不揮発メモリを用いたSiloロギング法の評価

田中 昌宏[1,a)]　川島 英之[1]

トランザクション処理技法 Silo のログ永続化法を設計・実装し，Optane を用いて評価した．スレッド数，バッファサイズなどのパラメタが性能に影響することが判明した．

1. はじめに

インメモリ・メニーコアを活用した高性能トランザクション技術として，Silo [1] を始め，様々な手法が提案されている．複数の手法が検証可能なベンチマークとして DBx1000 [2]，CCBench [3] がある．障害からのリカバリのため，インメモリ DB はロギングが不可欠である．ロギングの性能比較が可能なベンチマーク実装を我々は計画している．一方，トランザクションの性能は，ログを保存するストレージの性能にも依存する．高速かつ低遅延な永続化メモリとして期待されているのが，Intel Optane Data Center Persistent Memory Module (DCPMM) である．本稿では，DCPMM を用いたロギング性能の初期調査について報告する．

2. Silo ロギング法の実装

Silo[1]，SiloR [4] に記述されている Silo ロギング法には，エポックベース，value-logging，redo のみ行い undo は不要，という特徴がある．Silo ではロガースレッドをストレージの数に応じて複数起動し，各ロガースレッドは複数のワーカースレッドからログを受け取る．ロガーは，担当ワーカーのエポック及び未永続化ログのエポック e_l をチェックし，$\min(e_l) - 1$ 以下のトランザクションについては永続化が完了しているとして，クライアントに完了通知を行う．ロギングに設定が必要なパラメータには，ロガー数，ログバッファサイズ，ログバッファ数の 3 つがある．これらのパラメータにより Silo の性能は変動する．本研究ではこれらのパラメタと性能の関係を調査する．

今回の実装は，CCBench [3] を拡張する形で行った．Silo に対するロギングのみ実装し，チェクポイント，リカバリは実装していない．ログ書き出しにはシステムコールの open, write, fsync, close を用いた．

[1] 慶應義塾大学
Keio University
[a)] masa16.tanaka@keio.jp

表 1　使用マシン

CPU	Intel Xeon Platinum 8276 2.20GHz	
	1 チップの物理コア数	28
	ソケット数/NUMA ノード数	4/8
	全物理/論理コア数	112/224
	L1/L2/L3 キャッシュ	32 KiB/1 MiB/38.5 MiB
DRAM	512 GiB	
DCPMM	128 GiB × 8 モジュール	
SSD	440 GiB	

3. 測定環境

測定に用いたマシンは表 1 の通りである．DCPMM の使用にあたり，今回は File System DAX (fsdax) の設定で行った．

性能のためにはワーカースレッドとロガースレッドは同一の NUMA ノードに属する必要がある．今回の実装では，各ワーカースレッドとロガースレッドが動作する論理コアの ID をコマンドラインオプションで指定する．

```
silo.exe -affinity 0:1,2,3+4:5,6,10
```

この例では，最初のロガースレッドを論理コア`#0` に割当て，論理コア`#1`, `#2`, `#3` に割り当てられたワーカースレッドからログを受け取る．`+`以降は次のロガースレッドである．最初の (0 番目の) ロガースレッド`#0` は，実行ディレクトリ内の `log0` というディレクトリの下にログを保存する．

DIMM スロットに装着する DCPMM は，CPU との affinity が性能に大きく影響する．ロガースレッドの CPU とログ保存先の DCPMM を同一の NUMA ノードにする設定は次のように行う．まず，NUMA ノード`#0` に属する DCPMM モジュール `/dev/pmem0p1` に作成したファイルシステム内のディレクトリに `log0` という名前でシンボリックリンクを張っておく．次に，`-affinity` オプションによりロガースレッド`#0` に対して NUMA ノード`#0` に属する論理コア ID を指定してベンチマークを実行する．

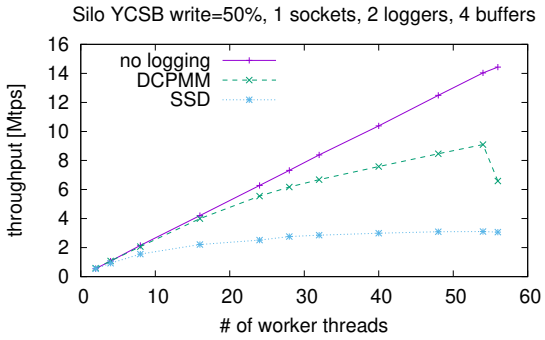

図 1 SSD と DCPMM にログ出力した場合の性能．(Silo YCSB write=50%，1 ソケット，2 ロガー，4 ログバッファ/スレッド)

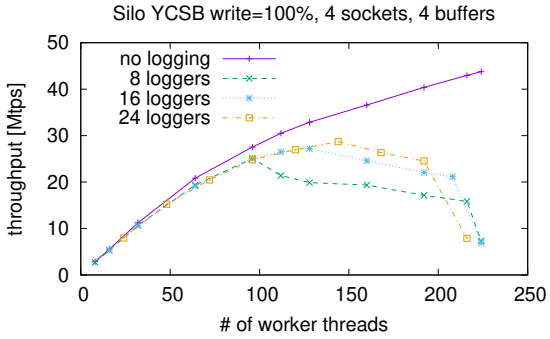

図 2 ロガースレッド数によるトランザクション性能．(Silo YCSB write=100%，4 ソケット，4 ログバッファ/スレッド)

4. 測定結果

今回の測定のワークロードは YCSB である．共通するパラメータは CCBench オプションで，トランザクション中命令数 10，RMW 命令無し，レコード数 100 万，Skew 0 である．同じ測定を 5 回行い，中央値を採用した．

図 1 に 1 ソケットのみを使用した r:w=50%:50% のトランザクションのスループットを SSD と DCPMM で測定した結果を示す．この結果から，ロギングによってトランザクション性能が低下しており，その低下の割合は SSD より DCPMM のほうが小さいことがわかる．DCPMM で 56 コアでのスループットが下がるのは，トランザクションが全コアを占め，ロガースレッドと干渉するためである．

図 2 に全 4 ソケットを使用して測定した write only のトランザクション性能を示す．全ロガースレッド数は 8, 16, 24 (DCPMM モジュール毎に 1, 2, 3) とした．この結果から，224 コアを用いたトランザクション性能を活かすには，今回の手法では不十分であることがわかる．

Silo ロギング法におけるログバッファ数の性能への影響を調査するため，ワーカー毎のログバッファ数を 2 から 32 まで変えて測定した．スループットを図 3 に，永続化遅延を図 4 に示す．バッファ数を増やすとスループット，遅延ともに悪化しており，バッファ数は 2 程度が最良だと

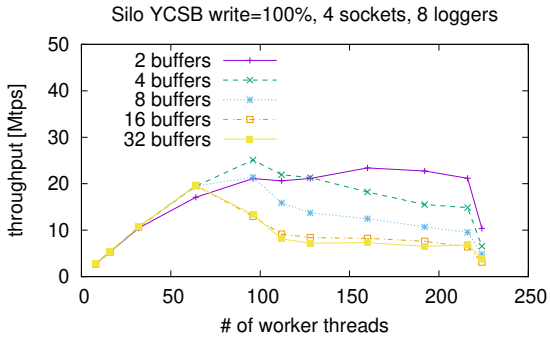

図 3 ログバッファ数によるトランザクション性能 (Silo YCSB write=100%，4 ソケット，8 ロガー)

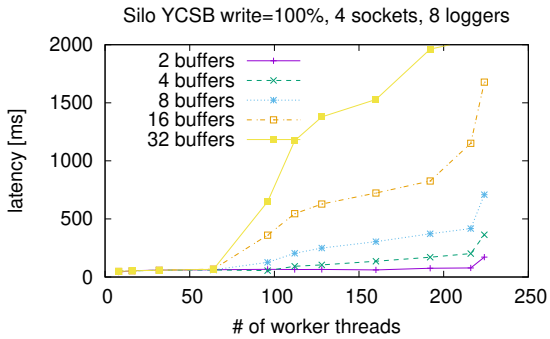

図 4 ログバッファ数による永続化遅延．(Silo YCSB write=100%，4 ソケット，8 ロガー)

言える．ログバッファが増えると遅延が増加する原因として，キューに滞在する時間の増加が考えられるが，トランザクションのスループットに与える影響については調査中である．

5. まとめ

Silo ロギング法を CCBench を拡張して実装し，DCPMM を用いて評価した結果，その性能はパラメタに大きく依存することが判明した．

謝辞 本研究は，NEDO「実社会の事象をリアルタイム処理可能な次世代データ処理基盤技術の研究開発」の支援により行った．

参考文献

[1] Tu, S., Zheng, W., Kohler, E., Liskov, B. and Madden, S.: Speedy transactions in multicore in-memory databases, *SOSP* (2013).
[2] Yu, X.: DBx1000, https://github.com/yxymit/DBx1000.
[3] Tanabe, T., Hoshino, T., Kawashima, H. and Tatebe, O.: An Analysis of Concurrency Control Protocols for In-Memory Databases with CCBench, *PVLDB* (2020).
[4] Zheng, W., Tu, S., Kohler, E. and Liskov, B.: Fast databases with fast durability and recovery through multicore parallelism, *OSDI* (2014).