社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

データインテンシブアプリケーションのI/O挙動解析評価とストレージ電力制御モデルの提案

西川　記史†　　中野美由紀†　　喜連川　優†

† 東京大学生産技術研究所
東京都目黒区駒場 4–6–1
E-mail: †{norifumi,miyuki,kitsure}@tkl.iis.u-tokyo.ac.jp

あらまし　我々は，アプリケーションと協調したストレージ省電力システムの構築を目指し，I/O 挙動特性に基づくストレージ電力制御モデルを提案する．データインテンシブアプリケーションの I/O 挙動特性を解析し，ストレージの省電力化に繋がる I/O パターンを抽出する．次に，抽出した I/O パターンの潜在的な省電力化可能性について検討し，アプリケーションの持つ情報と I/O 挙動特性を組み合わせることにより適切な省電力手法を選択するストレージ省電力制御モデルを提案する．

キーワード　ストレージ，省電力，電力制御モデル，データインテンシブアプリケーション，I/O 挙動解析

Storage Power Control Model based on Analyzing and Evaluating of I/O Behavior of Data Intensive Applications

Norifumi NISHIKAWA†, Miyuki NAKANO†, and Masaru KITSUREGAWA†

† Institute of Industrial Science, the University of Tokyo
4–6–1 Komaba, Meguro-ku, Tokyo
E-mail: †{norifumi,miyuki,kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract　We propose a storage power control model based on I/O behavior of data intensive applications. Our goal is to develop an application collaborative storage energy saving system. First, we analyze an I/O behavior of data intensive applications. Then we pick up I/O patterns that are useful to save energy of storages. We also discuss the power saving potentiality of the I/O pattern. Finally we propose a storage energy saving model which selects appropriate storage energy saving functions by combining both application information and I/O behaviors to storages.

Key words　storage, energy saving, power control model, data intensive application, I/O behavior analysis

1. はじめに

人類が生み出すデジタルデータは日々増加している．これらのデジタルデータはストレージに保持されており，生成されるデータ量の増加に伴いストレージの出荷量も年々増加している．文献 [1] によれば，2009 年からストレージの出荷容量は年率 49.8% のペースで増加し続けており，2014 年には 2009 年の約 7 倍以上のストレージが全世界に出荷されることとなる．これは，ストレージの運用における電力コストも増加することを意味している．

ストレージの消費電力は年率 19%のペースで増加し，2011 年には 2006 年の倍以上の電力コストが消費されると推定されている [2]．また，ストレージを含む全 IT 機器の利用コストのなかで電力コストは機器の導入コストの 75%程度の額にまで達するとの報告 [3] もある．IT 機器の運用時の消費電力削減，特に今後増大する一方のストレージの消費電力削減は重要な課題である．

データインテンシブアプリケーションが稼動する大規模ストレージの省電力化を効果的に行うには，ストレージ内に閉じた省電力制御のみではなくその上位に位置するアプリケーション層との協調が重要である．アプリケーションのワークロード，特に I/O の頻度や I／Oが行われるデータの局所性は TPC-C と TPC-H を比べるまでもなく，アプリケーションごとに顕著に異なる．また，アプリケーションの処理性能は実行時のユーザ数，データ量の増加等により変化するため，動的なワークロードに対して適応的に省電力調整を行う機構が必要である．

従来，ストレージの省電力化を目的とした研究が多数報告されている．その多くは，ストレージに対する I/O を監視し，I/O 発行の制御やデータ配置の制御を行うことによりストレージの省電力化の機会を作り出すものである [4]〜[11]．しかし，これらの手法はストレージ内部に実装される，あるいは，単純なストレージアクセスの統計値を用いるのみであり，アプリケーション情報を利用してはいない．

また，近年 DBMS によるストレージの省電力制御手法も提案されている [12], [13]．文献 [12] では DBMS においてもエネルギー効率を意識したチューニングが必要であることを述べている．また文献 [13] ではハードウェアの構成を変化させた場合の TPC-H の性能と消費電力の関係を評価している．しかし，文献 [12] ではデータセンタ運用開始後の実行時省電力化については述べられておらず，また文献 [13] は DBMS 稼働中の省電力化手法の省電力効果や性能への影響については述べられていない．

本論文では，アプリケーションと協調したストレージ省電力システムの構築を目指し，まず，今までに提案されたストレージの省電力手法を整理する．次に，それらの省電力手法を利用するための I/O パターンを明らかにする．そして，データインテンシブアプリケーションとして，オンライントランザクション処理 (OLTP)，意思決定支援システム (DSS)，及び大規模デジタルライブラリである Scientific Digital Library (DIAS) をとりあげ，それらの I/O 挙動解析を行うと同時にこれらインテンシブアプリケーションの潜在的な省電力化の可能性について検討する．最後に，これらの I/O 挙動特性を活用した，ストレージの電力制御モデルを提案する．

以下，2 章で関連研究を示し，3 章で省電力機能を有するストレージについて述べる．4 章でストレージの省電力機能を使うための I/O パターンについて述べ，5 章でデータインテンシブアプリケーションの I/O 挙動特性，及びデータインテンシブアプリケーションの省電力化の可能性を示す．6 章でストレージ電力制御モデルを提案し，7 章でまとめる．

2. 関 連 研 究

本章では，ストレージの省電力化に関する関連研勇について述べる．ストレージやディスクの省電力手法にはディスクあるいはストレージ自身が行う省電力化手法と，DBMS などのアプリケーションが行う手法がある．以下，これらについて述べる．

2.1 ストレージ省電力化手法

2.1.1 I/O 発行間隔制御

I/O 発行制御は，ディスクの省電力機能を使用する機会を増やすために，ディスクへの I/O の発行間隔を制御する [4]〜[7]．

これらの手法は，ストレージキャッシュなどの階層記憶を用いてディスクへの I/O の発行間隔を伸ばす．つまり read であれば先読みを行い cache にデータをロードし，write であれば cache に write されたデータのディスクへの更新の反映を遅延することにより，I/O 発行間隔を伸ばす．

データインテンシブアプリケーションへのこれらの手法適用について考える．まず OLTP であるが，OLTP はランダム I/O が行われるため I/O の発行先を知ることは困難である．このため先読みによりデータをストレージキャッシュにロードしておくこと難しい．DBMS の Write はトランザクションの実行とは非同期に実施されるため，Write 遅延を適用することによりディスクへの write 間隔を伸ばすことが期待できる．しかし，OLTP は小規模なものでもストレージに対して毎秒数回の read を発行しており，ディスクへの write 間隔を長くするのみではストレージの省電力化を行うために必要となる I/O 発行間隔を確保することは困難である．

DSS や DIAS はシーケンシャル read を行うため，先読みによりデータをストレージキャッシュにロードしておくことは可能である．しかし，従来の I/O 発行制御手法はアプリケーションの知識を用いないため，先読みにより省電力が期待できるデータがどれであるかを判断できない．従って，本手法だけでは大規模ストレージの省電力化には限界があると考えられる．

2.1.2 データ配置制御

データ配置制御は，ディスク上のデータの配置を調整することにより，ディスクの省電力機能が利用できるだけの I/O 発行間隔を生成する．具体的には，I/O 数の多いデータを同じディスクに配置し，I/O 数が少なくなった残りのディスクに省電力機能を適用する [5], [8]〜[11].

これらの手法は，いずれもストレージ内部で取得可能なブロック毎の I/O 数，あるいは OS で取得可能なファイル毎の I/O 数を用いてデータの配置制御を行う．しかし，省電力化で重要なのは単位時間当たりの I/O 数ではなく I/O 発行間隔の長さである．従来手法は I/O 発行間隔の長さについてはほとんど着目していない．このため，多くの I/O を発行するデータインテンシブアプリケーションに対して従来のデータ配置手法をそのまま適用しても，省電力化を効果的に行えない可能性が高い．

2.2 DBMS によるストレージ省電力手法

Harizopoulos らは，従来のハードウェアのみに閉じた省電力手法はソリューションの一部であり，データ管理ソフトウェアが大規模なデータセンタの省電力化に重要な役割を果たす可能性があることを示している [12]．具体的には，高性能を達成するアルゴリズムやハードウェア構成がエネルギー効率の観点では最適にはならない例を挙げ，DBMS においてもエネルギー効率を意識したチューニングやリソースの集約を考慮する必要があることを指摘している．また Poess らは，ストレージの構成 (ディスク台数，メディア) や CPU の省電力機能，メモリサイズを変化させた場合の TPC-H の性能と消費電力の関係を評価している [14].

しかし，1 章でも述べたように，文献 [13] はデータセンタが運用に入った後に発生する課題に対する解は示していない．また，文献 [12] では DBMS による省電力制御の定量的な評価や性能への影響については述べられていない．

3. 省電力機能を有するストレージ

本章では，省電力機能を有するストレージとその省電力機能および消費電力特性について述べる．

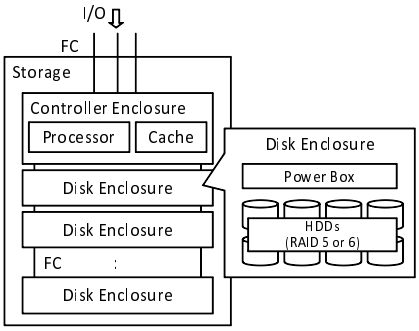

図 1　ストレージモデル
Fig. 1　Storage Model.

3.1　ストレージモデル

図 1 は，我々が省電力制御を行うストレージのモデルを示した図である．ストレージは 1 台以上のディスク筐体と 1 台のコントローラ筐体を持つ．ディスク筐体は 1 台以上の HDD を格納する筐体であり， Fibre Channel などのインタフェースによりコントローラと接続されている．またディスク筐体内には電源 box があり，内蔵している HDD に電力を供給している．ディスク筐体の HDD は RAID 構成 (例えば RAID5 や 6 等) を取る．

コントローラ筐体は，Fibre Channel などのインタフェース経由でサーバから I/O 要求を受け取り，ディスク筐体に対して I/O を発行する．コントローラ筐体は，I/O 要求の処理を行うプロセッサ及び，データを一時的に蓄えるバッテリバックアップされたストレージキャッシュを持つ．

3.2　ストレージ省電力機能

次に，ストレージの省電力機能について説明する．我々が省電力制御を行うストレージのモデルは，スタンバイ機能と電源 OFF 機能の 2 種類の省電力機能を有する．

スタンバイ機能　ディスク筐体内の全ての HDD の電源を OFF にする．以降，スタンバイ機能を使用している状態をスタンバイ状態と呼ぶ．

電源 OFF 機能　ディスク筐体の全ての HDD，及び電源 Box の電源を OFF にする．この状態を電源 OFF 状態と呼ぶ．

3.2.1　ストレージの消費電力

我々は，省電力機能を持つ商用のミッドレンジストレージを用いて，アイドル状態，スタンバイ状態，電源 OFF 状態の消費電力を計測した．計測に用いたストレージは日立製作所製の Adaptive Modular Storage 2500 である．ここで，アイドル状態とは，I/O が即時実行可能であるが I/O は行われていない状態である．計測の結果，アイドル状態の消費電力は約 257W，スタンバイ状態は 146W，電源 OFF 状態では 0W であった．

3.3　起動オーバヘッドと Break Even Time

スタンバイ状態あるいは電源 OFF 状態のディスク筐体を，アイドル状態に移行するために必要な電力と時間を起動オーバヘッドと呼ぶ．我々は，前節で消費電力の計測に用いたストレージを用いて起動オーバーヘッドを計測した．その結果を表 1 に示す．

表 1　起動オーバヘッド
Table 1　Spinup Overhead.

状態	起動エネルギー (W)	起動時間 (s)
スタンバイ	5,274	16
電源 OFF	13,245	69

ディスク筐体の省電力機能を用いると起動オーバーヘッドが発生するため，ディスク筐体の省電力機能を用いて消費電力を削減するためには，スタンバイ状態あるいは電源 OFF 状態をある程度の時間持続させる必要がある．この時間のことを Break Even Time と呼ぶ．起動に必要なエネルギーを E，ディスク筐体の省電力機能使用時とアイドル状態の消費電力の差を P とすると，Break Even Time T は，$T = E/P$ により求めることができる．我々が消費電力の計測に用いたストレージのディスク筐体では，スタンバイ状態の Break Even Time は約 27 秒，電源 OFF 状態の Break Even Time は約 51 秒であった．このことは，ディスク筐体のスタンバイ機能を使用するためにはディスク筐体への I/O 発行間隔が少なくとも 27 秒以上，電源 OFF 機能を使用するためには 51 秒なければならないことを示している．

4.　ストレージ省電力機能と I/O パターン

スタンバイ及び電源 OFF 機能を使用してディスク筐体の消費電力を削減するためには，ディスク筐体に対する I/O が Break Even Time 以上発行されないようにする必要がある．しかし，アプリケーションの I/O 挙動がその条件を満たさない場合は Break Even Time 以上の長さの I/O 発行間隔を積極的に作り出さなければならない．本章では，Break Even Time 以上の I/O 発行間隔を生成する手法と，それらの手法を用いることにより省電力化が期待できる I/O パターンについて述べる．

4.1　I/O 発行間隔の延伸方式

Break Even Time 以上の長さ以上の I/O 発行間隔を積極的に作り出す手法には，大きく I/O 発行間隔制御とデータ配置制御がある．I/O 発行間隔制御において I/O の発行間隔を伸ばす手法には，先読み，キャッシュ保持，及び Write 遅延がある．

先読み　Read されることが分かっているデータを事前に一括してストレージキャッシュに読み込み，アプリケーションからデータ read 要求があった場合はキャッシュされたデータをアプリケーションに渡すことにより，ディスク筐体への I/O 発行間隔を伸ばす．

キャッシュ保持　Break Even Time 以上の時間が経過してから再度 read されるデータをストレージキャッシュに保持し，アプリケーションから再 read 要求があった場合はキャッシュされたデータをアプリケーションに渡すことにより，ディスク筐体への I/O をなくすことで I/O 発行間隔を伸ばす手法である．本手法は，既に Break Even Time より長い I/O 発行間隔をさらに伸ばすための手法である．

Write 遅延　アプリケーションが更新したデータをストレージのキャッシュに保持しディスク筐体への反映を遅延することにより，ディスク筐体の I/O 発行間隔を伸ばす．

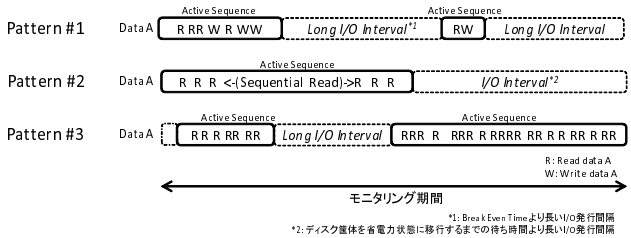

図 2　I/O 間隔の延伸が可能な I/O のパターン
Fig. 2　I/O Interval Enlargeable I/O Pattern.

データ配置制御　データ配置制御は，複数台のディスク筐体を使って I/O 発行間隔を伸ばす方式である．Break Even Time より長い I/O 発行間隔を持たないデータを同一のディスク筐体に配置することにより，他のディスク筐体に対する I/O の発行間隔を伸ばす．

4.2　ストレージ省電力化が可能な I/O パターン

次に，ディスク筐体の省電力機能 (スタンバイ及び電源 OFF)，及び I/O 発行間隔延伸方式の使用が可能な I/O パターンについて整理する．これらの I/O パターンを図 2 に示す．

ディスク筐体省電力機能の使用が可能な I/O パターン: スタンバイあるいは電源 OFF 機能を使ってディスク筐体の省電力化を行うためには，ディスク筐体をに対する I/O 発行間隔が Break Even Time 以上でなければならない．このため，ディスク筐体省電力機能の使用が可能な I/O パターンとは Break Even Time より長い I/O 発行間隔を含むものである (図 2 の I/O Pattern #1)．

先読みが可能な I/O パターン: 先読みを行うためには，現時点から Break Even Time の長さの間に read されるデータを知っている，あるいは予測出来る必要がある．Read を予測できる I/O パターンは Sequential Read のみのため，これが先読みにより I/O 発行間隔を伸ばすことが可能な I/O パターンである (図 2 の I/O Pattern #2)．

キャッシュ保持が可能な I/O パターン: キャッシュ保持は既に Break Even Time より長い I/O 発行間隔をさらに伸ばすための手法である．このため，キャッシュ保持により I/O 発行間隔を伸ばすことが可能な I/O パターンとは，Break Even Time 以上の間をおいて同一データに対する read を行う I/O パターンである (図 2 の I/O Pattern #3)．

Write 遅延が可能な I/O パターン: Write 遅延は Write が行われる I/O パターンであれば適用可能である．

データ配置制御が可能な I/O パターン: データ配置制御を行うためには，Break Even Time より長い I/O 発行間隔がなければならない．このため，データ配置制御が可能な I/O パターンとは Break Even Time より長い I/O 発行間隔を持つ I/O パターンである (図 2 の I/O Pattern #1)．

5.　データインテンシブアプリケーションの I/O 挙動特性と省電力化の可能性

本章では，データインテンシブアプリケーションの I/O 挙動を計測し，その解析を行う．さらに，得られた I/O 挙動特性に基づき，ディスク筐体の省電力化の可能性について述べる．

5.1　計測環境

計測に用いたサーバは日立製作所の SR16000，ストレージは同じく日立製作所の Adaptive Modular Storage 2500 である．ストレージは，4 つの I/O プロセッサ及び 2GB のバッテリバックアップされたキャッシュを有するコントローラ筐体 1 台，及び 15 台の HDD を格納するディスク筐体を 10 台有している．ディスク筐体内の 15 台の HDD はデータ用 HDD 13 台，パリティ用 HDD 2 台の RAID6 構成である．HDD は 7200 回転の SATA ドライブであり，容量は 750GB である．1 ディスク筐体当りの記憶容量は RAID 構成前で 11.25TB である．

サーバは 2 コアのプロセッサを 32 台 (合計 64 コア) と 512GB の主記憶を有している．サーバの OS は AIX5.3 64-bit 版，ファイルシステムは JFS2 である．サーバとストレージは 4 本の 4Gbit ファイバチャネルで接続されている．

データインテンシブアプリケーションとして，我々は OTLP(TPC-C)，DSS(TPC-H)，及び Scientific Digital Library (DIAS) を取り上げた．これらを先のシステム上で稼働させ，I/O トレースを取得した．I/O トレースの取得には，OS に付属の I/O トレース取得ツールを用いた．各アプリケーションの構成および計測条件は以下の通りである．

OLTP(TPC-C)　：TPC-C の DB サイズを示す Warehouse 数は 5,000，(データ量約 500GB)，同時実行スレッド数は 1,000，Think Time 及び Keying Time はともに 0 秒である．DBMS のキャッシュは 25GB，計測期間はトランザクションスループットが安定してから 1 時間である．ログをディスク筐体#1 に，DB をディスク筐体#2〜#10 にハッシュ分割して配置した．DB の表及び索引のパーティションを構成する DB ファイルの容量は最大 30GB である．

DSS(TPC-H)　：TPC-H の DB サイズを示すスケールファクタは 100 である (データ量約 100GB)．クエリ#1 から#22 を単一スレッドで順次実行し，I/O 挙動を計測した．計測期間は約 6 時間である．DBMS のキャッシュは 5GB である．ログをディスク筐体#1 に，DB をディスク筐体#2〜#9 にハッシュ分割して配置した．DB の表及び索引のパーティションを構成する DB ファイルの容量は最大 30GB である．

DIAS　：数名のユーザがデータインテンシブアプリケーションを実際に使用している状況下で I/O トレースを取得した．計測期間は 24 時間 (2010 年 5 月)，データ量は約 85TB，OS のバッファサイズは 512GB である．ファイルの平均サイズは約 2GB である．

5.2　ディスク筐体省電力機能の使用可能性

我々は，まずディスク筐体の省電力機能を使用できる可能性がどの程度あるかをまず調査した．TPC-C, TPC-H, DIAS の各アプリケーション毎にディスク筐体毎の I/O トレースを取得し，Break Even Time より長い I/O 発行間隔の有無を調べた (図 2 の Pattern #1 に対応)．この結果を図 3 に示す．

図から分かるように，TPC-C が使用するディスク筐体は省電力機能を使用できるだけの長さの I/O 発行間隔がなかった．逆に，TPC-H では全てのディスク筐体で，スタンバイ及び電源

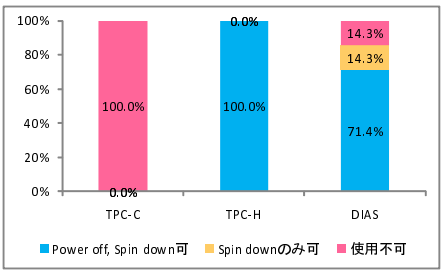

図 3　ディスク筐体毎の省電力機能使用可能性
Fig. 3　Potentiality of Power off and Spindown of Disk Enclosures.

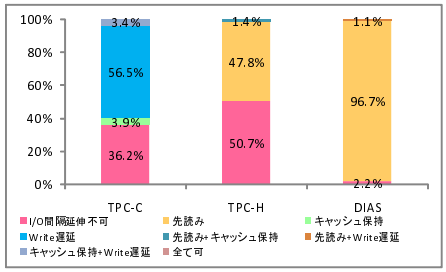

図 4　I/O 発行制御による省電力化の可能性
Fig. 4　Power Saving Potentiality of I/O Interval Control

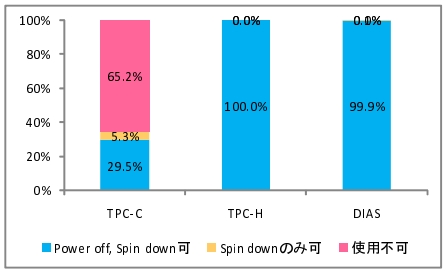

図 5　データ配置制御による省電力化の可能性
Fig. 5　Power Saving Potentiality by Data Placement Control

OFF 機能の使用が可能な長さ (Break Even Time 以上) の I/O 発行間隔があった．DIAS では 14.3%のディスク筐体はディスク筐体の省電力機能を使うことはできず，別の 14.3%のディスク筐体はスタンバイ機能の使用が可能であり，残りの 71.4%はスタンバイ及び電源 OFF 双方の機能の使用が可能なだけの長さの I/O 発行間隔が見られた．

5.3　I/O 発行制御による I/O 発行間隔の延伸の可能性

次に，I/O 発行制御に用いられる先読み，キャッシュ保持，及び Write 遅延の各手法により，I/O 発行間隔を伸ばすことが可能かどうかを調べるために，我々は図 2 の Pattern #2, #3 に相当するパターン，及びデータに対する Write の有無を調査した．これらはディスク筐体単位ではなくアプリケーションが認識するデータ単位で行う．我々は，TPC-C 及び TPC-H では DB ファイルを，DIAS では OS ファイルをそれぞれデータの単位とした．

我々は，read が行われると後続のアドレスのデータを常に先読みする場合に，先読みのヒット率が 50%以上となる I/O パターンを持つデータを，先読みにより I/O 発行間隔を伸ばすことが可能なデータとした．また，前回 read 完了後あるいは計測期間開始後から Break Even Time 以上の期間をおいて read が行われるデータを，キャッシュ保持により I/O 発行間隔を伸ばすことが可能なデータとした．Write が行われるデータを Write 遅延可能なデータとした．各アプリケーションにおいて，I/O 発行制御の使用が可能なデータ数の比率を図 4 に示す．

図 4 より，TPC-C では，DB ファイルの 56.5%が Write 遅延，3.9%がキャッシュ保持により，3.4%が Write 遅延とキャッシュ保持の双方の手法で I/O 発行間隔を伸ばすことができる可能性があることが分かる．しかし，36.2%の DB ファイルはどの手法を用いても I/O 発行間隔を伸ばすことができない．また先読みにより I/O 発行間隔を伸ばせる可能性があるデータは存在していない．

TPC-H では，47.8%が先読み，1.4%が先読み+キャッシュ保持により I/O 発行間隔を伸ばせる可能性があることが分かる．しかし，50.7%のデータはどの手法を用いても I/O 発行間隔を伸ばすことができない．

DIAS では 96.7%のデータが先読みにより I/O 発行間隔を伸ばすことができ，1.1%が先読みと Write 遅延を用いることで I/O 発行間隔を伸ばせる可能性があることが分かる．

これらの結果より，アプリケーションの種類により，I/O パターンは大きく異なっており，ストレージの省電力化により使用できる I/O 発行制御機能は非常に限られていること言える．これらの結果はす，アプリケーションが持つ情報を用いることにより適切な I/O 発行制御手法を選択することが，ストレージの省電力化にとって重要であることを示している．

5.4　Write 遅延による省電力化の可能性

Write が行われるデータが Write 遅延により省電力化できる可能性のあるデータである．データの単位は先読み及びキャッシュ保持と同じである．Write が行われるデータの比率は，TPC-C が 79.7%，TPC-H が 0.0%，DIAS が 3.0% であった．

5.5　データ配置制御による省電力化の可能性

図 5 は，データ毎の I/O 発行間隔のうち，Break Even Time より長い I/O 発行間隔を含むデータの比率を示したものである．データの単位は先読み及びキャッシュ保持と同じである．図より，TPC-C においても 29.5%，TPC-H と DIAS においてはほぼ 100%のデータで電源 OFF 及び Spin down 両方の省電力機能を使うことができることが分かる．これは，TPC-C においてもデータ配置制御を用いることにより，ディスク筐体の省電力機能を使用できる可能性があることを示している．

6.　ストレージ省電力モデルの提案

前章までの観測結果を元に，ストレージの省電力化をおこなうための電力制御モデルを提案する．本モデルの特徴は，ストレージに対する I/O 挙動のモニタリングに加え，アプリケーションの情報を取得し，その情報をストレージの省電力制御に用いる点である．本モデルは，I/O スケジューリング及びデータ配置制御を行うことにより，ディスク筐体に対する I/O 発行間隔を伸ばすことによりディスク筐体の省電力機能を使用できる機会を増やす．我々が提案する省電力モデルを，図 6 に示す．

モニタリング　アプリケーションの情報，及びアプリケーションがストレージに対して発行した I/O 挙動をモニタリングする．アプリケーション情報には，アプリケーション名やアプリケーションが利用するデータ (OS ファイル，DB の表や索引，パーティション，ファイルなどの DB オブジェクトなど) の名称，それらのディスク筐体上への配置などの静的な情報を含む．

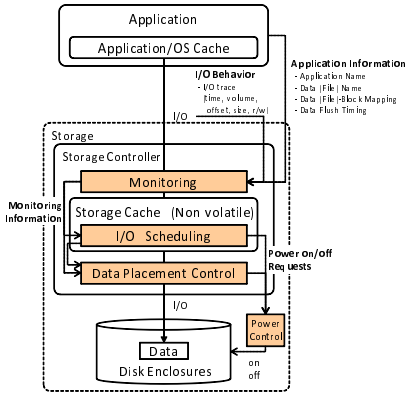

図 6　電力制御モデル
Fig. 6　Storage Power Saving Model

また，アプリケーションが独自のキャッシュを持つ，あるいは OS のキャッシュを持つ場合はキャッシュのフラッシュの契機をモニタリングする．

I/O 挙動のモニタリングは，アプリケーションがストレージに対して発行した I/O トレースを取得することにより行う．I/O トレースには時刻，I/O 先のディスク筐体名やオフセット，I/O サイズを含む．アプリケーションの知識を用いることにより，アプリケーションが認識するデータ単位での制御が可能となる．これにより，I/O 挙動特性が類似している範囲を，ストレージから得られる情報だけを用いる場合と比較して精度よく把握することが可能となり，省電力化の効率の向上が期待できる．モニタリングは，ストレージのコントローラで行う．

I/O スケジューリング　アプリケーションが認識するデータの単位でシーケンシャル Read の比率，I/O 発行間隔，及び write の有無を求める．その後，本機能はこれらの結果に基づき先読みやキャッシュ保持を行うデータを決定し，それらの先読みとキャッシュ保持を行う．また，アプリケーションキャッシュあるいは OS キャッシュのフラッシュの契機を受け取ると，ディスク筐体の電源を投入し，ストレージキャッシュに保持していた更新をディスク筐体にフラッシュする (Write 遅延)．アルゴリズムの設計は今後の課題である．

データ配置制御　データ配置制御はモニタリングにより取得した I/O トレースを用いてデータ毎の I/O 発行間隔を調査する．そして，Break Even Time より長い I/O 発行間隔を持たないデータを同一のディスク筐体の集める．また，Break Even Time より長い I/O 発行間隔を持つデータについても，I/O 発行間隔ができるだけ長くなるようデータを配置する．I/O スケジューリングと同様，アルゴリズムの設計は今後の課題である．

7. ま と め

本論文では，アプリケーションと協調したストレージ省電力システムの構築を目指した，I/O 挙動特性に基づくストレージの省電力化を目的として，ストレージ省電力機能の整理とそれらの機能を使うことが可能な I/O のパターンを整理した．次に，データインテンシブアプリケーションとして OLTP, DSS, 及びデジタルライブラリを取り上げ，これらのアプリケーションの I/O 挙動特性を解析を行った．ストレージの省電力化の潜在的な省電力化の可能性についての検討を行い，アプリケーションが認識するデータを単位として省電力制御を行うことにより，全てのアプリケーションで省電力化が期待できること，及びアプリケーションの種類により適切な省電力制御機能を選択する必要があることを示した．また，我々はアプリケーションの情報を用いたストレージ省電力制御モデルを提案し，その概要を示した．

今後は提案したモデルの詳細化及び実装を進め，データインテンシブアプリケーション向けの大規模ストレージの省電力化機構を完成させる予定である．

文　献

[1] R.L.V. Natalya Yezhkova, “Worldwide enterprise storage systems 2010?2014 forecast update,” IDC White Paper # 226223, 2010.
[2] D. Reinsel, “The real costs to power and cool all the world’s external storage,” IDC White Paper # 212714, 2008.
[3] M. Eastwood, S.J. Bozman, C.J. Pucciarelli, and R. Perry, “The business value of consolidating on power-efficient servers,” IDC White Paper # 218185, 2009. http://uk.logicalis.com/pdf/IDC_White_paper_energy.pdf
[4] A.E. Papathanasiou and M.L. Scott, “Energy efficient prefetching and caching,” Proc. of USENIX 2004 Annual Technical Conference, pp.255–268, USENIX Association Berkeley, 2004.
[5] D. Li and J. Wang, “Eeraid: Power efficient redundant and inexpensive disk arrays,” Proc. 11th Workshop on ACM SIGOPS European Workshop, pp.174–180, 2004.
[6] X. Yao and J. Wang, “Rimac: A novel redundancy based hierarchical cache architecture for power efficient,” High Performance Storage System Proc. 2006 EuroSys Conference, pp.249–262, 2006.
[7] T. Heath, E. Pinheiro, J. Hom, U. Kremer, and R. Bianchini, “Application transformations for power and performance-aware device management,” 11th International Conference on Parallel Architectures and Compilation Techniques, pp.121–130, 2002.
[8] D. Colarelli and D. Grunwald, “Massive arrays of idle disks for storage archives,” Supercomputing, ACM /IEEE 2002 Conference, pp.47–57, 2002.
[9] E. Pinheiro and R. Bianchini, “Energy conservation techniques for disk array based servers,” Proc. 18th Annual International Conference on Supercomputing, pp.68–78, ACM, 2004.
[10] O.M.Q.J. Weddle, C. and A.A. Wang, “Paraid: A gear-shifting power-aware raid,” 5th USENIX Conference on File and Storage, pp.245–267, USENIX Association, 2007.
[11] K.R.U.L. Verma, A. and R. Rangaswami, “Srcmap: Energy proportional storage using dynamic consolidation,” 8th USENIX Conference on File and Storage Technologies, pp.267–280, USENIX Association, 2010.
[12] S. Harizopoulos, M.A. Shah, J. Meza, and P. Ranganathan, “Energy efficiency: The new holy grail of data management systems research,” 4th Biennial Conf. on Innovative Data Systems, pp.112–123, 2009.
[13] M. Poess and R.O. Nambiar, “Tuning servers, storage and database for power efficient data warehouse,” 26th IEEE International Conf. on Data Engineering, pp.1006–1017, IEEE Computer Society, 2010.
[14] M. Poess and R.O. Nambiar, “Power cost, the key challenge of today’s data centers: a power consumption analysis of tpc-c results,” VLDB ’08 Proceedings, pp.1229–1240, ACM Press, 2008.