オンライントランザクション処理における高速フラッシュストレージの性能活用に関する実験的考察

早水 悠登 †　　合田 和生 ‡　　中野 美由紀 ‡　　喜連川 優 ‡
† 東京大学大学院情報理工学系研究科電子情報学専攻
‡ 東京大学生産技術研究所

1　はじめに

フラッシュストレージは半導体素子を記録媒体とする二次記憶装置の一種である。HDD と比べるとランダムアクセス性能が高く、機械駆動部品を持たないために耐衝撃性が高く、消費電力が低いという優位性を持つ。これらの性質のために、フラッシュストレージはこれまで主に携帯端末や家電製品等の組み込みデバイスにおいて用いられてきた。加えて、近年ではエンタープライズサーバのストレージとして広がりつつあり、フラッシュストレージの主要な市場となっている [1]。

エンタープライズシステムの中でも、フラッシュストレージが活用されている主な用途の一つがデータベースシステムである。データベースシステムは一般に高い IO 性能が求められる。特にオンライントランザクション処理 (OLTP) のように、多数のユーザからのトランザクション要求にできるだけ早く応答することが求められるアプリケーションにおいては、HDD よりも 10 倍以上アクセス遅延の短い [2] フラッシュストレージは有効なソリューションとなりうる。

フラッシュストレージを OLTP システムにおいて活用する際に注意するべき点は、その読み込み性能と書き込み性能の非対称性である。フラッシュストレージにおける書き込みは、対象データの含まれる 128〜256KB 単位のブロックで行われるため、これより細かい単位での頻繁な書き込みは非効率である。この点に着目した手法としては、ログ構造化手法 [3]、DB ページにログを同梱して書き込み回数を減らす In-Page Logging[4]、更新データを差分として記録する Page-Differential Logging[5] などの取り組みが行われている。

それに加えて、フラッシュストレージの性能を最大限に活用するためには、その高速性を考慮したソフトウェアアーキテクチャを改めて考える必要がある。現在のデータベースシステムは、主記憶に比べて二次記憶が極めて低速であり、二次 IO がボトルネックであるシステムに最適化されている。エンタープライズ向けフラッシュストレージの登場によって二次記憶が大幅に高速化された現在、必ずしも現在のデータベースシステムが十分にその性能を発揮するとは限らない。フラッシュストレージの更なる高速化や、相変化メモリや強磁性体メモリなど次世代の不揮発性デバイスの実用化を考慮すると、従前のデータベースシステムの構成を見直すことは、新たなストレージの性能を十分に活用するための重要な課題である。このような観点からの研究としては、フラッシュストレージをバッファプールの延長として利用する手法 [6] などがあるが、いまだ十分な取り組みが行われていない。

本研究では、フラッシュストレージが有する性能をデータベースシステムが十分に活用することができているか、OLTP システムを対象として実際の測定をもとに考察を行った。測定対象としては、エンタープライズクラス HDD、SSD に加えて、PCI-Express 接続フラッシュストレージ (以降 PCIe フラッシュと省略) の三種類を用いた。

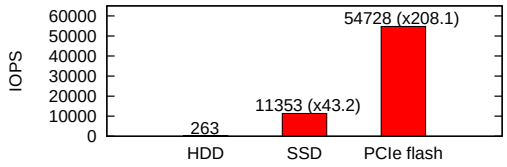

図 1: 16KB ランダム読み込みの IOPS 性能と、HDD に対する性能比

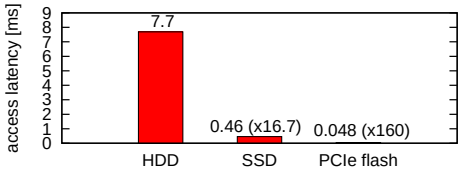

図 2: 16KB ランダム読み込みのアクセス遅延と、HDD に対する性能比

2　フラッシュストレージとデータベースシステム

データベースシステムにおけるストレージ性能の指標として重要なのが 1 秒あたりに処理可能な IO リクエスト数 (IOPS)、及び IO リクエストの発行から完了までにかかる時間 (アクセス遅延) である。本実験で用いた HDD、SSD、PCIe フラッシュ(詳細は 3 節を参照) の 16KB ランダム読み込みに関する IOPS 性能を図 1 に、アクセス遅延を図 2 に示す。これらのグラフから HDD に対して SSD は 43.2 倍、PCIe フラッシュは 208.1 倍の IOPS 性能を有し、また SSD は 16.7 倍、PCIe フラッシュは 160 倍アクセスが短いことがわかる。

従前のデータベースシステムは、主記憶に比べて二次記憶が極めて低速である記憶階層を有するシステムに対して最適化されてきた。即ち、CPU の命令実行コストやメモリアクセスコストは二次記憶への IO コストに比べると無視できる程度であるため、可能な限り二次記憶への IO を削減することが高性能化において最重要であった。例えば現在のプロセッサは 1 サイクルが 1 ナノ秒以下、また主記憶のアクセス遅延は 60 ナノ秒程度 [2] であるので、HDD のアクセス遅延は主記憶のそれに比べて約 100,000 倍程度低速である。

しかしながら、図 1,2 に示したように HDD に比べて SSD は 10 倍以上、PCIe フラッシュは 100 倍以上高速である。このように HDD と比べて高速な二次記憶を用いた場合、CPU の命令実行や、メモリアクセス、二次記憶のアクセスといった個々の要因がデータベースシステムの性能に寄与する割合が従前とは変化していることが予想される。すなわち、PCIe フラッシュのように高速なフラッシュストレージを二次記憶として用いた場合には CPU の命令実行やメモリアクセス等が部分的に律速要因となり、フラッシュストレージの有する IO 性能を十分に活用しきれない可能性がある。本研究では、現在普及しているフラッシュストレージを二次記憶として用いた場合に、データベースシステムの律速要因は二次記憶の

An Experimental Study on Exploitation of Fast Flash Storage for Online Transaction Processing

Yuto HAYAMIZU† Kazuo GODA‡ Miyuki NAKANO‡ Masaru KITSUREGAWA‡
†Institute of Industrial Science

みであるのか、十分にフラッシュストレージの有する IO 性能が活用されているのかを調べるため、OLTP をワークロードとして性能評価実験を行った。

3 実験環境

データベースサーバには HP DL580G7(Intel Xeon X7560 2.27GHz×4、PC3-10600 4GB DIMM×16) を用いた。また、評価対象のストレージは HDD として Toshiba MBF2600RC 6Gbps SAS HDD、SSD として Samsung SS805 3Gbps SATA SLC SSD、PCIe フラッシュとして Fusion-IO ioDrive Duo 320GB SLC の 3 つを用いた。PCI-Express で接続された HP Smart Array P410i Controller を介して HDD と SSD は接続されており、PCIe フラッシュは PCI-Express に直接接続されている。

データベースサーバの OS としては Linux 2.6.32 を用いた。データベース管理システムには MySQL 5.1.41、ストレージエンジンとしては InnoDB plugin を利用した。十分に IO が発生する状況で実験を行うために、InnoDB のバッファプールサイズは 512MB とした。各ストレージを用いる際には、デバイスファイルを RAW デバイスとして指定し、Direct I/O を用いることでファイルシステムのページキャッシュを迂回するよう設定した。

4 OLTP におけるフラッシュストレージ活用の評価

この評価実験では、HDD、SSD、PCIe フラッシュのそれぞれを OLTP システムの二次記憶とした場合に、下記の 2 項目について測定を行うことで、ストレージの性能がどれだけ活用されているかを評価した。

1. ストレージ性能が 100%活用された場合の IOPS
2. OLTP ワークロード実行時の IOPS

OLTP ワークロードとしては、標準的ベンチマークである TPC-C を利用した。TPC-C のデータベースサイズの尺度である warehouse 数は 200(初期化直後のデータサイズは約 20GB) とした。また TPC-C のターミナル数は 200 とし、各 warehouse に一つずつターミナルを割り当てた。TPC-C のその他のパラメータは version 5.11 に準拠するよう設定を行った。

1. の「OLTP ワークロード下でストレージ性能が 100%活用された場合の IOPS」の測定は、TPC-C の IO トレース再生を行うことで行った。IO トレース再生によって、データベースシステムにおける IO 以外のコストが全て存在しない状況を作り出すことにより、OLTP ワークロードに対する 100%のストレージ性能を測定することができる。測定に用いた IO トレースは、TPC-C 実行開始から 3600 秒経過以降の 300 秒間に採取した。

2. の「OLTP ワークロード実行時の IOPS」の測定は、前述の設定で TPC-C を 7200 秒間実行し、3600 秒経過後から 300 秒間の IOPS を測定した。

IO トレース再生、TPC-C 実行の測定は HDD、SSD、PCIe フラッシュのそれぞれに対して 1 回ずつ行った。いずれの測定においても、IOPS の測定には `iostat` コマンドを利用し、1 秒間隔で計 300 回記録した値の平均値をその測定における IOPS の値とした。

図 3,4 に実験結果を示す。図 3 では、HDD、SSD、PCIe フラッシュのそれぞれについて、下記のように定義される TPC-C 実行時ストレージ性能活用率 α を比較している。

$$\alpha = \frac{\text{TPC-C 実行時の IOPS}}{\text{TPC-C の IO トレース再生時 IOPS}}$$

また図 4 では、TPC-C のトランザクションスループットを表す MQTh(Maximum Qualified Throughput) の値を比較して

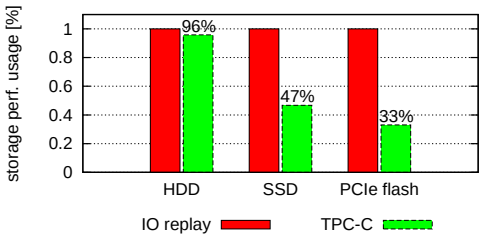

図 3: IO トレース再生時と TPC-C 実行時のストレージ性能活用率比較

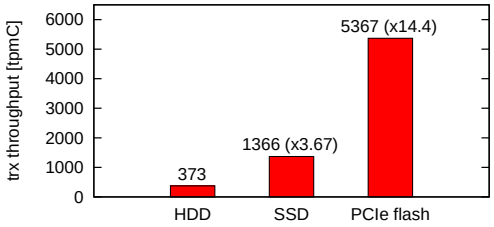

図 4: TPC-C のトランザクションスループットと、HDD に対する性能比

いる。これらのグラフより、HDD を二次記憶として用いた場合にはその性能の 95%が活用される一方、SSD では 47%、PCIe フラッシュでは 33%しか性能を活用できていないことがわかる。また、トランザクションスループットの HDD に対する性能向上率は SSD では 3.67 倍、PCIe フラッシュでは 14.4 倍に留まっている。ストレージの本来有する IOPS 性能の差としては、HDD に対して SSD は 43.2 倍、PCIe フラッシュは 208.1 倍であることを考えると、データベースシステムがフラッシュストレージの性能を十分に活用できていないことがわかる。

5 まとめ

本研究では、HDD に比べて大幅に高速化したフラッシュストレージが有する IO 性能をデータベースシステムが十分に活用することができているか、OLTP ワークロードのもとで実測により評価した。その結果、SSD を用いた場合にはその IO 性能の 47%、PCIe フラッシュでは 33%の IO 性能が活用されるに留まっていることが確認された。今後は更に性能解析を進め、フラッシュストレージの性能を十分に活用するための方法論の検討を進めたい。

References

[1] Jeff Janukowicz and David Reinsel. Worldwide solid state storage 2011-2015 forecast update. Technical report, IDC, December 2011.

[2] Richard Freitas. Storage class memory: Technology, systems and applications, August 2010. Hot Chips 22 Tutorial.

[3] Yi-Reun Kim, Kyu-Young Whang, and Il-Yeol Song. Page-differential logging: an efficient and dbms-independent approach for storing data into flash memory. In *Proceedings of the 2010 international conference on Management of data*, SIGMOD '10, pp. 363–374, New York, NY, USA, 2010. ACM.

[4] Sang-Won Lee and Bongki Moon. Design of flash-based dbms: an in-page logging approach. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pp. 55–66, New York, NY, USA, 2007. ACM.

[5] Gap-Joo Na, Bongki Moon, and Sang-Won Lee. In-page logging b-tree for flash memory. In *Proceedings of the 14th International Conference on Database Systems for Advanced Applications*, DASFAA '09, pp. 755–758, Berlin, Heidelberg, 2009. Springer-Verlag.

[6] Mendel Rosenblum and John K. Ousterhout. The design and implementation of a log-structured file system. *ACM Trans. Comput. Syst.*, Vol. 10, pp. 26–52, February 1992.