

アプリケーション指向ディスクドライブ省電力方式の一考察 : OLTP 系 DBMS の I/O 挙動特性に基づくディスクドライブ省電力の効果

西川 記史^{††}, 中野 美由紀[†], 喜連川 優[†]

† 東京大学 生産技術研究所 ‡ (株) 日立製作所 システム開発研究所

1. はじめに

サーバやストレージの集約によるデータセンタの高密度化に伴い、データセンタの消費電力は増加の一途を辿っている。中でも、データセンタが管理するデータ量の急増に伴うストレージの消費電力の増加は著しいものがある。デジタルコンテンツの容量は 2011 年には 2007 年の 6 倍の約 1773 エクサバイトに急増し、さらにその 85% が企業などの何らかの組織などのデータセンタによって管理されると予測されている [1]。データセンタにおけるストレージの電力消費量は急激に高まるものと考えられ、その消費電力の削減はデータセンタにおいて喫緊に解決しなければならない重要な課題である [2]。

本論文では、OLTP 系アプリケーションが用いるストレージの消費電力手法の確立を目指し、複数台のディスクからなる小規模環境において、アプリケーションの知識を用いることにより OLTP 系アプリケーションが用いるディスクの消費電力削減の可能性について示す。

2. OLTP 系 DBMS の I/O 挙動特性

OLTP 系アプリケーションの I/O 特性の活用方式を検討するために、我々は TPC-C ベンチマークの簡易実装である tpcc-mysql [2] を用い、DB バッファが DB サイズと比較して大きい環境下での OLTP 系アプリケーションの I/O 特性の調査を行った。この結果、DB バッファが DB サイズと比較して大きい環境下での I/O は、ログへの write が支配的であり、DB データに対する I/O はほとんど行われていないこと、及び DB データ部分においては OrderLine データを除き IOPS が 0.5 以下であり、かつ write が大半であることが分かった。これは、メモリが十分にある環境下では、ディスクレベルであっても消費電力の削減を行う余地があることを示していると言える。

A Study on Application-oriented Disk Drive Power Reduction: Power Saving Efficiency of Disk Drives based on I/O Behavior of OLTP Application

Norifumi NISHIKAWA^{††}, Miyuki NAKANO[†], and Masaru, KITSUREGAWA[†]

[†] University of Tokyo Institute of Industrial Science

^{††} Systems Development Laboratory, Hitachi, Ltd.

3. TPC-C の I/O 挙動特性を用いた省電力方式

我々は前述の I/O 挙動特性、及び DBMS の表及び索引データへの write 動作に着目した省電力方式を提案する。提案方式は、(i) ディスクへのデータ配置を、DB データ(ログ、表・索引)の I/O 頻度に基づき片寄せすることによる非ビジーディスクの生成、及び(ii) 表・索引データへの write を同一ディスク上のデータへの read が行われるまで遅延する I/O 発行制御、の 2 方式である。

後者は、DB バッファサイズが DB サイズと比較して大きい場合は、DBMS が DB に対して行う write のタイミングを DBMS が決定できる特性を活かしたものである。read 処理は問合せと同期して実施されるため要求が来た時点で実行する必要があること、及び write 回数をできるだけ減らすとの観点から、I/O 発行制御は以下の方式とした。

Step1. Write 受領時はディスク状態に関わらず write 保留。

Step2. Read 受領時、まず Read を実行。さらに前回 write を実行して以降 5 分以上経過していれば保留された write を実行。

Step3. 保留となった write が存在する状態で、10 分間 I/O 要求がなければ、保留となった write を実行。

4. 提案方式の評価

我々は、ディスク 2 台を用い、表 1 に示す 4 ケースのデータ配置について、データ配置管理を実施した場合の消費電力及びトランザクションスループット、トランザクション当り消費電力の計測を実施した。なお、ディスク電源制御機能は、ディスクに対して 5 秒間 I/O がなければ Standby 状態に移行し、Standby 状態にある時に I/O 要求があると Active 状態に移行する設定とした。

表 1. データ配置

	ディスク2(sdb)	ディスク3(sdc)
ケース1	ログ	Customer, District, History, Item, NewOrder, OrderLine, Orders, Stock, Warehouse
ケース2	ログ, OrderLine	Customer, District, History, Item, NewOrder, Orders, Stock, Warehouse
ケース3	ログ, NewOrder, OrderLine, Orders, Stock	Customer, District, History, Item, Warehouse
ケース4	ログ, Customer, NewOrder, OrderLine, Orders, Stock	District, History, Item, Warehouse

図 1 は、ログと DB データをそれぞれ異なるディスクに配置し、各ケースにおいてディスク電源制御機能を用いない場合の単位時間当りの消費電力を 100 とした場合に、ディスク電源消費機能を用いた場合のケース毎の単位時間当り消費電力を示している。図 1 から分かるように、ケース 2 ではディスク省電力機能を用いたことにより逆に単位時間当りの電力消費量が増加している。これは、ディスク 3 側の I/O 間隔が十分ではなく、ブレークイーブン時間より短い間隔で Active/Idle 状態と Standby 状態を繰り返したためである。その一方ケース 3, 4 においては単位時間当りの消費電力量はそれぞれ-9.2%, -23.1%であった。

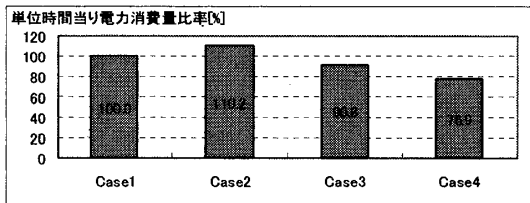


図 1. 単位時間当り消費電力

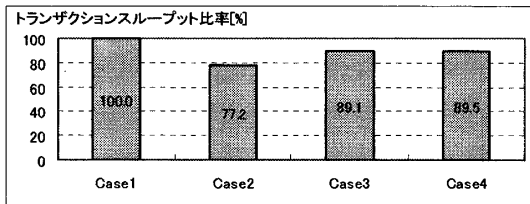


図 2. トランザクションスループット

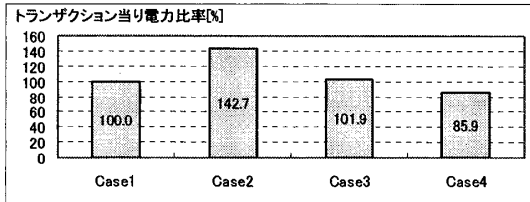


図 3. トランザクション当り消費電力

次に、トランザクションスループットの測定結果を図 2 に示す。ケース 1 を基準スループット(100%)とした場合、ケース 2 が最も悪く、ケース 3, 4 になるに従いスループットは上昇している。但し、省電力機能を使用しなかったケース 1 より減少していた。これはディスクの起動待ちの影響である。図 3 はトランザクション当りの消費電力であり、ケース 4 では省電力機能未使用時より電力の削減が可能となっている。

次に、データ片寄せ及び I/O 発行制御機能を併用した場合の単位時間当り消費電力、トランザクションスループット、トランザクション当り消費電力の関係をシミュレーションにより予測した結果を示す。図 4 に示すとおり、I/O

発行制御機能を用いることにより、省電力機能及び I/O 発行制御機能を用いない場合と比較して、単位時間当りのディスクの消費電力を約 37.9%削減することが可能であるとの結果を得た。また、トランザクションスループットについては、省電力機能及び I/O 発行制御機能を用いることにより、0.7%まで改善できることが示された(図 5)。さらに、トランザクション当り消費電力を 37.5%削減可能であるとの結果を得た(図 6)。

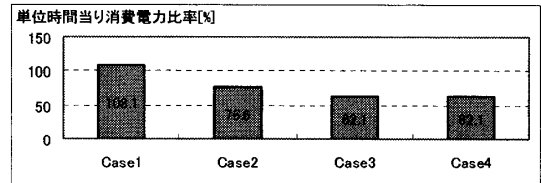


図 4. 単位時間当り消費電力(I/O 発行制御使用時)

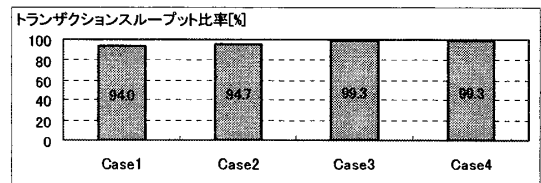


図 5. トランザクションスループット(I/O 発行制御使用時)

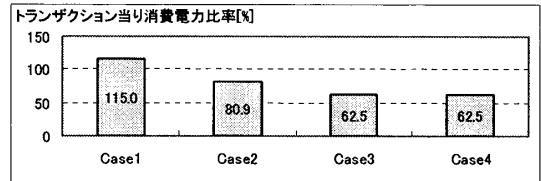


図 6. トランザクション当り消費電力(I/O 発行制御使用時)

5. まとめ

本論文では、OLTP 系 DBMS が持つ知識を用いてディスクの制御とデータ配置の制御を行うことにより、メモリが十分な環境下では小規模構成システムにおいてもディスクの消費電力の大幅な削減が可能になることを示した。

参考文献

- [1] P.B.Chu, E.Riedel, "Green Storage II: Metrics and Measurement," <http://net.educause.edu/ir/library/pdf/churiedel.pdf>, 2008.
- [2] M. Poess and R.O. Nambiar, "Energy cost, the key challenge of today's data centers: a power consumption analysis of TPC-C results", Proc. Int'l. Conf. on Very Large Data Base, pp.1229-1240, 2008.
- [3] tpcc-mysql, <https://code.launchpad.net/~percona-dev/perconatools/tpcc-mysql>.