

Light-MPI 実装方式¹

3ZB-06

(株)日立製作所 中央研究所² 坂口 明彦³(株)日立製作所 ソフトウェア事業部⁴ 岩月 秀樹 呉屋公英⁵

1 はじめに

近年、並列計算におけるメッセージパッシングライブラリの重要性が高まっている。特に、広く用いられているメッセージパッシングライブラリ MPI に対する高速化の要求が強い。一方で、これまでの MPI は科学技術計算の多様な要求に応えるために機能サポートの範囲を広げ、汎用性を高めてきた。高機能な関数の提供によりユーザプログラム工数の削減が可能とはなったが、利用頻度の低い多種の機能を実装することにより、コードの増大やプロトコルの複雑化に伴う性能低下、特にレイテンシ増大が問題となってくる。

本研究では、サポート範囲を利用頻度の高い基本通信関数のみに絞こむことにより高速性に特化した通信ライブラリ「Light-MPI」を提案するとともに、日立スーパーテクニカルサーバ SR8000 においてその実装と性能検証を行った。

2. Light-MPI 仕様

Light-MPI の開発目的は、サポートする機能を基本的な通信に限定することで、簡易性と高速性を追求することである。そのため、Light-MPI の仕様を以下のように定めた。

- ・MPI-1 レベルのみのサポートとし、さらに基本的な通信関数に限定する。
- ・ユーザプログラムのポータビリティの観点から関数インタフェースは MPI と同一とするが、性能向上のために仕様の一部に制限を加える。

通信関数は、基本的なプロセス間通信のために最低限必要なもののみに限定する。具体的には、コミュニケータは MPI_COMM_WORLD のみに限定し、データタイプやコミュニケータ操作関数等の通信とは直接かかわらない関数はサポートから外す。通信関数は、ブロッキング/ノンブロッキングそれぞれの送受信関数のみサポートする。

関数インタフェースは MPI に準拠することでユーザコードのポータビリティを高める。すなわち、Light-MPI で実行可能なユーザプログラムは通常の MPI を用いても実行可能とする。ただし、高速化のための制限事項として以下の三項目を課する。

(1) ワイルドカードの禁止

送受信リクエストのマッチングを明確化するため、ワイルドカードを禁止する。

(2) メッセージの追い越しの排除

送受信リクエストのマッチングは常に発行順とし、タグによるメッセージの追越を不可とする。

(3) 送受信メッセージ長の制約

対応する送受信関数間では同一メッセージ長を指定しなくてはならない。

3. 実装方式

Light-MPI の実装においては、通信メッセージ長によって二つのプロトコルを利用する。

(1) 短メッセージ用プロトコル

メッセージ長が短い場合には予め受信側に用意された通信バッファにデータを送りつける方式を用いる(図 1)。通信バッファは複数面用意し、先頭から順に使用することで送受信プロセスがお互いに使用するバッファの位置を知ることができる。送受信プロセス間でバッファ位置に関する同期を取る必要がないので、レイテンシ削減に有効である。

¹ Implementation of the Light-MPI² Hitachi, Ltd., Central Research Laboratory³ Akihiko Sakaguchi⁴ Hitachi, Ltd., Software Division⁵ Hideki Iwatsuki, Kimihide Kureya

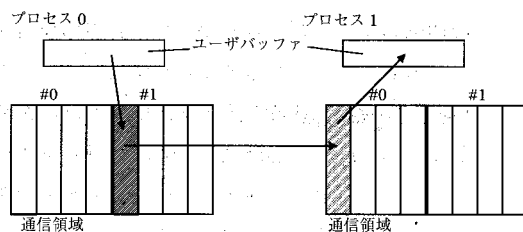


図1 短メッセージ用通信方式

本来のMPIの場合、タグの指定によるメッセージの追い越しが可能のため通信バッファが先頭から消費されるとは限らず、受信側の通信バッファの空き状況を受信側から通知する必要がある。Light-MPIではメッセージの追越しを禁止することにより、通信バッファの空き状況管理の単純化を可能としている。

(2)長メッセージ用プロトコル

長いメッセージの転送を行う場合には、メモリ使用量の制約のため、受信側にすべてのメッセージ長に対応できるバッファを予め用意することはできない。したがって、送受信プロセス間で予め同期を取って送受信するバッファのアドレスを通知しあう方式を用いる。日立SR8000では、ユーザプログラム中で静的に確保されたメモリ領域を高速通信領域(リモートDMA転送が可能なメモリ領域)にマップするコンパイルオプションが存在する。本機能を用いてユーザプログラムの配列領域を高速通信領域に割り当てる。通信プロトコルは、まず受信プロセスから送信プロセスに対して受信用ユーザバッファのアドレスを通知し、それを受けて送信プロセスがデータの転送を行う(図2)。Light-MPIの場合、ワイルドカードの使用を禁止したために、受信プロセスから送信の起動が可能となる。通常のMPI実装で見られる送信側からの通信起動を省略できるため、レイテンシ削減を図ることができる。

上記のようにLight-MPIの場合、メッセージ長によって通信動作を開始するプロセスが異なる。短メッセージ通信の場合には送信プロセスから、長メッ

セージ通信の場合には受信プロセスから動作が開始される。ここで、両プロセスがメッセージ長から自プロセスが動作を開始すべきであると判断すると通信の交差が生じてしまう。しかし、Light-MPIではマッチすべき通信の間でメッセージ長が等しいことを制限事項としているために、そのような通信の交差の可能性を排除することができる。

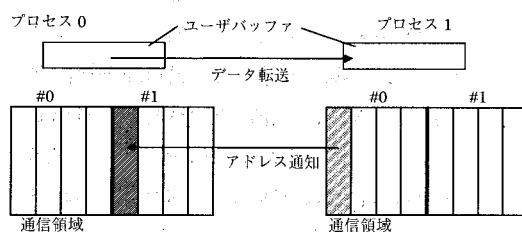


図2 長メッセージ通信方式

4. 性能の評価

Ping-Pong 性能についてLight-MPIと日立SR8000MPIの比較を行った結果を図3に示す(SR8000MPIの0Byteレイテンシの性能を100としたときの相対値)。プロトコルの単純化により、短-中メッセージ領域で通信レイテンシの半減を達成した。

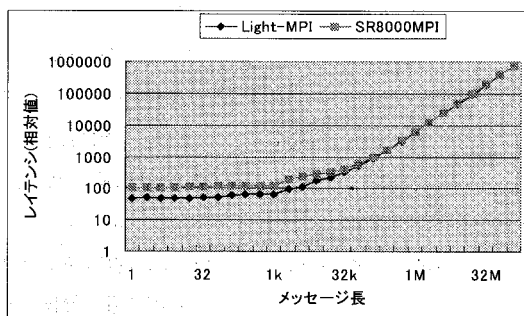


図3 SR8000MPIとLight-MPIの性能比

5. おわりに

機能を絞ることで高性能化を図ったLight-MPIの試作を行い、0Byteレイテンシ50%削減を達成した。

6. 謝辞

本研究は1996年9月に東京大学金田康正教授より示唆をいただいて開始いたしました。深く感謝いたします。