システムソフトウェアと
オペレーティング・システム 72－4
(1996. 5. 15)

タスク割り当て法における通信時間の取り扱いに対する検討

左木茂克　　小林真也

金沢大学 工学部

〒920 金沢市小立野 2-40-20

あらまし　マルチプロセッサシステムにおける有効なタスク割り当て法として，タスクの最長パス長をプライオリティとしたリストスケジューリングにより割り当てを決定する方法がある．最長パス長とはそのタスクの開始から全タスクの終了までに少なくとも必要となる時間を表すが，通信時間による影響を考慮していない．ところが，通信時間はタスクが終了までに要する時間を変化させ，最長パス長に誤差が生じる．そのため，通信時間が最長パス長に与える影響を扱う手段が必要となる．本研究では，通信時間が最長パス長に与える影響を定量的に扱う方法についてシミュレーションにより検討を行う．

キーワード　マルチプロセッサシステム，スケジューリング，通信オーバヘッド

Evaluation of Task Scheduling Methods Dealing with Influence of Communication Overheads

Shigekatsu Sagi　　Shin-ya Kobayashi

Faculty of Engineering, Kanazawa University

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

Abstract　One of effective task scheduling methods on multiprocessor system is list scheduling using critical path length of a task. Critical path length of a task indicates the time that is needed at least to complete executing all tasks, but it considers only size of tasks. And communication overheads influence the time that is needed to complete executing all tasks. Therefore task scheduling method must take account into influence of communication overheads on critical path length. In this paper, we evaluate the methods dealing with influence of communication overheads in task scheduling method.

key words　Multiprocessor system, Scheduling, Communication overheads

1 はじめに

マルチプロセッサシステムにおいて投入されたタスク集合の個々のタスクををどのプロセッサに割り当て，どのような順序で実行するかを決定するタスクスケジューリングはシステムの性能を決定する重要な要因となる．しかし，最適なスケジュールを求める問題は多項式時間では解くことができず，実用的な時間で最適解を求めることは不可能である．

そこで，準最適解をヒューリスティックなアプローチにより求めるタスクスケジューリングアルゴリズムが提案されている．そのようなタスクスケジューリングアルゴリズムの基本的なものの１つが，タスクの最長パス長をプライオリティとしたリストスケジューリングにより割当を決定するクリティカルパス法 (CP 法) である．CP 法ではそのタスクの実行を開始してから全タスクが終了するまでに少なくとも必要となる時間を示す最長パス長の大きなタスクから割り当ててゆくことにより比較的良いタスク割り当てを行う．

ところが，最長パス長はタスクサイズのみで決定され，通信時間は全く考慮されていないため，通信時間により全タスク終了までの時間が変化すると最長パス長との間に誤差が生じる．そのため，CP 法が良好な解を得ることができるのは，プロセッサ間の通信に要する時間がタスクの処理時間 (タスクサイズ) に比べ無視できる程小さい場合に限られる．

通信時間がタスクサイズに比べ無視できない程の大きさである場合にも良好な割当を行うためには，通信時間の最長パス長に対する影響を正しく取り扱う手段を備えたスケジューリングアルゴリズムが必要となる．

通信時間の最長パス長に対する影響を取り扱う手段を備えたアルゴリズムとして，我々の研究グループが提案している CP/RCO 法 [1][2] が挙げられる．CP/RCO 法ではプライオリティを最長パス長と通信削減量の和とすることでこの影響を考慮している．そのため，CP/RCO 法は他の CP 法から派生した通信時間の最長パス長に対する影響を考慮していない方式に比べて良好な割当を行えることが示されている．

本研究では，タスクスケジューリングアルゴリズムにおいて，通信時間が最長パス長に与える影響を正しく取り扱うためのいくつかの手段をシミュレーションによって評価し，それぞれを比較検討した．

2 諸定義

2.1 マルチプロセッサシステムの定義

本研究ではタスクの処理を行うマルチプロセッサシステムを，以下のような MIMD 型のマルチプロセッサシステムと定義する．

- 各プロセッサはすべて等しい処理能力をもち，タスクサイズが同一であれば実行したプロセッサによらず，タスクの終了までに要する時間は同一である．
- 各プロセッサはマルチポートメモリで構成される分散共有メモリを持ち，自分の持つメモリ (ローカルメモリ) や他のプロセッサの持つメモリ (リモートメモリ) 上のデータを元にタスクの処理を行い，ローカルメモリにその結果を書き込む．
- リモートメモリ上のデータの読み出しには通信オーバヘッドが必要となり，タスクの処理時間はリモートメモリへのアクセスが必要ない場合に比べ通信時間分だけ長くなる．
- 通信は通信路で直接に結合されたプロセッサ間でのみ行うことができ，直接に結合されていないプロセッサ間の通信は他のプロセッサを経由して行われる．そのため，通信時間は通信量とプロセッサ間の距離の積によって決定される．ここで，プロセッサ間の距離とは通信のために経由しなければならない通信路の数である．ただし，プロセッサ間の通信は常に最短距離で行われるものとする．
- メモリはマルチポートメモリであるので同時に複数のアクセスを受け付けることができるが，プロセッサは１つのアクセスしか実行できず，その間タスクの処理も行うことはできない．

2.2 タスクスケジューリング問題の定義

本研究が扱うタスクスケジューリング問題は与えられたタスク集合を先に定義したマルチプロセッサシステムで実行したときに，その全タスクが終了するまでに要する時間 (総処理時間) を最小にする割当 (スケジュール) をそのタスク集合の静的な情報に基づいて決定するというものである．

スケジューリングアルゴリズムに入力として与えられるタスク集合を以下のように定義する．

- タスク集合は n 個のタスクからなり，それぞれのタスク $T_i(1 \leq i \leq n)$ に対して固有の処理時間 (タスクサイズ)t_iが決まっている．
- 実行時においてあるタスクが実行されるのかどうかを決定する制御依存関係はないものとする．また，逆依存関係，出力依存関係もプログラムの変数名を付け替えることにより除去され，存在しないものとする．そのため，与えられるタスク集合は順依存関係のみを持つ．
- タスク集合は処理順序に関して半順序関係をなす．タスク T_iとタスク T_jが半順序関係 $T_i \ll T_j$を満たすとき，タスク T_iをタスク T_jの先行タスク，タスク T_jをタスク T_iの後続タスクと呼ぶ．$T_i \ll T_j$ であれば，タスク T_iの終了時刻 E_iとタスク T_jの開始時刻 S_jは $E_i \leq S_j$でなければならない．さらに，$T_i \ll T_k \ll T_j$となる T_kが存在しないとき，タスク T_iをタスク T_jの直前タスク，タスク T_jをタスク T_iの直後タスクという．

- 各タスクは実行のためにその直前タスクの処理結果を通信によって受け取らなければならない．タスクがその直前タスクから受け取らなけらばならないデータの量はタスクとその直前タスクの対ごとに固有である．

タスク集合は，タスクを表すノードとデータ依存関係を表すアークによってタスクグラフと呼ばれる有限無サイクル有向グラフで表される．ただし，ノード中の数値はタスクサイズを表し，アーク横の数値はアークで継れたタスク間で要する通信量である．その例を図1に示す．

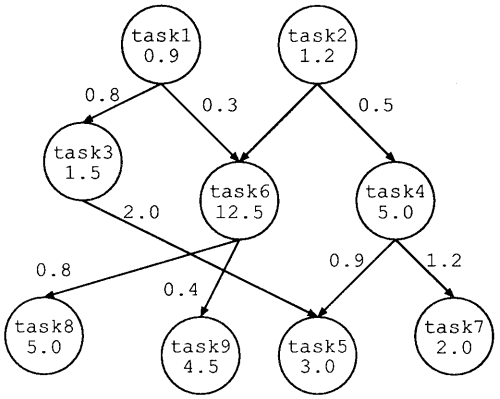

図 1: タスクグラフ

3 スケジューリングアルゴリズム

本研究が扱うスケジューリングアルゴリズムはリストスケジューリングの一種である．図 2にそのフローチャートを示し，以下ではその詳細について述べる．

1. 初期化 (Clock initialization)
   時計の表す時刻を 0 に初期化する．

2. 割り当て可能条件の判定 (Checking schedulable condition)
   時刻 Cにおいて，直前タスクがすべて終了したタスク (割り当て可能タスク) が存在し，かつどのタスクも割り当てられていないプロセッサ (空きプロセッサ) が存在する場合に割り当て可能条件が成立している．

3. プライオリティの計算 (Calculating priority)
   すべての割り当て可能タスクと空きプロセッサの組に対してプライオリティを計算する．

4. 割当の決定 (Allocation)
   最も大きなプライオリティ値をもつ割り当て可能タスクと空きプロセッサの組を次に割り当てるべき割り当て候補として選択し，割り当てる．割当が決定すれば，現在の時刻 Cに必要となる通信時間とタスクの処理時間を加えた時刻を求め，その

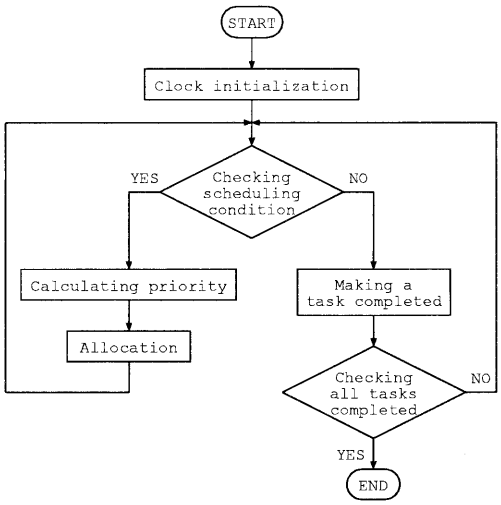

図 2: アルゴリズムのフローチャート

時刻を選択されたプロセッサがタスクの処理をを完了する時刻 (完了予定時刻) とする．

5. タスクの完了 (Making a task complated)
   現在の時刻においてタスクが割り当てられているプロセッサのうち，完了予定時刻が最も小さいプロセッサを選択し，そのプロセッサが処理を行っているタスクを完了状態，またそのプロセッサを空き状態とする．また，時計の時刻を選択された完了予定時刻まで進める．

6. 割当終了条件の判定 (Checking all tasks complated)
   すべてのタスクの処理が終了していれば，割当終了条件が成立している．

4 検討を行った方式

本研究では前節で述べたスケジューリングアルゴリズムのプライオリティ，及び割当の決定方式として有効であると思われるものをいくつか挙げ，それぞれの方式をシミュレーションにより比較評価し，タスクスケジューリングアルゴリズムにおける有効な通信時間の取り扱い方法について検討を行った．

以下では，検討を行ったプライオリティと割当の決定方式について説明する．

4.1 プライオリティ

- prio1
  プライオリティをタスクの最長パス長と通信削減量の和とする．すべての割り当て可能タスク T_iとすべての空きプロセッサ P_jの組に対して，プライオリティ$p(i,j)$ は式 (1) のように定義される．このプライオリティは CP/RCO 法が採用している

ものである．

$$p(i,j) = cpl(i) + rco(i,j) \qquad (1)$$

ここで，

$cpl(i)$ ： タスク i の最長パス長
$rco(i,j)$ ： 通信削減量

である．通信削減量とはタスク T_i をプロセッサ P_j で実行することで，最も通信時間が必要となる場合と比べて削減できる通信時間を表す．通信削減量は具体的には式 (2) で求められる．

$$rco(i,j) = \sum_{k=1}^{l_i} (mco(i,i_k) - co(i,i_k,j)) \qquad (2)$$

ただし，l_i はタスク i の直前タスク数であり，$co(i,i_k,j)$ はタスク i がプロセッサ j に割り当てられたときにタスク i の直前タスクであるタスク i_k が実行されたプロセッサとプロセッサ j との間で必要な通信時間である．また，$mco(i,i_k)$ は j を変化させたときの $co(i,i_k,j)$ の最大値である．同一のプライオリティ値を持つ組が複数ある場合の優先順位は次の規則に従う．

- 同一タスクに対して競合している場合
  $p(a,x) = p(a,y)$ であった場合，それぞれのプロセッサに対する 2 番目に大きなプライオリティの値が小さい方が高い優先順位をもつ．
- 同一プロセッサに対して競合している場合
  $p(a,x) = p(b,x)$ であった場合，それぞれのプロセッサに対する 2 番目に大きなプライオリティの値が小さい方が高い優先順位をもつ．

- prio2
  プライオリティをタスクの最長パス長と通信時間の差とする．すべての割り当て可能タスク T_i とすべての空きプロセッサ P_j の組に対して，プライオリティ $p(i,j)$ は式 (3) のように定義される．

$$p(i,j) = cpl(i) - com(i,j) \qquad (3)$$

ここで，

$cpl(i)$ ： タスク i の最長パス長
$com(i,j)$ ： 通信時間

である．タスク T_i をプロセッサ P_j で実行することで必要となる通信時間は具体的には式 (4) で求められる．

$$com(i,j) = \sum_{k=1}^{l_i} co(i,i_k,j) \qquad (4)$$

ただし，$co(i,i_k,j)$, 及び l_i の定義については prio1 と同様である．同一のプライオリティ値を持つ組が複数ある場合の優先順位もまた prio1 と同様である．

- prio3
  プライオリティをタスクの最長パス長と通信時間の和とする．すべての割り当て可能タスク T_i に対して，プライオリティ $p(i)$ は式 (5) のように定義される．

$$p(i) = cpl(i) + \min_j(com(i,j)) \qquad (5)$$

ここで，

$cpl(i)$ ： タスク i の最長パス長
$com(i,j)$ ： 通信時間

であり，$com(i,j)$ の定義については prio2 と同様である．ただし，タスクと組になるプロセッサはそのタスクを実行するにあたり，最も通信量が少なくて済むプロセッサとする．

4.2 割当の決定方式

- alg1
  最も高いプライオリティ値を持つ割り当て可能タスク T_q と空きプロセッサ P_r の組を選択し，即座にプロセッサ P_r にタスク T_q を割り当てる．

- alg2
  このアルゴリズムのフローチャートを図 3に示し，詳細を以下で説明する．

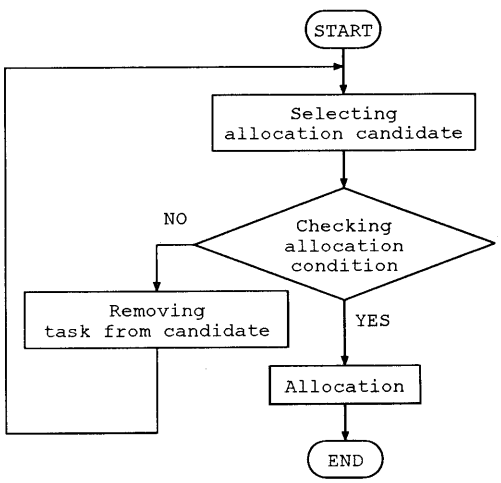

図 3: 割り当て決定方式

1. 割り当て候補の選択 (Selecting allocation candidate)
   割り当て可能タスクの集合と空きプロセッサの集合から最も高いプライオリティ値を持つタスク T_q とプロセッサ P_r の組を選択する．

2. 割り当て条件の判定 (Checking allocation condition)
   現在の時刻において処理中のプロセッサの中に現在行っている処理が終了するまでの待ち時間を含めても選択された空きプロセッサ P_r よりも早くにタスクを終了できるプロセッサが存在しない場合には割り当て条件が成立している.
3. タスクの割り当て候補からの削除 (Removing task from candidate)
   タスク T_q を割り当て可能タスクの集合から一時的に外す.
4. 割り当て (Allocation)
   割り当て可能タスクの集合から一時的に外されていたタスクを割り当て可能タスクへ戻し, タスク T_q をプロセッサ P_r へ割り当てる.

今回検討を行った方式は上で述べた, プライオリティ3通りと割当決定方式2通りを組み合わせた計6通りである. それぞれの方式の名称とそのプライオリティと割当決定方式の対応を以下の表1に示す.

表 1: 方式名

方式名	プライオリティ	割当決定方式
CP/RCO	prio1	alg1
CP/RCO/PA	prio1	alg2
CP/COM	prio2	alg1
CP/COM/PA	prio2	alg2
CP/NEW	prio3	alg1
CP/NEW/PA	prio3	alg2

5 評価

5.1 シミュレーション条件

割当を行うタスク集合として, 常微分方程式の数値解を求める4次の Runge-Kutta 法 [3] を用いた. Runge-Kutta 法のプログラムは常微分方程式 $y'' + \sin(y) = 0$ (初期条件 $y(0) = 3, y'(0) = 0$) を計算範囲 $0 \leqq x \leqq 16$ で解くものである.

C 言語で書かれたこのプログラムをループ展開した後, 各命令ステートメントをそれぞれ1つのタスクとして分割した. このタスク集合のタスク数は2969であり, タスクサイズの平均, 最小, 最大, 分散はそれぞれ 1.942, 0.728, 4.584, 1.216 である. なお, 各タスクの処理時間は SUN Sparc Station ELC(21MIPS) で実行し, その CPU タイム計測したものであり, その単位は $\mu sec.$ である.

タスク間の通信量を $0.00, 0.32, 0.64, 0.96, 1.28, 1.60\mu s$ と変化させてタスク集合の総処理時間の平均, 最大, 最小を調べた.

また, シミュレーションの際に用いたプロセッサ間の結合網と, その網のプロセッサ台数, プロセッサ間距離の平均, 最大, 分散を表2に示す.

表 2: プロセッサ間結合網

網形態	台数	平均	最大	分散
Completely connected (comp16)	16	0.94	1	0.05
Cube (cube16)	16	2.00	4	1.00
Mesh (mesh16)	16	1.88	3	0.73
Star (star16)	16	1.76	2	0.31
Chordal ring (chord16)	16	2.50	5	1.75
3-cube-connected cycle (3cube24)	24	3.08	6	2.16
Mesh (mesh121)	121	5.45	10	5.12
Cube (cube128)	128	3.50	7	1.75

5.2 シミュレーション結果と評価

表2の各結合網で接続されたマルチプロセッサシステムで先に述べたタスク集合を処理した場合のシミュレーションを行った.

タスク間の通信時間を 0.00, 0.32, 0.64, 0.96,1.28, $1.60\mu s$ とし, それぞれで表1の各方式を用いてタスク割当を行った場合のタスクの平均総処理時間を調べた.

その結果を表3に示す. ただし, 表中の数値は CP/RCO 法を用いて割り当てた場合の結果を1としている.

割当の決定法について注目するとプロセッサ数121のメッシュ網のようなプロセッサ間距離の分散が大きな結合網においては割当の決定法として alg2 を用いた方式が alg1 を用いた方式に比べて良い結果を示し, 最大で約22%程短い時間で全タスクの処理を終了できることが分かる.

ところが, プロセッサ間距離の分散が小さな完全網やスター型の結合網においては alg1 を用いた方法の結果と alg2 を用いた方法の結果の間では, ほとんど違いがない. この理由として以下のようなことが考えられる.

プロセッサ間の距離の分散が大きく, プロセッサ間の距離にばらつきがあるためにどのプロセッサにタスクを割り当てるかによって通信時間が大きく異なる. そのため, 空きプロセッサに即座にタスクを割り当てる alg1 よりもプロセッサ間距離の小さな処理中プロセッサの処理終了を待つ alg2 の方が待ち時間を含めたとしても, 結果的により早くタスクを終了できる場合が多く発生したからであると考えられる.

表 3: シミュレーション結果

	comp16	cube16	mesh16	star16
CP/RCO	1.00	1.00	1.00	1.00
CP/RCO/PA	1.01	0.96	0.96	0.98
CP/COM	1.00	1.00	1.00	0.85
CP/COM/PA	1.01	0.95	0.93	0.85
CP/NEW	1.03	0.97	0.98	0.82
CP/NEW/PA	1.01	0.93	0.92	0.82

	chord16	3cube24	mesh121	cube128
CP/RCO	1.00	1.00	1.00	1.00
CP/RCO/PA	1.00	0.88	0.78	0.91
CP/COM	1.00	1.00	1.00	1.00
CP/COM/PA	0.90	0.85	0.77	0.90
CP/NEW	0.98	0.96	0.97	0.98
CP/NEW/PA	0.89	0.85	0.75	0.89

つまり，alg1はプロセッサの処理終了までの待ち時間を避けるために次々と空きプロセッサにタスクを割り当て，タスクをプロセッサ全体に分散させてしまい，結果的にタスクの処理結果の収集のための通信時間を増加さてしまうという問題を持つ．それに対し，alg2はタスクを空きプロセッサに即座に割り当てるのではなく，スケジュールに待ちを挿入して割り当てを遅れさせることにより遠方のプロセッサを無駄に使うことなく使用するプロセッサ数を制限し，この問題を解決する．

つぎに，プライオリティについて注目すると，不完全網においてalg2と組み合わせた場合に，最長パス長と通信削減量の和にであるprio1を用いた方法が他のプライオリティを用いた方式に比べ，悪い結果を示していることが分かる．

prio1とprio2を比較した場合，それらの違いは最悪の場合の通信時間を加えるかどうかである．ここで，プライオリティ式中の最悪通信時間について考えると，最長パス長と通信時間が同じであるタスクとプロセッサの組が複数あった場合に，他のプロセッサで実行した場合により大きな通信時間が必要となる組を優先して割り当てるという意味を持つ．

ところが，上でも述べたようにalg1を用いることにより，通信削減量が考えるような最も通信時間が必要となるような状況が発生する可能性が低くなってしまう．そのような状況で最も通信時間が必要となる場合を基準とした割当を行うことには問題がある．その結果，他のプライオリティを用いた方法に比べ，悪い結果を示したと考えられる．

また，prio3については不完全網においては他のプライオリティに比べて良い結果を示すものの，完全網においては逆に悪い結果を示す．この理由を以下のように考えることができる．

prio2とprio3の違いは最長パス長が同じタスクが複数あった場合に通信時間の長い割当から行うか，それとも通信時間の短い割当から行うかである．prio3は通信時間の大きな組み合わせから割り当てることにより，通信時間が短くて済む他のタスクの割当を妨害する可能性がある．一方，prio2は通信時間の短い割当から行うことで，通信時間の長い割当を妨害し，更に長い通信時間が必要なものにするかもしれない．しかしながら，完全網においてプロセッサ間の距離は最大でも1でしかなく，この通信時間の増加は問題にならないと考えられる．そのため，完全網においてはprio3がprio2に比べ悪い結果を示すと考えられる．

6 おわりに

マルチプロセッサシステムでのタスク割り当て法における通信時間の取り扱い方式についての検討を行った．

本研究が扱うタスクスケジューリングアルゴリズムはリストスケジューリングの一種であり，プライオリティに従って割当を決定して行く．

今回はそのプライオリティ，及び割り当ての決定方式として有効であると考えられるものをいくつか挙げ，それを用いたタスク割り当て法をシミュレーションにより比較評価した．

その結果より，割当の決定方式としては，処理中のプロセッサ中に空きプロセッサよりも早くタスクを終了できるものが存在する場合にはタスクの割り当てを行わずに延期するという方式が不完全網，特にプロセッサ間距離の分散が大きな結合網において有効であることを示した．

また，不完全網でこの割当の決定方式を用いた場合には，通信削減量に基づいて割り当てを行うことには問題があり，通信量そのものに基づいた方が良い割り当てを行えることを示した．

最後に，プライオリティとして最長パス長と通信時間の和を用いた場合にはほとんどの不完全網で良好な割り当てを行えるものの，完全網においては問題があることを示した．

参考文献

[1] 小林, 木村, 武部, “マルチプロセッサシステムにおける通信時間を考慮したタスク割当て法”, 電子情報通信学会論文誌 D-I, Vol.J76-D-I, No.12, pp.689-694 (1993)

[2] 小林, “不完全結合マルチプロセッサシステムに対するタスク割当て法の提案と評価”, 電子情報通信学会論文誌 D-I, Vol.J79-D-I, No.2, pp.69-78 (1996)

[3] 戸川, “科学技術計算ハンドブック”, サイエンス社 (1992)