

並列処理ニューロシステムにおけるプロファイリング機能の実現

6 P-8

坪田浩乃* 王麗晶** 田中健一* 久間和生*

*三菱電機(株) 先端技術総合研究所

**University of British Columbia

1. はじめに

近年さまざまな分野で並列処理技術が適用されており、ニューラルネットワーク処理でも高速に演算実行できる並列ニューロシステムの開発がすすめられている(1)(2)。このようなシステムの開発では、高性能なH/Wの開発とともに、システムの能力を最大限に引き出すためのS/W開発環境の整備が極めて重要なものとなる。特に、同時に実行される各処理の状況を把握し、プロセッサ間の負荷の不均衡や処理のボトルネックとなっているデータ転送を発見することで、H/Wの性能を最大限に活かしたプログラム開発を可能とするためには、プロファイラが不可欠である。

本稿では、三菱電機で開発された並列ニューロシステム(NEURO4システム)用に開発したプロファイラについて報告する。まず、NEURO4システムおよび並列プログラムの表示ツールとして採用したParaGraphの概要について述べ、具体的な実現手法および実際のプログラムでの実行結果について論じる。

2. NEURO4システム

NEURO4は、12個のプロセッシングユニット(PU)を有するSIMD(Single Instruction stream Multiple Data stream)型マイクロプロセッサである(3)。このチップ

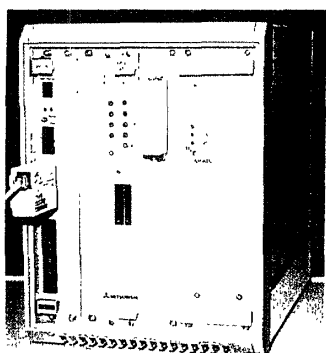


図1. NEURO4システム写真

Implementation of a profiler for a parallel neural-network system, Hirono TSUBOTA*, Li-Ching Wang**, Ken-ichi TANAKA*, Kazuo KYUMA*,
* Mitsubishi Electric Corporation Advanced Technology R&D Center
** University of British Columbia

を4個搭載したNEURO4_VMEボードが開発され、33MHz動作時に最高3.17GFLOPSが達成されている(4)(5)。このボードは、ホスト計算機の標準バス(VMEbus)を介して最大16枚まで接続可能である。チップ間の通信ポート接続を変更することで、チップ(12PU)単位、ボード(48PU)単位、あるいは全ボード(~768PU)でそれぞれ異なる動作をするSIMD処理ノードを構成することが可能である。

3. ParaGraph

ParaGraphは、実行時に生成されるトレース情報に基づいて、複数の処理ノード上の並列プログラムの動作やパフォーマンスを28種類のグラフやアニメーションで図的に表示するツールである(6)。解析の目的に合った表示方法を使うことで、NEURO4システムのように多数のプロセッサが同時に動作するシステムでの性能解析やボトルネックの特定の大きな助けとなる。

ParaGraphはメッセージパッシング方式のマルチプロセッサを仮定したツールであり、NX/2, VERTEXやPVM3.3といったいくつかのプラットフォーム用のトレース情報生成機能を有する通信ライブラリ関数PICL(Portable Instrumented Communication Library)が用意されている(7)。

4. NEURO4システムのプロファイラ実現

ParaGraphを用いて、有効なプロファイラを実現するためには、プログラムの実行状況を正確に記録したトレースファイルを生成する必要がある。

PICLを移植すればトレース情報は生成可能となる。しかし、ホスト計算機の制御のもとにSIMD型の動作をするスレーブボードが接続されたNEURO4システムに、メッセージパッシングマルチプロセッサ用の通信ライブラリを移植するのは、無意味である。そこで、NEURO4システムのS/W開発環境の一部として開発されていたホスト・ボード間の通信ライブラリであるボード制御ライブラリにトレース機能を追加することとした(図2)。

NEURO4システムでは、チップごとに異なるプログラムを実行させた場合には、1チップが、ボードごとの場合には、ボードが、実際の処理単位となる。この処理単位はダイナミックにも変更可能である。このた

め、実際の処理単位ではなく、ホスト計算機と各NEURO4チップをノードとする通信トレースデータを生成することとした。

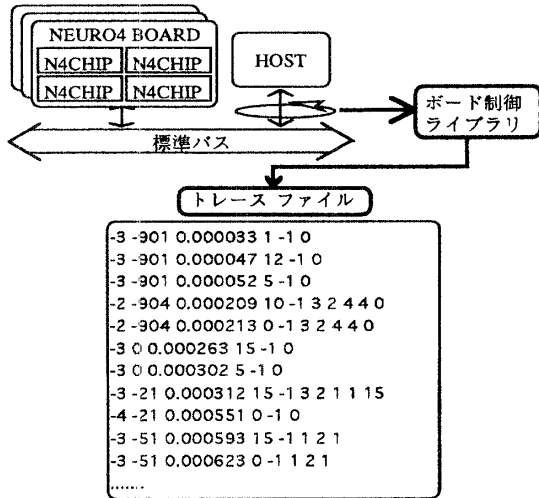


図2 トレースファイルの生成

5. インプリメンテーション

32個のボード制御ライブラリのうちで実際に通信を行っている29個の関数に対して、トレース機能を追加した。下記に、NEURO4のプログラム実行を開始させるN4PrgStart関数を例示する。関数の最初に実行開始時刻、最後に実行終了時刻をトレースバッファに格納するコード追加することで、トレース機能を実現している。

```

N4PrgStart()
{
#ifdef TRACEMODE
if (TraceFlag & MASK_PrgStart)
{
/* Obtain the action start time */
/* Save necessary information into the trace structure */
}
#endif

/* ... Original codes ... */

#ifdef TRACEMODE
if (TraceFlag & MASK_PrgStart)
{
/* Obtain the action stop time */
/* Save necessary information into the trace structure */
}
#endif
}

```

トレースバッファの準備等の初期化機能は、システムの初期化関数であるN4SessionStart中に、バッファデータのファイルへの格納は、システム終了関数N4SessionEnd中に実現した。これにより、アプリケーションプログラムのソースプログラムはそのまま、コンパイル時のライブラリの指定の変更だけでプロファイリングが可能となる。

また、ボード制御ライブラリの関数ごとにトレース

情報の生成の可否を指定するフラグを用意した(TraceFlag)。この指定によって必要度に合ったトレース情報を得ることが可能である。

6. 実行結果

図3に実際のParaGraphの表示例を示す。縦軸の番号は、各ノードに対応し、0がホスト計算機、1~4が4つのNEURO4である。横軸が経過時間を表し、各ノード間の通信の様子が、ボード制御関数ごとに異なる色で表示されている。この例では、1ボード、48PUでひとつの処理をさせている。このため、4つのノードは、ほぼ同じ挙動を示している。順に、ホスト計算機から各チップに、初期化(a)、プログラムロード(b)、データ転送(c)および演算開始、終了確認(d)の通信が起こっていることがわかる。この例では、全体の処理に対して、データ通信時間が長く、NEURO4ボードの性能に対して、粒度が小さい処理であったことがわかる。

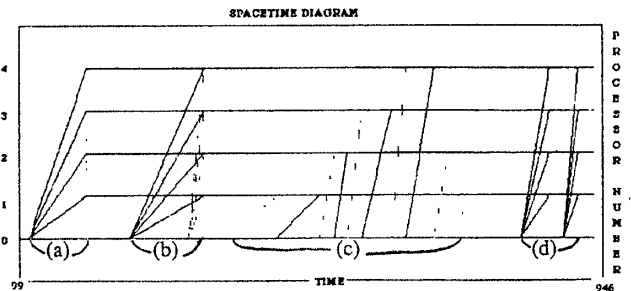


図3 ParaGraphの表示例

7. まとめ

並列ニューロシステムであるNEURO4システム用のプロファイラを実現した。並列プログラムのビジュアライザとしてParaGraphを適用することにし、そのために必要なトレースファイルの生成機能をNEURO4のボード制御ライブラリに追加した。これによって、各プロセッサでの負荷状況やデータ転送のオーバーヘッドを容易に観測することが可能になった。

参考文献

- 1) Adaptive Solutions Inc., <http://www.asi.com/hw/hw.html>.
- 2) HNC Inc., SNAP Technical Description, HNC Inc. 1993.
- 3) Y.Kondo, "A 1.2GFLOPS Neural Network Chip Exhibiting Fast Convergence", ISSCC Digest of Technical Papers, Vol.37, pp.218-219, 1994.
- 4) "汎用ニューロボード", 三菱電機技報・Vol 69.No.1, pp32, 1995
- 5) Y. Kondo et al, "Silicon VLSI Neural Network Chips for Real-Time Neural Applications", Proc. Int. Conf. Artificial Neural Networks (Paris), vol. A9, pp. 7-12, 1995.
- 6) Michael T.Heath, Jennifer Etheridge Finger, "ParaGraph: A Tool for Visualizing Performance of Parallel Programs", Technical Report, Oak Ridge National Laboratory, Oak Ridge, TN, November 1995.
- 7) G.A. Geist, M.T. Heath, B.W.Peyton, and P.H.Worley, "A Users' Guide to PICL, a Portable Instrumented Communication Library", Technical Report ORNL/TM-11616, Oak National Laboratory, Oak Ridge, TN October 1990.