プロセス移送機能を持つMPIライブラリの構築

3Z－4

岩村 卓成* 中村 嘉志† 多田 好克‡
電気通信大学 大学院情報システム学研究科§

1 はじめに

本発表では、MPI [1] とその一実装であるMPICH [2] を対象としたプロセス移送機能の追加方法について議論する。本システムは、移送時にプロセス間の通信路を再確立することによって、汎用実装で生じていた通信遅延時間の悪化を防ぐ。また、進歩の著しい負荷分散アルゴリズムを移送システムの仕組みから切り離すために、MPIと親和性の良い移送インタフェースを定義する。これによって、プログラマはMPIプログラムとして移送スケジューラを実装でき、自律的に移送する計算プログラムをMPIの枠組の中で作成できる。

2 動機

学校や企業のオフィスに導入されるコンピュータは短時間では高負荷になるが、長期的に見た場合には低負荷であることが知られている。こうした休眠ワークステーション (以後、休眠WSと省略) はいわば隠れた計算機資源として、主に金銭的コストの観点から有効利用が望まれている。

休眠WSを有効利用するための方法として、プロセス移送機能が考案された [3]。しかし、メッセージパッシング型の並列計算のような頻繁な通信を伴う計算を汎用のプロセス移送機能で移送する場合、従来の実装ではスタブによる通信路のオーバヘッドのために性能が劣化する。そこで我々はMPIライブラリに移送機能を付加し、移送時に通信路を再確立することで問題解決を目指した。

Attaching Process Migration Facility to MPI Library.
*Takashige Iwamura
†Yoshiyuki Nakamura
‡Yoshikatsu Tada
§Graduate School of Information Systems, The University of Electro-Communications.

3 本システムについて

本システムのイメージを 図1に示す。図中の斜線の部分が本研究で実装する部分である。

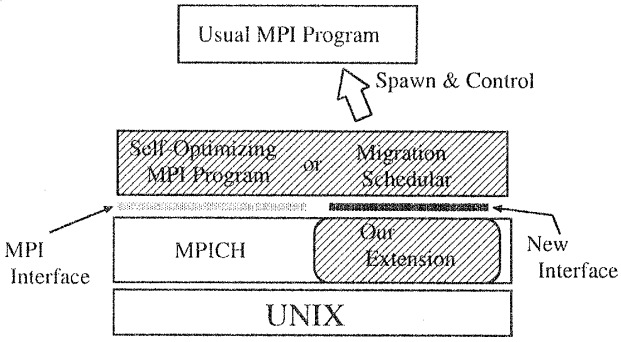

図 1: 本システムのイメージ

プロセス移送に必要な機能はMPICHを改造して組み込む。組み込む機能はコンテクスト保存と復元機能、実行環境取得のための処理部分、ユーザに提供する移送のためのインタフェースである。移送スケジューラや自律最適化機能を持つMPIプログラムは、我々が新しく定義したインタフェースとMPIを用いて構築される。

移送スケジューラは移送タイミングと移送先を決定するプログラムで、MPIのプロセス生成関数`MPI_COMM_SPAWN`でMPIの計算プログラムを生成し、移送タイミングをコントロールする。移送スケジューラを利用して負荷分散する場合、プログラムは自動的に移送されるため、コントロールされる計算プログラムはソースコードを変更する必要は無い。

自律最適化機能を持つMPIプログラムとは移送スケジューラの能力を内部に持つMPIプログラムである。プログラムに特化した負荷分散アルゴリズムを実装できるため、移送スケジューラより効率的に負荷分散を実装できる。

4 移送のためのインタフェース

移送スケジューラをMPIプログラムとしてポータブルに実装するために、新しくインタフェースを追加する。追加するインタフェースは移送のためのインタフェースと実行環境情報取得のためのインタフェースで構成される。これらとMPIの関数を利用することで、移送スケジューラや自律的最適化機能を持つMPIプログラムを作成できる。

4.1 移送インターフェース

移送関数は、移送するプロセスと移送先ノードを引数とする。移送関数は全てのMPIプロセスから呼び出せる。移送拒否を示す変数を真にしたプロセスは移送されない。

研究初期、我々はプロセスの指定方法をMPIのコミュニケータとランクの組合せで考えていたが、移送スケジューラから全てのプロセスを指定できない場合があることが判明した。そこで、プロセスID *を別途に定義して、プロセスを指定する方法を用意した。

この他に、移送関数にはアノテーションや複数のプロセスを同時に移送させるための関数を定義し、プロセス移送によるプログラムの停止を少なくするようにインタフェースを設計した。

4.2 実行環境情報の取得

ノンブロッキング関数を利用して情報取得とその他の処理の重畳を可能とするために、MPIの通信関数を用いて実行環境情報を取得するデザインを採用した。そのために、実行環境と通信するためのruntimeコミュニケータを取得、破壊する関数と、情報の受渡しのためのデータフォーマットを新たに定義した。

4.3 動作

移送スケジューラがMPIプログラムを移送する時の動作を図2に示す。

1. 移送スケジューラはruntimeコミュニケータを生成した後に、MPIの通信関数で実行環境情報を取得する。

2. 移送スケジューラは取得した実行環境情報を元にして、組み込まれた負荷分散アルゴリズムに従い移送タイミングと移送先ノードを選択する。

3. 移送スケジューラは移送したいプロセスを引数にして移送関数を呼び出す。

4. 指定されたプロセスは本来の処理を一時停止して、指定されたノードへ移動する。

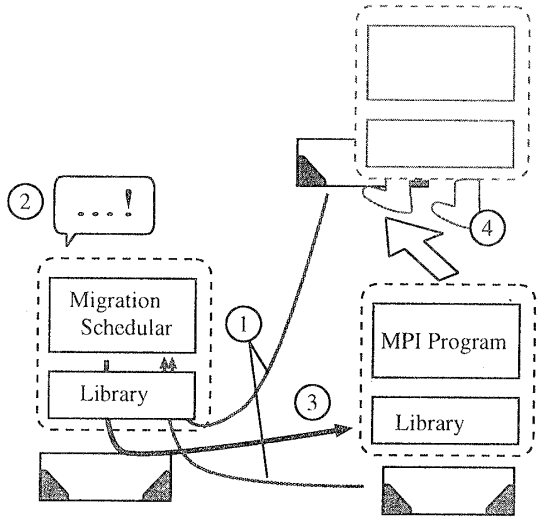

図2: 移送スケジューラを用いた移送の動作例

5 まとめ

MPICHに対するプロセス移送機能の追加手法について提案した。また、移送スケジューラをMPIプログラムとして実装するためのインタフェースの設計方針を説明した。

現在、プロセスコンテクスト保存、復元ルーチンを完成させ、MPICHとの統合を行っている。今後は、実装を完成させてさまざまな負荷分散アルゴリズムを実験評価し、既存のアルゴリズムの問題点を明らかにする予定である。

参考文献

[1] M. Snir et. al., MPI -The Complete Refecence, Vol. 1, The MPI Core, 2nd Ed. , MIT Press, 1998.

[2] W. Gropp et. al., A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard , *Parallel Computing*, Vol. 22, pp. 789-828, 1996.

[3] M. Themer et. al., Preemptable Remote Execution Facilities for the V-System, *Proc. 10th Symp. Operating System Principles*, Dec. 1985, pp. 2-12.

*計算に参加する全プロセスに一意に付けられる番号である。Unixのプロセス IDとは異なる。