2H-07　プロセス移送機能を持つMPIライブラリにおける負荷分散機構の実装

岩村 卓成† 中村 嘉志† 楯岡 孝道‡ 多田 好克†
†電気通信大学 大学院情報システム学研究科　‡電気通信大学 情報工学科

1　はじめに

本発表では, 我々が実装中のプロセス移送機能付きMPIライブラリにおける負荷分散機構 (以後, 負荷バランサと呼ぶ) の実装方法について述べる. 並列プログラムの実行時間の最適化を負荷バランサの目的とした場合, 様々なプロセス配置における実行時間をモデル予測することになる. しかし, 全ての並列プログラムを唯一のモデルで表現することは難しい.

そこで我々は,MPIに新しいインターフェースを加える事で, 負荷バランサを通常のMPIプログラムとして記述できるようにした. これによって, 開発者はスケーラブル, プログラマブル, ポータブルな負荷バランサを容易に実装可能となる.

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と略記) はいわば隠れた計算機資源として, 有効利用が望まれる.

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[1]. 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を用いているために通信性能が劣化する. 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.

An Implementation of Load Balancer for Process Migration System
Takashige Iwamura†, Yoshiyuki Nakamura†, Takamichi Tateoka‡, Yoshikatsu Tada†
†Graduate School of Information Systems, The University of Electro-Communications.
‡Department of Computer Science, The University of Electro-Communications.

3　システム概要

本システムは,MPICH-p4を拡張実装した, ユーザーレベル実装のプロセス移送機能付きMPIライブラリである[2]. 図1にライブラリの概観を示す. 図中の網かけ部分が我々の拡張部分である.

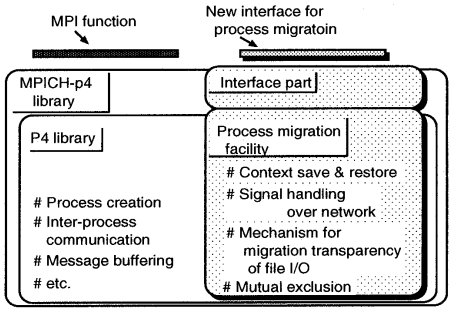

図1: 本ライブラリのイメージ

MPICH-p4は, 実際に通信機能を持つp4ライブラリと, p4の通信関数を用いてMPI関数を提供する部分に分かれる. 我々は,p4に対してプロセス移送機能を組み込み, そこで定義した関数を元に移送のためのインターフェースを提供した.

ライブラリの拡張部分にはプロセス移送に必要な部品を用意し, プロセス移送はこれらの部品とMPICH-p4の関数の一部を利用する事で実現した.

本システムはMPICH-p4とほぼ同じ方法で利用可能である. ユーザはソースコードのmain関数の始めに僅かな記述をするだけで, ソースコードの大規模な変更は必要ない. MPICH付属のコンパイル用スクリプトでコンパイル, リンクし, MPICH付属のmpirunコマンドで実行ができる.

プロセス移送は, 新しいインターフェースで定義した移送関数によって開始される. 通常, この移送関数を呼び出すのは負荷バランサと呼ばれるMPIのプロセスである.

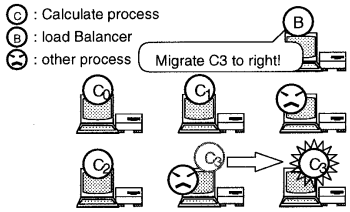

図 2: 負荷バランサ

4 負荷バランサの実装方法

負荷バランサとは, 計算機の実行性能が動的に変化する実行環境において, プロセスの再配置を行いながら並列プログラムの実行速度を高速に保つプロセスである (図 2). 負荷バランサは, 以下の処理を繰り返す.

1. 実行環境や制御対象のプロセスについて情報を集める
2. 様々なプロセス配置でのプログラムの実行時間をモデル予測する
3. 良い配置を見つけた場合, 移送関数を呼び出し, 各プロセスを再配置する

しかし, 実行時間を予測するモデルは並列プログラムによって異なり, 単一のアルゴリズムでは全てのプログラムを適切に表現できない. 効率を追求するためには, よりアプリケーションに特化した負荷バランサを使用する必要がある. そこで我々は, 負荷バランサ自身をMPI プログラムとして記述できるようにした.

負荷バランサの開発者は, MPI と我々が新たに定義したインターフェースで, 通常の MPI プログラムと同じように負荷バランサをコンパイル, 実行する.

負荷バランサの利用方法は, 並列プログラムに負荷バランサをあわせて実行する方法と,MPI-2 のプロセス生成関数を用いて負荷バランサが計算プログラムを生成する方法がある.

4.1 提供するインターフェースの概要

インターフェースは, 移送関数, 情報取得関数, ヒント関数, 保護, その他の 5 つに分類できる. インターフェースはライブラリに固有な機能への依存を少なくし, 負荷バランサのソースコード可搬性を高めるようにした.

- 移送関数:
  指定したプロセスを指定したホストへ移送する関数
- 情報取得関数:
  プロセスと実行環境から情報を取得する関数. 取得できる情報を表 1に示す.

表 1: 取得可能な情報の一覧

ホストから	CPU 速度, ロードアベレージ, バンド幅, 通信遅延, メモリ量, 直接通信可能ホストの一覧
プロセスから	使用メモリ量, rusage 構造体の情報 (user time, system time 等), ヒント関数の値

- ヒント関数:
  負荷バランサにとって有用な情報を MPI プログラマに提供してもらうための関数. 制御対象の MPI プログラムに埋め込んでもらう. ヒントは, プロセスが要求する処理速度, 頻繁に通信するプロセスのリストの 2 つを指定できる. 値の解釈は負荷バランサ依存である.
- 保護:
  計算プロセスが, 移送を拒否する仕組みを提供する.
- その他:
  MPI では不足する部分を補う関数を定義する.

本研究で定義した取得可能な情報やヒント関数は, 全ての要求を満たすものではない. これ以上の要求は MPI の Profiling インターフェースを用いるか, 制御対象プログラムのソースコードを変更し, 明示的に通信関数で負荷バランサに情報を送る必要がある.

5 終わりに

本発表では, 負荷バランサを通常の MPI プログラムとして記述するための方法について述べた. 現在, 本ライブラリのプロセス移送の基本部分が完成している. これと並行して, 情報取得機構の実装を進めている. 今後は,BSP モデルを利用した負荷バランサを実装し, 本システムの有効性を検証する予定である.

参考文献

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

[2] 岩村ほか, プロセス移送機能を持つ MPI ライブラリの構築, 情報処理学会 第 59 回全国大会 論文集 (1), pp. 145-146, Sep., 1999.