1F－2

超並列計算機SR2201における ソフトバリア同期機能の実装と評価

薗田浩二*、浅間知之**、小林耕三***、秋山幸広***、森山建三****
*（株）日立製作所　システム開発研究所
（株）日立マイクロソフトウェアシステムズ　*（株）日立製作所　ソフトウェア開発本部

1. はじめに

並列計算機を用いた科学技術計算では、同じプログラムコードのプロセスを複数のノード上に配置し、それぞれ異なるデータに対し演算を行うSPMD(Single Program Multiple Data streams)モデルに基づく並列プログラムがよく用いられる。このモデルでは、各ノード上のプロセスが互いに同期しながら演算を進める。このノード間の同期を高速にとるために、ハードウェアによるバリア同期機能が提供される事が多いが、ハードウェアバリア同期機能は高速である反面、使用条件に制約がある場合があり、常には使用できない。

我々は、SR2201において高速性と柔軟性を兼ね備えたソフトバリア同期機能を実現した。本発表では、ソフトバリア同期機能の実現方式を述べ、さらに256ノード構成時の性能評価について報告する。

2. バリア同期機能の機能要件

バリア同期機能に対する機能要件として、以下の３つがある。

(1) 高速性

バリア同期機能は、演算と通信をループ実行するプログラムにおいて演算と通信のフェーズ分けのために使用されることが多い。この場合、オーバヘッドはループ回数に比例して増加するため、バリア同期機能は高速性が要求される。

(2) 同時性

SPMDモデルの並列プログラムでは、１つのプロセスの実行の遅れが全体の性能に大きく影響を与える。このため、プログラマは各プロセスの演算量が等しくなるようにチューニングを行うが、全プロセスが同期点に達してから実行を再開するまでの間隔に、プロセス間でばらつきがあると、このチューニングが困難になる。

(3) 柔軟性

プログラムのデバッグ段階から並列計算機を独占して使用するのはコストが大きいため、通常はマルチユーザ環境下でデバッグを行い、プログラムが完成した後にハードウェアを占有しての高速実行を行う。従って、高速なバリア同期機能であっても、システムの独占運用などの制約は望ましくない。このため、バリア同期機能は使用条件を限定しない柔軟性を備える必要がある。

3. 実現方式

前章の要件を満たすため、以下の方式によるソフトバリア同期機能を実現した。

(1) ツリーとブロードキャストの併用方式

バリア同期機能は、全ノードの同期点到達通知の収集と全ノードへの同期完了通知の２フェーズで実現する。以下ではこれらの実現方式について検討する。

(a) 同期点到達通知収集方式

同期点到達通知の収集方式には、各ノードが単一ノードに通知する集中処理方式と、ツリー構造を利用して階層的に収集する分散処理方式が考えられる。

集中処理方式では、収集ノードの処理量がノード数に比例して増加する。

一方分散処理方式では、同期点到達通知の収集は並列に実行する事ができ、しかも各ノードに集中するメッセージ数は高々その子ノードの数であるため、ノード数増加の影響を受けにくい。従って超並列機では本方式の方が適している。

(b) 同期完了通知方式

同期完了の通知方式には、１ノードが全ノードに通知する方式と、ツリー構造を利用して通知する方式の２つが考えられる。

１対全通知方式では、システムがユニキャスト通信しかサポートしていない場合、ノード数に比例して処理時間が増加する。また、実行再開時間のノード間でのばらつきも大きい。一方、システムがブロードキャスト通信を提供している場合、１回の送信で全ノードに通知できるため高速であり、さらに全ノードが同時に処理を再開する同時性を実現することができる。

ツリー構造を利用する方式では、各ノードが送信するメッセージ数は高々その子ノード数であるが、ノード数が増加すると、ツリーの深さが深くなるため、実行再開時間のノード間でのばらつきが大きくなる。

SR2201は、一定のハードウェア境界内の全ノードへブロードキャストを行う機能を提供している。そこで、このハード機能と以上の検討結果から、ソフトバリア同期機能は、同期点到達通知収集にはツリー構造を用い、同期完了通知には、バリア同期参加ノードがハードウェア境界と一致している場合にはブロードキャスト機能を用い、一致していない場合にはツリー構造を用いる方式とした。

この方式において、１つのソフトバリア同期が対象とするプロセス群は、SR2201のOSが提供する並列プロセス

Software Barrier Synchronization Facility on SR2201

*Koji Sonoda, **Tomoyuki Asama, ***Kouzo Kobayashi, ***Yukihiro Akiyama, ***Kenzo Moriyama
*Systems Development Laboratory , Hitachi, Ltd.
***Software Development Center, Hitachi, Ltd. **Hitachi Microsoftware Systems, Inc.

生成機能で同時に生成されるプロセス群とする。このプロセス群の情報は、バリア同期に参加する全プロセスのノードリストと、自プロセスのノード番号を含み、各ノード上でローカルに取得することができるため、初期化時のツリーの作成は分散して行える。

また、バリア同期参加ノード群とハードウェア境界との一致もそれぞれのノードで判断し、処理を切り替えるため、実行環境の制約がない柔軟なソフトバリア同期機能が提供できる。

(2) リモートDMA機能の適用

SR2201は、高速なノード間通信手段としてリモートDMA機能を提供している[1]。リモートDMA機能では、ユーザプロセスに固定的に物理メモリを割り当て、ノードをまたがるユーザプロセス間でハードウェアが直接データの転送を行う。また、あらかじめネットワークハードウェア起動用のコマンドワードを作成しておき、それを繰り返し利用する送信インタフェース[2]や、受信データの到着確認に割り込みを用いずユーザプログラムが直接メモリをスピンループで読み出す受信インタフェースにより、低レイテンシの通信を可能としている。

そこで、ソフトバリア同期機能では、通信先と送信データ量は固定のため、このリモートDMAの利点を活用して、高速同期を実現した。

4. 性能評価

(1) 割り込み処理の影響

前章で述べた方式に従い、ソフトバリア同期機能の初期バージョンを作成し性能を測定した。測定は、ソフトバリア同期を10,000回ループ実行するプロセスを256ノードに配置して行った。表1に測定結果を示す。

表1 256ノードの同期性能(タイマ処理改善前) (μsec)

最小値	平均値	最大値
30.21	43.76	291.77

結果では平均値が最小値の1.45倍となっている。この原因を探るため、5μsec毎のバリア同期性能の頻度分布を求めた(図2)。

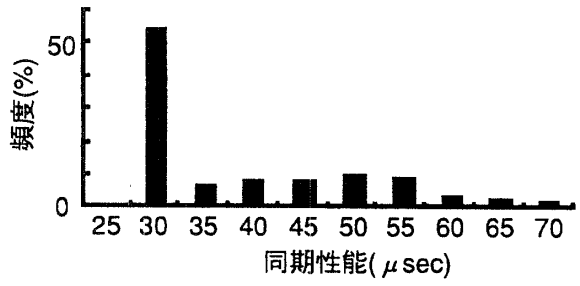

図2 同期性能の頻度分布(タイマ処理改善前)

これによると、最小値が含まれる30～35μsecの頻度は53%であり、次に50～55μsecを中心とした小さな山があることが分かる。その差は約20μsecであり、バリア同期参加ノードのいずれかで20μsec前後の処理が行われていることを示している。各ノード上で動作するプロセスは1つだけなので、これらはOSによる割り込み処理である可能性が高い。その中でも20μsec前後の処理時間で、コンスタントに発生する処理はタイマ割り込み処理である。

初期バージョンで使用したOSでは、各ノード上で10msec毎にタイマ割り込みを発生させる設定になっており、各ノード間での割り込み発生時刻の同期は行っていない。このため、256ノードでバリア同期を行うと、いずれかのノードがタイマ割り込み処理と遭遇する確率が高くなる。タイマ割り込みとぶつかった場合、同期点到達通知の送信が遅れるため、その分性能が劣化していると予測できる。

そこで、タイマ割り込み発生時刻を全ノードで同期させた。タイマ割り込み処理変更後のバリア同期性能分布を図3に示す。30～35μsecの頻度が99%と高く、大幅に性能改善できていることが分かる。この結果、平均性能値は31.91μsecとなり、改善前と比べて27%向上した。

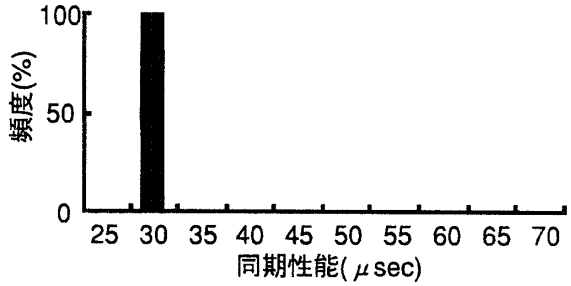

図3 同期性能頻度分布(タイマ処理改善後)

(2) ブロードキャストの効果

表2に256ノードでのブロードキャスト使用時と不使用時のソフトバリア同期性能を示す。ブロードキャストを使用すると、使用しない場合のほぼ半分の時間でバリア同期が完了しており、効果が大きいことが確認できた。

表2 256ノードでのブロードキャストの効果(μsec)

BC使用時の同期性能	BC不使用の同期性能
31.91	59.77

5. おわりに

SR2201のソフトバリア同期機能は、ツリー構造とブロードキャストを併用し、ノード間通信にリモートDMA機能を適用して実現する。この結果、柔軟性を持ちながらも、256ノード時の性能が31.91μsecと高速なソフトバリア同期機能を実現できた。

参考文献

[1] 岩嵜、他: 超並列計算機SR2201におけるネットワーク・インタフェース・アーキテクチャ　情報処理学会第54回全国大会(1997)

[2] 秋山、他: 並列計算機SR2201における高速ノード間通信APIの実現と評価　情報処理学会第52回全国大会(1996)