B-48 ユーザレベル通信ライブラリによるプロセスマイグレーション
Process Migration with User Level Communication Library

矢澤 慶樹† 堀口 進†
Yoshiki Yazawa Susumu Horiguchi

1. はじめに

今日、大規模な科学技術計算の多くに並列計算機が用いられている。並列計算機利用の普及により、高性能並列計算機に複数のユーザの並列ジョブが投入され、同時に実行されることが多くなっている。

このような環境では、異なるユーザの並列ジョブの一部が同一ノードを取り合って競合的に動作することが少なくない。競合によって並列ジョブを構成する部分プロセスが遅延すると並列ジョブ全体が影響を受けて遅延を生じる。

このため、複数利用者を許可する並列計算機では、部分プロセスの競合による並列ジョブ全体の処理時間遅延を回避する方法が必要である。

本論文ではマルチユーザ大規模並列計算機での処理時間遅延を解決する手法として、ギガビットクラスの通信速度を持つネットワーク装置と通信時のコピーオーバヘッドの少ないユーザ空間通信ライブラリを用いたプロセスマイグレーション機構について検討する。

2. 処理時間遅延の回避手法

同一ノード上でのプロセスの競合による処理時間の遅延を回避する方法として、従来、並列処理の高速化を目的に研究されてきた動的負荷分散手法を応用することが可能である。

一つの解決法は、ユーザがマスタ・スレーブ型の並列プログラム実装を行い、実行中に各ノードの進捗を見ながら割り当てる仕事量を変化させる方法である。

もう一つのアプローチとして、ノード負荷の監視をデーモン等によって行い、負荷の軽いノードが見出された場合、動作中のプロセスを中断し、負荷の軽いノードに移動してから実行を再開する方法がある (図 1)。この方法はプロセスマイグレーションと呼ばれる。

3. プロセスマイグレーション

プロセスマイグレーションによるアプローチは、OSやミドルウェア、ライブラリとしてシステムから提供されるため、透過的な利用が可能でユーザのプログラミング負担が少ない等の利点がある。しかしながら構造が複雑で動作オーバヘッドが大きく、これに見合った性能向上が得にくいため、これまでは有効な手法とは考えられてこなかった。[1]

プロセスマイグレーションにおけるオーバヘッドの最大の原因はプロセスのメモリ空間のコピーであることが知られている。

近年、並列計算機上ではギガビットクラスの高速通信インターフェースが用いられることが多くなっている。これらは主として計算中の値の交換の際に低遅延を実現

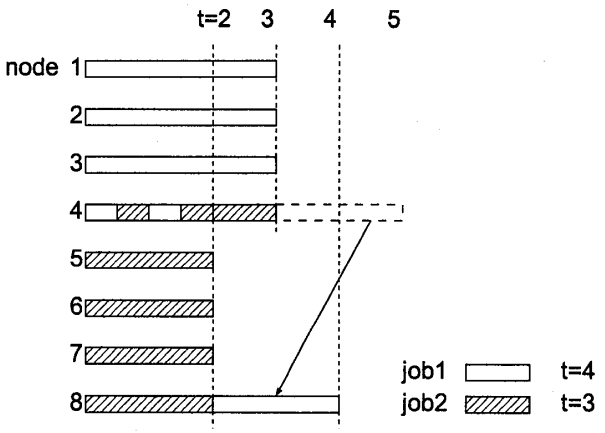

図 1: プロセスマイグレーションによる遅延の解消

することを目的としているが、バンド幅も大きく、データブロックの転送能力も非常に高い。また特に低遅延と高バンド幅が必要なアプリケーションのために、カーネル内のバッファへのコピーを行わず、ユーザ空間から直接ネットワークインターフェースを駆動するユーザレベル通信ライブラリを提供するものもある。

このような通信サブシステムをプロセスマイグレーションでのメモリ空間転送に用いることで、プロセスマイグレーションのオーバヘッドを縮小することが可能であると考えられる。

4. IBM SP/2での実装

4.1 SP Switch と LAPI

IBM SP/2 では、高速通信のためのインタフェースとして、SP Switch と呼ばれるネットワークインタフェースが利用可能である。SP Switch は、最大転送バンド幅 1200Mbit/s のスイッチ型ネットワークで、低レーテンシ、高バンド幅の通信を行うことができる。

SP Switch は、既存のプログラムの動作を目的としたカーネル経由の TCP/IP 通信の他に、LAPI と呼ばれるユーザレベル通信ライブラリから利用可能である。

4.2 メモリ転送性能

プロセスマイグレーション時間を決定する要因はメモリ空間の転送性能である。プロセスマイグレーションに LAPI を用いた場合のマイグレーション時間の短縮はおよそメモリ転送時間の短縮で予測することができる。

図 3 は SP/2 上でメモリ転送を SP Switch 上で LAPI, SP Switch 上で TCP, Ethernet(100Base-Tx) 上で TCP を用いて行い、所要時間を求めたものである。

図から LAPI のメモリ転送性能が高いことがわかる。バンド幅は LAPI で約 443Mbit/s, SP Switch で TCP を

†北陸先端科学技術大学院大学 情報科学研究科, JAIST

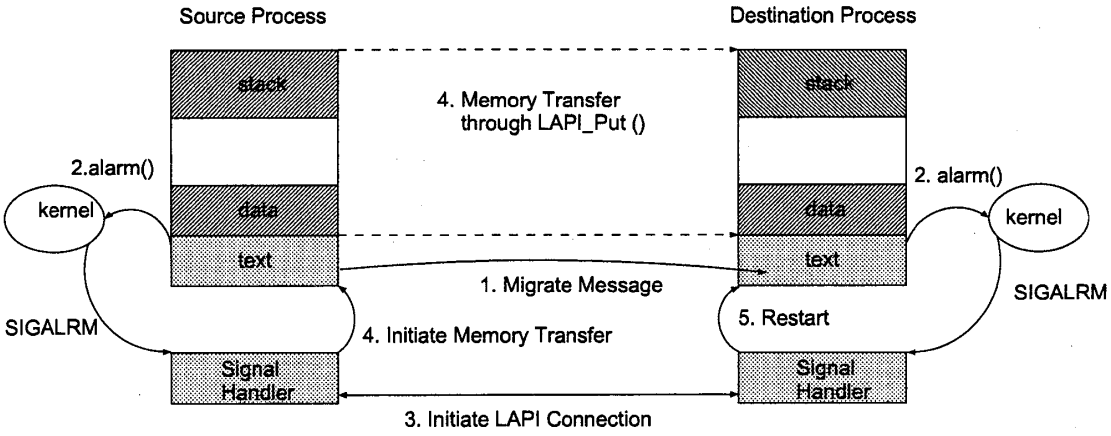

図 2: LAPI によるプロセスマイグレーション

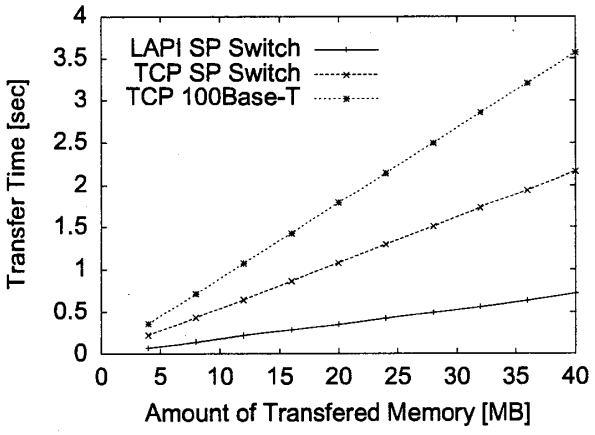

図 3: メモリ転送性能

使った場合 148Mbit/s, Ethernet で TCP を使った場合が 89.7Mbti/s であった。SP Switch で TCP を使った場合のバンド幅が低いが、これは TCP のオーバヘッド及びソケットへの書き込みサイズ制限があり、ループ処理を行っているのが原因と考えられる。

LAPI では 40MB の転送に平均 0.722 秒しか要さないことから、実際のアプリケーションではマイグレーション時間を 1 秒以下に抑えることが可能と予想される。

4.3 プロセスマイグレーションシステムの設計

実行中のプロセスを中断し、他のノードで中断したところから再開するためには、実行状態の再現に必要なプロセス内部状態とプロセッサのレジスタの値を保存することが必要である。プロセス内部状態として、データ領域とスタック領域を取得し、転送する。

現在実装中のプロセスマイグレーション機構の動作を図 2 に示す。このプロセスマイグレーション機構は、LAPI を用いたチェックポイント、プロセス状態の転送、リスタートの動作検証に目的を絞っているため、負荷分散で必要となる外部コントローラ等との協調動作機構は備えていない。

マイグレーション動作部分はシグナルハンドラとして実装される。これはプロセッサの内部状態を保存するためである [2][3]。メッセージでマイグレーション動作が開始されると、転送元・転送先双方のプロセスは SIGALRM によってシグナルハンドラに移行する。シグナルハンドラが新たに LAPI の通信路を開き、LAPI_Put() によってメモリ空間を転送する。転送が終了すると転送先では再始動コードが呼ばれ、実行が再開する。

5. まとめ

本論文ではノードでのプロセス競合による処理時間遅延問題を解決するためにプロセスマイグレーションを行う方法について、検討した。

提案する手法は、従来のプロセスマイグレーションの問題点であったメモリ空間コピーのオーバヘッドを、並列計算機の高速ネットワークとユーザレベルゼロコピー通信ライブラリを活用することによって解決する。

これまでにメモリ転送性能の測定と LAPI の制限に適合したプロセスマイグレーションシステムの設計を終了した。得られたメモリ転送性能から、データ使用量の多いプログラムでも高速にマイグレートすることが可能と予測される。

現在、この設計に基づきプロセスマイグレーション機構の実装を行っている。

参考文献

[1] D.L. Eager, E.D. Lazowska and J. Zahorjan, The Limited Performance Benefits of Migrating Active Processes for Load Sharing, in Proc. ACM SIGMETRICS, 1988.

[2] M. Litzkow, M.Livny, M.Mutka, Condor - A Hunter of Idle Workstations, Proceeding of IEEE 8th International Conference on Distributed Computing Systems, 1988.

[3] G. Stellber, CoCheck: Checkpointing and Process Migration for MPI, Proceeding of IPPS '96, 1996.