

メッシュ結合並列計算機用パーティショニングアルゴリズム 2D Buddy
 3 T-7 または Adaptive Scan を使った時分割処理の評価

須崎 有康 田沼 均 平野 聡 一杉 裕志 塚本 亨治

電子技術総合研究所

1 はじめに

筆者らは文献[1]においてメッシュ結合並列計算機のパーティショニングアルゴリズム 2D Buddy(2DB)[2] または Adaptive Scan(AS)[3] を使った時分割処理を提案した。ここでは時分割処理を行なうことで通常の2DB、ASよりも応答性とプロセッサ利用率が向上することを示した。本論文では提案した時分割処理の有用性を確かめるため、負荷限界について評価を行なった。計算機システムにある負荷でタスクを投入するとタスクが処理しきれず、タスクが滞る状態になる。ここで述べる負荷限界は、タスクが滞らない負荷の限界点である。

2 並列計算機における時分割処理

メッシュ結合並列計算機上で矩形のプロセッサ群を要求するタスクにプロセッサを効率的に割り当てるパーティショニングアルゴリズムに2DB[2]やAS[3]がある。これらのアルゴリズムは、プロセッサ数の大部分への割り当て要求をするタスクがあるとそのタスクの前後にあるタスクが同一時間帯に並列計算機に割り当てられず、プロセッサの利用率が上げられない状態(閉塞状態)になる。

この閉塞状態を解決する方法として筆者らは時分割処理を行なうことを提案した[1]。ここでは対象とする並列計算機の構造を表す仮想並列計算機(スライス)を複数提供する。各スライスへのタスクの割り当てはパーティショニングアルゴリズムで行なう。既存のスライスへ割り当てられなくなったら新しいスライスを作成し、そのスライスにタスクを割り当てる。スライスはラウンドロビンで一定時間実計算機に割り当てられ、スライスに割り当てられたタスクの処理を進める(図1)。また本方式では、あるスライスでタスクが占有している領域がもし別のスライスで未使用の場合には、その空いている領域にそのタスクを割り当てることができ、プロセッサの利用率を上げる。図1のsingle taskはスライスに唯一存在

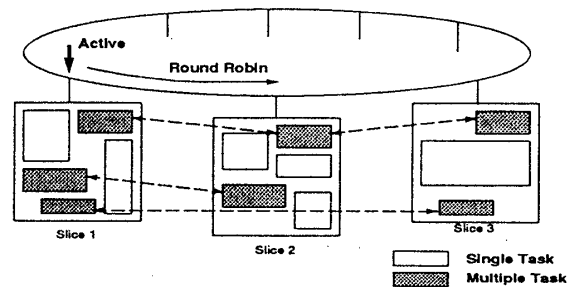


図1: スライスに基づく時分割処理

するタスク、multiple taskは複数のスライスに存在するタスクである。

3 負荷限界の評価法

本論文では、計算機システムに投入されたタスクが滞ることなく処理が行なわれる最大負荷を負荷限界と呼び、提案した時分割処理の負荷限界を求める。負荷限界の評価を行なうため以下のことを定義する。必要となるデータは、タスクの処理にかかるCPU時間、タスクの投入時刻、タスクの終了時刻である。

タスクの平均投入間隔($\bar{T}_{interval}$)は個々のタスクの投入時刻により、平均CPU時間(\bar{T}_{CPU})は個々のタスクのCPU時間(T_{CPU})により求められ、システムにかかる負荷(Load)は式1で定義される。

$$Load = \bar{T}_{CPU} / \bar{T}_{interval} \quad (1)$$

タスクを投入してから終了するまでの時間を経過時間(T_{elapse})とする。個々のタスクの応答(Response)は式2で定義される。応答はCPU時間に対する経過時間の乖離の程度を表している。

$$Response = T_{elapse} / T_{CPU} \quad (2)$$

応答は1以上の値をとり、1のときは利用者にとって望ましい処理が行なわれていることを示す。式2からわかるように時間割り当て待ちが起こると応答は大きくなる。投入されているタスクを処理しきれないうちに新しいタスクが投入される状況では時間経過とともにスライスが多くなり、応答が大きくなる。本論文では投入時刻に対する応答を一次関数で推定し、その傾きにより応答の時間変化を求める。この傾きが0であるならば、タスクは滞っておらず応答は投入時刻によって変化しないことがわ

Evaluation of TSS using partition algorithm "2D Buddy" or "Adaptive Scan" for mesh-connected parallel computers
 Kuniyasu SUZAKI Hitoshi TANUMA Satoshi HIRANO
 Yuuji ICHISUGI Michiharu TUKAMOTO
 Electrotechnical Laboratory

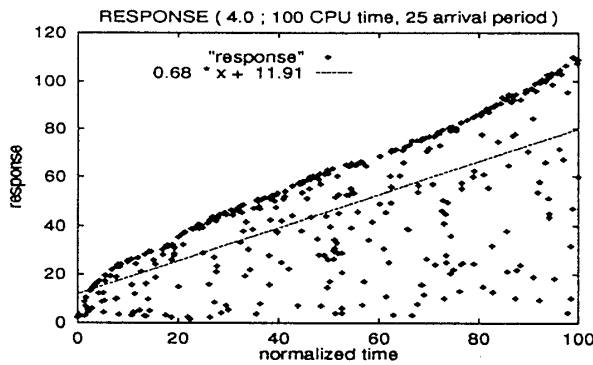


図 2: 投入時刻に対する応答

かる。そのときの負荷はシステムがタスクを滞らせずに処理可能な負荷である。ここで応答を処理の速度を表すものとみなすと傾きは加速度を表しているので、この傾きを応答の加速度と呼ぶ。ここでは応答の加速度を負荷に対して一意に扱うため、投入時刻を平均 CPU 時間で割ることにより正規化する。

応答の加速度はタスクの経過時間の増加の割合を意味しており、負荷に対する関数となる。この関数はタスクが滞らないうちは応答が投入時刻によって影響されないため 0 となる。しかし、ある負荷以上ではタスクが滞るため投入時刻が遅くなるのに従って応答が大きくなり、応答の加速度が正の値をとる。システムは応答の加速度が 0 である負荷が大きいくほど負荷限界が高く、システムは高い負荷でもタスクを滞らせずに処理できることを示す。

4 シミュレーションによる評価

2DB または AS を用いた時分割処理の負荷限界を 3 章の評価法に基づき、シミュレーションにより測定した。ここでは対象とするメッシュ結合並列計算機を 32×32 プロセッサとし、タスクのモデルを M/D/1 とした。このモデルはタスクの投入間隔は指数分布、タスクの CPU 時間は一定、タスクを受け付ける窓口が一つである。すべてのタスクの CPU 時間をシミュレーションの時間で 100、投入するタスクの数は 1000 個とし、測定は負荷が定常状態の間に処理されているタスクを対象とした。タスクが要求する矩形領域は、一辺を 1 から 32 までの乱数とした。

シミュレーションによって求められたタスクの応答の一例を図 2 に示す。図 2 は AS を使った時分割処理に負荷 4.0 (平均 CPU 時間 100 に対して平均投入間隔 25) のタスクを与えた結果である。この図より負荷 4.0 では投入時刻に対応して応答が大きくなっていることがわかる。しかし、筆者らが提案した時分割処理の特徴である複数のスライス間に存在するタスクの効果のため、投入時刻が遅くても応答が良いものがあることが示されている。また図 2 に示されている直線 (点線) は、投入時刻と応答の関係を示したものである。この直線の傾きが応答の加速

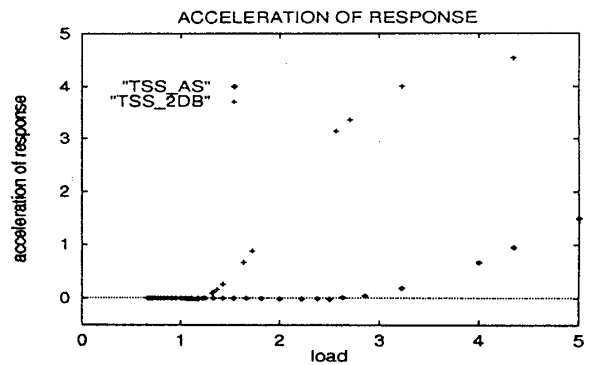


図 3: 負荷に対する応答の加速度

度を示す。図中の応答の加速度は、複数のスライス間に存在するタスクの効果のため、低く抑えられていることがわかる。この応答の加速度は 0.68 であり、この負荷状態ではシステムはタスクを処理しきれないため、タスクが時間経過ともに滞っていることがわかる。

負荷を変えて測定した応答の加速度を図 3 に示す。図 3 より、2DB を使った時分割処理では約 1.3、AS を使った時分割処理では約 2.6 が負荷限界あることが判った。これは同様に測定した通常の 2DB、AS それぞれが約 1.1、2.4 であったことより、時分割処理することにより負荷限界が上がるということがわかった。

5 おわりに

本論文では、メッシュ結合並列計算機にパーティショニングアルゴリズム 2D Buddy と Adaptive Scan それぞれを時分割処理に拡張した場合、負荷に対してタスクを滞らせずに処理が行なわれる限界を評価した。シミュレーションにより、時分割処理にすることで通常の 2D Buddy や Adaptive Scan より 1 割から 2 割負荷が強くなることが示された。今後はこのシミュレーション結果を元に実並列計算機上への実装を行なう予定である。

謝辞

本研究の一部は RWC 計画の一環として「超並列システムアーキテクチャに関する研究」で行なわれたものである。関係各位に感謝する。

参考文献

- [1] 須崎, 田沼, 平野, 一杉, 塚本: メッシュ結合並列計算機のパーティショニングアルゴリズム 2D-Buddy または Adaptive Scan を使った次分割処理, 情報処理研究会 OS-65(1994)
- [2] K. Li, K. Cheng: A Two Dimensional Buddy System for Dynamic Resource Allocation in a Partitionable Mesh Connected System, Journal of Parallel and Distributed Computing, vol.12(1991)
- [3] J. Ding, L. N. Bhuyan: An Adaptive Submesh Allocation Strategy for Two-Dimensional Mesh Connected Systems, Proceedings of ICPP(1993)