

不揮発性メモリを使ったデータ永続化システム NV-HTM の評価

1200283 飯干 寛幸 【プログラミング言語研究室】

1 はじめに

電源が失われても内容を保持するメモリとして不揮発性メモリが注目されている。不揮発性メモリを使ったシステムでは、不揮発性メモリに書き出されたときデータが永続化される。不揮発性メモリへのアクセスは揮発的なキャッシュを介して行われるため、キャッシュ上に残っているデータは計算機の電源喪失とともに失われる。さらに、キャッシュが書き戻されるタイミングはユーザ側から把握できない。そのため、不揮発性メモリ上にあるデータの一貫性を保つためには、適切にキャッシュを書き戻す命令を使って書き込み内容を永続化する必要がある。

不揮発性メモリを使った、データの一貫性を保ちつつ永続化されたデータ構造を作る方法が研究されている。その中には二つのアプローチがある。一つ目は、データの永続化機能をもつトランザクショナルメモリを開発し、それを使って汎用のデータ構造を永続化する方法である。トランザクショナルメモリは、スレッド間で同期が必要なメモリ操作をトランザクションとしてまとめて実行するための仕組みである。二つ目は、個々のデータ構造を不揮発性メモリに特化させて永続化する方法である。本研究では、一つ目の方法に基づいたトランザクショナルメモリの一つである NV-HTM[1] のパフォーマンス評価を行う。そのために、二つ目の方法に基づいて永続化した B⁺-木である FPTree[2] と NV-HTM を使い永続化した B⁺-木 (以下、B⁺-Tree_{NH} と表記) を比較する。比較実験にあたって、一般的な B⁺-木と FPTree に加えて、不揮発性メモリの領域を管理するためのメモリアロケータを実装した。さらに、公開されている NV-HTM のコードは性能評価用の DRAM を用いた不揮発性メモリのエミュレータを使うものだったため、不揮発性メモリの実機を使うよう修正した。

2 NV-HTM

NV-HTM は、データの永続化機能をもつトランザクショナルメモリの一つである。NV-HTM は、NVM 上の永続化されたユーザデータのコピーを DRAM 上に作る。ユーザプロセスは永続化されたデータにアクセスするとき、トランザクションを開始し、その中で DRAM 上のコピーにアクセスする。トランザクション内で書き込んだ内容は、同時に不揮発性メモリ上にログとして記録する。不揮発性メモリ上のログは、ユーザプロセスと並行に動作する Checkpoint プロセスと呼ばれるプロセスが、不揮発性メモリ上のユーザデータに反映させる。反映し終わったログは Checkpoint プロセスによって削除される。

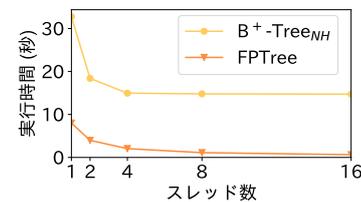


図1 スケール性能測定実験 (挿入操作)

3 実験と考察

B⁺-Tree_{NH} と FPTree の実行時間とスケール性能を比較するために、スレッド数を変化させたときの実行時間を測った。実験を行なった環境を示す。

CPU Intel Xeon Gold 6240 (2.60GHz, 18 コア, キャッシュラインサイズ 64B)

不揮発性メモリ Intel Optane DC Persistent Memory

OS Ubuntu 18.04.3 LTS

C コンパイラ GCC version 7.4.0

図1は250万回の挿入操作を行ったときの結果である。B⁺-Tree_{NH} は FPTree に比べて遅く、4スレッドまでしかスケールしなかった。

そこでさらに、NV-HTMのボトルネックを調べるために、B⁺-Tree_{NH}においてユーザプロセスが本来の処理以外に消費した時間を原因別に測定した。その結果、トランザクションがアボートされることによる消費時間は、スレッド数が少ないとき消費された時間の半分近くを占めており、ログが満杯になったことによる Checkpoint プロセスのログ削除を待つ時間はスレッド数によらず残りの消費時間のほとんどを占めていた。このことから、NV-HTMが遅くなった原因として、トランザクションのアボートが大量に発生していること、Checkpoint プロセスによるログ処理が遅いことがわかった。

4 おわりに

本研究では、データ永続化システム NV-HTM の評価を行った。その結果、B⁺-Tree_{NH} は FPTree に比べて遅く、4スレッドまでしかスケールしなかった。そこでさらに NV-HTM が遅い原因を調べた結果、ログの処理速度が遅いこととトランザクションのアボートが主な原因であることがわかった。

参考文献

- [1] C. Daniel et al. Hardware Transactional Memory Meets Memory Persistency. In *Proc. IPDPS'18*, pp. 368–377, 2018.
- [2] I. Oukid et al. FPTree: A Hybrid SCM-DRAM Persistent and Concurrent B-Tree for Storage Class Memory. In *Proc. SIGMOD'16*, pp. 371–386, 2016.