複数プラットフォーム上での受信メッセージ予測法の評価

3F−4

岩本善行　大津金光　吉永努　馬場敬信
宇都宮大学工学部

1.はじめに

我々の研究室では，これまでに商用並列計算機として富士通AP1000とNEC Cenju-3を取り上げ，それぞれの並列ライブラリ，OSのソースプログラムの提供を受け，メッセージ送受信処理部分の評価／分析を進めてきた．同時に，並列オブジェクト指向トータルアーキテクチャA-NETに関する研究を行い，この過程でメッセージ転送処理においては，受信側での受信処理に時間が多くかかること[1]や，アイドル時間を有効利用する可能性のあることが明らかになってきた．

これらを背景として，我々は，これから到着するであろうメッセージをアイドル時間を利用して予測し，その結果に基づいて受信後の処理を投機的に実行することによって，高速化を試みる受信メッセージ予測法を提案している[2]．本手法がハードウェアや並列化ライブラリなど，特定のプラットフォームに依存せず広範囲に応用が可能な技術であることを実証するため，AP1000やワークステーションクラスタ，A-NETマルチコンピュータに対して本手法を実装してきた．さらに，A-NETマルチコンピュータではさまざまなメッセージ予測アルゴリズムを実装し，ベンチマークプログラムを用いて評価してきた．

本稿では，受信メッセージ予測法について述べた後，その予測アルゴリズムを説明する．次に，各プラットフォーム上に対する実装について説明し，最後にA-NETマルチコンピュータ上で得られたベンチマークプログラムによる各予測方法での予測成功率，速度向上などの詳細な評価について議論する．

2．受信メッセージ予測法

本手法では，到着するメッセージの中からどの項目を予測するかということと，どのようなアルゴリズムで予測するかが非常に重要となる．以下では，今回実装したA-NETマルチコンピュータにおける予測項目と予測アルゴリズムについて簡単に述べる．

2.1 予測項目

A-NETマルチコンピュータにおけるメッセージパケットの中から実際に予測するのは，

1) メッセージサイズ
2) 過去／現在などのメッセージ型
3) メッセージの送信元
4) 起動すべきメソッド

の各項目とする．これ以外の項目は，前もって決定されている値や上記の項目から求めることができる内容である．また，メッセージの引数については今回は予測対象外とする．

1)のメッセージサイズを予測することによる効果としては，メッセージ到着前にあらかじめ格納領域を確保することが可能となることや，引数の数が計算可能となることなど，それぞれの項目について，その効果が挙げられる．

Evaluation of Receiving Message Prediction Method on Multi Platform
Yoshiyuki IWAMOTO, Kanemitsu OOTSU, Tsutomu YOSHINAGA and Takanobu BABA
Faculty of Engineering, Utsunomiya University

2.2 予測アルゴリズム

上記の予測項目から，具体的に考えられる予測アルゴリズムとしてここでは，

直前：直前に到着したメッセージのみを参照
最大値：これまでに到着したメッセージの最大値
平均値：これまでに到着したメッセージの平均値
最頻値：最も回数が多かったもの
マ連鎖：マルコフモデルに基づく連鎖

の5種類を考える．

3.複数プラットフォーム上での実現

今回実装したA-NETマルチコンピュータ/A-NETL，富士通AP1000/CellOS Lib，同AP1000/MPI(Message Passing Interface)，ワークステーションクラスタ/MPIのそれぞれに対する実装について説明する．

3.1 A-NETマルチコンピュータに対する実装

我々の研究室でプロトタイプとしてハードウェアを開発した並列オブジェクト指向計算機のA-NETマルチコンピュータは，処理を行うPE (Processing Element)とノード間通信を行うルータの16ノードの対によって構成される．PEはマイクロプログラムによって制御されている．これまでに，メッセージの送受信処理やオブジェクトの管理などを行うOSもマイクロプログラムにより実装している．そこで，受信メッセージ予測法をこのマイクロプログラム化したOSのメッセージ受信処理部に変更を加えることによって実装し，評価する．

3.2 AP1000に対する実装

AP1000は富士通によって開発された分散メモリ型並列計算機である．言語レベルでの並列化にはCell OS Libraryを使用し，ノード間通信をC言語（およびFortran）において実現している．このライブラリの中でノード間通信に一般的に使用される関数は，l_asend()とl_arecv()があり，ユーザプログラムにおいてl_arecv()が実行された時に，引数として指定したメッセージが到着してない場合に，アイドル時間となる．このアイドル時間を利用して，次に到着するメッセージを予測し，さらに受信処理とその後のユーザプログラムを投機的に先行実行することとする．よって，実装はl_arecv() を実現しているライブラリレベルに対して変更を加える．

3.3 MPIに対する実装

MPIは，現在，世界的に広く使用されている並列化ライブラリであり，Cenju-3やAP1000，ワークステーションクラスタなど，広範囲に実装され，活用されている．今回は，この中からAP1000およびワークステーションクラスタを選び，両プラットフォーム上のMPIに本手法を実装した．MPIの関数の中で，メッセージ送受信に一般的に使用される関数には，ブロッキングを行うMPI_Send()およびMPI_Recv()が挙げられるが，ワークステーションクラスタにおけるMPIの実装では，非ブロッキング型のMPI_IRecv()とメッセージが到着するまで待つMPI_Wait()の組み合わせによってMPI_Recv()が実現されており，今回はこのMPI_Wait()に変更を加えて，

本手法を実装する．

4.予測方法の定量的評価

ここでは，これまでに実装が完了し，評価が終了したA-NETマルチコンピュータについて説明する．先に述べた6種類の予測方法についてその特性を詳細に調べるために，A-NETマルチコンピュータ上に実装し，メッセージ通信パターンの異なる3種のベンチマークプログラムを用いて本手法を評価した．

4.1 評価項目

受信メッセージ予測法に対するメッセージ予測アルゴリズムについての評価項目は，1) 実装の容易さ，2) 予測に必要となる時間，3) ログを記録するために必要となるメモリ量，4) 予測成功率，5) 速度向上，6) 適用可能範囲の6項目とした．この中から本稿では，4)，5) について説明する．

4.2 予測成功率

表1に3種類のベンチマークプログラムを実行した時の，予測対象メッセージ数（予測対象数），ヘッダのみ予測成功（ヘッダのみ），引数を含めたメッセージ全体の予測成功（引数まで），予測失敗のそれぞれの回数を示す．このように予測の成功／失敗を段階的に切り分けることによって，可能な限り再実行のオーバヘッドを減らすことが可能となる．この表より，ヘッダのみの予測成功率は82.2～100%となり，性質の異なるアプリケーションにおいても比較的高い予測成功率となることが分かる．

さらに，表1の3種のベンチマークプログラムの中から，台形公式による定積分について，それぞれの予測アルゴリズム，予測項目と予測成功率の関係を図1に示す．

台形公式による積分は，数値演算に対するマスター－スレーブ型のリダクション処理を20回繰り返すプログラムである．スレーブ側のノードにおける受信処理は，すべてマスタ側から送られてくるメッセージであるため，どの項目についてもほぼ100%予測を成功させることができる．また，マスタ側ノードにおける受信処理は，スレーブとなる全ノードからほぼ同時にメッセージが送られてくるため，**直前／平均／最頻値**による予測は困難となる．

メッセージサイズの予測については，予測した値を用いてメッセージ領域を確保するため，実際に到着したメッセージサイズより大きな値として予測してあれば，領域の再確保を行う必要がないため，予測は成功とする．よって，サイズを**最大**で予測した場合の予測成功率は極めて高くなる．一方，**マルコフ連鎖**による予測成功率が他のアルゴリズムより高くなっているのは，マスタノード側で受信するメッセージが同時に多数のノードから送られてくる場合であっても，各スレーブで行われる処理やメッセージ転送距離の違いなどにより規則性が発生するためである．

4.3 速度向上

受信メッセージ予測法による速度向上の例として，先の台形公式による積分のベンチマークプログラムの実行ログを分析した．通常実行時には，ユーザプログラムが460マシンサイクル（以降MC，ただし1MC=167ns)動作した後サスペンドし，システムが17MC動作した後にアイドル状態となる．また，返値メッセージの到着後，OSよる受信処理に200MCを要している．予測実行時には，ユーザプログラムのオーバヘッドとして6MC(動作時間460MCの1.3%)増加している．ユーザプログラムがサスペンドした後，システムは合計381MC動作しているが，このうち，メッセージの予測と受信処理に要した時間は364MCであり，さらにその後通常ではアイドル状態となるところをユーザプログラムが動作しているためアイドル時間は存在しない．返値メッセージが到着した後，到着したメッセージと予測したメッセージとの比較やログの取得，受信処理などが行われるが，投機的先行実行による効果によって，合計159MCの動作でユーザプログラムに復帰可能となり，通常実行に比べて41MC (受信処理200MCの20.5%)の高速化となった．

表1　予測成功回数

	8-Queen	LifeGame	台形公式
予測対象数	82	296	297
ヘッダのみ	82	254	244
引数まで	0	0	0
予測失敗	0	42	53

（単位：回）

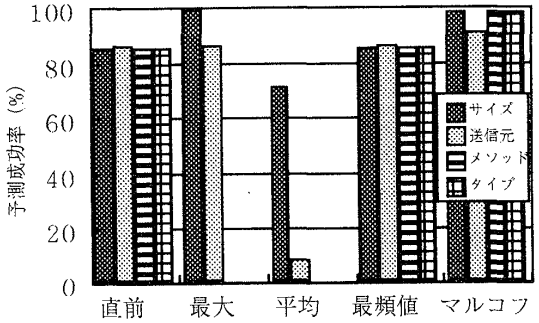

図1　台形公式による定積分における予測成功率

5.おわりに

本稿では，メッセージ転送処理を高速化するための手法である受信メッセージ予測法について述べ，メッセージの予測方式として6種類の手法を提案した．さらに，A-NETマルチコンピュータ，富士通AP1000，MPIライブラリに対する実装について述べた．最後にA-NETマルチコンピュータにおいて行った評価において，予測成功率が80%以上，速度向上が20.5%であるとの結果を得た．

謝辞

本研究は，一部文部省科学研究費（基盤(C)課題番号09680324，基盤(B)課題番号10558039，奨励(A)課題番号09780237），並列・分散処理研究推進機構の援助による．また，OSのソースプログラムを提供いただいた富士通並列処理研究センターおよびNEC C&C研究所並列処理センターに感謝する．

参考文献

[1] Y.Iwamoto, K.Ooguri, T.Yoshinaga and T.Baba: A Comparison of Communication Performance in the NEC Cenju-3 and FUJITSU AP1000, Proc. of the FIRST CENJU WORKSHOP, pp60-64(1997).

[2] 岩本善行，澤田東，阿部大輝，澤田康雄，大津金光，吉永努，馬場敬信: メッセージ転送処理の高速化法とその評価，情報処理学会論文誌，Vol 39, No.6, pp.1663-1671(1998).