

VPMT:分散メモリ型並列計算機のための 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

〒572 大阪府寝屋川市初町 18-8

^{††}(株) コンカレントシステムズ

〒550 大阪市西区江戸堀 1-22-4

効率のよい並列プログラムの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 VPMT を開発した。VPMT は、トランスピュータ並列計算機上で動作し、任意のユーザプログラムを可視化することができる。VPMT を使用することにより、並列プログラムの動作を視覚的に捉えることが可能になる。また、プロセッサでの処理効率を計算し、色情報でユーザに処理効率の情報を提供することができるので、ユーザの記述した並列プログラムの処理効率を向上させる指針となる。

本稿では、並列プログラム可視化システム VPMT の構成要素とその応用について述べる。

VPMT: Visual Performance Monitoring System for Transputer Networks

Hiroki Kugimura[†] Hiroshi Miyashita^{††} Hiroshi Umeo[†]

[†]Osaka Electro-Communication University

18-8, Hatsu-cho, Neyagawa 572, Japan

^{††}Concurrent Systems Ltd.

1-22-4, Edobori, Nishi-ku, Osaka 550, Japan

This paper describes our visual performance monitoring system VPMT, developed for visualizing behaviors of any parallel programs executed on Transputer networks. Design concept of the VPMT, its implementation on out 256-Transputer systems, and some applications showing its effectiveness are given.

1 はじめに

並列計算機は、多数のプロセッサとそれらをつなぐ通信リンクから構成されており、高速演算が可能である。しかし、並列プログラミング環境はよいとはいえ、並列計算機用のプログラムの記述やデバッグは非常に困難である。特にデバッグの時、通信によるデッドロックやボトルネックは、プログラムを一見しただけではわからない。そこで、並列計算機でのプログラミングを支援するためにいろいろな可視化ツールが開発されている [2][3]。我々の開発している VPMT もその 1 つである。

本稿では、我々が VPMT と呼んでいる並列プログラム可視化システムについて述べる。VPMT は、トランスピュータネットワーク上で動作し (図 1, 写真 1), ユーザの記述した任意の並列プログラムを可視化するシステムである。これを使用すると、並列プログラムの動作の確認や、停止プロセッサの特定とその原因の予測が可能になる。また、処理効率を色情報で示すので、プログラムのチューニングを容易に行なえる。

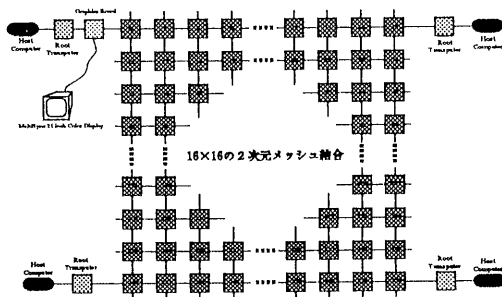


図 1 トランスピュータネットワーク

2 VPMT の概要

VPMT は、次に示す 5 つの部分で構成されている。

● 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。モニタプログラムの付加は自動的に行なえるが、ネットワーク記述ファイルを変更するとプログラムの再コンパイルが必要になる。前処理に関しては、3節でくわしく述べる。

● 初期化

ユーザプログラムを実行する前にモニタリングデータの格納領域を初期化する。モニタリングデータの格納領域は各ローカルメモリの後部 100KB を使用する。



写真 1 トランスピュータ並列計算機

● モニタリング

ユーザプログラムを実行し、その動作をモニタリングする。もしユーザプログラムが正常に終了しなくても、プログラムが停止するまでの動作はモニタリングできる。

● データの回収

モニタリングデータをホストコンピュータに回収し保存する。

● 可視化

回収されたデータに基づいてユーザプログラムの可視化を行なう。可視化画面には、動画として表示するものと数値的評価を表示するものがある。可視化に関しては、4節でくわしく述べる。

3 VPMT の前処理

3.1 グローバルクロックの設定

トランスピュータ並列計算機は、すべてのプロセッサが独立したクロックを持っている。しかし、VPMT は各プロセッサの動作状況を時系列データとして保存するため、各プロセッサの持つクロックのままではモニタリングを行なうと、プロセッサ間の通信の様子などの関連性を抽出することが困難になる。そこで、各プロセッサの動作状況をモニタリングする前に全プロセッサで統一された、グローバルクロックを得ることが必要になる。

2つのトランスピュータは、通信を行なうことにより両者間のクロックを近似的に一致させることができる。

¹ここで近似的という言葉を使ったのは、両者間で通信を行なった場合、送信側と受信側で同時にその処理が終わるものと仮定しているためである。

VPMTでは、ルートプロセッサのクロックを基準としてグローバルクロックの設定を行なう。サブプロセッサは、通信をしながらルートプロセッサのクロックに合わせる。また、グローバルクロックの設定時間を短縮するために生成木を並列に構成しながら、その設定を行なう。

3.2 モニタプログラム

ユーザプログラムの実行状況は、モニタプログラムによって記録される。VPMTのモニタプログラムには、プロセッサの状態を記録するプロセッサ状態モニタと、通信リンクの入出力の様子を記録する通信リンクモニタがある。

3.2.1 プロセッサ状態モニタ

プロセッサ状態モニタは、トランスピュータが仕事待ちになっているアイドル状態なのか、なんらかの仕事をしている実行状態であるのかを観察する。プロセッサの状態は、低優先度待ち行列レジスタの内容を調べるとわかる。もし、プロセッサがアイドル状態であれば、レジスタの内容は一定値(#80000000)となるが、実行状態であれば、レジスタの内容は#80000000以外の値となる。プロセッサ状態モニタは、状態が切り替わったときのグローバルクロック値を、各プロセッサのローカルメモリに保存していく。プロセッサ状態モニタによって生成されるモニタリングデータは32ビットで表され、それ全体でグローバルクロックが表現されている。プロセッサ状態モニタは、約3日モニタすることができる。

3.2.2 通信リンクモニタ

トランスピュータは、4本の双方向通信リンクを持っており、通信リンクモニタはその入出力リンクを観察する。通信リンクの状態は、メモリ空間の通信ポートの内容を調べるとわかり、通信をしていないとき通信ポートには一定値(#80000000)が書かれているが、通信している状態であれば通信ポートには#80000000以外の値が書かれている。通信リンクモニタによって生成されるモニタリングデータは、32ビットで表現される。各リンクの様子はその通信リンクが使われていれば1、使われていなければ0としてグローバルクロック値の上位8ビットに表現し、各プロセッサのローカルメモリに保存する。通信リンクモニタは、約17分モニタすることができる。

3.3 ネットワーク記述ファイルの変更

VPMTでは、ユーザプログラムをモニタリングするためにネットワーク記述ファイルの変更を行なわな

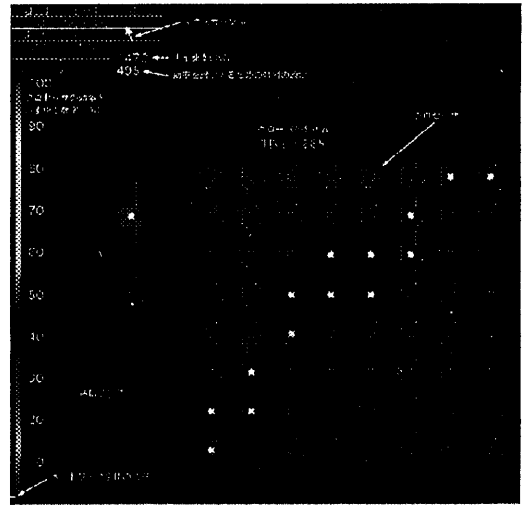


図2 可視化画面

ければならない。ネットワーク記述ファイルは、ユーザプログラムをプロセッサに割り当てるためのプログラムである。ネットワーク記述ファイルの変更は、自動的に行なえる。まず、ライブラリとして用意してあるモニタプログラムとグローバルクロック設定プログラムを付加する。次に、各プロセッサへのプログラムを割り当てるときに、はじめにグローバルクロックの設定をしたのちに、ユーザプログラムとモニタプログラムを並列に実行してユーザプログラムをモニタするように記述する。したがって、VPMTはユーザプログラムを直接変更することをしないので、ネットワーク記述ファイルの変更によってユーザプログラムに誤りが混入することはない。

4 可視化画面

モニタプログラムによって記録されたデータに基づいて、並列プログラムの動作の様子が動画として表示される。また、処理効率やプロセッサの動作状況などの数値的評価を行なうことができる。

4.1 動画による可視化

VPMTによる可視化画面を図2に示す。可視化画面の制御はマウスを用いて行なわれる。図2において、プロセッサは四角で、通信リンクは実線で示される。通信リンクは、使用されていれば水色、そうでなければ灰色で示される。プロセッサの内側の四角はアイドル／実行の状態を表し、アイドル状態であれば黒、そう

でなければ白で示される。内側の四角を囲む四角は処理効率を示すものであり、画面左のグラデーシンの縦棒で表される。また、システム全体の処理効率はインジケータが上下することで示される。

処理効率は、ある時間から過去の一定時間内に、実行状態にあった時間がどれだけあったかで示される。たとえば図3のように各プロセッサが動作したとすると、プロセッサ0では4000ユニットの間に1684ユニットだけ実行状態にあったので、処理効率は $(1684/4000) \times 100 = 42(\%)$ となる。システム全体での処理効率は、各プロセッサの処理効率の平均で示される。

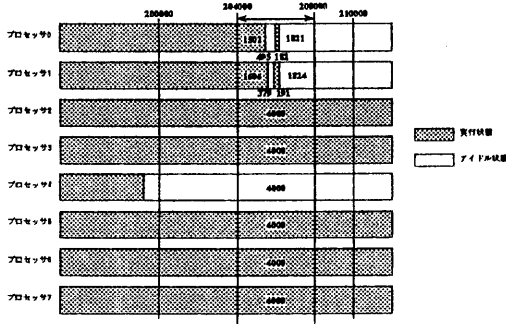


図3 処理効率の計算

表1 図3における処理効率

| プロセッサ | 処理効率 | プロセッサ | 処理効率 |
|--------|------|--------|------|
| プロセッサ0 | 42% | プロセッサ4 | 0% |
| プロセッサ1 | 45% | プロセッサ5 | 100% |
| プロセッサ2 | 100% | プロセッサ6 | 100% |
| プロセッサ3 | 100% | プロセッサ7 | 100% |
| 全プロセッサ | 73% | | |

動画による可視化によって、ユーザプログラムがどのように動作するかが視覚的にわかる。また、通信によるデッドロックやボトルネックなどが検出できるので、ユーザプログラムのデバッグに有効である。

4.2 数値的評価

並列プログラムの数値的評価は、モニタプログラムによって記録されたデータに基づいて行なわれ、処理効率、プロセッサの動作状況、並列度が示される。そして、これらは折れ線グラフ、棒グラフ、チャートの形で示される。

数値的な評価を行なうことによって、ユーザは自分が作成したプログラムの評価をおおよそでき、それに基づいて、より効率のよい並列プログラムを作成することができる。

4.3 VPMT の使用例

写真2-3に、マンデルブロ集合の計算を 16×16 の2次元メッシュ結合で行なったときの可視化画面を示す。マンデルブロ集合は、

$$z_{n+1} = z_n^2 + c \quad n = 0, 1, 2, \dots$$

を満たすすべての複素数 $c = a + ib$ で表される。この集合は各点において独立しているため、プロセッサファームと呼ばれる手法を用いることで高速に演算ができる。

写真2は、マンデルブロ集合の計算に必要なデータを各プロセッサにロードしている様子である。左上のデータを受け取ったプロセッサから計算をはじめていく様子がわかる。写真3は、計算結果をコントロールプロセッサへ返している様子である。早く結果を返しているプロセッサは、計算の付加が軽いことがみてとれる。また、早く計算が終了しているプロセッサが固まっているために、通信待ちが発生していることがわかる。

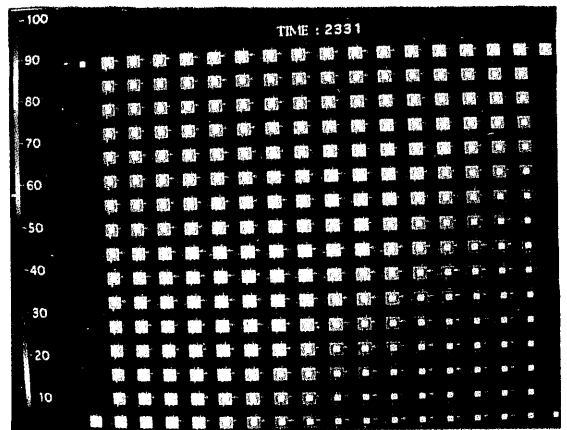


写真2 マンデルブロ集合計算の可視化画面1

5 演習用並列計算機への導入

我々の大学には、広く並列計算機にふれてもらうことを目的として4台のトランスペュータで構成された並列計算機がある(写真4)。そこで、演習を受講した学生に使用してもらうため、VPMTを演習用並列計算機に導入した。

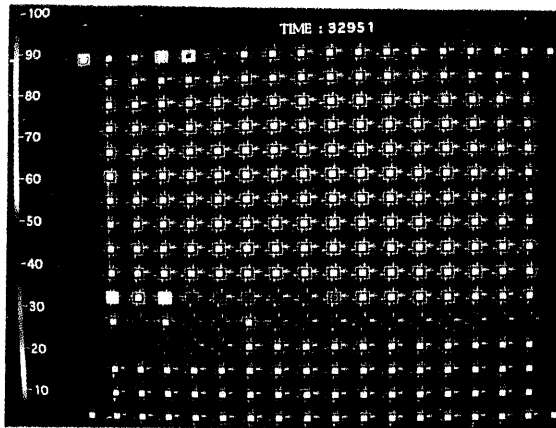


写真3 マンデルブロ集合計算の可視化画面2

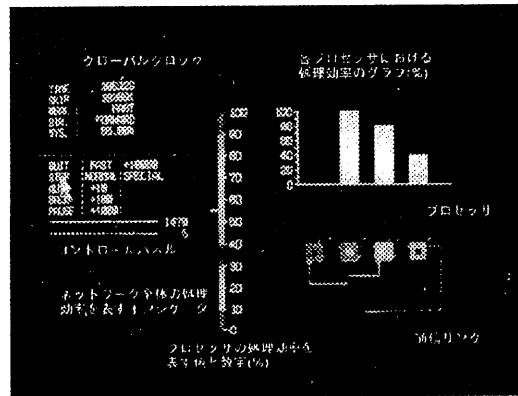


写真5 演習用並列計算機のための可視化画面

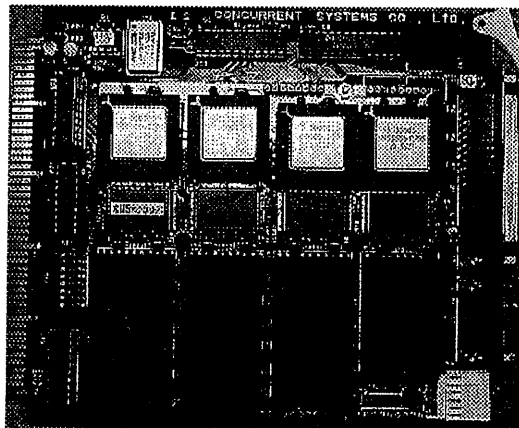


写真4 演習用並列計算機(このボードをPC-9801の拡張スロットに差し込むと使用できる)

5.1 演習用並列計算機のための可視化画面

演習用並列計算機のためのVPMTは、表示部が異なるだけで、あとは2節に述べたものと同じ処理を使用する。図5は、演習用並列計算機のための可視化画面である。プロセッサは四角で、通信リンクは実線で示される。また、処理効率は0-10%、11-20%、...、90-100%の10段階の色で表示され、その上部に対応付けて棒グラフによる処理効率の表示をしている。システム全体の処理効率は、インジケータの上下と数字で表示される。

5.2 演習用VPMTの使用例

写真6-7は、演習用並列計算機で、画像生成手法の1つであるレイトレーシングを実行したときの可視化例である。

レイトレーシングは、視点からスクリーンを通して視線(レイ)をとばし、物体に当たったその点の色と輝度を計算し、画像を生成する手法である。レイトレーシングの輝度計算は視点ごとに独立しているので並列化を行ないやすい。

写真6はすべてのプロセッサで計算をしているところである。物体の透過や屈折を考慮したプログラムを実行しているため、処理時間がかかり、可視化した画面での処理効率の表示は100%になっている。写真7は計算が終了データをコントロールプロセッサへと返しているところである。処理にかかる時間が視点によって違うので、可視化画面の処理効率はそれぞれ違う表示をしている。

6 結論

本稿では、並列プログラム可視化システムVPMTについて述べた。並列計算機のための可視化ツールは、並列プログラムを記述するうえで重要なものである。我々の開発しているVPMTは、トランスペユータ並列計算機上で動作する可視化システムである。VPMTを使うことによりプロセッサの動作を動的に理解することが可能となった。そのため、ユーザプログラムのデバッグを容易に行なうことが出来る。また数値的の評価をすることで、プログラムをチューニングするときにユーザに有益な情報を与えることが出来る。さらに、VPMTを演習用並列計算機に導入した。演習者は、自分の記述したプログラムをVPMTにより可視化することで、

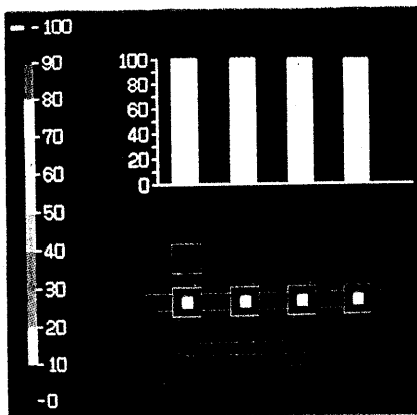


写真 6 レイトレーシングの可視化画面 1

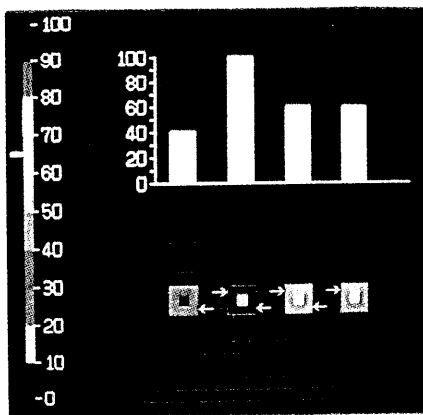


写真 7 レイトレーシングの可視化画面 2

並列処理の理解を深めることができると思われる。

参考文献

- [1] 釘村廣樹, 宮下浩, 梅尾博司. “VPMT:分散メモリ型並列計算機のための並列プログラム可視化システム”. 情報処理学会第 51 回全国大会講演論文集 (6), pp. 103-104, 1995.
- [2] Wolfgang Obelöer, Harald Willeke, and Erik Maehle. “Performance Measurement and Visualization of Multi-Transputer Systems with DELTA-T”. In *Performance Measurement and Visualization of Parallel Systems*, pp. 119-143. Elsevier Science Publishers B.V., 1993.

[3] U. Villano. “Monitoring parallel programs running in Transputer networks”. In *Performance Measurement and Visualization of Parallel Systems*, pp. 67-96. Elsevier Science Publishers B.V., 1993.

[4] Inmos Ltd. “*Transputer Instruction Set - a compiler writer's guide*”. Prentice Hall, 1988.