

優先度を用いたハードウェアトランザクショナルメモリの高速化手法

山田 遼平*, 間下 恵介, 三宅 翔, 津邑 公暁 (名古屋工業大学)

A Priority-Based Scheduling for Hardware Transactional Memory

Ryohei Yamada, Keisuke Mashita, Sho Miyake, Tomoaki Tsumura (Nagoya Institute of Technology)

1. はじめに

マルチコア環境における共有リソースへのアクセス調停にはロックが広く用いられているが、並列性の低下やデッドロックの発生などの問題がある。そこで、ロックを使用しない並行性制御機構として **Transactional Memory (TM)** ⁽¹⁾が提案されている。TM では投機実行が失敗した場合、再び競合が発生することを防ぐため、トランザクションの再実行までに待機時間を設定するアルゴリズムが採用されている。本稿では、この待機アルゴリズムを改良し、トランザクションの実行状況に応じた待機処理を行うことで高速化する手法を提案する。

2. Hardware Transactional Memory

TM はデータベース上のトランザクション処理をメモリアクセスに適用した手法であり、従来ロックで保護されていた処理範囲をトランザクションとして投機的に実行する。なお、TM におけるトランザクションの投機実行では、共有リソースに対する更新の際に更新前の値を保持するログを用意する必要がある (バージョン管理)。また、トランザクションを実行するスレッド間で、競合が発生していないかを常に検査する必要がある (競合検出)。TM のハードウェア実装である **Hardware Transactional Memory (HTM)** では、これらの機構をハードウェアで実装することで、処理を高速化している。

3. 優先度を用いた待機アルゴリズムの改良

HTM ではスレッド間で競合が発生した際、一方のトランザクションをアボートすることでデッドロックの発生を回避する。そして、アボートされたトランザクションはバックオフとして設定された時間だけ待機した後、再実行される。しかし、このように待機時間を設定する方法では、再び競合することなく再実行可能となる時間を超えて待機してしまい、バックオフによる待機時間の一部が無駄になってしまう可能性がある。

そこで本稿では、各スレッドが実行するトランザクションに **優先度 (Priority)** と呼ぶ値を設定し、この値を用いて待機処理を設定する手法を提案する。この優先度の値は、トランザクションの進行状況に比例して大きくなるように設定する。ここで、この優先度の値は

$$\text{Priority} = \frac{1}{\frac{1}{T} + (S_0 - S) + (L_0 - L)} \quad (1)$$

のように定義する。なお、 T はトランザクションを開始してから経過時間、 L , S は現在実行中のトランザクション内でロー

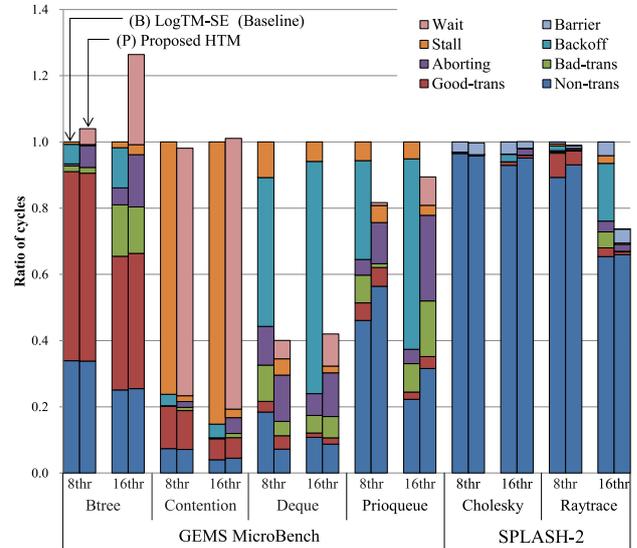


Fig.1 Execution cycles ratio

ド・ストアアクセスした回数、 L_0 , S_0 は現在実行中のトランザクションに関して、過去に実行した際のロード・ストアアクセス回数を示している。競合が発生した際には、式(1)を用いて算出された各トランザクションの優先度の値を比較し、より大きな値を持つトランザクションを優先的に継続実行させることで、残り実行時間が少ないトランザクションを優先的に実行できるようにする。

4. 評価

以上で述べた提案手法をシミュレーションにより評価した。評価には GEMS microbench および SPLASH-2⁽²⁾を使用し、各ベンチマークはそれぞれ 8, 16 スレッドで実行した。各ベンチマークの実行結果を Fig.1 に示す。結果は、既存の LogTM-SE(B)、提案した HTM(P) の実行サイクル数を示しており、LogTM-SE(B) の実行サイクル数を 1 として正規化している。評価の結果、本提案手法により無駄な待機時間が削減されたことで、既存の HTM と比較して最大 59.9%、16 スレッドで平均 11.2% の実行サイクル数を削減し、性能が向上した。

文献

(1) Maurice Herlihy and J. Eliot B. Moss: Transactional Memory: Architectural Support for Lock-Free Data Structures, *Proc. 20th Annual Int'l Symp. on Computer Architecture*, pp.289-300 (1993)

(2) Woo, S. C., et al.: The SPLASH-2 Programs: Characterization and Methodological Considerations, *Proc. 22nd Annual Int'l. Symp. on Computer Architecture*, pp.24-36 (1995)