

プロセッサのグループ化による 超並列プログラム実行の可視化*

品野 竜太
筑波大学理工学研究科†

山下 義行
筑波大学 電子・情報工学系‡

1 はじめに

並列プログラムの実行状況の把握を容易にすることを目的とした可視化の研究が行われているが、プロセッサ台数が増えると表示が複雑になり情報が読み取れないなどの問題が生ずる。このため、超並列計算機のプログラムを可視化するには、プロセッサ台数が増えても対応できるようなスケーラブルな可視化 [1] が必要となる。

本研究では、超並列計算機 CP-PACS[4] 上で動作する MPI[2] プログラムを対象に、超並列プログラムの実行状況を、プロセッサをグループにまとめることにより、より少ない面積で可視化するシステムを試作した。

2 プロセッサのグループ化

並列計算機上でプログラムを実行すると、各プロセッサごとにその受け持つ役割やプロセッサ間通信時の各種条件などにより挙動の違いが生じる。プロセッサ台数が多い場合、プロセッサ一つ一つの情報を表示するのは画面上の制約が大きく現実的ではないため、全体の状況を表示するには統計的な情報が使われる。しかし統計的な情報ではパフォーマンスチューニングやデバッグでのポイントとなる局所的な情報は得られにくい。そこで、プログラム実行時の情報を元に、実行状況の特徴を分類してプロセッサを複数のグループに分けることを考える。

一つのグループには一つ以上のプロセッサに関する実行時の情報を格納する。可視化の際には複数のプロセッサに関する情報をまとめて表示するため、グループ化を行った結果として表示の際の情報量は減る。可視化ツールがスケーラブルなものであるためには、グ

ループ化された情報が必要に応じて各プロセッサごとの詳細な情報に拡大できなければならない。

3 実装

実装は、MPIプログラムのプロファイリングを超並列計算機 CP-PACS (SR-2201) を対象に行い、グループ化表示を行う可視化ツールを Java 言語にて記述した。ユーザは、プロファイリング用ライブラリをリンクしたプログラムを実行することで実行記録を採取し、そのデータファイルをツールに読み込ませることで可視化を行う。

現在の実装では、「実行時間の処理別の内訳」「送信・受信プロセッサの対応関係」「プロセッサやグループの処理の流れ」を可視化することができる。

3.1 グループ化表示の例

図1は、256台のプロセッサを用いて3段のマスター・スレーブ型の並列計算プログラムを実行してその記録を取り、一つのサブマスターが管理するスレーブプロセッサを一つのグループとして可視化した例である。一本のバーは一つのグループの実行状況を示している。横軸は時間である。バーの色の濃淡は、グループ内で通信を行っているプロセッサの割合を示している。また、必要に応じてグループ間の通信の状況も表示可能である。図1を見ると、グループごとに実行状況に特徴が出ていることがわかる。

グループ化表示からより詳細な情報を読み取りたい場合には、バーをマウスでクリックすることにより、グループ内のプロセッサ一つ一つの詳細な実行状況が表示される(図2)。

グループ化を行う利点の一つとして、全プロセッサの実行状況の特徴を分類した場合に、それを明確にしつつより少ない面積で表示できることがあげられる。全体の概況を表示する方法として縮小表示が多く用いら

* Visualization of Massively Parallel Program by Grouping Processors

† Ryuta Shinano: Master's Program of Science and Engineering, Univ. of Tsukuba

‡ Yoshiyuki Yamashita: Institute of Information Sciences and Engineerings, Univ. of Tsukuba

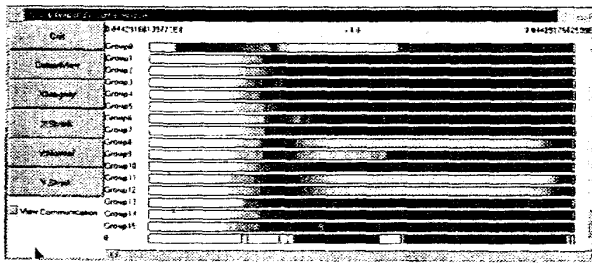


図 1: グループ化した可視化例

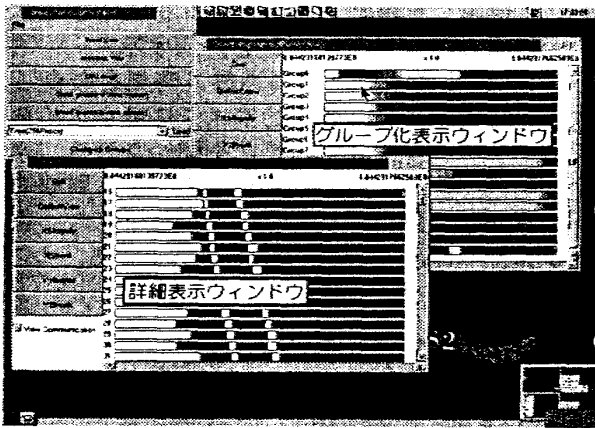


図 2: グループの展開

れるが、プロセッサ数が増えた場合、ある特徴を持つプロセッサが他に埋もれてしまう可能性がある。グループ化表示の場合は、特徴ごとに分類してまとめて表示することにより、注目すべき特徴を見逃さずに全体の概要を知ることができる。

4 自動グループ化の検討

可視化が有効に働くためには、どのようにグループ化を行うかが重要な問題となる。現在の実装では、グループ化に関する情報はすべてユーザが与えるものとしている。このため、「ユーザがプログラムのモデルにしたがってグループ化情報を与え、可視化を行うことによって動作の確認及びバグやパフォーマンスチューニングのヒントを得る」という使い方になる。しかしツールが自動的、あるいは半自動的にグループ化を行ってプログラムの動作の特徴を分類できれば、ユーザにとって問題解決のための助けとなることが期待できる。

実行時の情報をもとに自動的にグループ化を行う処理は、「グループ数を決定し」「グループに属するプロセッサを決定する」クラスタリング [3] の問題とするこ

とができる。

自動的にグループ化を行うために利用する情報としては、

1. プロセッサ間通信の頻度
(高いもの/低いもの同士をまとめる)
2. プロセッサの状態変化のパターンの類似性
(類似性の高いものをまとめる)

などがあげられる。これらの情報をもとにプロセッサ間の距離を定義し、クラスタリングを行うことによってグループ化を行う予定である。

5 まとめ

超並列計算機での並列プログラムの実行状況を可視化するために、プロセッサのグループ化による表示を提案し、可視化システムを試作した。

現在、自動グループ化に関して設計・実装を進めている。今後は、より多くの表示形式に関してグループ化のサポートを付加し、実用的なシステムとすることが課題としてあげられる。

参考文献

- [1] Alva L. Couch. Categories and context in scalable execution visualization. *Journal of Parallel and Distributed Computing*, No. 18, pp. 195-204, 1993.
- [2] Message-Passing Interface Forum. *MPI: Message-Passing Interface Standard*, 1995.
- [3] 長尾真. パターン情報処理. 電子通信学会大学シリーズ. コロナ社, 初版, 1983.
- [4] 中澤喜三郎, 中村祐一, 朴泰佑. 超並列計算機 cp-pacs のアーキテクチャ. *情報処理*, Vol. 37, No. 1, pp. 18-27, Jan. 1996.