アプリケーション協調型大規模ストレージ省電力システムの開発と評価

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　デジタルデータの急増に伴い，データセンタの運用コストは増加の一途を辿っている．特にストレージの電力コストの成長率は他のコストの成長率を圧倒しており，その削減はデータセンタの運用管理の最重要課題となっている．我々は，これまでアプリケーションの入出力挙動特性をストレージの省電力運転に活用する手法を提案している．本論文では，提案手法をストレージ省電力管理システムとして実装すると共にその実装の詳細について述べる．さらに OLTP 及び DSS をストレージ省電力管理システム上で稼働させ，ストレージの消費電力とアプリケーション性能を計測し，提案手法が有効に動作することを確認する．

キーワード　ストレージ，省電力制御システム，アプリケーション入出力挙動，OLTP，DSS

Development and Evaluation of Storage Power Control System utilizing 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　Datacenter's management cost are increased day by day according to rapid growth of digital data. Especially, the growth of power cost of storages overwhelms that of other IT equipments. Thus, reducing the power cost of storages is the most important problems at datacenters. We have been developed a novel storage power control system which utilizes application level I/O behaviors for saving a power consumption of storages. In this paper, we discuss an implementation of our storage power control system. We also run OLTP and DSS on our storage power control system, and evaluate our system. We measure power consumption of storages and application performance, and show our system has an enable to save a power consumption of large storage systems.

Key words　storage, power control system, application I/O behavior, OLTP, DSS

1. はじめに

人類が生成するデジタルデータの量は日々増加している．IDC のレポート [4] によれば，電子的に生成され蓄積される情報及びコンテンツの量は，2015 年には 7 ゼッタバイトを超えると予測されている．これら爆発的に増加するデジタルデータは大規模なストレージに格納され，センサデータアーカイブや検索エンジン，顧客情報管理 (オンライントランザクション処理システム) などのデータインテンシブアプリケーションにより管理・利用されている．このため，今後ストレージ出荷容量が急増することが予想されている [17].

一方，データセンタにおける IT 機器の電力消費量の増加は著しく [18]，その省電力が強く求められている．デジタルデータの増加とも相まって急増するストレージの消費電力の削減は，データセンタにおける最重要の課題の一つとなっている．

これまでにも多くのストレージ省電力手法が報告されている [2], [3], [5]〜[7], [11]〜[16]．これらの手法は，いずれもストレージレベルの入出力頻度に従いディスクを停止する等の手法により省電力を実現している．しかし，実際に入出力を発行するのはアプリケーションであり，ストレージレベルの入出力情報のみでは的確に省電力制御することは難しい．このため，アプリケーションが長期間入出力を行わない場合でもストレージを稼働し続けた結果電力を削減できない，あるいはアプリケーションが短期間で入出力処理を再開するにも関わらずストレージを省電力状態に移行してしまいアプリケーションの性能劣化を引き起こす等の可能性がある．

データセンタでは，主にオンライントランザクション処理 (OLTP) や意思決定支援システム (DSS) など，多くの入出力を伴うデータインテンシブアプリケーションが稼働している．これらアプリケーションの入出力の特性は，例えば OLTP ではランダムな入出力，DSS では一括読み取りなど，アプリケーション毎に異なる．我々は，アプリケーションの入出力挙動特性を用いることによりストレージに対する入出力の傾向を把握し，それをストレージの省電力に利用する新たな省電力手法を提案している [8]〜[10]．これらの提案手法はアプリケーションの入出力挙動に合わせてストレージの省電力を実行するため，実行中のアプリケーションの性能の劣化を抑えつつ，従来の手法と比較し高い省電力効果を得ることが期待できる．

本論文では，まず我々が開発した実行時ストレージ省電力フレームワークの概要を示す．我々が提案するフレームワークの特長は，i) アプリケーション実行時のストレージ省電力化，ii) アプリケーションレベルにおける入出力発行間隔の長さや read/write 入出力の頻度等のモニタリング結果に基づくアプリケーションレベルでの入出力挙動のパターン化，及び iii) アプリケーションレベルの入出力挙動のパターンに基づく，適切なストレージ省電力手法の選択及び適用，である．さらに我々は，本フレームワークをを商用ストレージ及び Linux サーバから構成されるシステム上に実装し，OLTP 及び DSS を動作させ評価する．

以下，2 章においてアプリケーションの入出力挙動特性を用いたストレージ省電力の概要を示し，3 章にて本手法の実装について述べる．さらに，4 章において OLTP 及び DSS を用いた評価について述べ，5 章においてまとめる．

2. アプリケーション協調型ストレージ省電力

大規模なデータセンタで稼働するデータインテンシブアプリケーションは，それぞれ固有の入出力挙動を持っている．例えば，E-comemrce やインターネットバンキングシステムなどの OLTP はマスタテーブルにランダム入出力を発行するとともにトランザクションテーブルにレコードを順次追加する．ストリーミングメディアは大きなビデオデータをシーケンシャルに読み出す．このようなアプリケーション固有の入出力挙動は複数のアプリケーションが同時に稼働しているデータセンタ等ではストレージデバイスレベルの入出力挙動のみを監視しても得ることはできない．アプリケーション固有の入出力挙動をストレージの省電力に使用することができれば，従来のストレージデバイスレベルの入出力に基づく省電力手法と比較してストレージデバイスに対する入出力挙動の傾向をより正確に把握することができ，アプリケーションの性能を維持しつつストレージの消費電力をより効率的に削減できる可能性が高まる．本章では，アプリケーションの入出力挙動特性を利用して MAID 機能を有するストレージデバイスの省電力を行う実行時ストレージ省電力フレームワークを提案する．

2.1 MAID 機能を利用した実行時ストレージ省電力フレームワーク

図 1 に，本研究において提案する省電力フレームワークを示す．図の左半分は，アプリケーションからストレージデバイスまでの入出力の経路を示している．

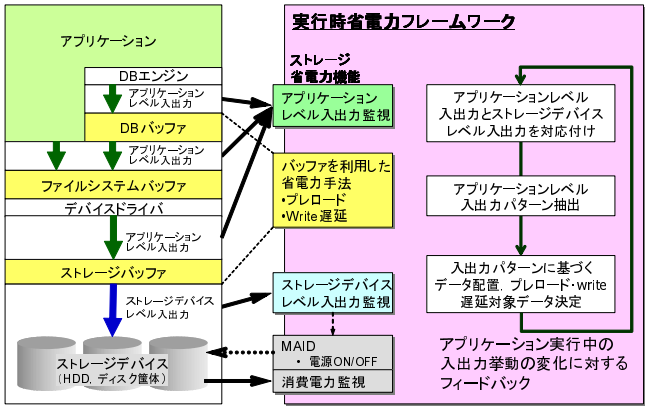

図 1　実行時ストレージ省電力フレームワーク
Fig. 1　Runtime Storage Power Saving Framework.

近年の DB サーバやファイルシステム，ストレージは大規模なバッファを持つが，このバッファはストレージデバイスの省電力に有用である．そこで，アプリケーションとストレージデバイス間に存在するバッファを省電力においても利用する．バッファは，DB バッファや FS キャッシュ，あるいはストレージキャッシュ層を含み，省電力対象のストレージデバイスによりどの層に組み込まれるかが異なる．

アプリケーションは，バッファに対してアプリケーションレベルの入出力を行う．アプリケーションレベルの入出力とは，アプリケーションが認識するデータであるファイルや DB の表・索引に対する入出力のことである．バッファはアプリケーションレベルのデータを受け取り，物理ブロックへと分解し，ストレージデバイスに対して，ストレージデバイスレベルの入出力を行う．

ストレージデバイスとは，HDD やストレージのディスク筐体などの電源 ON/OFF の単位となるデバイスのことである．ストレージデバイスレベルの入出力とは，ストレージデバイス内の物理ブロックに対する入出力のことである．

図の右半分は，実行時ストレージ省電力フレームワークと左半分の各レイヤが持つ実行時ストレージ省電力に関する機能を示している．

アプリケーションレベル入出力監視機能は，アプリケーションレベルの入出力挙動をストレージ省電力に用いるために，アプリケーションレベルの入出力を監視する．ストレージデバイスレベル入出力監視機能は，MAID 制御に必要な情報を得るために，ストレージデバイスレベルの入出力を監視する．

バッファを利用した省電力手法は，フレームワークからの指示に従い，バッファリングを利用した実行時ストレージ省電力であるプレロード及び write 遅延を行う．バッファを利用した省電力手法は，アプリケーションやストレージデバイスの種類に応じ，DB バッファ，ファイルシステムバッファ，あるいはストレージバッファの何れかに組み込まれる．

実行時ストレージ省電力フレームワークは，アプリケーションレベル入出力監視及びストレージデバイスレベル入出力監視

機能より得た入出力を対応付けると共に，アプリケーションレベル入出力挙動から入出力パターンを抽出する．そして，入出力パターンに基づきデータ配置の決定やデータ配置対象，プレロード対象のデータ決定などの省電力手法の選択を行う．

アプリケーションの入出力挙動特性は，データの更新や追加，ユーザ数の増加などに伴い，時間と共に変化する．このため，本フレームワークはアプリケーションレベル及びストレージデバイスレベルの入出力を監視し，アプリケーションレベル入出力パターンの変化を調査する．そしてアプリケーションレベル入出力パターンが変化した場合には，データ配置やプレロード対象，write 遅延対象データの選択などの省電力手法を再度選択する．これにより，アプリケーションの入出力挙動の変化に追随する．

また，ストレージデバイスレベル入出力監視機能は，ストレージデバイスレベル入出力を監視し，ストレージデバイスに対して入出力が行われていない場合にはストレージデバイスの電源を OFF にする．

2.2 論理入出力パターン

ストレージの省電力にアプリケーションの入出力挙動特性を用いるために，論理入出力パターンという概念を導入する．論理入出力パターンとは，アプリケーションの入出力挙動をストレージの省電力に適するように分類・パターン化したものであり，ストレージの省電力機能を適切に選択するために使用する指標である．アプリケーションの入出力挙動を論理入出力パターンを用いて識別することにより，アプリケーション実行中のストレージ省電力が可能な入出力挙動を容易に抽出でき，アプリケーション実行時の高いストレージ省電力を達成することが可能になると考えられる．

2.2.1 ストレージ省電力の単位

ストレージの省電力の単位として，ストレージ全体，ディスク筐体，及び HDD の 3 種類が考えられる．ストレージ全体の電源を OFF にする場合は，そのストレージを使用する全てのアプリケーションを停止しなければならない．さらに，ストレージ全体を起動するには非常に長い時間を要する．そのため，ストレージ単位の電源 OFF はアプリケーション実行時の省電力には不適切である．次に，HDD 単位の省電力について考える．多くのストレージは，ディスク筐体内の HDD を用いて RAID を構成する．ディスク筐体に対して発行された入出力は，ディスク筐体内の HDD に均等に発行される．これは HDD レベルの入出力挙動が，ストレージ省電力の観点ではディスク筐体レベルの入出力挙動と類似したものになる，すなわち，HDD の ON/OFF の契機はディスク筐体のそれとほぼ同じになることを示している．このため，HDD レベルの省電力は RAID を構成する個々の HDD の制御が必要なため管理が煩雑になるにも関わらず，その省電力効果はディスク筐体単位の省電力とほぼ同等になると考えられる．これらの理由により，ディスク筐体単位の省電力を選択する．

ストレージはバッテリバックアップされた大規模なキャッシュを持つ．ストレージ省電力手法は，ディスク筐体に対する入出力発行間隔を伸ばすためにストレージキャッシュを使用する．

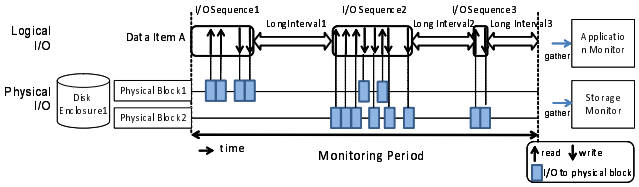

図 2　ロングインターバルと入出力シーケンス
Fig. 2　Long Interval and I/O Sequence.

ストレージキャッシュを用いて入出力間隔を伸ばす手法には，データをストレージキャッシュにプリロードし read 入出力間隔を伸ばす手法，及びディスク筐体への write 入出力を遅延することにより write 入出力間隔を伸ばす手法 (write 遅延) が考えられる．ディスク筐体に対する入出力発行間隔を伸ばすために，プリロードと write 遅延の双方を使用する．

2.2.2 データアイテムと論理入出力パターン

これまで述べたように，アプリケーションの入出力挙動はアプリケーション毎に大きく異なる．高いストレージ省電力効果を得るためには，アプリケーションの入出力挙動の違いを十分に考慮した上でストレージの省電力機能を選択・適用しなければならない．アプリケーションの入出力挙動をストレージの入出力挙動と結びつけストレージの省電力に取り入れるために，データアイテムと論理入出力パターンを導入する．

データアイテム　データアイテムとは，アプリケーションが使用するデータを，データが配置されているディスク筐体単位に切り分けたものである．データの単位はアプリケーション毎に異なる．OLTP や DSS など DBMS を利用するアプリケーションでは，データの単位はデータベースの表や索引である．ファイルサーバ上で動作するアプリケーションでは，データの単位はファイルである．データが複数のディスク筐体上に配置されている場合，それらは異なるデータアイテムである．データをデータアイテムに分割することにより，ディスク筐体上でのアプリケーションの入出力挙動を識別することが可能となる．

論理入出力パターン　論理入出力パターンとは，アプリケーションがデータアイテムに対して発行した入出力挙動をパターン化したものであり，適切な省電力機能を選択するための指標である．論理入出力パターンを識別するために，ロングインターバルと入出力シーケンスを導入する．ロングインターバルとはストレージを省電力状態に移行することにより消費電力を削減できる時間間隔 (Break Even Time と呼ぶ) より長い入出力発行間隔のことである．入出力シーケンスは，データアイテムに対するいくつかの read あるいは write 入出力と Break Even Time より短い入出力間隔から構成される一連の入出力のことである．

図 2 はデータアイテム A のロングインターバルと入出力シーケンスの例を示している．データアイテム A は，3 つのロングインターバルと 3 つの入出力シーケンスを有している．ロングインターバル#3 はモニタリング期間の終了と同時に終了している．入出力シーケンス#1 はモニタリング期間の開始時点から開始している．図 2 に示すように，論理レベルの入出力挙動

と物理レベルの入出力を組み合わせることにより，データアイテム A がディスク筐体#1 上に配置されていることを知ることができる．

アプリケーションレベルの入出力挙動をストレージレベルの入出力挙動と結びつけ省電力に活用することが可能となっても，個々のアプリケーション毎の入出力挙動に基づいて適切な省電力方式を決定することは依然非効率である．そこで，データアイテムを区別するための 4 種類の論理入出力パターンを導入する．これにより，個々のアプリケーションとは独立に，データアイテムに適した省電力管理を容易に選択することが可能となる．論理入出力パターンの定義を以下に示す．

- **入出力パターン P0**　モニタリング期間中，一度も入出力が発行されなかったデータアイテムを識別するための論理入出力パターンである．この論理入出力パターンは単一のロングインターバルのみを含み，入出力シーケンスは含まない．これらのデータアイテムは，電源 OFF 機能を適用するディスク筐体に配置する候補となる．
- **入出力パターン P1**　少なくとも一つのロングインターバルを含み，かつ入出力シーケンス内の合計 read 数が合計入出力数の 50%以上である．P1 に分類されるデータアイテムは read が多いため，これらのデータアイテムはストレージキャッシュへのプレロードの候補となる．
- **入出力パターン P2**　少なくとも一つのロングインターバルを含み，かつ入出力シーケンス内の合計 write 数が合計入出力数の 50%未満である．P2 に分類されるデータアイテムは write 数が多い．このためこれらのデータアイテムはディスク筐体への write 入出力を遅延させることにより write 入出力間隔を伸ばす，write 遅延の適用候補となる．
- **入出力パターン P3**　単一の入出力シーケンスのみを持ち，ロングインターバルを持たない，すなわち，全ての入出力間隔が Break Even Time より短い入出力パターンである．P3 に分類されるデータアイテムはロングインターバルを持たないため，省電力機能の適用対象外の候補である．

本研究では論理入出力パターンを 4 種類のみに分類している．本研究で提案する実行時ストレージ省電力フレームワークでは，ストレージは MAID 機能を持ち，キャッシュ層でデータ移動，プレロード，write 遅延を行う．このうち単一のディスク筐体の省電力に利用できる MAID，プレロード，write 遅延をまず考え，これらの省電力手法に適した P0，P1，P2 の 3 種類の論理入出力パターンを抽出した．しかし，データアイテムの中には常時入出力が発行され省電力に適さないものも存在する．高い省電力効果を得るためには，電源 OFF を適用するディスク筐体にこのようなデータアイテムを配置しないことが重要である．最後の入出力パターンである P3 は，このような省電力に適さないデータアイテムを識別するためのものである．

2.3　ストレージ省電力方式

図 1 に示したように，提案フレームワークは論理入出力パターンを分析し，その統計情報に基づいてディスク筐体に適切な省電力手法を適用する．ストレージ電力管理システムは，データ配置制御，及びストレージキャッシュを用いた入出力発行間隔制御を行う．

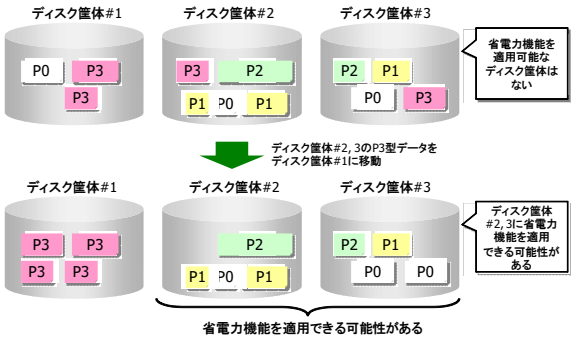

図 3　データ配置制御の効果
Fig. 3　An Effect of Data Placement.

2.3.1　データ配置制御

ディスク筐体に電源 OFF 機能を適用するには，ディスク筐体に対するいくつかの入出力発行間隔は Break Even Time より長くなければならない．このために，データ配置手法は P3 型のデータアイテムを同一のディスク筐体に配置し，残りのディスク筐体の入出力発行間隔を Break Even Time より長くすることを試みる．図 3 に示すように，ディスク筐体#2，#3 上の P3 型のデータアイテムをディスク筐体#1 に移動することにより，ディスク筐体#2，#3 に対する入出力発行間隔を Break Even Time 以上にできる可能性が高まる．

2.3.2　ストレージキャッシュを用いた入出力発行間隔制御

P0, P1 及び P2 型のデータアイテムのみを格納したディスク筐体は，長時間電源を OFF にすることが期待できる．しかし，データアイテムの入出力契機はデータアイテムごとに異なるため，ディスク筐体の入出力発行間隔は通常は個々のデータアイテムに対する入出力発行間隔より短くなる．そこで，エンタープライズストレージが RAID コントローラ内にバッテリーバックアップされた大容量のキャッシュを有している [1] ことを利用してディスク筐体に対する入出力発行間隔を伸ばすことを考える．

プレロード　まずプレロードを考える．プレロードは，データアイテムを，それらがアプリケーションから read される前にストレージキャッシュにロードしストレージキャッシュ中に保持する機能である．データアイテムがストレージキャッシュにロードされると，アプリケーションはこれらのデータアイテムをストレージキャッシュから read するため，ディスク筐体に対する read 入出力をゼロにすることが可能となる．通常，データアイテムをストレージキャッシュに読み込むために必要な時間は，アプリケーションの実行時間と比較して短い．このためディスク筐体に対する入出力発行間隔を Break Even Time 以上とすることが可能となる．

Write 遅延　次に，Write 遅延について考える．Write 遅延は，データアイテムの更新された部分をストレージキャッシュに保持し，それらのディスク筐体への書き出しをまとめて行うことにより，ディスク筐体に対する write 入出力の間隔を伸ばす機能である．通常，データをストレージキャッシュからディスク

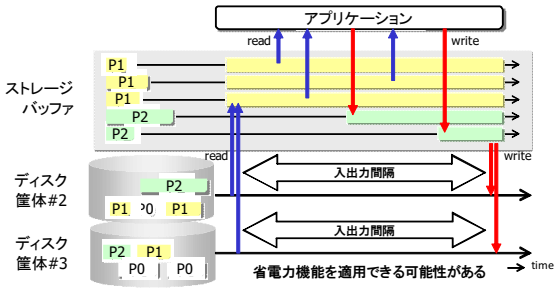

図 4　プレロード及び Write 遅延の効果
Fig. 4　An Effect of Pre-Load and Write Delay.

筐体に書き出す時間はアプリケーションが実行される時間より短いため，write 遅延によりディスク筐体に対する write 入出力の発行間隔を Break Even Time 以上にできる可能性が高まる．また，ストレージキャッシュはバッテリーバックアップされており，write 入出力を遅延しても DBMS の ACID 特性は保証される．

図 4 にプレロード及び write 遅延の効果を示す．図から分かるように，アプリケーションが P1 型データアイテムに入出力を行う前にそれらをバッファにロードすることにより，ディスク筐体#2，#3 の read 間隔を伸ばせる可能性を高めることができる．また，P2 型のデータアイテムに対する更新をバッファに維持し，まとめてディスク筐体に書き出すことにより，ディスク筐体に対する write 間隔を伸ばせる可能性を高めることが可能となる．

3. アプリケーション協調型ストレージ省電力システムの実装

3.1 実行時省電力ストレージ管理機構の設計

アプリケーション協調型ストレージ省電力システムは，アプリケーションレベル及びストレージデバイスレベルの入出力を監視・パターン化し，それに基づき入出力パターンに適した省電力手法を選択する．アプリケーションレベルの入出力の監視やデータ配置制御などの省電力手法の選択・変更に伴うアプリケーションへの影響を抑えるためには，ストレージ省電力システムは以下の要件を満たす必要がある．

軽量入出力統計取得　提案手法は，データに対するアクセス間隔が Break Even Time 以上であるか否かを判断してデータを配置する階層を決定する．このためには，入出力発行間隔に関する情報，特にデータ毎の Break Even Time 以上の入出力発行間隔の有無を軽量に取得できなければならない．

透過的ファイルアクセス　提案手法は，データ階層間，及びストレージ階層間でデータを移動することにより入出力間隔の長いディスク筐体を生成する．データの移動によりデータの実体へのアクセスパスは変化するが，アプリケーションに対してはこれを隠蔽しなければならない．

3.2 ストレージ省電力システムの実装

前節の要件に基づき，省電力管理システムの実装を図 5 に示すように規定した．図の構成要素の色は，フレームワークの各機能の色に対応している．

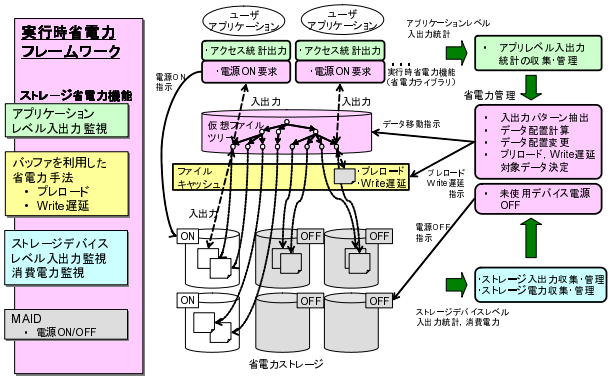

図 5　ストレージ省電力システムの実装
Fig. 5　Implementation of storage power control system.

省電力管理システムは，アプリケーションレベル入出力監視，バッファを利用した省電力手法，ストレージデバイスレベル入出力と消費電力監視，及び MAID 機能を有する．以下，これらの各機能及び実行時ストレージ省電力フレームワークの実現方式について説明する．

3.2.1 アプリケーションレベル入出力監視

アプリケーションレベル入出力監視は，アプリケーションの入出力を監視し，それを共有メモリに出力するアクセス統計出力機能とアクセス統計出力機能が共有メモリに出力したアプリケーションレベル入出力の統計情報を定期的に収集し，時系列情報として管理するアプリケーションレベル入出力統計収集・管理機能を有する．

アクセス統計出力機能はライブラリとして実装され，アプリケーションにリンクされる．アプリケーションレベル入出力統計収集・管理機能はアプリケーションとは独立した管理プロセスとして実装される．

3.2.2 バッファを利用した省電力手法

バッファを利用した省電力手法は，プレロード対象となったデータ，及び write 遅延対象となったデータの更新部分を保持するバッファ，データのプレロード機能及び write 遅延機能を有する．

本システムでは，前述のバッファとして，アプリケーションが DBMS の場合は DBMS の，そうではない場合はファイルシステムのバッファを用いる．プレロード機能は管理プロセスとして実装される．また，write 遅延機能は DBMS バッファあるいはファイルシステムバッファ内に実装される．

3.2.3 ストレージデバイスレベル入出力及び消費電力監視

ストレージデバイスレベル入出力及び消費電力監視は，ストレージのディスク筐体に対する入出力及び消費電力を収集し，時系列情報として管理する機能を有する．これらの機能は，管理プロセス内に実装される．

3.2.4 実行時ストレージ省電力フレームワーク

実行時ストレージ省電力フレームワークは，仮想ファイルツリー，省電力管理機能として入出力パターン抽出，データ配置計算，データ配置変更，プレロード・write 遅延対象データ決定機能を有する．さらにストレージデバイスレベル入出力及び消費電力監視が収集・管理している情報を参照し，入出力が行

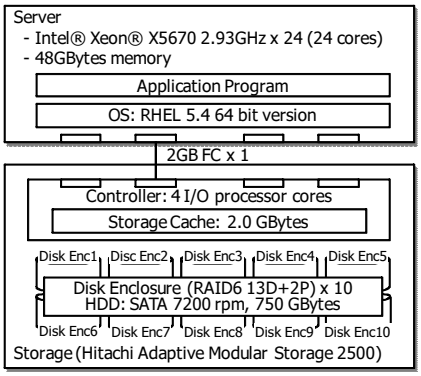

図 6　テストベッド
Fig. 6　Test bed.

われていないディスク筐体の電源を OFF にする未使用デバイス電源 OFF 機能を有する．

仮想ファイルツリーとは，ファイルの実体へのアクセスパスを抽象化し，ファイルがディスク筐体間を移動してもファイルへのアクセスパスが変わらないようにするための機構である．本実装では，シンボリックリンクを用いて仮想ファイルツリーを実現する．

省電力管理機能は，前述の管理プロセス内に実装される．

4. 実行時ストレージ省電力管理機構の評価

本節では，実行時ストレージ省電力管理機構を動作させたストレージ上で商用 DBMS を用いて TPC-C 及び TPC-H を独立に実行し，それぞれのストレージ消費電力，TPC-C のトランザクションスループット，及び TPC-H のクエリの応答時間を計測する．そして，実行時ストレージ省電力管理機構が，実システム上で有効に稼働することを示す．

4.1 評価環境

我々がストレージ省電力システム及びアプリケーションを稼動させるテストベッドを図 6 に示す．サーバの CPU は Intel Xeon X5670 2.93GHz × 24(cores)，主記憶は 48GB である．ストレージは日立製作所製の Hitachi Adaptive Modular Storage 2500 (AMS2500) である，本ストレージは，2GB のキャッシュを持つ RAID コントローラ，及び Seagate 社製の 750GB SATA (7,200rpm) を 15 台搭載したディスク筐体を 10 台有している．1 ディスク筐体内の HDD 15 台を用いて RAID6 (13D+2P) を構成している．RAID 構成前のストレージの全容量は 112.5TB である．また，サーバとストレージ間は 2GB の Fibre Channel ケーブル 1 本で接続されている．

我々は，ストレージの RAID コントローラ及びディスク筐体に HIOKI 社製の電力計 (Remote Measurement and Monitoring System 2300 シリーズ) を取り付け，ストレージの消費電力を計測した．

4.2 評価方法

4.2.1 計測項目

我々はストレージの実際の消費電力，及びアプリケーションの性能を計測する．消費電力には，データアイテムの移動，プレロード，及び write 遅延に要する電力，及びディスク筐体の電源の ON/OFF に要する電力も含む．アプリケーション性能は，OLTP の場合はトランザクションスループット及びトランザクション応答時間の分布を，DSS の場合はクエリの応答時間をそれぞれ計測する．応答時間は，ディスク筐体の起動に伴う遅延を含む．

表 1　TPC-C 及び TPC-H の設定

Application	Data Size	Workload	Cache Size
TPC-C	1TB	# of warehouse: 3000 # of threads: 200 Duration: 0.5 hr Put log to 1 Storage Device Put DB to 10 Storage Devices (hash distribution)	25GB (DBMS) 2GB (Storage)
TPC-H	1.2TB	SF=300 Run Q1 to 22 sequentially Duration: 6 hr Put log and work files to 1 Storage Device Put DB to 10 Storage Devices (key range distriution)	40 GB (DBMS) 2 GB (Storage)

4.2.2 アプリケーション設定

ストレージ上で動作する TPC-C 及び TPC-H の入出力挙動特性及び性能の計測に用いたアプリケーションの構成を表 1 に示す．TPC-C のデータベースサイズは約 1TB (Warehouse 数 3000)，DBMS のバッファサイズは 25GB，ストレージキャッシュサイズは 2GB とした．スレッド数は，Think Time を TPC-C の仕様に規定された通りとした場合にトランザクション応答時間を満たし，かつ最もスループットが高かった値である 400 とした．ログ及び作業表を図 6 中のディスク筐体#1 に，表と索引をディスク筐体#2 から 11 にハッシュ分割機能を用いて分散配置した．上記環境において，TPC-C を 1 時間実行した．

TPC-H のデータベースサイズは約 1.2TB (Scale Factor 300)，DBMS のバッファサイズは 40GB とした．ログ及び作業表を図 6 中のディスク筐体#1 に，表と索引をディスク筐体#2 から 11 にハッシュ分割機能を用いて分散配置した．上記環境において，TPC-H のクエリ 1 から 22 までを順次実行した．

提案手法においてはバッファを利用した省電力機構を DB バッファ層に組み込んだ．DB バッファを Hot データ用と Cold データ用に分割し．Cold データ用のバッファにプレロード対象データをプレロードするとともに Cold データ用バッファのチェックポイント間隔を最大値 (1 時間) とすることにより，プレロード及び write 遅延を実現した．

4.2.3 データインテンシブアプリケーションの論理入出力パターン

我々はまず，我々が導入した論理入出力パターンによりデータインテンシブアプリケーションの入出力挙動を把握することが可能かどうかを確認するために，TPC-C 及び TPC-H の論理入出力パターンを計測した．計測結果を図 7 に示す．図に示すよ

うに，TPC-C では 76.1%のデータアイテムが P3，0.9%が P2，8.9%が P1，14.1%が P0 型であった．TPC-H では，TPC-C と異なり 79.7%のデータアイテムは P0，20.1%が P1，0.2%が P2 型であり，P3 型のデータアイテムは見られなかった．図から，TPC-C と TPC-H 間で論理入出力パターンは大きく異なってることが分かる．これは，アプリケーション毎に適切な省電力戦略を取る必要があることを示している．

また，入出力パターンの安定性についても調査し，データアイテムの入出力パターンがアプリケーションの実行期間中に大きく変化しないことを確認した．

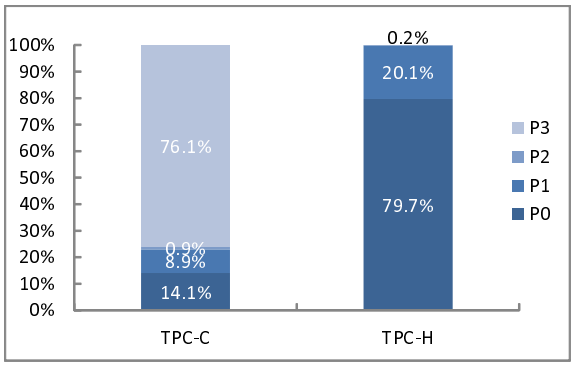

図 7　データインテンシブアプリケーションの論理入出力パターン
Fig. 7　I/O Pattern of Data Intensive Applications.

4.2.4　TPC-C の評価結果

省電力ストレージ管理機構を利用しない場合 (省電力制御なし)，MAID 機能のみを使用した場合 (MAID)，データ配置と MAID のみを使用し，バッファを利用した省電力手法を使用しない場合 (データ配置+MAID)，及び省電力ストレージ管理機構を利用した場合 (提案手法) のストレージ消費電力の平均値及びトランザクションスループットの計測結果をそれぞれ図 8 及び 9 にそれぞれ示す．

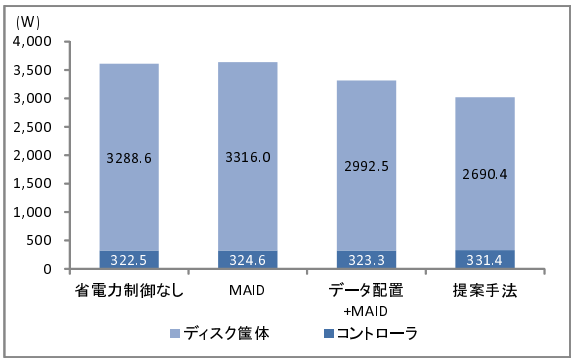

図 8　TPC-C が稼働するストレージの平均消費電力
Fig. 8　Storage Power Consumption of TPC-C.

図 8 から分かるように，提案手法は電力削減量が最も大きく，ディスク筐体の平均消費電力を 3288.6 W から 2690.5W に約 18.2%削減できた．これは，入出力が常時行われるデータを 8 台のディスク筐体に集めたこと，及び DB バッファを用いたプレロード及び write 遅延の効果である．また，図 9 から分かるように，MAID およびデータ配置+MAID はトランザクションスループットが大きく低下しているが，提案手法はトランザクションスループットは省電力制御なしとほぼ同等であった．

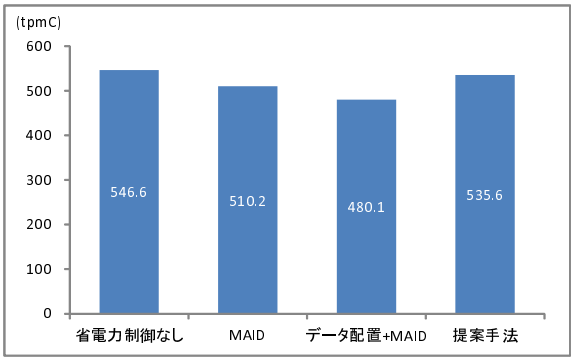

図 9　TPC-C のトランザクションスループット
Fig. 9　Transaction Throughput of TPC-C.

4.2.5　TPC-H の評価結果

省電力ストレージ管理機構を利用しない場合 (省電力制御なし)，MAID 機能のみを使用した場合 (MAID)，データ配置と MAID のみを使用し，バッファを利用した省電力手法を使用しない場合 (データ配置+MAID)，及び省電力ストレージ管理機構を利用した場合 (提案手法) のストレージ消費電力の平均値及びクエリ Q2, Q7 の応答時間の計測結果をそれぞれ図 10 及び 11 にそれぞれ示す．

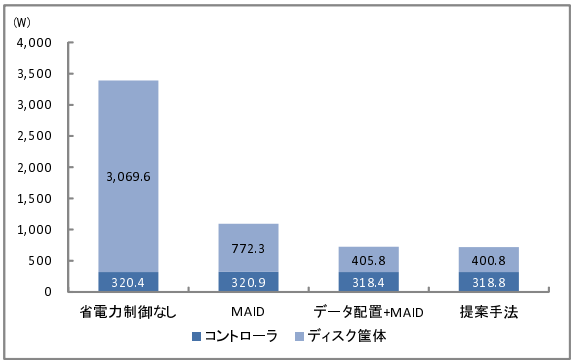

図 10　TPC-H が稼働するストレージの平均消費電力
Fig. 10　Storage Power Consumption of TPC-H.

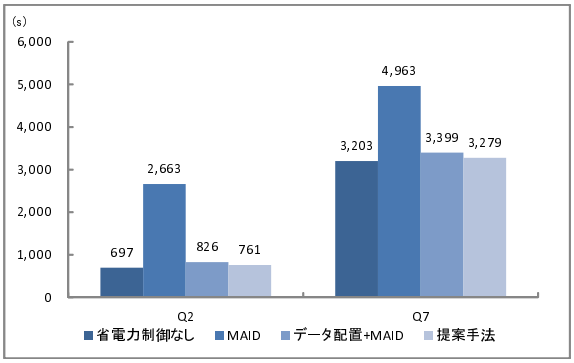

図 11　TPC-H のクエリ応答時間 (Q2, Q7)
Fig. 11　Query Resopnse Time of TPC-H Q2 and Q7.

図 10 から分かるように，全ての手法において消費電力を 50%以上削減できているが，提案手法を用いた場合の消費電力の削減量が最も大きく，提案手法はディスク筐体の平均消費電力を 3229.7 W から 698.1W に約 79%削減できた．これは，入出力が常時行われるデータを 2 台のディスク筐体に集めたことによる効果である．また，図 11 から分かるように，クエリ応答時間は Q2 が 9.7%，Q7 が約 2.4%倍増加したが，他の手法と比較して応答時間の増加を短く抑えられていることが分かる．Q7 の応答時間が増加したのは，入出力が 2 台のディスク筐体に集約されたことによる入出力応答時間の増加のためである．

これらの結果は，TPC-C 及び TPC-H の結果は，提案方式が大規模なストレージ上でも動作することを示している．

5. ま と め

本論文では，アプリケーションの入出力挙動を用いることによりストレージの省電力の機会を増加させるストレージ省電力フレームワークを提案した．また，ファイルサーバ，OLTP，DSS 等の実際のデータセンタで稼働するデータインテンシブアプリケーションの入出力パターンを調査し，それらが 4 つの入出力パターンに分類されることを確認した．さらに，本章で提案した実行時ストレージ管理機構の実装を示すとともに商用の DBMS を用いて TPC-C 及び TPC-H を動作させ，ストレージの消費電力と TPC-C のトランザクションスループット及び TPC-H のクエリ応答時間を計測した．この結果，TPC-C が稼働するストレージの消費電力を，TPC-C のスループットをほとんど低減することなく約 18%削減できること，及び数%程度の TPC-H のクエリ応答時間の遅延で TPC-H が稼働するストレージの消費電力を約 79%削減できることを確認した．

今後は大規模システムを対象とした評価を行い，大規模ストレージ環境においても提案手法が高いストレージ省電力効果を得られることを確認する．

文　献

[1] Hitachi Adaptable Modular Storage 2500. *HITACHI, Ltd.*, 2011.

[2] Dennis Colarelli and Dirk Grunwald. Massive Arrays of Idle Disks For Storage Archives. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, 2002.

[3] Austin Donnelly and Antony Rowstron. Write Off-Loading : Practical Power Management for Enterprise Storage. In *6th USENIX Conference on File and Storage Technologies*, pp. 253–267, 2008.

[4] Frank Gens. IDC Predictions 2011: Welcome to the New Mainstream. *IDC White Paper #225878*.

[5] Jorge Guerra, Himabindu Pucha, Joseph Glider, Wendy Belluomini, and Raju Rangaswami. Cost Effective Storage using Extent Based Dynamic Tiering Multi-Tiering : Design Choices. In *9th USENIX Conference on File and Storage Technologies*, pp. 1–14, 2011.

[6] T. Heath, E. Pinheiro, J. Hom, U. Kremer, and R. Bianchini. Application transformations for energy and performance-aware device management. In *Proceedings.International Conference on Parallel Architectures and Compilation Techniques*, pp. 121–130. IEEE Comput. Soc, 2002.

[7] Dong Li and Jun Wang. EERAID: Energy Efficient Redundant and Inexpensive Disk Array. In *11th ACM SIGOPS European Workshop*, 2004.

[8] N. Nishikawa, M. Nakano, and M. Kitsuregawa. Low power mnagement of oltp applications considering disk drive power saving function. In *21th International Conference of Database and Expert Systems Applications*, pp. 241–250. Springer, 2010.

[9] N. Nishikawa, M. Nakano, and M. Kitsuregawa. Potentiality of power management on database systems with power saving function of disk drives. In *The 22nd Australian Database Conference*, 2011.

[10] N. Nishikawa, M. Nakano, and M. Kitsuregawa. Energy efficient storage management cooperated with large data intensive applications. In *28th IEEE International Conference on Data Enginerring*, 2012.

[11] Ekow Otoo, Doron Rotem, and Shih-chiang Tsao. Dynamic Data Reorganization for Energy Savings. In *Proceedings of the 22nd international conference on Scientific and statistical database management*, pp. 322–341, 2010.

[12] Athanasios E Papathanasiou and Michael L Scott. Energy Efficient Prefetching and Caching. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, 2004.

[13] Eduardo Pinheiro and Ricardo Bianchini. Energy conservation techniques for disk array-based servers. In *Proceedings of the 18th annual international conference on Supercomputing - ICS '04*, pp. 68–78, New York, New York, USA, 2004. ACM Press.

[14] Akshat Verma, Ricardo Koller, Luis Useche, and Raju Rangaswami. SRCMap : Energy Proportional Storage using Dynamic Consolidation. In *Proceedings of the 8th USENIX conference on File and storage technologies*, 2010.

[15] Charles Weddle, Mathew Oldham, Jin Qian, An-I Andy Wang, Peter Reiher, and Geoff Kuenning. PARAID: A Gear-Shifting Power-Aware RAID. In *5th USENIX Conference on File and Storage Technologies*, Vol. 3, pp. 245–260, October 2007.

[16] Xiaoyu Yao and Jun Wang. RIMAC : A Novel Redundancy-based Hierarchical Cache Architecture for Energy Efficient , High Performance Storage System. In *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, No. i, pp. 249–262, 2006.

[17] Natalya Yezhkova and Richard L. Villars. Worldwide Enterprise Storage Systems 2010-2014 Forecast Update. *IDC White Paper #226223*.

[18] Alan G Yoder. Green Storage Technologies, CAPEX and OPEX. In *Storage Networking World Fall 2010 Conference*, 2010.