

疎行列専用並列計算機に関する研究

天野 英晴

慶應義塾大学理工学部電気工学科 助手

1. (SM)²プロジェクトの概観

膨大な計算量が要求される科学技術計算には、自然界に実際に存在する系の解析問題が多い。系の大きさが大きくなればなるほど同時に計算できる仕事が増えるため、このような問題は本質的に大規模な並列計算機によく適合する特性を持っている。ただし、対象とする系の性質によって、適合する並列計算機の構成は変わってくる。

流体力学や半導体デバイスのシミュレーション等の問題は、普通は規則的なメッシュで離散化される。これらの問題では、近接作用によって計算が進むことが多く、並列計算機の各プロセッサは周囲とだけ交信すればよいから、PAXに代表されるような正方格子状の構成が向いている。しかし、系の解析問題の中には規則的なメッシュで離散化されない問題も数多く存在する。例えば、構造解析等有限要素法で離散化を行なう問題ではメッシュは不規則になるのが普通である。また、電子回路解析や電力潮流解析の問題では系は不規則になるばかりか、近接作用だけでなく遠方の要素も影響を及ぼすため、正方格子状の構成は適さなくなる。

一般疎行列専用並列計算機(SM)²:(Sparse Matrix Solving Machine)プロジェクトは、このような不規則な構成を持つ系の解析を対象とする大規模並列計算機を開発する目的で1983年に慶應大学において開始された。この名前は、規則的なメッシュで離散化される問題が行列で表現すると帯行列になるのに対し、不規則な系は、大部分の零要素の中に非零要素が散在する一般疎行列になることに由来する。

(SM)²プロジェクトでは問題中に存在する膨大な並列性を生かすため、32bit程度のマイクロプロセッサを非常に多数、具体的には数千から数万台結合する。また、不規則な系の解析問題では系の部分によって仕事異なることが多いから、各プロセッサが独自にプログラムを実行するMIMD(Multiple Instruction Multiple Datastream)方式を用いる。

このような大規模なマルチプロセッサシステムを実現するためには次に示すように様々な問題がある。

●問題の記述

数千、数万といった多数のプロセッサを用いる並列性の高いプログラムをいかに記述するか。

●マッピング, プロセスコントロール

プロセスをどのようにプロセッサに割り付け、また実行時に制御するか。

- モニタ、デバッグ

並列に実行されているプロセスをどうやってモニタし、またデバッグするか。

- プロセッサの結合

どのように多数のプロセッサをデータ交換による効率低下を起こすことがないように結合するか。

(SM)²プロジェクトの大きな特徴は、NCモデルという簡単な計算モデルを提案し、これに基づいて全ての領域に対して一貫した設計を行なっている点である。このためにプロジェクトは言語、コンパイラ、デバッガ、OS、アーキテクチャ等の広い分野にわたっている。

まず、1983年にアーキテクチャの基本構想を提案し、1985年には改良版の(SM)²-11を提案した。

同年に言語、1986年から87年にかけて一種のOSの機能を持つ言語処理系と、マッピング法を提案した。この時期に20プロセッサから成るプロトタイプ1号機が稼働を始めた。回路シミュレータ、常微分方程式解析システムをはじめとしたアプリケーションの開発も同時に行い、これらに基づき、アーキテクチャ、言語についての評価を行なった。現在、研究の焦点はOSを一部ハードウェア化した新しいプロセッサアーキテクチャと、あらかじめプロセスの粒度を調整するプリプロセッサに移っている。以下本プロジェクトの代表的な成果を順に報告する。

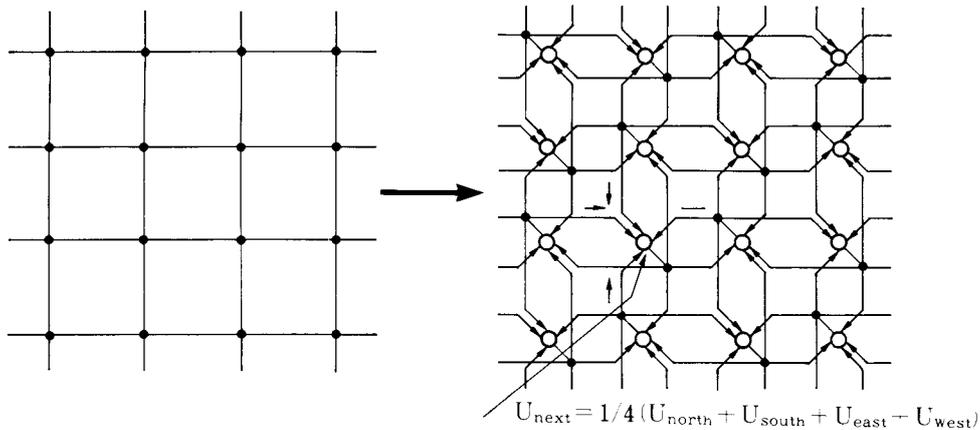
2. 問題の記述モデルおよび言語

従来、科学技術計算用の並列計算機の問題の記述にはFORTRANが用いられ、その拡張や並列化が研究されている。これに対して本プロジェクトでは、平行プロセス記述言語の考え方を導入し、記述モデルであるNC(Node & Connecting line)モデルを提案した。NCモデルはプロジェクトの対象とする問題の構造から直接導かれた非常に簡単な計算モデルである。このモデルでは系は能動的な働きをする節点と、それらを結合する結合線から構成される。図1はポアソン方程式を差分法で離散化した例である。この場合、格子点が節点にあたり、メッシュの辺が結合線にあたる。各格子点は上下左右の点からデータを受け取り、その平均を計算して出力することを繰り返す。境界に相当する節点のみ境界値を出力する。

このモデルは従来のようにシステム記述ではなく科学技術計算用に特殊化されており、CPSをはじめとする一般的な並行プロセスモデルに比べると、大きな違いが2つある。

- 1) NCモデルは節点、結合線共に静的で計算中、生成したり、消去したりできない。このため、このモデルは記述性が低く、プロセス間交信同期の表現力は自由選択ペトリネットという一般的なペトリネットのサブクラス程度でしかない。しかしそのかわりに単純であり、解析性が高い。

(2) 同期とデータ転送機構にある。NCモデルにおいては結合線の入力ポートは、サイズ1のバッファを持っており、データは一度に複数のポートに対し同時に転送される。このような1対多通信のことをマルチキャストという。マルチキャストでは、対応する入力ポートのバッファが全て空の時のみ、データの転送が可能である。もし、受信側がまだ処理を行っていないためバッファのどれかにデータが存在する場合、送信側は待ち状態になる。一方、受信側の節点は、データ受信時にバッファが空であれば待ち状態になる。



各節点は上下左右の節点から値を受け取り平均を計算し、出力する。

図1 ポアソン方程式のNCモデル表現

一対多のデータ転送は、科学技術計算では非常に頻繁に行なわれる。マルチキャストはCSPにおけるランデヴに比べ同期操作の厳密性に欠けるが、データが一度に複数プロセスに転送されるだけでなく、プロセスの不必要な待ち合わせが減り効率がよい。

さらに、我々はC言語をNCモデルの記述用に拡張した言語であるNCC (Node oriented Concurrent C) を提案し実装した。この言語は同様の処理を行なう複数のプロセスを容易に記述できるように、節点の行なう処理をテンプレートとして定義する部分 (プロセス記述部)、テンプレートから節点を生成する部分 (プロセス宣言部)、発生した節点を結合する部分 (ネットワーク記述部) の3つの部分によりNCモデルを表現する。データのマルチキャストは関数 `send()`、`receive()` を用いて行い、データの到着に関する非決定的な処理を記述するため `doeach` 構文と `doalt` 構文が用意されている。

NCCプログラム中の3つの部分はNCCブラウザを用いてウインドウシステム上で管理することができる。このウインドウシステム上ではネットワーク記述部を図的に表示、結合する機能も開発中である。

NCCはNCモデルに基づいているためOCCAMやAda等の並行プロセス記述言語に比べ、

合テーブルからは受信データメモリのローカルアドレスが得られ、そのアドレスに従って受信データメモリにデータが書き込まれる。ローカルアドレスは同時にキューの中に入れられ、プロセッサに対しデータの到着が通知される。ハンドシェイクフラグはデータの上書きを防ぐもので、データの書き込みの際にチェックされ、受信データメモリの内容がまだCPUによって読み出されていないならば送信側のプロセッサ及び他の受信中のプロセッサに転送の失敗が通知される。

RSMの特長は次の3点にまとめられる。

- (1) 各プロセッサは自分が必要としないデータの書き込み時には受信データメモリからデータを読み出すことができる。すなわち、通常の共有メモリと比較すると読みだし同士の競合がないだけでなく、書き込みと読みだしの競合が最小である。
- (2) 同期はデータ交換と同時にハンドシェイクフラグにより自動的に行なわれ、共有バスを余分に用いることが無い。
- (3) 受信データメモリはそのプロセッサにとって必要なデータのみを格納すればよい。問題の疎行列性が十分大きければメモリ容量は小さくて済む。

RSMはバスの能力を最大に引き出すことができるため、通常の共有メモリを持つバス結合に比べてはるかに多くのプロセッサを結合することができる。しかし、それでも単一バスに結合できるプロセッサ数は32程度が限界であり、数千数万のプロセッサを結合するためにはクラスタ構造を導入する必要がある。

RSMを用いるとプロセッサ間の結合とクラスタ間の結合はほぼ同様に行なうことができる。クラスタ間の結合はエージェントと呼ばれる一種の選択機構により行なわれる。エージェントはRSM-Managerを2組内蔵しており、双方向にデータを転送する。あるクラスタのバス上のデータがクラスタ外に送られる必要がある場合、対応するエージェントがこれを受け取り、アドレス変換を行い、接続されている他のクラスタのバス上にデータを送る。エージェントはクラスタ同士を直接結合することも、バスを介して結合することも可能であり、様々なクラスタ間結合方式が実現できる。図4はこのうちの一例である。

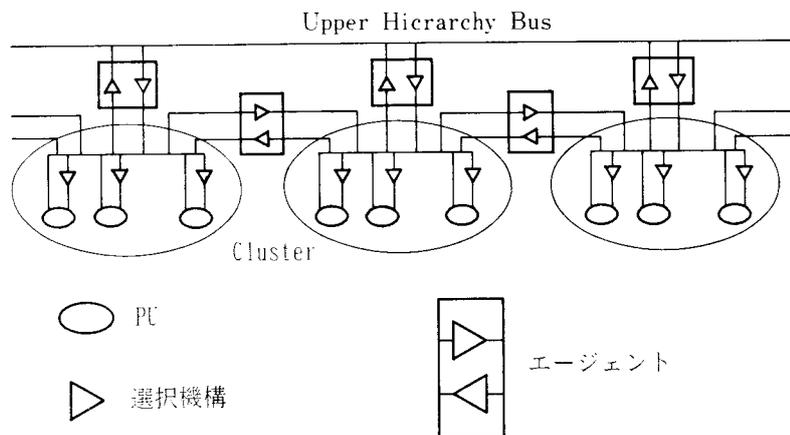


図4 クラスタ結合方式の一例

RSMを用いたクラスタ構成は、現在、商用化されているハイパーキューブマシンを代表とするリンク結合型の並列計算機に比べて、マルチキャストをサポートしている上に、データ中継と計算のオーバーラップが効率よく行なわれる点でも有利である。また、最近米国を中心に開発されている Butterfly, RP3等の多段結合網を用いた構成と比べても、交信の局所性が生かされるため、性能価格比の点で有利である。

4. プロセッサアーキテクチャ

ユーザがNCCで記述するプロセス数は、対象とする系が大きいほど多くなるため、1つのプロセッサは多数のプロセスを扱わなければならない。RSMによって、プロセッサ間の交信のオーバーヘッドが小さくなると、今度はプロセッサ内でのプロセス間交信及びプロセス管理のオーバーヘッドが効率低下の大きな原因として浮かび上がってくる。

(SM)²ではこのオーバーヘッドを小さくするため、プロセス管理用のコプロセッサを付加する。(SM)²のプロセッサアーキテクチャ Impulseは、汎用プロセッサであるタスクエンジンにコプロセッサであるIPC (Inter Process Communication) エンジンと、FPP (Floating Point Processing) エンジンが付加される構成を持つ。IPCエンジンはプロセッサ内のプロセス管理とプロセス間交信を行い、FPPエンジンは浮動小数点演算を高速に実行する。

ユーザの記述したプロセスは、タスクエンジン内でプロセス間交信が起きるまで連続して走り続ける。プロセス間交信が必要になった場合、IPCエンジンはプロセス間交信要求を受け付けるとともに、そのプロセス間交信がブロックし、プロセス切り替えが必要になるかどうかを高速に判定する。ブロックしない場合、タスクエンジンはそのまま走り続け、そうでなければIPCエンジンから次に実行するプロセスを受け取る。

IPCエンジンはRSMを管理して、他のプロセッサとデータ交信を行いながら次に実行できるプロセスを用意し、さらにプロセス内のデータのマルチキャストを高速に行なう。このために、IPCエンジンは独立したマイクロプログラム制御機構を持つ専用ハードウェアとなる。

FPPエンジンはスカラ計算を特に高速に行なう他、一般疎行列の処理で重要なアドレス計算をタスクエンジンと独立に高速に行える構成を持つ。

5. プロトタイプと現在の状況

20プロセッサから成るプロトタイプ1号機(図5)は1986年に稼働を開始した。このプロトタイプ1号機はNCCコンパイラ、プロセス管理の手法の確立、アプリケーションの開拓等の目的で作成されたため、規模も小さく性能も表1に示すようにさほど高くない。

このうち、RSM-Manager部は比較的高い機能を持っている。バスが16bitであることを除き、図3にほぼ等しい構成を持ち、SONY社の協力で開発したゲートアレイチップに

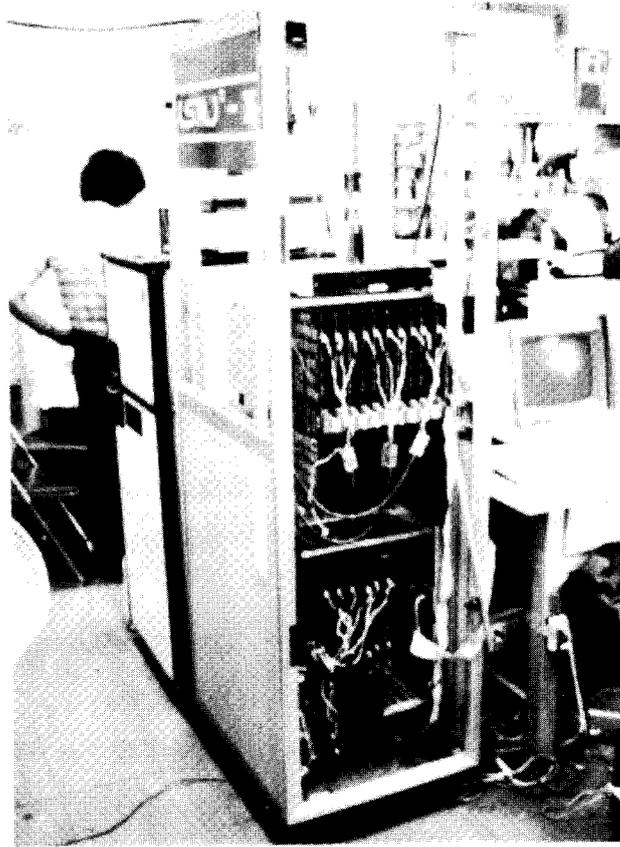


図5 プロトタイプ1号機

表1 プロトタイプ1号機の仕様

RSMの仕様

CPU部の仕様

Bit rate	40Mbit/sec
Local Address	64K by 8bit
メモリ要求量(Total)	288 kbit
Address Reference Table	128 kbit
Receive Data Memory	128 kbit
Arrival Detect Queue	32 kbit
デバイス	CMOSゲートアレイ コントローラ 50 TTL

CPU	MC68000
MMU	MC68451
Local Memory	2 Mbyte DRAM
FPP	MC68881 (0.15 MFLOPS)

より制御される。これに対しプロセッサ部は16bitのマイクロプロセッサ MC68000 が使用され、IPCエンジン、FPPエンジンは付加されていない。プロセス管理機構はソフトウェアで実装されている。

現在、以下に示すアプリケーションが稼働中である。

- 電子回路シミュレータ ESIM

緩和法により、バイポーラ素子の回路シミュレーションを行なう、SPICEライクの記述を用いる。今のところ、対象とする素子はバイポーラに限られ、MOS-EETモデルを組み込み中である。

- 常微分方程式、連立方程式解析システム ODESSA

式の形で入力した常微分方程式と連立方程式を反復法により解くシステム。プロセス融合の手法が用いられており、自動的にプロセッサ台数分に融合されたNCCプログラムが出力され、実行される。

そのほか、ニューラルネットワークのシミュレータ、OPS 5 を用いて記述されたエキスパートシステム、レイトレーシングシステム等の実装も行なわれている。

プロトタイプ1号機はプロセス管理機構がソフトウェアで実装されているため、プロセスの実行時間が全計算機時間の50%を越えることは難しい。さらに、浮動小数点演算装置があまり高速でないため、回路シミュレータで、VAX-11/750の約1.5倍程度の性能である。

現在IPCエンジン、FPPエンジンを組み込んだプロトタイプ2号機を開発中である。

6. おわりに

高柳先生及び財団から頂いた助成金は図5のプロトタイプの開発費として用いさせていただいた。現在大学レベルでこのような試作機を実際に制作できるのは資金面と技術面でやや困難になっている。このような機会が与えられたことを本当に感謝している。

プロジェクトの成果は国際学会を中心に様々な場で発表し報告した。代表的な論文を下記に示す。

- 1) H. Amano, T. Boku, T. Kudoh

“(SM)²: The Sparse Matrix Solving Machine,”

To be appeared in the IEEE Trans. on Computers.

- 2) H. Amano, T. Boku, T. Kudoh, H. Aiso

“(SM)²-11: The new version of the Sparse Matrix Solving Machine,”

Proc. of the 12th Annual International Symposium on Computer Architecture,

Jun. 1985.

3) T. Kudoh, H. Amano, T. Boku, H. Aiso

“NDL : A language for solving scientific problem on MIMD machines,”

Proc. of the 1st Super Computing Conference, Dec. 1985.

4) T. Boku, S. Nomura, H. Amano

“Impulse : A High Performance Processing Unit for Scientific Calculation”

Proc. of the 15th Annual International Symposium on Computer Architecture,

Jun. 1988.