 \times {}_5C_0 = ({}_5C_5)^2 = 1$$

であり、処理量は真の場合は、発生条件が真であるタスクが5個割り当てられたプロセッサが5となり、偽の場合は、発生条件が偽であるタスクが割り当てられたタスクの処理量が5となるため、いずれの場合においても処理量は5となる。

・ (○○○○×)

同様に、場合の数は

$${}_5C_4 \times {}_5C_1 = ({}_5C_4)^2 = 25$$

であり、処理量は4となる。

・ (○○○××)

同様に、場合の数は

$${}_5C_3 \times {}_5C_2 = ({}_5C_3)^2 = 100$$

であり、処理量は3となる。

・ (○○×××)

同様に、場合の数は

$${}_5C_2 \times {}_5C_3 = ({}_5C_2)^2 = 100$$

であり、処理量は3となる。

- ・ (〇××××)

同様に、場合の数は

$${}_5C_1 \times {}_5C_4 = ({}_5C_1)^2 = 25$$

であり、処理量は4となる。

- ・ (×××××)

同様に、場合の数は

$${}_5C_0 \times {}_5C_5 = ({}_5C_0)^2 = 1$$

であり、処理量は5となる。

従って、処理量の期待値を求めると、

$$3.2142$$

となる。

解析対象モデルにおいて確率タスクの個数を0個から100個まで変化させ、処理時間の改善比を求めると図12の様になる。

4.2 シミュレーション解析

次に、解析モデルを実際に割当法により割当を行い、処理時間を求めた。その結果、表1に示す結果を得た。

また、理論解析の結果とシミュレーション解析の結果を図12に示す。

このように、理論解析もシミュレーション解析もほぼ同程度の改善が行えることがわかる。

5 おわりに

確率タスクの排反関係に注目し、発生条件が排他的であるため共起性のない互いに排反なタスクを多重割当するタスク多重割当法の、改善効果について解析を行った。その結果、多重割当により期待できる改善効果の定量化が行えた。

実際のタスク間には今回のモデルとは異なり、データ依存関係が存在する。従って、タスクの優先順位決定法がタスク多重割当を行った際の処理時間に大きく影響すると考えられる。今後、確率タスクを考慮した、タスクの優先順位決定法の改良を行う必要がある。

謝辞 本稿を作成するにあたり、金沢大学工学部

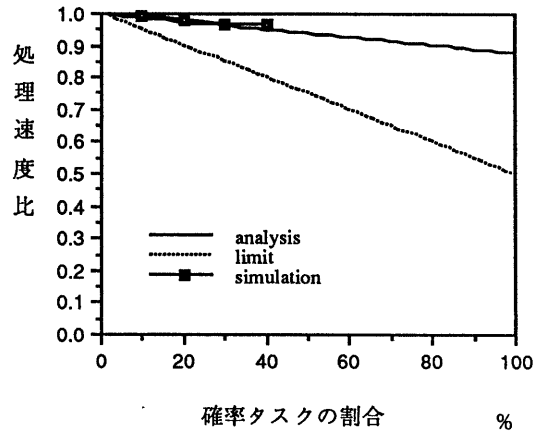


図12 処理速度比

表1 処理速度比

確率タスクの割合 %	処理速度比
10	0.9910
20	0.9792
30	0.9687
40	0.9652

電気・情報工学科学部学生の友永誠史君には、データの整理等で協力を頂いた。感謝の意を表す。

参考文献

- (1) 小林, 小巻, 中西, 手塚: “タスク間の排反性に注目したタスク割当法について”, 電子情報通信学会コンピュータシステム研究会, CPSY90-34(1990-10)
- (2) 小林, 中西, 手塚: “タスク多重割当法の提案”, 1990年電子情報通信学会秋季全国大会, D-56(1990-10)