5F−8

マルチプロセッサシステムにおけるUNIXの性能評価

友田 正憲　関戸 一紀　本田 亮

(株)東芝 情報処理・機器技術研究所

1 はじめに

近年、ビジネス分野においてもマルチプロセッサシステム上でUNIX[1]が広く用いられるようになってきた。このようなシステムにおいて、大量のデータを扱う場合には、データやプロセスのマイグレートの方法がプロセッサの使用効率や、システムの全体的な性能に影響を与える。

本発表では、データやプロセスに対するプロセッサのスケジューリングの方式が異なるいくつかのシステムをプロセッサの使用効率や、NFS コールのスループットを尺度にシミュレーションにより評価した。負荷がバランス良くプロセッサに与えられる場合には、データおよびプロセスにプロセッサを対応させ、できるかぎり同一のプロセッサに処理させる方式が、プロセスをある優先順位で選択する従来のスケジューリング方式よりもプロセッサの使用効率、システムのスループットともに向上することが定量的に確かめられた。その方法、結果について報告する。

2 プロセッサ割り当ての方針

ファイルサービスやデータベースサービス等、多数のクラアントを相手にするサーバ応用では、個々のサーバプロセスが独立に担当したクライアントの要求を処理するので、数値計算のように頻繁にプロセス間 (プロセッサ間) 通信が行なわれることはない。しかし、数値計算に比べてより大きなデータを取り扱うため、プロセスを実行するプロセッサが変わる (プロセス マイグレーション) とデータのマイグレーションが頻発して、スケーラビリティに限界が発生することが知られている [1]。従来のプロセッサ割り当てでは、データやプロセスに対して処理を行なうプロセッサは特定されていないことが多く、優先順位に従ったプロセススケジューリングや、データがどこまで処理されたかによって、実行を行なうプロセッサが決まる。また、ネットワーク処理では、データを処理するプロセッサが複数同時に存在する場合があり、データが処理されている間に実行するプロセッサが変わる可能性がある。このように、データやプロセスマイグレーションが起こると、プロセッサのキャッシュからデータの転送が起こり、そのオーバヘッドはプロセッサの使用効率に影響を与える。マイグレーションをできる限りおさえるようなプロセッサのスケジューリング方針が重要である。

ここで考える方式では、データやプロセスに対しプロセッサを一台決定する。以降、そのプロセッサのみが処理を行なう。これにより、マイグレーションによるオーバヘッドを回避し、効率良く処理を行なうことができると考えられる。

本論文では、本方式と従来方式を比較し、どの程度性能が改善できるかをシミュレーションにより定量的に評価した。詳細を以下で説明する。

3 処理モデル

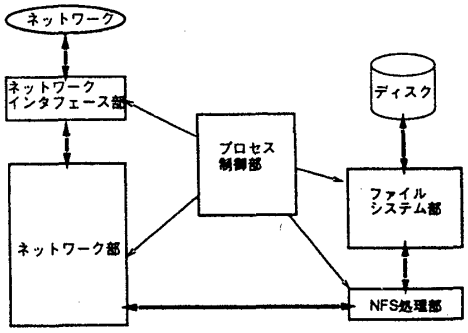

図 1: シミュレータの構成

データやプロセスマイグレーションによるプロセッサの使用効率の低下の様子を、現実的な使用状況で評価するために、大量のデータを扱う NFS(Network File System) サービスを用いた。

まず、OS 内部をソースレベルで解析し、その構造を明らかにした[2]。図 1 は、その結果からカーネルの NFS に関連する構造をおおまかに示したものである。カーネルを、機能別に以下のように分けた。

- ネットワークからのパケットを受けとるネットワークインタフェース部、ネットワークからのパケットを組立て、プロセスに受け渡すまでの処理を行なうネットワーク部
- ネットワーク部から受けとったパケットを解釈し、ファイルシステムへ受けわたす NFS 処理部
- 実際にディスクとのやりとりを行ないファイルシステムの管理を行なっているファイルシステム部
- プロセススケジューリング等の制御を行なうプロセス制御部

特にデータマイグレーションが多く発生するのは、ネットワーク部である。ネットワーク部は、モ

Performance Evaluation of UNIX on Multi Processor System.

Masanori TOMODA,Kazunori SEKIDO,Makoto HONDA

Information Systems Engeneering Lab. TOSHIBA Corp.

[1]UNIX は UNIX System Laboratories, Inc. が開発し、ライセンスしています。

ジュール単位に分けられ、データの処理はこの単位で行なわれる。ネットワーク部を実行するプロセッサが複数存在する場合、データを処理するプロセッサが変わることがあり、マイグレーションが発生する可能性が高いからである。また、プロセスがI/O待ちやロック待ちでスリープする、あるいはコンテキストスイッチが起こった場合は、次に実行可能になったとき、通常は以前に実行したプロセッサに割り当てられるとは限らない。このときにもマイグレーションが起こる可能性がある。

本稿で述べる方式の処理、特にプロセッサの割り当ての方法について以下で述べる。NFS などのネットワークから入力を受けるアプリケーションの場合、扱うデータはNFS コールを格納したネットワークからのパケットである。本方式では、パケット到着の割り込み処理の際にそのパケットを担当するプロセッサを負荷が偏らないように決定する。具体的には、各プロセッサの担当するパケット数を管理し、その数のもっとも少ないプロセッサに割り込みをかけることで決定する。以降、そのパケットの処理は担当するプロセッサのみが行なう。処理が進んで、複数のパケットが組み立てられ一つのパケットになる場合、もっとも負荷の軽いプロセッサが担当となる。さらに、ネットワーク部から、NFS サーバプロセスにパケットが渡されるときには、プロセスよりデータマイグレーションの方がコストが小さいことから、サーバプロセス担当のプロセッサにパケットが引き渡される。また、サーバプロセスからネットワーク部へ渡されるパケットに関しては、ネットワークよりパケットを受けとったときと同様に担当のプロセッサを決定する。

ソース解析に基づく構造を用い、従来方式、本方式の両方のシミュレータを構築した。これには、NFS に必要な計算機資源として、プロセッサ、バス、ディスク、ネットワークインタフェースをモデル化し、組み込んだ。また、マイグレーションのオーバヘッドはプロセスではその固有領域、パケットの場合はそのヘッダ部が主になる。これらは、OS のソース解析、実機での検証などによりその値を求めた。

4 シミュレーション

これまで説明してきた従来方式と、本方式のシミュレーションを以下のように行なった。

プロセッサ数を増やしていき、各々の最大スループットを計測した。また、プロセッサの使用率、マイグレーションが起こった回数、そのオーバヘッドの合計も同時に求めた。

シミュレータへの入力は、READ, WRITE, LOOKUP, GETATR の混合（割合は、実際のファイルサーバより求めた）からなるNFS コール（基本的なリモートファイル操作命令）の連続した系列を与える。シミュレータでは、ネットワークインターフェースまでを実装したので、実際には入力はIP パケットとして分割して表現されたものが与えられる。シミュレータにおいて、入力データはネットワーク部、NFS サーバ部、ファイルシステム部を通り、逆の道筋を通ってコールに対する応答をネットワークインタフェースへ返す。

5 結果

シミュレーションの出力から、各NFS コールごとの応答時間、プロセッサ、バスなどの計算機資源の平均使用率が求められた。以下、各方式について、プロセッサ数とその最大スループットから考察を述べる。図2では、横軸にプロセッサ数、縦軸に最大スループットをとっている。

1. 本方式では、マイグレーションによるオーバヘッドが少ないために、同じプロセッサ数の従来方式に比べて約 20 %スループットが大きい。

2. NFS コールの平均応答時間で比較すると、本方式は 10 %短くなっている。

3. 従来方式のマイグレーションのオーバヘッドの合計は、プロセスによるものとデータによるものがほぼ同じであった。プロセスマイグレーションはデータマイグレーションに比較すると発生回数が少ない。プロセスマイグレーションは、データマイグレーションに比べるとそのコストが大きいので、その発生をおさえることで、効率良くシステム性能をあげることができる。

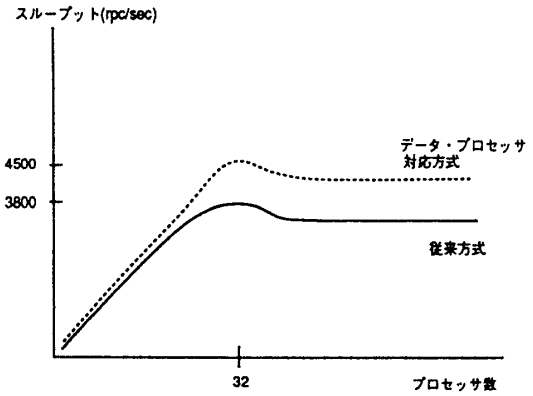

図 2: プロセッサ数とスループット

6 おわりに

本発表では、マルチプロセッサシステムにおけるOS 内部のデータやプロセスのマイグレーションによるオーバヘッドが、どのようにシステムの性能に影響するかをシミュレーションにより評価した結果を述べた。プロセススケジューリングなどの方策を改善することにより、プロセッサの使用効率を高め、システムのスループットを向上させることができることを確認した。しかし、プロセッサ間で負荷の偏りが生じた場合、特定のプロセッサが性能ネックになる可能性がある。今後は、動的な負荷分散を組み込んだ方式のシミュレーション、評価を行なっていく予定である。

参考文献

[1] Thakkar,S. S, et al. Performance of an OLTP Application on Symmetry Multiporcessor System. *17th Annual International Symposium on Computer Architecture.*

[2] 本田 亮, 他. UNIX システムのネットワーク性能評価. 情報処理学会第 45 回全国大会.