

Mach OS におけるプロセス移送機能の実現と応用*

6C-2

松口 正昭 砂原 秀樹 尾家 祐二

奈良先端科学技術大学院大学

1 はじめに

近年、電子技術の進歩およびネットワーク環境の整備によりインターネットなどのネットワークに接続されたコンピュータの数は急激に増加している。しかしこれらのコンピュータの全てが常に利用されているわけではなく、その中には全く利用されていないものも数多く含まれている。これらのコンピュータを有効に利用して分散処理を行うことができれば、負荷分散や信頼性・可用性など面において性能の向上を図ることができる。この分散処理を効果的に行う機構の1つとして、プロセスをノード間で動的に再配置するプロセス移送がある。これまでの研究において様々なOS上でプロセス移送が実現されている。しかしこれらの中で広く利用されているものではなく、プロセス移送を実現する手法は確立されていない。

そこで、本研究ではユーザ独自のメモリマネージャを作成できるMach OS上にプロセス移送機構を実装する。またプロセス移送における入出力の問題についても考察する。

2 プロセス移送機構の設計

プロセス移送は遠隔実行の場合と異なり、プロセスが実行するプログラムや状態情報を転送しなければならない。これまでのプロセス移送の研究においてプロセス移送機構はカーネル、ユーザ、マイクロカーネルなど様々なレベルで実装されている。本研究では機能性及び拡張性を考慮し、マイクロカーネル上のユーザプロセスとしてプロセス移送機構を実装する。

2.1 Mach

本研究で用いたMachはカーネギーメロン大学で設計された分散オペレーティングシステムである。特徴として、マイクロカーネル、UNIXサーバによるバイナリレベルでのUNIXとの互換性、外部ページャによる仮想記憶機構などがあげられる。本研究では、特に外部ページャを利用することでカーネルへの修正なしにユーザ独自のメモリマネージャを作成することが可能であるということに注目し、Mach上にプロセス移送機能を作成し

た。またMach自身に他ノードとの通信機能がないので、ネットワークを介したタスク間通信をサポートするためにネットメッセージサーバを用いた。

2.2 システム構成

プロセス移送のシステム構成を図1に示す。

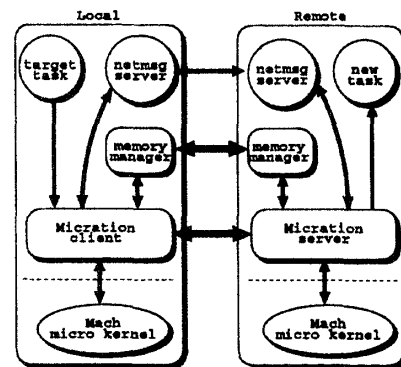


図1: プロセス移送の実装

2.3 プロセス移送の手順

次に、プロセス移送の手順を以下に示す。

1. 移送クライアントはまず指定されたホストの移送サーバと通信を行うため、ローカルのネットメッセージサーバに移送サーバのポートを問い合わせる。
2. ネットメッセージサーバ間の通信によって移送サーバへの送信ポートを得られたら移送するタスクをサスペンドする。
3. 移送するタスクの複製を作成しておく(途中で失敗しても処理が終了しないようにするため)
4. 次に転送先において新しいタスクするように移送サーバに要求し、移送サーバは新しいタスクを作成する(ただしアドレス空間の継承はしない)。
5. 移送クライアントは移送するタスクからアドレス空間を読み込んで、新しく作成したタスク中へ書き込む(一般にアドレス空間の移送は大量のデータ転送となり、長時間に渡ってプロセスを中断させることになってしまうため、プリページングやデマンド

* Implementation and application of Process Migration on Mach. Masaaki Matsuguchi, Hideki Sunahara and Yuji Oie. Nara Institute of Science and Technology(NAIST). 8916-5 Takayama,ikoma,Nara 630-01,Japan

ページングによって最適化したメモリサーバを作成する必要がある)。

6. 残りのタスク、スレッド、ポートに関する情報を取り出し、新しいタスクに設定する。
7. そして移送元のタスクを停止し、転送先でタスクを再開する。

3 実装

実装環境は、イーサネットによって接続された Pentium プロセッサ 90MHz, RAM 32M と i80486DX プロセッサ 33MHz, RAM 8M の AT 互換機である。OS は RT-Mach Keio Zushi version + Lites サーバである。

Mach においてタスクを移送するために必要な情報は、タスクのサスペンドカウント、メモリイメージ、タスク内にある全スレッドのサスペンドカウント、プライオリティ、スケジューリングポリシー、レジスタの値、そしてカーネルによって管理されているポートである。

3.1 カーネルへの修正

Mach のオブジェクトはカーネルポートによって表現され、オブジェクトの動作はそのカーネルポートへの送信メッセージによって行われる。したがって移送後もこれらのカーネルポートを維持しなければ処理を継続することができない。したがってこのカーネルポート扱うためのシステムコールを追加した。

4 今後の課題

今回プロセス移送の実装を行ったが移動できるプロセスは限られており、入出力を行うプロセスに対する処理が不十分であったと思われる。したがってプロセス移送における入出力処理の扱いについて以下で考察する。

プロセス移送において入出力の処理を行うため図 2 に示すようなシステムを考えている。

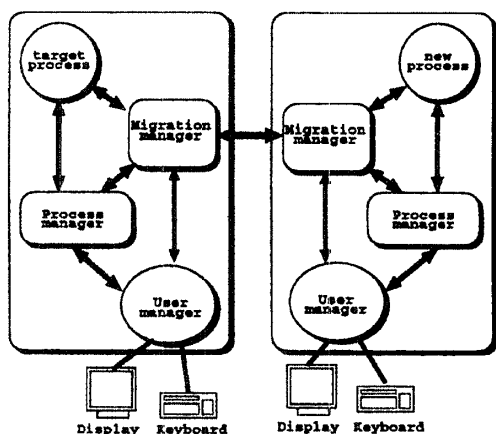


図 2: プロセス移送における入出力

- プロセスマネージャ
自ノードにおいて実行されているプロセスの管理を行う。
- ユーザマネージャ
ユーザからの要求をプロセスマネージャもしくは移送マネージャに伝える。またプロセスからの応答をユーザに提供する。
- 移送マネージャ
プロセス移送を行う。プロセスの移送先の管理を行う。

このシステムにおいて移送マネージャに入出力の処理の受渡しを行わせることで入出力を行うようなプロセスの移送を可能とする。またプロセス移送における入出力の扱いとして以下の 2 つの場合が考えられる。

- 移送元への入出力
プロセス移送によってリモートへ移送されたプロセスの入出力は、基本的には移送元のホストの入出力装置で行いたい。これはユーザからプロセスの移送が意識されないようにするために重要だと思われる。
- 移送先での入出力
次にプロセス移送を利用した分散環境では故障などによって元のホストの入出力装置が利用できない場合やが生じる。このような場合プロセス移送によって送られたプロセスをリモートホストで継続する。

5 まとめ

Mach OS 上にプロセス移送機構の設計し、実装を行った。また、プロセス移送における入出力処理について検討した。

参考文献

- [1] Dejan S. Milojić, Wolfgang Zint, Andreas Dangel and Peter Giese. Task Migration on the top of the Mach Microkernel. In *USENIX Mach Symposium, Santa Fe, New Mexico*, April 1993.
- [2] Open Software Foundation and Carnegie Mellon University. *Mach 3 Kernel Interface Manual*, Keith loepere edition, July 1992. Revision 2.2 NORMA-MK12.
- [3] 前川 守, 所 真理雄, 清水謙多郎. 分散オペレーティングシステム (UNIX の次にくるもの). 共立出版株式会社, 1991.