データベースシステムの問い合わ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

上野　裕也†　　合田　和生†　　喜連川　優†

† 東京大学生産技術研究所
E-mail: †{ueno,kgoda,kitsure}@tkl.iis.u-tokyo.ac.jp

あらまし　従来、コンピュータシステムにおける消費電力に関しては、主にモバイルコンピューティングにおけるプロセッサの省電力化を中心に検討がなされてきた。近年では、データセンタなどの大規模システムにおいてもその消費電力が問題となりつつあり、また、プロセッサだけでなく周辺の入出力機器を含めたシステム全体の省電力化が求められるようになっている。特に、システムの管理するデータ量が急激に増大している中、ディスクドライブの省電力化は極めて重要な課題である。本論文では、多数のディスクドライブから構成されるディスクアレイの省電力化を目指し、データベースシステムの有する問い合わせ実行計画を利用した新しいディスクドライブの制御方式を提案する。独自の方式により構築したディスクドライブに関する消費電力モデルを示すとともに、当該モデルに基づく解析的検討により、従来方式と比較して大きな効果が得られることを明らかにする。

A Study on Disk Array Power Reduction using Query Plan of Database Systems

Yuya UENO†, Kazuo GODA†, and Masaru KITSUREGAWA†

† University of Tokyo Institute of Industrial Science
E-mail: †{ueno,kgoda,kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract　So far, power consumption of processors has been studied mainly in mobile computing. Nowadays, the power consumption of enterprise systems, such as data centers, is beginning to be a new problem, and not only processors but also other subsystems including input/output devices are needed to be considered. Espcially, power reduction of disk drives is the most important issue because of the recent extraordinary increase of data managed by the systems. This paper aims at the power reduction of the disk array composed of many disk drives, and proposes a new control method of disk drives exploiting query plan of database systems. With showing our own power consumption model of the disk drive, we reveal that significant power reduction can be achivable by the analitical examination based on the model.

1. はじめに

1.1 動　　機

ディスクアレイは、サーバやデータセンタ等で利用されるコンピュータシステムを構成する主要デバイスであり、その消費電力は極めて大きくなってきている。その理由として、ネットワークの広帯域化によるデータ量の増大、データを長期にわたり保存することを要請する法律の制定などが挙げられる。また、万一災害時にリカバリが遅れてしまった時、場合によっては企業の倒産に繋がる恐れもあり、それに至らなかったとしても、時間に比例した莫大な損失が発生することは、現在の情報、通信の価値を考えれば想像に難くはない。よって、このような災害に備えたデータのレプリケーションによる電力も大きなコストであると言える [1]。このような、近年のデータ量の増大、データ管理コストの増大に伴って、ディスクアレイの省電力に関する研究がなされるようになってきた。[2] [3]。

1.2 研究の目的

これまでの方式では、ストレージが省電力に関する制御を行う際、idle 時間の長さやアクセス頻度など、ストレージの枠を出ない範囲の統計的な情報のみから、回転数変更などの省電力に関わる動作を行っていた。しかし、本研究では、DBMS の持つ問い合わせ実行計画の情報を有効利用し、上位のソフトウエ

アのレイヤから能動的にディスクの動作を決定することによって、従来に比べて飛躍的な消費電力の削減を目指す。

1.3 論文の流れ

本論文の流れは以下の通りである。

まず、2. で関連研究を紹介する。3. で提案する方式の概要を示す。4. で実験環境と、使用するディスクのスループットに対する電力モデルを示す。5. で提案するモデル、並びに省電力化方式の評価を示し、最後に 6. にて論文のまとめと今後の課題を示す。

2. 関 連 研 究

2.1 回転数のコントロール

ラップトップマシンのディスクに関する研究でもそうであったが、ディスク省電力の基本はディスクの不必要な時に回転数を低くするというものである [4]。また、これらの実験においては、実際のサーバやデータセンタの I/O リクエストをトレースしたものを入力として用いるが、それらのトレースにはローカリティ、つまり、データアクセスの偏りがある。ワークロードの種類にもよるが、ある調査では、新規に write されたデータのうち、50 %は再び read されることがなく、30 %はたった一度の read アクセスしかないという報告もある。このように、実際のワークロードにおいてはローカリティが顕著に現れ、これを利用して一部のディスクのみを高回転数で稼働させている傍らで、idle 状態のディスクの回転を低下、または停止させて電力を削減するということが可能になる。

しかし、ここで問題になるのは、高回転数のディスクを standby 状態にする時、あるいは逆に standby 状態のディスクを高回転数にする時は、余分な電力と遅延時間がかかってしまうということである。従って、あまり頻繁にディスクを spin up/down するという行為は逆に、電力とパフォーマンス両方面への性能低下を招いてしまう。更にそれだけにはとどまらず、ディスクの寿命にも影響が現れる。一般に、ディスクは数万回の spin up/down に耐えることができると言われているが、現実的な寿命を得るためには spin up/down を一日数十回程度に抑える必要があり、あまり過度に回転数を変えるというわけには行かない。

2.2 Hibernator

上記の特徴に基づき、近年発表された興味深い論文に、Hibernator [5] がある。

Hibernator の特徴として、まず tier という概念がある。これは「同じスピードで動作するディスクのグループ」を意味していて、いくつかの tier がそれぞれのスピードで動作している。そして、tier を構成するそれぞれのディスクは、都合に合わせて tier 間を移動する、つまり、動作スピードを変えることができ、数時間に一回という粗い粒度で見直されるため、ディスクスピードの遷移頻度を抑えることができる。

では、これらの tier がどのような特徴を持っていることが望ましいかを考えてみると、まず第一に、回転数の高い tier にはアクセス頻度の高いデータブロックが入っていることが望ましいと考えられる。なぜなら、より多くのリクエストが高いスループットのサービスを得られるからである。

そして第二に、同じ tier の中の各ディスクへのアクセス頻度は均等になっている方がよいと考えられる。これは、一つのディスクにアクセス頻度の高いデータが集中すると、そのディスクのリクエストキューが長くなってしまって、仮に全てのディスクが高回転数で動作していたとしても、このディスクがボトルネックとなって大きな遅延が発生してしまうことになるからである。Hibernator では、アクセス頻度は temperature と呼ばれる指標によってブロック単位で管理されている。実際には、このブロックの分散は Randomized Shuffling と呼ばれる、ランダムでパリティも含めてブロックを分散させる方法を取っている。下手にアクセス頻度を計算して配置するよりも効率よく分散できることが示されている。

2.3 その他の技術

回転数を低くすることに加え、アクセスのローカリティが高い場合は、data migration やキャッシング [6] も補助的な有効性を持つ。MAID [7] [8] [9] [10] という製品ではキャッシングを利用して消費電力を削減している。

3. 問い合わせ実行計画を利用した省電力化方式

3.1 概　　要

従来のラップトップ用ディスクの研究を始め、データセンタにおけるディスクに関する先進的な研究においても、ディスクの電力を削減する手段としては、ディスクの回転数を低くしたり停止させたりするのが、分かり易くかつ有効である [4]。本研究においてもそれは変わらないが、問題はいかにして回転数を変更するタイミングを制御するかということにある。

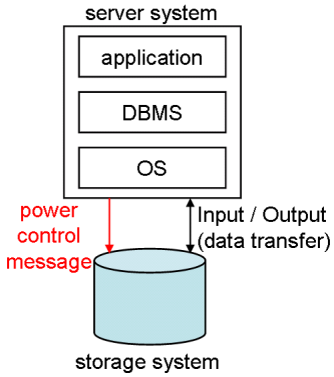


図 1　電力制御機構を持つデータベースシステム概念図

図 1 に、本研究で想定するデータベースシステムの概念図を示す。赤い線で示されている、電力モードを変更するメッセージを送れるようなインターフェースを加えることにより、以下のような考えを実現することができる。

従来の研究では、ディスクの idle 時間に応じて回転数を遷移させたり、各ディスクに対するアクセス頻度を学習してデータをディスク間で移動させるなど、ストレージ内部の情報をもとにディスクの動作を制御するしかなかった。しかし、本研究では、DBMS が自由に問い合わせ実行計画を決めることができることに着目し、その情報を元にディスクの動作を決定するこ

とを考える。つまり、データの場所が分かっていれば、動作する必要のあるディスクと必要のないディスク、またその時間などを把握することができ、アプリケーション、DBMS から能動的にディスクを制御することで、従来より効率のよい電力削減を図ることができる。

3.2 ディスクの状態遷移モデル

本研究では、ディスクを読み書き可能な回転数から低い回転数に遷移させることで電力削減を図る。本節では、ディスクの取りうる状態に関して説明を行う。

今回、データベースを保存するディスクとして、HGST の Deskstar T7K250 [11] を使用した。その理由は、idle 時の電力を数段階変えることができるからである。これを例にして、ディスクの各状態を説明していく。

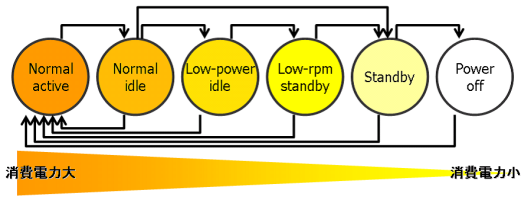


図 2　ディスクの取りうる状態と遷移

図 2 を参照しながら、ディスクのそれぞれの状態に関して簡単に説明する。

・normal active とは、ディスクが read や write を行っている最中の状態であり、最も電力が高い。

・normal idle とは、スループットが 0MB/s であり、ディスクは回転したまま、読み書きも、電力削減に繋がる特殊な動作も行っていない状態を表す。

・low-power idle とは、normal idle の特殊型で、ディスクのヘッドをランプに乗せ、ヘッドの動作に関わる電力を削減している状態を表す。

・low-RPM standby とは、low-power idle の状態から更にディスクを低速にすることにより、電力を削減している状態を表す。

・standby とは、low-RPM standby の回転数を 0round/min にしたものである。

・power off とは、ディスクに通電していない状態であり、消費電力は 0W となる。

本研究では、normal active、normal idle、low-RPM standby、standy の四つの状態に着目し、実ディスクにおけるそれぞれの状態の電力を実際に測定することにより、ディスク消費電力モデルを構築する。更に、そのモデルを用いて、三つのディスクからなるディスクアレイ上でのクエリ実行を想定した電力量解析を行う。

3.3 省電力化方式

図 3 を参照しながら、提案する方式の流れを説明する。

今回の実験を行った実環境では、全てのテーブルが一つのディスクに格納されている。しかし、本研究では、一つのテーブルを一つ、またはそれ以上のディスクにシーケンシャルに割り当てられているような状況を想定する。例えば、table1、table2、table3 をそれぞれを異なるディスク#1、#2、#3 に格納しておく。これら三つのテーブルの join を図 3 の問い合わせ実行計画に基づいて行うとする。

step1: table1 を build するためにディスク#1 を activate する。その間、ディスク#2、#3 を spin down し、table1 の build を待つ。

step2: step1 が完了したら、次にディスク#1 を spin down、ディスク#2 を activate して table2 による probe を行う。その間もディスク#3 は動作する必要がないため、probe と第 1join 後のテーブルが build されるのを待つ。

step3: step2 が完了したら、ディスク#2 を spin down、ディスク#3 を activate して、table3 の probe を行う。

ストレージに spin up/down を直接指示することができれば、このようにして、動作する必要のあるディスクのみを DBMS が定めた問い合わせ実行計画に基づいて特定することにより、より効率の良い省電力化方式が可能になる。

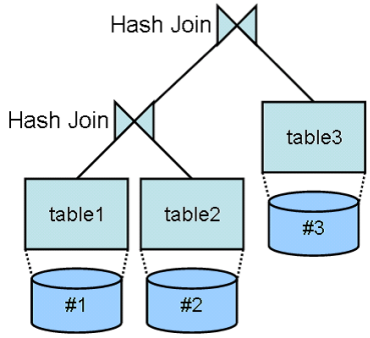


図 3　想定するテーブル配置

4. ディスク消費電力のモデル化

4.1 実験環境

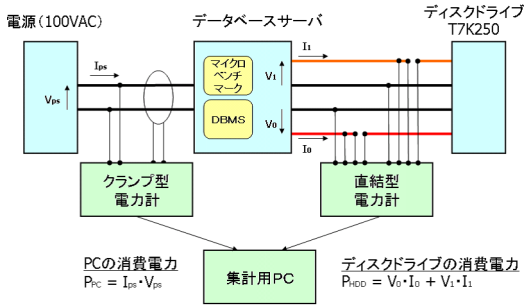


図 4　ディスク電力測定回路の概略

本節では、各種実験環境について説明する。

まず、図 4 はディスク電力を測定するための回路の概略図である。DB サーバからディスクへは 4pin の電源ケーブルを介して電力供給が行われ、赤線は 5V、黄線は 12V を表している。それらの線を YOKOGAWA 製の電力計、WT230 に通し、電流を測定する。また、赤線と黒線 (GND)、黄線と黒線の間にもクリップを噛ませ、電圧を測定する。この電流と電圧二つずつから得られた電力二つの合計が、ディスクの電力となる。更

に、DB サーバ全体の電力を把握するために、YOKOGAWA 製の電力計、CW120 を用いて交流電流と電圧を監視している。

次に、DB サーバの状態を示す。DB サーバとしては Gateway 製デスクトップマシンを用い、CPU は Pentium4 1496.361MHz、メモリは 381508kByte である。OS は、Redhat enterprise LINUX Version 3 を用いている。HGST 製 T7K250 を IDE インターフェースのセカンダリ/マスターに接続し、更に、raw キャラクタデバイス /dev/raw/raw1 を、ブロックデバイス /dev/hdc1 にバインドすることにより、raw デバイスとして取り扱っている。

T7K250 のスペックは [11] を参照されたい。(注1)。

4.2 active 状態におけるスループットベースのディスク消費電力モデル

本研究では、図 3 のようなディスクアレイの消費電力量を計算するために、独自の実験的手法により、実測値に基づいたディスク消費電力モデルを構築する。

ディスクの normal active 状態では、read または write を行っている。更に、それらは sequential と random に分けることができる。sequential ではヘッドの動きがほとんどないので電力は小さく、逆に random ではヘッドに費やす電力が大きいと考えられる。

また、各 I/O に対してハッシングやその他の処理で次の I/O までに時間がかかる場合もある。この時、I/O 以外の処理によってスループットが変化する。このスループットの違いによってもディスクの電力が変わってくると考えられる。

以上を踏まえ、本研究では、独自の I/O 負荷生成ツールによって作られた、sequential read、random read の電力を、スループットを調節しながら測定する(注2)これにより、様々なスループットにおける電力を測ることが可能になる。更に、今回用いたツールでは、1I/O のサイズを自由に変えることができる。よって、4096Byte、16384Byte、65536Byte の三種類で実験を行う。

4.2.1 シーケンシャルアクセス時の消費電力モデル

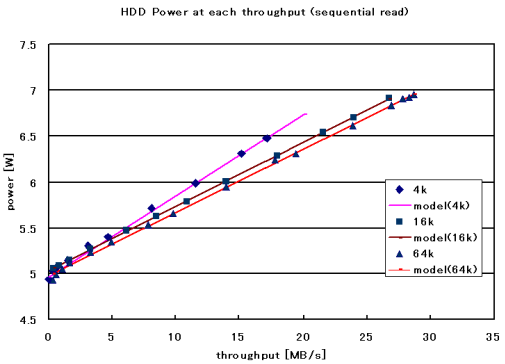


図 5　sequential read におけるスループットに対するディスク電力測定値とモデル

(注1)：データ転送速度は、DB サーバのチップセットの制約でモード udma2 のため、スペックより低い値となっている。

(注2)：今回は、ベンチマークが read のみを使用するため、write は省略する。

表 1　各 idle 状態における消費電力

normal idle	low-power idle	low-RPM standby	standby
4.744W	3.777W	2.223W	0.874W

図 5 は、I/O 負荷生成ツールにより作られた sequential read の負荷における、スループットに対するディスク消費電力の測定値と、それに最小二乗法を適用することにより求めた電力モデルを図示したものである。

それぞれ、かなり正確な直線になっており、また、1I/O のサイズが大きいほど電力効率が良い (傾きが小さい) ことが分かる。それぞれのデータに最小二乗法を適用して得られたモデルの直線の式は、以下のようになった。

$Y = 0.0849X + 5.0176 \quad (4k)$

$Y = 0.0701X + 5.0324 \quad (16k)$

$Y = 0.0686X + 4.9859 \quad (64k)$

(X:スループット [MB/s]　Y:消費電力 [W])

4.2.2 ランダムアクセス時の消費電力モデル

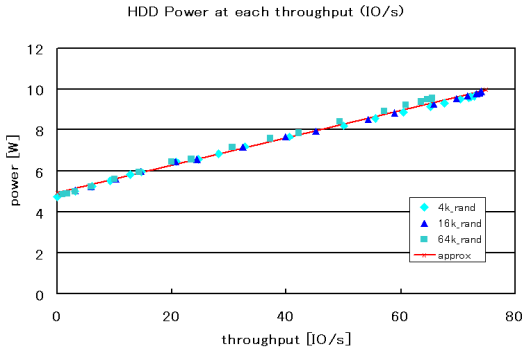


図 6　random read におけるスループットに対するディスク電力測定値とモデル

図 6 は、I/O 負荷生成ツールにより作られた random read の負荷における、スループットに対するディスク消費電力の測定値と、それに最小二乗法を適用することにより求めた電力モデルを図示したものである。横軸は、IO/s である。

こちらも sequential read 同様、ほぼ正確な直線となっているが、どの I/O サイズにおいてもほぼ同じ電力となっていることが分かる。これは、random アクセス特有のシーク時間がボトルネックとなって、ヘッドの動きの多寡が、I/O サイズよりも IO/s に依存しているためと考えられる。これら全てのデータに最小二乗法を適用してみると、得られた直線の式は、以下のようになった。

$Y = 0.0673X + 4.9177$

(X:スループット [IO/s]　Y:消費電力 [W])

4.3 idle 状態におけるディスク消費電力モデル

本節では、idle 状態における電力のモデル化を行う。

表 1 は、3.2 で説明した、normal idle、low-power idle、low-RPM standby、standby 状態における、各消費電力の実測値を示している。idle 時の電力なので、これにはスループットは

関係なく、固定の値を定める。

今回の実験では、通常の idle 状態として normal idle を、省電力状態として削減率の高い low-RPM standby、standby をモデルとして採用することにする。

4.4 各状態遷移コストのモデル

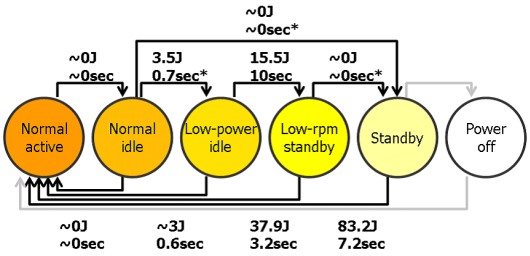


図 7　各状態遷移コスト

図 7 は、各状態遷移コストを表したものである。spin up に関しては、I/O リクエストを一つ与えた時点から I/O がコミットされるまでの時間を遷移時間とし、spin down に関しては、電力が低下し始めてから安定するまでの時間を遷移時間とする。

standby への spin down に関しては、状態遷移後に回転数を維持する必要がないため、スピンドルモータへの電力供給を止めることで遷移完了となり、1msec 以下の短い遷移時間となっている。また、spin down の始点と終点に関しては、1 秒精度の電力グラフに頼らざるを得ないので、遷移時間が非常に短いものは、やむを得ず仕様上の値を借りることにする。

以上で、図 2 で示された、power off 状態以外の全ての電力モデルが出来上がることになる。これにより得られた電力モデルに、ベンチマーク実行中のスループットと実行時間、idle 状態の時間を適用することによって、3.3 で紹介した各方式のディスク電力量を算出する。

5. 評価と考察

5.1 ディスクの消費電力モデルの評価

まず、4.2 で作成した消費電力モデルを評価する。実際のベンチマーク TPC-H [12] を、HITACH 製の DBMS HiRDB と、オープンソースの DBMS MySQL [13] を用いて実行した時のディスクにモデルを適用し、妥当性の評価を行う。具体的には、各テーブルの読み込みの際、sequential read、random read のモデルを適用し、スループットと読み込み時間から各フェーズの電力量を計算する。それを全てのフェーズで行い、更に、読み込み以外の時間を idle 状態のモデルに適用して電力量を合計する。こうして算出した電力の予測値と、実際にクエリを実行している間の実測電力量を比較する。

図 8 は、HiRDB における各クエリ実行時の、モデルによる予測消費電力量と実測値を比較し、値を正規化したものである。誤差は最大で 17 % (Q2) となっているが、Q2 以外は誤差 10 %に収まっている。また、どのクエリでもモデルによる値より実測値の方が上回っているが、これはバッファキャッシュやコントローラ系統による電力が上乗せされているためであると推測される。この原因を調査し、更に正確なモデルを構築することは今後の課題としたい。

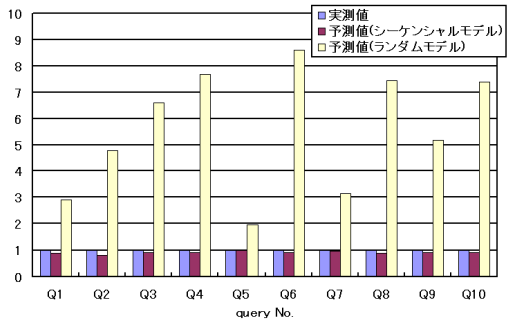


図 8　HiRDB における各クエリ実行時のモデルによる予測消費電力量と実測値

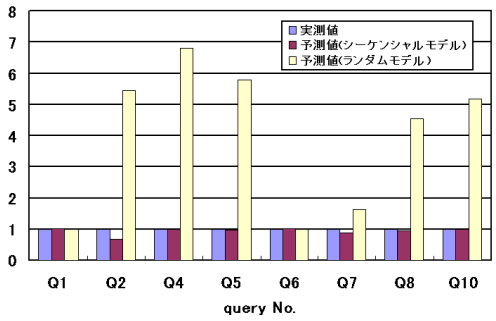


図 9　MySQL における各クエリ実行時のモデルによる予想消費電力量と実測値

更に、MySQL に関しても同様の実験を行った。図 9 は、MySQL における各クエリ実行時の、モデルによる予測消費電力量と実測値を比較し、値を正規化したものである。ほとんどの問い合わせに関しては、HiRDB に近い、ほぼ sequential なアクセスであったため、sequential のモデルがほぼ正確な値となっていた。実測値との誤差は、1 %から 6 %と非常に小さな値となった。

しかし、Q2、Q7 に関する実験結果は、ヘッドが頻繁に移動するランダムアクセスに近かったため、シーケンシャルモデルの予測値が他の問い合わせと比較すると大きくずれるものになっていたが、ランダムモデルによる予測値も正確であるとは言えなかった。

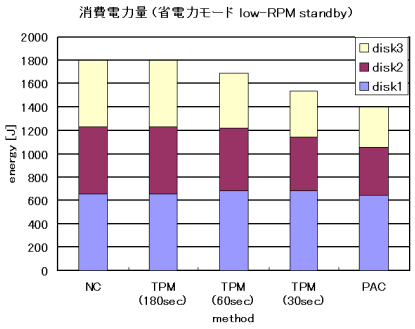
原因として考えられるのは、1I/O のサイズが大部分が 16kB であるが 1MB 以下の大きさも混じっていること、また、IO/s の値が特定しづらかったことなどが挙げられる。さらに、バッファキャッシュの影響で、実際にヘッドの動作とディスク読み取りの動作に関わっていない I/O が多数存在したため、IO/s の値が大きくなり、予測電力量が実測値より大きくなってしまったということも考えられる。よって、ランダムモデルに関しては、それらを考慮した検討の余地がある。

5.2 実験の補足条件

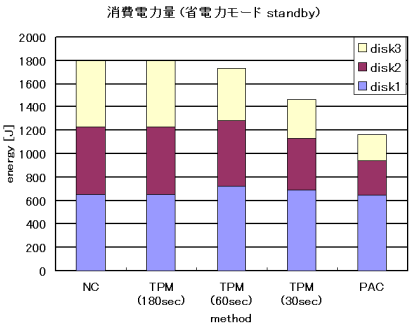
5.3 の解析は以下のような条件で行うものとする。

・ベンチマークとして TPC-H Q8、Q9 を用いる。

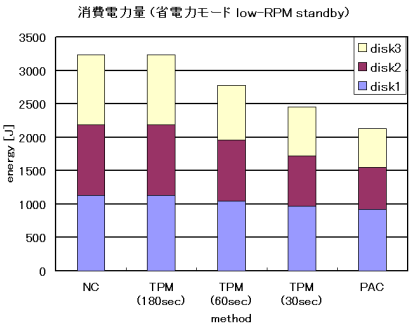
・三つのディスクに、lineitem テーブル、orders テーブル、その他のテーブルが保存されている状態を想定し、それぞれディ



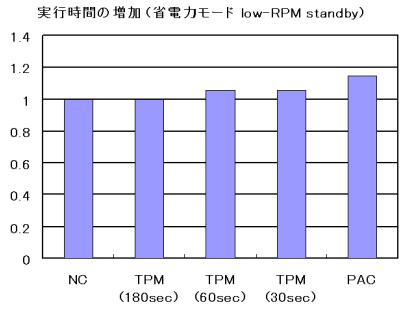
(a1)Q8 LD における各方式の消費電力量
(省電力モード low-RPM standby)



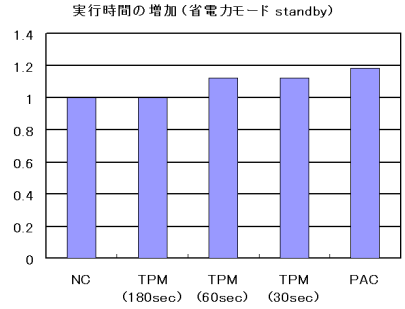
(b1)Q8 LD における各方式の消費電力量
(省電力モード standby)



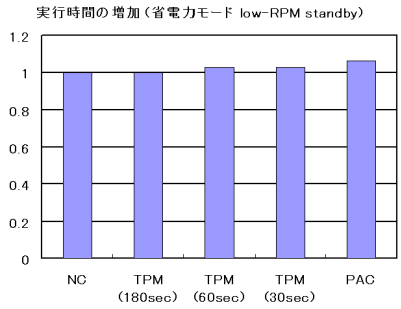
(c1)Q9 LD における各方式の消費電力量
(省電力モード low-RPM standby)



(a2)Q8 LD における各方式の実行時間
(省電力モード low-RPM standby)



(b2)Q8 LD における各方式の実行時間
(省電力モード standby)



(c2)Q9 LD における各方式の実行時間
(省電力モード low-RPM standby)

図 10　各方式の消費電力量と実行時間 (Q8 LD & Q9 LD low-RPM standby、Q8 LD standby)

スク 1、ディスク 2、ディスク 3 と呼ぶ。

・想定する状態としては、normal active、normal idle、省電力モードの三つを考え、それぞれを遷移するものとする。

・省電力モードとして、low-RPM standby と standby の二種類を考える。

・spin up/down は、図 7 を元に、時間的、電力的コストを含めた値を算出する。

・TPM(後述) の時間閾値は、180 秒、60 秒、30 秒とする。

・TPM、PAC(後述) では、初期状態は両電力モードではなく、normal idle であるものとする。つまり、PAC で初めから省電力モードに入るときは、一度 spin down する。

・小さなテーブルを対象とした細かい read のための spin up は無視する。

・それらの細かい read 部や idle 時間は非常に短いので無視する。ただし、Q9 の idle 時間は比較的長いので、全て idle として加算する。

5.3　TPC-H のクエリによる三つの方式の評価

本節では、図 3 のような三つのディスクからなるディスクアレイを想定し、TPC-H の Q8、Q9 の LD、RD を実行したときの消費電力量を、構築したモデルを用いて算出する。その際、後述する三種類のディスク制御方式それぞれの比較を行う。

図 11 は、Q8 LD における問い合わせ実行計画の木構造である。これにより、アクセスするテーブルの順番が確定し、各テーブルを格納するディスクを把握していれば、ディスクの能動的な省電力制御が可能となる。

以上の状況を踏まえ、ディスク制御方式三つの説明を行う。

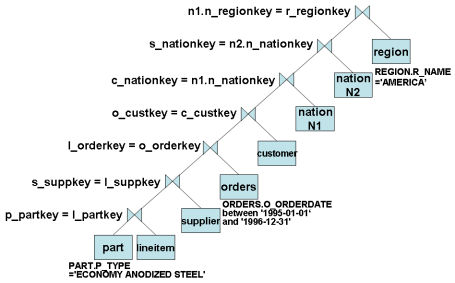


図 11　Q8 LD の問い合わせ実行計画

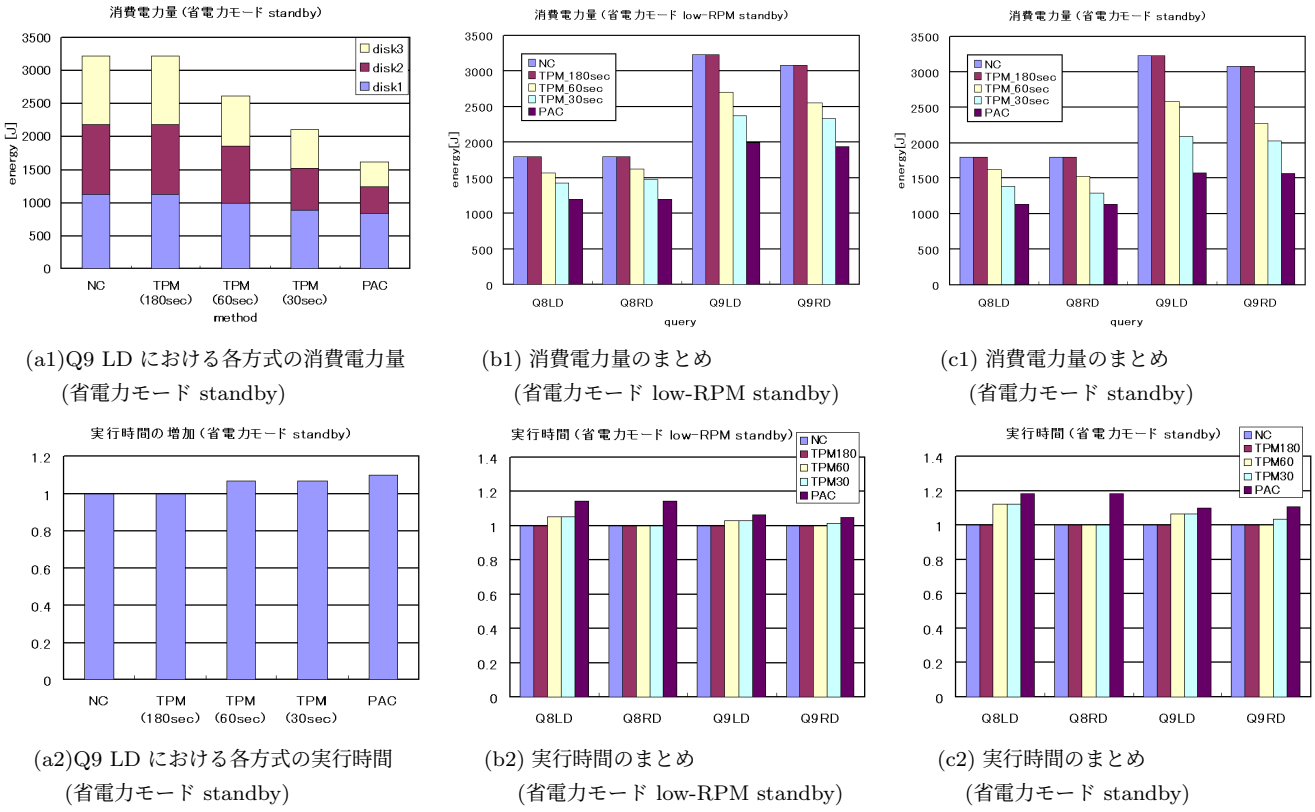
・NC:ディスクが idle であるときに何も省電力制御を行わない方式。これを基準として他方式を比較する。

・TPM:ディスクの idle 時間が一定閾値を超えたときに自動的にディスクが省電力モードに移行する方式。これは図 1 の赤線で示したような新しい機構を加えることなく実現できる方式である。

・PAC:DBMS がディスクの idle 期間を把握し、能動的にディスクを制御する方式。

Q8 LD、Q9 LD(注3) において NC、TPM、PAC によってディスク制御を行った時のそれぞれのディスクの電力量を計算すると、図 10(a1)〜図 12(a1) のようになった。また、このときのクエリ実行時間を、NC を 1 として正規化したグラフは図

(注3)：RD の個々のグラフに関しては紙面の都合上省略するが、データは図 12(b1)〜図 12(c2) にまとめて示す。



(a1)Q9 LD における各方式の消費電力量 (省電力モード standby)

(b1) 消費電力量のまとめ (省電力モード low-RPM standby)

(c1) 消費電力量のまとめ (省電力モード standby)

(a2)Q9 LD における各方式の実行時間 (省電力モード standby)

(b2) 実行時間のまとめ (省電力モード low-RPM standby)

(c2) 実行時間のまとめ (省電力モード standby)

図 12　各方式の消費電力量と実行時間 (Q9 LD standby) とまとめ

10(a2)〜図 12(a2) のようになった。これらのグラフには、ディスクの spin up、spin down に伴う時間的、電力的コストが含まれている。

各クエリの特徴を述べると、Q8 LD は、図 11 の左下のテーブルから順番にアクセスされる。Q8 RD はまず build を一度に行ってしまうが、テーブルの順序は任意であるため、基本的にディスク内のアドレスの順にテーブルが読み込まれるように最適化された。Q9 LD になると、どのディスクもアクセスされない idle 時間が多く見られるようになる。これは、probe の時間と、join の結果で得られたテーブルを新たに build している時間と考えられ、そのテーブルの大きさは、Q8 の時よりも大きくなっている (join の条件に合うタプルが多くなっている) と考えられる。この時間は、idle 状態として電力量に加算している。Q9 RD は、形としては Q8 RD と同様の形態になるが、lineitem テーブルによる probe の後に長い idle 時間が存在する。これは、Q9 LD における、各フェーズ間の idle 状態が、RD では最後にまとまったと考えられる。

5.4　実験結果のまとめ

図 12(b1)、図 12(c1) は、Q8、Q9 の RD も含めて、各クエリの電力量をまとめたものである。また、図 12(b2)、図 12(c2) は実行時間を正規化してまとめたものである。

省電力モード standby において、TPM では、Q8 に関して約 18 %〜28 %、Q9 に関して約 34 %の電力量削減が得られ、PAC では、Q8 に関して約 35 %、Q9 に関して約 50 %の電力量削減が得られた。実行時間に関しては、省電力モード low-RPM standby において、TPM では Q8 で 0〜約 5 %、Q9 で約 1〜3 %の増加、PAC では Q8 で約 14 %、Q9 で約 4〜6 %の増加となった。省電力モード standby においては、TPM では Q8 で 0〜約 12 %、Q9 で 0〜7 %の増加、PAC では Q8 で約 18 %、Q9 で約 10 %の増加となった。

Q9 では、join の際の条件に当てはまるタプルが多かったと見られ、LD での probe と再 build の際、RD での probe の際にディスクアクセスのない idle 状態が多く見られたため、その時の電力量削減の割合が Q8 より大きくなる結果となった。

次に、spin up/down の回数に関して考察すると、TPM では spin up が計 2 回、spin down が計 3〜5 回、PAC では spin up が計 3 回、spin down が計 5〜6 回となった。TPC-H のように一つのクエリが非常に大きい場合において各クエリでそれぞれのディスクに課す spin up/down の回数が数回という規模は、数万回の spin up/down に耐えうるディスクとしてはかなり現実的な数値であると考えられる。また、全体の処理時間を考慮に入れると、数回の spin up/down は比較的小さな割合の追加コストであるということができる。

また、今回は join の順序を DBMS の最適化にほぼ合わせたが、spin up/down の回数を最小にするような最適化を考えることにより、更に無駄を抑えることができると考えられる。加えて、テーブルを各ディスクにどのように配置するかも、spin up/down の回数に影響を与えることになる。

最後に、実行時間に関して考察すると、Q8 より Q9 の方が実行時間の増加の割合が小さかったが、これは単純に Q9 の方が実行時間が長かったためである。また、今回用いたデータベースは TPC-H のスケールファクタを 1.0 としているが、これを

高めたとしてもテーブルの数は変わらないため、spin up/down の回数も変わらない。従って、実際の DSS におけるデータベースのように、テーブルの数が本研究とほとんど変わらず、個々のテーブルサイズが何倍にもなるような状況において、spin up/down による実行時間の増加は、割合で表すと非常に小さくなることが予想される。また、PAC によって、spin up の時間的コストを考慮し、あるディスクの読み終わりと次に読むディスクの spin up の完了が同時に起こるような制御を行うことによっても、実行時間の増加を抑えることができると考えられるが、それは今後の課題となる。

6. まとめと今後の課題

6.1 ま と め

本研究では、巨大なデータベースシステムにおけるディスクアレイの消費電力問題を背景とし、問い合わせ実行計画に着目したディスク省電力の方式を提案した。特に複雑なクエリ処理を行う DSS に関しては、データベーステーブルの join に hash join を用いることで、長時間の idle 状態に置かれるディスクが多数存在することに着目し、それらのディスクを省電力モードにすることで電力削減を図った。また、データベーステーブルの特徴にあわせて問い合わせ実行計画を工夫しながら、left deep hash join、right deep hash join に関して実験を行った。

まず、想定するディスクアレイの消費電力量を算出するために、独自の I/O 負荷生成ツールを用いて実ディスクを動作させ、実際に電力を測定することにより、スループットに対するディスクの消費電力の関係をモデル化した。これは、過去の研究ではほとんど見られないものである。結果は、スループットの増加に対してほぼ直線的な電力上昇が見られ、I/O のサイズが大きいほど電力効率が良いことがわかった。特にシーケンシャルモデルにおいて精度の高い結果が得られたが、ランダムモデルに関してはまだ改良の余地がある。

次に、問い合わせ実行計画を用いることで読み込むテーブルの順序が確定すると、アプリケーション、DBMS、OS など、上位のレイヤから能動的にディスクの動作を決定することで、より効率的な省電力が実現できるということを示し、その環境を想定した三つの省電力化方式を提案した。そして、先の消費電力モデルを用い、三つのディスクを想定した省電力化方式の消費電力量を算出した。具体的には、独自の I/O 監視ツールを用いて、各テーブルを読み込む際のスループットを得た後、モデルに適用して消費電力を得る。更に、read、idle 状態の時間とそれぞれの電力の積を取ることにより、消費電力量を算出した。NC、TPM、PAC という三種類の方式を比較したところ、何も省電力の動作を行っていない NC に対して、TPM は 18〜34 %、PAC は 35〜50 %の電力量削減が得られることを示した。

6.2 今後の課題

まず、実環境として全てのテーブルが一つのディスクに入っているものを構築した。よって、実際に複数のディスクにテーブルが分散されている環境で試してみる必要がある。また、今回はベンチマークのスケールファクタを小さくし、小さいテーブルや細かい読み込みを無視した計算を行っていたため、それらも考慮に入れた正確な実験が今後必要となる。

テーブルのディスク上の配置や、spin up/down の回数を抑える最適化を考えることによっても、更なる電力量の削減が得られるものと考えられる。また、シングルスレッドで一度に一つのディスクしかアクセスしていなかった点も、まだまだ展開の余地がある。

更に、今回の PAC では idle 期間の先読みしか行っておらず、TPM の閾値を小さくすることで PAC に近づいてしまい、PAC の長所が薄れてしまったが、PAC の能動制御に、spin up 時間を考慮した先行 spin up や、idle 時間の長短を考慮した spin down の判断など、より高機能なものを追加することにより、TPM との差別を図れる可能性がある。その際、ディスクアクセスがより複雑なワークロードを用いる必要がある。

最後に、5.1 で説明したように、モデルによる予測値と実測値が若干異なっていたので、これらの原因を明らかにし、正確なモデル化を行うことも課題となる。

文 献

[1] 合田 和生, 喜連川 優, “ログ転送を用いたディザスタリカバリシステムにおけるディスクストレージの省電力化方式の検討”, DEWS2007.

[2] S. W. Son, G. Chen, M. Kandemir, “Disk Layout Optimization for Reducing Energy Consumption”, ICS’ 05.

[3] E. Carrera, E. Pinheiro, and R. Bianchini. “Conserving disk energy in network servers.”, ICS’ 03.

[4] H. Yada, H. Ishioka, T. Yamakoshi, Y. Onuki, Y. Shimano, M. Uchida, H. Kanno, and N. Hayashi, “Head Positioning Servo and Data Channel for HDD ’s with Multiple Spindle Speeds”, IEEE TRANSACTIONS ON MAGNETICS, VOL. 36, NO. 5, SEPTEMBER 2000.

[5] Qingbo Zhu, Zhifeng Chen, Lin Tan, Yuanyuan Zhou, Kimberly Keeton, John Wilkes, “Hibernator: Helping Disk Arrays Sleep through the Winter”, SOSP ’05.

[6] Qingbo Zhu and Yuanyuan Zhou, “Power Aware Storage Cache Management”, Dept. of Computer Science Univ. of Illinois.

[7] White paper by Dennis Colarelli, Dirk Grunwald, Michael Neufeld, “The Case for Massive Arrays of Idle Disks (MAID)”, Dept. of Computer Science, Univ. of Colorado.

[8] White paper by Fred Moore, Aloke Guha, “Introducing COPAN Systems’ MAID Architecture (Massive Arrays of Idle Disks)”, HORIZON Information Strategies.

[9] White paper, “The Rise of MAID: A New Tier in Disk Storage”, TANEJA GROUP.

[10] D. Colarelli and D. Grunwald, “Massive arrays of idle disks for storage archives”, ICS’ 02.

[11] HGST Deskstar T7K250. http://www.hitachigst.com/portal/site/jp/menuitem.e9e85a2f0b51ab518797c532aac4f0a0/

[12] TPC-H. http://www.tpc.org/tpch/spec/tpch2.5.0.pdf

[13] MySQL. http://www.mysql.gr.jp/