

5U-4

並列AIマシンProdigyの

視覚的デバッガ(VDB)

小柳 滋 藤田 純一 鈴岡 節

(株)東芝総合研究所

1. はじめに

我々は、知識情報処理の高速化を目指して高並列マシン prodigy の開発を行っており、その中で並列処理ソフトウェアの開発環境の整備に取り組んでいる。我々はその一環として高並列処理用視覚的デバッガ(VDB)を開発した。本稿では、高並列システム用デバッガとしてのVDBの設計方針およびその概要について述べる。

2. 背景

我々が対象とするプロダクションシステムや意味ネットワーク等の応用では、問題自体がネットワーク構造として表現される。この問題のもつネットワーク構造を、PEのネットワーク上にマッピングして並列処理する方式を想定している。すなわち、ネットワークのノードをPEに分散配置し、ネットワークのリンクをPE間のメッセージ通信により並列に辿る。このようなシステムを実現するには、各PEは同一プログラムを非同期に実行し、PE毎に異なるデータを割当てる並列処理方式が適している。そのため、本システムにおけるソフトウェアは図1に示すような構成をとる。ユーザ・プログラムはホスト上で動作するプログラム(ホストプログラム)とPE上で動作するプログラム(PE

プログラム)の2種類がある。各PEは同一のPEプログラムを実行する。また、これらを制御するOSはホスト上で動作するホストOSと、PE上で動作するPEカーネルの2種類がある。PEプログラム間、およびPEプログラム-ホストプログラム間の通信はホストOSとPEカーネルの通信機能を介して行われる。

このようなシステムは、ノードをオブジェクトとみなしたオブジェクト指向システムとして捉えることができる。本OSでは、オブジェクト間通信機構をサポートしている。すなわち、オブジェクトの物理的な配置に対するトランスペアレンシを実現している。これにより、ユーザプログラムではPEの台数やノードの割当方法に対して独立な記述が可能となる。

3. 設計方針

我々はprodigyのデバッガの開発に当たって次のような方針に基づいて検討を行った。

①ボトムアップ方式:

低レベルデバッグと高レベルデバッグの2つのレベルを想定し、各々別個のツールを開発する。すなわち、低レベルデバッグでは、個々のプログラムモジュールのテストを行い、高レベルデバッグでは、システム全体の動きを通信履歴をベースとしてデバッグする。低レベルデバッグとしては、ホストマシン上のマルチプロセスとしてPEプログラムを実行する環境を設定し、ホストマシンの既存のデバッガを用いる方式をとった。また高レベルデバッグとしてVDBを開発した。すなわち、VDBでは、基本部分のデバッグが終了したプログラムの全体の動作を確認するためのデバッグツールである。

②オブジェクト指向の導入:

システム全体の挙動をオブジェクト間のメッセージ通信として抽象化することによりVDBでは、高レベルのデバッグ機能を実現する。

③視覚化:

システムの挙動を時系列に沿ったネットワークとして2次元的に表現し、さらにネットワーク内の特定のノードの詳細情報の表示や、ネットワークの一部分の選択表示などの機能を備え、ビジュアルなイメージでデバッグを行えることを目指している。

4. VDBの構成

VDBはprodigyとはオフラインに動作し、prodigyの実行時にオブジェクト間のメッセージの送受信及びインスタンス変数の変更をイベントとした履歴を取り、この履歴に基づいてシステムの挙動を2次元的に表示する。

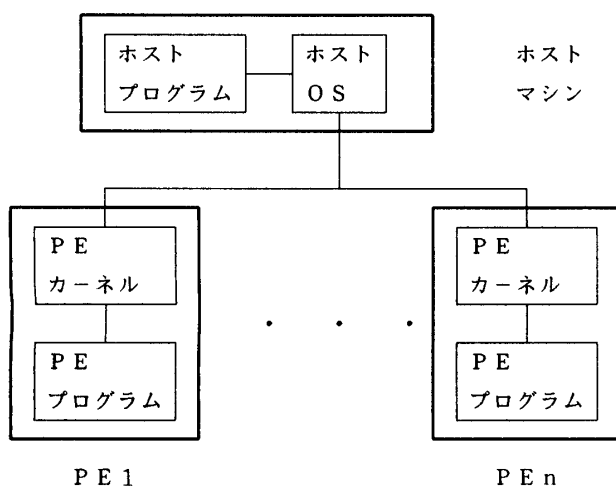


図1. ソフトウェア構成

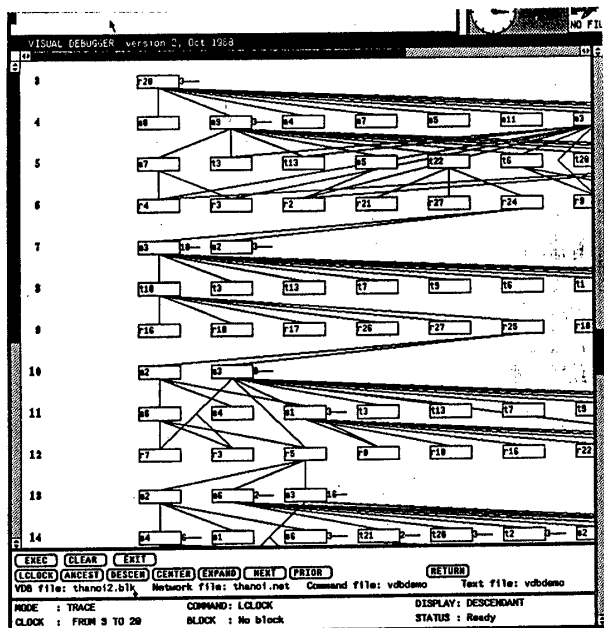


図2. VDBの表示例

また、このイベントを順序づけるために、論理時刻方式を用いる。すなわち、メッセージを受信したオブジェクトの論理時刻は、メッセージを送信したオブジェクトの論理時刻より大きい値を取ることで、イベントを順序づける。VDBにおいては、同一論理時刻をもつオブジェクトは論理的並列性をもつものと解釈し、物理的なPEの台数やノードの割当て方法とは独立に、オブジェクト間の論理的な依存関係のみに基づいてデバッグを行うことができる。履歴情報はメッセージの送受信及びインスタンス変数の変更などのイベントが発生した時点でPEカーネルによりとられる。これらの履歴情報は、PEカーネルによりバッファリングされ、バッファが一杯になった時点でホストに送信される。ホストOSはこの履歴メッセージを履歴ファイルに格納する。

5. VDB の機能

VDB の主な機能を以下に示す。

1) トレース

トレース機能は論理時刻に沿ってシステム全体の挙動を表示するものである。具体的には、縦軸に論理時刻をとり、その時刻に起動されたオブジェクトを水平方向に並べ、オブジェクト間のメッセージ通信を線分で表示する。表示例を図2に示す。なお、大規模なシステムにおいては、トレース機能を用いると同一論理時刻で多数のノードが起動される可能性があるため、これらを1画面で表示することはできなくなる。そこで、指定されたノードと関連のあるノードのみを選択してメッセージ・フロー図を1画面で表示できるような機能が必要である。その

ためVDB では次の3種類のコマンドを備えている。

• ancestor :

指定されたノードを終点とし、そのノードに到達するメッセージのフロー図を選択表示する。

• descendent :

指定されたノードを起点とし、そのノードから派生するメッセージのフロー図を選択表示する。

• center :

上記ancestorとdescendentの混合であり、指定されたノードを中心とするメッセージフロー図を選択表示する。

(2) ステップ

ステップでは、ユーザは表示したいオブジェクトについてその表示位置をマウスを用いて画面上に指定する。VDBは現在の論理時刻で行われたメッセージ通信の中でオブジェクトの表示位置が定義されているならば、そのメッセージ通信を対応するオブジェクト間を結ぶ線分で表示する。また、論理時刻の前進、後進を示すコマンドを用いることにより、論理時刻に沿って関連するオブジェクト間のメッセージ通信の流れを次々と表示することができる。

(3) 検索

検索は全メッセージフローの中で特定の検索条件を満たすオブジェクトを直接求める機能である。

検索対象としては、オブジェクトとメッセージパターンの2種類をそなえている。オブジェクトの検索では、オブジェクト名、受信メッセージ、送信メッセージ、インスタンス変数等の属性を指定して該当するオブジェクトを検索する。また、メッセージパターンの検索では、メッセージの流れを論理式で表現し、該当するパターンを検索する。

6. おわりに

高並列システムの視覚的デバッグVDB について述べた。VDB については現在prodigy 上でのプロトタイプの開発が終了し、実用評価をおこなっている。

参考文献

- 小柳他 “超並列AIマシンの構想”, データフローワークショップ1987, (1987.10)