

並列オブジェクト指向システムORAGA

-OragaItSelf のハードウェアシミュレータ

6F-3

河野真治・渡部真幸・神田陽治・田中英彦
東京大学工学部

1.OragaItSelf

OragaItSelf は、ソフトウェア支援システムOraga を支えるアーキテクチャである。OragaItSelf は、並列オブジェクト指向言語DinnerBellを、コンテキストと称する実行単位により並列実行する。コンテキストは、メッセージを、その引数単位に分解したものである。

DinnerBellの並列性は、並列に実行されるコンテキスト間の関係の程度により、二つ分類することができる。一つは、コンテキストの関係が親密な場合である。例えば、ベクトルを表すオブジェクトにメッセージが送信された場合、生成されるコンテキストは、ベクトルのデータ構造にしたがった関係をもっている。

もう一つは、コンテキストの関係が疎な場合である。後者の並列性は、ゆるく結合した多数のPE。(Processor Element) により実現することができる。この時に、問題となるのは、コンテキストを均等に各PEに分配することである。

この分配の仕方は、実際のプログラミングや、PEの結合状況によってダイナミックに変える必要がある。この分散の戦略を現実的な環境で試験する為にOragaItSelf のハードウェアシミュレータを設計した。以下では、モデルとするアーキテクチャと、そのシミュレータのハードウェア及びソフトウェア構成を述べる。

2.DinnerBellを直接実行するアーキテクチャ

コンテキストは、Destination,Reply,ReplySelf と一引数のメッセージを構成するKey とArgumentの五つの要素を持つ。複数のコンテキストにより一つの複数引数をもつメッセージ送受を構成する。これにより、単一のメッセージ送信をより基本的な単位に分割してオーバーラップしながら実行することが可能となる。一つのコンテキストの発火は、複数のコンテキストを生成する。OragaItSelf は、このコンテキストを変換していく実体である。

コンテキストが、OragaItSelf の実行単位であり、並列実行は、コンテキストをPEに分配することにより実現される。コンテキストの要素は、オブジェクトを指し示す単一代入変数であり、コンテキストの発火は、この単一代入変数への代入に支配される。

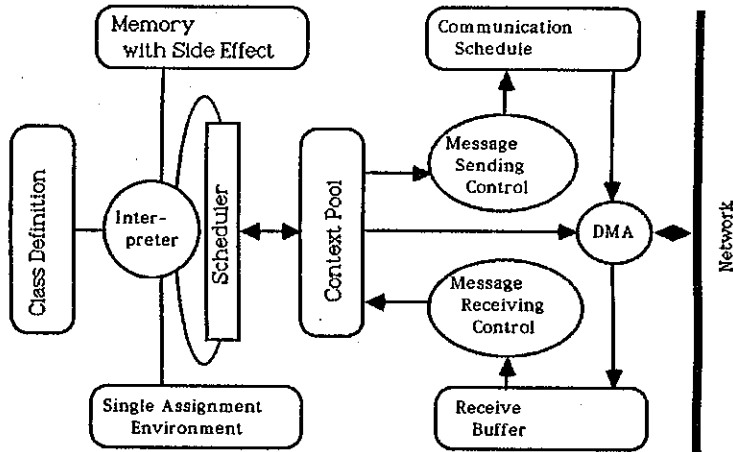


図2 シミュレータのプロセス

OragaItSelf の一つのPEは、オブジェクトを構成するメソッドと、オブジェクトの環境、コンテキストのスケジューラ、副作用を持つオブジェクトの作業領域の四つのメモリと、コンテキストの変換器からなる。

3.ハードウェアシミュレータの構成

OragaItSelf のハードウェアシミュレータは、1MB のメモリと68000 のワンボードコンピュータ、ネットワークインタフェースを一つのPEとしている(図1)。このPE、三台を結合する。このうちの一台は、ホストをかねる。

各PE内部は、VME バスにより結合されている。PE間には、共有メモリはない。PE間の通信は、16bit 幅のネットワークにのみによりおこなわれる。ネットワークは、68000 のアクセスに比べて十分に速く(最大6Mbyte/sec) 現在の構成でネットワーク自身がネックになることはないと考えている。

各PE上には、汎用マルチタスクOSであるOS-9が存在し、シミュレータプロセスの管理をおこなう。ダウンロード及び、シミュレータプロセスの監視は、ネットワークとは別のRS232C回線によりおこなう。

4. シミュレータのプロセス

一つのPE上に、四つのプロセスが存在する (図2)。

・インタプリタ

コンテキストの変換を逐次実行する。スケジューラより発火可能なコンテキストを取り出し、メッセージの受信を実行する。結果として生成したコンテキストをスケジューラに挿入する。

・通信管理 (2 プロセス)

PE間の通信を管理する。受信と送信の二つのプロセスよりなる。インタプリタと非同期的に通信しながら、受信したコンテキストと単一代入変数への代入、送信すべきコンテキストの選択などをおこなう。

・DMA

通信管理プロセスにより設定されたスケジュールにより、実際の通信をおこなう。他のPEへの転送は、コンテキストのキューから直接転送する。これに対し受信は、一旦、バッファに格納し、受信通信管理プロセスがこれを解釈する。

DMA のプロセスは、ハードウェアにより実現されるが、他のプロセスは、ソフトウェアで実現する。

5. ネットワーク上のデータ

ネットワークを通じて、以下の三つの要素が通信される。

・コンテキスト

コンテキストを他のPEに分配する事により、並列実行が実現される。

・Reply

単一代入変数への代入は、必要なPEへ転送される。これは、コンテキストの送信に対する答となる。

・Control

通信の協調をとる為に、制御用のメッセージを送る。コンテキスト転送のリクエスト、許可に用いられる。

オブジェクトのインスタンスの転送は行わない。

コンテキストおよびReply 内部のデータは、オブジェクトへのポインタと、それ自身転送してよいプリミティブオブジェクト (整数、浮動小数、ストリング、環状でないクラス) である。

6. コンテキスト分配の方針

・PE間にまたがるポインタ

通常メッセージは、オブジェクトのいるPEに対するコンテキスト転送で実現される。コンテキスト内Reply とReplySelf は、必ず変数である。また、keyは必ず定数である。したがってDestination と、Argumentにより、コンテキストの性質がおおよそ決定される。destination の確定していないコンテキストが、外のPEに転送されることはない。Argumentが確定していない場合は、Argumentがプリミティブの値が確定することがある時のみ外に転送することが望ましい。要約するとまたがるポインタをすくなくするようにコンテキストを転送する。

・インスタンスの分配

このような分散環境では、オブジェクトの間で、並列実行を引き出すためには、オブジェクト、特異インスタンスを均等に各PEに分配することが重要で基本的に、オブジェクト生成のメッセージは、自分に拡大するように、送信される。このシステムではインスタンスの転送を行わないので、オブジェクトの生成時に確定してしまう。したがって、オブジェクトの生成メッセージを含むコンテキストの転送先の決定が重要となる。この決定は、コンテキストを見るだけでは、判断できずDinnerBell自身に、させることも考えている。

