3N-1

複数問い合わせ処理のワークロードに着目した SSDを用いたデータベースの最適化

鈴木 恵介 †　早水 悠登 †　横山 大作 †　中野 美由紀 ††　喜連川 優 †,†††

† 東京大学　†† 芝浦工業大学　††† 国立情報学研究所

1 はじめに

SSDは、HDDと比較して高いI/Oスループットをもつストレージデバイスとして注目を集め、データセンターなど、大規模データ処理が必要とされる現場において導入が進んでいる。データベースシステム(DBMS)においても、大規模データを扱う問い合わせ処理では、ストレージの転送レートがボトルネックになる場合が多く、SSDを使用する利点は大きい。最近のSSDは、内部でFlashチップが複数接続した構造をもち、並列I/O処理によってスループットが増加する[1]特徴がある。本稿では、SSDを用いたDBMSでは、複数問い合わせの同時処理において、並列I/O処理能力を利用し、さらにキャッシュを分割して使用することで全体の処理を高速化できることを明らかにする。データベース演算の中でも、大規模データ処理で多用され、負荷の高いハッシュ結合を中心として扱った。

2 関連研究

SSD上での結合演算の研究としては、列指向のデータ配置を利用し、I/Oの削減を実現したFlash join[2]や、複数の結合演算アルゴリズムの性能を比較し、SSD上において有利になるアルゴリズムについて分析した[3]などが挙げられる。本稿では、複数問い合わせの同時処理によってシステムの処理性能を向上する方法について論じている。[3]で触れられている、結合演算におけるキャッシュ利用効率が全体の処理性能に及ぼす影響についても分析している。

3 ハッシュ結合の処理コスト

HDDの使用を前提としたDBMSではハイブリッドハッシュ結合が多用されており、本稿でも、ハッシュ結合といえばこれを指すものとする。

ハッシュ結合のI/Oパターンは、ワーキングメモリサイズによって決まる。ワーキングメモリが小さいときは、多数の小さなパーティションファイルが作られることになり、フラグメンテーションが発生するため、アクセス時に大量のランダムI/Oを伴う。ランダムI/Oが低速なHDDでは、I/Oスループットが大きく減少するため、HDD上でのハッシュ結合では、大きなワーキングメモリが必要とされる。一方でSSDにおいては、HDDと比較してランダムI/Oが高速であるため、ワーキングメモリサイズをHDD使用時より小さくできる。

ワーキングメモリサイズは、キャッシュミス数にも関係する。ワーキングメモリサイズをキャッシュサイズ以下に定めることで、ハッシュテーブルをキャッシュに収め、キャッシュミス数を低減できる。SSDを使用したハッシュ結合では、I/Oスループットを低下することなくワーキングメモリサイズを縮小できるので、キャッシュミス数減少による性能向上も見込むことができる。さらに、ワーキングメモリサイズを小さくすることで、キャッシュにも空きスペースが生じるので、これを他の処理で使用できるようになる。

表1: 計測に使用した計算機環境

CPU	Xeon X7560 (L3 cache: 24MB) @ 2.27GHz x 4
DRAM	64GB
Storage (SSD)	ioDrive Duo x4 (8 Logical units, Software RAID0)
kernel	linux-2.6.32-220
File system	ext4
DBMS	PostgreSQL 9.2.4 (Shared buffer = 8GB)

4 複数問い合わせの同時処理

SSDは、並列I/Oによりスループットが増加するため、複数問い合わせ処理によって同時に発行される複数I/Oを高速に処理できる。本節では、(1)ハッシュ結合を行う問い合わせの同時処理と、(2)ハッシュ結合を行う問い合わせとスキャンを行う問い合わせの同時処理の2つのワークロードについて全体処理時間を計測し、処理性能の向上について分析する。

表1に計測環境を示す。SSD上にソフトウェアRAID0(チャンクサイズ = 64kB)を構築し、ext4ファイルシステムを用いる。また、I/Oスケジューラには、noopを適用した。問い合わせ実行時の、CPU使用率の内訳についてはmpstat(1)を用いて取得した。

4.1 ハッシュ結合の複数同時処理

TPC-H Scale factor=10のデータベースを複数用意し、それぞれでlineitem表とpart表をハッシュ結合する問い合わせを同時実行し、計測した。ワーキングメモリサイズ(work_mem)を64kB-256MB、同時処理数1-16

Exploiting SSD-based Database on Multiple Queries Processing
†Keisuke Suzuki　†Yuto Hayamizu　†Daisaku Yokoyama
††Miyuki Nakano　†,†††Masaru Kitsuregawa
†University of Tokyo
††Shibaura Institute of technology
†††National Institute of Informatics

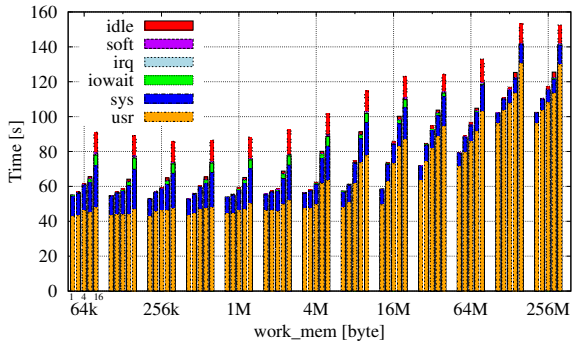

図 1: 結合演算を行う問い合わせ複数同時処理の実行時間

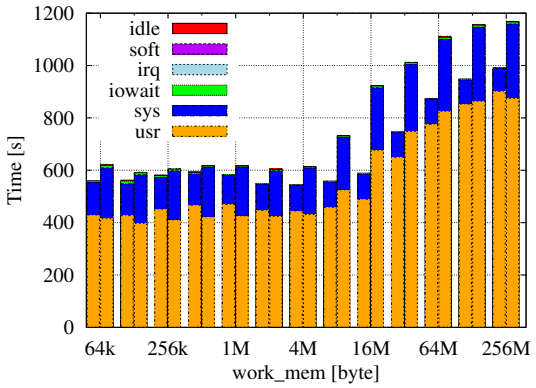

図 2: 結合演算を行う問い合わせの実行時間 (左:結合演算単体、右:スキャン同時実行時)

で変化させ計測した結果を図 1 に示す。図中では、各 work_mem において左から同時処理数 1、2、4、8、16 のときの結果を示しており、usr が CPU コスト、sys と iowait、idle の合計が I/O コストを示す。

work_mem ≤ 1MB の領域では、ハッシュテーブルが全て L3 キャッシュに収まっており、キャッシュミス数が小さく抑えられている。I/O コストについては、同時実行数 1-8 ではほぼ増加しておらず、並列 I/O 処理によってスループットが増加しており、16 でスループットが飽和している。全体では、単位時間当たりの問い合わせ処理数で、同時実行数 16 のときで逐次の場合の 13 倍の処理性能向上を示している。

4.2 ハッシュ結合とスキャンの同時処理

TPC-H Scale factor=100 のデータベースで lineitem 表と part 表をハッシュ結合する問い合わせを実行し、同時に他のデータベースでテーブルの順次スキャンを繰り返した時の処理性能を計測した結果を図 2 に示す。図中では、各 work_mem において結合演算を行う問い合わせのみを実行した場合の結果を併載している。

スキャンの同時実行により、work_mem ≥ $8MB$ と結合演算単体での実行時より小さい値からキャッシュミス数の増加による CPU コストの増加が見られる。全体の処理性能の向上に関しては、例として work_mem = 2MB の場合を挙げると、まず順次スキャンしたデータ量は 369GB であった。順次スキャン単体実行時のスループットは 750MB/s であったので、ハッシュ結合とスキャンを逐次実行すると $578[s] + (369[GB]/750[MB/s]) = 1070[s]$ となり、同時実行により 1.7 倍処理性能が向上している。

5 考察

前節の計測結果を参照すると、キャッシュミス数の増加は全体の処理性能に大きな影響を与えており、同時処理時は互いのキャッシュの干渉を抑えることが重要である。また、SSD ではフラグメンテーションや読み書き I/O の混在によってスループットが低下する [1] ことが知られている。同時処理による I/O スループットの低下を避けるためには、個々の問い合わせの I/O ワークロードを把握し、SSD の帯域や内部アーキテクチャにあわせて制御を行う必要がある。

6 おわりに

本稿では、SSD を用いた RDBMS において、複数問い合わせの同時処理によって、システム全体の処理性能が向上することを示した。SSD 上のハッシュ結合では、ワーキングメモリを小さくすることで、キャッシュミス数を低減し、さらにキャッシュを他の処理と共有することが可能であった。また、SSD の並列 I/O 処理能力により、I/O スループットが増加し、処理性能向上につながった。今後の研究課題としては、より複雑なワークロードの同時処理において、キャッシュの干渉を抑え、SSD の帯域や内部アーキテクチャを考慮し、I/O スループットの低下を防ぐ、同時処理数制御などが考えられる。

参考文献

[1] Feng Chen, David A. Koufaty, and Xiaodong Zhang. Understanding Intrinsic Characteristics and System Implications of Flash Memory Based Solid State Drives. *SIGMETRICS Perform. Eval. Rev.*, Vol. 37, No. 1, pp. 181–192, June 2009.

[2] Dimitris Tsirogiannis, Stavros Harizopoulos, Mehul A. Shah, Janet L. Wiener, and Goetz Graefe. Query Processing Techniques for Solid State Drives. In *Proceedings of the 2009 ACM SIGMOD*, pp. 59–72, New York, NY, USA, 2009. ACM.

[3] Jaeyoung Do and Jignesh M. Patel. Join Processing for Flash SSDs: Remembering Past Lessons. In *Proceedings of the Fifth International Workshop on Data Management on New Hardware*, DaMoN '09, pp. 1–8, New York, NY, USA, 2009. ACM.