Cellプロセッサにおける ソフトウェアトランザクショナルメモリの実装と評価

出宮 健彦† 高山 征大† 境 隆二†

†（株）東芝 デジタルメディアネットワーク社 コアテクノロジーセンター

1 はじめに

近年，プロセッサの性能向上は周波数向上からマルチコア化へと急速にシフトしている．一方，ソフトウェアの開発においては，マルチコアプロセッサの性能を引き出す並列プログラミングの手法が課題になっている．一般に，並列プログラムにおいては複数のスレッドが共有する変数が存在する．このような共有変数へのアクセスは排他的に行う必要がある．従来の排他制御の実装としてロック方式があるが，同期処理を正しく行うことが困難である．そこで，排他制御を用いないロックフリー方式のうちトランザクショナルメモリ [1] が注目されている．トランザクショナルメモリでは複数スレッドに対して共有変数への同時アクセスを許可する．競合が発生した場合には関連する処理の中断および再実行を行う．これにより，並列実行の度合いが高くなり，高い性能が期待できる．

本論文では，マルチコア CPU の一つである Cell プロセッサにおいてソフトウェアトランザクショナルメモリ [2] の実装を行い評価した．

2 ソフトウェアトランザクショナルメモリ

トランザクションとは，不可分操作と直列実行性の性質を満たす有限な一連の処理をさす．プログラマは共有変数の一貫性を保証するため，共有変数へのアクセスを含む一連の処理をトランザクションとして指定する．

トランザクショナルメモリにおける読み出しと書き込みはアトミックな処理として行われる．トランザクションを実行しているスレッドで更新内容を保持しておき，コミットと呼ばれる共有メモリに更新内容を反映させる操作で他のトランザクションとの競合を解決し更新内容を確定させる．

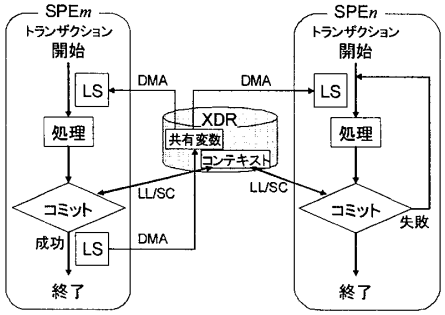

図 1: Cell におけるトランザクショナルメモリ

トランザクショナルメモリはロック方式と比べてアトミック処理したい操作をブロック単位で指示する合成性と，排他制御がコミットに局所化されることによるスケーラビリティ向上の面で有利である．

3 実装と評価

3.1 実装

Cell プロセッサを用いた一般的なプログラミングでは，主記憶である XDR メモリ 上に置かれているデータを SPE のローカルストア (LS) に DMA 転送でコピーして処理を行い，その結果を DMA 転送で XDR に書き込むという形である．これは，トランザクショナルメモリがスレッドで更新内容を保持しておくというアルゴリズムに合っていると言える．実装はまず，x86 プロセッサ環境で行い ([3], [4])，その後 Cell プロセッサ上に移殖した (図 1)．コミット時に使用されるアトミック命令について，x86 は Compare-and-Swap (CAS) 命令で，PowerPC 系の命令体系である Cell は，Load-Link/Store-Conditional 命令を用いて実装している．

3.2 評価

実装したトランザクショナルメモリに対して，[1] を参考にカウンタとキューによる評価を行った．従来手法としてミューテックスを用いた排他制御を評価の比較対象とした．以降のグラフで横軸は SPE の数を，縦軸は単位時間当たりの各々2 つの評価で行う処理の実

Implementation and Evaluation of Software Transactional Memory for Cell Processor

Takehiko DEMIYA†, Motohiro TAKAYAMA†, and Ryuji SAKAI†

†Core Technology Center, Digital Media Network Company, Toshiba Corporation

{takehiko.demiya, motohiro.takayama, ryuji.sakai}@toshiba.co.jp

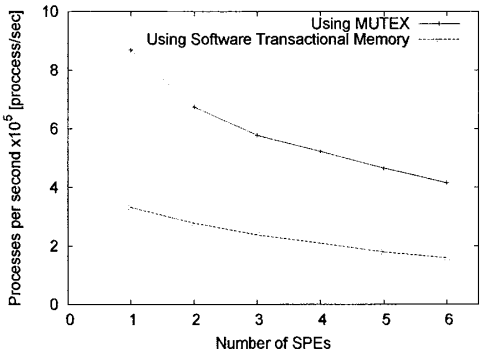

図 2: カウンタ増分での比較

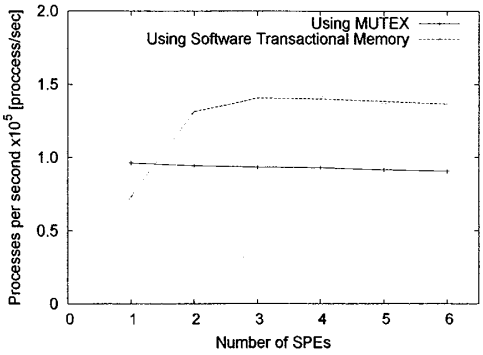

図 3: 双方向リストキュー操作での比較

行回数を示す.

3.2.1 カウンタによる評価

カウンタ値を共有メモリにおいて，複数 SPE から増分を行う評価を行った．SPE 数を n として，1SPE あたり $(5400/n)$ 回の増分を実行した．

評価結果を図 2 に示す．どちらの手法も SPE 数が増加するに伴って，実行速度が低下している．これは，SPE 数の増加に伴い DMA 転送のキャッシュ・ミス・ヒットが発生することによるオーバーヘッド考えられる．一方，ミューテックスとトランザクショナルメモリとの間に差がある．これは，トランザクショナルメモリが，共有メモリ書き換えのログ情報を格納しているコンテキストの読み書きなどで，ミューテックスよりも DMA 転送を多用しているためと考えられる．以上から，共有変数 1 つでは，実行時間が DMA 転送に比べて十分大きい処理量の場合には，従来手法とトランザクショナルメモリには大きな差が生まれないと考えられる．次節で複数の共有変数を用いて評価を行う．

3.2.2 キュー操作による評価

双方向リストを用いたキュー構造で評価を行った．初期状態としてキューには 2 つデータが入っており，SPE 数を n として，1SPE あたり $(5400/n)$ 回のエンキューとデキューの組を実行した．従来手法はエンキューとデキューそれぞれで共通のミューテックスを取得・解放し，トランザクショナルメモリでは共通のトランザクショナルのコンテキストを持ちエンキューとデキュー毎にコミットを行う実装とした．本評価プログラムを実行したところ，まだ，処理量が小さかったため，擬似的に負荷を増やすためそれぞれの手法のエンキュー，デキューに繰り返し数が同じビジーループを挿入し DMA 転送に比べて処理量が大きくなるようにした．

評価結果を図 3 に示す．DMA 転送のオーバーヘッドについてはカウンタと同様の傾向が見られた．トランザクショナルメモリでは，エンキューとデキューが並列実行できるため 2SPE 以上で実行速度が増加しており，加えて 3SPE でも若干，実行速度が増加している．これは，トランザクショナルメモリではコミット時に排他処理が局所化されることで，キュー操作および擬似的に与えた負荷が並列に実行されたためと考えられる．以上から，トランザクショナルメモリの投機的実行による恩恵が確認できた．

4 おわりに

マルチコア CPU の一つである Cell プロセッサにおいてソフトウェアトランザクショナルメモリを実装し評価した．評価の結果，トランザクショナルメモリの投機的実行による恩恵が確認できた．今回実装したコードは，ダブルバッファーを用いた DMA 転送時間の隠蔽などの最適化は施していない．今後は，Cell プロセッサに最適な実装を行い，ロック方式よりもプログラミングが容易であるとする点を評価したい．

参考文献

[1] Maurice Herlihy and Nir Shavit, *The Art of Multiprocessor Programming*, MORGAN KAUFMANN PUBLISHERS, 2008.

[2] Nir Shavit and Dan Touitou, "Software Transactional Memory," in *Proc. of the 14th ACM SPDC'95*, 1995, pp.204-213.

[3] "steps to phantasien (2007-06-23)", http://www.dodgson.org/omo/t/?date=20070623, 2008 年 12 月 19 日アクセス.

[4] Tim Harris and Keir Fraser, "Language Support for Lightweight Transactions," in *Proc. of the OOPSLA'03*, 2003, pp.388-402. Laboratory, 1987.