計算機アーキテクチャ 91−11
（1991. 11. 22）

並列計算機ＡＤＥＮＡＲＴにおける データの分散配置方式

森康浩　右田学　若谷彰良　岡本理
松下電器産業（株）半導体研究センター
Email:ymori@vdrl.src.mei.co.jp

並列計算機ＡＤＥＮＡＲＴにおいては、ホスト計算機上の配列データは複数のプロセッサエレメント（ＰＥ）に分散配置され、各ＰＥはこの分散配置されたデータを独自のネットワーク（ＨＸｎｅｔ）を用いて交換、編集しながら処理を行ない、処理結果は各ＰＥから再びホスト計算機上の配列データに戻される。このため、ＨＸｎｅｔの特徴を生かした効率的なＰＥ間転送を可能にさせる最適なデータの分散配置方式が必要となる。本稿では、本計算機における配列データの分散配置方式について概説し、これを高速に実現するためのハードウェア機構（ＩＵＰＬＵＳ）の構成について述べる。最後に、このハードウェア機構を実装した上でのＡＤＥＮＡＲＴのシステム性能の評価に関して報告する。

The data distribution method in the parallel computer ADENART

Yasuhiro Mori, Manabu Migita, Akiyoshi Wakatani, Tadashi Okamoto
Matsushita Electric Industrial Co.,Ltd. Semiconductor Research Center
3-15, Yakumonakamachi, Moriguchi 579, Japan

In the parallel computer, ADENART, data on the host-computer is distributed to each processor element(PE). Each PE proceeds with their computation while exchanging and editing data via the unique network called HXnet. The result data on each PE is combined and again sent back to the data-set on the host computer. In order to realize efficient inter-processor communication featuring advantages of HXnet, appropriate data distribution strategy is crucial. In the report, we summerize the datadistribution method in ADENART and show the acceralating mechanism andthe hardware structure for it.

1．はじめに

我々は、京都大学工学部（野木達夫助教授）と共同で、並列計算機ＡＤＥＮＡＲＴ（以前はＡＤＥＮＡと呼んでいた）を開発した。[1][2]ＡＤＥＮＡＲＴは２５６個のプロセッサエレメント（以後ＰＥと呼ぶ）を用いたＭＩＭＤ型の並列計算機であり、ＰＥ間通信ネットワーク（以後ハイパークロスネットと呼ぶ）によって２回のＰＥ間データ転送により、すべてのＰＥ間のデータ転送が可能である。このハイパークロスネットの特徴を生かすため、各ＰＥに対して最適な配列データの分散配置が必要となる。また、ＡＤＥＮＡＲＴでは、並列動作を”陽に”意識してプログラミングするユーザーのために、ＦＯＲＴＲＡＮに並列化構文を追加したＡＤＥＴＲＡＮ[3]を用意している。一方、既にＦＯＲＴＲＡＮで開発されたプログラムをそのまま流用したいというユーザーに対しては、並列プログラムに変換する手段が必要となる。この手段の一つとして自動並列化コンパイラＡＰＡＲＣ[4]を開発中である。ＡＰＡＲＣによって生成される並列プログラムは、”陽に”並列動作を意識して記述されたプログラムに比べ、逐次処理部と並列処理部とのデータ転送が頻発することになり、このデータ転送を高速化することはシステム性能を上げる上で重要である。

以上のように、ＡＤＥＮＡＲＴのようにホスト計算機と並列処理用の”エンジン”－ＡＤＥＮＡＲＴ本体から構成され、ホスト計算機のデータを複数のＰＥで分散処理する形式の並列計算機システムにおいては、データの分散配置方式と、ホスト－ＡＤＥＮＡＲＴ本体間のデータ転送によって生じるオーバーヘッドが問題となる。

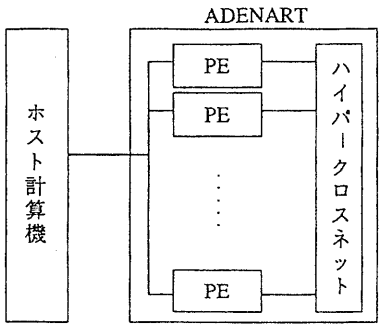
富士通のＡＰ１０００においては、データの分散／収集時のセル（ＡＤＥＮＡＲＴのＰＥにあたる）の切替えによるオーバーヘッドを解消するため、Ｂｒｏａｄｃａｔ　ｎｅｔｗｏｒｋと呼ぶネットワークと専用のＬＳＩを導入し、データのブロードキャストと同様の手順／速度でデータの分散／収集ができるようにしている。[5]これに対し、ＡＤＥＮＡＲＴでは、後述するようにハイパークロスネットの特徴を生かすため、データの分散配置にともなうオーバーヘッドが大きい。

本稿では、ＡＤＥＮＡＲＴにおけるデータの分散配置方式を示した後、これを高速化するためのハードウェア機構（ＩＵＰＬＵＳと呼ぶ）について触れ、最後にこれを実装した上でのＡＤＥＮＡＲＴのシステム性能の再評価に関して述べる。

2．並列計算機ＡＤＥＮＡＲＴ

2．１　ＡＤＥＮＡＲＴシステム

ＡＤＥＮＡＲＴシステムは、第１図に示すようにホスト計算機とＡＤＥＮＡＲＴ本体から構成される。

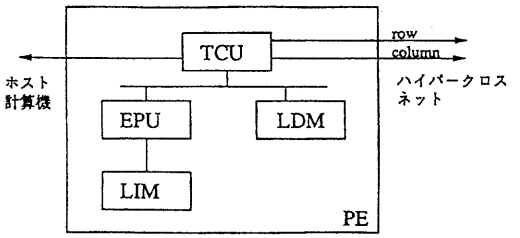


第１図　ADENARTシステム

ホスト計算機はＵＮＩＸ環境下でユーザーに対するあらゆるサービスを提供するとともに、プログラムの逐次処理部分を実行する。ＡＤＥＮＡＲＴ本体はMIMD型の並列計算エンジンで、プログラムの並列処理部分を実行する。

ＡＤＥＮＡＲＴ本体は、２５６個のＰＥとハイパークロスネットから構成される。一つのＰＥは、ピーク性能２０ＭＦＬＯＦＳの浮動小数点演算プロセッサ：ＥＰＵ(Element Processing Unit)、データ転送用コントローラ：ＴＣＵ(Transfer Control Unit)、命令メモリ：ＬＩＭ(Local Instruct

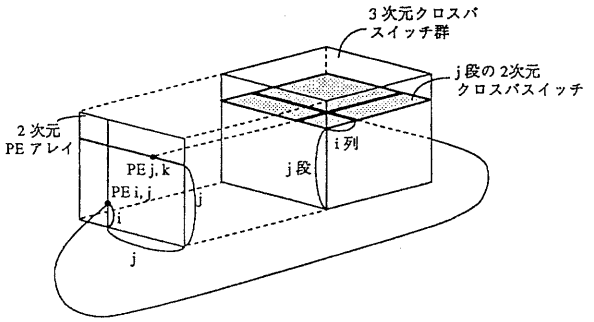
ion Memory, 24ビット＊64Kワード)、データメモリ：ＬＤＭ(Local Data Memory, 72ビット＊256Kワード, ECC8ビットを含む）の４つのパーツによって第２図の接続により構成される。



第2図　PEの構成

２．２　ハイパークロスネット

ハイパークロスネットはＡＤＥＮＡＲＴを特徴づけるＰＥ間接続ネットワークであり、自社開発した２種類のＬＳＩによって構成される。第３図にその接続を示すブロック図を示す。



第3図　ハイパークロスネットの構成

ハイパークロスネットは、１６×１６個の２次元のクロスバスイッチを３次元的に１６段重ねて配置したクロスバスイッチ群を構成し、１６×１６個のＰＥのそれぞれに２方向のバスを持たせ、その２方向のバスによって前記のクロスバスイッチに対して２方向から接続する形態になっている。すなわち、ＰＥi,jはｊ段ｉ列の１６個のクロスバスイッチを介してｊ行の１６個のＰＥ（ＰＥj,k k=0,15）と接続されることになる。ＰＥi,jがＰＥj,kに接続されているので、（式１）に示す２回のＰＥ間転送で任意のＰＥ間のデータ転送を実現できる。

$$PEk,l \leftarrow PEj,k \leftarrow PEi,j \qquad (1)$$

３．データ転送に伴う問題点

ＡＤＥＮＡＲＴでは、データの転送に伴って、次の２点の問題が発生する。

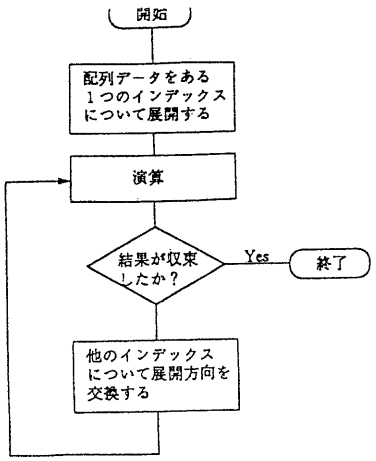
（１）データの再配置

（２）データの分散／収集

上記の問題点について、次にその詳細を述べる。

３．１　配列データの再配置

２章でＡＤＥＮＡＲＴのハイパークロスネットは、ＰＥi,jとＰＥj,kがクロスバスイッチ群を介して接続された形態のネットワークであることを述べた。ところで、このアーキテクチャーはもともと数値解法アルゴリズムの一つであるＡＤＩ法(Alternating Direction Implicit method)を高速に処理することを目的として考案されたものである。ＡＤＩ法によって問題を処理するとき、第４図に示すようなフローチャートに従って結果が収束するまで演算を繰り返す。ここで、配列データ：Ui,j,kをインデックス：ｉに着目して１次元化することをｉ方向に展開するという。また、演算ステップ間で行なうデータの展開方向を交換する操作をＡＤＥ(Alternating Direction Edition)操作といい、これは（式２－１，２，３）で示される。

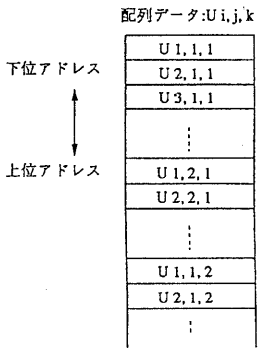


第4図　ADI法による問題処理手順

$$Ui/,j,/k \leftarrow Ui,/j,k/ \qquad (2-1)$$
$$U/i,j/,k \leftarrow Ui/,j,/k \qquad (2-2)$$
$$Ui,/j,k/ \leftarrow U/i,j/,k \qquad (2-3)$$

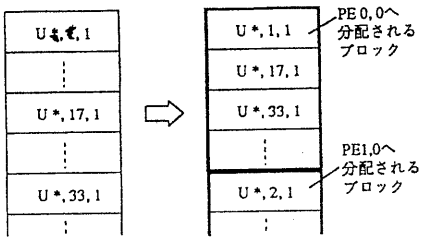
ハイパークロスネットはＰＥi,jとＰＥj,kが接続された形態のネットワークであるので、／／で囲まれたインデックスを論理的なプロセッサ番号として配列データを各ＰＥに割り当てるとき、（式２－１，２，３）は唯１回のＰＥ間転送によって実現されることになり、より効率的で自然なＡＤＥ操作を実現できる。現実には、ＰＥは有限個であるので、１つの物理ＰＥが複数の論理ＰＥを兼ねることになる。（ＰＥの多重使用）

一般に配列データ：Ui,j,kは第５図に示すようにＦＯＲＴＲＡＮコンパイラによってより左側にあるインデックスがより頻繁に変化するようにメモリー上に割り付けられる。



第5図　FORTRANコンパイラによる
配列データの配置

ＡＤＥＮＡＲＴでは配列データのインデックスによって配置先のＰＥを決定するため、前記したような理由により、あるＰＥに割り当てるデータはホスト計算機のメモリー上で分散していることになる。このため、このままでは最悪の場合１ワード毎に転送先のＰＥを切り替えなければならないといった事態が発生し、転送時のオーバーヘッドが大きくなり過ぎる。したがって、ＡＤＥＮＡＲＴでは第６図に示すように一つのＰＥに割り当てるデータ毎にブロック化して再配置する必要がある。このことに起因するオーバーヘッドは、データサイズの増加とともに増大する傾向にあり、転送効率を上げる上で障壁となる。



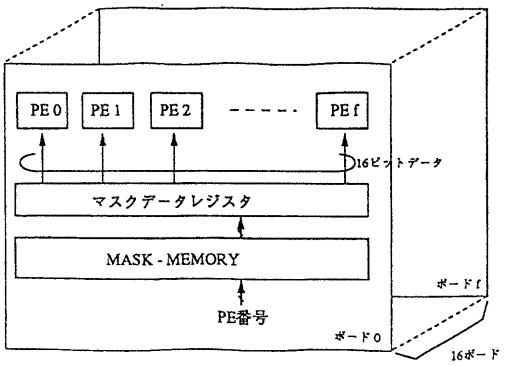
第6図　i方向へ展開する場合の
データのブロック化

3．2　データの分散／収集

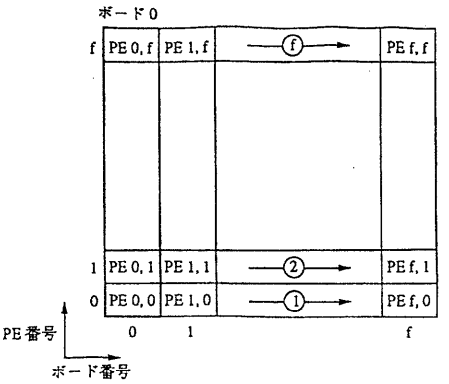
ブロック化されて再配置されたデータを各ＰＥのＬＤＭに分配するためには、１）データを切り分けて、２）転送先のＰＥを切替えながら転送するといったステップが必要である。

従来、ＡＤＥＮＡＲＴにおいては、このオーバーヘッドをおさえるため、第７図に示すハードウェア機構を設けている。[6]

ＡＤＥＮＡＲＴでは１枚のボードに１６個のＰＥが実装され、計１６枚のボードによって２５６個のＰＥが提供される。各ボードにはボード番号が付けられ、固有のアドレスが割り振られる。



第7図



第8図　PEのアクセス順序

また、各ＰＥにはボード上に物理的に実装された位置によってＰＥ番号が決められており、このＰＥ番号はボードに実装されているＭＡＳＫ－ＭＥＭＯＲＹのアドレスに置き換えられる。ＭＡＳＫ－ＭＥＭＯＲＹにはあらかじめ、あるＰＥ番号＝アドレスにそのＰＥだけが選択されるようなデータが記憶される。ＭＡＳＫ－ＭＥＭＯＲＹから出力されるデータの１ビット１ビットは、１つ１つのＰＥのチップセレクトとして入力される。

一つ一つのＰＥは、ボード番号とＰＥ番号によって特定され、識別されるが、これらは、実際には、ボードのアドレスとＭＡＳＫ－ＭＥＭＯＲＹのアドレスに置換されて用いられることになる。

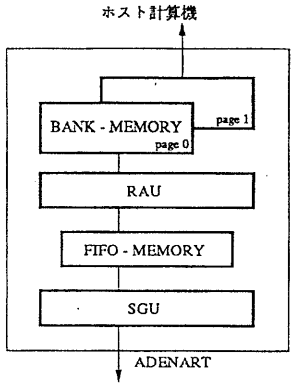
ＰＥへのアクセスは、第８図に示すようにＰＥ0,0, ＰＥ1,0と続き、ＰＥf,0までいって次の行へ入る。次行へ移行する際に全ボードにＰＥ番号をブロードキャストしておき、１ＰＥ／１ボードを選択しておく。その後、ボード番号をインクリメントしながら、データを送る。

したがって、ホストは各ボードに割り当てられたアドレスに対して連続的にデータを書き込んで行くことによって、ＰＥにデータを転送できるが、データの切りわけや転送先のＰＥのアドレス計算はホスト計算機で実行されているのでより高速化を図る必要がある。。

4. ＩＵＰＬＵＳ

ＩＵＰＬＵＳは、ホスト計算機とＡＤＥＮＡの間にあって、３章で述べたデータの再配置、並びに分散／収集によって発生するオーバーヘッドを小さくすることを目的とするハードウェア機構である。ＩＵＰＬＵＳのブロック構成図を第９図に示す。

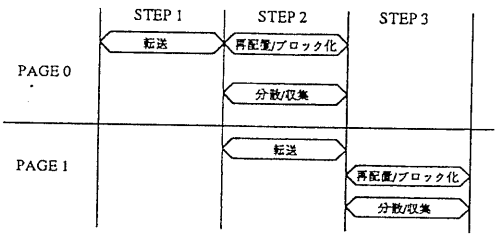
ＩＵＰＬＵＳは、ＢＡＮＫ－ＭＥＭＯＲＹ, ＲＡＵ(Re-Arrangement control Unit), ＦＩＦＯ－ＭＥＭＯＲＹ, ＳＧＵ(Scatter/Gather control Unit)の４つのブロックに分割できる。



第9図　IUPLUSの構成

第１０図はＩＵＰＬＵＳの動作をしめすチャート図である。

ホスト計算機のデータは、まず、ＢＡＮＫ－ＭＥＭＯＲＹに１次的に記憶される。（この時点では配列データはＦＯＲＴＲＡＮコンパイラの定めるフォーマットのままである。）つぎに、ＲＡＵは配列データの中から一つのＰＥに転送するデータを連続的に拾いだし、ＦＩＦＯ－ＭＥＭＯＲＹに詰め込んでいく動作をする。これによって、一つのＰＥに送るデータがブロック化されフォーマットの変換がなされることになる。（データの再配置）ＳＧＵは、ブロック化されて詰め込まれたデータをＦＩＦＯ－ＭＥＭＯＲＹから順に読み出して、順次転送先のＰＥを切替え、分散（収集）させる。（データの分散／収集）ここで、ＲＡＵとＳＧＵがＦＩＦＯ－ＭＥＭＯＲＹを挟んで非同期に動作できるので、データの最配置と分散／収集といったステップがオーバーラップして実行でき、これらを原因とするオーバーヘッドを削減し、システム性能を向上させることができる。また、ＢＡＮＫ－ＭＥＭＯＲＹとして２変数分＝２ページのメモリを用意してあるので、２変数以上のデータをＡＤＥＮＡＲＴに転送するとき、ホスト－ＡＤＥＮＡＲＴのデータ転送とデータの再配置、分散／収集もオーバーラップさせることができる。



第10図　IUPLUSの動作

5．性能予測

従来のシステム（ＡＤＥＮＡＲＴ）において、ｎワードのデータを転送する場合、データ転送に要する時間：Ｔは、データ再配置に要する時間：Ｒ(n)とデータの分散／収集に要する時間：Ｔｃ×ｐと、実際のデータ転送時間：Ｔｔ×ｎの和になり、（式３）で書き表される。

$$T = R(n) + Tc \times p + Tt \times n \qquad (3)$$

ｐ：ＰＥ数
Ｔｃ：ＰＥ切替え時間
Ｔｔ：１ワードのデータ転送時間

一方、ＩＵＰＬＵＳを実装したシステム（ＡＤＥＮＡＲＴ＋）の場合、データ転送時間：Ｔiuは、データ再配置と分散／収集がほぼオーバーラップされるため、データ再配置時間：Ｒiu（ｎ）と、実際のデータ転送時間：Ｔｔ×ｎの和となり、次式で書き表される。

$$Tiu = Riu(n) + Tt \times n \qquad (4)$$

ここで、Ｒ(n)、Ｒiu(n)はデータ再配置の時間関数であり、単純にデータのワード数：ｎにのみ依存する増加関数である。

第１表はＡＤＥＮＡＲＴにおいて、（３）を実測した結果である。

n	Ｒ（ｎ）	Tc*p+Tt*n
256k	0.28sec	0.14sec
2048k	1.49sec	1.0 sec

第１表　再配置の実行時間

ＡＤＥＮＡＲＴにおいては実際のデータ転送能力として2MB/sec程度あるが、再配置に要する時間を考慮すると、約1/3に落ちる。

ＡＤＥＮＡＲＴ＋においては、２回のメモリ間転送：

１）ホスト計算機→BANK-MEMORY,

２）BANK-MEMORY→FIFO-MEMORY

（FIFO-MEMORY→PEは２）にかくれる）によってデータの分散配置が実行される。したがって、１）あるいは２）のデータ転送速度で遅い方の速度が支配的となる。１）の転送速度は10MB/sec程度であるので、ＡＤＥＮＡＲＴ＋においてはＡＤＥＮＡＲＴの10倍程度の速度でデータの再配置が実行できると予測される。

ＡＰＡＲＣはＦＯＲＴＡＲＮで記述されたプログラムを入力としてＡＤＥＮＴＲＡＮプログラムを出力する並列化コンパイラである。ＡＰＡＲＣを用いて並列化したプログラムは、初めからＡＤＥＴＲＡＮを用いて”陽”に並列記述したプログラムに比べ、並列処理部分の粒度が小さくなりがちである。このため、データ転送に伴うオーバーヘッドが大きいと、十分に満足のいく処理性能を得られないことになり、場合によっては並列化のメリットがないということもありうる。従って、ホスト計算機とＡＤＥＮＡＲＴ本体とのデータ転送に伴うオーバーヘッドを削減することは、ＡＰＡＲＣの適用範囲を拡大し、ＡＤＥＮＡＲＴの並列プログラム開発環境の向上に貢献する上で重要である。

6．おわりに

並列計算機ＡＤＥＮＡＲＴの配列データの分散配置の方式とこれを効率良く行なうためのハードウェア機構について述べた。本ハードウェア機構によって、ホスト計算機とＡＤＥＮＡＲＴとの間でのデータ転送時におけるオーバーヘッドを削減し、ＡＰＡＲＣの適用範囲を拡大するとともに、その負担を下げることができる。システムにインプリメントした上で、各種のアプリケーションを用いて評価していく予定である。

参考文献

(1)T.Nogi: Parallel computation, patterns and wavesqualitative analysis of nonlinear differen- tial equations, pp279-318(1986)

(2)谷川他：並列計算機ＡＤＥＮＡ，情　報学会計算機アーキテクチャ研究会　報告，CPSY88-11, pp33-40(1989)

(3)若谷他：並列処理言語ＡＤＥＴＲＡ　Ｎの実装，情報学会ソフトウェア基　礎論プログラミング言語合同研究会　報告，pp1-10(1990)

(4)材木他：並列計算機ＡＤＥＮＡＲＴ　用自動並列化コンパイラ：ＡＰＡ　ＲＣ，並列処理シンポジウムＪＳＰ　Ｐ’９１，pp293-300(1991)

(5)加藤他：高並列計算機ＣＡＰ－ＩＩ　のブロードキャストネットワーク，　情報学会計算機アーキテクチャ報告　83-39, pp229-234, 1991

(6)森他：並列計算機におけるＰＥへの　データ転送制御方式，電子通信学会　春期全国大会No. 6, D-318(1989)