

商用並列計算機におけるメッセージ受信予測による高速化

2J-02

板野修次 八幡義家 足立涼子 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 をとりあげ、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、メッセージ処理受信の評価・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、そのメッセージ処理転送において、受信側での受信処理にかかる時間が大きいことや、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[1]。

これらを背景として、我々は、メッセージ受信予測法を提案している[2]。本手法がハードウェアや並列ライブラリなど、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し、さらにその性能を評価するため、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、そして今回、商用並列計算機 Sun Enterprise3500 に対して実装を行った。

本稿ではまず、Sun Enterprise の特徴について簡単に述べる。次に、メッセージ受信予測法について述べ、その予測アルゴリズムについて説明する。その後、この予測法を Message Passing Interface(MPI)ライブラリに実装した結果に基づき、評価・考察する。

2. システム構成

Sun Enterprise は、最大で 30 ノードまでの UltraSPARC プロセッサを搭載可能な並列計算機で、Sun microsystems 社によって開発されたものである。この Sun Enterprise は高い性能、且つ多様性と簡易性を同時に可能にした“Giga

High Performance Message Receiving on A Parallel Computer
Syuuji ITANO, Yoshiie YAHATA, Ryouko ADACHI, Yoshiyuki IWAMOTO, Kanemitsu OOTSU, Tutomu YOSHINAGA, and Takanobu BABA

Faculty of Engineering, Utsunomiya University

plane バス“とよばれるシステムバスを持つことを特徴としている。今回使用した、Sun Enterprise3500 は CPU が 8 ノード、メモリは 3GB の構成のものである。

3. メッセージ受信予測法

メッセージ受信処理を行う際、メッセージ待ちアイドル時間が生じる。このアイドル時間を利用して、次に来るメッセージを予測し、その結果に基づいて、後に続く処理を投機的に実行を行ない、処理の高速を目指すのがメッセージ受信予測法である。

その予測手法として今回は、最も実装が容易である直前予測を用いて実装を行った。直前予測とは、直前に到着したメッセージを保存しておき、そのメッセージを用いて受信後の処理を行うものである。図 1 にメッセージ受信予測による高速化の流れを示す。図は上が通常実行時の流れであり、下がメッセージ受信予測を行ない、すべて成功した時の流れである。その際、後処理に要する時間①と、メッセージ比較する時間②との差③を短縮でき、その分ユーザーのプログラムを早く実行することができる。

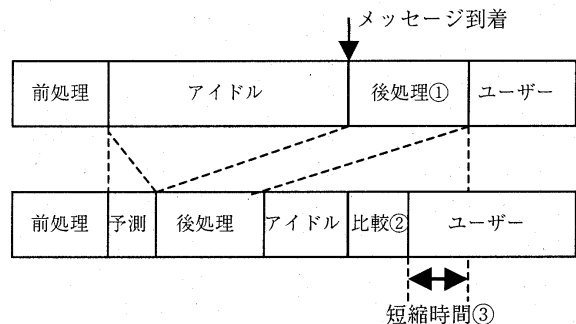


図 1 メッセージ受信予測による高速化

4. 実装

本手法の実装に際し、変更を加えた MPI_Recv

の流れについて簡単に説明する。この関数が実行されると、先ず引数のチェックなどの前処理が行われる。次に、引数によって指定されたメッセージが到着しているかどうかチェックを行なう。もし到着していたら、そのメッセージを使って後処理を行なう。後処理とは、メッセージ到着直後から MPI_Recv 終了直前までに行われる処理のことを指す。また、メッセージが未到着の場合は、先に述べたように、そのままメッセージを待ち続けるため、このアイドル時間中に前回のメッセージを用いて、後処理を先行実行する。その後、後処理の直前まで戻り、実際のメッセージが到着するのを待つ。メッセージが到着したら前回のメッセージと比較し、同じであれば予測が成功しているため、後処理をすべて飛ばして MPI_Recv を終了させることができる。

また、メッセージ予測に失敗した場合には、もう一度、正しいメッセージを用いて通常の後処理を行わなければならない。その際、メッセージを予測して、先行実行している時と区別がつかないため、再び後処理の始まる直前まで戻ることになる。これを避けるために、先行実行中にだけ戻り、メッセージの到着を待てるようにフラグを用意する。

5. 評価

評価は、通常実行時と、本手法を用いて予測がすべて成功した場合と、すべて失敗した場合の3種類において“後処理にかかった時間”を計測し、比較した(図2)。

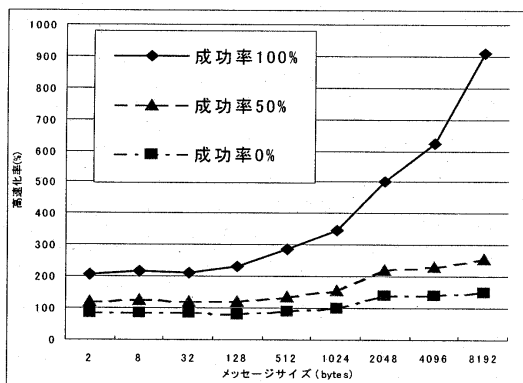


図2 メッセージサイズと高速化の関係

図2の成功率100%、成功率50%、成功率0%とは、それぞれの確率で予測が成功したときに、どれだけ速くなるかを示している。また、高速化率とは、通常実行時の後処理の時間を、本手法を実装した後処理の時間で割ったもので、数値が大きいほど高速化したということになる。

この結果から、予測が成功した場合、最大で912%(約9倍)高速化したことがわかる。これは、後処理の時間だけを比較したもので、全実行時間に対する割合で見ても、最大で約15%高速化している。

また、予測に失敗した場合にも高速化しているところもあるが、これは一度先行実行しているため、後処理の命令コードがキャッシュに置かれるため、次に実行した時には、以前より速くなっていると考えられる。

6. おわりに

本稿では最初に、今回使用した Enterprise の構成を示した。次に、メッセージ受信を高速化するための手法であるメッセージ受信予測法について述べ、それについての実装、評価について述べた。評価では、全体的に高速化がみられる。予測が成功した場合、速くなるのは当然のことだが、予測に失敗したときでも、受信するメッセージのサイズが十分に大きければ、通常実行時より速くなっており、この手法が有効であると確認できた。

謝辞

本研究では、一部文部省科学研究費(基盤(B)課題番号10558039、奨励(A)課題番号11780190)、並列・分散処理研究推進機構(PDC)の援助による。

参考文献

- [1]Y.Iwamoto, K.Ooguri, T.yoshinaga, and T.Baba:
A Comparison of Communication Performance in the
NEC Cenu-3 and FUJITSU AP1000: proceeding of the
FIRST CENJU WORKSHOP, pp.60-64(1997).
- [2]Y.Iwamoto, K.Ootsu, T.Yoshinaga, and T.Baba:
Message Prediction and Speculative
Execution of the Reception Process, Proceedings of
the Eleventh IASTED International Conference,
Parallel and Distributed Computing and Systems
(PDCS '99), pp.329-334(1999).