

VPMT: 分散メモリ型並列計算機のための 2P-9 並列プログラム可視化システム

釘村 廣樹[†] 宮下 浩^{††} 梅尾 博司[†]
[†](大阪電気通信大学) ^{††}(コンカレントシステムズ)

1. はじめに

近年、集積回路技術の向上により、プロセッサ数が数千～数万個にもなる超並列計算機が実現してきた。しかし、現在の並列プログラミング環境はよいとはいえない。その理由の1つとして、デバッグの際個々のプロセッサの動作を知ることが重要であるが、それを知るためのツールが少ないためである。そこで、並列プログラム開発の支援を目的として並列プログラム可視化システムを開発した。この可視化システムは、トランスピュータネットワーク上で動作する(図1、写真1)。これにより、停止プロセッサの特定やその原因の予測などのデバッグや、プログラムのチューンアップ作業を容易に行うことができる。

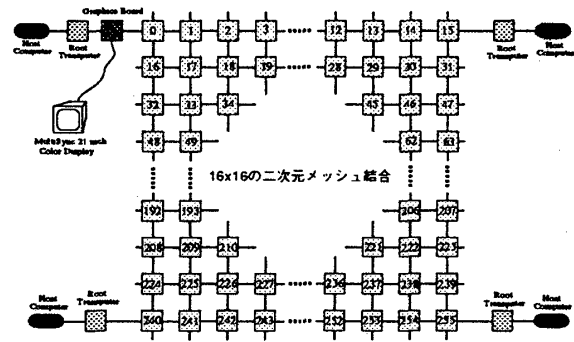


図1. トランスピュータネットワーク

2. 可視化システムの概要

並列プログラム可視化システムは、「前処理」「初期化」「モニタリング」「データ回収」「表示」の5つから成り立っている。それぞれの役割は以下の通りである。

・前処理

モニタプログラムをユーザプログラムのネットワーク記述ファイルに付加する。ネットワーク記述ファイルを変更すると、再コンパイルが必要となる。

・初期化

コンパイルが終了すると、プログラムを実行する前に、状態モニタとリンクモニタのデータを記録するためのメモリ領域を確保しなければならない。この領域は、ローカルプロセッサのメモリの内の後部100kバイトである。

・モニタリング

再コンパイルされたユーザプログラムを実行し、その動作をモニタする。

・データの回収

実行されたユーザプログラムのモニタリングデータは、回収プログラムによって、ホスト計算機上のハードディスクに回収される。

・データの表示

回収したデータを一連の動画として表示する。

VPMT: Visual Performance Monitoring System for a Large-Scale of Transputer Networks

[†] Osaka Electro-Communication University

^{††} Concurrent Systems Limited



写真1. トランスピュータ並列計算機

3. モニタ

モニタは、ユーザプログラムの実行状況を記録する。並列プログラム可視化システムには、プロセッサ状態モニタと通信リンクモニタの2種類がある。

3.1 プロセッサ状態モニタ

トランスピュータには、実行状態とアイドル状態の2つの状態が存在する。実行状態とは、トランスピュータが計算を行っている状態のことである。アイドル状態は、トランスピュータの全てのプロセスが通信待ちになっている状態のことである。

プロセッサ状態モニタは、トランスピュータが実行状態であるか、アイドル状態であるかを調べる。このモニタは、アイドル状態から実行状態に、また、実行状態からアイドル状態に切り替わったときのタイマ値をとり、その結果をトランスピュータのローカルメモリに格納する。プロセッサ状態モニタのデータは32ビットで表現され、32ビット全体でグローバルクロックを表す。

3.2 通信リンクモニタ

通信リンクモニタは、トランスピュータの入出力リンクをモニタする。このモニタは、各リンクの入出力の状態を調べ、その結果をトランスピュータのローカルメモリに格納する。通信リンクモニタのデータは32ビットで表現され、上位8ビットでリンクの入出力の状態を表す。そして、残りの24ビットでグローバルクロックを表す。

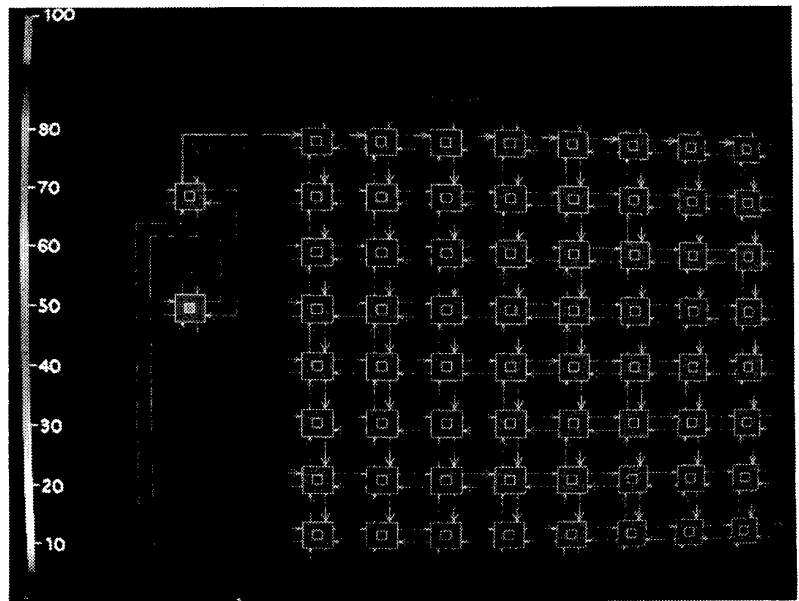


写真2. 可視化画面1

4. 処理効率

4.1 処理効率の計算

各プロセッサの処理効率は、プロセッサが実行状態にあった時間をある一定期間で割ると求められる。また、全プロセッサの処理効率は、各プロセッサの処理効率の平均として表される。求めた処理効率は、それに応じた色によってディスプレイに表示される。

4.2 処理効率のグラフ化

並列プログラム可視化システムは、特定のプロセッサの処理効率のグラフを表示する。特定のプロセッサは、可視化画面中でマウスを操作して選択することができる。グラフを用いると、1%の効率の違いをよりはっきりと表現でき、ユーザプログラムのチューンアップに役立てることができる。

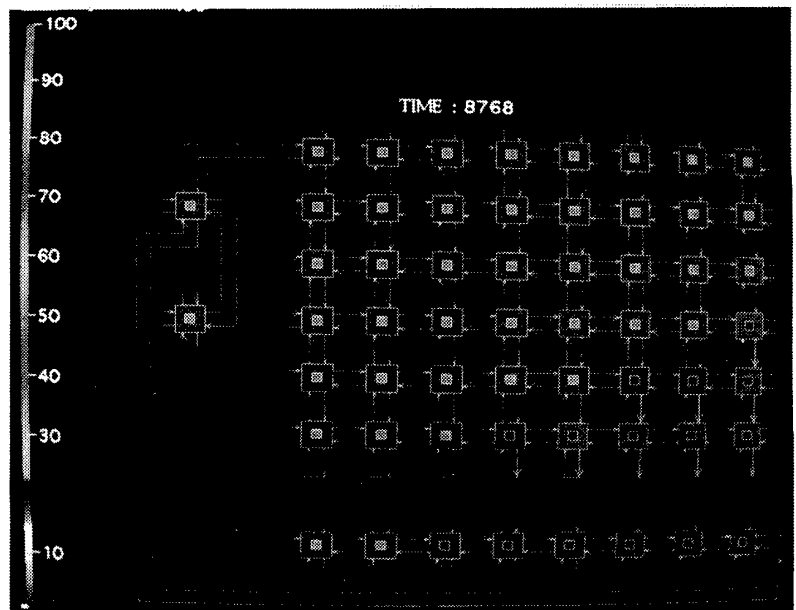


写真3. 可視化画面2

5. 結論

並列プログラムを作成する上で、各プロセッサがどのように動作しているか、処理効率がどれほどかなどを知ることが重要である。そこで、並列プログラム可視化システムを使ってプロセッサの動作をモニタすることにより、各プロセッサがどのように動作しているかを理解しやすくなった。その結果、ユーザプログラムが停止しても、その原因を把握しやすくなりデバッグを容易に行うことができる。また、処理効率の表示を色とグラフを併用することによって、特定のプロセッサにおける1%の効率の違いをよりはっきりと示すことができる。よって、プログラムのチューニングのときに有益である。

参考文献

- [1] 宮下浩, "並列プログラム可視化システムの開発とその応用", 大阪電気通信大学修士学位論文 (1995.2)
- [2] R.S.Cok; "Parallel Programs for the Transputer", Prentice Hall, 1991. 梅尾 (監訳), 松田, 丸山, 山本 (訳); "Transputer/occam による並列プログラム入門", 共立出版, 1993.
- [3] T. G. Moher; "Provide: A process visualization and debugging environment", IEEE Trans. Software Eng., Vol. 14, No. 6, pp. 849-857, June, (1988).
- [4] H. Burkhart and R. Millen; "Performance-measurement tools in a multiprocessor environment", IEEE Trans. on Computers, Vol. 38, No. 5, pp. 725-737, May, (1989).