

データ駆動計算機 EDDEN とその性能評価

岡 本 一 晃[†] 川 口 正 樹[†]
三 浦 宏 喜[†] 清 水 雅 久[†]

我々は、実用的な並列処理計算機の実現に向け、要素プロセッサ数 1000 規模の高並列データ駆動計算機 EDDEN (Enhanced Data Driven ENgine) を開発中である。EDDEN は、循環パイプライン方式の演算処理機能とベクトル演算機能、さらにルーティングのための通信制御機能とを有する要素プロセッサ LSI が多数結合して構成される高並列データ駆動計算機であり、その開発の第一段階として、現在 64 台の要素プロセッサからなる中規模並列処理システム pEDDEN (*personal*-EDDEN) が稼働中である。本論文では、この中規模システム上でいくつかの試験プログラムを実行させることにより EDDEN の性能評価を行ったので、その結果を報告する。今回の評価は、特に基本性能の中でも、1) 数値演算性能、2) 待ち合わせ性能、3) 通信性能、の 3 点に注目し、ベンチマークプログラムを用いて行った。その結果、リバモアループにおいて 300 MFLOPS という実効性能を得た。また、別のベンチマークにより 63.0 MB/sec, 56.1 Msync/sec という高い実効値を確認した。そして、EDDEN の特長であるベクトル演算機構の導入が、これらの高い実効値に大きく貢献していることが示された。さらに、いくつかのアプリケーションを実行させることにより、より実用的なプログラムにおいても高い性能を得られることが確認できた。

Performance Evaluation on the Highly Parallel Dataflow Machine EDDEN

KAZUAKI OKAMOTO,[†] MASAKI KAWAGUCHI,[†] HIROKI MIURA[†]
and MASAHISA SHIMIZU[†]

The EDDEN (Enhanced Data Driven ENgine) is a highly parallel dataflow computer. A floating-point operation unit, a vector operation control unit and a communication control unit are integratedly packed into a single chip processor which is connected as a component of the torus network in the system. This paper describes the performance based on several evaluation programs on the pEDDEN (personal-EDDEN) system with 64 processor elements. Combination of dataflow architecture and vector operation control function was proved to be efficient by the programs. The pEDDEN system has recorded 300 MFLOPS on the Livermore Loop benchmark, and recorded 63.0 MB/sec, 56.1 Msync/sec on another benchmark program.

1. はじめに

近年、計算機の処理性能向上を目的とした並列処理技術が注目を集めており、並列処理計算機の実用化に対する期待が高まっている。中でもデータ駆動アーキテクチャは、①問題に内在する並列性を自然に抽出することができる、②命令の実行が関数的となるためプロセッサ内部における命令実行のパイプライン化が容易である、③プロセッサ間の通信と同期のメカニズムを含んだアーキテクチャでありデータ転送のレイテン

シーに強い、などの点で大規模並列処理に適したアーキテクチャであると考えられている¹⁾⁻³⁾。

我々は、データ駆動アーキテクチャに基づいた実用的な並行処理計算機の研究を進めており、プロセッサ数 1000 規模の高並列データ駆動計算機 EDDEN を開発中である^{4), 5)}。EDDEN ではその要素プロセッサとして、1) 32 ビット浮動小数点演算器を備えた循環パイプライン方式の数値演算機構、2) 定型的な繰り返し演算を高速に行うためのベクトル演算機構、3) オペランド対生成のための待ち合わせ機構およびその待ち合わせ用メモリ、4) プロセッサ結合網の要素として自己ルーティングを実現するための通信制御機構、を 1 チップの CMOS-LSI 上に実現しており、このプロセッサ LSI を多数結合することによりシステムを構

[†] 三洋電機(株)東京情報通信研究所コンピュータ基礎技術研究室

Computer Research Laboratory, Tokyo Information & Communication Research Center, SANYO Electric Co., Ltd.

成している。すでに高並列処理システムの開発に向けての第一段階として、この要素プロセッサを 64 台接続した中規模並列処理システム pEDDEN (personal-EDDEN) を開発しており、同機は 1991 年 9 月より稼働中である⁹⁾。

今回、特にこのデータ駆動計算機 EDDEN の基本性能の中でも、1) 数値演算処理性能、2) 待ち合わせ処理性能、3) 通信性能、の 3 点に注目して性能評価を行った。これらの評価は、各種のベンチマークプログラムならびに幾つかのアプリケーションプログラムを、中規模並列処理システム pEDDEN 上で実行することにより行った。さらに、pEDDEN 上で流体解析をはじめとする幾つかのアプリケーションプログラムを実行させた時の実効性能を測定し、より実用的な問題に対する適応性についての評価を行った。

本論文では、まず EDDEN の基本アーキテクチャと実験に用いた中規模並列処理システム pEDDEN について述べ、次に今回行った基本性能評価の実験方法とその結果を示す。また、幾つかのアプリケーションプログラムによる実効性能の評価を行い、得られた評価結果について考察を行う。最後にこれらの性能評価に基づき、EDDEN の今後の課題について述べる。

2. データ駆動計算機 EDDEN

EDDEN は複数の要素プロセッサがトラス状に相互結合された分散メモリ型並列計算機であり、データ駆動モデルに基づいてすべての要素プロセッサが同期式クロックで制御される、高並列データ駆動計算機である。各々の要素プロセッサは、我々が独自に開発したシングルチップのプロセッサ LSI と、局所メモリのみで構成される。このプロセッサ LSI は、データ待ち合わせ機能、演算実行機能などの実行制御機能とともに、結合網の要素としてのデータ交換機能を内蔵しているため、ルーティング制御のための外付け回路は不要であり、したがって要素プロセッサ同士を直接結合することで結合網を構成することができる。

プロセッサ LSI の命令実行部は、データ駆動アーキテクチャに基づく循環パイプライン上にあって、32 ビットの浮動小数点演算器を備えている。さらに循環パイプラインとは独立に、局所メモリ上のデータとの間で直接演算を行うベクトル演算機構を装備してい

る。これにより、データ駆動計算機が本質的に得意としない定性的な繰り返し演算を高速に処理することができる。また、待ち合わせ機構としては連想記憶方式を採用している。待ち合わせメモリは 2 段階に階層化されており、制御回路とともにプロセッサ LSI 上に実装されている。さらに EDDEN ではトラス/メッシュ結合網を採用しており、各プロセッサ LSI は隣接するプロセッサとの結合のために、16 ビット幅の双方向通信リンクを 4 本備えている。

中規模並列システム pEDDEN では、16 台の要素

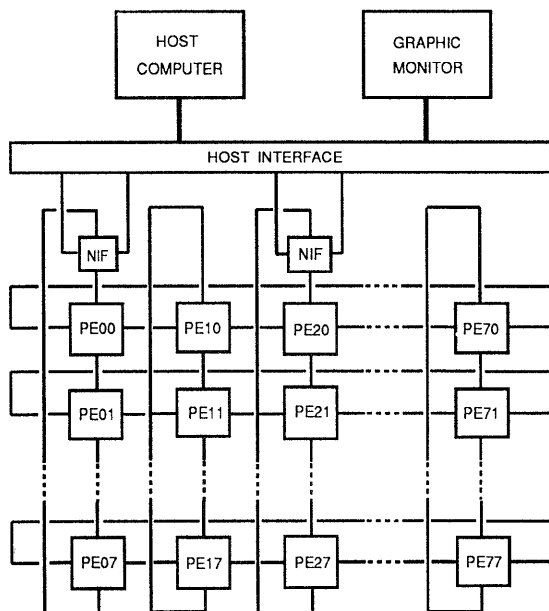


図 1 中規模並列システム pEDDEN の基本構成
Fig. 1 The organization of the pEDDEN.

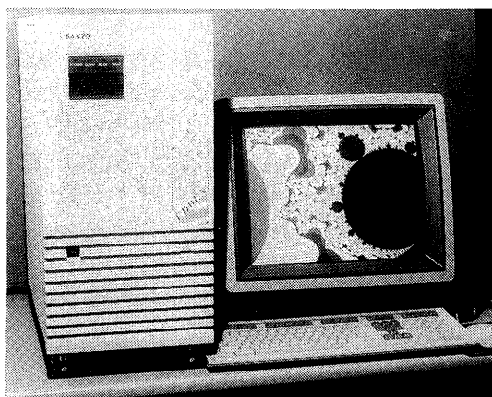


図 2 中規模並列システム pEDDEN の外観
Fig. 2 General view of the pEDDEN.

プロセッサを1枚のプロセッサ基板上に実装し、このプロセッサ基板4枚を幅30cm、奥行き50cm、高さ66cmの筐体に収めている。プログラムおよびデータ格納部として1プロセッサあたり1MBの局所メモリを備え、全体で64MBのメモリを有している。また、演算結果の即時表示機能を備えていて、グラフィックモニタを直接接続することができる。pEDDENの構成図と全体像を、図1および図2に示す。

3. 基本性能評価

今回、EDDENの基本性能のうち、特に①数値演算処理性能、②データ待ち合わせ処理性能、③プロセッサ間通信性能、の3項目について評価を行った。

3.1 数値演算処理性能の評価

EDDENでは、特にピュアなデータフロー計算機が得意としない定型的な繰り返し演算を高速に処理するために、演算部にベクトル演算機構を導入した。これは局所メモリ上のデータ群に対し、循環パイプラインを経由せずに連続的に演算を実行するものであり、繰り返し演算のように循環パイプラインが充足されないような逐次演算をより効率的に行うことを目的としている。図3に、EDDENのプロセッサLSIの内部構成図を示す。

図に示すとおり、ベクトル演算機構は命令実行部と局所メモリの間に位置し、メモリ上のデータ群に対して直接演算を施すことができる。ベクトル演算実行時には、循環パイプライン上で実行される通常のスカラ演算と、ベクトル演算機構によるベクトル演算が、時分割で1つの演算器を共用する。

今回、このベクトル演算機構の導入により数値演算処理性能がどの程度向上したかを、リバモアループによるベンチマークテストを用いて評価した。評価にはリバモアループの中からベクトル化の可能ないくつかのループを選び、pEDDEN上での実行時間を測定してFLOPS値を計算した。それぞれのループについて、ベクトル演算機構を用いた場合と、用いなかった場合との比較結果を表1に示す。これらはループ間でデータの依存性がないので、ループごとに並列化し各PEに割り付けた。またスカラ演算についてはさらに細かく並列化し、PE内で多重実行して循環パイプラインを常に充足させるように配慮した。

実験結果に示されるとおり、ベクトル演算機構を用いることにより最高300MFLOPSに近い実効性能を引き出せることが確認できた。EDDENのベクトル演算機構は10MHzのクロックで動作しており、1PEあたりの最大処理性能は単項演算で10MFLOPSである。しかしリバモアループで実行されるような2項演算の場合、2つのオペランドを逐次にメモリから読み出さなければならないため、演算は2クロックに1回しか実行されず、最大処理性能は1PEあたり5MFLOPS、システムでは320MFLOPSになる。よって今回の実験では、実効性能として最大性能の93%以上を引き出していることになり、ベクトル演算機構による命令実行部の使用効率が非常に高いことがわかる。このように高効率でベクトル演算機構が使用できるのは、ベクトル演算の前処理および後処理にあたるスカラ演算の部分をデータ駆動方式で実行していて、これが命令実行部においてベクトル演算と時分割で処理されるためその処理時間が見かけ上隠蔽され、これによりベクトル演算の処理の立ち上がりが非常に鋭くなっているからである。

また、LOOP12のように単純な計算の繰り返し処

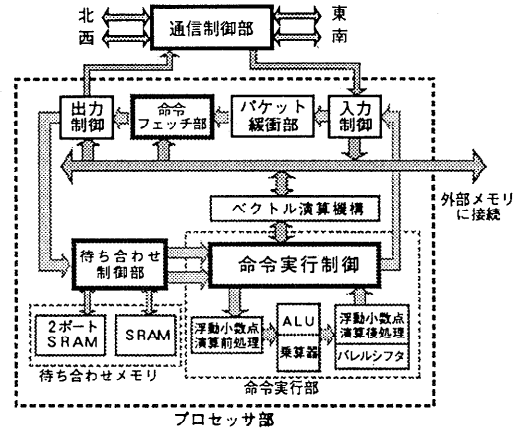


図3 プロセッサLSIの内部構成
Fig. 3 EDDEN processor LSI.

表1 リバモアループによる数値演算処理性能の評価
Table 1 Performance evaluation using the Livermore Loop benchmark.

	処 理 性 能 [MFLOPS]					
	LOOP 1	LOOP 3	LOOP 7	LOOP 9	LOOP 12	LOOP 21
スカラ演算	35.0	23.0	58.8	44.8	11.7	30.5
ベクトル演算	292.	152.	298.	197.	259.	236.

理では、ループのためのオーバーヘッドが相対的に大きくなるため、データ駆動方式によるスカラ演算では多重実行しているにもかかわらず十分な性能が得られていないが、特にこうした問題に対してベクトル演算機構が大きな効力を発揮できることが明らかになった。ちなみに、LOOP 12 の処理内容は次のとおりである。

LOOP 12:

```
for(k=0; k<N; k++)
```

```
  X[k]=Y[k+1]-Y[k];
```

3.2 待ち合わせ処理性能の評価

動的なデータ駆動計算モデルは、データフローグラフの1つのリンク上にタグによって識別される複数のトークンの存在を許すことで、再入可能性などの柔軟な記述を可能にしている。一般に、動的データ駆動モデルを実現する待ち合わせ機構として、ハッシュなどを用いた連想記憶方式によるもの⁷⁾と、直接マッチング方式によるもの⁸⁾があるが、ハッシュを用いた連想記憶方式による待ち合わせ機構では、

- ①一回の連想処理に要する時間が大きい。
- ②ハッシュ衝突時の処理オーバーヘッドが大きい。
- ③ハッシングのためのハードウェアコストが大きい。

などの問題点が指摘されており、直接マッチング方式によるものが有利であるとされている。しかし直接マッチング方式は、待ち合わせ制御の論理そのものは単純であるが、メモリの使用効率が悪く待ち合わせのための大きなメモリが必要になるため、待ち合わせ機構全体のハードウェア量は連想記憶方式に比べはるかに大きくなり、現状の半導体素子技術では待ち合わせメモリを含めての要素プロセッサの1チップ化を実現することは、極めて困難であると思われる。

EDDEN では、待ち合わせメモリを含めての1チッププロセッサを実現するために、待ち合わせ処理に要するハードウェア量の削減を図っている。そこでその待ち合わせ機構としては、処理効率に優れる直接マッチング方式ではなく、メモリ使用効率に優れる連想記憶方式を選択し、待ち合わせメモリ量の節減を実現した。そのために直接マッチング方式を取る計算機に比べ多少の処理効率が犠牲になっているが、今回この処理効率の低下がどの程度のものなのかを、幾つかのテストプログラムを実行させることにより評価した。

EDDEN の待ち合わせ機構は、図4に示すとおりハッシュメモリとバッファメモリの2段階の待ち合わせメモリを持っていて、ハッシュ衝突が起こらない場合の待ち合わせをハッシュメモリ上で直接行う。また、

ハッシュ衝突が生じた場合は、バッファメモリ上に順次チェイニングして格納していく。ここで、ハッシュメモリ上で直接待ち合わせを行う場合のマッチング処理は最短時間(1 clock)で行えるため、本方式における処理効率の低下は、ハッシュ衝突の回数に依存すると言える。そこで、まず幾つかの簡単なテストプログラムを用いてハッシュ衝突の状況を観察した。評価に用いたテストプログラムは、

- (1) SOR 法によるポアソン方程式の求解
- (2) ガウス掃き出し法による連立1次方程式の求解
- (3) クイックソート
- (4) マンデルブローのフラクタル図形描画

の4種類である。これらについて、ハッシュ空間の大きさを16から128まで変えた時の消費バッファ数を測定し、ハッシュ衝突の状況を調べた。表2に示されるとおり、今回のように規模もそれほど大きくなく(ノード数1500程度)、カラーを使わないような場合においては、ハッシュ空間が128もあれば十分であり、またハッシュ衝突の回数も極めて少ないことが明らかになった。

次に、カラーを用いた関数共用や関数の再帰呼び出しが、ハッシュ衝突にどのような影響を及ぼすかを、画像のアフィン変換プログラムとフィボナッチ関数の計算プログラムを用いて観察した。図5に示すとおり、関数共用に関してはハッシュ衝突にそれほど大き

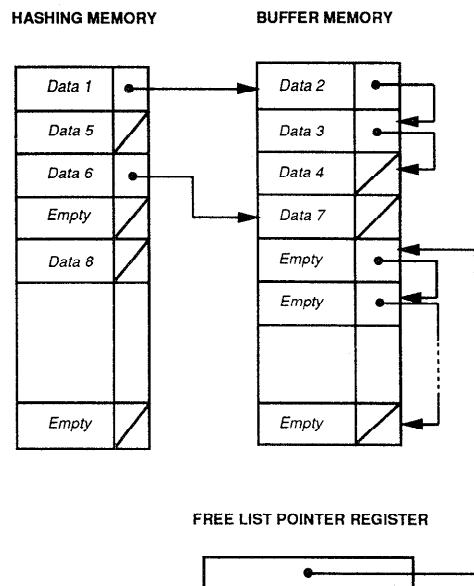


図4 EDDEN の待ち合わせ機構
Fig. 4 Matching mechanism of the EDDEN.

な影響を及ぼしていないことがわかる。これに対し図 6 が示すとおり、再帰呼び出しにおいては消費ハッシュ数に対する消費バッファ数が非常に大きく、ハッシュ衝突に対して深刻に影響することが明らかになった。しかもバッファメモリ上の連鎖が長くなるため、待ち合わせの処理オーバーヘッドがかなり大きくなっていることが容易に推測できる。こうした事態が生じるのは、ハッシュキーにカラーの情報が全く反映されていないからである。当初から EDDEN では、1 プロセッサあたりのカラーを 16 しか用意しておらず、こうしたネストの深い再帰問題は対象としていなかった。しかしこのように大きな処理効率の低下は、できるだけ避けるのが望ましい。EDDEN のハッシュキーは可変になっているので、現在カラーの一部をハッシュキーとして用いた場合について検討している。

表 2 ハッシュ衝突によるバッファメモリの消費数
Table 2 Buffer memories used when hash conflict occurs.

ハッシュ空間		SOR	GAUSS	QSORT	MANDEL
128	ハッシュ数	98	94	60	127
	バッファ数	0	1	0	6
64	ハッシュ数	62	54	46	64
	バッファ数	3	1	0	11
32	ハッシュ数	32	32	30	32
	バッファ数	8	1	1	11
16	ハッシュ数	16	16	16	16
	バッファ数	15	2	1	19

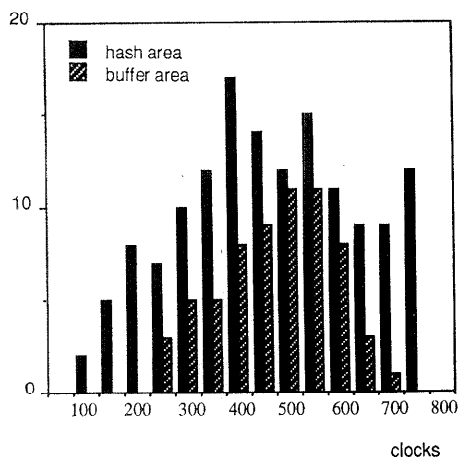


図 5 関数共用による影響

Fig. 5 Influences of duplicate call of the function.

3.3 通信性能の評価

EDDEN ではプロセッサ LSI に通信制御機構を内蔵しており、自己ルーティング方式による通信制御が行われる。したがってルーティングのための外部回路は不要であり、プロセッサ同士を直接結合することができる。

今回の評価は、通信性能の中でも特に並列性能に関わる大きな要因である、各プロセッサのデータ転送性能と、プロセッサ間の同期性能に注目して行った。

3.3.1 データ転送性能

データ転送性能については、2 種類のベンチマークプログラムによって評価を行った。

まず最初に、1つのプロセッサが別の1つのプロセッサの局所メモリ上にあるデータを順次読み出して加算する、というベンチマークプログラムを実行した。この場合、データ転送のレイテンシーはプロセッサ間の距離に依存するので、2つのプロセッサの選び方を4通りに変えて実験を行った。一加算あたりの平均の処理時間を表 3 に示す。ここで単方向通信の場合は、

表 3 データ転送性能の評価 (1)
Table 3 Data transfer performance (1).

データを転送する プロセッサの組み合わせ	処理時間 [μsec]	
	単方向通信	双方向通信
1) PE(0,0) $\leftrightarrow$ PE(1,0)	10.91	8.10
2) PE(0,0) $\leftrightarrow$ PE(1,1)	13.71	8.91
3) PE(0,0) $\leftrightarrow$ PE(4,4)	13.71	13.71
4) PE(0,0) $\leftrightarrow$ PE(7,7)	13.71	9.31

注：データ転送を行わない時のループの処理時間：6.82 μsec

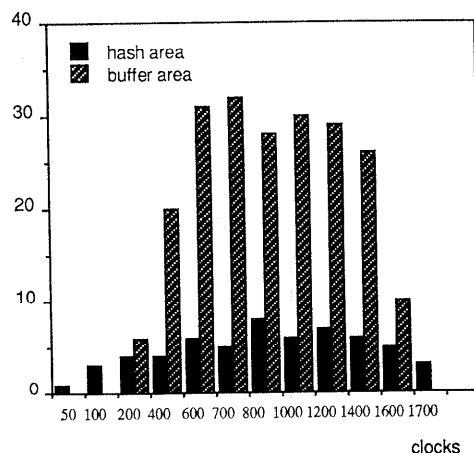


図 6 再帰呼び出しによる影響

Fig. 6 Influences of recursive call of the function.

組み合わせ 1) 以外はすべて同一距離（最長距離：要求パケットもしくは結果パケットが x, y 方向にそれぞれ 1 周ずつ転送される）であるため、処理時間が同じになっている。これに対し双方向通信の場合は、プロセッサ間の距離がすべて異なっており、それらの距離に比例して処理時間が増大しているのがわかる。ここで PE (0, 0) と PE (7, 7) の間にはホスト計算機とのインタフェースがあるため、組み合わせ 1) より組み合わせ 4) の方が距離が長くなっている。また、同じ処理でデータ転送を行わず、自プロセッサのデータを順次加算していく処理では、一加算あたりの処理時間が $6.82 \mu\text{sec}$ であり、これと表 3 の結果から隣接プロセッサへの転送時間を計算すると、組み合わせ 3) の場合で 430 nsec となる。これにより転送時間の理論最短値 400 nsec に近い値が実効値で得られていることがわかる。

上述の評価では、常に通信経路がクリアな状態でのレイテンシーを考えているが、次の評価では通信経路上が多数のパケットで混雑している場合について考えた。この評価には、あるプロセッサが他のすべてのプロセッサの局所メモリ上のデータを読み出し、その和を求めるというテストプログラムを用いた。そしてこのテストプログラムを複数のプロセッサ上で並列に実行し、通信量を増大させることにより通信経路の混雑を引き起こして、その時の処理オーバヘッドの増加状況について評価した。結果を表 4 に示す。

この問題において、加算処理本体の実行時間が別の測定で $13.8 \mu\text{sec}$ であることがわかっており、並列度 1 の時の他プロセッサへの平均的なメモリ読み出し時間は $0.53 \mu\text{sec}$ と計算される。また、並列度 64 の時と同じ負荷を与えた時の加算処理本体の実行時間が $31.0 \mu\text{sec}$ であることが測定よりわかっており、これから並列度 64 の時の平均的なメモリ読み出し時間は、計算により $4.26 \mu\text{sec}$ であることがわかる。この差がパケットの混雑による通信の遅延であると考えられ

る。

3.3.2 プロセッサ間同期性能

次にプロセッサ間の同期性能についての評価を行った。評価には、各プロセッサに単純なループを実行させ、1 回のループが終了するごとにプロセッサ間で同期をとる、というテストプログラムを用いた。そしてプロセッサ数を増加させることにより同期に要する処理時間がどのように増大していくかを測定した。評価結果を表 5 に示す。ここで 1 PE だけにループ本体を実行させ、同期をとらなかった場合の処理時間は平均して $4.64 \mu\text{sec}$ であったので、このことから 64 プロセッサ全体の同期処理がたかだか $36.2 \mu\text{sec}$ で行えることが明らかになった。

同期性能について、もう少し詳細に分析してみる。隣接プロセッサ間の同期の処理時間の実測値は、表 5 の双方向通信の 2 PE での値 $8.05 \mu\text{sec}$ からループ本体の処理時間 $4.64 \mu\text{sec}$ を引いた $3.41 \mu\text{sec}$ となる。よってシステム全体の同期性能は、 18.7 Msync/sec になる。一方 EDDEN では、パケットの出力に 5 クロック、入力に 6 クロックかかり、内部の循環パイプラインが 11 段あるので、同期のためのパケットが隣接するプロセッサに行って返ってくるのに要するクロック数は、 $(5+6) \times 2 + 11 = 33 \text{ clocks}$ となる。さらに、同期処理に伴う余分のパケットコピーが 1 回生じるので、同期処理に要する時間の理論値は 34 clocks となり、実測値とはほぼ一致する。

ここでデータ転送のレイテンシーによるプロセッサの遊休時間をなくすために、このテストプログラムを多重実行させてみると、5 並列実行までは同期の処理時間が変わらなかった。すなわち、多重度 6 ではじめて内部の循環パイプラインが充足されたことになる。したがって同じプログラムを全プロセッサで実行することを考えると、半分の多重度 3 でパイプラインが充足されることが予想され、この時の全体のデータ同期

表 4 データ転送性能の評価 (2)

Table 4 Data transfer performance (2).

並列度数	処理時間 [μsec]
1	47.5
2	50.4
4	64.8
8	85.0
16	129.6
32	195.8
64	299.5

表 5 同期性能の評価

Table 5 Evaluation of synchronization cost.

PE 数	処理時間 [μsec]	
	単方向通信	双方向通信
1 PE	4.64	
2 PE	10.8	8.05
4 PE	11.3	10.5
8 PE	12.8	12.0
16 PE	19.2	16.1
32 PE	26.7	24.8
64 PE	40.8	40.8

性能は 56.1 Msync/sec となる。

3.3.3 構造体通信

これまで述べてきたデータ転送性能の評価では、ランダムなデータ転送のみを考えていたが、ここであるプロセッサの局所メモリ上に置かれている大量のデータ群（例えば配列データなど）をまとめて転送する場合を考えてみる。一般にデータ駆動計算機において、プロセッサ間でデータを転送する際、各々のデータにヘッダ情報をつけパケット化して転送するので、まとまったデータ群を転送する場合にはパケット化のための処理オーバーヘッドが全体の処理時間に大きく影響する。そこで EDDEN では、配列データなどのひとまとまりのデータ群を効率良く転送する手段として、構造体通信機能をサポートしている。これは最大 4095 個までのデータ群をひとまとまりのデータとして扱い、先頭データのみにヘッダ情報を付けて構造体パケットと呼ばれる 1 つのパケットに構成することにより、転送処理の効率化を図るものである。ただしこの構造体通信はデッドロック防止のため、隣接するプロセッサ間でのみ許される。

EDDEN における構造体通信導入の効果を評価するため、あるプロセッサの局所メモリ上に置かれている 200×200 の大きさの 2 次元配列データを、全プロセッサにブロードキャストするというテストプログラムを実行し、ノーマルパケットによる通常通信と構造体パケットによる構造体通信との通信時間を比較した。前述のとおり、EDDEN では構造体通信は隣接するプロセッサ間にしか適用されていないが、今回の評価実験では構造体通信の方法として、①全プロセッサを数珠つなぎにして順次転送を行っていく逐次転送と、②まず南北方向の 7 プロセッサに転送してからそれぞれのプロセッサから東西方向に 8 並列で転送する並列転送、の 2 種類を選んだ。実験結果を表 6 に示す。この結果から、構造体通信を用いることで通常の 20 倍以上の通信性能を得られることが明らかになった。このように構造体通信機能が低い通信性能を示すのは、先に述べたパケットの効率化以外に、ベクトル演算機構導入の効果が大きい。すなわち通常のスカラパケットが循環パイプラインを回って生成されるのに対し、構造体パケットはベクトル演算機構により循環パイプラインを介さずに直接生成されるためである。浮動小数点演算器を持つようなプロセッ

サでは、どうしても循環パイプライン長が長くなってしまい、そのためノーマルパケットを生成するような単純なループ処理では、パイプラインバブルによる効率の低下が著しい。こうしたことから、ベクトル演算機構を利用した EDDEN の構造体通信機能が、非常に有効であることが明らかである。

4. アプリケーションによる評価

これまで簡単なテストプログラムにより EDDEN の基本性能についての評価を行ってきた。ここで、より実用的なアプリケーションプログラムを実行することにより、実用に近いレベルでの演算処理性能を測定し、EDDEN の実効性能について評価する。評価に用いたアプリケーションは、次のとおりである。

- (1) アフィン変換による画像の拡大回転
- (2) レイトレーシング法による画像レンダリング
- (3) 2 次元粘性流体シミュレーション
- (4) FFT による 2 次元フーリエ変換、逆変換

今回は、これらのアプリケーションプログラムを C 言語で記述し、いくつかの計算機上で実行してその処理時間を比較した。比較に用いた商用計算機は、

- SPARC STATION 2 (4.2 MFLOPS)
- TITAN/300 (64 MFLOPS)
- CONVEX/C 200 (100 MFLOPS)

の 3 つである。評価結果を表 7 に示す。

実験結果より、例えばレイトレーシングのように並列性が高くかつベクトル化率が低い問題に対して、EDDEN が特に優れた性能を示しているのがわかる。また、アフィン変換や流体シミュレーションのように

表 6 構造体通信の評価 (配列データ転送)
Table 6 Vector transfer performance.

通常通信	構造体通信	
	逐次通信	8 並列通信
3.40 秒	0.66 秒	0.16 秒
2.96 MB/秒	15.2 MB/秒	63.0 MB/秒

表 7 アプリケーションによる演算処理性能の評価
Table 7 Parallel performance using several applications.

アプリケーション	処 理 時 間			
	pEDDEN	SS 2	TITAN	CONVEX
画像アフィン変換 [ミリ秒]	5.40	102.2	34.9	17.8
レイトレーシング [秒]	4.22	91.0	55.2	75.8
流体シミュレーション [秒]	713.5	1482.7	1351.5	766.7
高速フーリエ変換 [秒]	0.40	9.16	4.92	4.35

ベクトル化の容易な問題でも、並列性が高ければ十分な性能を引き出せることが、実験により明らかになった。さらに、アフィン変換についてはベクトル演算機構を使用しなかった時の実行時間が 11.6 ミリ秒であり、ベクトル演算機構の導入で 2 倍以上の高速化を実現できたことがわかる。一方、2 次元 FFT についてはベクトル化に適したアルゴリズムを選ばなかったこともあってか、ベクトル演算機構の導入効果が 30% 程度の高速化にとどまった。

今回の実験では、アフィン変換と 2 次元 FFT についてはベクトル演算機構を利用して評価しているが、レイトレーシングと流体シミュレーションについてはスカラ演算のみで評価している。これらの問題についても、ベクトル演算機構を利用することでより一層の性能向上が予想できるが、現在このベクトル演算機構はライブラリを用いて利用しなければならない状況であり、ユーザがベクトル化を意識してプログラミングする必要がある。したがってここに挙げたような規模の大きな実用レベルのアプリケーションでは、ベクトル演算機構が利用しにくい状況にある。今後の課題として、コンパイラが自動ベクトル化をサポートすることが必須であると考えている。同時に、構造体通信もコンパイラがサポートしていく必要がある。これによって EDDEN の実用性がさらに向上するものと思われる。

5. おわりに

我々が開発中の高並列データ駆動計算機 EDDEN について、基本性能と実効性能の評価を行った。基本性能については、64 プロセッサからなる中規模並列システム pEDDEN 上で評価プログラムを実行することにより、1) 数値演算処理性能、2) 待ち合わせ処理性能、3) 通信性能、の 3 項目についての評価を行った。その結果、リバモアループによるベンチマークテストで実効性能 300 MFLOPS を得ることができ、EDDEN のベクトル演算機構が有効に作用していることが確認できた。また、このベクトル演算機構が構造体通信機能を通じて、通信性能の向上にも大きく寄与していることが明らかになった。さらに、待ち合わせ機能に関しては、そのハードウェア量から考えて十分な処理性能を有していることが立証された。ただし、ネストの深い再帰問題に対しては処理効率が低下するという問題点も明らかになった。通信性能としては、ベンチマークにより 63.0 MB/sec, 56.1 Msync/sec を実効

値として得ることができた。さらに、実用的なアプリケーションに対しても高い演算性能を引き出すことができ、実用に向けての見通しを立てることができた。

今後は、コンパイラ、デバグをはじめとするソフトウェア環境を拡充して、実用性を向上させていく予定である。

参考文献

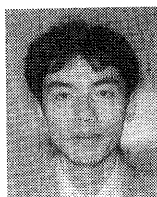
- 1) Dennis, J.B. and Misnas, D.P.: A Preliminary Architecture for a Basic Dataflow Processor, *Proc. 2nd Ann. Int. Symp. Computer Architecture*, pp. 126-132, IEEE (1975).
- 2) Arvind, Dertouzos, M. L. and Iannucci, R. A.: A Multiprocessor Emulation Facility, MIT-LCS Technical Report 302 (1983).
- 3) Gurd, J.R., Kirkham, C.C. and Watson, I.: The Manchester Prototype Dataflow Computer, *CACM*, Vol. 28, No. 1, pp. 34-52 (1985).
- 4) 三浦ほか: データ駆動計算機 EDDEN の概要, 第 38 回情報処理学会全国大会論文集, 2T-2 (1989).
- 5) 三浦ほか: データ駆動計算機 EDDEN のアーキテクチャ, 情報処理学会論文誌, Vol. 32, No. 7, pp. 838-848 (1991).
- 6) 大橋ほか: データ駆動計算機 EDDEN の中規模並列システムの開発, 第 43 回情報処理学会全国大会論文集, 7Q-2 (1991).
- 7) 平木, 西田, 島田: 科学技術計算用データ駆動計算機 Sigma-1 のハッシングによるマッチングユニットの評価, 第 26 回情報処理学会全国大会論文集 (1983).
- 8) 坂井, 平木, 山口, 児玉, 弓場: データ駆動計算機のアーキテクチャ最適化に関する考察, 情報処理学会論文誌, Vol. 30, No. 12, pp. 1562-1571 (1989).
- 9) 三浦ほか: データ駆動計算機 EDDEN の要素プロセッサ LSI, 第 42 回情報処理学会全国大会論文集, 6W-3 (1991).
- 10) 岡本ほか: データ駆動計算機 EDDEN の発火制御機構の評価, 第 43 回情報処理学会全国大会論文集, 7Q-3 (1991).
- 11) 石川ほか: データ駆動計算機 EDDEN におけるレンダリング処理, 第 44 回情報処理学会全国大会論文集, 3D-3 (1992).
- 12) 岡本ほか: データ駆動計算機 EDDEN の基本性能評価, 並列処理シンポジウム JSPP '92 予稿集, pp. 337-344 (1992).

(平成 4 年 9 月 16 日受付)

(平成 5 年 1 月 18 日採録)

**岡本 一晃 (正会員)**

1962 年生. 1986 年慶應義塾大学理工学部電気工学科卒業. 同年三洋電機(株)に入社. 以来, 計算機アーキテクチャ, 特にデータ駆動計算機などの並列処理計算機の研究に従事. 同社東京情報通信研究所に所属. 1992 年 10 月より(技組)新情報処理開発機構へ出向, 現在に至る.

**川口 正樹 (正会員)**

1961 年生. 1986 年東京理科大学理工学部物理学学科卒業. 同年三洋電機(株)入社. 現在, 同社東京情報通信研究所に所属. 並列プログラミング環境, 数値シミュレーション手法等の研究開発に従事.

**三浦 宏喜 (正会員)**

1959 年生. 1982 年京都大学工学部情報工学科卒業. 同年三洋電機(株)に入社. デジタル信号処理, 計算機アーキテクチャ, 並列処理, データフローマシン等の研究に従事. 現在, 同社東京情報通信研究所に所属.

**清水 雅久**

1952 年 9 月 13 日生. 1975 年大阪大学工学部通信工学科卒業. 1977 年同大学大学院工学研究科修士課程修了. 同年三洋電機(株)に入社. 現在同社東京情報通信研究所に所属. デジタル信号処理, 並列処理システムの研究に従事. 電子情報通信学会会員.