分散オペレーティングシステムDM-1におけるスレッド分配機構

2H－8

大西祐史† 岡部寿男† 大久保英嗣†† 津田孝夫†
†京都大学工学部情報工学科 ††立命館大学理工学部情報工学科

1 はじめに

分散システムでは、負荷分散を行なうことによってシステム全体の性能を向上させることができる。そのため数多くの研究が行なわれてきたが、プロセス間の通信量を考慮した負荷分散についての研究は少ない。

我々が研究開発している分散OS DM-1では、複数のノードで並列に実行されるプログラムをノード間通信を意識せずに記述することができるため、従来の分散システムよりもネットワークを介したスレッド間通信の頻度が大きくなると予想している。したがってDM-1において負荷分散を行なう際には、ネットワークの通信量を適度な量に保つためにスレッド間の通信量をも考慮すべきであると考える。

本稿では、負荷分散に加え通信量抑制をも行なうスレッド分配機構を提案し、シミュレーションによる評価を示す。

2 分散OS DM-1

分散OS DM-1は、LAN(local area network)によって接続された計算機上に分散システムを構築することを目指して研究開発しているOSである[1]。

DM-1では、メモリ資源の位置透過性をネットワーク上に広がる仮想的な単一の記憶空間（分散仮想記憶と呼ぶ）によって実現している。分散仮想記憶の実体はネットワーク上の二次記憶であり、各ノードの主記憶はキャッシュとして使われる。分散仮想記憶の管理はハードウェアのページング機構を利用することによりページ単位で行なっている。同一ページの複製が複数のノードに存在する場合があるので、ページの整合性制御をwrite-invalidate方式によって行なっている。

DM-1では、資源割当ての単位（タスクと呼ぶ）とプロセッサ割当ての単位（スレッドと呼ぶ）を分割することにより、オーバヘッドの小さい並列処理を行なえるようにしている。一つのタスクに所属する複数のスレッドにそれぞれ異なるサイトのプロセッサを割り当てることを可能にしている。

DM-1における並列プログラムの作成は、一つの仕事や解くべき問題をタスクとして設定してから、並列に実行できる処理は分割し、それぞれ別々のスレッドに割り当てるという手順で行われる。プログラムの並列実行可能な部分の分割は、ユーザもしくはコンパイラがスレッド操作のプリミティブを利用することで行なう。

3 スレッド分配機構

通信量抑制のためのスレッド移送の主な目的としてタスクの応答時間の短縮を考える。つまり、タスク内のスレッドがノード間通信を頻繁に行なっていて1ノードにすべてのスレッドを配置した場合より応答時間が長くなりそうであれば、頻繁に通信し合っているスレッドを1ノードに集めるようなスレッド移送を行なうことにする。

分散させた方が応答時間が長くなる条件を簡単な例について示す。あるタスクの処理量が均等にn個のスレッドに分割でき、それらのスレッドは並列に実行可能で、CPU実行時間はt_eであるとする。今、それらのスレッドをシステムに他のタスクがない状態で実行させたとする。一つのノードにそれらのスレッドをすべて配置して実行させた場合、つまり、ノード間通信がまったく不必要な場合は、応答時間はnt_eである。それに対し、n台のノードに1ノード1スレッドずつ分散させた場合、すなわち、各スレッドの通信時間が最大になる場合の応答時間は、通信時間をt_cとすると、$t_e + t_c$である。それで、分散させることにより応答時間の短縮を図れるt_cの上限は、$(n-1)\,t_e$となる。

そこで、通信量抑制の方式として、最近の一定期間

A Thread Distributor in the Distributed Operating System DM-1
Yuji Ohnishi†,Yasuo Okabe†,Eiji Okubo††,Takao Tsuda†
† Department of Information Science,Kyoto University
Kyoto,606-01,Japan
†† Department of Computer Science and Systems Engineering,Ritsumeikan University
Toujiin Kitamachi,Kita-ku,Kyoto,603,Japan

の通信時間Δt_cとCPU時間Δt_eの比$\Delta t_c/\Delta t_e$が$n-1$ (nはタスク内のスレッドの数) を越えた時、そのスレッドを通信量抑制のために移送する方式を提案する。移送するノードは、そのスレッドが過去一定期間に行なった通信の回数が最も多いノードとする。なお、スレッドが移送された直後には、プログラムや必要なデータがノード上にないためにページ転送が頻発するが、そのような通信によって移送元のノードに再び移送されないように、ノード上に存在していなかったページに対するアクセスのために発生したページ転送[1]にかかる時間はΔt_cには含めないことにする。

負荷分散のアルゴリズムは次のようなものにした。ノードの負荷を過去一定期間の実行可能スレッド数の平均で表す。定期的に起動されるスレッド分配機構が、ノードの負荷を計算し、その値がもしあらかじめ定められた閾値を越えていればスレッド移送を行なう。移送先は、各ノードから定期的に送られてくる負荷情報を用いて、負荷がある閾値以下のノードの中からランダムに選ぶ。移送するスレッドは、最近のCPU利用率がもっとも大きいスレッドとする。

4 シミュレーションによる評価

シミュレーションを行なうにあたって、次のような仮定を置いた。

- 各ノードにおけるスケジューリング方式はラウンドロビンとする。
- 各ノードは1本のバスで結合されている。
- 通信量がページより少ない通信は無視する。例えば、ページの整合性制御のための通信やノードの負荷情報の通知のための通信を無視する。

アクセスパターンが実行途中に変化する例についてシミュレーションを行なった結果を図1に示す。この例は並列化の対象とするループを途中で変えて行列積の計算を行なったもので、最初はスレッド間の通信が少ししか発生しないが、ある時点から頻繁に通信し始めるようにしてある。このグラフは行列の大きさとすべての処理を1スレッドで行なった場合に対する速度向上比の関係を表したものである。図中の“集中配置”は最初にすべてのスレッドを1ノードに配置した場合、“分散配置”は最初に各ノードに1スレッドずつ配置した場合を表す。なお、ノード数およびスレッド数は5である。

負荷分散のみを行なった場合、スレッドを分散させることは行なうが、アクセスパターンが変化して頻繁に通信するようになってもそのままなので、最初に1ノードに1スレッドずつ配置してスレッド分配を行なわなかった場合と処理時間はほとんど変わらない。それに対し、通信量抑制をも行なった場合では、アクセスパターンが変化した時に1ノードに集中させて応答時間の短縮を図るので、結果として全処理を1ノードで行なった場合より速くなっている。

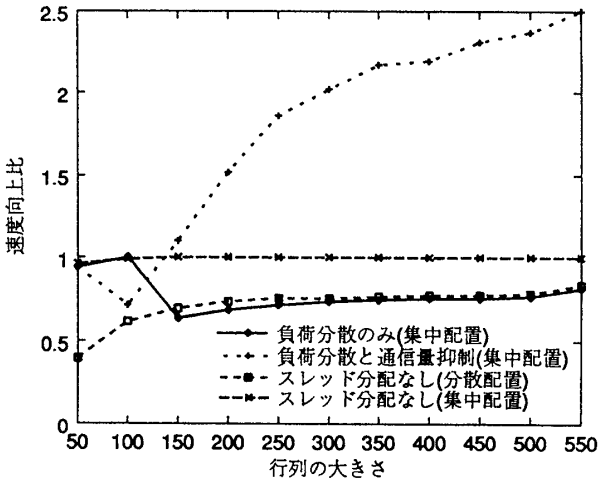

図1 通信量抑制の効果

5 おわりに

本稿では、プロセス間の通信量を考慮しつつ負荷分散を行なうスレッド分配機構を提案し、シミュレーションによってその有効性を示した。なお、あるタスクにおいてそれに属するスレッドの移送が頻繁に起こりスレッドの配置が安定しないという現象 (スレッドスラッシングと呼ぶ) が発生する可能性があるので、それを防ぐ方法も検討したが紙面の都合上省略した。

参考文献

[1] 篠原拓嗣, 藤川賢治, 大久保英嗣, 津田孝夫: 分散仮想記憶に基づくオペレーティングシステムDM-1 の構成, 情報処理学会研究報告 93-OS-61,Vol.93,No.68,pp.49-56,1993.

[1] ページ転送が発生するもう一つの原因は無効化されていたページに対するアクセスである。