マルチクラスタ用ソフトウェア分散共有メモリの提案

吉川克哉† 城田祐介† 吉瀬謙二† 本多弘樹† 弓場敏嗣†
電気通信大学 大学院情報システム学研究科†

1 はじめに

スーパークラスタ[3]やグリッドを利用し複数のクラスタシステムを繋げたマルチクラスタが今後普及することが期待されている。このようなシステムを効率良く利用するためには並列プログラミング環境の構築が重要である。ページベースソフトウェア分散共有メモリ(SDSM)は、分散メモリシステム上で共有メモリプログラミングモデルを提供する一方で、パフォーマンスのスケーラビリティが得られないため実用的なレベルに達しているとは言い難い。しかし、SDSM の用途はアプリケーションプログラマが直接 SDSM コードを記述するだけのものではない。例えば、従来は共有メモリシステムで実現されていた OpenMP が、SDSM に対して OpenMP プログラムをコンパイルすることでクラスタ上で実現されているように、スケーラブルな SDSM の実現が望まれている。

そこで、我々はまず、マルチクラスタを PC クラスタ上でエミュレートするシステムを構築し、予備評価として既存のページベース SDSM を同システム上でそのまま用いた場合のオーバヘッドを求める。マルチクラスタ上でページベース SDSM を実行する場合、ページ転送などでクラスタ間のトラフィックが頻繁であるため、本稿ではキャンパスグリッド規模までのマルチクラスタを前提とする。そして、明らかになったオーバヘッドを削減する足掛かりに、マルチクラスタの階層的アーキテクチャを考慮したマルチホーム方式を提案し、既存の SDSM 上に実装して、予備評価を行う。また、同方式を用いることによって、大規模クラスタシステムを、クラスタ間通信遅延がスイッチングディレイのみの論理的なマルチクラスタとしてみることができる。大規模クラスタシステムでのスケーラブルな SDSM の実現に向けて、マルチホーム方式が大規模な並列システムとして標準的な SMP クラスタシステムでも有効である場合があることを予備評価で示す。

A Proposal of a Software Distributed Shared Memory for Multi-clusters

† Katsuya YOSHIKAWA, Yusuke SHIROTA, Kenji KISE, Hiroki HONDA and Toshitsugu YUBA

† Graduate School of Information Systems, The University of Electro-Communications

2 キャンパスグリッド規模マルチクラスタで既存の SDSM をそのまま用いる問題点

2.1 SDSM の選択

マルチクラスタとの親和性などを考慮した上で、既存の各種 SDSM から JIAJIA[2]を選択する。JIAJIA は、高性能なばかりでなく、使用可能なメモリサイズが 1 ノードの物理メモリサイズに限定されないので、大規模問題を前提とする我々の要求に合致する。JIAJIA が利用するページコヒーレンシプロトコルのメモリアーキテクチャは、ライトバックの戻り先であるホームノードがページごとにある固定ノードに決められているホームベース方式である。

2.2 マルチクラスタ上での実行に伴うオーバヘッド

JIAJIA の実装は、他の多くの各種 SDSM と同様に全ての CPU をフラットな構成とみるものになっている。このため、マルチクラスタ上でそのまま用いると頻繁なページ転送などでクラスタ間通信遅延が性能ボトルネックになってしまう[1]。

3 マルチホーム方式の提案

3.1 クラスタキャッシュ

クラスタ間通信遅延を隠蔽するために、クラスタのデータローカリティを利用する。具体的には、更新されたページをクラスタごとにクラスタ内のあるノードにキャッシュし、同キャッシュをクラスタキャッシュとして利用する。これにより、本来ならばクラスタ外のホームノードとのページ授受が必要なページフォルト処理をクラスタ内で解決することが可能になる。

3.2 クラスタごとのホームノード

本稿では、「クラスタごとのホームノード」という概念を導入し、同ノードをクラスタキャッシュとして利用することを検討する。このクラスタキャッシング方式を、全てのクラスタごとのホームノードに対してライトバックを行うマルチホーム方式（MH）とそうでないプロキシホーム方式に分類する。これに対して従来方式をシングルホーム方式（SH）と呼ぶ。マルチホーム方式では、ライトバックを積極的に全てのクラスタのホームノードに対して行うので、複数のホームノード間

のコンシステンシをとるオーバヘッドが大きい。このため、パフォーマンス向上は見込めないが、クラスタキャッシュとしての有効性の検証という目的で我々はまずマルチホーム方式を実装する。その後、マルチホーム方式のコンシステンシの制約を緩めるなど、より効率的なクラスタキャッシング方式であるプロキシホーム方式について検討する[1]。

3.3 マルチホーム方式の実装方法

本稿で行うマルチホーム方式の実装は、最もシンプルな方式を採用する。具体的には、ホームノードの数はクラスタごとに1つ[2]とし、すべてのクラスタの構成がホモジニアスであるとの前提のもと、ホームノードをクラスタ間で対称となるように配置する。ライトバックはすべてのホームノードに対して行い、ページリクエストの発行はクラスタ内のホームノードに対して行う。

4 予備評価

4.1 予備評価システム

マルチクラスタ(MC)をシングルクラスタ(SC)上でエミュレートするシステムを構築した。具体的には、SMP-PC クラスタを論理的に2つの SMP-PC クラスタ構成とみなす疑似マルチクラスタシステム(図1)を構築した。クラスタ外ノードとの通信をクラスタ間ルータにみたてた3台の SMP-PC ノードを経由して行わせることで、クラスタ間遅延をエミュレートした。予備評価に使用した SMP-PC クラスタシステムの各ノードは、CPU が PentiumIII 866MHz(×2)、メモリ1GB、OS が Red Hat Linux 7.1、コンパイラが gcc 2.96、NIC は 100BASE-TX である。[2]

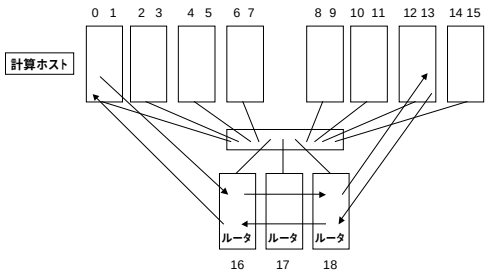

図1：疑似マルチクラスタシステム

4.2 予備評価用ベンチマークプログラム

予備評価用ベンチマークプログラムには、行列積を計算する MM(Matrix Multiply)を用い、問題サイズを 2048×2048 とした。ホームノードの配置は、シングルホーム方式においてメモリアクセスローカリティが最適になるようにアプリケーション側で設定した。

4.3 予備評価結果

予備評価結果を表1に示す。

	SC･SH	SC･MH	MC･SH	MC･MH
1CPU	155.35			
2CPU(1+1)	89.66	89.07	109.56	103.04
4CPU (2+2)	61.29	60.14	80.46	73.09
8CPU (4+4)	40.48	41.30	60.24	54.11
16CPU (8+8)	46.18	32.91	63.79	45.52

表1：MM の実行時間[sec]

SC 及び MC において、マルチホーム方式ではライトバックをクラスタごとのホームノードに対して行うため、ページ読み出しのコンテンションが緩和されたと考えられる。MC の場合は、これに加えて、ホームノードがクラスタごとにあることで、クラスタローカリティでクラスタ間通信遅延を隠蔽できた結果、高い性能向上を示している。

しかし、ベンチマークプログラムによっては性能向上が期待できない。マルチホーム方式はホームページ対して実行された書き込みでも diff を作成し、もう一つのホームノードに対してライトバックを行う必要があるからである。

5 おわりに

本稿ではクラスタごとのホームノードのクラスタキャッシュとしての有効性があることを示した。現在は、効率的なクラスタキャッシング方式であるプロキシホーム方式の実装を進めている。クラスタキャッシングをシステムが動的に行うか、プロファイリング手法を用いるのかも今後検討したい。SCore システムソフトウェア上で動作する SDSM SCASH 上でも実装を行うことも今後の課題としたい。

参考文献

[1] Arantes,L.ら: The Impact of Caching in a Loosely-coupled Clustered Software DSM System., Proceedings of the IEEE International Conference on Cluster Computing, pp.27-34 (2000).

[2] Hu,W.ら: JIAJIA: A Software DSM System Based on a New Cache Coherence Protocol, HPCN Europe, pp.463-472 (1999).

[3] 工藤知宏ら: AIST スーパークラスタ構想、情報処理学会研究報告, Vol.2002, No.91, pp.103-106 (2002).

[1] プロキシホーム方式のキャッシングは必ずしもライトバック時に行う必要はない。クラスタ外のホームノードから受け取ったページをキャッシングする場所として利用することも考えられる。

[2] 最適なホームノードの数が有り得る。