5N－7

スレッド移送の実装と スレッドレベル動的再構成システムの検討

鈴木信雄　脇　英世
東京電機大学　工学部

1.はじめに

ネットワーク上で結合されている計算機環境における資源の有効利用法の一つに動的再構成によるものがある[1]。動的再構成は、あるノード上で動作している一連の実行単位の組織の中からユーザの指示により、ある一つの要素を任意のノード上で動作中の組織へ再配置し処理を継続する機構である(図1)。

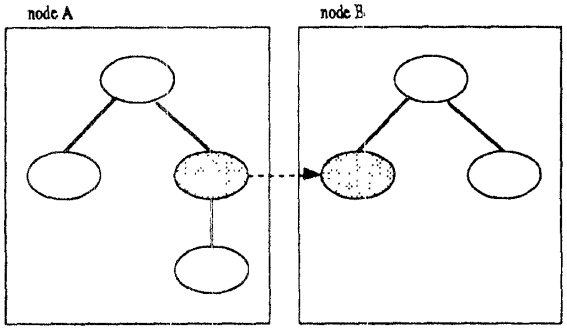

図1. 動的再構成の例

これまで提案されてきた動的再構成はプロセスを単位とする粗粒度のレベルで行われてきた。著者らは、より粒度の細かいスレッドレベルでの動的再構成の実現について検討を進めている。

スレッドレベルの動的再構成を行うためにはスレッドの移送が必要となる。これまで提案されているスレッド移送では、カーネルレベルでの実装が主であり、RPCモデルの改善等に利用されてきた。著者らの実装では、スレッドレベル動的再構成に必要となるスレッド移送をSunOS等の既存OS上のユーザレベルで実現する方法について検討した[2][3]。

本稿では、著者らが行ったSunOS5.3上でのスレッド移送の実装について報告し、本方式を用いたスレッドレベル動的再構成について検討する。

2.スレッド移送の実装

2.1.スレッド移送機構の構成

動的再構成において必要な移送形態は、移送先ノードですでに起動されているプロセス内に、スレッドを任意の時点で組み入れる形である。このような形態を実現するために次のような手順により移送を行う。移送元ノード上のプロセスにおいて、ユーザから移送開始指示を受け取ると、スレッドコンテキストをgetcontext()を用いて保存し該当スレッドを停止する。次に、移送先ノードにスレッドコンテキストを転送し、setcontext()を用いてスレッドの実行を再開する。

2.2.スレッドコンテキストの移送

スレッド移送処理において最も重要な機能は、スレッドコンテキストを適切に保存し、移送先で回復することである。今回の実装では、以下のような手法を用いることによりこれを実現している。

（1）テキスト領域：移送対象となるスレッドのテキスト領域は移送先で組み込まれるプロセスの起動時にすでに存在している必要がある。これに対しては、予想できる再構成の形態の候補を予めいくつか用意しておき、移送先プロセスを生成するソースコード内に予め移送対象スレッドのソースコードを入れておく。これにより、移送先プロセス起動時に移送対象スレッドのテキスト領域を生成することができる。

（2）スタック領域：移送先プロセス内で移送対象スレッド用のスタック領域を用意し、スレッド生成時にこの領域を設定する。移送開始時には移送対象スレッドのスタック領域を移送先ノードへ転送し、移送対象スレッド起動直後にスタック領域の上書きを行う。

（3）データ領域：ローカルデータ領域はスタック内に置かれるため移送先においても参照可能である。静的データ領域は異なるアドレス上にあるため、そのままでは参照不可能である。これに対してはコンパイルの前処理により、移送先プロセス上に予め変数を宣言しておく。ヒープ領域も異なるアドレス上にあるため、そのままでは参照不可能である。これに対応するために、新たにスレッド移送用malloc関数(thm_malloc)を設け、この関数が割り当てられた領域の情報を内部テーブルに保存する。このテーブルには、割当の回数、取得された領域の先頭アドレス、領域の長さ、移送元における先頭アドレス保持変数のスタック内でのオフセット等を保存する。移送先では、

A Realization of Thread Migration and Study of Thread Level Dynamic Reconfiguration
Nobuo SUZUKI and Hideyo WAKI
Tokyo Denki University

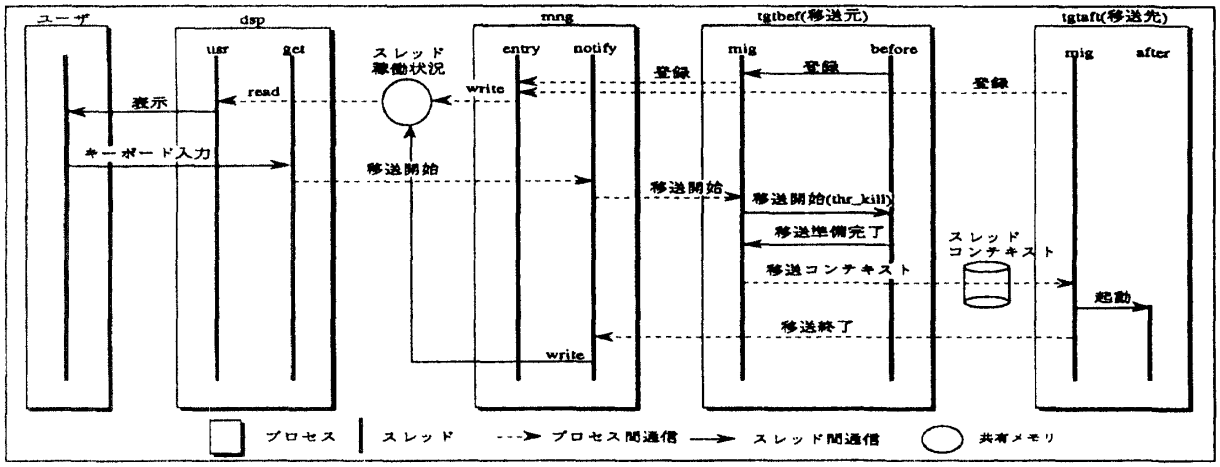

図3. スレッドレベル動的再構成システムの基本通信シーケンス

スレッドのスタック上で、該当する変数をオフセットにより探し出し、新しく領域を割り当てた上でそのアドレスを設定する。

（4）レジスタ類：プログラムカウンタ(PC)は、テキスト領域内アドレスが異なるため移送前後で異なる。スレッドテキストの先頭アドレス(移送処理関数のエントリアドレス)と現在のPCとのオフセットを保持し、移送後のスレッドテキスト先頭アドレスからこれを差引いたものを新PCとする。また、nPC(次に実行される命令のアドレス)の値も同様に変更する。

スタックポインタ(SP)については、移送前のSPとスレッドスタック領域の先頭アドレスのオフセットを、移送後に移送先のスレッドスタック領域の先頭アドレスに加えることで設定する。

関数からの復帰アドレスも設定し直す必要がある。このためには移送先関数のアドレスと移送元関数のアドレスとのオフセットを移送先のスレッドコンテキスト内に保持することで対応する。

3.スレッドレベル動的再構成の検討

（1）構成

スレッドレベル動的再構成システムの構成を図2に示す。各ノード上の実行単位組織については単純な1プロセス1スレッドの場合について示している。

各ノード上には、移送マネージャ・プロセス(mng)があり、移送元プロセス(tgtbef)および移送先プロセス(tgtaft)と連携して移送処理を進める。

（2）移送シーケンス

移送時の基本的な通信シーケンスを図3に示す。ユーザより移送対象スレッドの指定を受けると、自ノードのmngプロセスを経由して移送先ノード上のmngへ通知される。通知と同時にスレッドコンテキストを保存する。移送先ノード上の移送先プロセス内

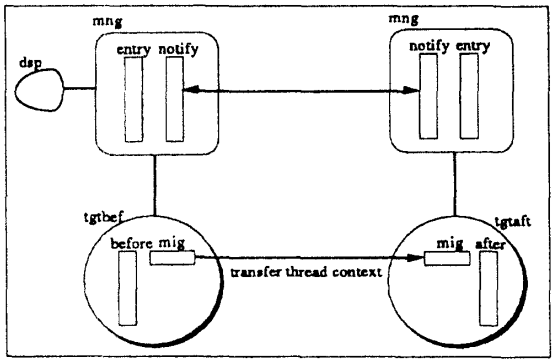

図2. スレッドレベル動的再構成システムの構成

では、mngからの指示により、転送されたスレッドコンテキストを再現し、新しい環境でのスレッドの処理が継続される。

4.おわりに

本稿では、著者らが実装したスレッド移送機構についての概要を示した。本手法を用いてSunOS5.3上に実装し実現性を確認することができた。また、本手法を基にしたスレッドレベル動的再構成の基本構成についても検討した。

今後は、スレッドレベル動的再構成についての詳細な検討および実装を進める予定である。

参考文献

[1]Christine Hofmeister,et al:"Dynamic Reconfiguration in Disributed Systems: Adapting Software Modules for Replacement",IEEE 13th Int. Conf. on Disributed Computing Systems,pp.101-110,May 1993

[2]鈴木,脇:"SunOSにおけるマルチスレッドプロセス移送方式の提案",情処全大5H-5,1995.3

[3]鈴木,脇:"スレッドレベル動的再構成のためのスレッド移送方式の検討",通信学会総大D-56,1996.3