

並列オブジェクト指向トータルアーキテクチャA-NET

2L-3

- マルチコンピュータ開発と言語実装の現状 -

吉永 努

馬場 敬信

宇都宮大学 工学部*

1. はじめに

A-NETは、並列オブジェクト指向を核概念として、プログラミング言語、ハードウェアアーキテクチャ、および各ノード上のローカルOSの設計を統合的に進めるトータルアーキテクチャである[1]。本研究において、我々は、柔軟な同期機構やネットワークポロジに適合した負荷分散などに特徴を持つ並列オブジェクト指向言語A-NETLとその効率的実行を目的としたマルチコンピュータを開発した。

本稿では、マルチコンピュータ開発と言語実装の現状について報告する。

2. A-NET マルチコンピュータ

A-NET マルチコンピュータは、独自に設計したノードプロセッサを種々の直接結合網で結合可能な分散メモリ型並列計算機である。図1に、16ノードプロトタイプの写真を示す。きょう体サイズは、100cm(幅)×85cm(高さ)×104cm(奥行き)である。

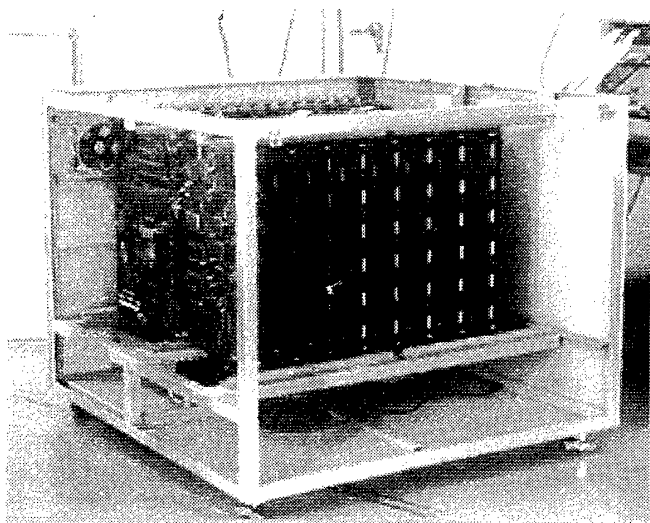


図1: A-NET マルチコンピュータ

ノードプロセッサは、メソッドを実行するPEとメッセージ通信を行うルータの2枚の基板で構成される。基板サイズは、共に500mm×375mmで、内層をGND,VCCとする4層基板である。で直接結合される。配線は、2.54mm 間隔に3本のルールで行った。

*A Parallel Object-Oriented Total Architecture A-NET -Current status of multicomputer development and language implementation-, Tsutomu YOSHINAGA and Takanobu BABA, Utsunomiya University.

2.1 PE

PEは、32ビットALU, FPU, 8ビットTPU (Tag Processing Unit), 280KBの局所メモリなどで構成され、オブジェクトを単位として割り付けてメッセージ駆動でプログラムを実行する。TPUは、本ノードプロセッサの特徴の一つであり、ALU, FPUの計算と同時にデータ型やFutureフラグの検査を行う。メモリは、1語40ビットで構成され、データはA-NETLの未来型メッセージと動的データ型付けなどを支援するための8ビットのタグを持つ。その他、可変長機械命令のフェッチとデコードを支援する命令前処理ユニット、コンテキスト切替えを高速化するための2セットの特殊レジスタセットなどに特徴がある。

プロトタイプPEに対しては、試作を容易にするためにマイクロプログラム制御方式を採用しており、76×8KWの制御記憶を持つ。システムクロック30MHz, 1マシンサイクル167nsで動作する。

2.2 ルータ

ルータは、6つの隣接ノード結合用ポートと1つのホスト用ポート、およびメッセージ送受信ユニットなどからなる。プログラマブル・コントローラにより、任意のネットワークポロジによる結合が可能である。各ポートとメッセージ送受信ユニットは、8ビットのデータバスを持つクロスバスイッチによって接続されるため、ルータ内でも同時に複数のメッセージ転送が可能である。ノード間結合リンクは、データ8ビット、制御信号4ビットの双方向リンクであり、データ転送速度は15MB/s/ポートである。メッセージ転送のフロー制御には、バーチャルカットスルー方式を採用しており、衝突パケットは2KBのFIFOを持つパケットバッファに一時退避される。経路選択は、宛先ノード番号に対して、複数の出力ポートを割り当てることが可能であり、プログラムされた出力候補から、その時に空いている経路を適応的に選択できる。プロトタイプで転送時間を実測した結果、35バイトを隣接転送するのに約5.2μsかかり、転送距離が伸びる毎に約1.0μsの遅延が発生することが分かった。

2.3 マルチコンピュータの現状

現在、ハードウェアの調整がほぼ完了し、A-NETLアプリケーションの実行テストとマイクロプログラムのチューニングを行っている。現時点では、N-QueensやライフゲームなどのA-NETLプログラムの動作を確認している。

3. 並列オブジェクト指向言語 A-NETL

A-NETL は、大規模並列プログラムの記述が容易に行え、且つ、マルチコンピュータ上で効率的な実行ができることを目的として設計された。その特徴を要約する。

(1) 多数オブジェクトの静的・動的生成

高並列計算においては、同じような性質を持った計算オブジェクトが多数必要なことが多い。このために、A-NETL ではクラスを介した動的なオブジェクト生成とインデックストオブジェクトを用いた静的定義機能を設けている。

動的生成に対しては、実行開始時にすべてのノードにクラスをブロードキャストしておき、ユーザプログラムで生成先のノード番号を指定できるようにしている。

インデックストオブジェクトは、定型な問題に対して動的生成の手数を減らすと共に、言語処理系による負荷分散を行うために設けている。

(2) メッセージ構文

メッセージパッシングとしては、ABCL/1 とほぼ同様な過去、現在、未来型の3つを定義できる [3]。過去、現在、未来のすべての型についてメッセージのマルチキャストが可能である。

ABCL/1 との違いは、未来型メッセージの実装において、未来オブジェクトに対して明示的に返値の利用可能性を問い合わせるのではなく、タグを用いた暗黙的な返値の到着確認をサポートする点、および、待ち合わせの際にオブジェクト内並列性を活用する点にある。

(3) 同期機構

メソッドは、通常単一のメッセージによって起動されるが、複数のメッセージを待ち合わせるメソッドも定義可能である。複数メッセージによって起動されたメソッドからは、複数のセンダに対して同一の、あるいは異なる返答をすることができる。

また、メソッドの実行順序指定や二重起動の禁止、条件付きメソッド起動などが定義できる [2]。

4. A-NETL 実装

A-NETL の実行環境としては、現在、A-NET マルチコンピュータとワークステーションクラス上のシミュレータ、富士通 AP1000 が利用可能である [4]。この他、広く世の中に公開することと機種独立の並列オブジェクト指向言語の高性能実装方式の検討などを目的として、PVM や MPI などのメッセージパッシング・ライブラリを利用した実装を開始している。

これらの環境に対して、A-NETL ユーザに統一的なプログラミング環境を提供するために、図2に示すように X ウィンドウ・インタフェースを用いてプログラムの編集からコンパイルまでを一貫して行えるプログラミング支援システム APSS、並列デバッガ、ネットワークポロジ独立なアロケータなどを整備している。

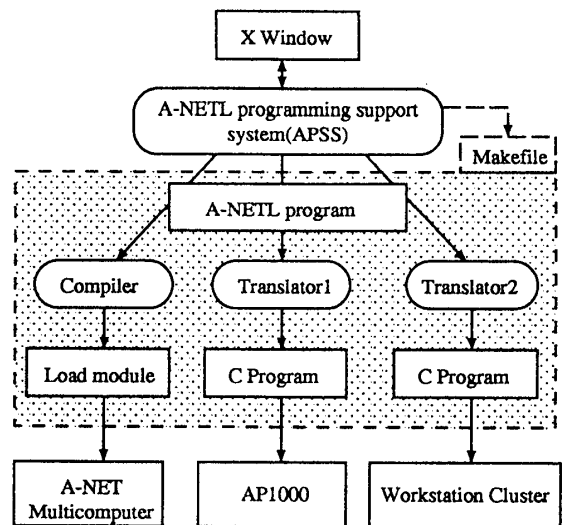


図2: A-NETL プログラミング・実行環境

5. おわりに

A-NET マルチコンピュータ開発と言語実装の現状について述べた。現在、評価データ収集のためのエバリユエータをマイクロプログラムとローカル OS レベルで実装しているところであり、詳細な評価は今後の課題である。

謝辞：ハードウェアの試作に関して御協力頂いた日本電気 (株) C&C システム研究所、(株) 東芝研究開発センタ、(株) 日立製作所日立研究所、(株) 大晶電子、富士精密 (株)、日本 AMD (株) の各社に感謝する。

本研究は、一部、文部省科学研究費 (試験 (B) 04555077, 重点「超並列処理」04235104, 一般 (C) 07680334, 奨励 06780233)、電気通信普及財団の補助による。

参考文献

- [1] 馬場, 吉永: “並列オブジェクト指向トータルアーキテクチャ A-NET における言語とアーキテクチャの統合”, 信学論 (D-I), J75-D-I, 8, pp.563-574(1992).
- [2] T. Baba, N. Saitoh, T. Furuta, H. Taguchi and T. Yoshinaga: “A Declarative Synchronization Mechanism for Parallel Object-Oriented Computation”, IEICE Trans., E78-D, 8, pp.969-981 (1995).
- [3] A. Yonezawa: “ABCL: An Object-Oriented Concurrent System”, The MIT Press, Cambridge, Mass. (1990).
- [4] 岡嶋, 吉永, 馬場: “並列オブジェクト指向 A-NETL の高並列計算機 AP1000 への実装”, 文部省重点領域研究 (超並列) 第5回シンポジウム予稿集, pp.55-60 (1994).