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
‡株式会社日立製作所 システム開発研究所
E-mail:　†{norifumi, miyuki, kitsure}@tkl.iis.u-tokyo.ac.jp,　‡norifumi.nishikawa.mn@hitachi.com

あらまし　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, データセンタで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く，その電力削減はデータセンタにおける重要な課題となっている．我々は，複数のディスクドライブから構成されるストレージの省電力化を目的に，TPC-C ベンチマーク相当の OLTP 系アプリケーションの I/O 挙動に基づくディスクドライブの省電力化方式の検討及び評価を実施した．本論文では，OLTP 系アプリケーションの I/O 挙動特性を活用することにより，メモリが十分にある状況下ではディスクドライブの消費電力の削減が可能であることを示す.
キーワード　データベース管理システム，ディスクドライブ，省電力，OLTP

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.
E-mail:　†{norifumi, miyuki, kitsure@tkl.iis.u-tokyo.ac.jp,　‡norifumi.nishikawa.mn@hitachi.com

Abstract　Power consumption is increased rapidly in today's datacenters. Storage is the most power consuming unit at a datacenter. Power savings of disk storage become a major problem at datacenters. In this paper, we describe a power saving approach of multiple disk drives used by OLTP applications (TPC-C). Features of our approach are (i) use characteristics of I/O behavior of OLTP application, and (ii) delay writes to database data by using a behavior of I/O of DBMS. We then show experimental and simulated results of our power saving approach. The results show that our power saving approach enables to save power consumption of disk drives substantially under DBMS has enough memory.
Keyword　Database Management System, Hard Disk Drive, Power Saving, OLTP

1. はじめに
1.1. 動　　機

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[1]．中でも，データセンタが管理するデータ量の急増に伴い，ストレージの消費電力が急増することが予想されている[2].

データベース管理システム(DBMS)は, ストレージの主要なアプリケーションである．ハイエンドストレージの DBMS 向け出荷容量は全出荷容量の 6 割以上を占め，その半数以上が ERP や CRM などの Business Processing と呼ばれるオンライントランザクション処理(OLTP)系アプリケーションである[3]．これは，データセンタにおいては OLTP 系アプリケーション用のストレージが主要な電力消費源の一つと考えられることを示している．この消費電力を削減することはデータセンタにおけるストレージ消費電力の削減に大きく貢献するものと考えられる.

一方，ストレージの消費電力の内訳に目を向けると，その約 7 割はディスクドライブに起因するとされている[4]．ディスクドライブの主要な電力消費源はスピンドルモータやボイスコイルなどの機械的部分である[5]．近年のディスクドライブには，スピンドルモータの停止やヘッドの退避などにより機械的部分の消費電力を削減する機構が組み込まれているが，機械的部分の再起動時には通常時の数倍の電力と十数秒の時間を要する[6]．このため，ディスクドライブの消費電力を削減するためには，I/O がいつ発行されるかを精度よく知り，適切な機会に上記の省電力機構を利用するこ

とが重要である.

これに対し，処理が比較的長時間に及ぶ分析系アプリケーションの挙動を解析することにより I/O の発行時期を予測し，これを用いてディスクドライブの省電力化を行う手法が提案されている[7,8,20]．しかし，個々のトランザクションの挙動解析に基づく I/O 予測を用いるこれらの手法をトランザクションの応答時間が 1 秒に満たない OLTP 系アプリケーションに適用することは困難であり，OLTP 系アプリケーションのディスクドライブ省電力化のためには新たな手法の開発が求められている.

1.2. 研究の目的

このような背景の下，我々は，OLTP 系アプリケーションで用いられるディスクドライブを対象とした省電力化方式を提案する．提案方式の特長は，トランザクション処理を多数実行した場合の DBMS のマクロ的な挙動に着目し write 主体環境において Idle 時間の延伸を図る点である．具体的には，OLTP 系アプリケーションの I/O 挙動特性と DBMS の内部挙動特性を活用する.本論文では提案方式と評価結果について述べる.

1.3. 本論文の構成

本論文の構成は以下の通りである．まず，2 章において関連研究を示し，3 章において OLTP 系アプリケーションの I/O 挙動特性を示す．その後 4 章においてディスクドライブの消費電力特性を，5 章において提案方式を，6 章において提案方式の評価を示す．最後に 7 章にて論文のまとめを示す.

2. 関連研究

ストレージの省電力手法は，大きくディスクドライブの回転制御，I/O 発行間隔の制御，及びデータ配置制御に分けることができる．本章では，まずこれらの手法について概観した後，我々のアプローチと関連の深いアプリケーション連携による省電力化手法について述べる.

2.1. ディスクドライブ制御

本手法は，主にディスクに対して何らかの制御を実施する手法である．従来，モバイル機器を中心に一定時間ディスクのアイドル状態が続くとディスクをスタンバイ状態等の省エネルギーモードに移行することが行われている．これに対し，ディスクを停止するまでの時間を動的に変更することによりアイドル時間を短くし電力消費の低減を図る手法[9,10],ディスクは回転数が低い方が低消費電力であることに着目し，単にディスクを省エネルギーモードに移行するのみではなく複数の回転数で動作させることにより省エネルギー化を図る手法[11,12]が提案されている.

2.2. I/O 発行間隔制御

本手法は，ディスクを省エネルギーモードで動作する機会をなるべく多くするようディスクへの I/O の発行を制御するものである．本手法の特長は，キャッシュなどの階層記憶構造を有効に使うことによる Idle 時間の延伸である[13-15].

2.3. データレイアウトの制御

本手法は，データのレイアウト（配置）を調整することにより省エネルギー化を図るものである．その基本的な発想はアクセス頻度の高いデータを少数のディスクに集中させ，他のディスクをスタンバイ状態とすることにより省電力化を図るものである[16-19].

2.4. アプリケーション連携省電力

本手法は，アプリケーションの知識を用いて I/O 発行時期や発行先ディスクドライブを知り，その情報を用いて前述の各制御を行うことを特長とする．これにより，前述のディスクストレージ内部のみで得られる情報に基づき I/O を予測する手法と比較して高い省電力効果を得ることが期待できる.

代表的な研究としては，プログラムカウンタを用いて I/O の時期を予測しディスクの制御を行う手法[20],ディスクドライブの Idle 時間を増加させるようアプリケーションを変形する手法[21]がある．また，よりアプリケーションを意識した手法として，科学技術計算向けアプリケーションのソースコードを解析し Idle 時間が延びるようループ処理を変形すると共にデータを再配置する手法[22]，DBMS のクエリプランを利用して DBMS が行う I/O の時期や I/O 先を知り，これを用いてディスクドライブの回転制御を行う手法[7,8]が提案されている.

本論文では，OLTP 系 DBMS で用いられるディスクドライブの省電力化を目的としている．本論文で提案するアプローチは，個々の問合せの特性を活用するのではなく，問合せを多数実行した場合の DBMS のマクロ的な挙動を活用する点に特長がある．これにより，個々のトランザクションの所要時間がディスクドライブの省電力制御に必要な時間より短い場合における省電力化を狙う.

3. ディスクドライブの消費電力特性

まず我々は，ディスクドライブの電力特性の把握を目的として，ディスクドライブの消費電力特性の計測を行った.

3.1. 消費電力特性の計測環境

図 1 は，ディスクドライブの消費電力の計測のために使用した機器構成の概略図である.負荷生成 PC は，計測対象ディスクドライブに対し，4 ピンの電源ケーブルを介して電力供給を行う.赤は 5V, 黄色は 12V で

ある．赤及び黄色の線をそれぞれデジタル電力計(YOKOGAWA 製 WT1600)に通して電流を計測し，さらに赤線と黒線(GND)，及び黄線と黒線間にクリップをはさみ電圧を計測する．ディスクドライブの消費電力は，これら両者の電力の合計値である．

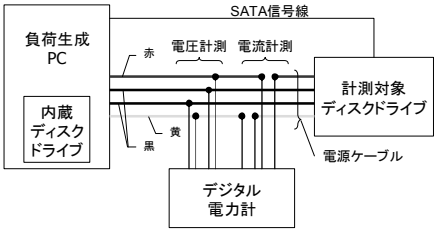

図 1. 実験機器構成

負荷生成 PC の CPU は AMD Athlon 64 FX-74 3GHz, cache 1MB, 4 コア×2，主記憶は 8GB である．計測対象ディスクドライブは Seagate 社の Barracuda ES ST3750640NS(750GB, 7200rpm)である．また，計測時はディスクドライブの write キャッシュを無効化している．これは，DBMS では信頼性の観点から通常ディスクドライブの write キャッシュを使用しないためである．

3.2. ディスクドライブの電力状態

本研究において用いたディスクドライブの電力状態は以下の通りである．

Active: ディスクドライブに対して I/O が行われている状態であり，消費電力が最も高い．

Idle: ディスクドライブに対する I/O は行われていないが，即座に I/O を実行できる状態である．

Standby: ヘッドを退避するとともにディスクドライブの回転を停止した状態であり，ディスクを回転させるための電力を削減している．キャッシュの使用は可能である．

Sleep: ヘッドの退避，回転の停止とともに，キャッシュへの電力供給も停止している状態である．消費電力は Standby と同じである．

これらの状態のうち本研究では Active, Idle, Standby の 3 状態を用いた．Sleep 状態の GAIN は Standby と同等であるにも関わらず Sleep 状態から他の状態への遷移にはディスクドライブのリセットが必要であり，Standby 状態で代用可能と判断したためである．

3.3. Active/Idle 時の消費電力

Active/Idle 時の消費電力と秒当りの I/O 数(IOPS)の計測結果を図 2 に示す．I/O サイズは 16KB である．図 2 に示すように，Random I/O では IOPS が増加するに従い消費電力量は増加傾向にあるが，消費電力の伸びは小さくなっている．また，Sequential Write では Random I/O と異なりヘッドの移動がほとんどないため，IOPS が増加しても消費電力は Random I/O ほどには増加しないことが分かる．なお，図 2 中の近似曲線は 90 IOPS までを示しているが，Random I/O の最大 IOPS は read の比率により異なっており，100% read 時で約 90 IOPS，以降，read の比率が 40%減少する毎に約 10 IOPS ずつ最大 IOPS が減少する結果となった．

また，図 2 では I/O サイズ 16KB についてのみ示しているが，異なる I/O サイズ 4KB, 8KB, 32KB, 64KB, 128KB についてもほぼ同様の結果であった．

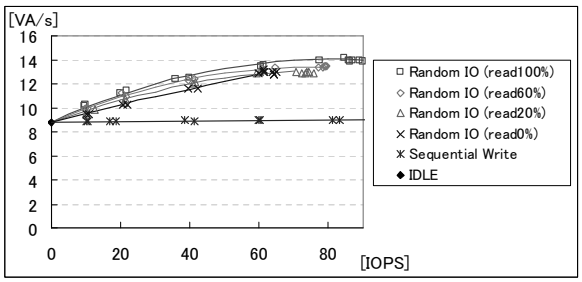

図 2. ディスクドライブの I/O 数と消費電力

3.4. Standby 時の電力とブレークイーブン時間

Standby 中のディスクドライブ消費電力，Active/Idle 状態から Standby の状態への移行時の消費電力，Standby 状態から Active/Idle 状態へ移行時の消費電力，及びブレークイーブン時間は図 3 の通りであった．ブレークイーブン時間とは，Standby 状態に移行した場合と Idle 状態を維持した場合の消費電力が拮抗する時間のことである．

図 3 に示すとおり，Standby 中のディスクドライブの消費電力は約 1.5VA/秒であった．これは Idle 時の 1/5 以下である．その一方で，Standby 状態から Active/Idle 状態への移行には，平均 23.2VA/秒以上の電力消費と 8 秒の時間が必要であった．また，Active/Idle 状態から Standby 状態への移行においても，約 3.5VA/秒の電力消費が 0.2 秒間持続した．上記の数値よりブレークイーブン時間を計算したところ約 23.9 秒であった．つまり，省電力機構を利用するためには，少なくとも 24 秒以上，入出力の停止，延長が必要となる．

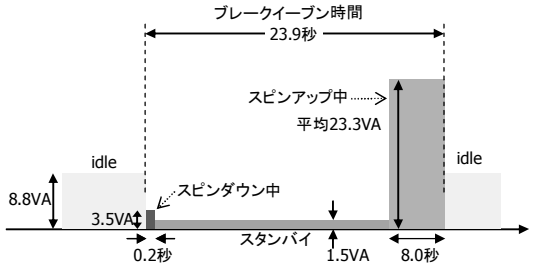

図 3. Standby 中の電力とブレークイーブン時間

4. OLTP 系アプリケーションの I/O 挙動特性

次に我々は，OLTP 系アプリケーションの I/O 挙動特性を活かしたディスクドライブの省電力方式を検討す

るため，OLTP 系アプリケーションの代表的ベンチマークの TPC-C ベンチマーク[23]の簡易実装である tpcc-mysql[24]を用いて I/O 特性の調査を行った．

4.1. I/O 挙動特性の実験環境

I/O 挙動特性の計測に用いた機器は前章と同一のものを用いた．ソフトウェアは，OS に CentOS 5.4 (32 ビット版)，DBMS に MySQL Community Server 5.1.40 Linux 版を，OLTP 系アプリケーションの負荷生成プログラムに TPC-C ベンチマークの簡易実装である tpcc-mysql をそれぞれ用いた．

今回計測に用いた DB の規模は約 1GB(Warehouse 数 10)である．ただし，このサイズにログは含まない．我々は，計測対象のディスクドライブを 10 個のパーティションに均等に分割し，全てに ext2 ファイルシステムを作成した後，そのうちの一つにログファイルを，残りの 9 個に表及び索引を格納したファイルを配置した．パーティションと表・索引を格納したファイルは 1:1 対応となるようにし，OS レベルの統計情報を参照することにより DB のどのデータに対して I/O が行われたかが分かるようにした．また，本計測では DB バッファのサイズを 2GB に設定し，またファイルシステムのキャッシュ及びディスクドライブのキャッシュを無効化した．DB バッファサイズを DB サイズと比較して十分大きく取った理由は，高トランザクションスループットを狙う DBMS を対象としたためである．

計測にあたっては，まず DB を作成し，次に DBMS を再起動して DB バッファをクリアした後アプリケーションを実行した．計測期間はスループットが安定してから 10 分間とした．

4.2. TPC-C における I/O の計測結果

データ毎の Read/Write 別の秒当りの I/O 数及び平均 I/O 発行間隔を図 4 に，Idle 時間の分布を図 5 にそれぞれ示す．

図 4 に示すとおり，DB バッファサイズが DB サイズより大きな環境下では，ログへの write が支配的であること，及び表・索引に対する I/O は最大でも 2 IOPS 程度と非常に少なくかつ write が大半であることが分かる．District, Item, Warehouse 表・索引に対する I/O は観測されなかった．また，IOPS より求めた平均 I/O 発行間隔は NewOrders 表・索引を除きブレークイーブン時間である 23.9 秒より短かった．

しかし，図 5 を見ると Customer, OrderLine, Orders, Stock の各表・索引データの Idle 時間には偏りがあり，ブレークイーブン時間より長い Idle 時間が多数存在することが分かる．これは，OLTP 系アプリケーションであっても DB バッファサイズが DB サイズより大きい環境下ではディスクドライブの省電力化が可能であることを示している．

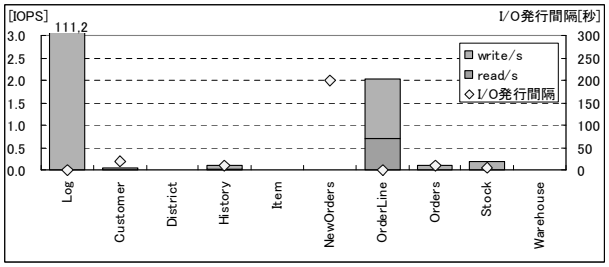

図 4. TPC-C のデータ毎の IOPS と平均 I/O 発行間隔

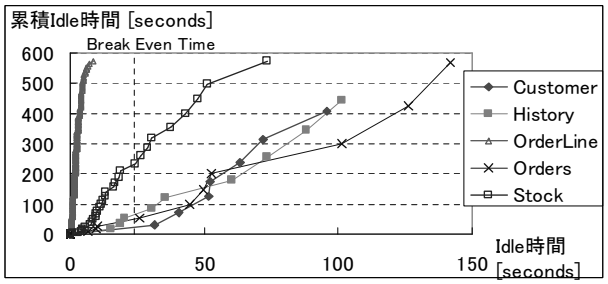

図 5. TPC-C のデータ毎の I/O 時間分布

5. OLTP 系アプリケーションの I/O 挙動特性を用いた省電力方式

前章で示したように，DB バッファが DB サイズと比較して大きい環境における OLTP 系アプリケーションの I/O 挙動には，ログへの write が支配的でること，及び表及び索引データの I/O の大半は write である，という特性がある．

我々はこの 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，ディスクドライブへのデータ配置を 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ドライブの生成，及び表・索引データへの write を同一ディスクドライブ上のデータへの 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．

5.1. データ配置に基づくディスクドライブ省電力化

データ配置の制御は主に設計時に実施する省電力化であり，ディスクドライブの許容できる容量及び IOPS を超えない範囲で I/O の多いデータを少数のディスクに片寄せする方式である．DBMS が管理するデータの単位を用いて設計時に片寄せを行う以外は，2.3 節で述べた手法と発想は同じである．

5.2. I/O 発行制御に基づくディスクドライブ省電力化

DBMS が表及び索引データに対して行う write は，主にチェックポイントと呼ばれる更新された DB バッファページのディスクへの書込み，及び DB バッファに空きがない状態で新たに DB バッファへのページの読み込みが必要となった際に更新されたページをディスクに書き込み空きを作る場合の 2 通りである．このうち，後者は DB バッファに空きがある間は発生せず，

前者はDBMSの問合せとは非同期に実施される処理であり遅延させることが可能である．

遅延方式は何通りか考えられるが，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 を保留．

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．write の実行時はディスクドライブ当りの IOPS が 60 を超えないよう制御する．

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. write の実行時はディスクドライブ当りの IOPS が 60 を超えないよう制御する．

6. 提案方式の評価

続いて，我々は前章で議論した省電力化方式の評価を実施した．評価を行うに当り，我々はまずディスクドライブ 2 台からなるごく小規模な環境を用いて効果を確認し，次いで省電力化方式をディスクドライブ 5 台からなる環境に適用した場合の評価を実施した．

6.1. ディスクドライブ 2 台使用時の評価

提案方式の効果を確認するため，まずデータ配置の制御のみを実施した場合の省電力効果を，実機を用いて評価した．その後，I/O 発行制御を併用した場合の省電力効果をシミュレーションにより求めた．

6.1.1. データ配置制御の評価

(1) 機器構成

評価に用いた機器の構成を図 6 に示す．

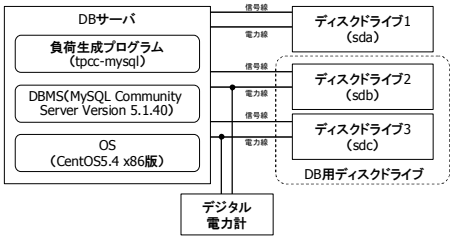

図 6. ディスク 2 台使用時の実験環境

3 章, 4 章で用いた実験環境に SATA ディスクドライブを 1 台追加し，DBMS が使用するディスクドライブを 2 台とした構成である．追加したディスクドライブも 3 章，4 章で述べたものと同一種類のディスクドライブ(Seagate 社の Barracuda ES ST3750640NS)である．追加したディスクドライブの電力線にもデジタル電力計を取り付け消費電力の計測を可能とした．

また，実験に用いた DB の規模及び DBMS の設定も 4 章と同じものを用いた．両ディスクドライブとも，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

(2) データ配置

本実験では，データ毎の I/O 頻度に基づくデータ片寄せの効果を確認するために，表 1 に示す 4 つのケースについて，データ配置制御の評価を実施した．本配置では，一方をアクティブなディスクドライブに，他方を非アクティブなディスクドライブとし，非アクティブ側のディスクドライブをスピンダウンすることによる消費電力の削減を狙っている．ディスクドライブ 3(sdc)側を非アクティブなディスクドライブとし，ケース番号が大きくなるにつれ, sdc 側の I/O が減少する構成とした．

表 1. データ配置と平均 IOPS

ケース	ディスク 2(sdb)	ディスク 3(sdc)
Case #1	ログ	Customer,District,History, Item,NewOrder,OrderLine, Orders,Stock,Warehouse
	111.2 IOPS	2.5 IOPS
Case #2	ログ, OrderLine	Customer,District, History,Item,NewOrder, Orders,Stock,Warehouse
	113.2 IOPS	0.5 IOPS
Case #3	ログ, NewOrder, OrderLine, Orders, Stock	Customer,District, History,ItemWarehouse
	113.5 IOPS	0.2 IOPS
Case #4	ログ, Customer, NewOrder,OrderLine, Orders,Stock	District,History,Item, Warehouse
	113.6 IOPS	0.1 IOPS

(3) 評価結果

各ケースの単位時間当りの消費電力とトランザクションスループットをそれぞれ図 7, 8 に示す．各ケースとも，ディスクドライブの省電力機能を使用しない場合の消費電力及びスループットで正規化した値を示している．

図 7，8 から分かるように，ケース#2 では単位時間当りの消費電力は 10%以上上昇し，トランザクションスループットは 25%以上減少している．しかし，ケース#3，#4 では逆に消費電力は減少し，スループットも 1 割減程度にとどまっていることが分かる．

ケース#2 でディスクドライブの短時間当たり消費電力が増加しトランザクションスループットが大きく落ち込んだ理由は，ディスクドライブ 3(sdc)側の Idle 時間が 5 秒より長いもののブレークイーブン時間である 23.9 秒より短く起動損が発生したこと，またこれによりトランザクションが待たされディスクドライブ 2(sdb)側への I/O が停止しさらなる起動損が発生したためである．一方，ケース#3，#4 において消費電力を削減できた理由は，ブレークイーブン時間(23.9 秒)よ

り長い Idle 時間を確保できたためである．ケース#3, #4 のトランザクションスループット低下は，ディスクドライブ 3(sdc)の起動待ちによる．

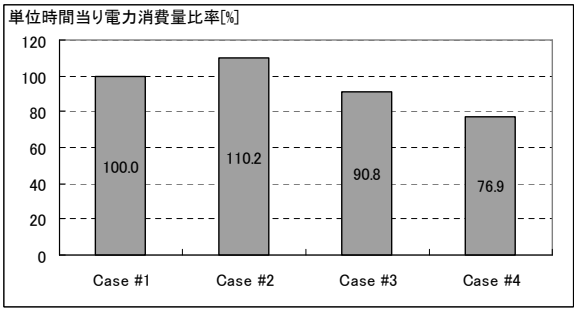

図 7. ディスク 2 台構成時の消費電力
（データ配置変更時の実測値）

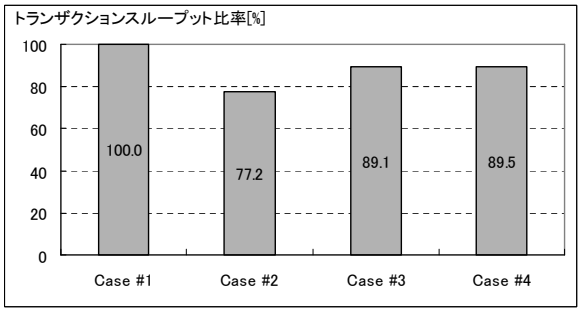

図 8. ディスク 2 台構成時のスループット
（データ配置変更時の実測値）

6.1.2. I/O 発行制御方式の評価

次に我々は，I/O 発行制御を用いた場合の消費電力の削減効果及びトランザクションスループットをシミュレーションにより評価した．シミュレーションにおいて，各ディスクドライブの消費電力は 3 章に示した値を，各データへの I/O は 4 章で取得したデータを用いた．

この結果を図 9，10 にそれぞれ示す．結果は前節と同様，ディスクドライブの省電力機能を使用しない場合の消費電力及びスループットで正規化している．

図 9 から分かるとおり，ケース#1 を除いて単位時間当り消費電力は減少し，消費電力を最大 37.9%削減できる結果となった．一方，トランザクションスループットはケース#1 で 6%減，ケース#2 で 5.3%減，ケース#3 及び#4 の減少率は 1%未満であった．ケース#1 において消費電力の増加とトランザクションスループットの低下が見られ，またケース#2～#4 において消費電力の削減とトランザクションスループットの減少率の低下が見られたのは，前節と同様の理由による．

以上の結果より，2 ディスクドライブのみを有するごく小規模なシステムにおいても，DB バッファサイズを DB サイズ以上の場合には，データ配置の調整と I/O 発行制御を行うことにより高い消費電力削減効果を得られることが確認できた．

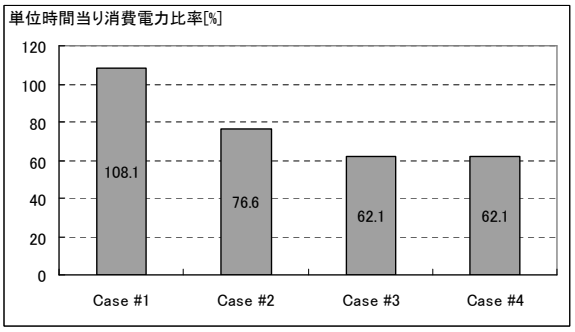

図 9. ディスク 2 台構成時の消費電力
（データ配置変更+I/O 発行制御時のシミュレーション値）

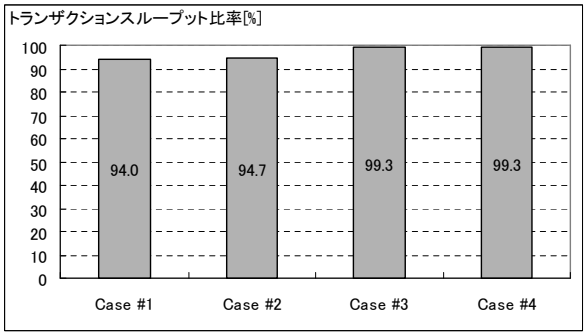

図 10. ディスク 2 台構成時のスループット
（データ配置変更+I/O 発行制御時のシミュレーション値）

6.2. ディスクドライブ 5 台使用時の評価

次に，より多数のディスクドライブを用いた場合の効果を確認するため，ディスクドライブ 5 台を用いた場合についてシミュレーションによる評価を実施した．

(1) シミュレーション条件

シミュレーションを行うに当り，各ディスクドライブのデータ量の差が小さくなるようデータを配置した．各ディスクドライブへのデータの割当て及び容量は表 2 に示すとおりであり，ディスクドライブ#1 にはログを，ディスクドライブ#2 には Customer 表及び索引を，ディスクドライブ#3 には OrderLine 表及び索引を，ディスクドライブ#4 には Stock 表及び索引を，ディスクドライブ#5 にはこれら以外の表及び索引をそれぞれ配置した．ディスクドライブ#1 から#4 の容量はそれぞれ全データの 17%～30%，ディスクドライブ#5 は全データの 8%である．

表 2. データ配置(5 ディスクドライブ)とデータ量

ディスクドライブ	データ	容量[%]
Disk #1	ログ	20
Disk #2	Customer	17
Disk #3	OrderLine	25
Disk #4	Stock	30
Disk #5	District, History, Item, NewOrders, Orders, Warehouse	8

また，前節のシミュレーションと同様，ディスクドライブの消費電力として 3 章に示した値を，各データ

への I/O として 4 章において取得した I/O トレースをそれぞれ用いてシミュレーションを行った.

(2) シミュレーション結果

シミュレーション結果を図 11, 12 に示す. 図 11 は単位時間当りの消費電力であり, 省電力機能を用いない場合を規準に正規化している.図から分かるように, ディスクドライブ省電力機能と I/O 発行制御機能を併用することにより, 消費電力を約 41.2%削減できることが分かる.図 12 はトランザクションスループットを示しており, ディスクドライブ省電力機能と I/O 発行制御機能の併用時は省電力機能未使用時とほぼ同等のスループットであることが分かる.

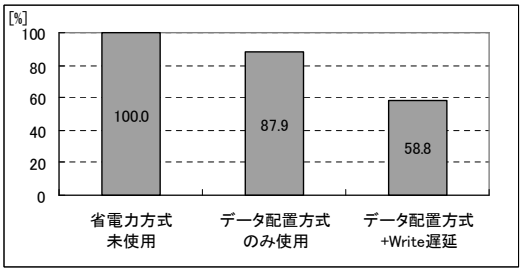

図 11. 単位時間当り消費電力(5 ディスクドライブ時)

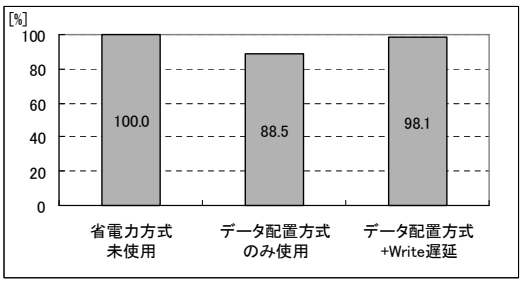

図 12. トランザクションスループット
(5 ディスクドライブ時)

6.3. DB バッファサイズと省電力効果の関係

これまで示した結果は, DB バッファサイズが DB サイズより十分大きな場合についてのものである. しかし, DBMS は常に DB バッファサイズが DB サイズより大きい状態で用いられるとは限らない. そこで我々は, DB バッファが DB サイズと同等, 及び小さな場合についても評価を行った.

評価に当り, 我々は DB サイズを 1 とした場合に DB バッファサイズを 0.1, 0.5, 0.75, 1.0, 1.25 と変化させて消費電力及びスループットをシミュレーションした. 各ディスクドライブへのデータ配置は前節表 2 と同一である. また, シミュレーションに用いた I/O トレースは, 4 章で示した環境及び方式を用いて DB バッファサイズ毎に取得したものを用いた.

シミュレーションの結果を図 13, 14 に示す. 図 13 は単位時間当りの消費電力の比較であり, 各 DB バッファサイズにおいて省電力機能を用いない場合を規準に正規化している.図 14 はトランザクションスループットの実測値と省電力機能を用いない場合と比較した場合の比率を示している.

図 13 から分かるとおり, DB バッファサイズが DB サイズの 10%の場合は省電力化方式を用いた場合と用いない場合の差はほとんどない. ここから DB バッファサイズを増加させるに従い消費電力, トランザクションスループットとも悪化し, DB バッファサイズが DB サイズを超えると省電力効果が現れていることが分かる.

この理由を探るため, ディスクドライブ毎の I/O 数の比率の変化, 及び I/O 発行制御を実施した場合の平均 Idle 時間(シミュレーション結果)の変化を調査した. この結果を図 15 に示す.

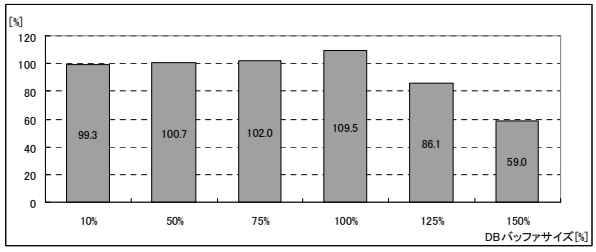

図 13. 単位時間当り消費電力
(5 ディスクドライブ, DB バッファサイズ変化時)

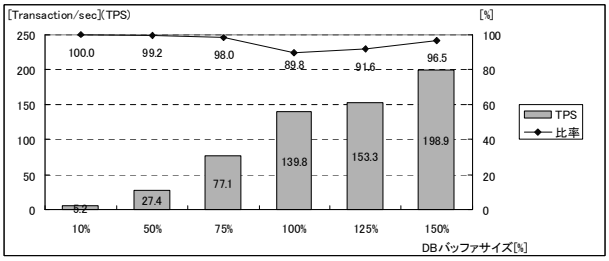

図 14. トランザクションスループット
(5 ディスクドライブ, DB バッファサイズ変化時)

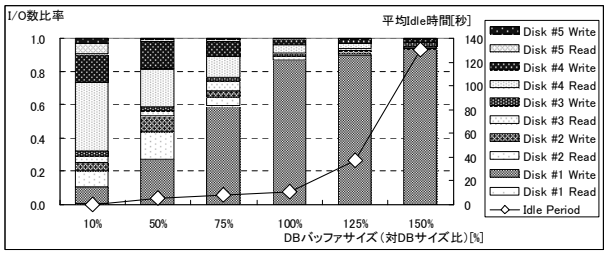

図 15. ディスク毎の I/O 数比率及び平均 Idle 時間

図 15 から分かるように, DB バッファサイズが 10%の場合は表及び索引への I/O(特に read)が多く各ディスクドライブに I/O が分散している. また DB バッファサイズが増加するに従い Disk1(ログ用ディスク)への Write の比重が高くなっていた. また平均 Idle 時間は DB バッファサイズが 10%の場合は 0.0 秒, 50%～100%の間はディスクドライブの省電力機能が動作する 5 秒以上であるがブレークイーブン時間(23.9 秒)より短く, 125%以上の場合にブレークイーブン時間より長くなることが分った.

この結果より，DB バッファサイズが 10%の場合はディスクドライブの省電力機能の使用に至らず，50%～100%の場合は，省電力機能は動作したものの損が発生したことにより，省電力化することはできなかったことが分かる．

以上の結果より，DB バッファサイズが小さい場合には提案方式の効果はほとんどない反面，DB バッファサイズが DB サイズと比較して大きい場合には，データ配置制御及び I/O 発行制御が有効性に機能することが示された．

7. まとめ

本論文では，OLTP 系 DBMS が持つ知識を用いた複数ディスクドライブの省電力方式の提案と評価を実施した．提案方式の特長は問合せを多数実行した場合の DBMS のマクロ的な挙動に着目し write 主体環境において Idle 時間の延伸を図る点であり，OLTP 系アプリケーションの I/O 挙動特性と DBMS の内部挙動特性を活用した省電力方式について述べた．また，提案方式の評価を行い DB バッファサイズが DB サイズより大きな環境下では複数ディスクドライブの消費電力の大幅な削減が可能になることを示した．今後は，本来の課題である大規模 RAID システムにおいて，DBMS の情報を利用することによる省電力の可能性について検討を行いたい．

参 考 文 献

[1] R.Bauer, “Building the Green Data Center: Towards Best Practices and Technical Considerations,” Storage Networking World Fall 2008 Conference, http://net.educause.edu/ir/library/pdf/bauer.pdf, 2008.

[2] P.B.Chu, E.Riedel, “Green Storage II: Metrics and Measurement,” http://net.educause.edu/ir/library/pdf/churiedel.pdf, 2008.

[3] D. Reinsel, “WHITE PAPER Datacenter SSDs: Solid Footing for Growth”, IDC #210290, 2008.

[4] 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.

[5] M.Allalouf, Y.Arbitman, M.Factor, R.I.Kat, K.Meth, D.Naor, “Storage Modeling for Power Estimation,” Proc. of SYSTOR 2009: The Israeli Experimental System Conference, 2009.

[6] “Product Manual Barracuda ES Serial ATA,” http://www.seagate.com/staticfiles/support/disc/manuals/enterprise/Barracuda%20ES/SATA/100424667b.pdf, 2006.

[7] 上野裕也，合田和生，喜連川優，”データベースシステムの問合せ実行計画を利用したディスクアレイ省電力化に関する一考察,” DEWS 2007.

[8] 合田和生，Q. Wenyu, 喜連川優，”複数問合せを意識したディスクストレージ省電力化に関する一考察,” DEWS 2009.

[9] F. Douglis, P. Krishnan, B. Bershad, “Adaptive Disk Spin-Down Policies for Mobile Computers,” Proc. of 2nd USENIX Symposium on Mobile and Location Independent Computing, 1995

[10] D. P. Helmbold, D. D. E. Long, T. L. Sconyers, B. Sherrod, “Adaptive Disk Spin Down for Mobile Computers,” Mobile Networks and Applications, Vol. 5, No. 4, 2000

[11] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, H. Franke, “DRPM: Dynamic Speed Control for Power Management in Server Class Disks,” 30th Annual International Symposium on Computer Architecture, 2003

[12] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, J. Wilkes, “Hibernator: Helping Disk Arrays Sleep through the Winter,” Proceedings of the twentieth ACM symposium on Operating systems principles, 2005

[13] A. E. Papathanasiou, M. L. Scott, “Energy Efficient Prefetching and Caching,” Proc. of the USENIX 2004 Annual Technical Conference, 2004

[14] D. Li, J. Wang, “EERAID: Energy Efficient Redundant and Inexpensive Disk Arrays,” Proceedings of the 11th workshop on ACM SIGOPS European workshop, 2004

[15] X. Yao, J. Wang, “RIMAC: A Novel Redundancy based Hierarchical Cache Architecture for Energy Efficient,” High Performance Storage System Proceedings of the 2006 EuroSys conference, 2006

[16] D. Colarelli, D. Grunwald, “Massive Arrays of Idle Disks For Storage Archives,” Supercomputing, ACM /IEEE 2002 Conference, 2002

[17] E. Pinheiro, R. Bianchini, “Energy Conservation Techniques for Disk Array Based Servers,” Proceedings of the 18th annual international conference on Supercomputing, 2004

[18] C. Weddle, M. Oldham, J. Qian, A. A. Wang, “PARAID: A Gear-Shifting Power-Aware RAID,” FAST ’07: 5th USENIX Conference on File and Storage, 2007

[19] S. W. Son G. Chen M. Kandemir, “Disk Layout Optimization for Reducing Energy Consumption,” Proceedings of the 19th annual international conference on Supercomputing, 2005

[20] C. Gniady, Y.C. Hu, Y.H. Lu, “Program Counter Based Techniques for Dynamic Power Management,” High Performance Computer Architecture, 2004

[21] T. Heath, E. Pinheiro, J. Hom, U. Kremer, R. Bianchini, “Application Transformations for Energy and Performance-Aware Device Management,” Parallel Architectures and Compilation Techniques, 2002

[22] S. W. Son, M. Kandemir, A. Choudhary , “Software-Directed Disk Power Management for Scientific Applications,” Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, 2005

[23] Transaction Processing Performance Council, “TPC-C, an online transaction processing benchmark,” http://www.tpc.org/tpcc/.

[24] “tpcc-mysql,” https://code.launchpad.net/~percona-dev/perconatools/tpcc-mysql.