

信号処理におけるバス結合型と多段結合網型の並列計算機の性能比較

水野 政治 宮田 裕行 磯西 徹明 菅 隆志

三菱電機(株) 情報電子研究所

概要

並列計算機の処理性能は、そのアーキテクチャと対象とするプログラムの形態に依存する。本論文では、特に実際に使用されている信号処理のアプリケーションを取り上げ、バス結合型と多段結合網型の二つのアーキテクチャにおける処理性能を比較した。その結果、最終的にはほぼ同等の性能が得られたが、バス結合型におけるメモリやバスの競合による性能低下、多段結合網型におけるプログラミング上の問題点等、各アーキテクチャにおける信号処理特有の特徴が明らかになった。

Performance Evaluation of Signal Processing on Two Shared Memory Multiprocessors

Masaji Mizuno Hiroyuki Miyata Tetsuaki Isonishi Takashi Kan

*Mitsubishi Electric Corporation
Computer & Information Systems Laboratory
Kamakura, Kanagawa, Japan*

Abstract

The performance of a multiprocessor system depends heavily on its architecture and the executing application programs. In this project, we compare the performance of signal processing program on two multiprocessors having different architecture - one with a common bus, and the other with a multistage interconnection network. Final results demonstrate performance similarities between the two systems. We will present these similarities, and also analyze each system's specific behavior in response to the application.

1 はじめに

並列処理の研究においては現在までに様々なアーキテクチャが考案され、種々の並列計算機の商用化が進められている¹⁾²⁾³⁾。しかしながら、各アーキテクチャには性能や使い易さの点で一長一短がある。また、その処理性能も並列処理の対象とするプログラムの形態に依存し、実行される並列プログラムにより大きく変化する。従って、実際の各種アプリケーションについて、どの様な並列処理の形態が存在するか検討し、各アーキテクチャにおける並列処理の有効性・問題点を明確にすることが重要となる。

本論文では特に信号処理のアプリケーションを取り上げ、バス結合型、多段結合網型の二つのアーキテクチャの性能を比較する。同様な研究としては、これまで理論的なモデル上での評価は多く行なわれている⁴⁾が、本評価のように実際の信号処理について検討したものは少ない。

以下、2章では使用したバス結合型と多段結合網型の並列計算機についてその概要を示す。3章では使用した信号処理のアプリケーションの特徴と並列化の手法について示す。4章では今回得られた評価結果及び考察を示す。

2 バス結合型と多段結合網型の並列計算機

バス結合型並列計算機とは、複数のプロセッサ、メモリ及びその他の共有資源をバスによって結合した並列計算機である。一方、多段結合網型並列計算機とは、プロセッサやメモリなどから構成されるPE(Processor Element)を多段スイッチにより複数個結合した並列計算機である。

本評価では、バス結合型並列計算機としてオムロン社製 LUNA-88K を、多段結合網型並列計算機として米国 BBN アドバンスド・コンピュータズ社製 TC2000 を使用した。図 1, 2は各並列計算機のシステム構成を示したものであり、表 1には

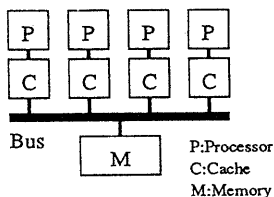


図 1: バス結合型並列計算機のシステム構成

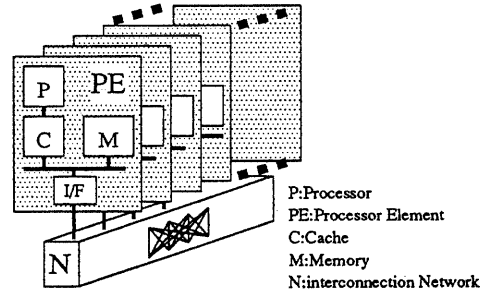


図 2: 多段結合網型並列計算機のシステム構成

各々のアーキテクチャについてまとめた。

表 1に示したように、2台の並列計算機のどちらもプロセッサとしてはモトローラ社製 RISC プロセッサ MC88100 を使用している。しかしながら、両者の間には動作周波数やプロセッサの結合方式、メモリ等の構成に相違があるため、本評価では各並列計算機の逐次処理に対する相対性能でアーキテクチャの評価を行なった。

以下、この問題について概説する。

2.1 プロセッサ結合方式

バス結合型並列計算機では、共有バスによって複数のプロセッサ及び共有メモリ、その他周辺装置等がこの共有バスによって接続されている(図 1)。そのため、各プロセッサからメモリや周辺装置へアクセスする場合、必ずこの共有バス経由で行われる。この構成は各プロセッサからみた場合、従来のシングルプロセッサでの構成と大差なく扱うことができ、従来の計算機技術の導入が比較的容易に行える利点がある。しかし、メモリや周辺装置へのアクセスが各プロセッサで多発する場合、バスの負荷が増加しシステムの処理能力の低下を招く。また、プロセッサ数が増大した場合にも同様な問題が生じ、接続できるプロセッサ数に限界がある。

多段結合網型並列計算機では、多段スイッチによって複数の PE が接続されている⁵⁾(図 2)。多段スイッチは比較的規模の小さいクロスバー・スイッチを複数個組み合わせることで多数のノードを接続するものであり、同数のノードを単独のクロスバー・スイッチで構成する場合に比較しハードウェア量が小さくすむ利点がある。また、同一のノードにアクセスが競合しない限り、複数のプロセッサは同時にアクセスを行うことができ、シス

表 1: 各並列処理計算機のアーキテクチャの比較

	バス結合型並列計算機 (LUNA-88K)	多段結合網型並列計算機 (TC2000)
プロセッサ (動作周波数)	MC88100(25MHz)	MC88100(20MHz)
プロセッサ数 (最大構成時)	4 (4)	4 (512)
キャッシュ (容量)	MC88200(16KB) x 2 /CPU	MC88200(16KB) x 2 /CPU
メモリ・アーキテクチャ	共有メモリ (32MB)	分散メモリ (16MB/CPU)
メモリ・アクセス・モデル	UMA	NUMA
プロセッサ結合方式	共有バス	多段スイッチ

テム全体の通信バンド幅は大幅に増大する。そのため、システムにはプロセッサを数百台程度まで接続できる。しかし、一つの PE 内で行われるメモリのアクセス時間に比較し、多段スイッチを介し複数の PE 間で行われるメモリのアクセス時間は遅くなるため、頻繁に PE 間のメモリアクセスを行う場合は処理能力が低下してしまう。

2.2 メモリ・アーキテクチャ

バス結合型並列計算機にはシステム全体に共有メモリが一つ、多段結合網型並列計算機には各プロセッサにメモリが分散して存在している。バス結合型並列計算機は全てのメモリを同一のコストでアクセスであることから UMA (Uniform Memory Access) に、多段結合網型並列計算機はメモリへのアクセスコストが不均一であることから NUMA (Non-Uniform Memory Access) に分類される⁶⁾。UMA では特に問題にならないが、NUMA ではデータのメモリ配置がその性能に大きく影響を与える。従って、多段結合網型並列計算機では、データの依存性を解析するなど並列処理独特の作業が必要となるのに加え、データをどのメモリに配置するかについても考慮する必要がある。

2.3 キャッシュ・アーキテクチャ

本評価で使用した 2 台の並列計算機はいずれも各プロセッサに命令用 16KB、データ用 16KB のキャッシュメモリが用意されており、物理的には違いがない。

バス結合型並列計算機では、キャッシュは共有メモリへのアクセス回数を減少させ共有バスの負荷を軽減する効果をも有する。さらにスヌーピング機能によりプログラムからは特にキャッシュを制御することなく、複数のキャッシュ間のデータ

の一貫性を保つことができる。一方、多段結合網型並列計算機ではデータの一貫性を保つためにはディレクトリ方式などのより複雑な制御が必要である。本評価で使用した多段結合網型並列計算機にはデータの一貫性を保つためのハードウェア機能が特に用意されていないため、キャッシュを利用するためにはプログラムの中で明示的にデータの一貫性を保つ必要がある。

3 信号処理の概要と並列化手法

本評価では信号処理のアプリケーションの並列プログラムを作成し、前述の 2 台の並列計算機上で実行し、その実行時間を元に各々の並列計算機の性能を比較した。以下、評価に使用した信号処理の概要及び並列化手法について示す。

3.1 信号処理概要

本評価では信号処理のアプリケーションとして多種のセンサで用いられる対表的な実用レベルの信号処理を使用した。一般に信号処理は、高速な処理が要求される上、処理の途中で二次元データの行と列を入れ換え (コーナーターン) が生じたり、データ量が増えるなどの処理が存在する。そのため、並列化の手法、データのメモリ配置、コーナーターンの扱い、変化するデータ量の扱い等、並列処理を行なう上での信号処理特有の問題点が多い。

さらに信号処理はその処理形態により次の二つに分類することができる。

1. 処理の途中でコーナーターンが生じるもの
2. 処理の途中でコーナーターンが生じないもの

本評価ではこの二つの処理形態の代表的な実用例である「信号処理 A」「信号処理 B」について検討する。

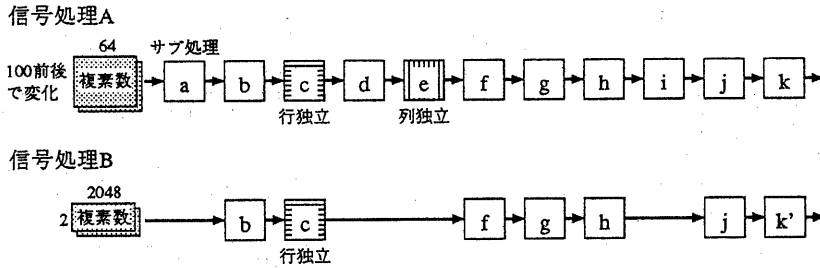


図 3: 信号処理の処理フロー

これらの処理は二次元配列を入力して、そのデータに対し順に処理を施し、何らかの結果を求めるものである。この流れを図3に示す。

信号処理 A は a ~ k の計 11 個のサブ処理を順に行うものである。ここでサブ処理の具体的な内容は、例えばサブ処理 b は重み付け、サブ処理 c は高速フーリエ変換 (FFT)、サブ処理 f は実数化などである。入力される二次元配列の大きさは $(64) \times (100 \text{ 前後で変化})$ であり、約 50KByte の大きさとなる。11 個のサブ処理のうち、サブ処理 c は行についてのみ独立に処理を行なえ、サブ処理 e は列についてのみ独立に処理を行なえる。サブ処理 c, e 以外のサブ処理は行にも列にも独立に処理を行なえるものである。このため、信号処理 A では処理の高速化のためにはコーナーターンが必要となる。

信号処理 B は b ~ k' の計 7 個のサブ処理を順に行うものである。ここでサブ処理 b ~ j は信号処理 A とまったく同様の処理である。入力される二次元配列の大きさは $(2048) \times (2)$ であり、32KByte の大きさとなる。7 個のサブ処理のうち、サブ処理 c のみが行についてのみ独立に処理を行なえ、他のサブ処理は行にも列にも独立に処理を行なえる。

3.2 並列化手法

まず、各プロセッサに分割する処理の大きさについて検討する。各プロセッサにサブ処理毎に分割すると、3.1に示したように各サブ処理は順に処理する必要があるため、並列化の効果はまったく表れない。ところが各サブ処理の内容は、基本的に、入力される二次元配列の全要素について同一の処理を施すため、行方向と列方向の各々の繰り返し処理によって処理される。そこでこの繰り返し処理をうまく分割し各プロセッサに割り当て

る。

次に、繰り返し処理の分割方法について検討する。個々の繰り返し処理は条件分岐等はほとんど含まれないため、その処理時間はほぼ一定である。そこで繰り返し回数をプロセッサ数で均等に分割し各プロセッサに割り当てる。

続いて、行方向と列方向のどちらの繰り返し処理を分割するかを検討する。

信号処理 A においては図3に示すように、サブ処理 c は行独立、サブ処理 e は列独立である。そこで、サブ処理 e のみ列方向で分割し、その他のサブ処理はすべて行方向で分割することとした。この結果、サブ処理 d とサブ処理 e で行から列に、サブ処理 e とサブ処理 f で列から行に変更するため、コーナーターンが生じる。また、サブ処理 f 以降では各プロセッサの処理自体は独立になるが、それ以前のサブ処理において他のプロセッサで得られた処理結果を参照するため、幾つかのデータをプロセッサ間で交換する必要がある。

一方、信号処理 B ではサブ処理 c のみが行独立であり、他のサブ処理は行にも列にも独立であるため、全てのサブ処理において行方向に分割することとした。信号処理 A と同様に、サブ処理 f 以降では各プロセッサの処理自体は独立になるが、それ以前のサブ処理において他のプロセッサで得られた処理結果を参照する必要があり、プロセッサ間のデータ交換を必要とする。

以下、各並列計算機における並列処理の実現方法について示す。

3.2.1 バス結合型並列計算機での並列処理

本評価で使用したバス結合型並列計算機では、OS として米カーネギメロン大学で開発された Mach が採用されている。この OS では、並列処理を行う場合は一つのプログラム (Mach というタスク) 上で複数のスレッドを生成してやればよい。

スレッドを取り扱うために用意された C 言語用のライブラリが C thread ライブラリであり、従って並列処理もこのライブラリを使用することにより可能となる⁷⁾。

本評価ではスレッドの生成や起動、削除にかかるオーバーヘッドを軽減するため、以下の方法で並列処理を実現した(図4)。この方法は後述する

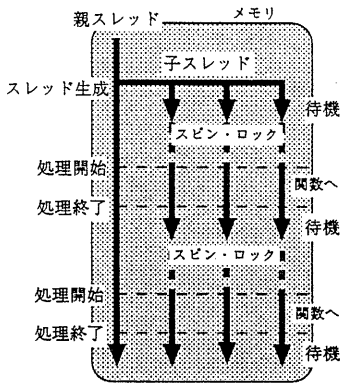


図 4: バス結合型並列計算機における並列処理

多段結合網型並列計算機での方法とほぼ等しい。

1. プログラムの初めの部分で必要な数だけスレッドを生成する。
2. 生成したスレッドはスピン・ロックして、実際の処理の開始を待つ。
3. 並列処理を行いたい関数を指定し、並列処理を開始させる。
4. これまでスピン・ロックして待機していたスレッドはスピン・ロックを中止し、指定された関数へ処理を移す。
5. 関数の処理を終了した後、再びスピン・ロックに移り、次の処理開始を待つ。

ここで、各スレッドは実行されるプロセッサは特定されず、実行状態により異なるプロセッサ上で実行される。また、これらのスレッドは同一のメモリ上で実行されるため、コーナーターンの生じる場合でもプロセッサ間で意識的にデータ交換を行わない。

3.2.2 多段結合網型並列計算機での並列処理

本評価で使用した多段結合網型並列計算機では、nX と呼ばれる独自の OS が動作している。

この OS では、Uniform System ライブラリという高級言語から利用できる並列化ライブラリが提供されており、本評価においてもこのライブラリを使用し、並列処理を実現した⁸⁾。

Uniform System ライブラリを使用した場合、次のような方法で並列処理が実現される(図5)。

1. Uniform System ライブラリの初期化。
→ 必要な数だけプロセスが生成され、実際の処理の開始を待つ。
2. 並列処理を行いたい関数を指定し、並列処理を開始させる。
→ 各プロセスは指定された関数へ
3. 関数の処理を終了した後、再び待ち状態に移り、次の処理開始を待つ。

ここで、各プロセスは実行されるプロセッサは特定されており、常に同一のプロセッサ上で実行される。また、各プロセスは個々のメモリ上で実行されるため、コーナーターンの生じる場合には性能向上を図るためにプロセッサ間でデータ交換を行なう。

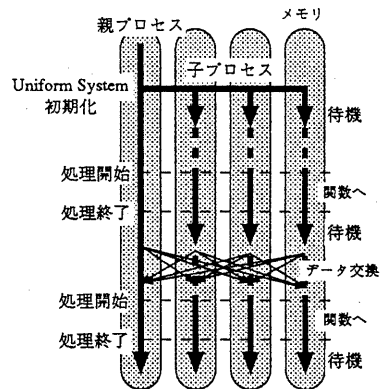


図 5: 多段結合網型並列計算機における並列処理

4 評価

4.1 信号処理における並列化の効果

プロセッサ数を1台から4台まで変更した場合について、3.1に示した2種類の信号処理のアプリケーションの実行時間を測定し、その実行時間の変化により各並列計算機の並列化の効果について検討する。ここで示す実行時間はデータの入力や

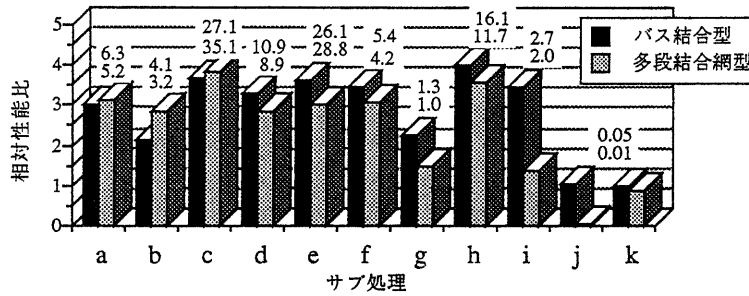


図 6: 信号処理 A の各サブ処理における並列化の効果

プログラムの初期化 (スレッドの生成や Uniform System の初期化を含む) 等を除いた、信号処理自身の実行にかかった時間である。

4.1.1 信号処理 A における並列化の効果

図 6 に信号処理 A の各サブ処理における並列化の効果、図 7 に信号処理 A におけるプロセッサ数の変化に伴う相対性能比の変化の様子を示す。図 6 において、横軸は各サブ処理名、縦軸は逐次プログラムの性能を“1”とした場合のプロセッサ数 4 台の時の性能比である。また、各棒グラフの上の数字は、逐次処理プログラムにおける各サブ処理の実行時間の割合 (単位は %) であり、上にバス結合型並列計算機の値を、下に多段結合網型並列計算機の値を示した。図 7 において、横軸はプロセッサ数、縦軸は逐次プログラムの性能を“1”とした場合の各プロセッサ数における性能比である。

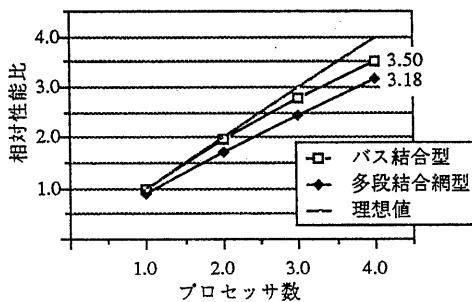


図 7: 信号処理 A における相対性能比の変化

いずれの並列計算機においても並列処理を行うことにより、プロセッサ 4 台で 3 倍以上の性能向上があった。

2 台の並列計算機の並列化の効果と比較した場合、約 0.3 倍ほどバス結合型の方が優れている。図 6 をみるとわかる通り、サブ処理 a ~ c までは多段結合網型並列計算機の方が、それ以降はバス結合型並列計算機の方が優れている。信号処理の内容から考察すると、サブ処理 a ~ c まで各プロセッサはまったく独立にデータをアクセスしているのに対し、サブ処理 d 以降は何らかの形で他のプロセッサのデータを参照する。従って、本評価で使用した信号処理については多段結合網型並列計算機における PE 間のメモリアクセスによる性能低下に比較し、バス結合型並列計算機におけるバス競合による性能低下の方がその影響が少ないことが考えられる。

4.1.2 信号処理 B における並列化の効果

図 8 には図 6 と同様に信号処理 B の各サブ処理における並列化の効果、図 9 には図 7 と同様に信号処理 B におけるプロセッサ数の変化に伴う相対性能比の変化の様子を示した。

バス結合型並列計算機においてはプロセッサ 4 台で 2.8 倍の性能向上が、多段結合網型並列計算機においてはプロセッサ 4 台で 3.4 倍の性能向上があった。2 台の並列計算機の並列化の効果と比較した時、多段結合網型の性能向上が 0.6 倍ほど優れている。これは、図 8 をみるとわかる通り、逐次処理全体の約 70% を占めるサブ処理 c での違いが全体の違いとなって表れたものである。サブ処理 c では具体的には FFT を計算しているが、そこで各プロセッサが使用するデータの大きさは 32KB であり、データキャッシュの容量 (16KB) を大きく越える。このため、バス結合型並列計算機ではメモリアクセスの競合が生じ性能低下が生じたものと考えられる。一方、多段結合網型並列計算機ではデータはローカルなメモリに格納でき

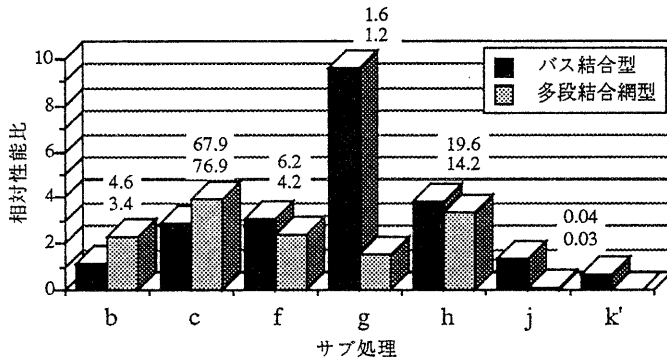


図 8: 信号処理 B の各サブ処理における並列化の効果

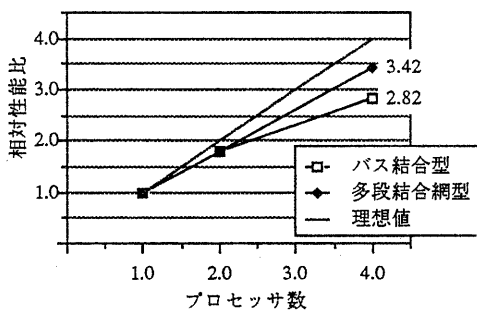


図 9: 信号処理 B における相対性能比の変化

るため、特に性能低下は生じなかったものと思われる。

また、サブ処理 g ではバス結合型並列計算機において、約 9 倍の性能向上がみられた。このサブ処理もデータの局所性が非常に高い処理であり、各プロセッサが使用するデータの大きさは 16KB である。この様な性能向上が生じた原因については次章で検証することとする。

4.2 データサイズの処理性能への影響

図 8 に示すように、バス結合型並列計算機においてサブ処理 g の性能向上は 4 倍を大きく越えたものとなっている。ここでは、この原因を追求した。

サブ処理 g は、データの局所性が非常に高い処理であり、処理の大部分はデータの read/write を行なうものである。そこで、ここではデータサイズを意図的に変更し、そのときの実行時間の変化を測定した。

図 10 は評価の結果を示したものである。同図において、横軸は各プロセッサで扱うデータの大き

き、縦軸は逐次プログラムの性能を“1”とした場合のプロセッサ 4 台の時の性能比である。

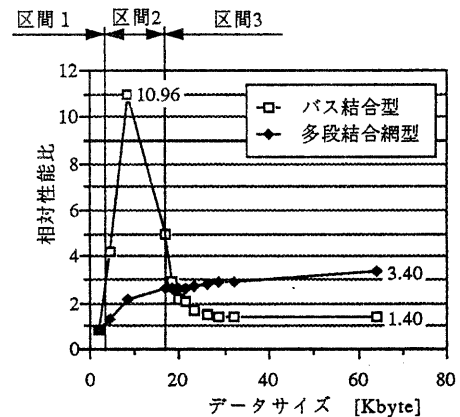


図 10: データサイズによる処理性能の変化

同図において、区間 1 は 1 プロセッサで実行する場合も 4 プロセッサで実行する場合も扱うデータの大きさがキャッシュの容量を越えない場合である。区間 2 は 1 プロセッサで実行する場合はキャッシュの容量を越えるが、4 プロセッサで実行する場合はキャッシュの容量を越えない場合である。区間 3 は 1 プロセッサで実行する場合も 4 プロセッサで実行する場合もキャッシュの容量を越えてしまう場合である。

バス結合型並列計算機では、区間 2 において大幅な性能向上がみられる。これは逐次処理においてデータをメモリへアクセスしていたものが、並列処理の場合にはキャッシュへのアクセスで行なえるため、データのアクセス時間が短縮されたためと考えられる。しかし区間 3 では急激にその効果は衰え、4 プロセッサで並列処理を行なった場

合でも 1.4 倍しかならない。これは区間 3 ではデータが大きくなるにつれバスの負荷が増大するためと考えられる。区間 1 については、実際の処理内容に対し並列処理のオーバーヘッドの割合が大きくなるため、処理性能が低下していると考えられる。

多段結合網型並列計算機においては区間の区別なく、データサイズが大きくなるにつれ、性能向上が大きくなっている。このサブ処理の演算結果は後に他のプロセッサのサブ処理で使用されるため、データの一貫性を保つためにライト・スルーの Protokol を用いている。つまり、逐次処理でも並列処理でも常にメモリにアクセスすることとなり、特にキャッシュの効果は表れない。従って、データが大きくなることにより並列処理の一単位の大きさが増加し、処理に占める並列処理のオーバーヘッドの割合が小さくなるためとみられる。

4.3 プログラミング性

4.1 の評価の際、並列プログラムを作成した経験に基づき、各並列計算機のプログラミング性について以下に示す。

信号処理 A について、逐次プログラムから特にアーキテクチャを考慮せず並列プログラムを作成した際、バス結合型並列計算機では 3.4 倍、多段結合網型並列計算機では 0.9 倍の性能が得られた。アーキテクチャを考慮し、以下の点について並列プログラムのチューニングを行なった結果、多段結合網型並列計算機では 3.2 倍の性能が得られた。

- 各プロセッサで使用するデータをプロセッサと同一の PE 内のメモリに配置した。
- コーナーターンが生じる時 プロセッサ間で交換するデータ量をできるだけ少なくした。
- プロセッサ間でデータを共有する場合、デフォルトではそのデータはキャッシュされないが、それをキャッシュを利用するように変更し、データの一貫性を保つようキャッシュ制御用の命令 (キャッシュの無効化やフラッシュを行なう命令等) を挿入した。

この様に、バス結合型並列計算機に比較し、多段結合網型並列計算機ではハードウェア構成を理解した上で並列プログラムを作成することが重要であり、高い性能を得る並列プログラムを作成する

にはハードウェア構成を考慮した多くの工夫が必要となる。

5 おわりに

様々なアーキテクチャが提案されている中、本論文では信号処理という特定の応用について、バス結合型と多段結合網型の 2 つのアーキテクチャの性能を比較した。

信号処理は他の数値計算にない処理が存在し、並列処理を行なう上での問題点が多い。これらの問題点は、特に多段結合網型並列計算機の性能に大きな影響を与えており、逐次プログラムから特にアーキテクチャを考慮せず並列プログラムを作成した時にはプロセッサ 4 台で実行する並列処理の性能が逐次処理の性能より劣るものとなってしまった。アーキテクチャを考慮しチューニングを行なった結果、バス結合型並列計算機と同程度の性能を得ることができたが、データのメモリ配置やキャッシュの制御など複雑なプログラミングが要求される。

なお、多段結合網型並列計算機はスケラビリティに優れており、数百台程度の並列処理を行なえる利点があり、本来ならもっと多くのプロセッサ数まで評価するべきであるが、本評価においては 4 台までしか使用していない。

謝辞 本研究を進めるにあたり、御指導、御討論いただいた次世代部岩瀬部長ならびに並列 1 グループ諸氏に感謝致します。

参考文献

- 1) 富田真治, 末吉敏則: 並列処理マシン, オーム社 (1989).
- 2) 高橋 義造編: 並列処理機構, 丸善 (1989).
- 3) 奥川峻史: 並列計算機アーキテクチャ, コロナ社 (1991).
- 4) A. Sivasubramaniam, et. al., : Experimental Evaluation of Algorithmic Performance on Two Shared Memory Multiprocessors, in *Proc. of the International Symposium on Shared Memory Multiprocessing*, pp. 13-24 (Apr. 1991).
- 5) BBN Advanced Computers Inc.: *Inside the TC2000 Computer*.
- 6) M. Young, et. al., : The Duality of Memory and Communication in the Implementation of a Multiprocessor Operating System, in *Proc. 11th Symp. on Operating Systems Principles* (Nov. 1987).
- 7) オムロン (株): LUNA88K UniOS-Mach 解説書.
- 8) BBN Advanced Computers Inc.: *Uniform System Programming in C*.