

リバモアループによるクラスタ型
マルチプロセッサの性能評価

2N-1

Performance Evaluation of the Clustered Multiprocessor by Livermore Loops

堀口 進 川添良幸 重井芳治
Susumu HORIGUCHI Yoshiyuki KAWAZOE Yoshiharu SHIGEI
東北大学情報工学科 東北大学 東洋大学情報工学科
情報処理教育センター

1. はじめに

高速計算処理の1つの方法として、演算処理を分割し同時に並行して処理を行う並列計算機が近年注目され、種々のアーキテクチャが提案・試作されている。我々は種々の理論的検討に基づいて、32台のマイクロプロセッサを8台ずつ4クラスタにまとめたクラスタ型試作システムMUGENを設計構築した[1]。本報告では、このクラスタ方式バス結合マルチプロセッサで動作する並列処理記述言語を用いて、リバモアループによるMUGENの性能評価を行なった結果について述べている。

2. クラスタ型マルチプロセッサシステム

4クラスタからなる32台のスレーブプロセッサ(SP)を持つクラスタ方式バス結合マルチプロセッサの試作システム機MUGEN(MULTiprocessing system with pErfect connection Network between clusters)システムの構成を図1に示す。システム全体を管理するマスタプロセッサ(MP)の下に4つのクラスタがある。各クラスタは、クラスタ内のバスを管理するバスコントローラ(BC)、データの受け渡しに使用するクラスタ共通メモリ(CM)、および8つのSPで構成されている。それぞれのBC間はパラレルポートで完全結合され、異なったクラスタ間でのデータ通信を容易にしている。また隣接するSP間はシリアルポートで接続され、これによって共有バスを使用しないでデータの交換を行なうことができる。各SPとBC間および各BCとMP間はパラレルポートを通して結ばれる。システムのハードウェア仕様を表1に示す。試作システ

MP (MASTER PROCESSOR)	Number	1 Unit
	CPU	Z80A-CPU
	ROM	8 KB
	RAM	56 KB
	PIO	Z80A-PIO×6 (12 Ports)
	SIO	Z80A-SIO (2 Ports)
	CTC	Z80A-CTC
BC (BUS CONTROLLER)	Number	4 Units
	CPU	Z80A-CPU
	ROM	8 KB
	RAM	8 KB
	PIO	Z80A-PIO×8 (16 Ports)
CM (CLUSTER COMMON MEMORY)	Number	4 Units
	RAM	32 KB
SP (SLAVE PROCESSOR)	Number	32 Units
	CPU	Z80A-CPU
	RAM	32 KB
	PIO	Z80A-PIO (2 Ports)
	SIO	Z80A-SIO (2 Ports)
	CTC	Z80A-CTC

表1 ハードウェア仕様

ムはSP32+BC4+MP1の計37個のマイクロプロセッサからなり、メモリは、ROMがMP8+BC8×4の40KB、RAMがMP56+BC8×4+SP32×32+CM32×4の1242KBで計1282KBから構成されている。

3. システムソフトウェア

MUGENのソフトウェア環境は、大きく分けてモニタと並列処理言語から構成される。基本システムソフトウェアの構成概念を図2に示す。計算機では、システムの立ち上げ時から動作を制御するシステムモニタが必要となる。MUGENでは、モニタはシステムの各プロセッサのメモリ上に常駐し、システムの動作の制御を行ない、またプログラムの実行やデバッグなどの際にはシステムの運用・操作が効率的に行えるよう便宜を図っている。また実際のプログラムを作成する時のために、各種の基本的な操作用ルーチンをシステムコールとして用意している。

実際の並列処理の応用プログラム開発を行う上で、アセンブラで種々の数値計算アルゴリズムを全て記述することは非常に時間がかかり、新しい並列処理アルゴリズムなどの開発を行うことは不可能に近い。そのため、プログラム開発が効率よく行え、システムの機能を十分に発揮できる高位言語は必要不可欠である。そこで、本システム上で実行可能なオブジェクトを生成できる高位言語のコンパイラを開発した。具体的には、SIMD型処理が可能な並列C言語『ext-C』と、MIMD型処理が可能である並列C言語『para-C』を設計・開発した。前者は複数のプロセッサ上で同一のプログラムによってデータを処理するだけの機能であるが、後者は複数のプロセッサ上で異なるプログ

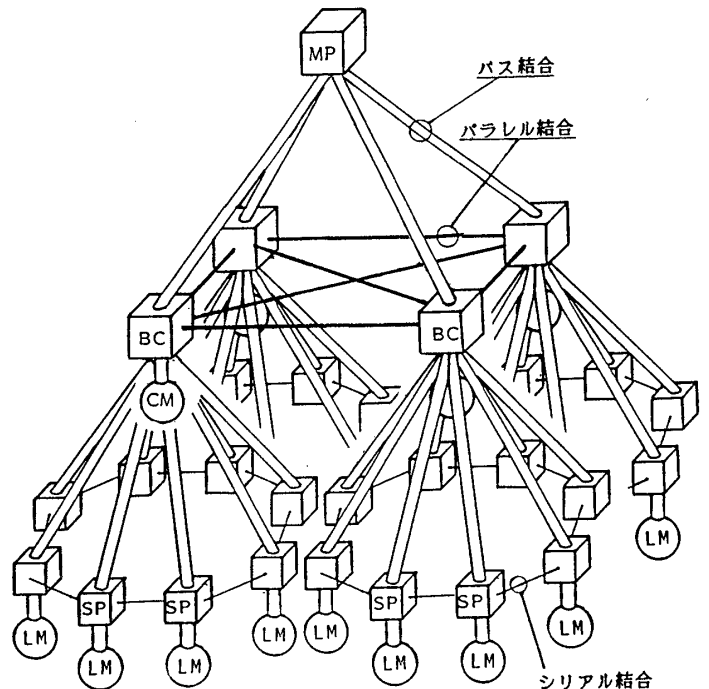


図1 システム構成概念図

Performance Evaluation of the Multiprocessor by Livermore Loop
S. Horiguchi: Dept. of Information Science, Tohoku University
Y. Kawazoe : ECIP, Tohoku University, Sendai 980, JAPAN
Y. Shigei : Deapnt. of Information Science, Toyo University

ラムの処理も可能とする。para-C では、並列処理をプログラム上に陽に記述する方法を取る。この方法によって、並列処理の単位をプログラマが任意に決定できるので、実効的にスループットを上げることができる。反面、並列処理を行なうためには、プログラマが並列処理を直接記述する必要があり、通常のC言語で記述されている既存のプログラムを para-C によって並列に実行することはできない。

この問題を解決するために、通常のC言語で記述されたプログラムを para-C 用のプログラムに変換するプリプロセッサ P3C (Pre-Processor for Para-C) を設計し、インプリメントした。P3C は、C言語ソースプログラム中の for ループを解析し、para-C の並列処理ルーチンへの展開を行なう。これを用いることによって、通常のC言語で記述されたプログラムを、自動的に para-C 用プログラムに変換し、高速に並列実行することができる。

4. リバモアループによる性能評価

C言語で記述したプログラムを実行する際のシステムのスループットを測定するために、リバモアループによるベンチマークテストを行なった[2]。リバモアループの基本プログラムは、流体、内積などの一般的な大規模計算の中核コードを集めた14個のD0ループで構成されている。

このリバモアループをC言語で記述し、SIMD型の処理が可能であるext-C言語のコンパイラを用いてコンパイルした場合のMUGENシステム上での処理時間を測定した。測定結果を図3に示す。縦軸の値は、MP1台で計算した場合の実行時間に対する速度比を表している。また、処理時間にはデータ転送時間も含まれている。並列化による速度向上比は、2から17倍と大きなばらつきが見られる。これは、演算のアルゴリズムによって並列化できる度合いが違ふことと、SPに転送するデータ量が大きく違ふ事による。また同一のループではSPの数を、8, 16, 32台と増加させるにつれて、速度比は増加していくが、増分は少なくなっていく。問題の大きさに依存するので一般論を述べるのは難しいが、ここで扱った規模のループに対しては、数十台

ソフトウェア構成

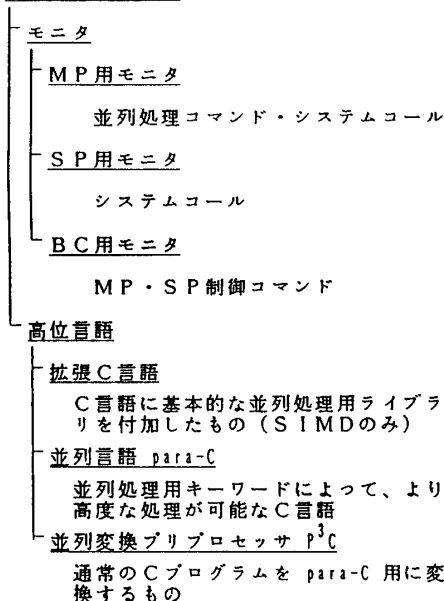


図2 ソフトウェア構成図

のプロセッサからなるMUGEN程度の並列処理システムは処理速度向上に有効であると言える。

また、各ループをext-C言語を用いてコンパイルした場合の処理速度比 (SPの台数は32台) とスーパーコンピュータ SX-1 のベクトルプロセッサを用いた場合の速度向上比を図4に比較した。ループのアルゴリズムの違いによって、速度向上の度合いがベクトルプロセッサと並列計算機とで異なるが、処理速度が全く向上しないループは共通である。ループ13および14に対してSX-1より本システムが相対的に良い結果を出しているが、両者とも非常に速度向上比が悪い。SPの台数を増やした場合、ループ2および7で本システムは秀れた処理速度向上比を実現できる。このように、並列計算機とベクトル計算機では、それぞれ処理に適する問題が異なっている。一般に、異なる型の演算処理が混在している場合には、並列計算機が優位であり、この意味で汎用性が高いと言える。

5. 結論

ベンチマークテストとして、スーパーコンピュータの性能評価用として良く用いられているリバモアループをクラスタ方式マルチプロセッサMUGEN上に設けるシステムの性能を評価した。その結果、ループごとに相当のばらつきはあるものの、一般的に相当程度処理速度の向上が得られた。また、para-C を用いてMIMD型の処理を行なうことにより、より高速に実行できることが明らかになった。

参考文献

- [1] 中田 他: "クラスタ方式マルチプロセッサシステム" 電子情報通信学会論文誌 vol1J70-D, pp. 1469-1477 (Aug. 1987).
- [2] 堀川 他: "リバモアループによるクラスタ型マルチプロセッサシステムの性能評価" 電子情報通信学会 コンピュータ研究会 (Jul 1988)

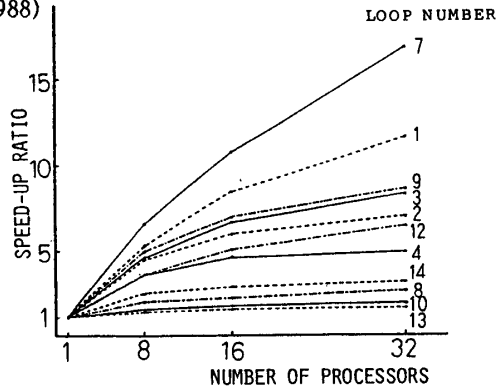


図3 リバモアループに対する処理速度比

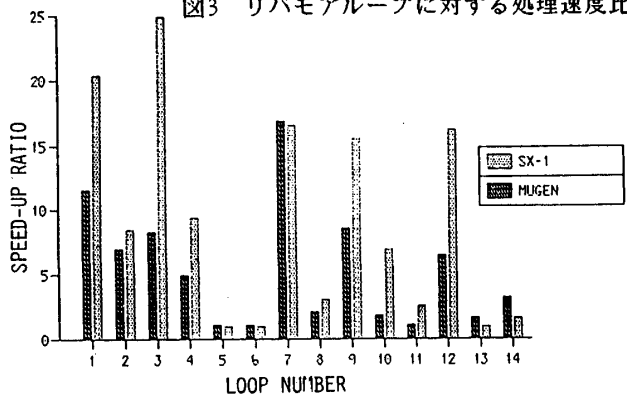


図4 ベクトルプロセッサとの比較