3R-05

Scopeごとにコンシステンシモデルの変更が可能なソフトウェア分散共有メモリ

城田祐介† 吉瀬謙二‡ 弓場敏嗣‡
電気通信大学電気通信学部† 電気通信大学大学院情報システム学研究科‡

1 はじめに

計算機クラスタの分散メモリを透過的に扱うソフトウェア分散共有メモリ (DSM) において, false sharing を緩和する LRC(Lazy Release Consistency), ScC(Scope Consistency)[2] や, Multiple Writer キャッシュコヒーレンスプロトコル等が提案されている. 同一 scope(クリティカルセクション) を複数 PE で更新する集計処理等では, scope 内の共有変数の更新と転送が繰り返しながら逐次的に実行されるため, 並列効果が得られない.

本論文では, ScC モデルを拡張し, scope ごとにコンシステンシモデルを変更可能とするソフトウェア DSM を提案する. CC-NUMA 型のソフトウェア DSM [3] に変更を加えて, 本方式を実装し, 評価を行った結果について報告する.

2 Scopeごとにコンシステンシモデルの変更が可能なDSMの提案

2.1 拡張 ScC モデルの提案

Scope は, 同一 lock でつくられた全てのクリティカルセクションである. lock(i) 操作から unlock(i) 操作までをセッション i という. ScC は, ある scope のセッション i の変更点が, 同一 scope の次セッション $i+1$ に反映されることを保証する. つまり, セッション間では scope ごとに, LRC で一貫性が維持される.

高い局所性を示すアプリケーションでは, 全ての scope に関して LRC を適用すればよい. しかし, 本論文が対象にしている集計処理等の実行比率が大きい場合, 十分な並列効果が得られない. そこで, ScC を拡張することを提案する. つまり, scope ごとに, セッション間のコンシステンシモデルを変更可能とする. scope ごとに, LRC と集計処理に特化した Log Based Consistency(LBC) モデル [1] が選択可能となる (図 1). これにより, ソフトウェア DSM が, 集計処理等にまで適用可能となる.

2.2 LBC モデル

一般に, 集計処理は, 可換/結合法則が成立する場合が多い. そこで, このような操作に対して, LRC より緩いコンシステンシモデルである LBC モデルが提案されている [1].

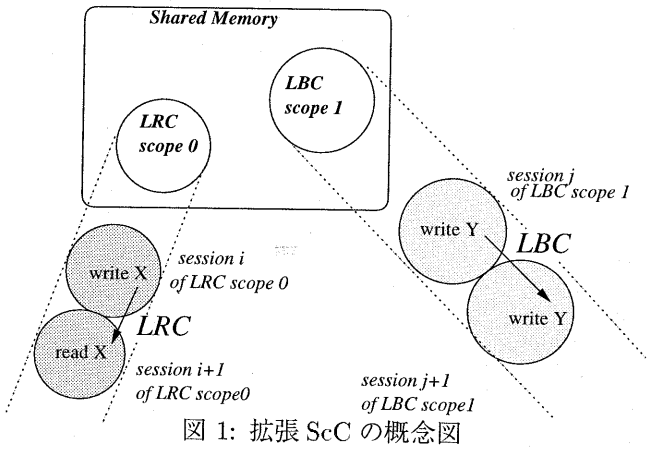

図 1: 拡張 ScC の概念図

例えば, アドレス X に, Node0(アドレス X のホームノード), Node1, Node2 がそれぞれ, a,b,c のインクリメントを繰り返し実行した結果, それぞれ, A, B, C だけ加算したとする. LRC モデルを用いた JIAJIA では, 同一アドレスへの変更は排他制御され, それぞれのインクリメント操作が逐次化される (図 2). 提案する DSM では, 並列に同一アドレスへのインクリメント操作を許す. つまり, セッション間では一貫性はとられないで, 明示的な一貫性維持操作によって各ノードがインクリメントした値がホームノードに送られる. 全ノードがローカルにインクリメントした値の和をホームノードが計算することで, 一貫性がとられる (図 3). 一貫性維持操作が実行されるまでの間, アドレス X へのインクリメントは完全にローカルな処理になり, 並列実行される.

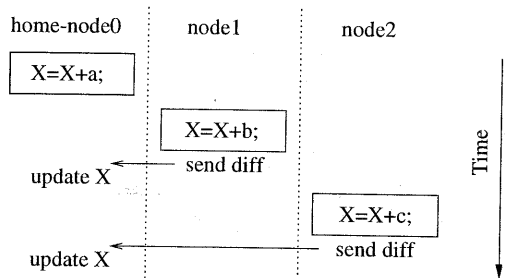

図 2: JIAJIA による同一クリティカルセクションの更新

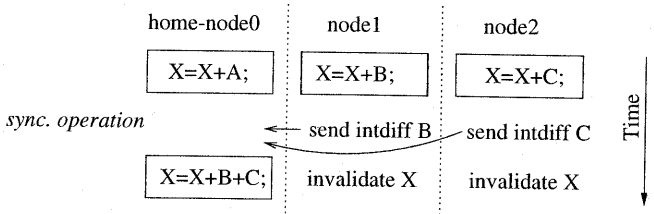

図 3: LBC の一貫性維持操作

Selecting a consistency model for scopes in homebased software DSM
Yusuke SHIROTA, Kenji KISE, and Toshitsugu YUBA
† The University of Electro-Communications
‡ Graduate School of Information Systems, The University of Electro-Communications

3R-05

2.3 LBC モデルの実装

LBC モデルを実装する. ホームノード以外のノードは, 次のいずれかの状態で必要なページをキャッシュする: Read-Write(RW), Read-Only(RO), Invalid(INV)(図 4)[1] .

read 操作がミスした場合, ページのホームノードよ

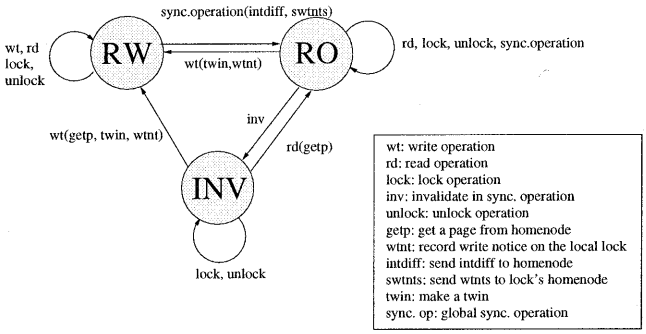

図 4: LBC モデルの状態遷移図

りページをフェッチし, RO でキャッシュに入れる.

write 操作がミスした場合, ダーティなキャッシュページを検出するため, ローカルな同期変数 lock にフォルトページのアドレス (wn) を記録する. write 操作を再開する前に, キャッシュページのコピー (twin) をつくる.

一貫性維持操作が実行されると, twin を基準値にインクリメントした intdiff を求める. intdiff はホームノードに転送される. ホームは, 複数ノードの intdiff の和をホームページに加算/反映させ, 一貫性をとる. 次に, lock マネージャノードに, lock に記録された wn を転送する. lock マネージャノードは, 全ノードの wn の和集合をブロードキャストする. これに, 自ノードでキャッシュしているページがあれば, ページを無効化する. そうでない場合は RO にする. これは, キャッシュへの write 操作を新たに検出するためである. このため, lock に記録されている wn は初期化する.

3 評価

評価には, データマイニング応用における Apriori アルゴリズム [4] の相関ルール検索を用いた. 同アルゴリズムでは, 膨大なトランザクションデータベースを走査し, 集計処理する.

Myrinet 接続の PentiumIII PC クラスタ上に, 提案する DSM と JIAJIA をそれぞれ実装した. 同 DSM 上に相関ルール検索プログラムを移植し, 実行/比較した (表 1). トランザクションデータは [4] に従い, 人工的に作成した: 総アイテム数 =600 件, レコード中のアイテム数 = 平均 10 件のポアソン分布, 最小サポート =0.8%, トランザクション数 =2000 件.

表 1 より, JIAJIA では, 並列処理が逆効果となっている. 本 DSM でも台数効果は得られていない. これは JIAJIA 上で大きい問題サイズで走らせると, 余りにも

[1] unlock で, 状態遷移しないのは, 現実装では, LRC と LBC が同時実行する場合, それぞれ別々にアロケートしなくてはいけないという, プログラミング制約がある為である.

DSM	PE 数	実行時間
本 DSM	1	3
	2	3
	4	3
JIAJIA	1	3
	2	280
	4	292

表 1: 相関ルールの検索時間 [s]

時間を要するため, 問題サイズを小さくしたことが原因である. 最小サポート値, トランザクション数をそれぞれ 0.5%, 400,000 件に変化させ, 問題サイズを大きくし, 本 DSM 上で実行した (表 2). 表 2 より, 十分な台数効

PE 数	pass1	pass2	pass3	pass4	pass5	pass6	合計	台数効果
1	10	618	31	37	31	21	758	1.00
2	16	360	17	19	16	13	443	1.71
4	8	187	9	9	8	7	229	3.31

表 2: 本 DSM 上での相関ルールの検索時間 [s]

果が得られていることが確認できた.

4 おわりに

Scope ごとに LRC と LBC の双方を選択することが有効なアプリケーションを移植し, 評価を行うことが今後の課題である.

参考文献

[1] 平山 秀昭, 本多 弘樹, 弓場 敏嗣. 可換/結合法則が成立する操作を対象としたログベース更新型分散共有メモリ. 電子情報通信学会論文誌 Vol. J83-D-I, pp.440-458, May 2000.

[2] L. Iftode, J.P. Singh, K. Li. Scope consistency: A bridge between release consistency and entry consistency. Proc. of the 8th ACM Annual Symp. on Parallel Algorithms and Architectures(SPAA'96), pp.277-287, June 1996.

[3] Weiwu Hu, Weisong Shi, Zhimin Tang. JIAJIA : An SVM system based on a new cache coherence protocol. Proc. High Performance Computing and Networking (HPCN'99), pp.1147-1150, April 1999.

[4] R. Agrawal, R. Srikant. Fast algorithms for mining association rules. Proc. of 20th VLDB Conference, pp.487-499, Sept. 1994.