目 次

図 目 次

表 目 次

第1章　序論

1.1　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 のレポートによれば [29], 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．爆発的な増加が予想されるこれらのデジタルデータは，大規模なストレージに格納されている．このため，ストレージの容量も今後急増することが予想されている．例えば，文献 [105] では，2014 年のストレージ出荷容量は，2009 年の 7 倍に増加すると報告されている．

今日，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い [106]．特に，IT 機器の消費電力に占めるストレージの消費電力は，デジタルデータの増加とも相まって急増している [101]．例えば，文献 [80] に示すように，大規模なオンライントランザクション処理システム (OLTP) におけるストレージの消費電力は，IT 機器全体の消費電力の 70% 以上を占めるとの報告もある．すなわち，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．

当該問題を解決すべく，従来より，大容量ハードディスク (HDD) や 2.5 インチ HDD, Solid State Disk (SSD) など容量当りの電力効率が高いストレージデバイスの利用や，データ圧縮や重複排除などのデータの格納効率の向上などによるストレージの省電力が行われている [111]．これに対し，本論文は，近年のストレージ [64, 9] が，アクセスが行われていない HDD の回転を停止させることによりストレージの省電力化を図る MAID (Massive Arrays of Idle Disks) 技術 [20] を搭載し動的な省電力を可能としつつあることに着目する．MAID を活用することにより，従来では困難であったアプリケーション実行中のストレージ省電力を目指す．本論文は，アプリケーション実行中にその性能を低下することなくストレージ省電力を実現するためのフレームワークの提案とその実証を目的とする．すなわち，ストレージデバイスレベルに加え，アプリケーションの論理レベルの入出力挙動を能動的に利用することでストレージ省電力を可能とする，新たな実行時ストレージ省電力技法を提案する．

これまで，MAID 機能を利用したストレージ省電力手法がいくつか報告されている．これらの手法には，ストレージデバイスの物理ブロックに対する入出力頻度を監視し，入出力の頻度が高い物理ブロックの先読みや書き出を一括して行うことによりストレージデバイスに対する入出力間隔を延伸する手法 [72, 55, 104, 38] や，物理ブロックの入出力頻度を監視し，入出力頻度に対してストレージデバイスの入出力間隔が伸びる方向でデバイス間で物理ブロックを再配置することによりストレージ省電力を目指す手法 [19, 75, 97, 23, 92, 69, 34]

等がある．

次に，データセンタで稼働するアプリケーションに着目する．データセンタでは，常時多くのアプリケーションが稼働している．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) を例に挙げるまでもなく，これら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．例えば，OLTP の代表的ベンチマークである TPC-C[7] は，マスタテーブルに対するランダム入出力を行う．意思決定支援 (DSS) の代表的ベンチマークである TPC-H[10] は，巨大なトランザクションテーブルに対する逐次的な一括読み取りを行う．また，多くのアプリケーションでは，アプリケーションレベルの入出力の傾向は頻繁には変わらないことが報告されている [25].

従来のストレージ省電力手法は，ストレージデバイスレベルの入出力挙動のみに着目しており，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に活用することができない．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アプリケーションの性能劣化を引き起こすなどの可能性がある．

1.2 研究の目的と概要

ストレージを利用するアプリケーションは OLTP や DSS，大規模なファイルサーバなど多岐に渡る．また，省電力の対象となるストレージデバイスには HDD やストレージがあり，その省電力手法にはデバイスの ON/OFF による消費電力の削減の他に，RAID 構成の選択による消費電力の削減などがある．本論文では，アプリケーションやストレージデバイス，省電力手法に依存しない高い汎用的性を持つ実行時ストレージ省電力フレームワークを提案する．

本論文で提案す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の入出力挙動とストレージデバイスレベルの入出力挙動を統計的に解析することにより，ストレージデバイスに対するアプリケーションのデータ毎の入出力の傾向を把握し，これを利用してストレージデバイスに対する入出力の間隔を伸ばすことにより，アプリケーション実行中のストレージの電力消費を削減する省電力機構を提案する．すなわち，アプリケーションレベルの入出力トレースの統計的解析結果より省電力の機会を得る可能性能ある入出力パターンを抽出する．そして，抽出された入出力パターンに対してストレージデバイスレベルの入出力間隔を最も伸ばすことが可能な省電力手法を選択し実行する．入出力パターンに適した省電力手法を動的に選択するため，アプリケーション実行時であっても最も効果のある省電力手法の適応が可能となる．

データセンタで稼動するストレージコンポーネントの中で最も消費電力が高いのは HDD である．そこで，まず代表的なデータインテンシブアプリケーションである OLTP 実行中の HDD 省電力手法を検討する．一般に OLTP ではデータアクセスが頻繁なため，OLTP 実行中の HDD の省電力は困難とされている．OLTP のアプリケーションレベルの入出力挙動特性を詳細に解析し，OLTP が入出力を行うデータには常時高頻度で入出力が行われるデータと，数百秒から数千秒は入出力が行われないデータがあることを明らかにする．さらに，OLTP が入出力を行うデータとデータが配置されている HDD との対応関係を利用

して．常時入出力が行われるデータと長時間入出力が行われないデータを異なる HDD に配置すると共に，DBMS がログ先行書き出しプロトコルに従い DB のデータを HDD に書き出すことを利用して長時間入出力が行われないデータの HDD への書き出しを遅延させることにより，HDD に対する入出力間隔を伸ばす新たなアプリケーション実行時省電力手法を提案する．さらに，実機を用いた実験により，提案手法が従来手法と比較して OLTP 実行中の処理性能を劣化させることなく HDD の消費電力を大きく削減できることを示す．

次に，多様なアプリケーションが稼働するストレージの場合について検討する．既に，OLTP の入出力挙動特性を用いることにより HDD の消費電力を大きく削減できることを述べたが，個々のアプリケーション毎，あるいは個々のアプリケーションのデータ毎に省電力手法を仕立てたのでは，アプリケーションやデータの数だけ省電力手法を構築する必要があり効率が悪い．このため，特定のアプリケーションによらずアプリケーションデータに対する入出力挙動特性を識別するための入出力パターンを導入すると共に，入出力パターンに適したストレージ省電力手法を動的に選択しストレージの消費電力を削減するフレームワークを提案する．本フレームワークの特長は，次の通りである．

1. 入出力発行間隔の長さや read/write 数の比率などに基づき，アプリケーションレベルの入出力挙動をストレージの省電力手法と対応した入出力パターンに分類する．個々のアプリケーションの入出力挙動ではなく，入出力パターンを基に実行時ストレージ省電力手法を実行することにより，アプリケーション毎の入出力挙動によらずに省電力を可能とする高い汎用性を提供すると同時に，従来のストレージデバイスレベル入出力挙動を用いた手法と比較して高い省電力効果が期待できる．

2. アプリケーションレベルの入出力挙動を反映した入出力パターンに基づき，常時入出力が行われるアプリケーションデータを識別し省電力効果の高いデータ配置をアプリケーションレベルで行う．さらに，実行時のストレージ省電力を効果的にするため，常時入出力が行われるデータを配置していないストレージデバイスに配置されたデータについて，read 間隔が短いデータをキャッシュにプレロードする，あるいは write 間隔が短いデータのストレージデバイスへの write を遅延することにより，省電力化の可能性を高める．

データセンタで稼動する典型的なアプリケーションであるファイルサーバ，OLTP，及び DSS を用いて提案手法を定量的に評価する．そして，提案手法が従来の手法と比較してアプリケーション実行中のストレージの消費電力を大きく削減できることを示す．さらに省電力ストレージ管理機構を開発し，その実装を示すとともに，実アプリケーションを用いた評価を行う．そして，提案手法が実環境で有効に動作することを示す．

近年，データセンタで用いられるストレージは RAID を構成するドライブの数や種類 (HDD, Solid State Disks (SSD))，RAID レベルの異なる様々な RAID 構成を取ることが可能である．従来，RAID は容量効率や信頼性，性能に重点を置いて構成されることが多かったが，本研究では，省電力の観点からの RAID の構成を議論する．つまり，広くデータセンタで用いられている十数台のドライブから構成される RAID 5 や RAID 6 は，電源投入時のエネルギーや起動時間などのオーバヘッドが大きく省電力の観点からは必ずしも望ましい RAID 構成ではないことを明らかにすると共に，アプリケーションの入出力挙動に適した RAID 構成があること，例えば read のみを行うアプリケーションでは RAID 4 や RAID

0+1 のような冗長データのみを格納するドライブとデータを格納するドライブとを独立に設ける RAID 構成が省電力効果が高いことを，実アプリケーションの入出力トレースを用いたシミュレーションにより示す．

また，データセンタでは，増大し続けるデータに対しストレージを階層化し，データに求められる入出力性能などの要件を満たしつつストレージのハードウェアコストの低減を狙う階層的データ管理が着目されている [74]．階層的データ管理は，大規模なストレージほど多く導入される傾向にあり，大規模顧客ではその半数以上が何らかの階層的なデータ管理手法を導入している．しかし，これまで階層的なデータ管理を運用しているストレージを対象とした省電力手法は提案されていない．そこで，これまで提案した手法に階層的データ管理手法を取り入れることにより，データの利用者が求める性能要件を満たしつつ，大規模なストレージの省電力とハードウェアコストの低減を支援する，新たな省電力手法を提案する．すなわち，データの入出力要件に基づきデータの管理階層を決定し，データの管理階層に求められる性能などの要件に基づきストレージ階層を構築し，データ管理階層とストレージ階層を対応付ける．実験環境として用いているシステムであるデータ統合・解析システム (DIAS)[4] を用いた評価結果を示し，提案手法が，実運用されている大規模なストレージにおいても高い省電力効果を発揮することを示す．

1.3 本論文の構成

本論文は以下の章からなる．まず，序論として本章があり，続いて第 2 にて本論文にて提案する実行時ストレージ省電力フレームワークについて説明する．第 3 章では，従来のストレージ省電力の研究について，HDD の省電力，RAID 及びストレージの省電力，アプリケーションによる IT 機器の省電力，及びデータセンタの省電力の観点から簡単に概観するとともに，本研究との関係について述べる．第 4 章では，省電力の対象とする HDD 及びストレージの消費電力特性を明らかにし，第 5 章では，OLTP 実行中の HDD の省電力手法の研究について述べる．第 6 章では，データセンタで稼動する大規模なデータインテンシブアプリケーション実行中のストレージ省電力手法の研究，及び省電力ストレージ管理機構の実装と評価結果について述べる．第 7 章では，ストレージの省電力の観点からの RAID 構成について議論する．第 8 章では，階層的データ管理手法を用いたストレージの省電力手法に関する研究，及び階層的データ管理を支援するストレージ管理機構の評価結果について述べる．最後に，第 9 章にて，本研究で得られた成果と今後に残された課題について述べる．

第2章　実行時ストレージ省電力フレームワーク

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 2.1 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．

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．

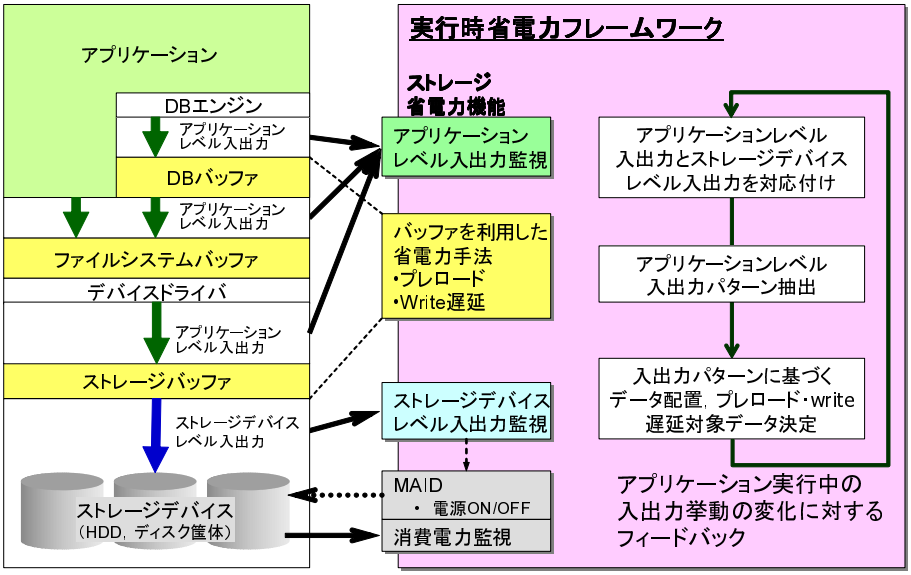

図 2.1: 実行時ストレージ省電力フレームワーク

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　実行時ストレージ省電力フレームワークの流れ

本節では，本研究において提案する実行時ストレージ省電力フレームワークを，従来の省電力手法と対比しつつ説明する．

2.2.1　従来のストレージ省電力におけるストレージデバイスレベル省電力

図 2.2 は，従来のストレージ省電力手法におけるストレージデバイスレベル省電力の流れを示している．図から分かるように，従来手法ではアプリケーションレベルの入出力では自明であるファイルや DB の表・索引の入出力挙動特性をストレージ省電力に用いていない．

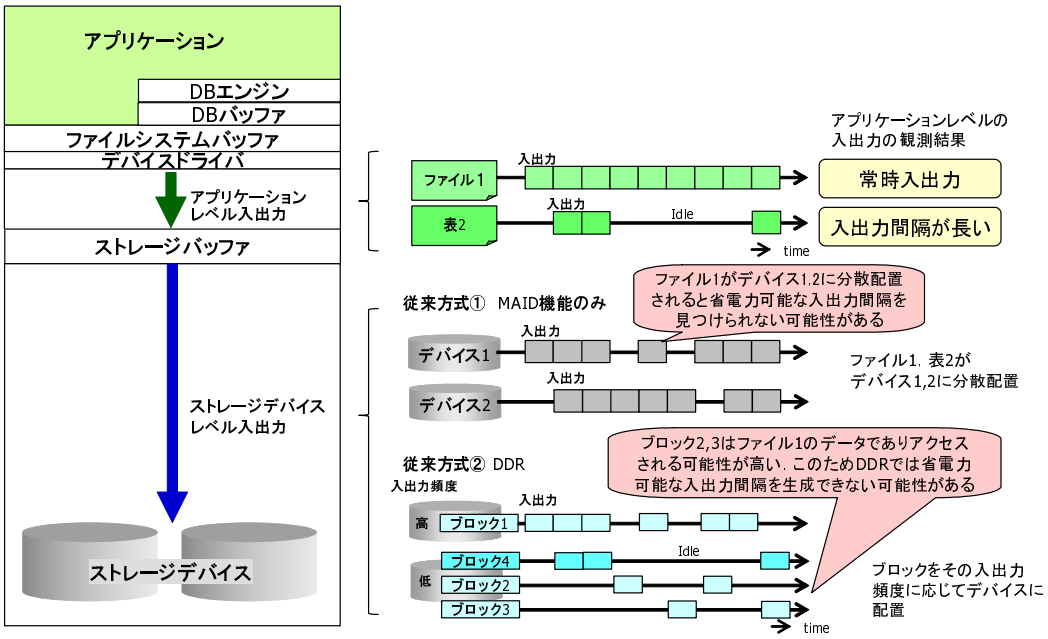

図 2.2: 従来のストレージ省電力におけるストレージデバイスレベル省電力の流れ

図 2.2 では，アプリケーションが使用するデータとしてファイル 1 と表 2 がある．アプリケーションの入出力挙動を観測することにより，ファイル 1 は常時入出力が行われ，表 2 は入出力間隔が長いことを知ることができる．

以下，従来方式の問題点について見てゆく．従来方式　は単純な MAID 機能のみを用いた場合を示している．図に示すようにファイル 1 がデバイス 1 と 2 に分散配置された場合，単純な MAID 機能のみを用いた場合はデバイスに対して省電力可能な入出力間隔を見つけられない可能性がある．

従来方式　はブロックの入出力回数に基づきブロックの配置を決める DDR について示している．この例では，ファイル 1 を構成するブロックがブロック 1，2，3，表 2 を構成するブロックがブロック 4 である．ブロックの入出力回数に基づきデバイスにブロックを配置した結果，ブロック 1 が入出力頻度が高のデバイスに，ブロック 2，3，4 が入出力頻度低のデバイスにそれぞれ配置されている．しかし，ブロック 2，3 はファイル 1 のブロッ

クでありアクセスされる可能性が高い．このため DDR では省電力可能な入出力を生成できない可能性がある．

2.2.2　実行時ストレージ省電力におけるアプリケーションレベル省電力

図 2.3 は，本節で提案する実行時ストレージバッファ層にバッファを利用した省電力手法を組み込んだ場合のストレージ省電力フレームワークにおけるアプリケーションレベル省電力の流れを示している．

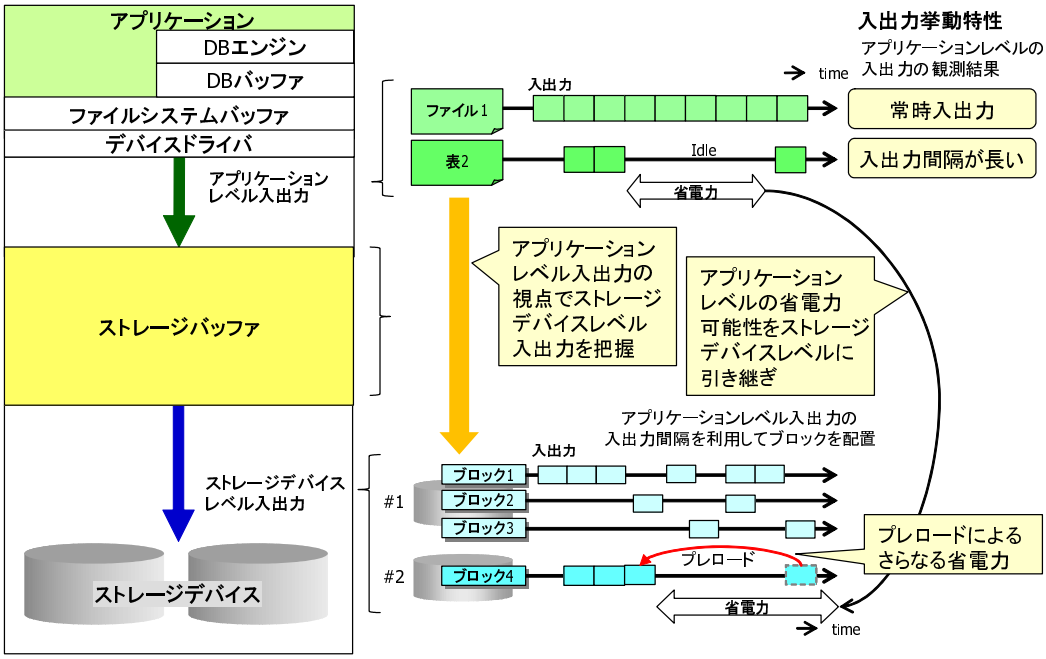

図 2.3: 実行時ストレージ省電力におけるアプリケーションレベル省電力の流れ

実行時ストレージ省電力フレームワークでは，バッファ層に位置する実行時ストレージ省電力機能がまずアプリケーションレベル入出力の視点でストレージデバイスレベル入出力を把握する．そしてアプリケーションレベルの省電力の可能性をストレージデバイスレベルに引き継ぐ．具体的には，アプリケーションレベル入出力の入出力間隔を利用してストレージデバイスに対するブロックの配置を決定する．

図 2.3 の例では，アプリケーションレベル入出力は図 2.2 に示した場合と同様，ファイル 1 と表 2 がありファイル 1 に対して常時入出力が行われ，表 2 に対する入出力間隔は長く省電力可能な入出力間隔が存在する．アプリケーションレベル入出力の入出力間隔を利用してストレージデバイスに対するブロックを配置することにより，ファイル 1 のブロック 1，2，3 がデバイス#1 に，表 2 のブロック 4 がデバイス#2 にそれぞれ配置される．この結果，表 2 の持つ省電力の可能性をデバイス#2 に引き継ぐことができ，デバイス#2 の省電力の可能性を高めることができる．さらに，入出力間隔が長いデータのブロックをバッファ層にプレロードすることにより，デバイスの入出力間隔をさらに伸ばすことができ，デバイスの省電力の可能性を高めることができる．

第3章　関連研究

本章では，関連研究をまとめ，本研究との関連について述べる．まず単一の HDD，及び複数台の HDD を対象とした省電力手法について述べ，本研究の関係をまとめる．次に，ストレージの省電力手法として RAID を構成する HDD 群，及び大規模なキャッシュと複数の RAID を持つストレージの省電力手法について延べ，それらと本研究の関係をまとめる．その後，アプリケーションの情報を活用したストレージの省電力に関する手法，及び本研究との関係をまとめ，最後にデータセンタにおける省電力手法及び本研究との関係を述べる．

3.1　HDD の省電力に関する研究

HDD の省電力は，従来ノート PC などのモバイル機器の省電力を対象に発展してきた．その手法は，大きく，HDD の省電力状態を制御する方法，キャッシュ等を用いて入出力発行間隔を変更する方法，及び入出力発行が伸びるようデータ配置を変更する方法，に分けることができる．

3.1.1　HDD の省電力状態の制御

HDD の省電力状態の制御には，省電力状態へ移行する契機を制御する手法，複数の回転数を取ることができる HDD を前提とした，HDD 回転数の動的な変更手法に分けられる．

HDD の省電力状態の制御

従来，モバイル機器等を中心に一定時間 HDD の Idle 状態が続くと HDD を Standby 状態等の省電力状態に移行する省電力手法が採用されている [56, 44, 58]．これらの手法は伝統的に用いられていることから Traditional Power Management(TPM) と呼ばれている．ところが，HDD への入出力は，HDD を利用するユーザやアプリケーションにより大きく異なるのが普通である．このため TPM では，例えば入出力発行間隔が上記一定時間よりは長いが Break Even Time より短い場合に消費電力を削減できない，あるいは今後長時間入出力が行われない場合でも一定時間入出力を待つため無駄に電力を消費する，等の問題がある．

HDD の省電力状態の制御手法とは，HDD を省電力状態に移行するまでの時間を動的に変更することによりディスクのアイドル時間を短くすることにより，上記の問題を解決しようとする手法である [24, 51, 39]．文献 [24] は，HDD の利用パターンがユーザや時間の経過とともに変化するため HDD を Standby 状態に移行するまでの待ち時間 (閾値) を常に

固定値とすることは適切ではないことを指摘している．そして，過去の HDD へのアクセスパターンに基づき HDD を省電力状態に移行するまでの時間を動的に変化することにより HDD の省電力を図る手法について述べている．また，この問題を機械学習を用いて解決しようとするアプローチも見られる．文献 [51] は，HDD を省電力状態に移行する契機の決定を Rent-to-Buy 問題とみなし，それを解くためのアルゴリズムを提案している．文献 [39] は，ディスクに対する入出力のトレースを用いた機械学習により入出力の発行を予測し，予測結果に基づきディスクを省電力状態に移行するか否かを判断する手法を提案している．

HDD の内部で得られる情報のみでは入出力挙動の予測が十分ではないことから，OS 層との連携やアプリケーションとの連携を試みる研究も報告されている．文献 [32, 31] は，OS の入出力発行契機を予測するためにプログラムカウンタを用いる手法を提案している．本手法は，プログラムカウンタの特定の命令列と Idle 期間との関係を学習し，将来の Idle 期間の発生を予測する．そして，Idle 時間が十分長いと予測される場合に HDD を省電力状態に移行する．また，文献 [99] は，Cooperative 入出力 (Coop-I/O) と呼ばれる，拡張されたファイル操作インタフェースを提供する．Coop-I/O は，HDD が省電力状態の場合に，ファイルに対する入出力を遅延しても良いことを OS に伝える．OS はこの情報を利用して，対象の HDD が (他のオペレーションにより) Active/Idle 状態となる，あるいはインタフェース経由で指定された時間が経過するまで入出力を遅延する．これによりこれによりアプリケーションへの性能の影響を抑えつつ HDD の省電力状態の持続時間を増やすことを狙っている．

HDD 回転数の制御

HDD の消費電力は，主にプラッタを回転させるスピンドルモーターが消費している．モーターの消費電力は回転数の二乗に比例するため，スピンドルモーターの回転数を下げることができれば HDD の消費電力を大きく減らすことが可能となる．HDD の回転数の制御とは，HDD を異なる回転数で動作させることにより省電力を図る手法である．

文献 [36, 37] は，HDD の回転数を動的に変化させ，スピンドルモータの消費電力を削減する Dynamic Rotation Per Minute (DRPM) と呼ばれる手法を提案している．DRPM は，一定期間入出力の応答時間を観測し，観測が終了すると過去 n 期間分の入出力応答時間の平均値と直近に観測された入出力の平均応答時間を比較する．もし直近の応答時間の増加率が *upper tolerance* を超えていれば，DRPM は入出力の応答時間を下げるために HDD の回転数を最高速度に設定する．増加率が *lower tolerance* を下回れば，DRPM は HDD の消費電力を削減するために，HDD の回転数を下げる．

文献 [15] は，ネットワークサーバに用いられる HDD について，HDD を省電力状態に移行する契機の制御，高性能 HDD を複数の低性能 HDD に置き換える方法，高性能 HDD とラップトップ HDD を組み合わせる手法，及び複数の回転数を持つ HDD を用いる方法について，それぞれの消費電力を比較している．そして，2 段階の回転数を持つ HDD がネットワークサーバの省電力に適していることを述べている．

これまで述べた，複数の回転数をもつ HDD は，省電力の効果は高いものの商用システムではほとんど用いられておらず，実用性に欠ける．一方で近年の HDD は Acoustic Mode と呼ばれる，HDD の性能と騒音レベルを利用者の好みに応じて設定できる機能を有して

いる [43, 42]．本機能は，HDD の騒音レベルに加え，消費電力も削減することが可能である．文献 [43] によれば，Acoustic Mode を用いることより，主にシークの間の消費電力が 13W から 11W に低減されることが示されている．文献 [16] は，この Acoustic Mode を用いた HDD の省電力の効果について報告している．そして HDD の Acoustic Mode が power capping に有効であること，IOPS が一定である場合やマルチスレッドアプリケーションの場合に効果が高いこと，及び Idle 時やシーケンシャルアクセスに対しては効果がないことを示している．

本研究との関係

HDD の電力状態の制御に関する研究は，そのほとんどが HDD に対する入出力の挙動を観測することにより得られる情報に基づいて，HDD を省電力状態に移行するかどうかを決定している．

また，文献 [32, 31] は，OS 層のプログラムカウンタの情報を用いて入出力発行契機を予測しようとしている．しかしデータセンタの主要なアプリケーションである DBMS では，アプリケーションから発行されたクエリやデータベースに格納されているデータの値により挙動が決まる．このため，プログラムカウンタを用いても，HDD の電源 ON に要する時間より先の入出力発行時期の予測は困難と考えられる．

文献 [99] に示すようなインタフェースを用いる手法は，アプリケーションのソースコードが公開されている場合は活用できる可能性があるが，商用ソフトウェアなどソースコードの入手が困難である場合は適用できず，データセンタにおいてこれらの手法を用いることは困難である．

一方，本論文において提案する手法は，アプリケーションレベルの入出力挙動特性をストレージの省電力に用いる．アプリケーションレベルの入出力挙動は OLTP や DSS などアプリケーションの種類によりほぼ決まっている．この情報を用いることで，データセンタで稼動する多様なアプリケーションの入出力挙動をより正確に予測することが可能になると考えられる．

3.1.2 入出力発行間隔の制御

入出力発行間隔の制御は，ディスクを省電力状態で動作する機会を増やすために，HDD への入出力の発行を制御する手法である．本手法には，コントローラなどのキャッシュを用いて入出力発行間隔を制御する手法，入出力発行間隔が伸びるようアプリケーションを改変する手法に分けられる．

キャッシュを用いた入出力発行間隔の制御

文献 [72] は，ファイルのアクセスパターン (ランダム，シーケンシャル) やアクセス時刻などのヒント情報を元に，どのデータをキャッシュに先読みするかを決定すると共に，Idle 期間中に発生したミスの種類により先読みに使用するキャッシュサイズを決定する省電力手法を提案している．

アプリケーションの改変による入出力発行間隔の制御

また，HDD の省電力に適したようにアプリケーション側を改変する手法も提案されている．文献 [38] は，HDD の Idle 時間が伸びるようアプリケーションのコードを変換するとともに OS 層に Idle 時間の長さを伝えることにより，OS 層による HDD の電力状態の制御を支援する．さらに，コードの変換を自動的に行うコンパイラフレームワークを提案している．

文献 [87] は，科学技術向けアプリケーションを対象に，コンパイラがアプリケーションコードを解析し，適切な箇所に HDD の電源 ON/OFF などの制御コードを追加するとともに，HDD 上のファイルレイアウトを参照し，HDD への入出力発行間隔が広がるように，プログラム側を再構成する手法を提案している

本研究との関係

本論文で提案する手法においても，キャッシュを用いて入出力発行間隔の延伸を図る．しかし，従来の研究が read 入出力のみを対象とし，HDD の入出力性能に関する指標である入出力数やランダム/シーケンシャル入出力などの情報を用いて入出力発行間隔の延伸を試みているのに対し，提案手法は DBMS がログ先行書き出しプロトコルを用いていることを知った上で DB 領域のみ write 入出力の発行間隔を延伸するなど，アプリケーション側の入出力挙動との連携を意識している．さらに，入出力性能に関する指標のみではなく，HDD の省電力に必要となる入出力発行間隔などの情報を用いて省電力を試みている点が異なる．

また，提案手法は文献 [38, 87] 等と異なりアプリケーションコードの改変は行わず，より汎用的に使用できるフレームワークの構築を目指している点がこれらの手法とは異なっている．

3.1.3 データ配置の制御

HDD の省電力効果を高めるために，単一 HDD 内，あるいは複数 HDD 間でデータの配置を制御する手法も提案されている．

単一 HDD 内のファイル配置の制御

HDD では，大きなファイルの read/write と小さなファイルの read/write に要する消費電力にはあまり差がない．文献 [54] において述べられている手法は，この HDD の電力特性に基づき，同時にアクセスされる可能性が高いファイルをグループ化して配置する．そして，グループ内のファイルのどれかにアクセスがあると，グループ内のファイルをまとめて読み出すことにより，HDD を省電力状態にできる時間を増やすことを目指している．

複数 HDD に跨るデータ配置の制御

複数の HDD に跨ってデータ配置を制御することにより HDD の消費電力を低減する手法は多数提案されている．これらの手法は，全ての HDD が常に最大入出力性能で動作している訳ではなく性能余力が存在すること，及び HDD に配置されているデータ毎の入出力数は，データにより大きく異なることを利用するしている．すなわち，入出力頻度が高いデータと入出力頻度が低いデータを異なる HDD に配置し，アクセス頻度が少ない HDD を作り出す手法である．そしてアクセス頻度が少ない HDD を省電力状態とすることにより，HDD の消費電力の削減を試みる．

文献 [19, 20, 3] は，今日では省電力機能を有するディスクアレイを意味する Massive Arrays of Idle Disks (MAID) を提案している．MAID ではキャッシュ用の HDD を設け，高頻度でアクセスされるブロックをキャッシュ用の HDD にコピーする．そしてキャッシュ用の HDD 以外の HDD を長時間 Idle 状態とし，これらの HDD を省電力状態とすることにより HDD の省電力を図る (図 3.1).

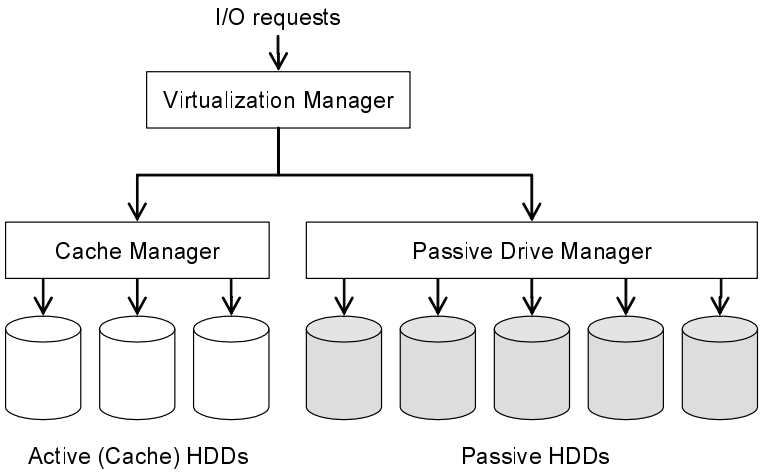

図 3.1: キャッシュHDD を持つ MAID の構成

文献 [75] は，Popular Data Concentration (PDC) と呼ばれる手法について述べている．本手法は，MAID と似ているが，入出力頻度が高いブロックのコピーではなく，HDD 間のデータ移動をファイル単位で行う．これにより，少数に HDD にアクセスを集中させ，残りの HDD を省電力状態に移行する．ファイルシステムに提案手法を実装し，2 段階の速度を持つ HDD を用いた場合に MAID より効果が高いことを示している．

また，アプリケーションとの連携を意識した手法も提案されている．文献 [86] で述べられている手法は，まず典型的な入力データに基づきアプリケーションコードをプロファイリングし，その入出力情報を取り出す．次に，入出力情報に基づき，いつ，どのブロックがアクセスされるかを計算する．そして，この計算結果に基づき HDD の Idle 時間を伸ばすためのブロック配置を決定する．文献 [86] は本手法のアルゴリズムを示すと共に，コード変換との組合せによる Idle 時間のさらなる延伸について議論している．文献 [70] は，科学技術計算向けのファイル配置を応答時間と消費電力の観点から解析するためのモデルを提案し，それを用いて HDD のファイル配置を行う手法を提案している．アプリケーショ

ンとの連携を意識した手法では，ファイルシステム層から HDD 群の省電力を図る手法もある．文献 [28] は，Log Structured File System の write が，常にファイルの最後尾への追記であることに着目し，ログが書かれる HDD 以外の HDD を省電力することにより，HDD 群の消費電力の削減を図っている．

これまで，性能と消費電力の観点からデータ配置に関する研究について述べてきたが，信頼性維持と消費電力の低減を目指した研究もある．文献 [88] は，HDD に NVRAM 及び CPU を組合せたノードを多数ネットワークで接続した，安価な低消費電力アーカイブシステム Pergamum を提案している．Pergamum は，ブロック内及びノード間でパリティ情報を持つ．これにより，ブロックの破損時に他のノードを起動することなくデータの回復を行うと共に，ノード自体が破損した場合に最小限のノードの起動のみでデータの回復を可能としている．

本研究との関係

従来のデータ配置制御に関する研究は，主に物理ブロックの挙動を観測しブロック配置を決定する手法，及びアプリケーションが使うファイルなどのデータの入出力挙動を観測しデータ配置を決定する手法に分けられる．本論文で提案する手法は，アプリケーションの個々のデータの入出力挙動を用いてデータの配置を決定する点で後者に近いが，従来技術がファイルの入出力頻度のみを利用してファイルの配置を決めるのに対し，提案手法はアプリケーションが使用するデータの入出力挙動をパターン化し，それを用いてデータ配置を決定する汎用的なフレームワークを提案している．データ毎の入出力パターンを用いることにより，提案手法をデータセンタで稼動する多様なアプリケーションに容易に適用することが可能となる．

3.2 RAID 及びストレージの省電力に関する研究

データセンタで稼動するストレージは，入出力性能の向上と信頼性向上の観点から複数台の HDD を用いた RAID [73, 78] 構成を取ることが普通となっている．本節では，RAID を構成する HDD，及び複数の RAID を持つストレージを対象とした省電力手法について述べると共に，本研究との関係をまとめる．

3.2.1 RAID を構成する HDD の省電力に関する研究

RAID を構成する HDD の省電力に関する研究は，コントローラが持つキャッシュを利用して入出力数の削減や発行間隔の制御を行うことにより RAID を構成する HDD の一部を省電力状態に移行する手法，HDD へのデータ配置やパリティ配置を工夫することにより同様の効果を狙った手法などが見られる．

キャッシュを利用した省電力に関する研究

文献 [55, 53] は，RAID1 及び RAID5 を構成する HDD の消費電力の削減を目的とした，入出力スケジューリングとキャッシュ管理ポリシーである EERAID について述べている．

RAID1 の入出力スケジューリングを行う EERAID1 は，プライマリ HDD あるいはミラー HDD に N リクエストごとに入出力を交互に発行し，入出力が発行されていない側の HDD を省電力状態に移行する．RAID5 を対象とした EERAID5 は，キャッシュからブロックを追い出す場合に，当該ブロックの書き出し先となる全ての HDD が Active/Idle 状態であるブロックを優先して書き出すことにより，HDD をより長時間省電力状態に保とうとする．

また，文献 [94] は，RAID1 を対象とした，eRAID と呼ばれるエネルギー削減ポリシーについて述べている．eRAID は，EERAID1 で示された入出力の分配ポリシーの性能をキューイングネットワークモデルを用いて解析すると共に，従来のミラーリングよりより省電力可能なデータ配置について述べている．さらに，文献 [95] は，eRAID を RAID5 に拡張し，同様の性能解析を行っている．

文献 [109, 22] は，キャッシュを利用して，RAID を構成する HDD を省電力する手法について述べている．本手法は，キャッシュされたブロックをキャッシュから追い出す際に，当該ブロックの再読み込み電力が最も小さいと考えられるブロックをを追い出す．入出力トレースを用いてブロックの入出力発行間隔のヒストグラムを作成してブロックの入出力間隔の長さの確率を求め，それをもとにブロックを優先 (入出力間隔が長いブロック)，及び通常の 2 つのクラスに分割する．そして優先クラスに対して先に述べたキャッシュ追い出しアルゴリズムを適用することにより，RAID を構成する HDD の省電力を図る．

文献 [104, 93] は，RIMAC と呼ばれる，RAID の冗長性と大規模なキャッシュを活用した省電力手法を提案している．RIMAC は，RAID を構成する HDD が省電力状態である場合に，当該 HDD に格納されているデータを Active 状態の HDD に格納されたブロックを用いて復元する．これにより当該 HDD の passive spin up を回避し，入出力性能の維持と HDD の省電力を図っている．

ブロック配置を利用した省電力に関する研究

文献 [77] は，Diverted Access と呼ばれる，RAID の冗長性を活用した省電力手法を提案している．Diverted Access は，オリジナルブロックと冗長ブロックを異なる HDD に配置し，冗長データが入っている HDD を省電力モードに移行することにより，RAID を構成する HDD の省電力を図る．さらに，Diverted Access や従来の手法を用いた場合の HDD の消費電力を予測するモデルを導入し，実ワークロードを用いたシミュレーションにより評価を行っている．

文献 [98] は，データセンタの入出力負荷が変化することに着目し，RAID を構成する HDD において電源を ON にする HDD を動的に変更する手法 (PARAID) について述べている．PARAID は，RAID 内のパリティ配置を不均一にすることで Active/Idle 状態の HDD 数の変更を可能としている．これにより，入出力負荷が低い場合に RAID を構成する HDD の省電力が可能となる．

文献 [102] は，RAID を構成する HDD を Hot と Cold の 2 種類に分け，Cold 側の HDD を低速で回転することにより RAID を構成する HDD の消費電力の削減を図っている．Cold 側を低速で回転させた場合の性能への影響を小さくするために，ブロックを入出力数が多いブロックと少ないブロックの 2 種類に分け入出力数が多いブロックを Hot HDD に，入出力数が少ないブロックを Cold HDD に配置する．

本研究との関係

これらの手法は単一の RAID グループ内の HDD の省電力を対象としているのに対し，本研究はデータセンタで用いられている複数台のストレージの省電力を対象としている．本研究で提案する手法は，アプリケーションレベルの入出力挙動をパターン化し，データ配置制御やキャッシュを利用した入出力間隔の延伸など入出力パターンに適した省電力手法を選択する．このため，従来の手法と比較して汎用性が高い．

3.2.2 複数の RAID を持つストレージの省電力に関する研究

ストレージの省電力に関する手法は，そのほとんどが RAID グループ間でデータを移動もしくは複製することにより，アクセス頻度が高い RAID グループと低い RAID グループを作成し，アクセス頻度が低い RAID グループを省電力状態に移行する手法を採用している．

RAID グループ間のデータ移動又は複製を利用した省電力に関する研究

文献 [108] は，PDC のアイデアを拡張した Hibernator について述べている．Hibernator は，HDD アレイを複数の Tier に分割し，Tier 毎に HDD の回転数を設定する．そして，ブロックの応答時間の閾値などに基づき，ブロックを Tier 間で移動する．性能と消費電力のバランスを適切に取ることが期待できる．

文献 [112, 110, 26] は，HPC 向けアプリケーションと連携した，2 次ストレージの省電力手法について述べている．HPC などで使用されるスーパーコンピュータは，ジョブと呼ばれる単位でプログラムを実行する．本手法は，HPC アプリケーションが使用するジョブスケジューラからジョブのスケジューリング情報を取得しこれを基にジョブが実行される時期を予測する．そして，予測に基づき，ジョブがアクセスするデータが格納されたディスク筐体の電源を ON にする．ジョブが終了すれば当該ディスク筐体の電源を OFF にする．これらの処理により，2 次ストレージの省電力を図っている．

文献 [68, 69] は，Dynamic Data Reorganizaiton (DDR) と呼ばれる，RAID グループ間でのブロック交換によりストレージの消費電力を削減する手法について述べている．DDR は，RAID グループに対する入出力数に基づき RAID グループを Hot RAID グループと Cold RAID グループに分割する．そして，Cold RAID グループを省電力状態に移行する．Cold RAID グループ内のブロックにアクセスがあると，DDR は当該ブロックを，Hot RAID グループ内で最も入出力数が少ないブロックと交換する．また，DDR は Hot RAID グループ間で入出力の負荷が均一になるように，ブロックの交換を行う．

文献 [92] は，SRCMap と呼ばれる省電力手法について述べている．SRCMap は，RAID グループ内の入出力数が多いブロック群のレプリカを少数の RAID グループ内に作成し，他の RAID グループを省電力状態に移行する．ストレージの仮想化層でこれらの処理を行う点に特徴がある．

文献 [34] は，近年のストレージに採用される，SSD, SAS, SATA 等の性能や容量，電力特性が異なるメディアを搭載したマルチ Tier ストレージを対象とした省電力手法である EDT について述べている．EDT は，アプリケーションの入出力トレースを解析して適切

な Tier を提案する構成アドバイザーと，構築された Tier 間でデータを動的に再配置する動的 Tier マネージャから構成される．EDT は，SSD の使用，及び使用されていないデバイスを省電力状態に移行することにより，ストレージの省電力する．

その他の研究

文献 [59] は，アプリケーションから受け取ったヒント情報に基づき，データの先読み及び write の遅延を行う手法である，GreenStor について述べている．GreenStor は，アプリケーションからどのブロックがいつ必要であるかを示す情報を受け取り，先読みのスケジューリングを行う．さらにストレージ内にキャッシュディスクを設け，write を一時的に遅延することにより，ストレージの省電力を行う．

文献 [23] は，write offloading と呼ばれるストレージ省電力手法について述べている．Write offloading は，write されたブロックを格納するボリューム (ディスクアレイ又は RAID グループ) が省電力状態であった場合に，当該ブロックを Active/Idle 状態のボリュームに一時的に書き出し，後で本来のボリュームに書き戻す．これによりボリュームの spin up 回数を削減する．本手法は，エンタープライズシステムでは，write が主体のボリュームがいくつかあり，これらのボリュームの write を 0 とした場合に active/idle 状態としなければならないボリュームの数を大幅に削減できるとの観測結果に基づいている．

文献 [33] は，実際のエンタープライズワークロードを解析し，ストレージを 40%から 75%省電力できる可能性があることを述べている．さらに，アプリケーションの性能要件や入出力挙動を特徴づけし，これまでに提案されている様々な手法が，どの特徴に適しているかを整理している．

本研究との関係

提案手法も，RAID グループ間でデータを再配置し，Hot な RAID グループと Cold な RAID グループ数を生成し，Cold な RAID グループを省電力状態に移行する点においては，これらの手法と似ている．しかし，これら従来の研究が，文献 [112, 110, 26] を除きストレージ内部の情報のみを用いてブロックの入出力挙動を予測しようとしているのに対し，提案手法はアプリケーション側の入出力挙動特性を用いている点が異なる．アプリケーションが用いるデータの入出力挙動はストレージの物理ブロックの入出力挙動より安定している．このため提案手法は省電力効果を高めることができると考えられる．

また，文献 [112, 110, 26] では HPC 向けアプリケーションで多く用いられるジョブスケジューラから得られる情報に基づき入出力が発行される契機を予測している．しかしジョブスケジューラを用いない対話的処理に対してはこのような手法を採用することはできない．提案手法はアプリケーションの入出力挙動特性を用いるため，対話的な処理に対しても適用可能である．

3.3 アプリケーションによる IT 機器の省電力に関する研究

文献 [83] では，TPC-H を用いたシステムの IT 機器の消費電力を削減するためのハードウェアの構成と DBMS の設定について議論している．文献 [83] では，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に重要な役割を果たす可能性があると述べている．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，DBMS の性能と消費電力の関係を調査した研究も報告されている．文献 [62] では，意思決定支援システムのベンチマークプログラムである TPC-H[10] を実行した場合の，サーバ及びディスク等の主要なハードウェアの消費電力を計測し報告している．さらに，性能と消費電力の関係を調査し，大幅な性能低下をさけつつ消費電力を大きく下げられる可能性があることを報告している．文献 [79] では，文献 [62] よりも多くの種類のハードウェアを用いて，ストレージの構成 (ディスク台数やメディアの種別 (ハードディスク, SSD など)) や CPU の省電力機能の使用の有無，メモリサイズを変化させた場合の TPC-H の性能と IT 機器の消費電力の関係を調査した結果を報告している．文献 [90] は，DBMS が行うスキャンやジョイン，ソートなどの処理と CPU スケジューリングとの関係を調査している．調査の結果，現在のサーバでは最も性能のよいクエリプランが最もエネルギー効率が高くなることを示している．将来ハードウェアのエネルギー効率が上昇し Idle 時の消費電力が減少すれば，より省電力に適したクエリプランの選択の余地が広がること，及び最大消費電力を抑えるパワーキャッピングを採用する場合は省電力に適したクエリプランの選択の余地が高まる可能性があることを述べている．文献 [113] は，オンライントランザクション処理における DB サーバの省電力を目的として，CPU をはじめとするサーバの各コンポーネントの消費電力の計測を行うと共に，DVFS を適用した際の効果について報告している．

アプリケーション側からの IT 機器の消費電力を削減する具体的な手法としては，次のようなものがある．文献 [52] は，DBMS の実行中にプロセッサの電圧や周波数を変化させる PVC と呼ぶ手法，及び同一コンポーネントに対するクエリを集約する QED と呼ぶ手法について述べている．文献 [11] は，金融機関向けのアプリケーションを例に，ソフトウェアによるサーバの省電力手法について述べている．本手法は，関数の実行結果を再利用することにより計算量を減らし，決められた処理を行うために必要となるサーバの消費電力を削減する．文献 [48] は，GreenHDFS と呼ばれる，HDFS を実行するサーバの省電力手法について述べている．本手法は，Yahoo で使われているシステムを対象としており，ファイルが配置されるディレクトリ名とファイルのライフサイクルに関係があることを利用してファイルのライフサイクルを予測する．そして，予測したライフサイクルに基づき，ファイルの階層管理を行う．アクセス頻度が低い階層のサーバを省電力状態にすることにより，HDFS が稼働するサーバの消費電力を低減する．文献 [63] は，HPC 向けアプリケーションの将来のハードウェア利用予定を用いてサーバの省電力を行う手法について述べている．本手法は，ハードウェアの利用予定を伝えるコードをアプリケーションに挿入することにより，将来のハードウェア利用を省電力機構に伝える．文献 [27] は，SLA 違反と消費電力を最小化する手法について述べている．本手法は，まずアプリケーションのワークロードトレースを解析してワークロードのベースとなる長期的な負荷変動パターンを識別する．そしてこのパターンに基づき Active 状態のサーバ数を変更する．

本研究との関係であるが，これらの手法はいずれも DBMS などのアプリケーションが

持つ知識を用いて IT 機器の省電力を行おうとしている点で提案手法と類似している．これらの手法はデータセンタの設計や構築に関する一つの解を提示しているが，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．または，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．これまで提案されているアプリケーション側からの IT 機器の省電力手法は，Hadoop や HPC などバッチ的な挙動特性を持つアプリケーションが稼動するサーバの省電力を対象としている点が異なる．提案手法は，対話的なアプリケーションも含めたストレージの省電力を対象としている．

3.4 データセンタの省電力に関する研究

データセンタの省電力に関する研究には，ファシリティ制御に関する研究，空調制御に関する研究，サーバ及びネットワークの省電力に関する研究等がある．本節は，これらの研究動向及び本研究の位置付けについてまとめる．なお，ストレージの省電力に関する研究，及びアプリケーションと連携した省電力に関する研究については前節まで述べており，本節では述べない．

3.4.1 ファシリティ制御に関する研究

ファシリティ制御とは，空調機器と IT 機器を併せた電力の削減を試みる研究である．そのほとんどが，ワークロードの IT 機器への配置を調整することによる消費電力の削減を試みている．

まず，空調及びサーバの双方を対象とした省電力に関する研究について説明する．文献 [96] は，ラックに収められた blade server を対象に，blade server の消費電力と blade server に冷気を送るファンの消費電力の合計を最小化するよう，blade server のワークロードを管理する手法を提案している．文献 [12] はサーバのアイドル時消費電力と空調の消費電力のトレードオフを図ると同時に，サーバにジョブを過剰に配置することによりサーバの稼働台数を減らす手法を，文献 [71] は空調電力とサーバ電力を最小化するジョブの配置を Linear Programming により求める手法をそれぞれ提案している．文献 [17] は，動的なリソース割当て，ワークロード配置，及び空調制御の間の相互関係を考慮した，データセンタの省電力手法について述べている．文献 [57] は，温度とサーバの電力を抑えつつ，MapReduce のスループットを最大化する手法について述べている．

文献 [30] は，サーバ，空調に加え，ストレージも含めたデータセンタの省電力手法について述べている．本手法は，まずエネルギー消費を意識した階層ストレージを構築し，次に冷却とワークロードの配置変更を組み合わることによりデータセンタの消費電力を削減する．ストレージも含めたデータセンタ全体の省電力を図っている点が特徴である．

データセンタの可視化に関する研究もいくつか報告されている．文献 [60] は，データ解析，可視化，知識発見技術の使い方の調査結果，およびこれらを電力，冷却，計算の 3 サブシステムに適用する際のユースケース，効果的な使い方を提案している．また，文献 [40] は，エネルギー効率，IT 機器の吸気温度，及び空調の効率を可視化するダッシュボードについて述べている．膨大な量の情報を要約し，時計を模した単純な GUI で表示している点が特徴である．

本研究との関係であるが，ここで述べられた手法のほとんどが，サーバへのワークロードの配備による省電力を図っている．本研究はストレージの省電力を対象としている．ストレージ間のデータの移動は，移動するデータ量がプログラムや仮想サーバと比較して大きいため，プログラムや仮想サーバの移動と比較して時間が掛かる．本研究ではこれを意識し，データ配置を決定するにあたりデータ移動量をできるだけ削減しようとしている点が，従来のサーバを対象とした移動手法とは異なる．

3.4.2 空調制御に関する研究

空調制御に関する研究とは，データセンタの消費電力の 50%以上を占めるとも言われる空調機の消費電力削減を目的とした手法である．

文献 [65] は，データセンタ内の冷却能力が場所により異なるため，ジョブをサーバに均一に配備したのではホットスポットが生じ冷却により多くの電力が必要になることを指摘し，温度が低いサーバにより多くのジョブを配備する手法を提案している．文献 [35] は，IT 機器のピーク吸気温度を最小化することが，冷却電力を最低限に抑えることを示している．線形の熱循環モデルを用いて，データセンタにおけるタスク割当によるピーク吸気温度の最小化問題を定義し，GA 及び逐次 2 次計画法を基にした手法によりこの問題を解いている．文献 [91] は，データセンタにおけるジョブのスケジューリングによる空調電力削減に，温度に加えエア・フローを取り入れる手法について述べている．スケジューリングはワークロードの短期間の変化など予測が困難な部分に，エア・フローは長期間の変動にそれぞれ対応する．文献 [45] は，データセンタにおける，温度を意識した 2 階層型の電力最適化手法について述べている．データセンタの HCAC(Heating, Ventilation and Air Conditioning) の電力とサーバファンの電力間で電力を融通することにより，データセンタの省電力を図る点に特徴がある．文献 [85] は，Hadoop が稼動するストレージを中心としたデータセンタの温度と電力を考慮したタスクスケジューリング手法について述べている．本手法は，ノードの信頼性を確保するため，ノードの温度が閾値を超えないようにしつつ空調の電力を最小化しようとする点に特徴がある．

本研究で提案する手法はストレージの省電力を対象としている．空調も含めたストレージの省電力は今後の課題である．

3.4.3 サーバ及びネットワークの省電力に関する研究

データセンタにおけるサーバやネットワークの消費電力の削減に関する研究も見られる．そのほとんどが，負荷が低い場合にワークロードあるいは仮想サーバ (VM) を少数の物理サーバに集約し，他の物理サーバを省電力状態に移行する．

まず，ワークロードのプロビジョニングに関する手法について述べる．文献 [76] は，ワークステーションあるいは PC のクラスタの消費電力削減を目的に，クラスタを構成するサーバの電源を動的に ON/OFF する手法のアルゴリズムについて述べている．本アルゴリズムは，サーバの負荷や電力，性能に基づきワークステーションの構成案を生成する点に特徴がある．文献 [66] は，仮想サーバによる物理サーバリソース共有環境下における物理サーバの省電力手法について述べている．本手法は，仮想サーバの独立性の保障，既存の

アプリケーションポリシーの活用，及びヘテロ環境への対応が特徴である．文献 [67] は，仮想サーバ視点での電力管理手法について述べている．本手法の特徴は，仮想サーバ間における電力使用を融通し合うことにより，仮想サーバが動作する物理サーバの電力制約を満たす点である．文献 [21] は，サーバの消費電力及び性能が目標を満たすための手法及びその実装について述べている．統合管理ではなく，エージェントを用いてサーバを分散管理する点に特徴がある．文献 [50] は，クラウド環境下における，金融などのリアルタイムサービス向けの仮想サーバの省電力プロビジョニング手法について述べている．本手法の特徴は，リアルタイムサービスをリアルタイム仮想マシンリクエストとしてモデル化した点，及び Dynamic Voltage Frequency Scaling スキーマを用いた仮想サーバのプロビジョニングである．文献 [13] は，サーバの電源 ON/OFF，省電力指向の集約，性能を最大化するための機械学習等，多様な省電力手段を提供するためのフレームワークについて述べている．文献 [49] は，Data Aware Scaling Down (DASCA) と呼ぶ，MapReduce のスケールダウンによる省電力手法について述べている．MapReduce をスケールダウンする際は，どのノードを省電力状態に移行するかが問題となるが，本手法は，チャンク数が最も少ないノード，及び利用できないデータが最も少ないノードをベストエフォートアプローチで選択する．また，MapReduce は信頼性を確保するため最低 3 つのノードにレプリカを作成するが，ノードが省電力状態の場合はレプリカを利用できないという課題がある．これに対し，本手法はレプリカの数がより少なくても信頼性を維持できることを示すとともに．適切な数のレプリカ数を計算する手法について述べている．文献 [100] は，実行時のリソース配置調整と DVFS を用いた，MapReduce 環境の消費電力削減手法について述べている．実環境での評価により，適切なノー度数の選択と DVFS スケジューリングを用いることにより消費電力を大きく下げることが可能であると述べている．

また，異なる種類のサーバ群の省電力を対象とした手法も見られる．文献 [81] は，均一な構成のクラスタより，Tier 毎にサーバの種類が異なるヘテロ構成のクラスタの方が省電力効果が高いという観測結果を示している．さらにこの観測から，Web サービスのワークロードをプロファイルし，与えられた QoS の下でワット数当りのスループットを最大化するようクラスタ構成を決める手法について述べている．

文献 [107] は，単一サーバの省電力であるが，サーバの温度を意識したバッファ管理を行う．本手法は，LRU 中のブロックを追い出す際に，LRU の後方にあるブロックのうち，メモリ中の同一 rank 内のブロックを優先的に追い出すことにより，メモリのアクセスされる範囲を局所化することにより温度の上昇を低くしている．

また，文献 [84] は，データセンタにおける高密度ネットワークの省電力ルーティングについて述べている．その主たるアイデアは，ネットワーク性能の劣化を抑えつつできるだけ少数のデバイスを使用するようにルーティングを設定すること，及びアイドル状態のネットワークデバイスを省電力状態に移行することである．

本研究との関係であるが，先に述べたように，本研究はストレージの省電力を対象としており，データの移動量も考慮している点が，これらの研究とは異なる．

第4章　MAID機能とハードディスク及びストレージの消費電力特性

本章では，HDD及びストレージのMAID機能について説明する．次に，HDD及びストレージの消費電力を計測し，その特性について述べる．また，ストレージ省電力の指標であるBreak Even Timeについて説明する．

4.1　MAID機能

4.1.1　HDDの省電力機能

本節では，市販のHDDを例に，HDDの取りうる電力状態，及びそれら電力状態のアプリケーションからの制御方法を述べる．

HDDの電力状態

本節では，Seagate社製のSATAハードディスク(HDD)[5]を例に，HDDの省電力機能について説明する．HDDは，通常表4.1に示す4つの電力状態を持つ．以下，これらの状態について述べる．

表4.1: ハードディスクの電力状態

電力状態	ヘッド	スピンドル	バッファ
Active	Tracking	回転	利用可能
Idle	Tracking	回転	利用可能
Standby	Parked	停止	利用可能
Sleep	Parked	停止	利用不可

Active HDDはread/write処理，あるいはseek処理を実行中である．

Idle HDDのバッファは利用可能であり，HDDは全てのコマンドの受付が可能である．ディスクアクセスが必要な場合は即座にActive状態に移行する．

Standby サーバがStandby ImmediateコマンドをHDDに送ると，ドライブはStandby状態に移行する．サーバがStandbyタイマを設定した場合は，タイマにより指定された間HDDがinactiveであれば，HDDは自動的にStandby状態に移行する．Standby

状態では，HDD のバッファは利用可能である．ヘッドは退避しドライブの回転は停止する．HDD は全てのコマンドの受付が可能であり，ディスクアクセスが必要になれば，HDD は Active 状態に移行する．

Sleep サーバから Sleep コマンドを受け取ると，HDD は Sleep 状態に移行する．Sleep 状態ではバッファは利用不可であり，ヘッドは退避，ディスクの回転は停止する．サーバから Hard Reset または Soft Reset コマンドを受け取ると，HDD は Standby 状態に移行する．

上記の 4 種類の状態のうち，Standby と Sleep が省電力状態である．HDD の省電力では，HDD を Standby あるいは Sleep 状態に移行させることにより，省電力を実現する．

これらの状態のうち本研究では Active, Idle, Standby の 3 状態のみを用いる．これは，Sleep 状態の消費電力は Standby とほぼ同等であること，及び Sleep 状態から他の状態への遷移には HDD のリセットが必要であるが，アプリケーション実行中にはリセット操作を行うことができないためである．

HDD の電力状態の制御方法

ディスクの電力状態には，ホスト側が主導してディスクを省電力状態に移行する Host Initiated Link Power Managemet (HIPM) と，ディスク側が主導してディスクを省電力状態に移行する Device Initiated Link Power Management (DIPM) が存在する[1][61, 18].

HIPM はホストがデバイスに対して省電力状態への移行を指示する機能であり，ホスト側のハードウェア又はソフトウェアで実現される．通常，ホストはデバイスに対して発行されるコマンドが何であるかを知っているため，タイムアウト期間を発生させることなく即座にデバイスを省電力状態に移行することができる．HIPM の例としては，Linux の *hdparm* コマンド [2] がある．hdparm コマンドではオプションにより，HDD を即座に Standby あるいは Sleep 状態に移行することができる．

一方の DIPM はデバイス側からホストに対して省電力状態への移行を要求する方法であり，デバイス側で実現される．ドライブは，特定のコマンドが完了するまでにどの程度時間がかかるかを知っているため、コマンドの処理が完了しないうちに，省電力状態への切り替えをホストに要求できる．例えば Seagate の HDD[5] では，Standby timer と呼ばれるタイマ機能を持っており，タイマで指定された時間内に HDD を Active 状態に移行するコマンド (read, write あるいは seek) が発行されなければ HDD を Standby 状態に移行する．タイマの設定には例えば前述の hdparm コマンドを用いることができる．

アプリケーションは，前述の hdparm コマンドを用いて HDD を即座に Standby 状態に移行する，及び hdparm コマンドを用いてタイマを設定することにより，HDD を省電力状態に移行することができる．

[1] SATA Interface の場合

4.1.2　ストレージの構成と MAID 機能

ストレージの構成

本節では，省電力の対象として用いる商用ストレージである日立製作所製の Hitachi Adaptive Modular Storage 2500 (AMS2500) [9] の構成について述べる．

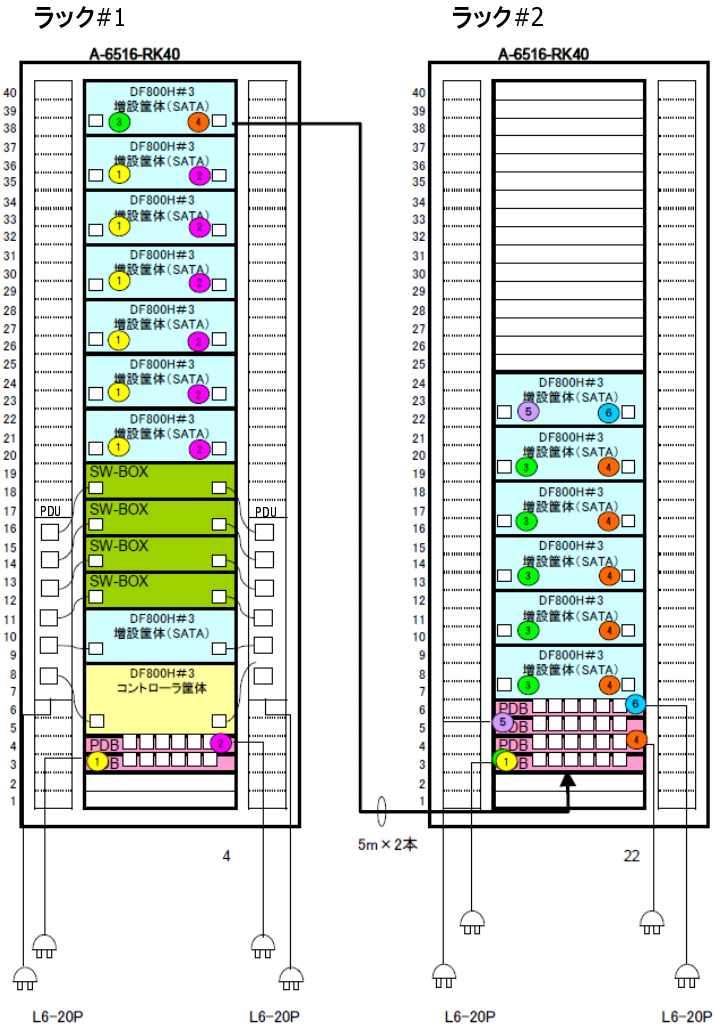

図 4.1: ストレージの構成 (AMS2500)

図 4.1 は 1 台の AMS2500 の構成例を示している．本例では，AMS2500 は 2 本のラックから構成されている．ラック#1 には，Power Distribution Board (PDB)2 台，コントローラ筐体 1 台，Power Distribution Unit (PDU)2 台，増設筐体 8 台，スイッチボックス (SW-BOX)4 台が搭載されている．PDB 及び PDU は 200V のコンセント (L6-20P) から電源供給を受ける．コントローラ筐体，増設筐体，及び SW-BOX は PDB 及び PDU から電力供給を受ける．コントローラ筐体はサーバとのインタフェース及びプロセッサ，キャッシュ等を持ち，サーバとのデータのやり取りや RAID パリティ計算などを行う．増設筐体 (以下，本論文ではディスク筐体と呼ぶ) は HDD を格納する筐体である．本ストレージは，1 台のディスク

筐体は 15 台の 7200 回転の SATA HDD を格納しており，コントローラと Fibre Channel で接続されている．SW-BOX はコントローラとディスク筐体間の Fibre Channel ネットワークスイッチである．ラック#2 には 2 台の PDB と 6 台のディスク筐体を有している．これらのディスク筐体もコントローラ筐体と接続されている．

ストレージの MAID 機能

AMS2500 は MAID 機能を持つストレージである．AMS2500 の MAID 機能は，ディスク筐体内に格納された HDD 自身の省電力機能と，ディスク筐体単位の省電力機能の 2 種類である．本節では，AMS2500 のディスク筐体が取りうる電力状態，及びそれら電力状態のアプリケーションからの制御方法を述べる．

ストレージの電力状態

AMS2500 のディスク筐体は，Active，Idle，Spindown，及び電源 OFF という 4 種類の電力状態を持つ．以下，これらの状態について述べる．

Active ディスク筐体内の HDD は read/write 処理，あるいは seek 処理を実行中である．

Idle ディスク筐体は全てのコマンドの受付が可能である．HDD へのアクセスが必要な場合は即座に Active 状態に移行する．

Spindown RAID を構成する全ての HDD の電源が OFF となった状態である．HDD に対するアクセスはできない．ディスク筐体内の全ての HDD を用いて RAID を構成した場合，当該ディスク筐体内の全ての HDD のみが電源 OFF となる．RAID を構成する HDD 群を Active 又は Idle 状態に戻すためには，外部から RAID グループを構成する HDD の Spinup 指示を与える必要がある．

電源 OFF HDD を含む，ディスク筐体の全ての構成要素の電源が OFF となった状態である．消費電力は 0 Wとなる．HDD に対するアクセスはできない．Spindown 状態と同様，ディスク筐体を Active 又は Idle 状態に戻すためには，外部からディスク筐体の Spinup 指示を与える必要がある．ディスク筐体を Spindown 状態に移行させる場合も，外部から Spindown 指示を与える必要がある．

ストレージの電力状態の制御方法

AMS2500 のディスク筐体の電力状態を制御するためには，AMS2500 に付属するコマンドを用いる．コマンドを用いてディスク筐体の電力状態をコントローラに指示すると，ディスク筐体は指示された状態に移行する．コマンドは，ストレージのコントローラと LAN で接続された管理サーバにインストールして使用する (図 4.2).

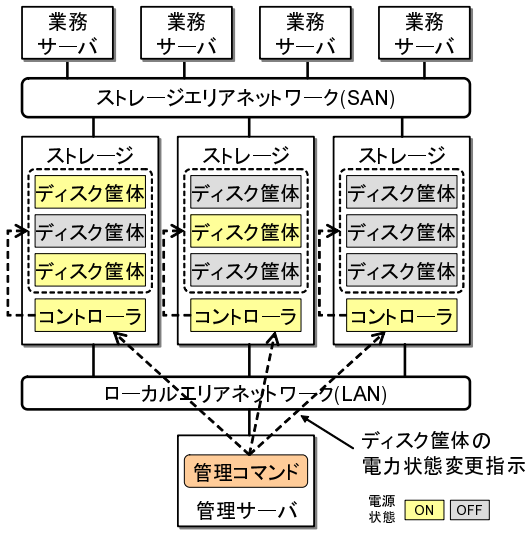

図 4.2: ストレージ電力状態制御コマンド

4.2 HDD 及びストレージの消費電力

本節では，HDD 及びストレージの消費電力の計測値を示し，HDD 及びストレージの省電力のためには省電力機能の活用が重要であること，及び省電力機能の活用には大きなペナルティ(性能及び電力) があることを述べる．さらに，複数台の HDD とディスク筐体の消費電力特性の差異について論じる．

4.2.1 HDD の消費電力特性

HDD の省電力機能を有効に活用すべく，まず HDD の消費電力特性を実測し，その結果を解析した．

計測環境

図 4.3 は，HDD の消費電力の計測に用いた機器の構成図である．

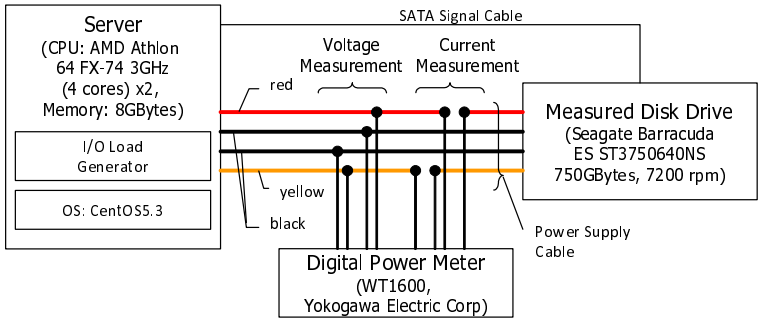

図 4.3: HDD の消費電力の計測環境

電力を計測した HDD は，サーバに直結されている．HDD はサーバから 4 ピンの電源ケーブルにより電力供給されている．図 4.3 中，赤色の線は 5V，黄色の線は 12V である．5V および 12V 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するとともに，5V とグランド (黒線)，及び 12V とグランド間の電圧を計測する．HDD の消費電力は，これら両者の電力の合計値である．サーバの CPU は AMD Athlon 64 FX-74 3GHz(キャッシュサイズ 1MB), 4 コア ×2，主記憶サイズは 8GB である．計測対象の HDD は Seagate 社の Barracuda ES ST3750640NS (750GB, 7200rpm) である．また，計測時は HDD 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 HDD の write キャッシュを使用しないためである．

Active 時及び Idle 時の HDD 消費電力特性

図 4.3 に示した環境を用いて，まずランダム入出力を発行した場合の Active 時及び Idle 時の HDD の消費電力を計測した．計測結果を図 4.4 に示す．図の横軸は HDD に対する秒当りの入出力数 (IOPS)，縦軸は HDD の消費電力を示す．

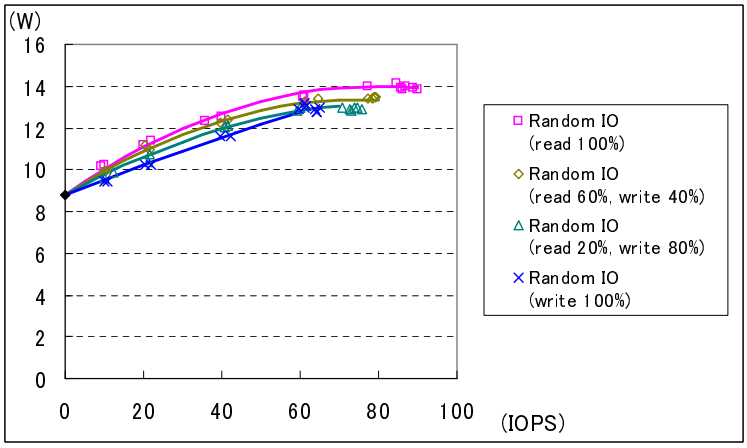

図 4.4: Active 時及び Idle 時の HDD の消費電力特性

図から分かるように，Idle 時の消費電力は約 8.8W であった．そして，入出力数が増加するにつれ HDD の消費電力も増加した．ランダム read が最も消費電力が高く，最大消費電力は約 14.0W，Idle 時消費電力からの増加率は最大 59.1%であった，秒当りの入出力数が増えた倍の消費電力の伸びは線形ではなく，入出力数が増加するにつれ伸び率は低下した．

省電力機能使用時の HDD 消費電力特性

次に，HDD を Standby 状態に移行した場合，及び Standby 状態から復帰した場合 (Spin up) の消費電力を計測した．この結果を図 4.5 に示す．図の横軸は経過時間 (秒)，縦軸は HDD の消費電力を示す．6 秒目で HDD の Spin down を，32 秒目で HDD の Spin up をそれぞれ開始している．

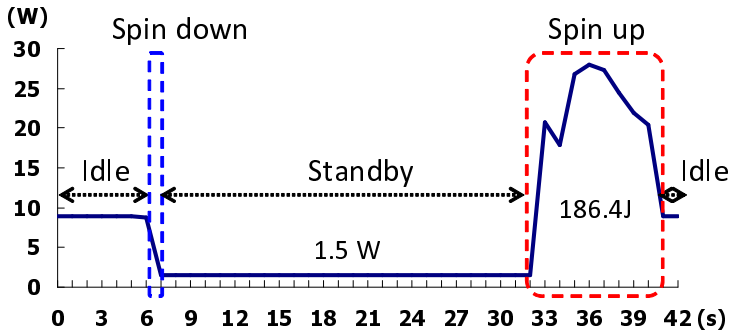

図 4.5: HDD を Standby 状態に移行した場合の消費電力特性

図に示すように，Standby 時の HDD の消費電力は約 1.5W と，Idle 時の約 1/6 まで低下している．その所要時間は約 1 秒である．一方，HDD を Standby 状態から Active/Idle 状態に移行する場合のエネルギーは 186.4J であり，最大消費電力は約 30W であった．これは主に HDD のスピンドルモーターを起動するために消費されたエネルギーである．また，HDD を Standby 状態から Active/Idle 状態に移行するために必要な時間は約 8 秒であった．

これらの計測結果から分かるように，HDD の省電力機能を用いることで HDD の消費電力を大きく削減できる．しかし，同時に HDD の Spin up には多大なエネルギーと長い起動待ち時間が必要であることを示している．これらの結果は，HDD の省電力機能を用いて HDD を省電力するためには，省電力機能を適用する契機や対象を慎重に選ぶことが重要であることを示している．

4.2.2 ストレージの消費電力

次に，ストレージの消費電力特性を実測し，その結果を解析した．

計測環境

図 4.6 は，消費電力の計測に用いたストレージと電力計の接続図である．ストレージは，日立製作所製の Hitachi Adaptive Modular Storage 2500 である．本ストレージは 4 つの入出力プロセッサを有するコントローラ，及び 15 台の HDD を格納するディスク筐体を複数台有することが可能である．ディスク筐体内の 15 台の HDD はデータ用 HDD13 台，パリティ用 HDD 2 台の RAID6 を構成している．HDD は 7200 回転の 750GB SATA ドライブである．計測に用いた環境のストレージのディスク筐体数は 10 である．コントローラ及び各ディスク筐体はそれぞれ 2 本の 200V の電源ケーブルを有している．各電源ケーブルにクランプセンサを接続し，それらを電力計に接続することによりストレージの消費電力を計測した．電力計には HIOKI 製の遠隔計測及びモニタリングシステム 2300 シリーズを用いた．

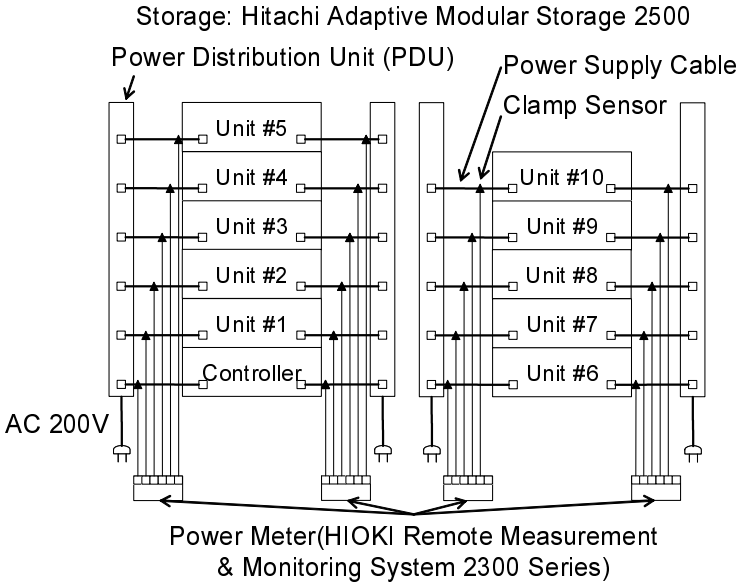

図 4.6: ストレージと電力計の接続

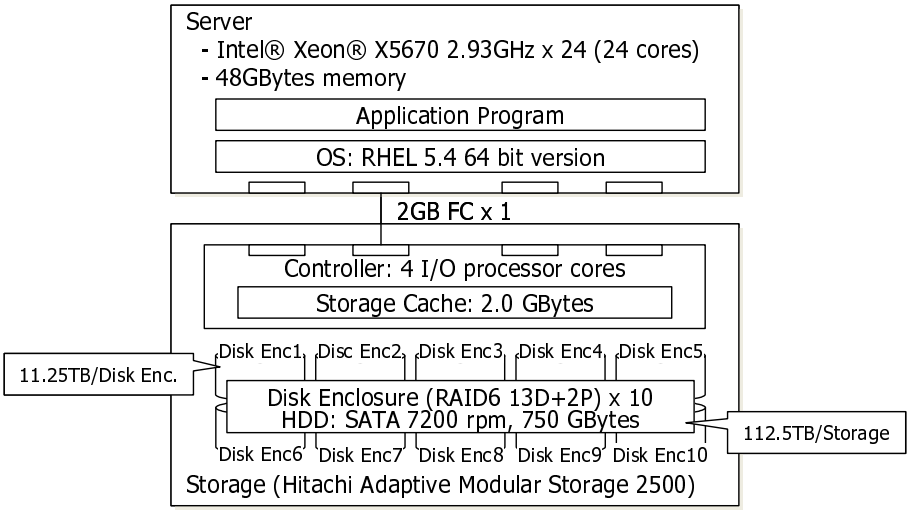

図 4.7: ストレージの消費電力の計測環境

次に，ストレージの消費電力の計測に用いた機器構成を図 4.7 に示す．サーバのプロセッサは Intel Xeon X5670 2.93GHz (合計 24 コア)，主記憶は 48GB である．サーバの OS は Red Hat Enterprise Linux 5.4 (64 ビット版)，ファイルシステムは EXT2 である．ストレージのキャッシュ容量は 2GB，RAID 構成前のディスク筐体の容量は 11.25TB，合計容量は 112.5TB である．サーバとストレージは 4 本の 2Gbit ファイバチャネル 1 本で接続されている．

Active 時及び Idle 時のストレージ諸費電力

ストレージのコントローラに対する秒当り入出力数 (IOPS) とコントローラの消費電力との関係，及びディスク筐体とディスク筐体に対する秒当り入出力数 (IOPS) との関係を図 4.8，4.9 にそれぞれ示す．入出力サイズは共に 8KB である．

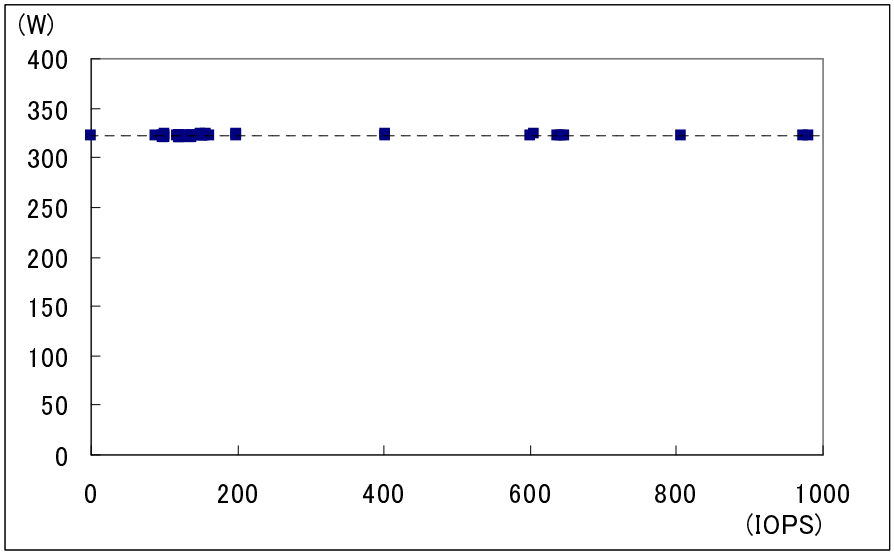

図 4.8: ストレージのコントローラの消費電力特性

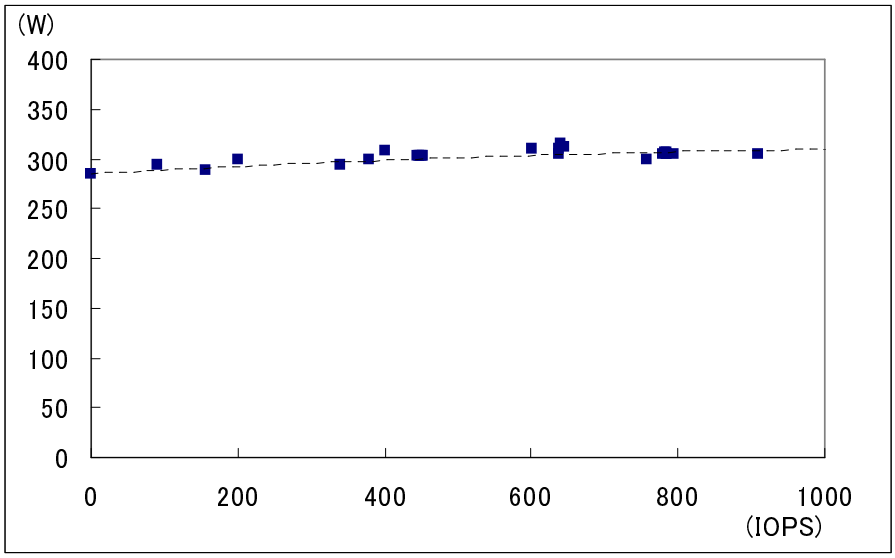

図 4.9: ディスク筐体の消費電力特性

図 4.8 から分かるように，コントローラの消費電力は IOPS によらず約 321W でほぼ一定であった．また，図 4.9 から分かるように，Idle 時のディスク筐体の消費電力は約 285W

であった．ディスク筐体の消費電力は IOPS が増加するにつれ増加している．最大電力消費量は 315W であり，これはディスク筐体の Idle 時の消費電力より 10.6%高い．

省電力機能使用時のストレージ筐体の消費電力

図 4.10 にディスク筐体の省電力機能を用いた場合のストレージコントローラ及びディスク筐体の消費電力を示す．ストレージの構成は図 4.7 に示した通りである．図中，Spindown とは全てのディスク筐体を Spindown 状態に移行した場合のストレージの消費電力を，Power off は全てのディスク筐体の電源を OFF にした場合のストレージの消費電力をそれぞれ示している．

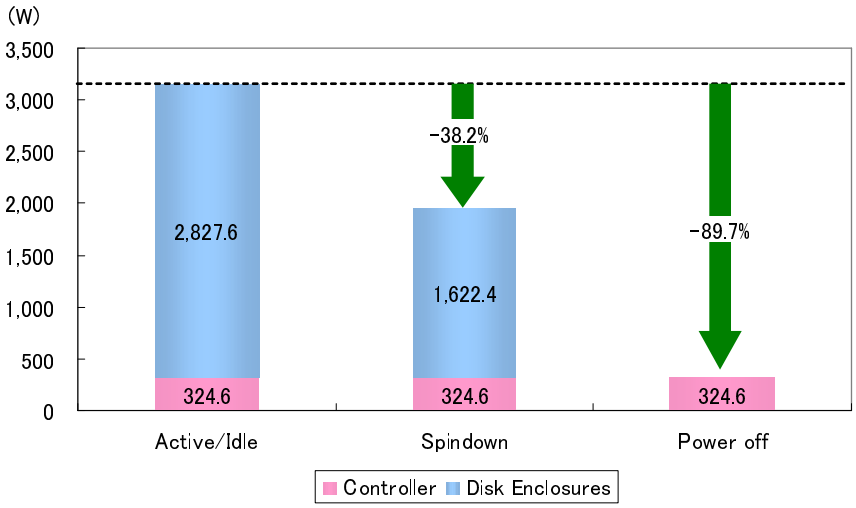

図 4.10: 省電力機能使用時のストレージ消費電力特性

図から分かるように，ディスク筐体を Spindown 状態に移行した場合，ディスク筐体の消費電力は 2,827.6W から 1,622,4W に減少した．コントローラの消費電力は変化していない．ストレージの消費電力は約 38.2%減少した．ディスク筐体の電源を OFF にした場合，ディスク筐体の消費電力は 0W であった．コントローラの消費電力は Spindown 時と同様変化していない．

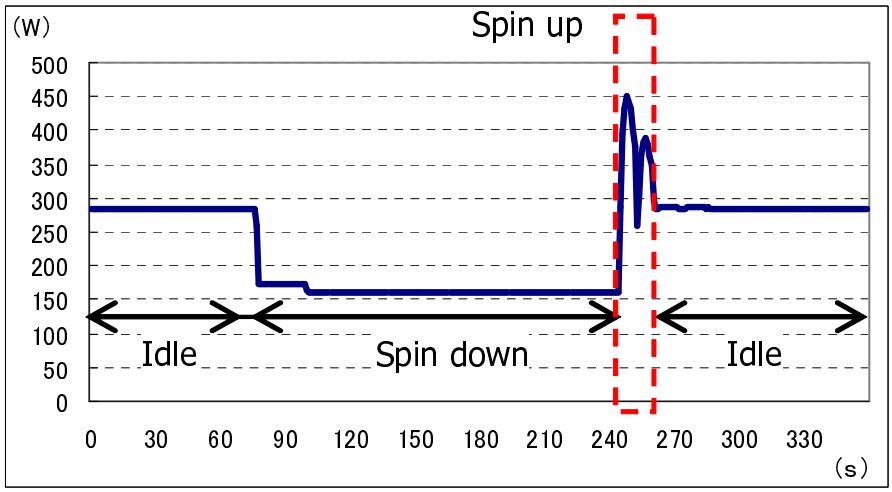

図 4.11: ディスク筐体を Spin down 状態に移行した場合の消費電力推移

図 4.11 は，ディスク筐体を Spin down 状態に移行した後再度 Idle 状態に移行 (Spin up) した場合の消費電力の推移を示している．

図から分かるように，ディスク筐体の Spin up 時の最大消費電力は最大約 450W に達していた．これはディスク筐体内の HDD のスピンドルモーターの起動のためである．ディスク筐体の Spin up に要するエネルギーは約 5,860J，Spin up に要した時間は約 16 秒であった．

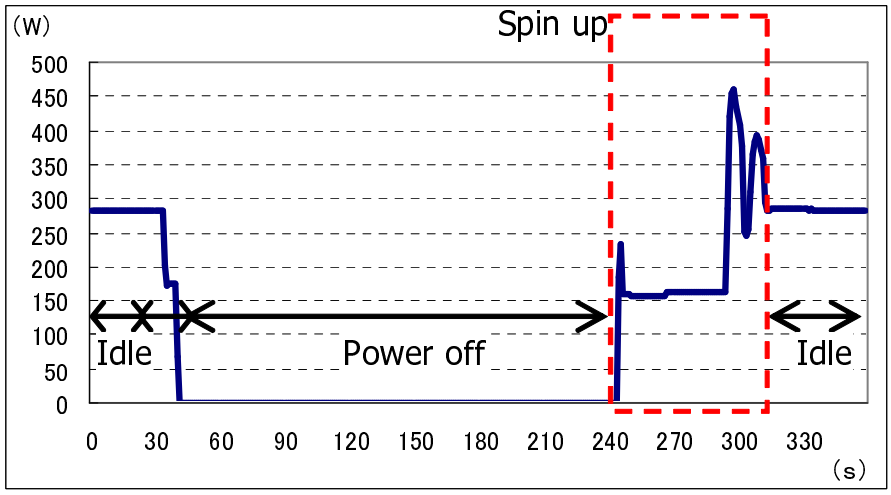

図 4.12: ディスク筐体を電源 OFF 状態に移行した場合の消費電力推移

図 4.12 は，ディスク筐体の電源を OFF にした後再度 Idle 状態に移行 (Spin up) した場合の消費電力の推移を示している．図から分かるように，ディスク筐体の Spin up 時の最大消費電力は Spin down 時の場合とほぼ同じ最大約 460W であった．一方，ディスク筐体の Spin up に要するエネルギーは約 14,717J，Spin up に要した時間は約 69 秒と，Spin down 状態からの Spin up と比較して大幅に増加した．

これらの結果より，ストレージの消費電力を削減するにはディスク筐体に対する秒当た

り入出力数を削減するのみでは大きな効果を得ることが難しく，Spindown や電源 OFF などのディスク筐体単位の省電力機能を最大限活用することが必要であることが分かる．

4.2.3　複数台の HDD とディスク筐体の消費電力特性の違い

ディスク筐体と，ディスク筐体内に格納されている HDD 数と同数の HDD の電力特性を明確化するために，15 台の HDD とディスク筐体の消費電力及び起動に要する時間を比較した．HDD 15 台の消費電力は，4.2 節で計測した値を 15 倍することにより求める．ディスク筐体の消費電力は，4.2.2 節で計測した値である．

Idle 時及び Active 時の消費電力

図 4.13 は，Idle 状態の 15 台の HDD と，同じく Idle 状態のディスク筐体の消費電力を比較したものである．図から分かるように，Idle 状態の 15 台の HDD の消費電力の合計値は 132.4W，Idle 時のディスク筐体の消費電力は 285.0W であった．ディスク筐体の消費電力の方が約 150W 高い結果となった．

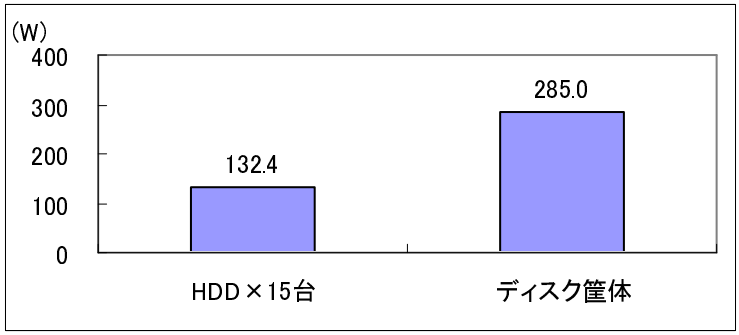

図 4.13: HDD 15 台とディスク筐体の消費電力比較 (Idle 時)

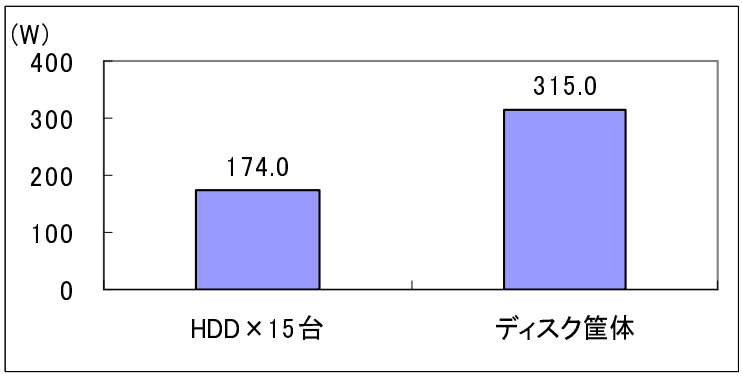

図 4.14: HDD 15 台とディスク筐体の消費電力比較 (Active 時)

図 4.14 は，Active 状態の 15 台の HDD と，同じく Active 状態のディスク筐体の消費電力を比較したものである．ディスク筐体への秒当り入出力数は，消費電力が最も高かった時の値 (641.1 IOPS, read 100%) を用いた．HDD への入出力数は，この値の 1/15 である 42.7 IOPS の場合の値を用いた．図から分かるように，Idle 状態の 15 台の HDD の消費電力の合計値は 174.0W，Idle 時のディスク筐体の消費電力は 315.2W であった．ディスク筐体の消費電力の方が約 140W 高い結果となった．

これらの比較結果から，ディスク筐体の消費電力は，単に HDD を 15 台並べた場合よりも約 150W 高いことが分かる．これは，ディスク筐体が内蔵している電源等が消費する電力である．

省電力機能使用時の消費電力と起動時間

次に，HDD 及びディスク筐体の省電力機能を用いた場合の消費電力と起動時間を比較する．図 4.15 は，15 台の HDD を Standby 状態から Idle 状態に移行するために必要なエネルギー，Spin down 状態のディスク筐体を Idle 状態に移行するために必要なエネルギー，及び電源 OFF 状態のディスク筐体を Idle 状態に移行するために必要なエネルギーをそれぞれ比較した結果である．

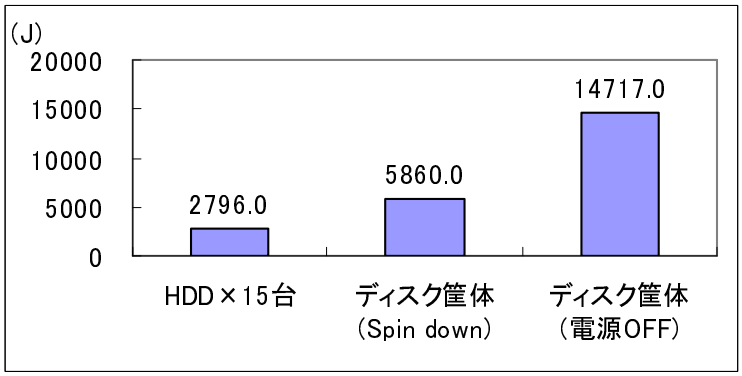

図 4.15: HDD 15 台とディスク筐体の Spin up 時のエネルギー比較

図から分かるように，電源 OFF 状態のディスク筐体を Spin up するために必要となるエネルギーが最も高いことが分かる．これは，ディスク筐体内の電源等の起動に必要なエネルギーを含むためである．また，15 台の HDD を Spin up する場合のエネルギーが最も低い．Spin down 状態のディスク筐体を Spin up するために必要となるエネルギーは，HDD15 台の場合の約 2 倍であった．これは，ディスク筐体を Spin up する際の最大消費電力を低く抑えるため，ストレージがディスク筐体内の HDD を半数ずつ起動しているためである．

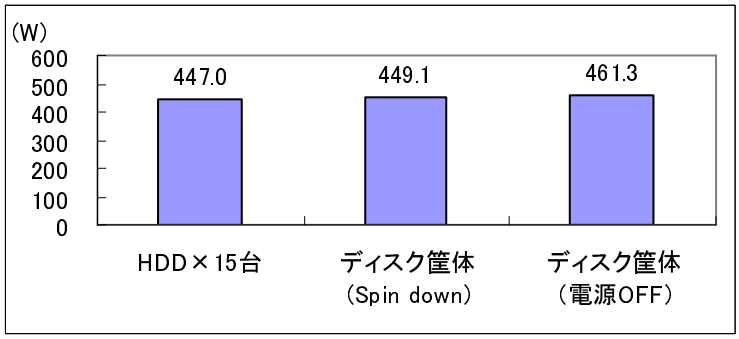

図 4.16: HDD 15 台とディスク筐体の Spin up 時の最大消費電力比較

図 4.16 は，15 台の HDD を Standby 状態から Spinup した場合，Spin down 状態のディスク筐体を Spin up した場合，及び電源 OFF 状態のディスク筐体を Spin up した場合の最大消費電力を比較した結果である．

図に示したように，各場合とも最大消費電力は約 450W でありほとんど差がないことが分かる．この結果からも，ストレージがディスク筐体の最大消費電力を抑えるように HDD の起動を制御していることが分かる．

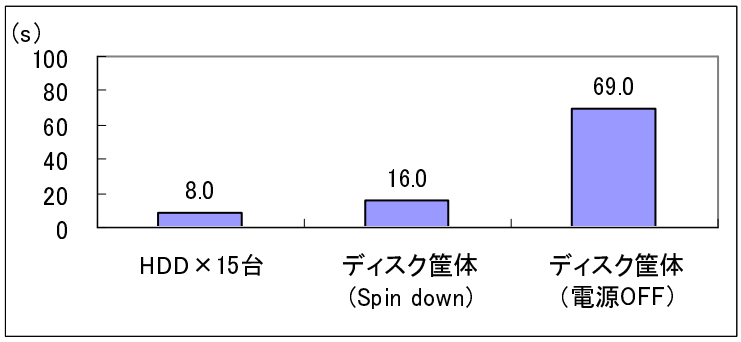

図 4.17: HDD 15 台とディスク筐体の Spin up 所要時間比較

図 4.17 は，15 台の HDD を Standby 状態から Spinup した場合，Spin down 状態のディスク筐体を Spin up した場合，及び電源 OFF 状態のディスク筐体を Spin up した場合の所要時間を比較した結果である．

HDD 15 台を同時に Spin up するのに必要であった時間は 8 秒，Spin down 状態のディスク筐体を Spin up するのに必要であった時間は 16 秒，電源 OFF 状態のディスク筐体を Spin up するのに必要であった時間は 69 秒であった．これらの結果からも，ストレージはディスク筐体中の HDD を，時間をかけて Spin up していることが分かる．電源 OFF 状態のディスク筐体の Spin up が Spin up 状態のディスク筐体の Spin up より 50 秒以上長いのは，ディスク筐体内の電源等の起動のためである．

ディスク筐体は，同数の HDD を束ねただけの JBOD と比較して Spin up 時の消費電力は高く起動に要する時間も長いことが分かる．つまり，ディスク筐体の省電力機能の使用は，JBOD の省電力機能の使用と比較して困難であると言える．

4.3 Break Even Time

Break Even Time とはストレージ省電力の指標である．本章では，まずディスク筐体の場合を例に，Break Even Time について説明する．次に，HDD 及びディスク筐体の Break Even Time の長さを示す.

4.3.1 Break Even Time の定義

ディスク筐体の電源が一旦 OFF になると，その起動にはいくらかの電力が必要になる．一方，入出力を待っている間，ディスク筐体の Idle 状態を維持するためにも電力が必要である．これは，電源 OFF と Idle 状態の間にはトレードオフが存在することを示している．もし，次の入出力が来るまでに十分な時間あれば，ディスク筐体の電源を OFF にした方が Idle 状態を続けるより消費電力を削減できる．しかし，前節で述べたように，ディスク筐体の Spin up には約 15KJ のエネルギーが必要である．ディスク筐体の電源を OFF にし，エネルギー削減量 (Idle 状態を継続した場合と比較した場合) が 15KJ に達する前に再度ディスク筐体を Spin up した場合，ディスク筐体が消費するエネルギーは逆に増加する．

ディスク筐体の Idle 状態を維持した場合に消費する電力と，ディスク筐体の電源 ON に必要な電力が等しくなる，Idle 状態の持続時間を Break Even Time と呼ぶ．電源 OFF 機能を用いてディスク筐体の消費電力を削減するためには，ディスク筐体の電源 OFF の持続時間が Break Even Time より長くなければならない．このことは，電源 OFF 機能を用い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いくつかの入出力間隔は，Break Even Time より長い必要があることを示している．

Break Even Time の長さを L_b とすると，L_b は式 4.1 により計算することができる．

$$L_b = (E_{spd} + E_{sup})/(P_{idle} - P_{save}) \tag{4.1}$$

ここで，E_{spd} はディスク筐体を省電力状態に移行するために必要なエネルギー，E_{sup} はディスク筐体を Spin up し Active/Idle 状態に移行するために必要なエネルギー，P_{idle} はディスク筐体の Idle 時の消費電力，P_{save} はディスク筐体が省電力状態の場合の消費電力である．

なお，本節ではディスク筐体を例に Break Even Time について説明したが，HDD の場合も同様に計算することができる．

4.3.2 Break Even Time の値

図 4.18 に，HDD 一台，Spin down 状態に移行する場合のディスク筐体の消費電力，及び電源 OFF 状態に移行する場合のディスク筐体の Break Even Time の長さの計算値を示す．HDD 及びディスク筐体は，4.2 節で用いたものである．Break Even Time の長さの計算には，式 4.1 を用いた．

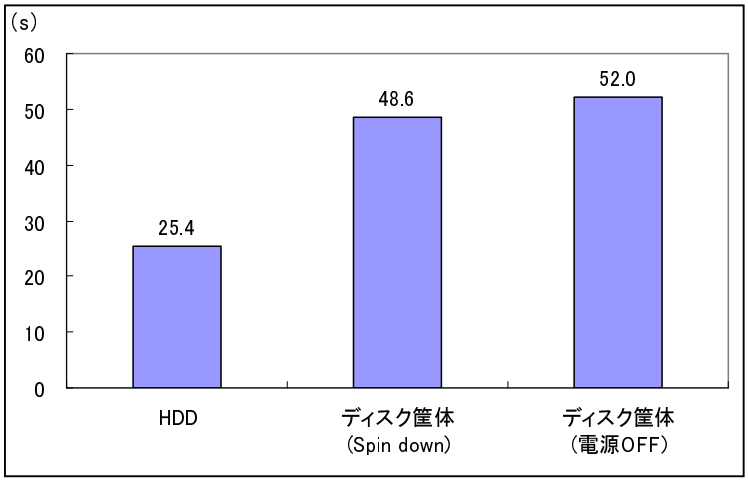

図 4.18: Break Even Time の長さ

HDD の Break Even Time は 25.4 秒，Spin down 状態に移行する場合のディスク筐体の Break Even Time は 48.6 秒，電源 OFF 状態に移行する場合のディスク筐体の Break Even Time の長さは 52.0 秒であった．Spin down 状態に移行する場合と電源 OFF 状態に移行する場合のディスク筐体の Break Even Time の長さはどちらも約 50 秒であった．HDD の Break Even Time の長さはディスク筐体の Break Even Time の長さの約 1/2 であった．

第5章　オンライントランザクション処理の入出力挙動特性を利用したハードディスクの実行時省電力技法

5.1　はじめに

ストレージを構成するコンポーネントの中で，HDD は消費電力が最も高い．そこで，まず代表的なデータインテンシブアプリケーションである OLTP 実行中の HDD 省電力手法を検討する．

現在出荷されている HDD のほとんどは，ヘッドの退避やプラッタの回転停止等，消費電力を削減する機能を搭載している．しかし，CPU の省電力機能である Dynamic Voltage and Frequency Scaling と異なり，省電力状態中の HDD は入出力処理ができない．また HDD を省電力状態から入出力処理可能な状態に復帰させるために数秒以上のオーバーヘッドを要する．アプリケーション実行中に，その処理性能を低下させずに HDD の省電力機能を適用することは，CPU の省電力機能の適用と比較して非常に困難である．

大量のデータを処理する代表的なアプリケーションとしてデータベース管理システム (DBMS) がある．HDD の DBMS 向け出荷容量は HDD の全出荷容量の 6 割以上を占め，さらにその半数以上がバンキングや証券取引，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) に使用されている [82]．OLTP 処理中の HDD の消費電力は，OLTP に使用される IT 機器の全消費電力の 60%以上との報告もあり [80]，OLTP における HDD の消費電力の削減は，IT 機器全体の消費電力を削減する上で重要である．

多量の HDD を利用する OLTP は，高スループットで動作する場合は HDD に対して毎秒数十回ものランダム入出力を発行する．従来のストレージデバイス (HDD) レベルの入出力挙動のみを用いた省電力手法は，入出力発行間隔が数十秒から数分と長い場合や，アプリケーションコードの解析により実行前に入出力アクセス挙動が予測できる場合，あるいは，現在商用化されていない回転数を変更可能な HDD を用いなければ，高い省電力効果を得ることはできない．高スループットで動作する OLTP は上記のケースにはあてはまらないため，ストレージデバイスレベルの入出力挙動のみを用いる従来手法での省電力は困難である．

本章では，まず HDD 上で OLTP を稼働し，その入出力トレースを取得する．取得した入出力トレースを解析し OLTP の入出力挙動特性を明らかにする．そして OLTP においてもディスクの省電力機能を利用する機会がある可能性を示す．次に，実行時ストレージ省電力フレームワークに基づく省電力手法を提案する．提案手法の特長は，アプリケーションレベルのデータ (表・索引等) で HDD の省電力を考える，つまり表・索引毎の入出力発

行間隔をモニタリングし，省電力可能なアクセスパターンを持つ表・索引を抽出し，そのHDD 上の配置を決定する．さらに，表・索引に対する入出力の挙動が write が支配的であるものに着目し，DB への write がログ先行書き出しプロトコル (WAL) に従えばよいことを利用した write 遅延を試みる．実機および実入出力トレースを用いて提案手法の評価を行い，提案手法が既存のブロックレベルの省電力手法 [69]，およびファイル単位の入出力数のみを用いた省電力手法 [75] と比較して高い省電力効果が得られることを示す.

5.2 HDD 上で稼動する OLTP の入出力挙動特性と省電力の機会

OLTP の入出力挙動特性を利用した HDD の省電力手法を検討するため，OLTP の代表的ベンチマークである TPC-C ベンチマーク [7] を用い，OLTP の入出力 挙動特性を詳細に解析した.

5.2.1 計測環境

OLTP の入出力挙動特性の計測に当り，図 4.3 に示した環境に，同一種類の SATA HDD クを 1 台追加し，HDD 2 台構成とした．OLTP の入出力 挙動計測に使用したソフトウェアおよびその設定を表 5.1 に示す.

表 5.1: HDD 上で動作する OLTP の入出力挙動特性解析ソフトウェアおよびその設定

OS	Cent OS 5.3 (32bit)
File System	ext2
DMBS	MySQL Comunnity Server 5.1.40
OLTP Program	tpcc-mysql
DB Size	4GB (Warehouse=10)
DBMS Buffer Size	2GB
# of Threads	5
Think Time & Keying Time	0 s

SQL 発行環境の構築には tpcc-mysql[89] を用いた．また，当該システム構成で最も高負荷，すなわちトランザクションスループットが最も高くなる DB サイズとして，Warehouse 数が 10(DB サイズ 4GB) の場合の入出力 挙動を計測した．ファイルシステムのキャッシュ及び HDD のキャッシュは無効化している.

Warehouse 数が 10 より小さい場合，行レベルの競合が発生しスループットが低下する．Warehouse 数が 10 より大きい場合は DB バッファ内の更新されたページの数が増えるため，チェックポイント時の更新されたページの出力と当該ページへのアクセスの競合が増加し入出力数は減少する．Warehouse 数が 10 より小さい場合，及び Warehouse 数が 10 より大きい場合のいずれの場合も入出力 数が減少するため，省電力は容易になると考えられる．従って，最も高スループットが得られる Warehouse 数 10 の場合で入出力挙動を解析した.

本計測では，TPC-C を用いて DB のデータである表・索引に対する入出力のトレースを取得した．トランザクションスループットが安定してから 100 万入出力 を超える程度のトレース (約 3 時間分) を blktrace (入出力トレース取得ツール)[14] を用いて取得し，ブロックと DB の表・索引との対応関係を用いて表・索引単位の入出力とレースを生成した．

5.2.2 TPC-C の入出力挙動特性と省電力の可能性

DB データ毎の Read/Write 別の秒当りアクセス回数を図 5.1 に示す．

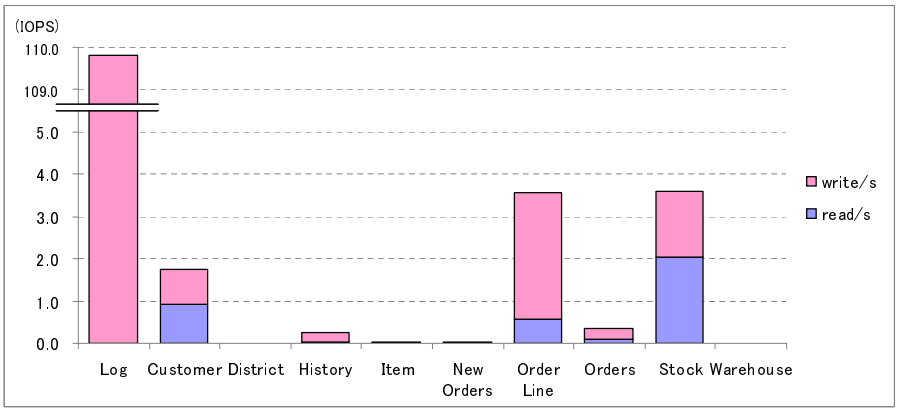

図 5.1: TPC-C の平均入出力数

図より，ログに対しては毎秒 100 回以上 write 入出力が発行されており，省電力の余地はほどんどないことが分かる．しかしログ以外の表・索引に対する入出力は最大でも毎秒 3.5 回程度である．また，District や History，NewOrders，Warehouse などは入出力が少なく，発行された入出力 の 90%以上が write 入出力であった．District や NewOrder, Warehouse 表は，Stock や Customer 表等と比較してサイズは小さい．このため District や NewOrder, Warehouse 表を格納したページは，Stock 表などの大きな表と比較し，DB バッファ上にキャッシュされたデータの読み込みが多い．つまり，Stock 表などに比べ read の入出力 数は少なくなり，相対的に write の入出力 比率が高くなっていると考えられる．また，History 表はデータの追加が主体の表であるため，write 入出力の比率が高くなっていると考えられる．

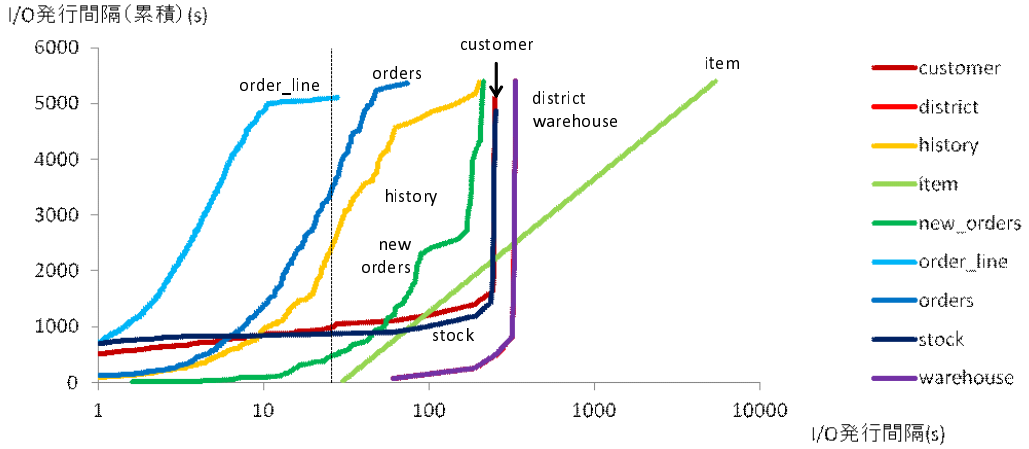

図 5.2: TPC-C の入出力発行間隔 (0 秒から 5400 秒まで)

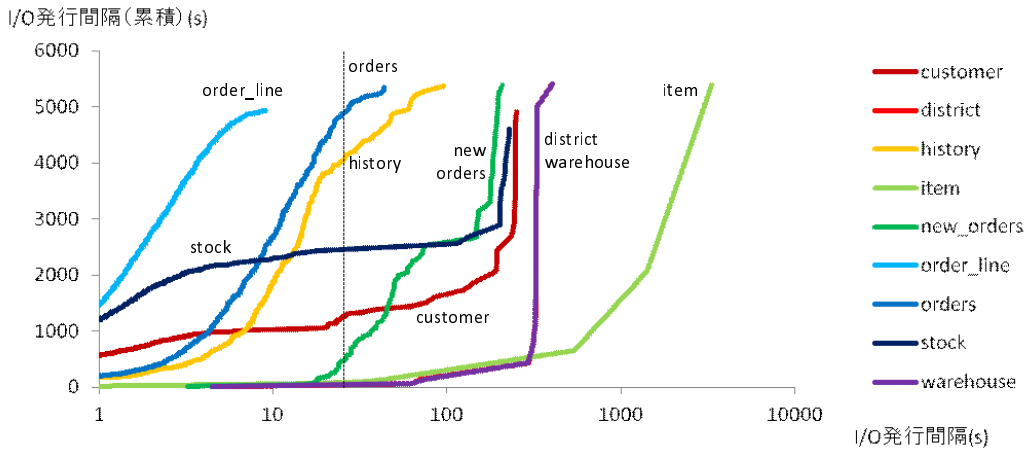

図 5.3: TPC-C の入出力発行間隔 (5400 秒から 10800 秒まで)

図 5.2 は計測開始から 1.5 時間分の，図 5.3 は 1.5 時間後から 3 時間後までの表・索引への入出力 の発行間隔の長さの分布を示している．横軸は入出力 発行間隔の長さ (ログスケール)，縦軸は入出力発行間隔の累積値である．

図中の縦の一点鎖線は Break Even Time を示している．図 5.2 を見ると OrderLine を除く各表の入出力発行間隔の長さには Break Even Time より長いものが多数存在し，特に Warehouse，District(Warehouse とほぼ重なっている)，Item，NewOrders などの入出力発行間隔が数百秒以上と長いものがあることが分かる．これは，高スループットで実行中の OLTP 系アプリケーションであっても HDD の省電力の可能性があることを示している．また，図 5.2 と 5.3 を比較すると，表・索引毎の入出力発行間隔の分布は時間経過に対してほとんど変化していないことが分かる．

これらの計測結果より，DB の表・索引に対する入出力の発行間隔は Break Even Time より長いものが多数あること，入出力発行間隔が長い表・索引に対する入出力はそのほとんどが write であること，及び表・索引単位では入出力挙動の変化はほとんどないことが明

らかとなった．これらの特徴は，HDD やブロック単位の入出力挙動特性解析では把握することはできない．

ここまで計測に用いた環境は，表 5.1 に示すように，DB の Warehouse 数は 10(DB サイズは 4GB)，DB バッファサイズは 2GB である．一方，TPC-C ベンチマークのレポート等では，DB バッファサイズは DB サイズの 5%程度であり計測を行った環境より小さい．そこで DB バッファサイズが DB サイズの 5%となる Warehouse 数 100(DB サイズ 40GB) の DB を用いた場合についても入出力発行間隔の調査を行った．この結果を図 5.4 に示す．

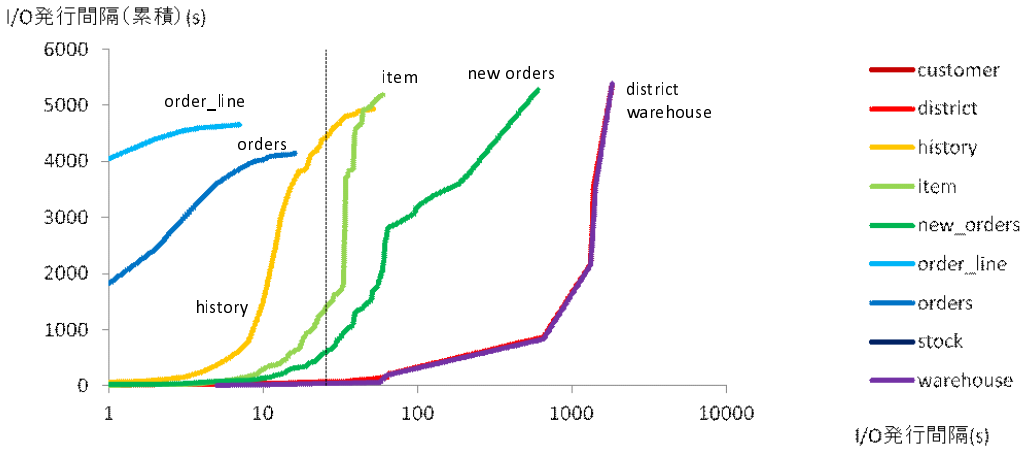

図 5.4: TPC-C の入出力発行間隔 (DB サイズ 40GB)

図より，入出力発行間隔が短く，省電力が期待できない OrderLine や Order 表では Break Even Time より長い入出力発行間隔は見られなくなった．しかし，Warehouse や District 表など入出力 発行間隔が長く省電力が期待できる表では，Break Even Time 以上の入出力 発行間隔の長さは DB バッファサイズが DB サイズの 50%の場合と比較し，同等かそれ以上であることが分かる．省電力が期待できる表の入出力発行間隔が長い方が HDD の省電力には有利である．そこで，より HDD の省電力が厳しい環境で提案手法の有効性を評価するため，DB の Warehouse 数を 10(DB サイズ 4GB)，DB バッファサイズを 2GB とした．

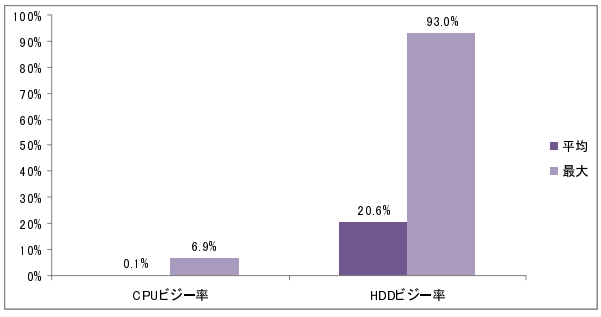

図 5.5: Warehouse 数 10 の場合の CPU 及び HDD のビジー率

次に，今回の TPC-C 実行時 (Warehouse 数:10,DB サイズ:4GB) における計算機資源のビジー率を調査した．この結果を図 5.5 に示す．CPU ビジー率は平均 0.1%，最大 6.9DB デー

タを格納した HDD のビジー率は平均 20.6%，最大 93.0%であった．本計測環境では CPU ビジー率と比較してディスクのビジー率の方が高く，アプリケーションスループットは入出力性能が支配的である．従って，アプリケーション実行時のディスク省電力にとって，本計測環境は厳しい環境と考える．

5.3 OLTP の入出力挙動特性を用いた HDD の省電力手法

本節では，第 4 章で示した TPC-C の入出力挙動特性を基に，アプリケーション実行中にも省電力可能な HDD の省電力手法を提案する．前節の結果から TPC-C においても表・索引に対する入出力発行間隔は Break Even Time より長いものが多数あり，また表・索引単位では入出力挙動の経時変化はほとんどない．これらの知見から，入出力 発行間隔が短い表・索引を同一の HDD に配置することにより他の HDD の入出力発行間隔を伸ばす可能性があると考えられる．さらに，入出力発行間隔が長い表・索引に対する入出力のほとんどが write 入出力である．この特性と DBMS の更新ログ先行書き出しプロトコルの性質を利用し，HDD に通常適用される write 入出力の実際の書き込みを伸ばすことができる可能性がある．以下，OLTP が稼働する HDD を対象とした実行時ストレージ省電力フレームワーク，及びアプリケーションの入出力挙動特性を用いた実行時の HDD 省電力手法である，データ配置制御および write 遅延について説明する．

5.3.1 OLTP が稼働する HDD の実行時ストレージ省電力フレームワーク

図 5.6 に，OLTP が稼働する HDD を対象とした実行時ストレージ省電力フレームワークを示す．アプリケーションは TPC-C，DBMS は MySQL，ストレージデバイスは HDD である．HDD 上で稼働する TPC-C では，DBMS が持つ DB バッファが HDD に対する入出力間隔の延伸に重要な役割を果たす．このため，バッファを利用した省電力手法を DB バッファ層に組み込む．

実行時ストレージ省電力フレームワークは，DB の表・索引毎の入出力，及び HDD の入出力を監視し，TPC-C の表・索引単位で HDD に対する入出力を把握する．そして TPC-C の表・索引毎の入出力パターンを抽出し，抽出した入出力パターンに基づき HDD への表・索引の配置，及び write 遅延を適用する表・索引を決定する．また，TPC-C 実行中の入出力挙動の変化に伴い入出力パターンが変化した場合には，データ配置や write 遅延対象の表・索引を再度選択する．さらに，本フレームワークは HDD に対する入出力を監視し，HDD に対して入出力が行われていない場合には HDD の電源を OFF にすることにより，HDD の消費電力を削減する．

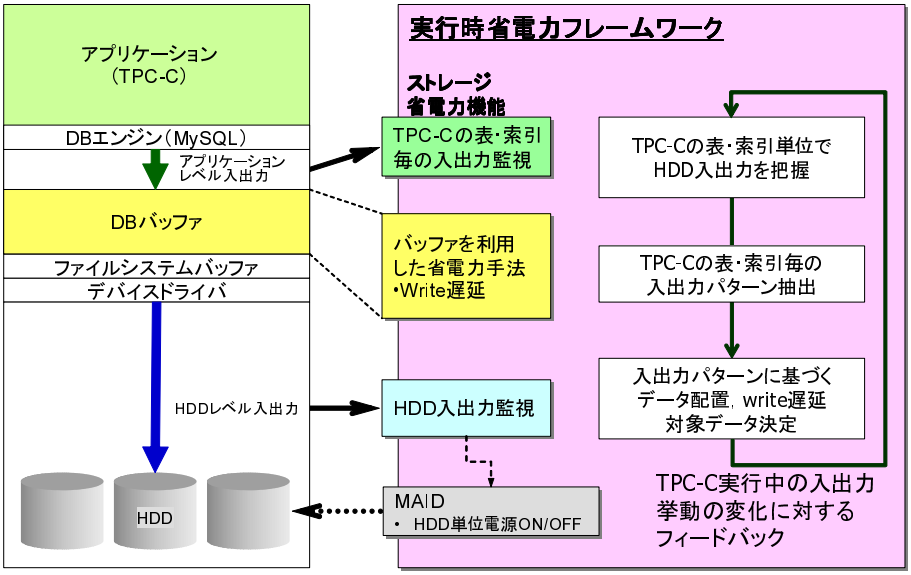

図 5.6: OLTP が稼働する HDD を対象とした実行時ストレージ省電力フレームワーク

5.3.2 データ配置制御

ほとんどの入出力発行間隔が Break Even Time より短い，つまり入出力頻度の高い表および索引をなるべく少数の HDD に集め，残りの HDD を省電力の対象とする．従来の手法では，物理ブロックアドレスやファイル等デバイスレベル，ファイルシステムレベルのアクセス頻度に着目していたが，本手法では OLTP が認識するデータであるログ，及び表・索引を対象とする．ログおよび表・索引は物理ブロックと比較して大きな単位だが挙動が安定しており，前節の分析結果 (図 5.2，5.3) から表・索引単位での入出力発行間隔分布に基づくデータ配置でも高い省電力効果を得られる可能性があると考えられる．

実行時のデータ配置制御の概要をアルゴリズム 1 に示す．データの初期配置に関しては，IOPS が複数 HDD 間でできるだけ均等になるように配置されていると仮定する．また，入出力挙動モニタリングは，通常の DBMS のモニタと同じ頻度で行う．

Hot・Cold データ計算 モニタリングの結果を用いて OLTP のデータを Hot データと Cold データに分類する．Cold データとは同じ HDD に格納されている他のデータへのアクセスは考えず，当該データのみのアクセス履歴から計算して HDD の消費電力を削減できるデータである．Cold データ以外のデータを Hot データとする．アルゴリズム 1 を開始，あるいは前回データ再配置を実行してから現在までの時間間隔を T，データ j のみを HDD に配置したと仮定した場合に，期間 T 内で HDD に対して発行される入出力の発行間隔のうち Spindown タイムアウトより長い入出力発行間隔を l_i とする．期間 T 内に k 回 l_i が観測された場合の HDD の消費電力削減量は $\sum_{i=1}^{k}((\text{Idle 時電力-Standby 時電力}) \times l_i - \text{HDD 起動エネルギー})$ である．

Hot データを配置する HDD 計算 Hot データを格納する HDD 数 N を求める．I_{max} を HDD が提供できる最大 IOPS，S_{max} を HDD の容量とすると，$N = max(\lceil Hot \text{ データの}$

Algorithm 1 HDD の実行時省電力制御アルゴリズム

```
物理ブロックと TPC-C の論理的データ単位の入出力挙動の監視を開始;
HDD の省電力状態と HDD の消費電力の監視を開始;
while DBMS 実行中 do
  Hot・Cold データ計算 ();
  if Hot データが Cold データとなる，又は Cold データが Hot データとなる then
    Hot データを配置する HDD 計算 ();
    移動対象データおよびデータの移動先の計算 ();
    if 新配置の電力削減量 > 現在の配置の電力削減量 then
      データ再配置;
      if ColdHDD 数 > 0 then
        Write 遅延開始;
      else
        Write 遅延停止;
      end if
    end if
    if Cold HDD 数 >0 かつ省電力機能が適用されていない Cold HDD がある then
      省電力機能が適用されていない Cold HDD に省電力機能使用のためのパラメタを
      設定;
    end if
  end if
end while
```

合計 $IOPS$ の最大値/$I_{max}\rceil, \lceil Hot$ データの合計サイズ/$S_{max}\rceil$) である．$IOPS$ とは秒当たり入出力の数のことである．次に，HDD 内の合計 Hot データ量の降順にソートし，上位 N HDD を Hot データを配置する HDD とする (以降これらの HDD を Hot HDD，残りの HDD を Cold HDD と呼ぶ)．Hot データの容量でソートするのはデータの移動量を削減するためである．

移動対象データおよびデータの移動先の決定 Cold HDD 中の Hot データを移動対象データとする．まず，Hot HDD のうち，空き容量が，移動しようとする Hot データのサイズ以下の HDD を選択する．次に，それらの HDD の中から，移動対象の Hot データを配置した後の最大 IOPS が I_{max} 以下かつ Hot データ移動後の最大 IOPS が最小となる Hot HDD を移動先として選択する．Hot HDD に十分な空きがなく Hot データをいずれの Hot HDD にも移動できない場合は，Hot HDD 上の Cold データを Cold HDD に移動する．その際，移動する Hot データ以上の容量を持つ Cold データを選択する．次に，当該 Cold データを移動した場合に入出力発行間隔が最も長くなる Cold HDD を選択し当該 Cold データを移動する．容量および性能の要件を満たす Hot HDD がなく，かつ Cold データの移動もできない場合，Hot HDD の数を 1 増やして再度本ステップを実行する．

HDD の電力削減量 アルゴリズム 1 を開始，あるいは前回データ再配置を実行してから

現在までの時間長を T，期間 T 内で HDD に対し発行される入出力の発行間隔のうちの Spindown タイムアウトより長い入出力発行間隔を l とする．期間 T 内に k 回 l が観測された場合の HDD の電力削減量は，$\sum_{i=1}^{k}((Idle$ 時電力 $- Standby$ 時電力 $) \times l_i - HDD$ 起動エネルギー) である．現在のデータ配置，および新たなデータ配置のそれぞれについて上記を計算する．

5.3.3 Write 遅延

WAL プロトコルは，DB バッファ上の DB データを HDD に書き出す前に，当該データの更新をログに書き出すことを保証する．これによりトランザクションのコミットとは独立した契機で DB バッファ上のデータを HDD に書き出すことが可能となる．write 遅延はこのプロトコルを利用して write をまとめて HDD に書き出すことにより Cold HDD への入出力発行間隔を伸ばす手法である．

DBMS が DB に対して write 入出力を行う契機は，i) チェックポイントと呼ばれる更新された DBMS バッファページの HDD への書込み，ii) チェックポイントによる HDD への負荷を低減するための (チェックポイントに先立つ) データの write，及び iii)DB バッファに空きがない場合に空きを作るために更新されたデータを DB バッファから追い出す場合, の 3 通りである．

Write 遅延はこのうち ii) のチェックポイントに先立つデータの write を遅延する．Cold HDD 上のデータは入出力数が少ないため ii) を利用するメリットがなく，また Cold HDD に対する更新を一括で write しても HDD の負荷は低いままと考えられるためである．図 5.7 にその概要を示す.

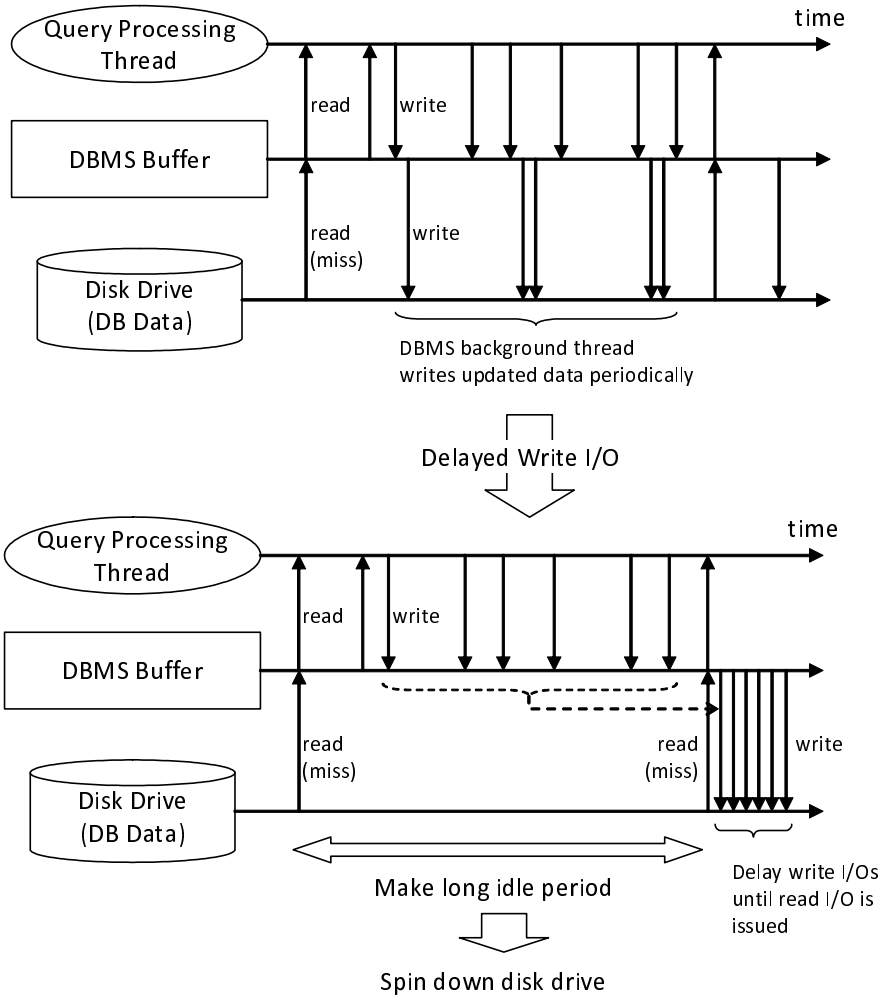

図 5.7: Write 遅延方式

Write 遅延は，アルゴリズム 1 で Write 遅延対象データと判定されたデータを対象に行う．DB バッファ上のこれらのデータが更新された場合，これらのデータをチェックポイント開始時，あるいは DB バッファの更新ページ数の比率が α を超えるまで保持する．そしてチェックポイント開始あるいは更新ページ数の比率が α を超えた時点で，write 遅延対象データの更新を更新順序を変更せずに HDD に反映する．Write 遅延処理は DBMS の WAL プロトコルと整合しており，write 遅延により DBMS の信頼性が低下することはない．

図 5.8 に OrderLine 表および索引に write 遅延を適用した場合と適用しない場合の入出力トレースを示す．図 5.8 より，write 遅延により数百秒の入出力発行間隔が生成できていることが分かる．

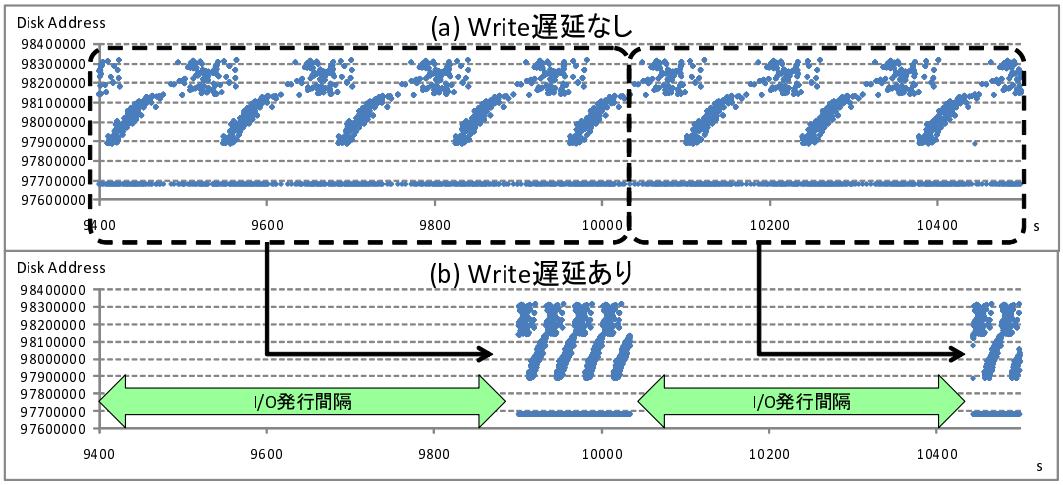

図 5.8: Write 遅延による入出力発行間隔の延伸

5.4 評価

本章では，提案手法の評価方法および結果について述べる．提案手法の効果を確認するために，HDD 2 台および 5 台を用いた構成で評価を行った．また，既存の実行時 HDD 省電力手法として，HDD のブロックの配置制御を行う手法 (Dynamic Data Reorganization; DDR) [69]，及びファイルの配置制御を行う手法 (Popular Data Concenteration; PDC)[75] との比較を行う．

5.4.1 評価方法及びパラメタ

提案手法，PDC，DDR のそれぞれについて，HDD の消費電力，およびトランザクションスループット，データの移動量を用いて比較する．5.2 節の TPC-C 実行時の DB データに対する入出力トレース (基本入出力トレース) を基に各手法を適用した後の入出力トレースを生成し，それらを HDD 上で再生することにより HDD の消費電力を計測した．

基本入出力トレース

HDD 2 台の場合は 5.2 節で取得した入出力トレースを基本入出力トレースとした．5 台の場合は，5.2 節で述べた構成にさらに同一の SATA HDD を 3 台追加し基本入出力トレースを取得した．提案手法及び PDC は表・索引単位の入出力とレースを用いた．DDR の場合は blktrace ツールで取得した物理ブロックの入出力トレースを使用した．HDD 5 台構成は，HDD 2 台の場合と同一規模の DB を用い，その構成で最も高いスループットを達成するためにスレッド数を増やした．TPC-C の warehouse 数，Think Time，DB バッファサイズは HDD 2 台の場合と同じとしている．

表 5.2: データの配置 (HDD 5 台)

Disk #	Data on Disk
Disk #1	Log
Disk #2	Stock
Disk #3	OrderLine
Disk #4	Customer
Disk #5	District, History, Item, NewOrders, Orders, Warehouse

各手法の入出力トレース生成

提案手法の入出力トレースは，アルゴリズム 1 を基本入出力トレースの先頭から適用することにより生成した．つまり，データ配置制御に合わせ，それ以降データの移動先の HDD の同じ物理アドレスにアクセスするように変更した．Write 遅延を適用したデータの write はチェックポイント時又は DB バッファの更新ページ数の比率が α を超えるまで遅延させている．また，データの移動に必要な新たな入出力も付加している．DDR ではブロック交換後の入出力先の変更，及びブロック交換に必要な新たな入出力を基本入出力トレースから生成した．PDC では，ファイル移動後の入出力先の変更，及びファイル移動に必要な新たな入出力を基本入出力トレースから生成した．提案手法と PDC では，データ移動のための IOPS が $I_{max}/2$ を超えないよう IOPS を制御した．DDR は Cold HDD 上のブロックにアクセスが行われる毎にブロックを交換するため，それに従ってデータ移動のための入出力を付加した．HDD の Spindown タイムアウトは OS で設定可能な最小値である 5 秒，Write 遅延における DB バッファの更新ページ数の比率 α は 50%とした．

データ配置の計算

(a) データ初期配置 HDD 2 台の場合は，各手法ともログを Disk#1 に，表・索引を Disk#2 に配置した．HDD 5 台の場合は，ログを Disk#1 に，表・索引を 4 章で取得した表・索引毎の IOPS に基づき 5 章で述べた初期配置に従い Disk#2-4 に配置した．実際の配置は表 5.2 に示すとおりである．

(b) データ再配置 提案手法はアルゴリズム 1 を用いてデータの再配置を行った．DDR，PDC はそれぞれ文献 [69, 75] に示された方式に基づき再配置を行った．

DDR における $TARGET\_TH(Threshold)$ は 50，$HIGH\_TH$ は 100，LOW_TH は 25 とした．DDR におけるデータ再配置の契機は，文献 [69] に従い IOPS が LOW_TH 以下の HDD 上のブロックに入出力が行われた時点とした．PDC のキュー数は文献 [75] に従い 12 とした．HDD 2 台の場合はキュー 0 から 5 を Disk#1 に，6 から 11 を Disk#2 にそれぞれ対応させた．HDD 5 台の場合は，キュー 0,1 を Disk#1，1,2 を Disk#2，3,4 を Disk#3, 6-8 を Disk#4, 9-11 を Disk#5 にそれぞれ対応させた．PDC におけるデータ再配置の契機は文献 [75] に従い 30 分とした．

HDD 消費電力の計測

前節で生成した入出力トレースを HDD 上で再生し，HDD の消費電力を計測した．計測期間は基本入出力トレースの先頭から 30 分である．トレース再生には blktrace[14] を用いた．提案手法及び PDC は，DB データ単位の入出力トレースを，DB データと物理ブロックの対応関係を用いて HDD の物理ブロック単位の入出力とレースに変換して使用した．

トランザクションスループットの計算

提案手法では，まずアルゴリズム 1 により求めたデータ配置に合わせてデータを HDD に配置し，TPC-C を実行してスループットを計測した．スループットの計算値 TP_E は，TP を計測されたスループット，w を計測期間中のスピンアップ待ち時間の合計値，d を計測期間の長さとすると，$TP_E = TP \times (1 - w/d)$ により求めた．PDC については PDC のデータ配置計算アルゴリズムにより求めたデータ配置に合わせてデータを HDD 上に配置し，提案手法と同じ方法でスループットを計算した．DDR については数十入出力毎にブロック配置が変化する可能性があるため，基本入出力トレース取得時に取得したトランザクションスループットを前述の式に当てはめスループットを計算した．

5.4.2　評価結果

HDD 2 台の場合

図 5.9. に HDD 2 台の場合の HDD 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示す．

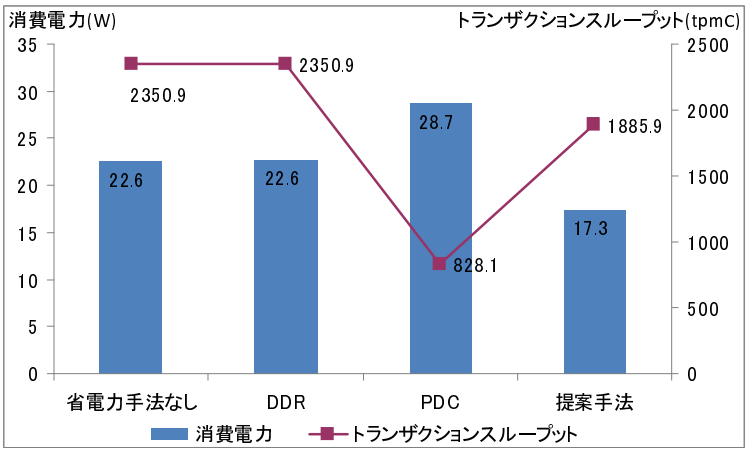

図 5.9: HDD 消費電力とトランザクションスループット (2HDD)

図 5.9 より，提案手法の消費電力は 17.3W であり省電力手法なしの場合と比較して消費電力を 23.3%削減できた．DDR の消費電力は省電力手法なしの場合と同等，PDC の消費電力は 28.7W であり省電力手法なしの場合と比較して 26.9% 増加した．DDR，PDC と比較し提案手法は消費電力を削減できている．

提案手法では，入出力の発行間隔が短いデータを一方の HDD (Disk#1) に配置したこと，および他方の HDD (Disk#2) に配置したデータへの write 遅延の適用により，Disk#2 の入出力発行間隔を Break Even Time 以上に伸ばすことができた．DDR の省電力効果が見られないのは，2 台の HDD の合計 IOPS を $TARGET\_TH$ で除した値を切り上げた値 (Hot HDD 台数) が 2，Cold HDD が 0 となり省電力機能が適用されなかったためである．PDC は消費電力が約 27%上昇した．これはデータの配置を HDD への入出力発行間隔ではなく IOPS に基づき実施しているためである．この結果，IOPS が少ないデータを格納した HDD の入出力発行間隔が，HDD の Spindown タイムアウト時間である 5 秒よりは長く Break Even Time よりは短い状態となり，消費電力を削減することはできなかった．

提案手法のトランザクションスループットは 1885.9tpmC であり省電力手法なしの場合と比較して 19.8%減，DDR は省電力していないのでトランザクションスループットに変化はない．PDC は 828.1tpmC(64.8%減) であった．

提案手法で，DBMS のログと DB の表・索引が同一の HDD に配置されたことによるログ入出力の応答時間の低下のため，トランザクションスループットが低減している．PDC がトランザクションスループットが大きく低下した理由は提案手法と同様にログ入出力の応答時間の低下に加え，HDD の起動待ちが 30 分間に約 50 回発生したためである．

初期配置からのデータ移動量は DDR は省電力手法が適用されなかったため 0，PDC は 4,522MB，提案手法は 472MB であった．PDC の移動量が多い理由は，容量の大きなデータ (Stock, OrderLine 等) が PDC が IOPS が少ないデータを配置する HDD と判断した HDD 上にあり，それらを移動したためである．提案手法では HDD 内の Hot データの容量により Hot HDD を決定しているため，データ移動量を削減できた．

トランザクション当たりの消費電力は省電力手法なしの場合は 9.6W/tpmC であるのに対し，提案手法が 9.2W/tpmC，DDR が 9.6W/tpmC，PDC が 34.6W/tpmC であった．省電力手法の場合と比較して提案手法は 4.3%減と最もすぐれており，DDR は同等，PDC は 260.4%増であった．

提案手法はアプリケーションの入出力挙動を HDD の省電力に利用している．このため最悪の場合においても消費電力やスループットを省電力なしの場合と同等程度に抑えることが可能である．

HDD 5 台の場合

HDD 消費電力とトランザクションスループット 図 5.10. に，HDD 5 台の場合の HDD の消費電力とトランザクションスループットを示す．

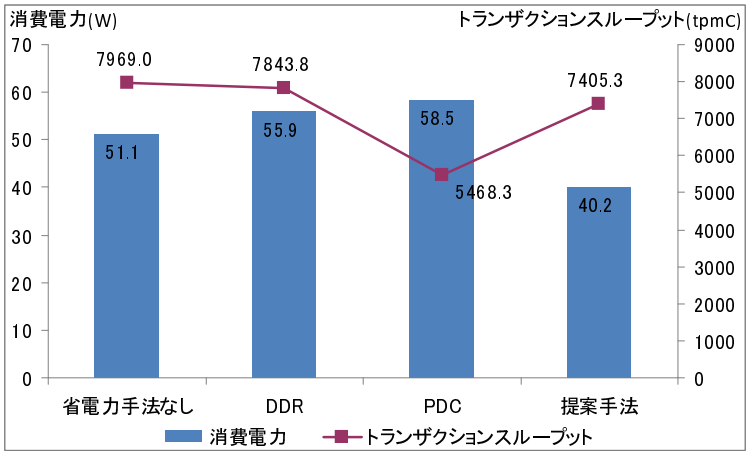

図 5.10: HDD 消費電力とトランザクションスループット (5HDD)

提案手法の消費電力は 40.2W であり，省電力手法の場合と比較して 21.4% 低減できた．DDR は 55.9W(9.4%増)，PDC は 58.5W(14.5%増) であった．DDR の消費電力が増加した理由は，OLTP はランダム入出力のため IOPS が多いブロックの予測を誤ったためである．この結果 Cold HDD への入出力発行間隔が Break-Even Time より短くなり，HDD の Spinup に伴う電力が増加した．PDC の消費電力が増加した理由は，HDD 2 台の場合と同様，IOPS に基づきデータ配置を決めたため，各 HDD 毎に入出力発行間隔が平均化され Break Even Time より長い入出力発行間隔がほとんどできなかったためである．

また，提案手法のトランザクションスループットは 7405.3tpmC であり省電力手法なしの場合と比較して 7.1%減であった．DDR は 7843.8tpmC(1.6%減)，PDC は 5468.3tpmC(31.4%減) であった．PDC のスループットが低いのは，入出力発行間隔が短く HDD の起動待ちが多数発生したためである．提案手法と DDR はそれほど低下していない．

初期配置からのデータ移動量は DDR は 7,184MB，PDC は 1,684MB，提案手法は 180MB であった．DDR の移動量が多い理由は IOPS の高いブロックの予測に失敗し Cold HDD 上のブロックの移動が多発したこと，およびデータ交換をおこなっているためである．HDD 2 台の場合と同様の理由により，提案手法はデータ移動量を削減できている．

トランザクション当たりの消費電力は，省電力手法なしの場合が 6.4W/tpmC であるのに対し，提案手法が 5.4W/tpmC，DDR が 7.1W/tpmC，PCD が 10.7W/tpmC であった．消費電力制御なしの場合と比較して DDR が+11.1%，PDC は+66.8%と悪化したが，提案手法は-15.4%と大幅な削減を達成している．以上の結果より，より高いスループットが求められる環境においても提案手法がアプリケーション実行時の HDD の消費電力削減できることが示された．

提案手法はアプリケーションの入出力挙動を HDD の省電力に利用している．このため最悪の場合においても消費電力やスループットを省電力なしの場合と同等程度に抑えることが可能である．

入出力発行間隔 図 5.11 に HDD 5 台を用いて評価を行った際の Break Even Time 以上の長さの入出力発行間隔の合計時間と回数を，5.12 にスピンダウンタイムアウト (5 秒) 以上 Break Even Time 未満の入出力発行間隔の合計時間と回数をそれぞれ示す.

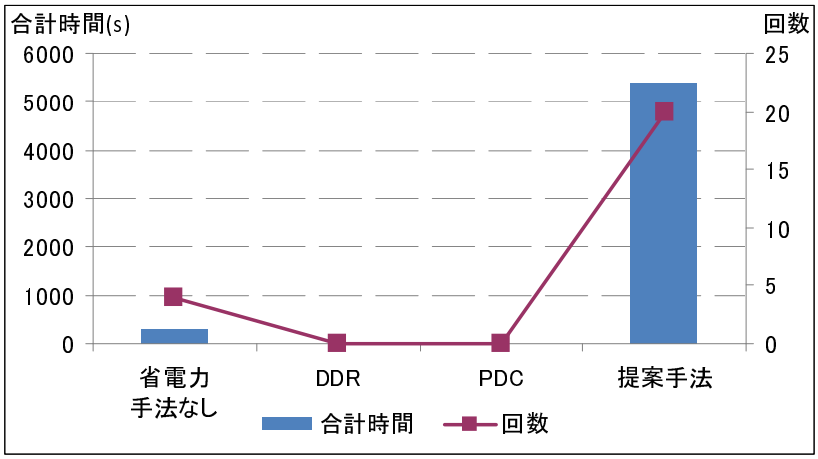

図 5.11: 入出力発行間隔 (2HDD)

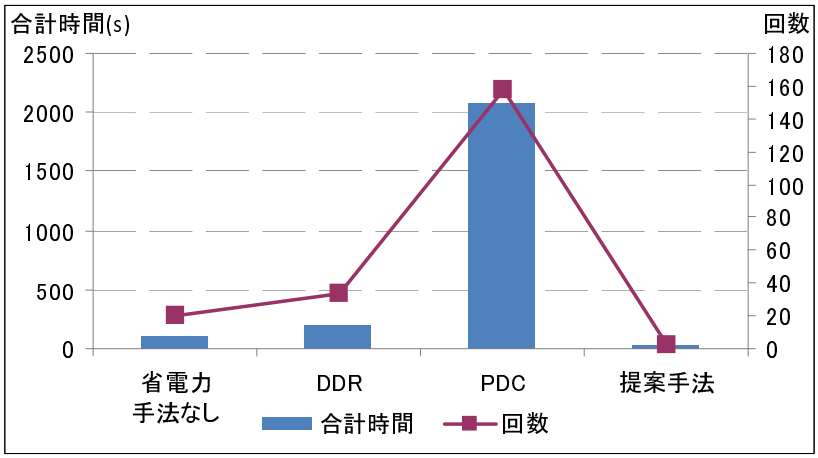

図 5.12: 入出力発行間隔 (5HDD)

図から分かるように，提案手法が，タイムアウト時間以上 BreakEven Time 未満の入出力発行間隔を増加させることなく Break Even Time 以上の長さの入出力発行間隔を多く作り出せていることが分かる．これは，提案手法が入出力発行間隔を基準にデータ配置を決定していること，および write 遅延の効果である．Write 遅延を行った場合，HDD 5 台構成では HDD に対する write 間隔は約 350 秒であったが，write 遅延を行わない場合は約 11 秒であった．この結果より，アプリケーションの入出力挙動特性と DBMS の HDD への write 方法に関する知識を活用することで，入出力発行間隔を大きく伸ばすことが可能であることを確認できた.

HDD 起動待ち時間の影響 4.2 節で述べたように，HDD を Standby 状態から Active/Idle 状

態に移行するには約 8 秒の時間を要する．そこで，HDD の起動待ちが TPC-C の応答時間にどの程度影響を与えるかを調査した．TPC-C では，StockLevel を除く 4 つのトランザクションの応答時間の 90%が 5 秒未満，StockLevel は 20 秒未満であることが規定されている [7]．そこで，省電力手法なしの場合とデータ配置制御のみを行った提案手法 (HDD 省電力機能は使用せず) に起動待ち回数を加算した場合のそれぞれについて，HDD 5 台の場合の応答時間の 90%値を調査した．この結果両ケースとも NewOrder, Payment, OrderStatus, StockLevel の応答時間の 90%値は 0.2 秒，Delivery が 0.4 秒であり，TPC-C の性能要件を満たしていた．

DB サイズに関する考察 本評価で用いた DB は Warehouse 数が 10，サイズは約 4GB である．これは，HDD 5 台を用いた場合の全 HDD 容量の約 0.1%と小さい．しかしながら，TPC-C 実行時の HDD へのアクセス頻度は DB サイズではなく，トランザクションスループットにより決まる．実験結果から分かる通り，HDD 5 台の場合のトランザクションスループットは HDD 2 台の場合と比較し十分に高く，Warehouse 数が 10(DB サイズ 4GB) の場合は 5 台の HDD に対する入出力負荷は高いと考えられる．一方，DB サイズを増やした場合，図 5.4 に示した通り OrderLine 表などの省電力効果の低い表の入出力発行間隔は短くなる (入出力数が増加する) が，Warehouse や District 表などの省電力効果の高い表の入出力発行間隔は長くなる．これは，DB サイズ (Warehouse 数) が大きくなると，入出力に伴う DB バッファのページのロック回数が増え入出力オーバーヘッドが増加するためである．この結果，DB サイズが大きい場合は省電力効果の高い表の入出力発行間隔が伸び，省電力の機会が増えると考えられる．本計測環境 (Warehouse 数: 10, DB サイズ 4GB) は，HDD の省電力にとって厳しい設定となっているが，十分に省電力が可能となっている．従って，DB サイズがより大きな環境でも，提案手法は HDD の省電力に有効であると考える．

5.4.3 考察

データ再配置の契機

提案手法は，Cold データと判断されていたデータが Hot データと判断された契機，あるいは Hot データと判断されていたデータが Cold データと判断された契機でデータの配置を再計算している．

今回の実験では高スループットで実行しているため生じないが，トランザクションリクエスト数が増えると，Cold データと判定されていたデータが Hot データになり，HDD の起動ペナルティが増加し消費電力が削減できなくなる可能性がある．これに対し，Cold データが Hot データになった契機でデータ再配置することにより，大幅な消費電力の上昇を抑えることが可能であると考える．

逆にトランザクションリクエストが減ると，入出力発行間隔は広くなり，Hot データと判断されていたデータが Cold データになる場合がある．これに対し，Hot データが Cold データになった契機でデータ再配置を行うことにより，さらに消費電力を削減することが可能になると考える．

PDC ではデータ再配置は 30 分毎に行うため，次のデータ再配置までの間のロスが大きくなると考えられる．DDR は物理ブロック単位のデータ再配置であるが TPC-C では IOPS

が増えるブロックの予測が難しく高い省電力効果を得ることは困難であると考えられる．アプリケーションレベルの入出力挙動特性を利用することで一層の省電力が可能となる．

OLTP 以外の大量データ処理アプリケーションへの適用

大量のデータを処理するアプリケーションとして，OLTP 以外にも意思決定支援システム (DSS) や科学技術計算向けの大規模デジタルライブラリなどがある．これらのアプリケーションでは，データに対する順次アクセスが中心であることが分かっている．このため HDD デバイスレベルから得られる情報を用いても省電力機能の適用が可能な入出力発行間隔を見つけることは容易であると考えられる．しかし，HDD デバイスから得られる情報を用いたデータ移動では，Cold HDD 上のブロックに入出力が行われるとそれを複数の Hot HDD の中で最も IOPS が少ないブロックと交換する．このため，順次アクセスが行われるデータが入出力の発行順序とは異なる順序で複数の HDD に配置されることになり，入出力性能が低下する可能性がある．一方，提案手法ではデータ単位として表・索引を考え，アプリケーションの論理的な入出力挙動特性を用いるため，アクセスの順序を保持したままデータを再配置する．このため，性能への影響は低いと考えられる．

5.5 まとめ

今後も，デジタルデータは爆発的に増大すると考えられ，それらのデータを格納するために増大し続ける HDD の省電力は急務である．本章では，アプリケーション実行時の HDD 省電力手法を開発し，OLTP を用いて評価した．HDD 上で動作する OLTP の入出力挙動特性の解析結果に基づき，アプリケーションレベルの入出力挙動特性を用いた新たな HDD の実行時省電力手法を提案した．既存手法の DDR，PDC と比較した結果，5 台の HDD 構成の場合には提案手法により数%のトランザクションスループットの低下で HDD の消費電力を 20%以上削減できることを示した．また，提案手法はアプリケーションの入出力挙動を利用するため，最悪の場合においても消費電力やスループットを省電力なしの場合と同等程度に抑えることが可能である．

今後は，提案手法をストレージに拡張すべく，データインテンシブアプリケーション固有の入出力挙動特性をストレージの省電力に活用するためのストレージ管理システムの研究を行う．

第6章　大規模データインテンシブアプリケーションと連携した実行時ストレージ省電力技法

6.1　はじめに

本章では，アプリケーションが持つ固有の入出力挙動特性をストレージの省電力に活用する実行時ストレージ省電力手法を提案する．まず典型的なデータインテンシブアプリケーションであるファイルサーバ，OLTP，及び DSS の入出力トレースを用いて，これらデータインテンシブアプリケーションの入出力挙動特性を明らかにする．ファイルサーバを用いた評価では，Microsoft Research (MSR) のマルチプロダクションエンタープライズワークロードの入出力トレース [23] を用いる．OLTP では TPC-C ベンチマークを実行し取得した入出力トレースを用いる．DSS では TPC-H ベンチマークを実行し取得した入出力トレースを用いる．そして，データインテンシブアプリケーション実行中のストレージ省電力の可能性を示す．

次に，本節において提案する実行時ストレージ省電力フレームワークの設計を示し，それがどのように大規模なデータインテンシブアプリケーションの入出力挙動とストレージ省電力手法を結びつけるかについて述べる．提案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，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．

その後，前述のデータインテンシブアプリケーションであるファイルサーバ，OLTP，及び DSS の入出力トレースを用いて提案手法の定量的評価を行う．ストレージ上で入出力トレース再生ツール [14] を用いて入出力トレースの再生を行い，その消費電力を実際に計測する．さらに，ファイル単位のデータ再配置を行う Popular Data Concentration [75]，及びストレージのブロック単位のデータ再配置を行う Dynamic Data Reorganization [69] と提案手法を比較する．ファイルサーバや OLTP，DSS 等のデータインテンシブアプリケーションとの連携による省電力の可能性を明らかにするとともに，ストレージの実行時省電力においてアプリケーションの入出力挙動特性を用いることの優位性を示す．

6.2　データインテンシブアプリケーションの入出力挙動特性

本章では，データインテンシブアプリケーションとして取り上げるファイルサーバ，OLTP，及び DSS の入出力挙動特性を明らかにする．さらに，これらアプリケーション実行時の

ストレージ省電力の可能性について議論する．

6.2.1 ファイルサーバの入出力挙動特性と省電力の機会

まず，ファイルサーバの入出力挙動特性と省電力の機会について述べる．

計測環境

ストレージ上で動作するファイルサーバ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 6.1 に示す．

表 6.1: ストレージ上で動作するファイルサーバの設定

Application	Data Size	Workload	Cache Size
File Server (MSR Trace) (6 hr)	19,800,000 records	Replay 入出力 trace using trace reply tool [14] Duration: 6 hr Create 36 volumes on 12 disk enclosures, and assign each volume in MSR trace to volumes in alphabetical order of the volume names.	2 GB (Storage)

ファイルサーバの入出力挙動特性を調査するために，Microsoft Research の入出力トレース [23] をトレース再生ツール [14] を用いてストレージ上で再生した．トレースレコード数は約 19,800,000，再生時間は約 6 時間，ボリュームの数は 36 個である．36 個のボリュームを 12 台のディスク筐体 (#1〜#12) 上に 3 つずつ，ボリューム名の順に配置した．ストレージキャッシュサイズは 2GB である．

ファイルサーバの入出力挙動特性

図 6.1 は，サーバからストレージに発行された平均 read 数と write 数を，図 6.2 はストレージのコントローラからディスクに発行された平均 read 数と write 数をそれぞれ示している．

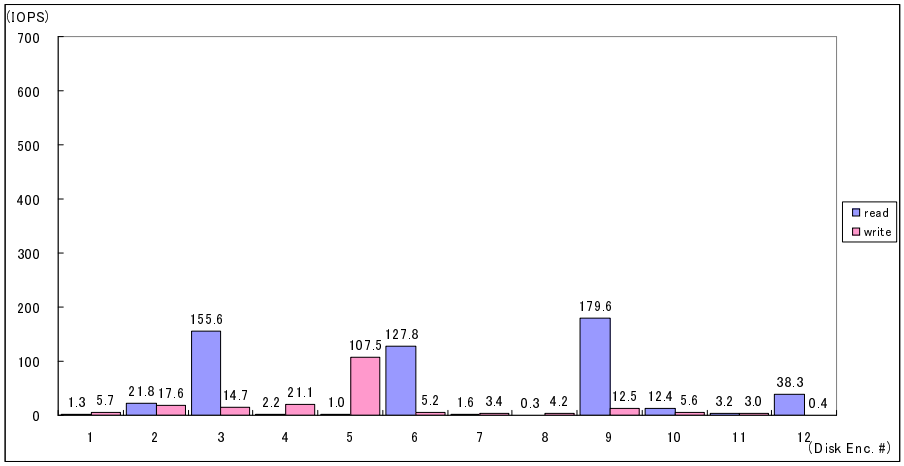

図 6.1: ファイルサーバ稼動中のディスク筐体毎の入出力数 (サーバからコントローラに発行された入出力)

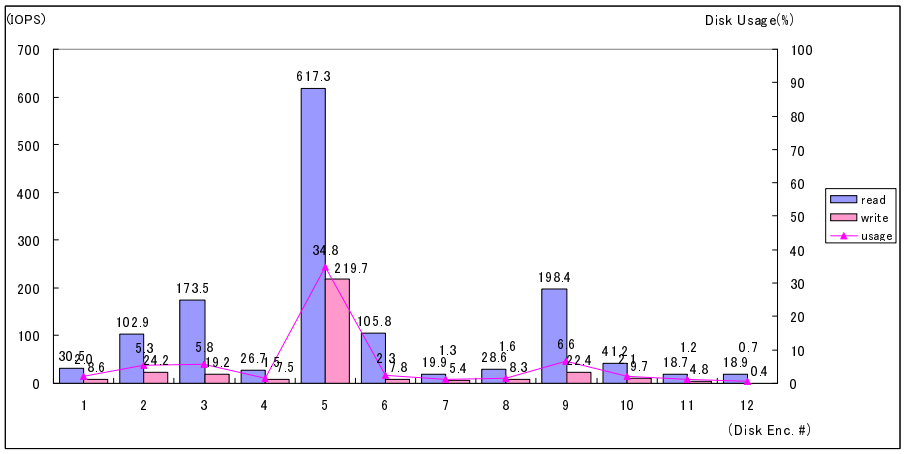

図 6.2: ファイルサーバ稼動中のディスク筐体毎の入出力数 (コントローラから HDD に発行された入出力)

図 6.1 より，ディスク筐体ごとに平均 read 数，平均 write 数はばらついており，最大平均 read 数はディスク筐体#9 の 179.6IOPS，最大平均 write 数はディスク筐体#5 の 107.5IOPS である．また，平均入出力数が少ないディスク筐体であっても，毎秒数 IOPS の入出力が出ていることが分かる．

図 6.2 より，ディスク筐体内の HDD に対する平均入出力数はディスク筐体#5 が最も高い．これは，TPC-C の場合と同じく write ペナルティのためである．また，何れのディスク筐体も，サーバからコントローラに発行された入出力数よりもコントローラからディスク筐体内の HDD に発行された入出力数の方が多い，これはディスク筐体に対する入出力がランダム入出力であることを示している．

ファイルサーバの入出力応答時間

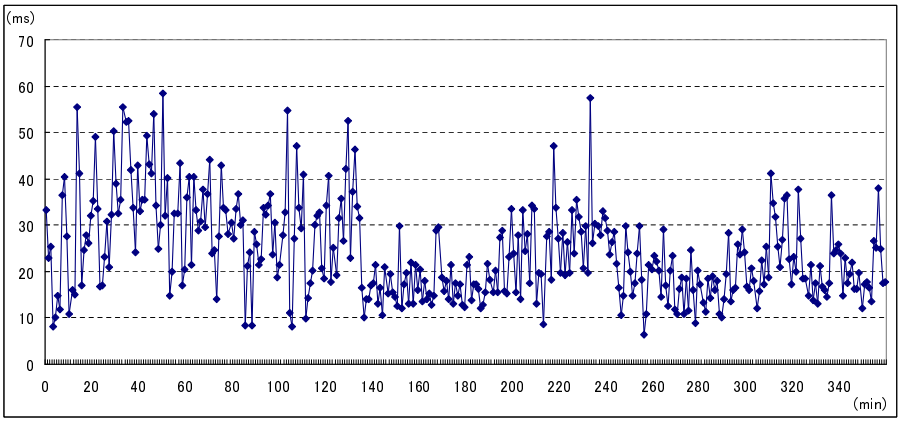

図 6.3: ファイルサーバの入出力応答時間

ファイルサーバの平均入出力応答時間を図 6.3 に示す．縦軸は入出力応答時間の平均値，横軸は経過分数である．図 6.3 から分かるとおり，入出力応答時間にはばらつきがあるが，ほぼ 10ms から 50ms の間に収まっている．その平均値は 21.2ms であった．

ファイルサーバの省電力の機会

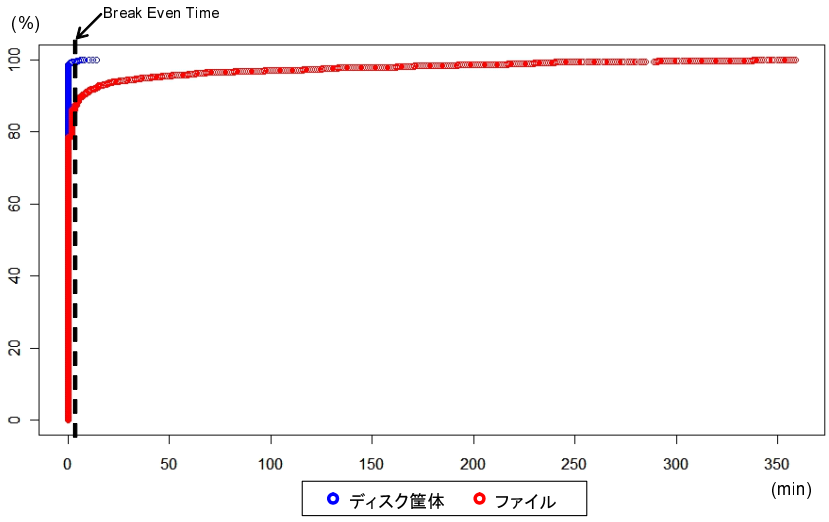

図 6.4: ファイルサーバの入出力発行間隔の分布

次に，ファイルサーバの省電力の機会を調べるために，ファイルサーバの Idle 時間の長さの分布を調査した．図 6.4 は，ディスク筐体毎，及びファイルサーバ内のファイル毎の入出力発行間隔の分布を示している．図の X 軸は分単位の Idle 時間の長さ (また Idle 時間 0 は入出力が行われていたことを示す)，Y 軸は計測期間に占める累積比率である．

図から，ディスク筐体単位では，ほぼ全ての時間で，ディスク筐体に対して入出力が行われていたことが分かる．すなわち，省電力の機会はほとんどない．また，ファイル単位の場合は，電源 OFF の適用が期待できる Break Even Time より長い入出力発行間隔は全累積実行時間の 10%程度であった．電源 OFF を使用できる Idle 時間は全体の 18.0%あることが分かる．これらの結果から，ファイル単位の入出力挙動特性に基づき省電力を考えることで，ファイルサーバにおいてもストレージを省電力できる可能性が高まると考えられる．

6.2.2 OLTP の入出力挙動特性と省電力の機会

次に，OLTP の入出力挙動特性を活かしたストレージの省電力手法を検討するため，HDD 上の OLTP の挙動解析と同じく TPC-C ベンチマークを用い，OLTP の入出力 挙動特性を詳細に解析した．

計測環境

図 4.7 に示した実験環境を用いて，ストレージ上で動作する OLTP の入出力挙動を解析した．

表 6.2: ストレージ上で動作する OLTP の設定

Application	Data Size	Workload	Cache Size
OLTP (TPC-C)	500GB	# of warehouse: 5000 #of threads: 1000 Think time: 0 Duration: 1.8 hr Put log to 1 Storage Device Put DB to 9 Storage Devices (hash distribution)	25GB (DBMS) 2GB (Storage)

OLTP プログラムとして，OLTP の代表的ベンチマークである TPC-C を使って，500GB クラスの DB(Warehouse 数 5,000) を対称に計測を行った．DBMS には Linux 用の商用 DBMS を用い，ダイレクト入出力オプションを使用した．DBMS のバッファサイズはデータベースサイズの 5%である 25GB とした．データベースのサイズにはログのサイズは含んでいない．ログを図 6 中のユニット 1 に，表と索引をユニット 2 から 10 にハッシュ分割機能を用いて分散配置した．また，TPC-C のスレッド数は 1000，Think time 及び Keying time はそれぞれ 0 秒とした．計測にあたっては，TPC-C を停止状態から起動した後 1 時間走らせ，挙動が安定してから 10 分間データをとる作業を 3 回実施しその平均値を用いた．スレッド数が 1000 以上の場合，応答時間は伸びるがスループットは変化しないため，スレッド数は 1000 固定とした．

OLTP の入出力挙動特性

図 6.5 は，サーバからストレージに発行された平均 read 数と write 数を，図 6.6 はストレージのコントローラからディスクに発行された平均 read 数と write 数をそれぞれ示している．

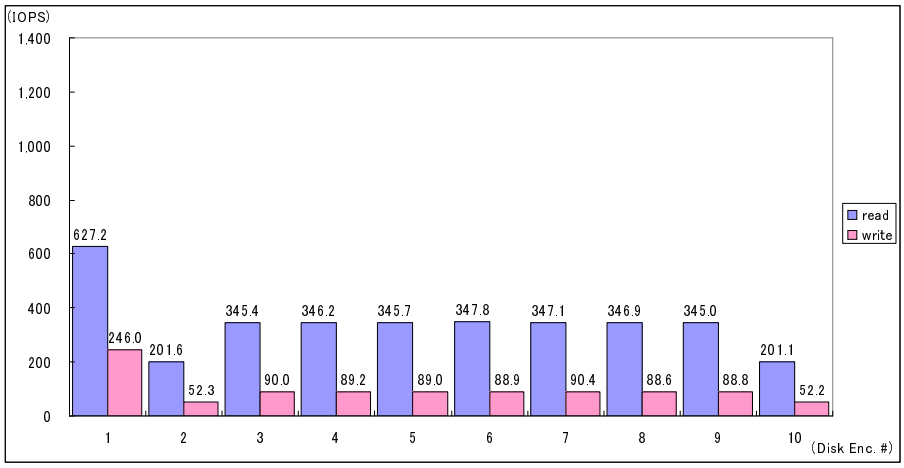

図 6.5: OLTP 稼動中のディスク筐体毎の入出力数 (サーバからコントローラに発行された入出力)

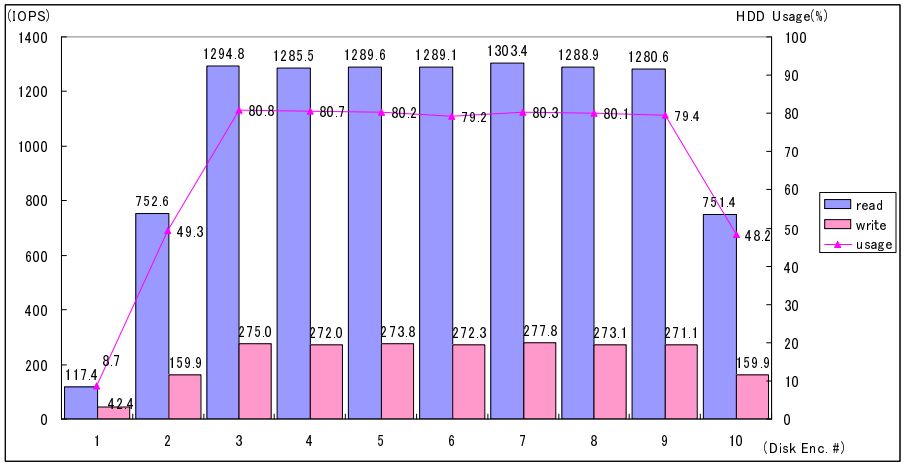

図 6.6: OLTP 稼動中のディスク筐体毎の入出力数 (コントローラから HDD に発行された入出力)

図 6.5 より，read, write ともディスク筐体#1 が他の unit より高いことが分かる．ディスク筐体#1 では，平均 read 数は，平均 write 数の約 2.5 倍である．またディスク筐体#2-10 では，平均 read 数は平均 write 数の約 3.8 倍である．

一方，図 6.6 を見ると，ログを格納したディスク筐体に対する入出力数は少なくなっており，逆に表及び索引を格納したディスク筐体の入出力数は約 3 倍になっている．これは，ログへの入出力はシーケンシャルであるため read や write がまとめられて入出力数が減少

したのに対し，表や索引への入出力が増えたのは RAID の write ペナルティのためである．Write ペナルティとは，RAID のパリティ生成のために必要となる入出力であり，旧パリティデータや旧データの read，新パリティの write が含まれる．これらより，OLTP のようにランダム write が行われるアプリケーションでは，write ペナルティがディスクの負荷を高めていることが分かる．実際，ディスク筐体#1 のディスクの使用率は 8.7%程度であるのに対し，入出力数の少ないディスク筐体#2 及び#11 を除く他のディスク筐体の使用率は 80%を超えていた．

OLTP のトランザクションスループット

次に，OLTP の性能の計測結果について述べる．OLTP の秒当りのトランザクション処理量 (tps) の推移を図 6.7 に示す．図 6.7 の縦軸は秒当りのトランザクション処理量，横軸は経過秒数である．図 6.7 から分かるとおり，トランザクション処理量は計測期間を通じてほぼ一定であり，その平均は約 71.5 tps であった．

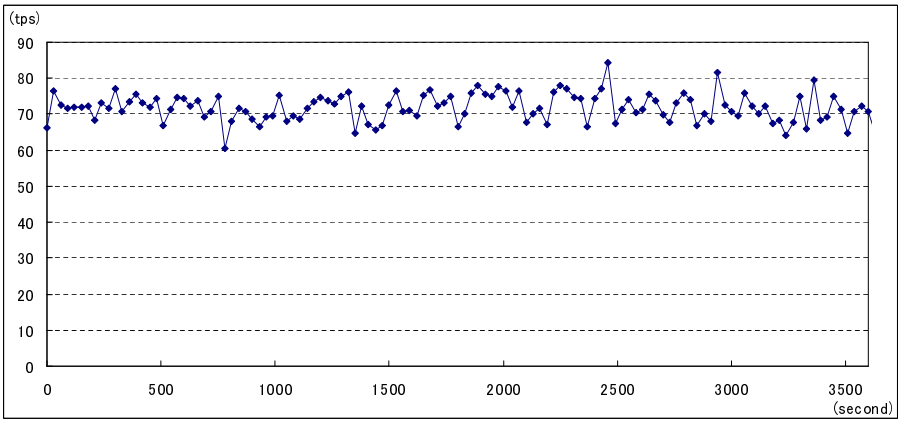

図 6.7: OLTP のトランザクションスループット

OLTP の省電力の機会

次に，OLTP における省電力の機会を調べるために，OTLP の Idle 時間の長さの分布を調査した．図 6.8 は，ディスク筐体毎，及び OLTP の表・索引毎の入出力発行間隔の分布を示している．図の X 軸は分単位の Idle 時間の長さ (Idle 時間の最大値は 60 分としている，また Idle 時間 0 は入出力が行われていたことを示す)，Y 軸は計測期間に占める累積比率である．

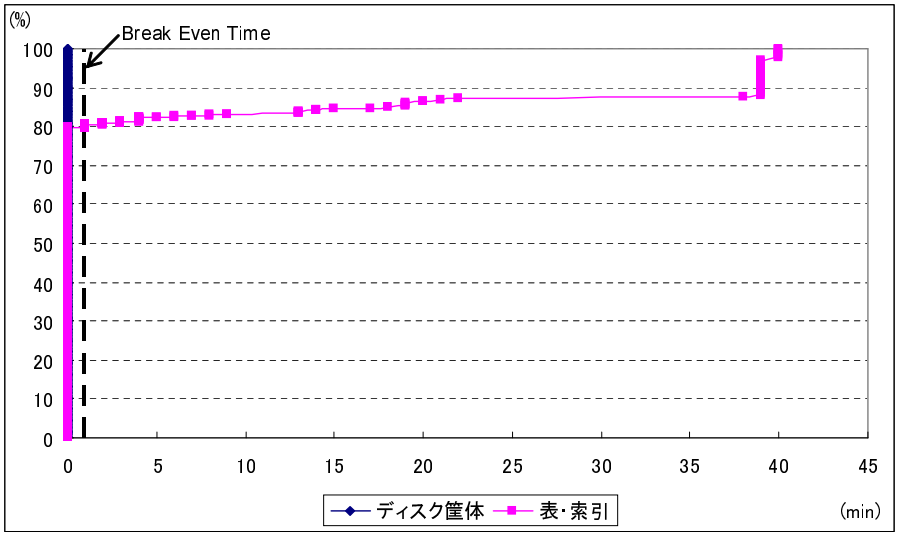

図 6.8: OLTP の入出力発行間隔の分布

図から，OLTP は全ての計測期間に渡って入出力が発行されており，ディスク筐体単位の省電力では，省電力機能を使用する余地がないことが分かる．一方，表・索引単位の入出力では，Break Even Time より長い入出力発行間隔が存在しており，全累積実行時間のうち 18.7%は電源を OFF にできる可能性があることが分かる．この入出力発行間隔の分布は，時間が経過してもほぼ同じ傾向を示した．これらの結果から，表・索引単位の入出力挙動特性に基づき省電力を考えることで，高スループットで動作する OLTP が使用するストレージを省電力できる可能性が高まると考えられる．

6.2.3 DSS の入出力挙動特性と省電力の機会

最後に，DSS の入出力挙動特性と省電力の機会について述べる．

計測環境

ストレージ上で動作する DSS 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 6.3 に示す．DSS プログラムとして，やはり DSS の代表的ベンチマークである TPC-H ベンチマークを用いて計測を行った．DBMS には OLTP の場合と同じ商用 DBMS を用い，ダイレクト入出力オプションを使用した．データベースサイズは約 100GB (Scale Factor 100)，DBMS のバッファサイズは 5GB (データベースサイズの 5%) とした．データベースのサイズにログ及び作業表領域のサイズは含んでいない．ログ及び作業表を図 4.7 中のディスク筐体#1 に，表と索引をディスク筐体#2 から 9 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．

表 6.3: ストレージ上で動作する DSS のの設定

Application	Data Size	Workload	Cache Size
DSS (TPC-H)	100 GB	SF=100 Run Q1 to 22 sequentially Duration: 6 hr Put log and work files to 1 Storage Device Put DB to 8 Storage Devices (hash distribution)	5 GB (DBMS) 2 GB (Storage)

DSS の入出力挙動特性

図 6.9 は，サーバからストレージに発行された平均 read 数と write 数を，図 6.10 はストレージのコントローラからディスクに発行された平均 read 数と write 数をそれぞれ示している．

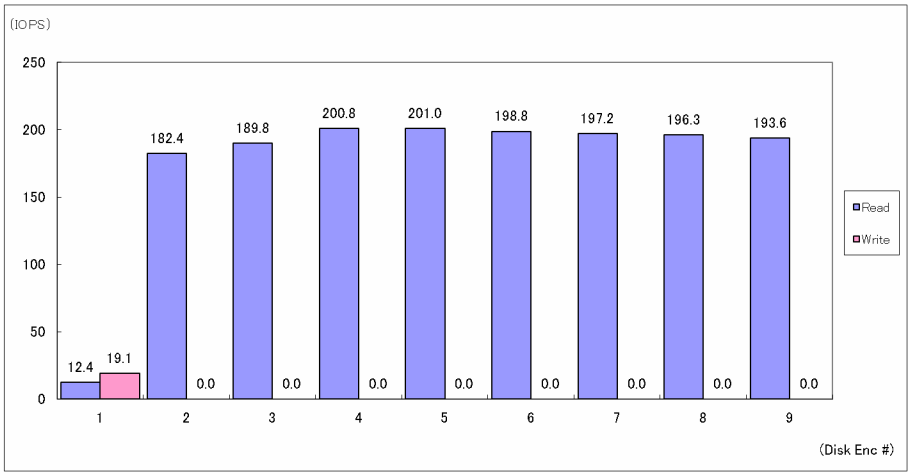

図 6.9: DSS 稼動中のディスク筐体毎の入出力数 (サーバからコントローラに発行された入出力)

図 6.9 より，平均 read 数はディスク筐体#2-9 間でほぼ等しいことが分かる．また，ディスク筐体#2-9 に対する write は行われていないことが分かる．ディスク筐体#1 では，read 数の方が write 数より多い．これは作業表をディスク筐体#1 に格納しているためである，

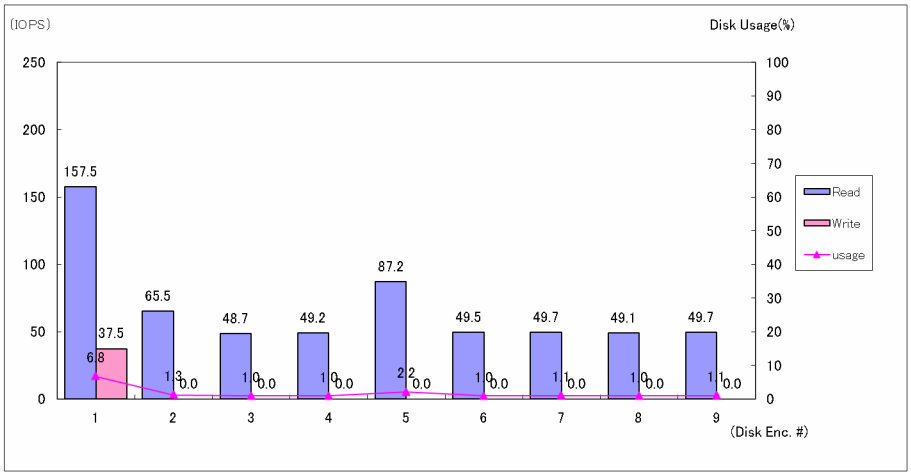

図 6.10: DSS 稼動中のディスク筐体毎の入出力数 (コントローラから HDD に発行された入出力)

図 6.10 よりディスクに対する入出力はディスク筐体#1 を除いて減少していることが分かる．ディスク筐体#1 の入出力数が増加した理由は先に述べた write ペナルティのためである (作業表に対する write)．ディスク筐体#2 から#10 の入出力数が減少したのは，これらのディスク筐体に対する入出力は LINEITEM 表などの大きな表の Full Scan が主であり，ストレージが入出力をまとめて 1 つの入出力として先読みしたためである．

DSS のクエリ応答時間

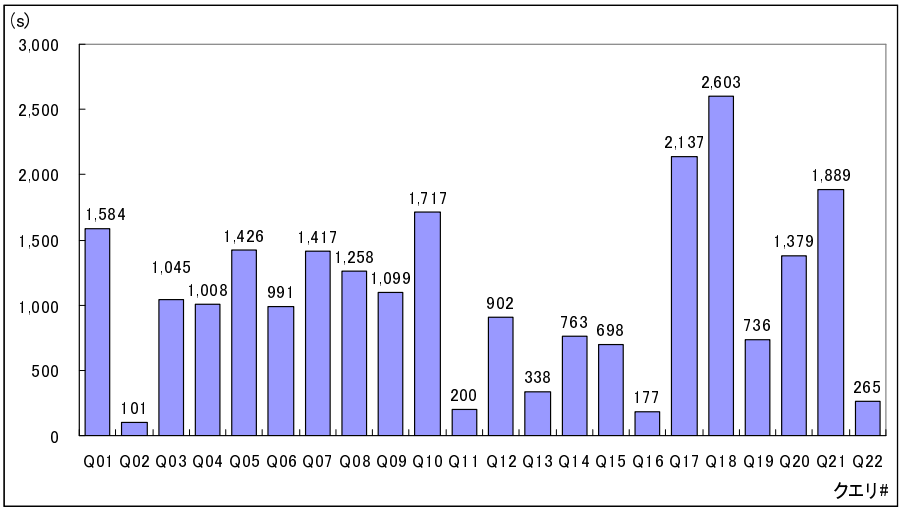

図 6.11: DSS のクエリの応答時間

DSS のクエリ毎の平均応答時間を図 6.11 に示す．クエリの応答時間はクエリ毎に異なっているが，最も応答時間が短いクエリ (Q2) の応答時間は約 94 秒，最も長いクエリ (Q18) は約 2,584 秒であった．

DSS の省電力の機会

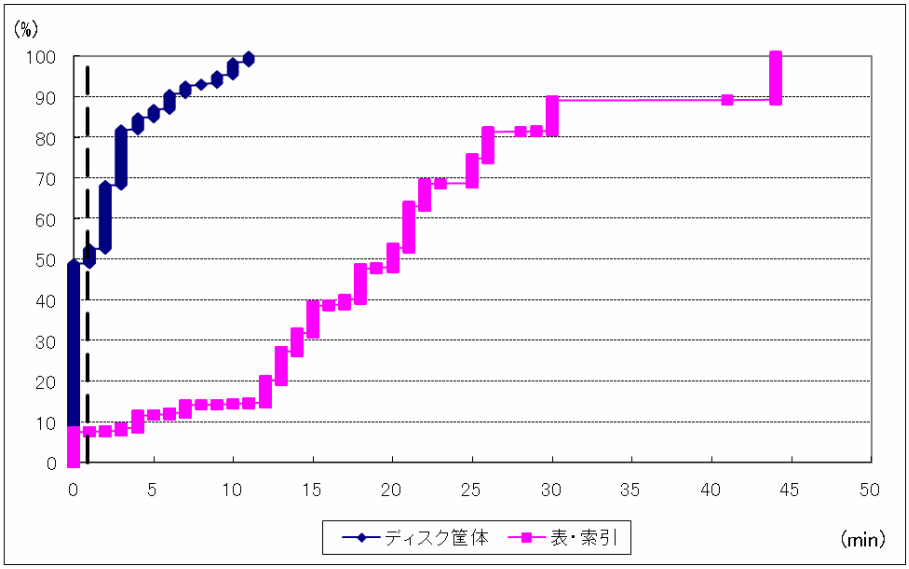

図 6.12: DSS の入出力発行間隔の分布

次に，DSS における省電力の機会を調べるために，DSS の Idle 時間の長さの分布を調査した．図 6.12 は，ディスク筐体毎，及び DSS の表・索引毎の入出力発行間隔の分布を示している．図の X 軸は分単位の Idle 時間の長さ (Idle 時間の最大値は 60 分としている，また Idle 時間 0 は入出力が行われていたことを示す)，Y 軸は計測期間に占める累積比率である．

図から，ディスク筐体単位では，入出力が行われていた時間が全計測時間 (累計) 中の約 49.0%，電源 OFF を使用できる Idle 時間は全体の約 52%あることが分かる．また，DSS の表・索引単位では全累積実行時間の 90%以上の時間，ディスク筐体の電源を OFF にできる可能性があることが分かる．

6.3 データインテンシブアプリケーションと連携した実行時ストレージ省電力システムの設計

今日，プロセッサの Halt や DVFS 機能に見られるように，多くの IT 機器が省電力機能を提供している．プロセッサと同様，エンタープライズストレージも単純な HDD の電力制御 [103, 47, 41] に加え，MAID[19, 20] に代表されるような，ディスク筐体の電源 ON/OFF 等の省電力機能を持つようになってきている [64, 9]．しかし，ストレージの省電力機能は，それを使用すれば常に高い省電力効果を得られるわけではない．例えば，ディスク筐体の電源 OFF 機能を用いて省電力効果を得るためには，少なくとも一つの入出力発行間隔はディスク筐体の電源 OFF 及び ON に必要な時間より長くなければならない，

6.2 節で述べたように，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．またそれらの挙動は時間変化に対し

て安定的である．もし，このような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，従来の研究と比べてアプリケーションの実行時の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．

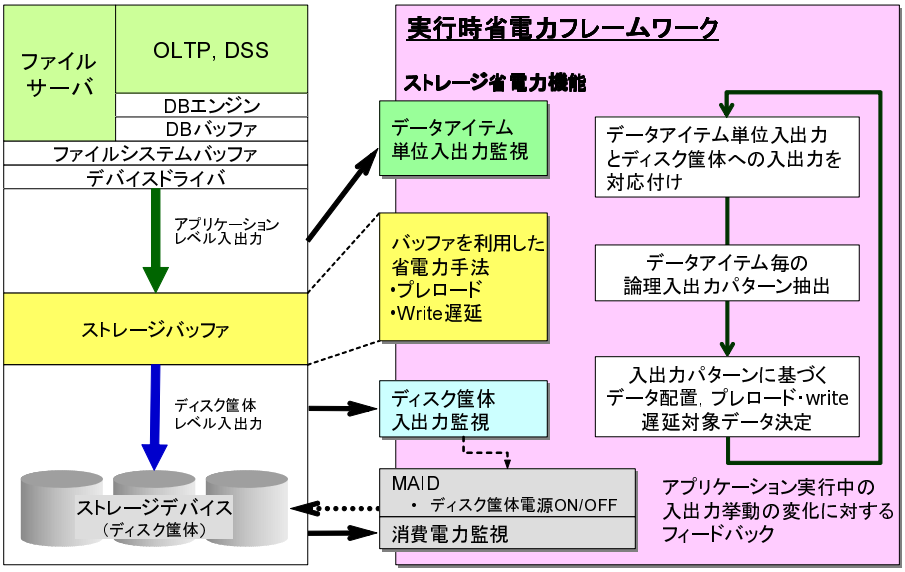

図 6.13: ストレージを対象とした実行時ストレージ省電力フレームワーク

2 章にて提案した実行時ストレージ省電力フレームワークのストレージへの適用を図 6.13 に示す．ストレージは大規模なバッファを有しており，これはディスク筐体に対する入出力間隔の延伸に重要な役割を果たす．このため，ストレージ省電力機能をストレージバッファ層に組み込む．

ストレージを対象とした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としてデータアイテム毎の入出力を，ストレージレベル入出力としてストレージのディスク筐体毎の入出力をそれぞれ監視する．ここでデータアイテムとはファイルサーバのファイルや OLTP 及び DSS の表・索引をディスク筐体の単位で分割したものである．データアイテムの詳細は 6.3.2 節で述べる．実行時省電力フレームワークはまずデータアイテム単位入出力とディスク筐体単位入出力を対応付ける．次にデータアイテム毎の論理入出力パターンを抽出し，入出力パターンに基づきデータアイテムの配置やプレロード，write 遅延の対象とするデータアイテムを決定する．また，アプリケーション実行中の入出力挙動の変化により論理入出力パターンが変化した場合には，データアイテムの配置やプレロード，write 遅延の対象とするデータアイテムを見直す．さらに，本フレームワークはディスク筐体に対する入出力を監視し，ディスク筐体に入出力が行われていない場合にはディスク筐体の電源を OFF にする．

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．入出力パターンは次の 4 種類である．第一は，モニタリング期間中にアプリケーションから入出力が発行されなかったこ

とを識別するための入出力パターンである．本パターンに該当するデータを識別することにより，容易に電源 OFF などのストレージ省電力機能を適用できる可能性が増加する．第二はストレージキャッシュを用いることで read 入出力間隔を延伸できる可能性があるデータを識別するためのパターンである．第三は同じくストレージキャッシュを用いるが，read ではなく write 入出力間隔を延伸できる可能性があるデータを識別するためのパターンである．最後はストレージ省電力機能を適用することができないデータを識別するための入出力パターンである．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る．この結果，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．

6.3.1 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．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．

6.3.2 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．

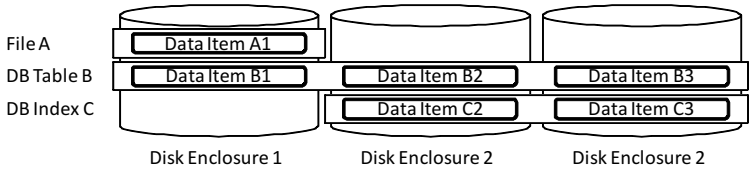

図 6.14: データアイテム

図 6.14 はデータアイテムの例を示している．ディスク筐体#1 上のファイル A は，一つのデータアイテム A1 を有している．ディスク筐体#1 から#3 に跨っている DB テーブル B は，3 つのデータアイテム B1, B2, B3 を持つ．ディスク筐体#2，#3 上の DB 索引 C は，2 つのデータアイテム C2 及び C3 を持つ．

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 Break Even Time 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．

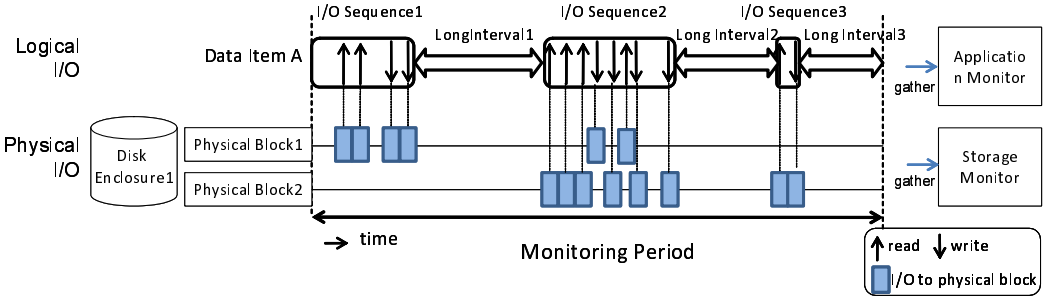

図 6.15: ロングインターバルと入出力シーケンス

図 6.15 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 6.15 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論理入出力パターンである (すなわち，全ての入出力間隔が Break Even Time より短い)．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．

6.3.3　実行時ストレージ省電力フレームワーク

従来のストレージ省電力手法と，本節において提案する実行時ストレージ省電力フレームワークを図 6.16，6.17 にそれぞれ示す．図 6.16 の左側のブロック図はシステム機能と従来のブロックベースのストレージ電力管理モジュールを示している．従来のブロックベースのストレージ省電力手法は，ストレージモニタを持つ．ストレージモニタ機能はディスク筐体に対する入出力の挙動をトレースし，それらを物理ブロック[1]に対する入出力トレースとしてストレージモニタのリポジトリに格納する．

従来手法では，アプリケーションレベルの入出力挙動とストレージレベルの入出力挙動は結び付けられていない．このため，従来の手法では，同一物理ブロック上の複数のデータアイテムに対する入出力を識別することができず，アプリケーション毎に適切な省電力機能を選択・適用することが難しい．

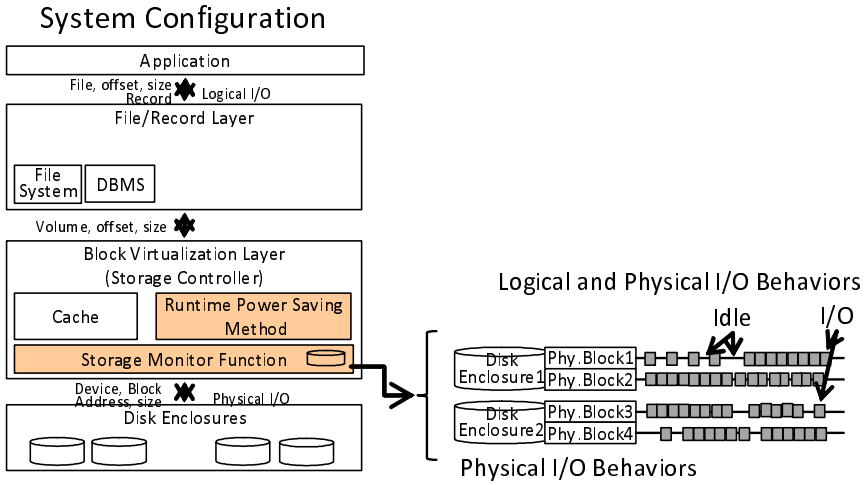

図 6.16: 従来のストレージ電力省電力手法

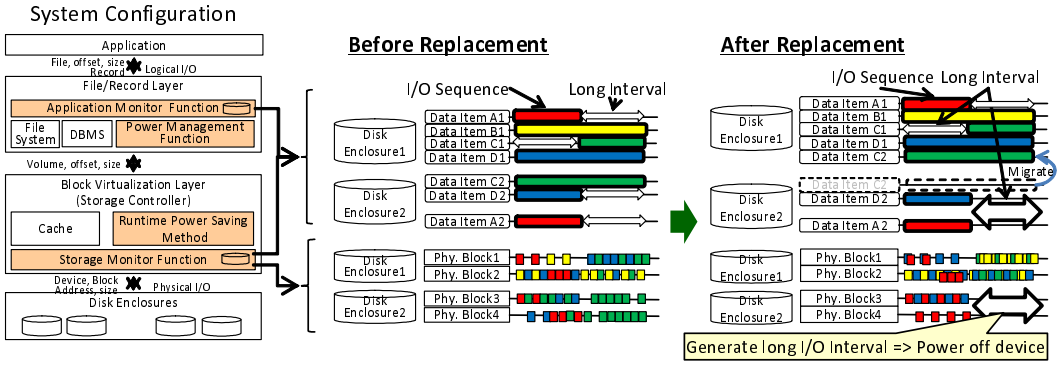

図 6.17: 実行時ストレージ省電力フレームワーク

図 6.17 の左側のブロック図はシステム機能とストレージ電力管理モジュールを示して

[1]ストレージ内のデータ移動単位であり，サイズは数百 GB から数 GB である．

いる．ストレージ省電力フレームワークでは，アプリケーションレベルとストレージレベル双方の入出力をモニタリングする．アプリケーションモニタ機能はアプリケーションの入出力挙動を監視し，それらを論理入出力トレースとしてアプリケーションモニタのリポジトリに格納する．ストレージモニタ機能はディスク筐体に対する入出力の挙動をトレースし，それらを物理入出力トレースとしてストレージモニタのリポジトリに格納する．電力管理機能はこれら蓄積された入出力トレースを収集する．また，電力管理機能は記録された論理入出力トレースをデータアイテムと関連付け，各データアイテムの論理入出力パターンを決定する．その後，電力管理機能はデータアイテムを配置するディスク筐体，及びディスク筐体に対する適切な省電力手法を決定する．電力管理機能は，実行時省電力機能に省電力手法を伝える．実行時省電力機能はディスク筐体の電源 ON/OFF，及びストレージキャッシュを用いた省電力手法を実行する．

実行時ストレージ省電力フレームワークは，図 6.17 に示すようにアプリケーションレベルの入出力挙動と物理レベルの入出力挙動を結びつける．図 6.17 の左側はデータアイテムに対するアプリケーションレベルの入出力挙動と物理ブロックに対する物理レベルの入出力挙動との関係を示している．図 6.17 の右側は，ストレージ電力管理システムにおいてこれらの関係を利用することによるストレージ省電力の可能性を示している．

図 6.17 左に示すように，提案システムではデータアイテムと関連付けられた論理入出力トレースを用いることでデータアイテム A1, C1,D2 及び A2 がロングインターバルを含むことを見つけることができる．さらに，データアイテムとそれらの物理ブロックとの対応関係を用いることにより，これらのデータアイテムが配置されているディスク筐体を識別することが可能である．これにより，図 6.17 の右側に示すように，提案システムでは，データアイテム C2 をディスク筐体#1 に移動することができ，ディスク筐体#2 に電源 OFF 機能を適用する機会を増やすことが可能となる．アプリケーションレベルの入出力パターンを用いない場合，物理ブロック#1 から#4 においてロングインターバルを見つけることができず，従ってディスク筐体の電源を OFF にすることができない．

6.3.4 ストレージ省電力方式

図 6.17 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使用する．

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 6.17(b) に示すように，ディスク筐体#2 上のデータアイテム C3 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 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．

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．エンタープライズストレージは，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 [9]．そこで，この大容量のキャッシュを用い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．図 6.18 に示すように，データアイテム A1 と B2 をストレージキャッシュにプレロードすることにより，ディスク筐体に対する read 入出力の間隔を Break Even Time 以上とできる可能性が高まる．

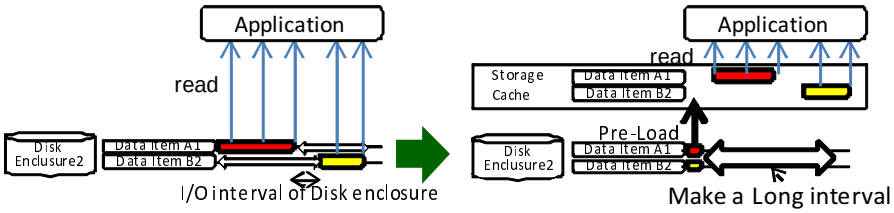

図 6.18: プレロードの効果

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．図 6.19 に示すように，データアイテム A1 と A2 に write 遅延を適用することにより，ディスク筐体に対する write 入出力間隔を伸ばすことが可能となる．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．

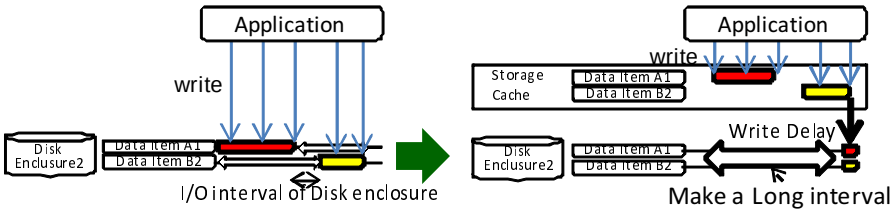

図 6.19: Write 遅延の効果

6.4 モニタリング機能

本節では，実行時ストレージ省電力手法のモニタリング機能について述べる．図 6.17 に示すように，ストレージ電力管理システムは，アプリケーションモニタとストレージモニタを有し，アプリケーションレベルの入出力挙動とストレージレベルの入出力挙動をそれぞれ監視する．アプリケーションモニタは図 6.17 のファイル/レコード層に配置され，ストレージモニタはブロック仮想化層に配置される．

6.4.1 アプリケーションモニタ

アプリケーションモニタは，論理マッピング情報及び論理入出力トレースという 2 種類の情報を収集する:

- **論理マッピング情報**: 論理マッピング情報はデータとボリュームの対応関係を保持する．ボリュームとは，ファイル/レコード層に提供される仮想的な外部記憶の管理単位である．ブロック仮想化層がこれを提供する．
- **論理入出力トレース**: 論理入出力トレースとはアプリケーションがデータに対して発行した入出力のトレースであり，入出力が発行された時刻，データの識別子，入出力先の位置 (データ内の)，入出力サイズ，及び入出力種別 (read あるいは write) を含む．

論理マッピング情報は，データの作成，拡張，縮小，削除に応じて作成，更新，削除される．マッピング情報はアプリケーションモニタのリポジトリに格納される．論理入出力トレースはアプリケーションが入出力を発行した時に捕捉され，アプリケーションモニタのメモリに蓄積される．メモリが一杯になると，入出力トレースはアプリケーションモニタのリポジトリに出力される．論理入出力トレースは DBMS が収集している情報と同等の情報であり，収集のオーバヘッドは小さいと考える．

6.4.2 ストレージモニタ

ストレージモニタは物理マッピング情報，物理入出力トレース，ディスク筐体の電力状態，及びディスク筐体の消費電力を収集する．これらは，それぞれ次のような情報を持つ．

- **物理マッピング情報**: 物理マッピング情報はボリューム内のオフセットとディスク筐体のブロックアドレスとの対応関係を持つ.
- **物理入出力トレース**: 物理入出力とはブロック仮想化層がディスク筐体に対して発行する入出力である．物理入出力トレースは，入出力が発行された時刻，入出力が発行されたディスク筐体の名称， 入出力が発行されたブロックのディスク筐体内の位置，及び入出力種別 (read あるいは write) を含む.
- **ディスク筐体の電力状態**: ディスク筐体の電力状態には，ディスク筐体の名前，電力状態，及びディスク筐体の電力状態が変化した時刻を含む.
- **ディスク筐体の電力消費量**: ディスク筐体の消費電力情報は，ディスク筐体の名前，ディスク筐体の消費電力を収集した時刻，及びディスク筐体の消費電力値を含む.

6.5 電力管理機能

6.5.1 概要

実行時ストレージ省電力手法は，アプリケーションとストレージの入出力挙動を一定期間モニタする．モニタ期間が終了すると，管理システムは電力管理機能を呼び出す.

電力管理機能は，まずデータアイテムの論理入出力パターンを決定する．その後，データアイテムの論理入出力パターンに基づきディスク筐体を Hot と Cold に分割する．Hot ディスク筐体とは，主に P3 型のデータアイテムを格納するディスク筐体である．過剰なデータ移動を避けるために，提案手法はできるだけ初期データ配置を維持しようとする．このため Hot ディスク筐体は通常全てのパターンのデータアイテムが格納される．Cold ディスク筐体は P0，P1，P2 型のデータアイテムのみを格納する．Cold ディスク筐体に格納されたデータアイテムの論理入出力パターンに基づき，本電力管理機能は各ディスク筐体に適した省電力機能を選択する．本機能は，データアイテムの論理入出力パターンに基づきそれらデータアイテムの配置を決定する．論理入出力パターンを用いることで，以降に示すように全てのディスク筐体に P3 型のデータアイテムが配置された場合でも Cold ディスク筐体を生成する可能性を持つ.

次に，本機能は Cold ディスク筐体に格納されたデータアイテムに対して write 遅延とプレロードを適用し，Cold ディスク筐体の入出力発行間隔の延伸を試みる．Cold ディスク筐体の入出力発行間隔はは Hot ディスク筐体のそれと比較して長い予想できる．このため，Hot ディスク筐体の入出力発行間隔の延伸を試みるより，Cold ディスク筐体の入出力発行間隔を伸ばした方が，高い省電力効果を得ることができると考えられる．従って，Cold ディスク筐体上のデータアイテムにのみ write 遅延とプレロードを適用する.

本機能は，最初に write 遅延を適用し，次にプレロードを適用する．これは，write 遅延の方がプレロードより高い省電力効果を容易に得られるためである．エンタープライズストレージのキャッシュは不揮発である．エンタープライズストレージは，アプリケーションが write したデータを一旦キャッシュに保持し，その後ディスク筐体に書き出す．このためストレージはディスク筐体にデータを書き出す契機を，アプリケーションからの write 入出力とは独立に決めることができる．一方，read が発行される契機はアプリケーション

の実行時の状態によって決まる．このため，write の予測と比較して read の予測は困難である．

その後，本機能は Cold ディスク筐体のみに電源 OFF 機能が適用されるよう，ストレージを設定する．最後に，本機能は，省電力に適したモニタリング期間を選ぶために，最新のモニタリング期間における入出力発行間隔の分布に基づき，次のモニタリング期間の長さを決定する．

アルゴリズム 2 に本機能の概要を示す．

Algorithm 2 電源管理機能

```
Start application monitor and storage monitor;
while applications are running do
    Wait unitl a monitoring is finished;
    Determine Logical 入出力 pattern of data item;
    Determine Hot and Cold Disk Enclosures;
    Determine Data Placement;
    Determine Write Delay Applicable Data Item;
    Determine Pre-Load Applicable Data Item;
    Determine the Power Control Method of Disk Enclosures;
    Determine a length of next monitoring period;
end while
```

6.5.2 論理入出力パターンの決定

入出力パターン決定機能は，データアイテムの論理入出力パターンを次のように抽出する．

- **Step1. ロングインターバルの識別**: まず，本機能は論理入出力トレースをデータアイテムごとに分割する．次に，データアイテムごとに分割された論理入出力トレースが Break Even Time より長い入出力発行間隔を含むかどうかをチェックする．もしデータアイテムごとに分割された論理入出力トレースが Break Even Time より長い入出力発行間隔を含む場合，その入出力発行間隔をロングインターバルとして記録する．データアイテムに対して一度も入出力が発行されていない場合，本ステップはモニタリング期間を入出力間隔と見なし，それをロングインターバルとして記録する．この場合，ロングインターバルの長さはモニタリング期間と等しくなる．

- **Step2. 入出力シーケンスの識別**: 次に，本機能はデータアイテムごとに分割された論理入出力トレースから入出力シーケンスを抜き出す．入出力シーケンスは少なくとも一回の入出力と，Break Even Time より短い 0 個以上の入出力発行間隔を持つ．

- **Step3. データアイテムの論理入出力パターンの決定**: データアイテムに対して一度も入出力が発行されていない場合，それらのデータアイテムの入出力パターンを P0 とする．データアイテムがロングインターバルを持たない場合，それらの論理入出

力パターンを P3 とする．残りのデータアイテムについて，本機能はデータアイテムに対する read と write の入出力発行回数をカウントする．もしデータアイテムに対する入出力の半数以上が read 入出力であれば，本機能はそのデータアイテムの論理入出力パターンを P1 と判定し，半数以上が write 入出力であれば P2 と判定する．

6.5.3 Hot 及び Cold ディスク筐体の決定

次に，電力管理機能は Hot ディスク筐体と Cold ディスク筐体を決定する．本機能は，P3 型のデータアイテムに基づき Hot ディスク筐体を選択する．P3 型のデータアイテムは高頻度でアクセスされるため，P3 型データアイテムの入出力性能の劣化はアプリケーションの性能に大きな影響を及ぼす．従って，本機能は以下の条件を満たすディスク筐体を Hot ディスク筐体として選択する．

1. Hot ディスク筐体として選択されたディスク筐体が提供できる IOPS の合計値が，P3 型のデータアイテムの IOPS の合計値より大きい．

2. Hot ディスク筐体として選択されたディスク筐体の容量の合計値が，P3 型のデータアイテムの容量の合計値より大きい．

ディスク筐体間でのデータ移動のコストは高い．P3 型のデータアイテムの Cold ディスク筐体から Hot ディスク筐体への移動量を削減するために，本機能はディスク筐体に格納されている P3 型データアイテムの容量が多いディスク筐体から順に Hot ディスク筐体を選ぶ．Hot 及び Cold ディスク筐体の選択方式を示す:

- **Step1. P3 型データアイテムの合計 IOPS の計算**: 最大 IOPSI_{sum} は次の式により計算する: $I_{sum} = max(\sum_{i=0}^{n-1} \sum_{t=0}^{k-1} I_{it})$．$n$ は P3 型のデータアイテムの個数，k はモニタリング期間 (秒数)，I_{it} は P3 型のデータアイテム i の時刻 t における IOPS である．

- **Step2. Hot ディスク筐体数の計算**: Hot ディスク筐体数 N_{hot} の計算式は以下の通りである: $N_{hot} = max(\lceil I_{sum}/O \rceil, \lceil \sum s_i/S \rceil)$．$O$ は 1 台のディスク筐体が提供可能な IOPS の最大値，S はディスク筐体の容量，s_i は P3 型のデータアイテム i の容量である．本機能は，ディスク筐体が要求された IOPS を満たし，かつ全ての P3 型のデータアイテムを格納できるように N_{hot} を選ぶ．

- **Step3. Hot 及び Cold ディスク筐体の選択**: まず，本機能はディスク筐体を，ディスク筐体内の P3 型データアイテムの容量が多い順にソートする．次に，上位 N_{hot} 個のディスク筐体を Hot ディスク筐体として選ぶ．残りのディスク筐体が Cold ディスク筐体である．もし N_{hot} がディスク筐体の数より多い場合，本機能は全てのディスク筐体を Hot ディスク筐体として選択する．上位 N_{hot} 個のディスク筐体を Hot ディスク筐体として選ぶことにより，本機能は，Cold ディスク筐体から Hot ディスク筐体に移動しなければならないデータ量を削減している．

6.5.4 データ配置の決定

データアイテムの配置を決定するために，本電力制御機能は，P3 型のデータアイテムを配置するためのアルゴリズムと，P0，P1，P2 型のデータアイテムを配置するアルゴリズムをそれぞれ設ける．本機能は，入出力数の多い P3 型データアイテムを，Hot ディスク筐体間の負荷を分散できるように配置するのに対し，省電力効果が期待できる P0，P1，P2 型のデータアイテムは，入出力発行間隔の長いディスク筐体数をできるだけ多く保つように配置する．

Algorithm 3 P3 形データアイテムの配置の決定

$M \leftarrow$*P3 data items in cold disk enclosures*;
Sort elements of M *by IOPS/size in descending order*;
$i = 0$;
for number of elements in M **do**
  $d \leftarrow M[i]$;
  $s \leftarrow$ *hot disk enclosure which average IOPS is minimum*;
  if *d.averageIOPS+s.averageIOPS*$< O$ **and** *d.size+s.usedSize*$< S$ **then**
    Set s as a target disk enclosure of d;
  else if *d.averageIOPS+s.averageIOPS* $>= O$ **then**
    Increase N_{hot} *and retry this algorithm*;
  else if *d.size+s.usedSize* $>= S$ **then**
    $s \leftarrow$ *hot disk enclosure which average IOPS is next minimum and retly*;
  end if
  increment i;
end for

P3 型データアイテムのデータ配置を決めるためのアルゴリズムをアルゴリズム 3 に示す．アルゴリズム 3 では，まず Cold ディスク筐体に配置されている P3 型のデータアイテムの配置を考える．本機能は，Cold ディスク筐体に配置されている P3 型のデータアイテムを，以下の条件を満たす Hot ディスク筐体に配置する: i) Hot ディスク筐体が提供できる IOPS の余力が，当該 Hot ディスク筐体に配置しようとする P3 型データアイテムの最大 IOPS より大きい，ii) #1 の中で，Hot ディスク筐体が提供できる IOPS の余力が最も大きい，及び iii) Hot ディスク筐体の空き容量が，配置しようとする P3 型データアイテムの容量より大きい．

本アルゴリズムは Hot ディスク筐体内の P3 型データアイテムの移動は行わない．もし全ての Hot ディスク筐体が前述の 3 番目の条件のみを満たさない場合，本アルゴリズムは Hot ディスク筐体中の P0，P1，P2 データアイテムを，アルゴリズム 4 を用いて Cold ディスク筐体に移動する．このようにして，Hot ディスク筐体内の空き容量を増やす．条件を満たすディスク筐体が見つからない場合は，N_{hot} を 1 増加し，再度ディスク筐体の選択及びデータ配置の決定を行う．

P0，P1，P2 型のデータアイテムを配置するためのアルゴリズムをアルゴリズム 4 に示す．アルゴリズム 4 は，P3 型のデータアイテムを格納するための十分な容量が Hot ディスク筐体にない場合に起動され，Hot ディスク筐体内の P0，P1，P2 型のデータアイテムの

Algorithm 4 P0, P1, P2 型データアイテムの配置の決定

```
M ←P1 and P2 data items in the hot disk enclosures;
i = 0;
for length of M do
  d ← M[i];
  s ← a cold disk enclosure that its l_max is maximum;
  if d.size < S - s.usedSize and l_max+d.iops < O then
    Set s as a target disk enclosure of d;
  else
    s ← a disk enclosure which l_max is next larger and retry;
  end if
  increment i;
end for
```

Cold ディスク筐体への移動先を決定する．本アルゴリズムは，以下の条件を満たす Cold ディスク筐体を移動先として選ぶ: i) 提供可能な IOPS が，移動しようとする P0，P1，P2 型のデータアイテムの IOPS より大きい，ii) #1 のうち，Cold ディスク筐体内で IOPS が最も多い，及び iii) 空き容量が，移動しようとする P0，P1，P2 型のデータアイテムより大きい．

6.5.5 Write 遅延を適用するデータアイテムの決定

エンタープライズストレージは，データの更新に伴う RAID のパリティ生成に要する時間をアプリケーションから隠ぺいするために，アプリケーションがデータを更新する契機とは非同期にデータをディスク筐体に書き出す．この write 動作は通常ディスク筐体に対する write 入出力を遅延させ，ディスク筐体に対する write 入出力発行間隔を伸ばす．しかし，このストレージの非同期 write 機能はアプリケーションのデータアイテムを識別せず更新されたデータの write を全て遅延する．P3 型のデータアイテムは通常高い頻度で更新されるため，この write 動作は write 遅延用に割当てたキャッシュを多く消費する．従って，P3 型のデータアイテムと Cold ディスク筐体に格納されている P0，P1， P2 型のデータアイテムを同一のキャッシュに保持するエンタープライズストレージ固有の write 動作では，Cold ディスク筐体に対する入出力発行間隔が短くなる．このため，独自の write 遅延機能を導入する．

Write 遅延機能は，Cold ディスク筐体内にある全ての P2 型のデータを write 遅延の対象とする．理由は，P2 型のデータアイテムは入出力数の半数以上が write であり，write を遅延した場合に write 入出力間隔を伸ばせる可能性が高いためである．もしストレージキャッシュに余裕がある場合，本機能は Cold ディスク筐体に格納された P1 型の中で Write 数が多いデータも Write 遅延の対象とする．

6.5.6 プレロードを適用するデータアイテムの決定

エンタープライズストレージは，シーケンシャルに read されるデータをストレージキャッシュにプリフェッチする．しかし，プリフェッチ機能の目的は read 入出力発行間隔の延伸ではなく read 入出力応答時間の短縮である．Read 入出力の発行間隔を伸ばすために，電力管理機能は Cold ディスク筐体内の P1 型データアイテムをプレロードの適用候補とする．本機能は，Cold ディスク筐体内の P1 型のデータアイテムをデータサイズ当りの容量が大きい順にソートする．そして P1 型のデータアイテムの容量がプレロード用に割当てたキャッシュサイズに到達するまで Cold ディスク筐体内の P1 型のデータアイテムを選択し，これらをプレロード対象データとする．

エンタープライズストレージは一つのディスク筐体上に複数個のボリュームを作成し，それぞれにストレージキャッシュを割当てることが可能である．プレロード機能は以下のように実装することが可能である:

1. Cold ディスク筐体上に複数個のボリュームを作成,
2. 個々のボリュームにストレージキャッシュを割当て,
3. プレロード対象データアイテムを他のデータアイテムと異なるボリュームに格納.

プレロードでは P2 データアイテムをストレージキャッシュに一括ロードするため，ストレージキャッシュへのデータの読み込みがアプリケーションの入出力性能に影響を及ぼす可能性がある．しかし，現在のエンタープライズストレージの内部帯域は 100GB/s を超えており，その影響は小さいと考える．

6.5.7 ディスク筐体の電力管理方式の決定

プレロード対象のデータアイテムを決定した後，電力管理機能は Cold ディスク筐体にのみ電源 OFF 機能が適用されるよう，ストレージを設定する．

6.5.8 次回のモニタリングの期間の決定

最後に，電力管理機能は次回のモニタリングの期間の長さを決定する．電力管理機能は，ロングインターバルの平均長に基づき次回のモニタリング期間を決定する．計算式は以下の通りである: $I_{new} = average(I_{cur}) \times \alpha$ I_{cur} は現在のモニタリング期間中に観測されたロングインターバルの長さ，I_{new} は次回のモニタリング期間の長さである．

次回のモニタリング期間を決めるために，1 以上の定数である α を導入する．これは，実際の入出力発行間隔がモニタリング期間より長い場合に，モニタリング期間を伸ばすための定数である．α を導入しない場合，入出力間隔がモニタリング期間より長い場合であってもモニタリング期間の終了毎に電力管理機能が起動され，無駄に CPU を消費する．

6.6 実行時省電力手法

6.6.1 ディスク筐体の電源制御

実行時省電力手法は，電源が OFF となっているディスク筐体上のデータアイテムに read あるいは write を行う必要が生じた場合，そのディスク筐体の電源を ON にする．ディスク筐体の電源を ON とする契機は以下の通りである:

- アプリケーションから read 入出力を受け取ったが，ストレージキャッシュ上に read 入出力が発行されたデータアイテムのブロックが存在しない場合，当該データアイテムを格納しているディスク筐体の電源を ON にする．
- データアイテムをディスク筐体間で移動する場合，データアイテムの移動元及び移動先のディスク筐体の電源を ON にする．
- アプリケーションから write 入出力を受け取ったが，それが write 遅延対象ではない場合，write 入出力が発行されたデータアイテムを格納しているディスク筐体の電源を ON にする．
- データアイテムをキャッシュにプレロードする場合，プレロード対象となるデータアイテムが格納されているディスク筐体の電源を ON にする．
- Write 遅延されていたデータアイテムをディスク筐体に書き出す場合，当該データアイテムが格納されているディスク筐体の電源を ON にする．

また，実行時省電力手法は，ストレージモニタが収集した物理入出力トレースを参照し，一定期間ディスク筐体に対して入出力が発行されなかった場合は，ディスク筐体の電源を OFF にする．

6.6.2 データアイテムの移動

電力管理機能の実行後，実行時省電力手法は 6.5.4 節で決定したデータ配置に基づきディスク筐体間でデータを移動する．実行時省電力手法は，アプリケーションの入出力性能に影響を与えないようにデータ転送のための入出力スループットを調整する．実行時省電力手法は，まず Hot ディスク筐体上の P0，P1，P2 データアイテムを Cold ディスク筐体に移動する．これらのデータ移動は，P3 型データアイテムを移動する Hot ディスク筐体に空きスペースを作成するために必要である．実行時省電力手法は，アルゴリズム 3 及び 4 内のリスト M の順序に従いデータアイテムを移動する．

6.6.3 Write 遅延の制御

Write 遅延では，実行時省電力手法は，ストレージコントローラに対し，write 遅延対象となったデータアイテムの更新されたブロックを，Write 遅延用に割当てたキャッシュに蓄積するよう指示する．そして，更新ブロック比率を設定し，キャッシュの更新されたブロックの比率が更新ブロック比率を超えると，更新されたブロックがまとめてディスク筐

体に書き出されるようにする．更新ブロック比率とは，キャッシュ内の更新ブロック数の最大比率を決定するためのパラメタである．

また，本手法はデータアイテムが write 遅延対象ではなくなった場合に，write 遅延用キャッシュに格納されたこれらデータアイテムのブロックをディスク筐体に反映した後削除するよう，ストレージコントローラに指示する．ディスク筐体へのデータの書き出しにはストレージが持つ機能を用いる．データの書き出し時にディスク筐体の電源が OFF であった場合には，ストレージコントローラに当該ディスク筐体の電源 ON を指示する．

6.6.4 データアイテムのプレロード

プレロードでは，実行時省電力手法は，まずプレロード対象ではなくなったデータアイテムをプレロード用のキャッシュから削除する．その後，プレロード対象であるがまだキャッシュにロードされていないデータアイテムをキャッシュにロードする．これらの動作はモニタリングの開始前に行う．既にキャッシュにロードされているデータはそのまま保持する．

プレロードの実行時に，プレロード対象となるデータアイテムが格納されているディスク筐体の電源が OFF である場合には，本機能は当該ディスク筐体の電源を ON にしプレロードを行う．

6.6.5 入出力挙動の変化への追従

実行時省電力手法は入出力挙動の変化を示す以下の条件のうち少なくとも一つが成立した場合，即座に電力管理機能を実行する:

1. Hot ディスク筐体の入出力間隔が Break Even Time より長くなる，
2. モニタリング開始時刻 t_e と現在時刻 t_c の間の Cold ディスク筐体の電源 ON 回数が m 回を超える．$m = 2 \times (t_c - t_e)/l_b$ である．l_b は Break Even Time の長さである．

前者はさらに消費電力を削減できる可能性に対応するため，後者は無駄な電力の消費を削減するための条件である．電力管理機能を即座に実行することにより，本手法はモニタリング期間中に入出力挙動が変化した場合においても省電力効果の減少を防ぐことができる．

6.7 評価

本節では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，提案手法の評価方法，評価結果，及び提案手法の有効性について述べる．

6.7.1 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

ディスク筐体の消費電力を削減するためには，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を知ることが重要である．そのため，データセンタで稼働するファイルサーバ，OLTP 及び DSS の 3 種類の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を調査した．

実験環境

実験環境は図 4.7 に示す通りである．また，データインテンシブアプリケーションの設定及び負荷は，表 6.1，6.2 及び 6.3 に示したとおりである．

データインテンシブアプリケーションの負荷を生成するにあたり，ファイルサーバでは，Microsoft Research の入出力トレース [23] をストレージ上で再生する．OLTP では，OLTP の代表的ベンチマークである TPC-C を図 4.7 に示す実験環境上で動作させる．DSS では，OLTP と同様 DSS の代表的ベンチマークである TPC-H を図 4.7 に示す実験環境上で動作させる．モニタリング期間はアプリケーションの実行開始から終了までである．

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

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の計測結果を図 6.20 に示す．図に示すように，ファイルサーバでは，89.6%のデータアイテムは P1 型，9.9%が P3 であった．P2 型のデータアイテムはほとんど見られなかった．TPC-C では 76.2%のデータアイテムが P3，23.3%が P2 であり，P1 型のデータアイテムは見られなかった．TPC-H では，TPC-C と異なり 61.5%のデータアイテムは P1，38.5%は P2 であった．P3 型のデータアイテムは見られなかった．本計測では，P0 型のデータアイテムは観測されていないが，これは計測期間中に最低 1 回はデータがアクセスされたためである．図から，ファイルサーバ，OLTP，DSS 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た．この結果，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．

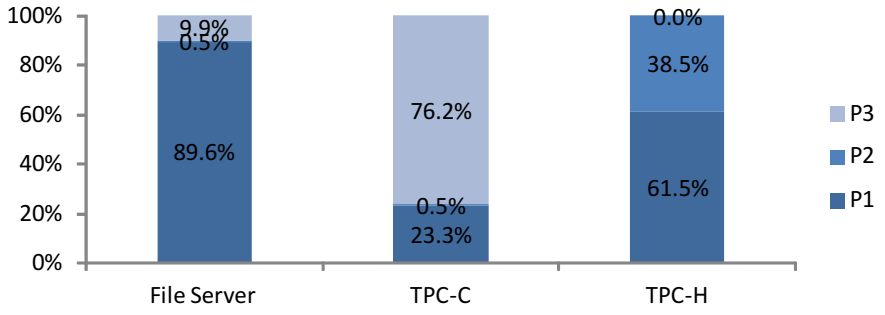

図 6.20: データインテンシブアプリケーションの論理入出力パターン

6.7.2　評価方法

比較対象

本評価では，提案手法を，物理入出力挙動に基づく省電力手法 (Dynamic Data Reorganization (DDR)) と論理入出力挙動に基づく省電力手法 (Popular Data Concentration (PDC)) のそれぞれと比較する．

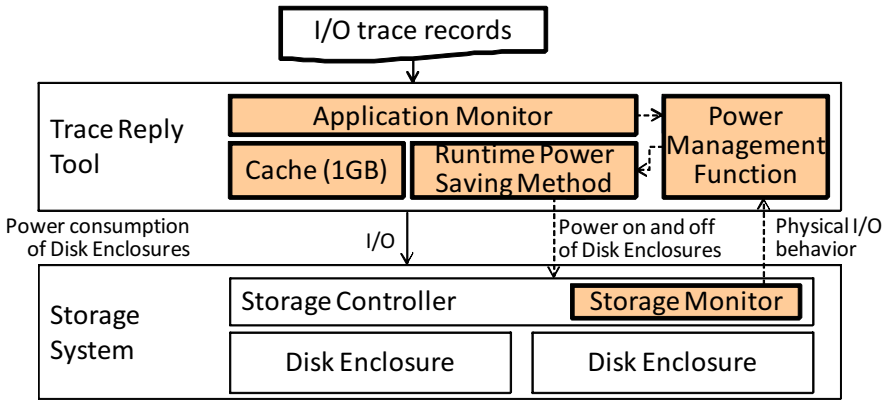

図 6.21: 省電力手法を組み込んだトレース再生ツール

- **Dynamic Data Reorganization (DDR)**[69]: 物理入出力挙動に基づくデータ配置制御を行う手法として，DDR を第一の比較対象とした．DDR は物理入出力挙動に基づきストレージの RAID グループ間でブロック (RAID のストライプ) を移動することによりストレージの省電力を行う手法である．
- **Popular Data Concentration (PDC)**[75]: 論理入出力レベルの情報のみを用いた省電力手法として，PDC を第二の比較対象とした．PDC はアクセス頻度が高いファイルを少数の HDD に集め，他の HDD に省電力機能を適用することにより HDD の省電力を図る．

省電力機能を有するトレース再生ツール

提案手法をトレース再生ツールである blktrace[14] に組み込み，6.2 章で取得した入出力トレースをストレージ上で再生しその消費電力を計測した．本節ではトレース再生ツールの構成について述べる．図 6.21 は提案手法の実装を示すブロック図である．データ移動，プレロード，及び write 遅延されたデータのディスク筐体へのフラッシュを模した入出力も本ツールに実装した．さらに，従来手法との比較を行うために，PDC と DDR 機能を電力管理機能に実装した．

テストベッド

図 4.7 に示した実験環境をテストベッドとして用いる．ストレージに電力計を取り付け，ストレージの実際の消費電力を計測する．

計測項目と計測期間

ストレージの実際の消費電力，入出力応答時間，及び入出力スループットを計測する．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．入出力レスポンスタイムと入出力ス

表 6.4: 評価用パラメタの値

Parameter	Value
Break Even Time	52 sec
Spin down Timeout	52 sec
	(Equal to Break Even Time)
Maximum IOPS of Disk Enclosure	900 (Random 入出力)
	2800 (Sequential 入出力)
Size of Volumes on Disk Enclosure	1.7TB
Storage Cache Size	2GB
Cache Size for Write Delay	500MB
Cache Size for Pre-load	500MB
Dirty Block Rate for Write Delay Cache	50%
Coefficient of Monitoring Period (α)	1.2
Initial Monitoring period for our method	520 sec
Monitoring period for PDC	30 min
TargetTH of DDR	450

ループットはトレース再生ツール内のアプリケーションモニタを用いて計測する．レスポンス及びスループットには，ディスク筐体起動の待ち時間，及びデータ移動，プレロード，write 遅延に伴う入出力の影響も含む．TPC-C と TPC-H では，それぞれトランザクションスループットとクエリレスポンスタイムを計測する．これらの値の計測方法は次節で示す．

トランザクションスループットとクエリレスポンスの計算

TPC-C のトランザクションスループット t の計算式は $t = t_{orig} \times (r/r_{orig})$ である．ここで，t_{orig} は 6.2 節で計測したトランザクションスループットの値，r は省電力機能を適用した場合の平均 read 応答時間，r_{orig} は 6.2 節で計測した read 応答時間の平均値である．

TPC-H のクエリ応答時間 q は次のように計算する: $q = q_{orig} \times (sum(r)/sum(r_{orig}))$. q_{orig} は 6.2 節で計測したクエリ応答時間，$sum(r)$ は省電力機能を用いた場合のクエリ実行時の read 入出力の応答時間の合計値，$sum(r_{orig})$ は 6.2 節で計測した read 入出力応答時間の合計値である．

6.7.3 パラメタ設定

表 6.4 に提案手法，及び比較に用いた各手法のパラメタの設定値を示す．Break Even Time，最大 IOPS，ボリュームの容量，及びストレージキャッシュの容量はテストベッドのストレージの実際の値である．ディスク筐体の電源を OFF にするまでの待ち時間は Break Even Time と同じ値とする．

テストベッドのストレージのキャッシュサイズは 2GB である．トレース再生ツールは，そのうち 500MB がプレロードに，別の 500MB が write 遅延用に割り当てられていることを

認識している．更新ブロック比率は 50%とする．モニタリング期間の初期値は Break Even Time の 10 倍の 520 秒とする．モニタリング期間は，6.4 節に示した方式に従い動的に変更する．PDC のモニタリング期間は 30 分とする．これは文献 [75] で用いられていた値である．他の PDC のパラメタ値も文献 [75] で用いられた値を用いる．DDR の TargetTH は 450 とする．TargetTH とは DDR で用いられる指標であり，ディスク筐体がアプリケーションに要求される応答時間及びスループットを提供できる最大の IOPS である．TargetTH をディスク筐体が提供できる IOPS の最大値の 1/2 とする．DDR のモニタリング期間は文献 [69] に示された値と同一である．

6.7.4　ワークロード

本評価には，6.7.1 で取得したアプリケーションレベルの入出力トレースを用いる．計測期間は表 6.1，6.2 及び 6.3 に示した期間と同じである．

6.7.5　評価結果

ファイルサーバ

ファイルサーバを実行した場合のストレージコントローラとディスク筐体の消費電力を図 6.22 に示す．図に示すように，提案手法はディスク筐体の消費電力を 2977.9W から 2209.2W に約 25.8%削減している．PDC の消費電力は 2873.9W(3.5%減)，DDR は 2869.7W(3.6%減)であった，

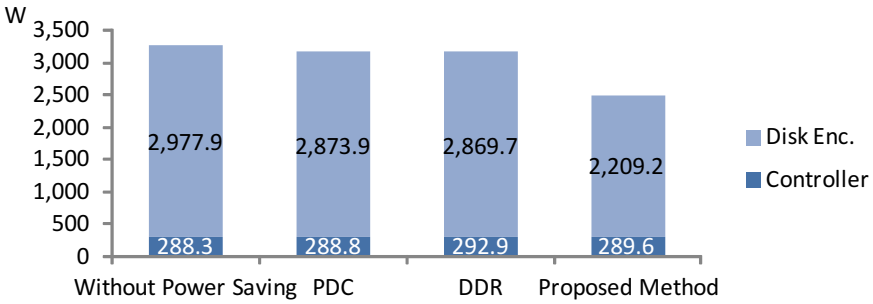

図 6.22: ファイルサーバの消費電力

図 6.23 はアプリケーションモニタで計測した入出力応答時間の平均値である．入出力応答時間には，ディスク筐体の電源を ON にするための待ち時間，データ移動，プレロード，及び write 遅延による入出力応答時間の劣化分も含む．図に示すように，提案手法の入出力応答時間の平均値は 17.1ms である．PDC は 22.6ms，DDR は 27.0ms であった．提案手法の入出力応答時間は，省電力なしの場合より短い．これは，提案手法が P1 型のデータアイテムをプレロードした結果ディスク筐体に対する入出力数が省電力なしの場合と比較して減少したためである．

提案手法はアプリケーションの入出力挙動を利用するため，最悪の場合においてもディスク筐体の消費電力や入出力性能を省電力なしの場合と同等程度に抑えることが可能である．

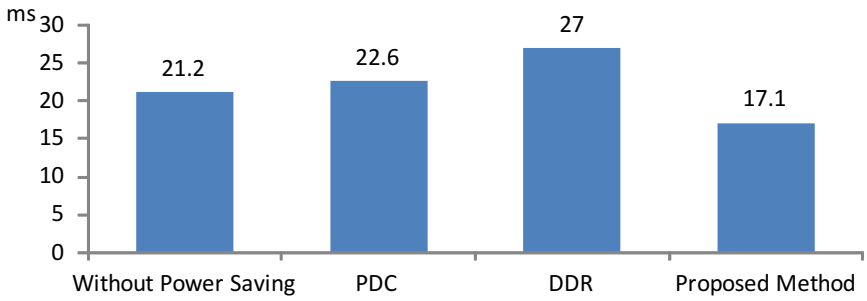

図 6.23: ファイルサーバの入出力応答時間

図 6.24 は評価期間中に移動したデータ量を示している．図に示すように，提案手法のデータ移動量は約 23.1GB であった．提案手法では，P3 型のデータアイテムを Cold ディスク筐体から Hot ディスク筐体に移動するのみであった．これによりデータ移動量を削減できた．PDC のデータ移動量は 3TB を超えている．これは PDC は Hot ディスク筐体と Cold ディスク筐体間のデータ移動のみではなく，Hot ディスク筐体間，及び Cold ディスク筐体間でもデータを移動したためである．DDR のデータ移動量は 1.3GB である．DDR のデータ移動量は他の方式と比較して少ない．これは，全てのディスク筐体の平均 IOPS 数が，DDR が用いるパラメタである LowTH(225, TargetTH の 1/2 の値) より高かったためである．LowTH はディスク筐体が Cold であるか否かを判断するための値である．DDR は Cold ディスク筐体のブロックがアクセスされるとそれを Hot ディスク筐体のブロックと交換する．しかし，本評価では Cold と判断されたディスク筐体がほとんどなくデータの移動量はごく少量となった．

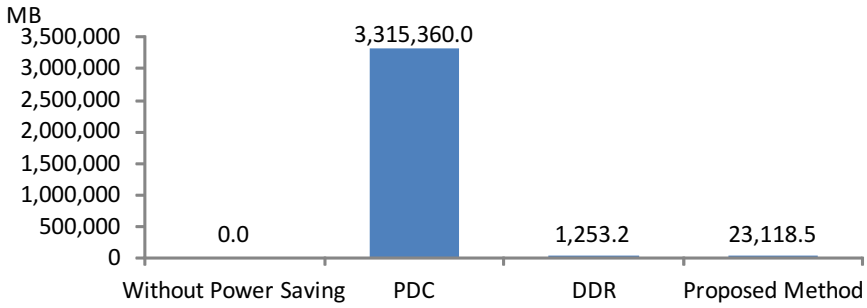

図 6.24: ファイルサーバのデータ移動量

ワークロードの実行時間は約 6 時間である．データ配置の決定回数は PDC が 11 回，DDR は 91,000 回であった．提案手法は 5 回であった．ファイルサーバの入出力トレースは多くのロングインターバルを含み，その傾向は時間が経過してもあまり変わらなかった．これは，モニタリング期間がより長くてもよいことを示している．このため提案手法はモニタ

リング期間をより長く設定した．この結果提案手法はデータ配置の決定に使用する CPU サイクルを削減できた．

TPC-C

図 6.25 は TPC-C を用いた場合のストレージコントローラとディスク筐体の消費電力を示している．図に示すように，提案手法はストレージの消費電力を 2656.4W から 2238.1W に約 15.7%削減している．PDC の消費電力は約 2873.9W(10.7%減)，DDR はディスク筐体の消費電力を削減することはできなかった．

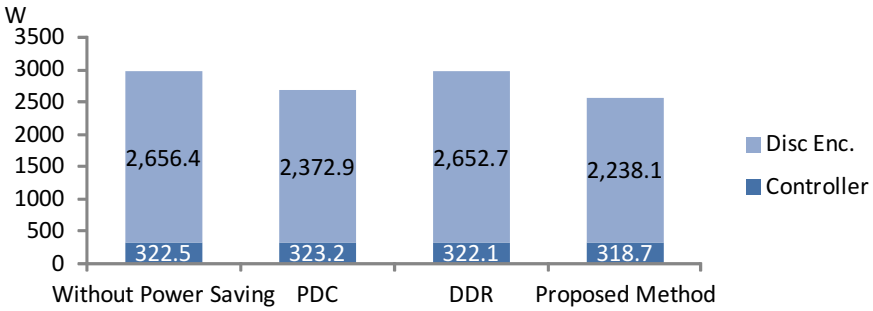

図 6.25: TPC-C の消費電力

図 6.26 は各手法のトランザクションスループットを示している．提案手法のトランザクションスループットは 1701.4tpmC(約 8.5%減) であった．PDC と DDR もトランザクションスループットは減少しているが，その減少率は提案手法より高かった．これは，提案手法はプレロード機能を用いたことにより，read 応答時間が他の手法より短くなったためである．

提案手法はアプリケーションの入出力挙動を利用するため，最悪の場合においてもディスク筐体の消費電力やトランザクションスループットを省電力なしの場合と同等程度に抑えることが可能である．

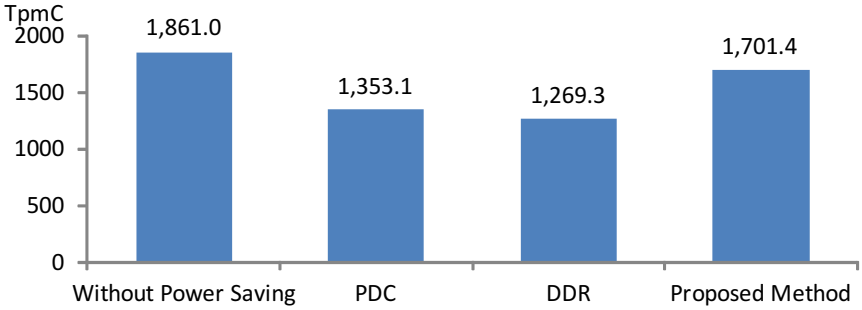

図 6.26: TPC-C のトランザクションスループット

図 6.27 は評価期間中の転送データ量を示している．図に示すように，PDC の移動デー

タ量は 1TB を超えており，DDR が最小である．これは，ファイルサーバの場合と同様の理由による．

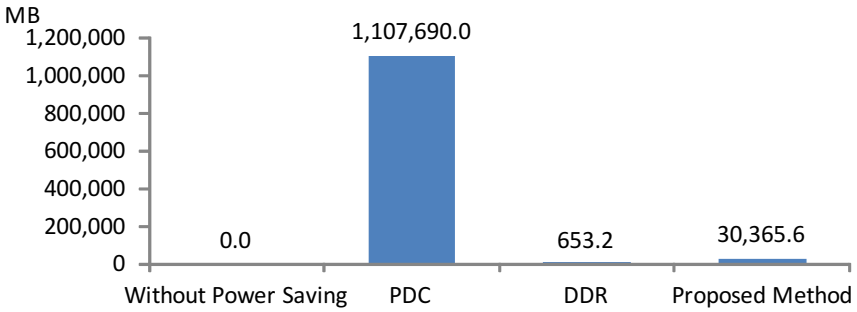

図 6.27: TPC-C のデータ移動量

ワークロードの実行時間は約 1.8 時間である．提案手法のデータ配置の決定回数は 7 回，PDC は 3 回，DDR は約 90,000 回であった．提案手法のデータ配置の決定回数は PDC より多いが，提案手法は Cold ディスク筐体の P3 型のデータアイテムを Hot ディスク筐体に移動するのみであったため移動量を少なく抑えていることが分かる．

TPC-H

図 6.28 は，TPC-H を用いた場合のストレージコントローラとディスク筐体の消費電力を示している．図に示すように，全ての方式が 50%以上消費電力を削減できた．提案手法はストレージの消費電力を 2191.2W から 638.8W に 70.8%以上削減した．PDC の消費電力は 965.2W(約 55.9%減)，DDR の消費電力は 657.9W(約 69.9%減) であった．

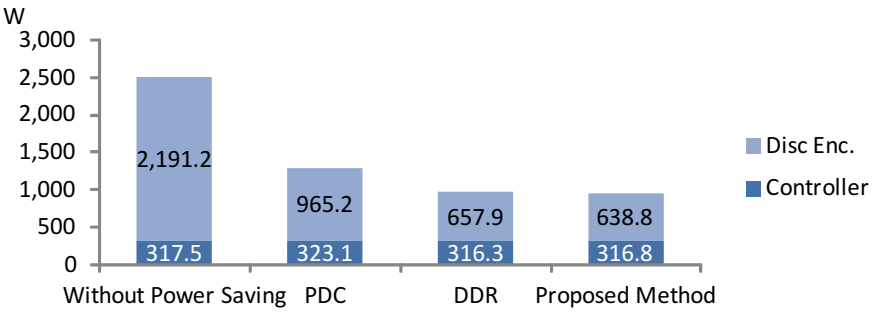

図 6.28: TPC-H の消費電力

図 6.29 はクエリ Q2，7，21 の応答時間を示している．全ての手法でクエリ応答時間は悪化していたが，提案手法のクエリ応答時間は PDC や DDC よりは良い結果であった．これは提案手法はプレロード機能を用いているため，read 入出力の応答時間が PDC 及び DDR より短くなったためである．DDR の応答時間は提案手法の約 3 倍であった．これは，DDR はデータの移動がほとんどなくデータ配置が初期配置からほとんど変化しなかったためで

ある．テストベッドでは，TPC-H のデータは全てのディスク筐体に分散して配置されている．このため DDR では TPC-C 実行時に電源 ON しなければならないディスク筐体数が増加した．

提案手法はアプリケーションの入出力挙動を利用するため，最悪の場合においてもディスク筐体の消費電力やクエリ応答時間を省電力なしの場合と同等程度に抑えることが可能である．

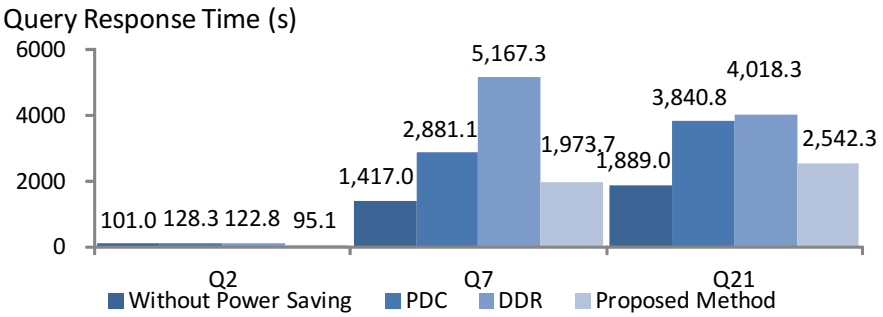

図 6.29: TPC-H のクエリ応答時間

図 6.30 は評価期間中の移動データ量を示している．図に示すように，提案手法と PDC は DDR と比較して多くのデータを移動した．これはこれらの手法がディスク筐体を Hot と Cold に分割し，Cold ディスク筐体内の Hot データを Hot ディスク筐体に移動したためである．DDR の移動データ量はごくわずかである．DDR は Cold ディスク筐体上でアクセスされたブロックを Hot ディスク筐体のブロックと交換する．しかし，テストベッドでは TPC-H のデータは全てのディスク筐体に分散して配置されている．このためクエリ実行中は全てのディスク筐体が Hot(Cold ディスク筐体がない) と判断され，DDR はブロックを移動することができなかった．

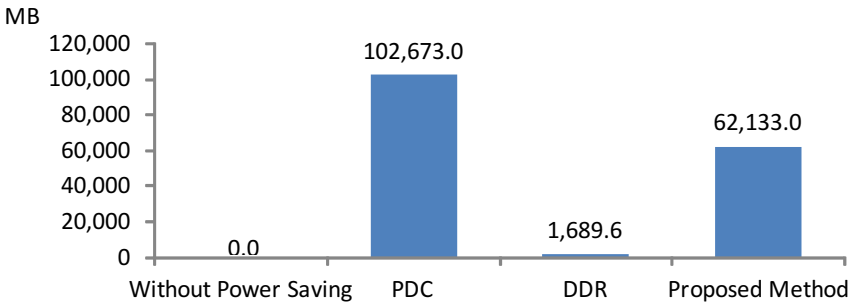

図 6.30: TPC-H のデータ移動量

ワークロードの実行時間は約 6 時間である．提案手法がデータ配置を決定した回数は 10 回，PDC は 8 回，DDR は約 205,000 回であった．提案手法のデータ配置の決定回数は PDC より多いが，データ移動量は PDC より少ない．これは PDC が Hot ディスク筐体間，Cold ディスク筐体間でもデータ移動を行ったに対し，提案手法が P3 型のデータアイテムを Cold ディスク筐体から Hot ディスク筐体に移動したのみであったためである．

6.7.6　分析

本節では，提案手法がなぜアプリケーション性能の大幅な低下を引き起こすことなくディスク筐体の消費電力を削減できたかについて述べる．提案手法は，データアイテムの論理入出力パターンに基づきそれらのディスク筐体への配置を決定する．ディスク筐体の省電力は，ディスク筐体に対する入出力発行間隔の長さに大きな影響を受ける．そのため，提案手法及び比較に用いた手法のそれぞれについて入出力発行間隔の分布を比較した．

図 6.31，6.32 及び 6.33 は，ファイルサーバ，TPC-C 及び TPC-H の Break Even Time 以上の長さの入出力発行間隔とその合計長との関係をそれぞれ示している．X 軸は入出力発行間隔の長さを，Y 軸は入出力発行間隔の累積の長さを示している．

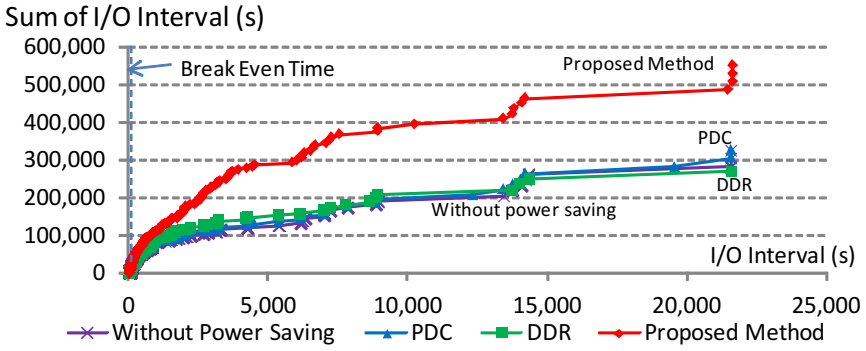

図 6.31: ファイルサーバの入出力発行間隔の分布

図 6.31 に示すように，ファイルサーバでは入出力発行間隔の最大値はどの手法もほぼ同じである．しかし，提案手法の入出力発行間隔の累積値は他の手法の約 2 倍あり，提案手法が他の手法と比較してストレージの消費電力を削減する可能性が高いことを示している．

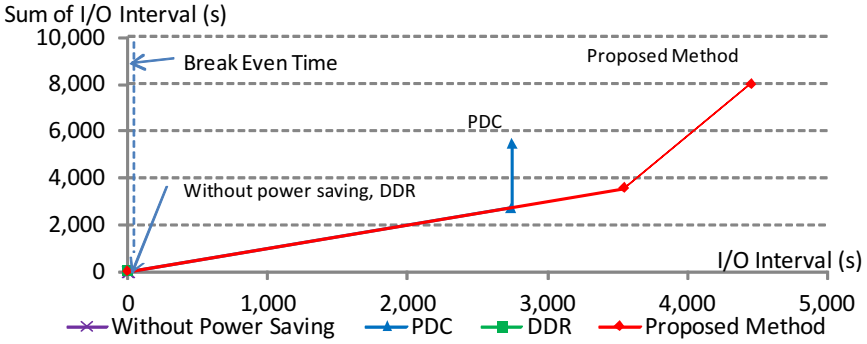

図 6.32: TPC-C の入出力発行間隔の分布

図 6.32 に示すように，TPC-C では提案手法の入出力発行間隔の最大値は PDC 及び DDR と比べて長くなっていることが分かる．DDR では Break Even Time より長い入出力発行間隔は見られなかった．提案手法の入出力発行間隔が他の手法比較して長い理由は，提案手法がプレロード及び write 遅延を用いたことで Cold ディスク筐体の入出力発行間隔を延

伸できたためである．この結果は，高負荷で動作する OLTP であっても提案手法がアプリケーションの入出力挙動を用いることにより高い省電力効果を得られることを示している．

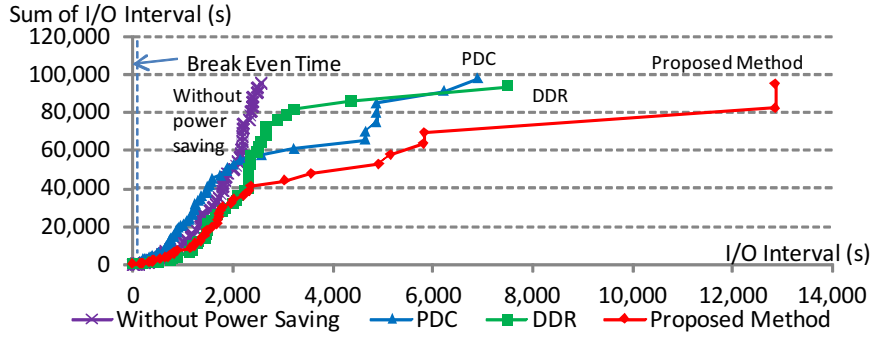

図 6.33: TPC-H の入出力発行間隔の分布

図 6.33 に示すように，TPC-H においては PDC や DDR も入出力発行間隔を伸ばすことが可能であるが，提案手法は PDC や DDR より入出力発行間隔を長く伸ばすことが可能である．理由は TPC-C の場合と同様である．

6.7.7 OLTP アプリケーションを対象としたストレージキャッシュの省電力効果

近年のストレージは，バッテリバックアップされた大容量のキャッシュを搭載している．本節では，このキャッシュを活用して OLTP が稼動するストレージの消費電力をさらに低減する手法について述べる．

本節で提案する手法は，キャッシュを活用して RAID のパリティ生成オーバヘッド (write ペナルティ) を削減し，省電力機能の適用が可能なディスク筐体数を増やすことにより OLTP が稼働するストレージの省電力化を図る．提案手法は，P3 型のデータアイテムの中で，ランダム write 入出力が多いデータアイテムに特に効果を発揮する．

本節では，ストレージキャッシュサイズと RAID パリティ生成オーバヘッドの削減率の関係をシミュレーションにより示す．さらに，シミュレーションによる評価を行い，提案手法の有効性を示す．

OLTP の入出力挙動特性

既に 6.2.2 節にてストレージ上で動作する TPC-C の入出力挙動特性を示した．そして図 6.5，6.6 より，表及び索引を格納したディスク筐体中の HDD に対してコントローラが発行した入出力数は，サーバがコントローラに対して発行した入出力数の約 3 倍であること，及びディスク筐体内の HDD のビジー率は 80%程度と非常に高くなっていることを示した．そして，これらが TPC-C の表・索引に対する入出力にランダム write 入出力が多数含まれていることによる RAID パリティ生成のオーバヘッドのためであることを述べた．

テストベッドのストレージキャッシュ容量は 2GB である．これは入出力挙動の計測に用い TPC-C の DB 容量の 0.2%以下である．一方，現在のストレージは大容量のキャッシュ

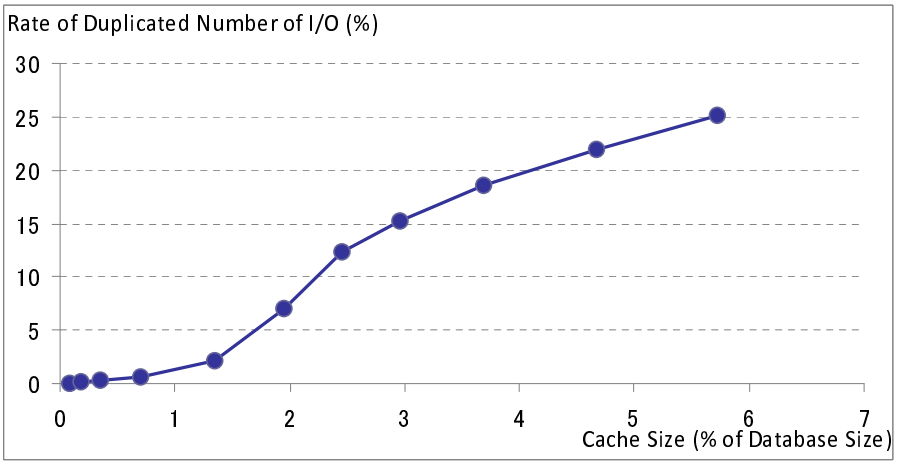

図 6.34: 入出力の重複比率

を有している．この大容量のキャッシュを用いた場合に，TPC-C の HDD に対する入出力挙動がとのように変化するかを調べるために，ストレージのキャッシュ容量を増加させた場合に入出力をどの程度削減できる可能性があるかをシミュレーションにより調査した．シミュレーションに用いた入出力トレースは 6.2.2 節で取得したものを用いた．

図 6.34 は，ストレージキャッシュの DB 容量に対する比率と同一アドレスへの入出力による入出力の重複数の比率を示している．図からわかるように，キャッシュサイズが DB 容量の 1%未満の場合は重複入出力数も 1%未満である．しかし，ストレージキャッシュのサイズが DB 容量の 1%より大きくなると，入出力数の重複率は急激に高くなり，DB 容量のわずか 5%のストレージキャッシュを用いるだけでディスク筐体内の HDD に対する入出力数を 20%以上削減できることが分かる．

ストレージキャッシュを利用した TPC-C アプリケーションの省電力

提案手法は，ディスク筐体内の HDD に対する TPC-C の入出力の局所性を用いてストレージの消費電力を削減する．提案手法の主たるアイデアは，ディスク筐体内の HDD の同一ブロックに対する write 入出力を 1 回の write 入出力にまとめることにより write ペナルティを削減し，DB データを初期配置より少数のディスク筐体に集約し，他のディスク筐体を省電力状態に移行する．ディスク筐体内の HDD への入出力数を削減するための 2 種類の単純なアプローチを提案する．

適切なサイズのストレージキャッシュの割当て ：ストレージキャッシュサイズが DB 容量の 1%より小さい場合，TPC-C ではディスク筐体内の HDD の同一ブロックへの write 入出力はほとんどないこと，及び一方，TPC-C DB 容量の数%のキャッシュ容量を用いることで，HDD に対する入出力数を削減できることを述べた．第一の提案は，TPC-C 用に適切なサイズのキャッシュの割り当てによる write ペナルティの削減である．テストベッドにおけるストレージのキャッシュサイズは 2GB であるが，近年のストレージの最大キャッシュ容量は数百 GB である．提案手法は，ストレージキャッ

シュを LRU キャッシュ置き換えアルゴリズムを使用する DBMS の入出力バッファとして用いる．これによりディスク筐体内の HDD のビジー率を低下させ，高アクセス頻度のデータを少数のディスク筐体に移動する．そして，低負荷となったディスク筐体の電源を OFF にすることにより，ストレージの消費電力を削減する．

ストレージキャッシュの write 遅延：第二のアプローチはこれまでにも述べた write 遅延の適用である．ディスク筐体内の HDD に対する入出力の大部分は write ペナルティである．ストレージキャッシュ内に一定量の write 入出力を蓄積することにより write 入出力の重複を排除し，HDD に対する write ペナルティを削減する．ストレージ内に蓄積する write 入出力の数は，RAID のパリティ生成に必要な領域，及び read 入出力のキャッシュ用の領域を残してできるだけ多く確保する．

キャッシュ割り当ての効果

提案手法を評価するため，まず提案手法を用いることによりどの程度 HDD に対する入出力数を削減できるかをシミュレーションにより計算した．その後，この計算結果に基づきストレージの消費電力を予測した．評価に用いたトレースは 6.2.2 節で取得したものである．また，評価環境は図 4.7 に示した構成を用いた．

ストレージキャッシュ割当ての効果を確認するために，ストレージキャッシュのサイズを DB サイズの 0.4%，1.0%，3.0%，5.0%，10.0%，及び 20%とした場合の入出力数の削減率をシミュレーションにより評価した．さらに，キャッシュの更新ブロック比率を 1%，10%，25%，75%，及び 95%に変化させた場合の入出力数の削減率も評価した．ストレージキャッシュ内の更新されたブロックの HDD へのフラッシュは更新されたブロックの比率が更新ブロック比率を超えた場合に行われる設定とした．HDD に対する入出力は，次の契機で発行するとした．

- サーバから read 入出力が発行され，要求されたブロックがキャッシュに存在しない場合．この場合，HDD に read 入出力を発行する．
- 更新ブロック数が更新ブロック比率を超え，更新されたブロックを HDD にフラッシュする場合．以下の場合に HDD に入出力を発行する．
  - write がランダム write であり，旧パリティブロックがキャッシュ中にない場合，旧パリティブロックをキャッシュに read する．
  - write がランダム write であり，旧データブロックがキャッシュ中にない場合，旧データブロックをキャッシュに read する．
  - write がランダム write であり，新パリティブロック及び新データブロックを HDD に write する．
  - 更新されたブロックが 1 ストライプ分揃っている場合は，1 ストライプ分の新データ及びパリティを HDD に wirte する．

同一ブロックに対して複数回 write 入出力が発行された場合，それらはまとめて 1 回の write 入出力が発行されるとした．

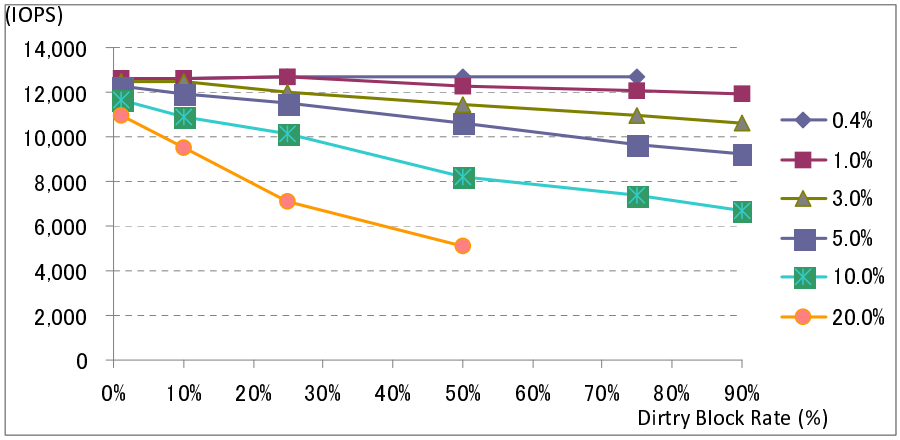

図 6.35: ディスク筐体内の HDD に対する入出力数の変化

図 6.35 は，ストレージキャッシュサイズと更新ブロック比率を変化させた場合の入出力数を示している．図から分かるように，ストレージキャッシュサイズが DB サイズの 0.4%の場合は更新ブロック比率を増やしてもほとんど入出力数を削減できていないことが分かる．一方，ストレージキャッシュサイズを DB サイズの 3%，更新ブロック比率を 75%とした場合，入出力数の削減率は 13%，ストレージキャッシュサイズが DB サイズの 5%の場合は 23%以上入出力数を削減できることが分かる．キャッシュサイズが DB サイズの 20%の場合は約 60% 入出力数を削減できている．この結果は，DB サイズの数%のキャッシュを用いるだけで，同等の入出力性能を維持するために必要なディスク筐体数を減らせる可能性があることを示している．

ディスク筐体の集約と省電力効果

ディスク筐体の集約 図 6.35 に示した結果に基づき，更新ページ比率を 75%とした場合の，キャッシュサイズとディスク筐体数を計算した結果を表 6.5 に示す (ストレージキャッシュサイズが DB サイズの 20%の場合は，更新ページ比率は 50%である)．1 ディスク筐体の最大入出力数は 1,430 IOPS である．表から分かるように，キャッシュサイズを大きくするにつれ，初期状態と同一の入出力性能を達成するために必要なディスク筐体数を大きく削減できていることが分かる．

ストレージ消費電力 次にストレージの消費電力をシミュレーションにより計算した．キャッシュ割当ての効果と write 遅延の効果を別々に確認するために，キャッシュ割当てのみを使用した場合と，キャッシュ割当てと write 遅延を併用した場合のそれぞれについて消費電力を計算した．この結果を図 6.36 に示す．

表 6.5: ストレージキャッシュサイズとディスク筐体数

Storage Cache Size (%)	Number of 入出力 s to be Served (IOPS)	Number of RAID Groups
0.4	12,657	9
1.0	12,065	9
3.0	10,954	8
5.0	9,690	7
10.0	7,365	6
20.0	5,120	4

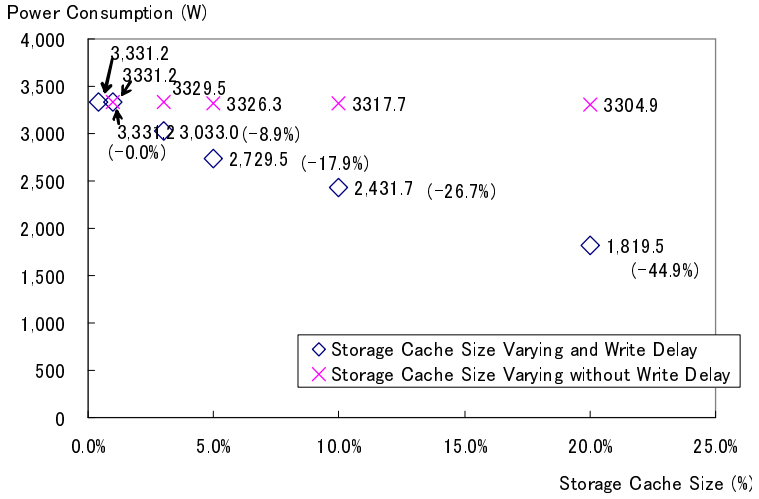

図 6.36: ストレージキャッシュサイズを変えた場合の消費電力

図から分かるように，write 遅延を使用しない場合，キャッシュサイズを大きくしても省電力効果はほとんどないことが分かる．これは，ストレージキャッシュサイズを大きくするのみで更新ブロック比率を小さいままとした場合，write 入出力の重複の排除がほとんどなされないためである．

しかし，write 遅延と併用することで，ストレージキャッシュサイズが DB サイズの 3.0%の場合で 8.9%，5.0%の場合で 17.9%，20.0%の場合で約 45%と消費電力を大きく削減できることが分かる．1TB の DB の場合，キャッシュサイズ 5%は約 50GB である．ストレージの最大キャッシュ容量は数百 GB であり，キャッシュサイズを大きくしても現在のストレージはこの構成をとることは可能である．

トランザクションスループット トランザクションスループットは TPC-C における重要な要素である．そこで，提案手法を適用した場合の TPC-C の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計算した．計算に当たって，ディスク筐体当りの入出力数がディスク筐体の入出力数を超えない場合は入出力応答時間は変化しない

と仮定した．図 6.37 にキャッシュサイズを変更した場合のトランザクションスループットの推移を示す．

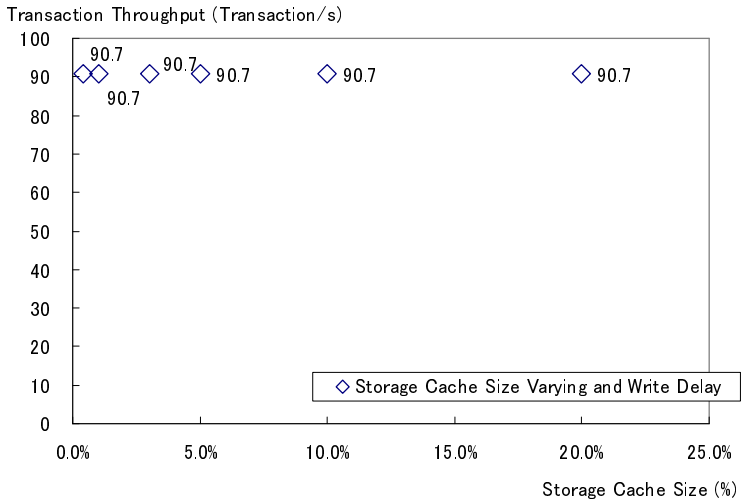

図 6.37: ストレージキャッシュサイズを変えた場合のトランザクションスループット

図から分かるように，提案手法を採用してもトランザクションスループットは減少していない．提案手法はキャッシュサイズの拡大と write 遅延の併用によりディスク筐体に対する入出力数を削減する．ディスク筐体の最大 IOPS を超えないように TPC-C の表・索引データを集約した．このためディスク筐体の最大 IOPS を超えることはなく，トランザクションスループットも低下しなかった．

6.8　実行時ストレージ省電力管理機構の実装と評価

本節では，これまで我々が提案してきた実行時ストレージ省電力管理機構の実装，及び TPC-C，TPC-H を用いた評価結果を示す．

6.8.1　実行時省電力ストレージ管理機構の設計

省電力管理機構は以下の要件を満たす必要がある．

データ投入支援　データの規模が大きくなると，データの移動には数日から数週間を要する場合がある．データの移動はストレージリソースを消費しアプリケーションの I/O 性能に影響を与える．このため，データ投入後に頻繁にデータを移動することは望ましくない．本機構には，データ投入後のデータ移動を極力削減することが求められる．

軽量 I/O 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，I/O 発行間隔

に関する情報，特にデータ毎の Break Even Time 以上の I/O 発行間隔の有無を軽量に取得できなければならない．

電源 ON 要求の軽量化 I/O を行おうとしたディスク筐体の電源が OFF であった場合，当該ディスク筐体の電源を ON にした後 I/O を行う必要がある．しかし，1 I/O 毎にストレージ電源制御機能 (状態の取得も含む) を用いて電源状態を取得していたのでは，I/O 性能に多大な影響を及ぼす．ストレージ省電力機構では，この問い合わせの軽量化が必要である．

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 I/O 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．

6.8.2 省電力ストレージ管理機構の実装

前節の要件に基づき，省電力管理機構の実装を図 6.38 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．

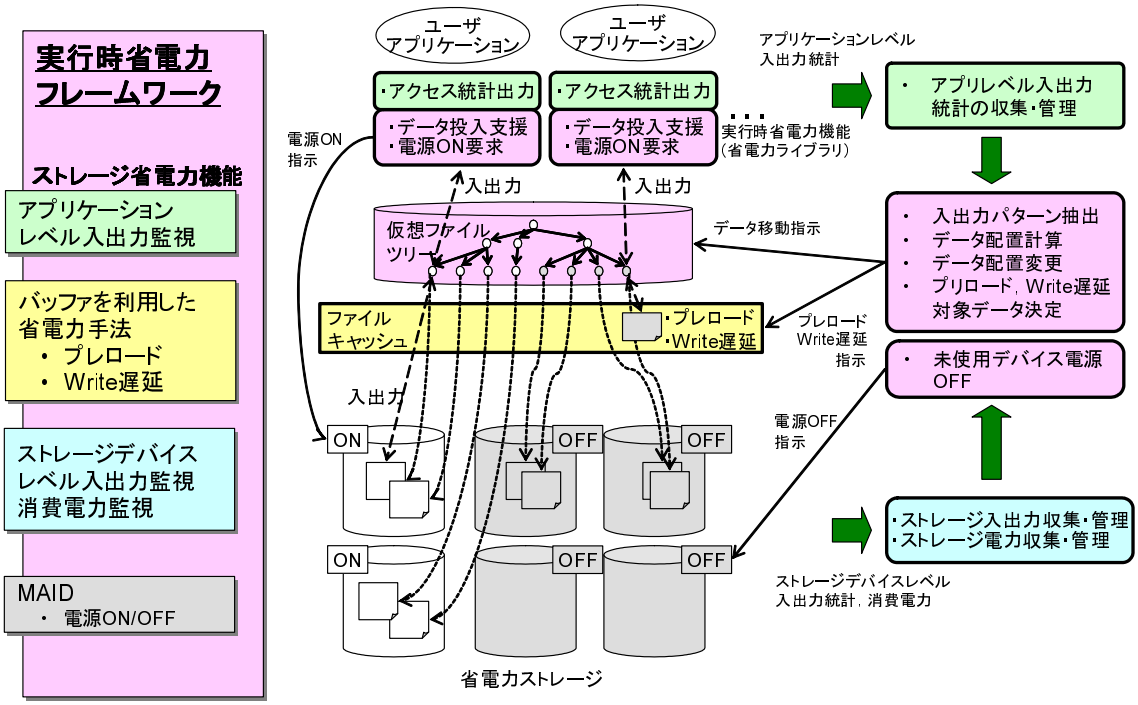

図 6.38: 省電力管理機構の実装

省電力管理機構は，モニタリング機構，省電力管理機能，実行時省電力機能を持つ．モニタリング機構は，論理入出力統計，及びストレージ性能・消費電力を収集・管理する．省電力管理機能は，データ要件管理，データ階層・ストレージ階層構築，ファイル配置計算，及びファイル配置変更機能を持つ．実行時省電力機能は，ユーザアプリケーション実

行時にアプリケーションに動的にリンクされるストレージ省電力ライブラリ，仮想ファイルツリーファイルキャッシュ，及びストレージ電源制御から構成される．以下，これらの実現方式について説明する．

省電力管理

本節では，省電力管理機構の実装について述べる．

ファイル配置計算 データの要件を参照し，データ階層，及びストレージ階層を決定する．さらに，データを配置するディスク筐体，及び配置方法 (分散するか否か) を決定する．決定したデータ配置は，リポジトリに格納する．

ファイル配置計算 データアクセス統計を参照し，プレロード及び write 遅延対象データを決定する．また，データ階層及びストレージ階層へのデータの配置を見直す．

ファイル配置変更 データを配置するディスク筐体を変更する．また，プレロード対象のデータを，ファイルキャッシュに読み込む．さらに，ストレージ省電力ライブラリに write 遅延データ指示する．

実行時省電力

次に，実行時省電力管理機能について述べる．

ストレージ省電力ライブラリ ストレージ省電力ライブラリは，データ投入支援，並列入出力，アクセス統計出力及び電源 ON 要求機能を有する．

データ投入支援 データ投入支援は，新規データ投入時にそれらのデータの配置予定ディスク筐体をリポジトリから取得し，それに基づきデータをディスク筐体に配置する．

並列入出力 I/O 先のデータが複数のディスク筐体に配置されている場合はマルチスレッド機構を用いてそれらを並列に read する．

アクセス統計出力 低レベルの I/O をフックし I/O 統計を共有メモリに出力する．この情報はモニタリング機構により収集される．

並列入出力 I/O が入出力エラーとなった場合に，I/O 先のディスク筐体が電源 OFF 状態であると判断し，ストレージ電源制御機能に当該ディスク筐体を Spin up するよう依頼する．

本ライブラリは，アプリケーションの実行時に動的にアプリケーションにリンクされる．

仮想ファイルツリー 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．

ファイルキャッシュ プリロード対象となったデータのファイルキャッシュへのプリロード，及び Write 遅延対象となったデータの write 遅延を行う．

ストレージ電源制御 ストレージ電源 ON/OFF コマンドが多重に実行されないよう制御する．また，デバイス I/O を監視し，一定時間 I/O が行われていないディスク筐体の電源を OFF にする．

6.8.3 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．また，TPC-C についてトランザクションスループットと消費電力の関係を示し，TPC-C が稼動するストレージの Energy Proportion について考察する．

評価環境

評価に用いたハードウェア構成は，第 4 章の図 4.7 に示した環境と，ディスク筐体数が 1 台多いことを除いて同一である．

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 6.6 に示す．TPC-C のデータベースサイズは約 320GB (Warehouse 数 1000)，DBMS のバッファサイズは 15GB，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最もスループットが高かった値 160 とした．ログ及び作業表を図 4.7 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 4.7 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．

表 6.6: TPC-C 及び TPC-H の設定

Application	Data Size	Workload	Cache Size
TPC-C	320GB	# 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)

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．

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 機能のみを用いた場合 (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 6.39 及び 6.40 にそれぞれ示す．

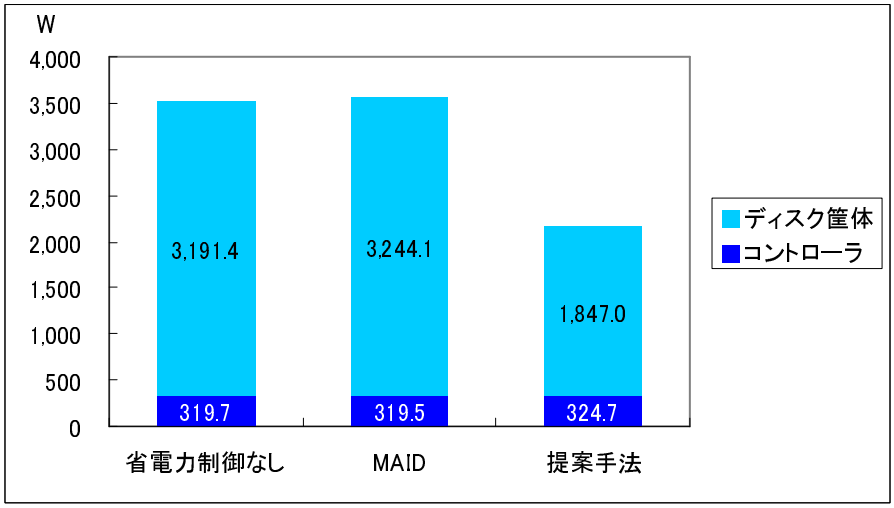

図 6.39: TPC-C が稼働するストレージの平均消費電力

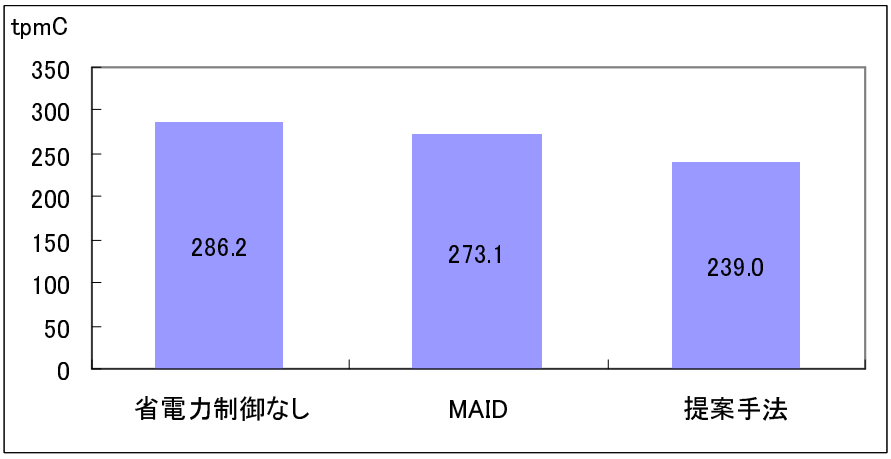

図 6.40: TPC-C のトランザクションスループット

図から分かるように，提案手法はディスク筐体の平均消費電力を 3191.4 W から 1847.0W に約 42%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．MAID の消費電力がほとんど削減できていない理由は，全てのディスク筐体に対して常時入出力が行われ，省電力を行う機会がなかったためである．またトランザクションスループットは省電力制御なしと比較して MAID が 4.6%減，提案手法は 16.5%減であった．

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) と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 6.41 及び 6.42 にそれぞれ示す．

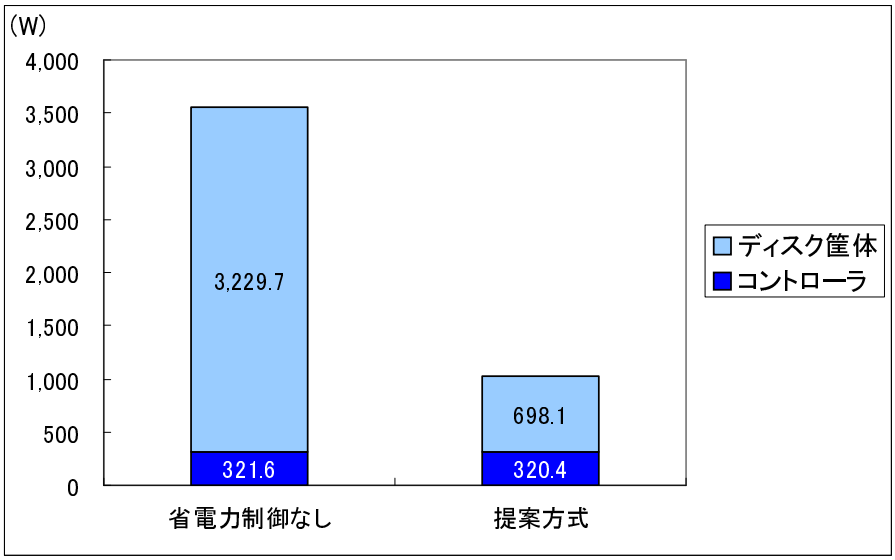

図 6.41: TPC-H が稼働するストレージの平均消費電力

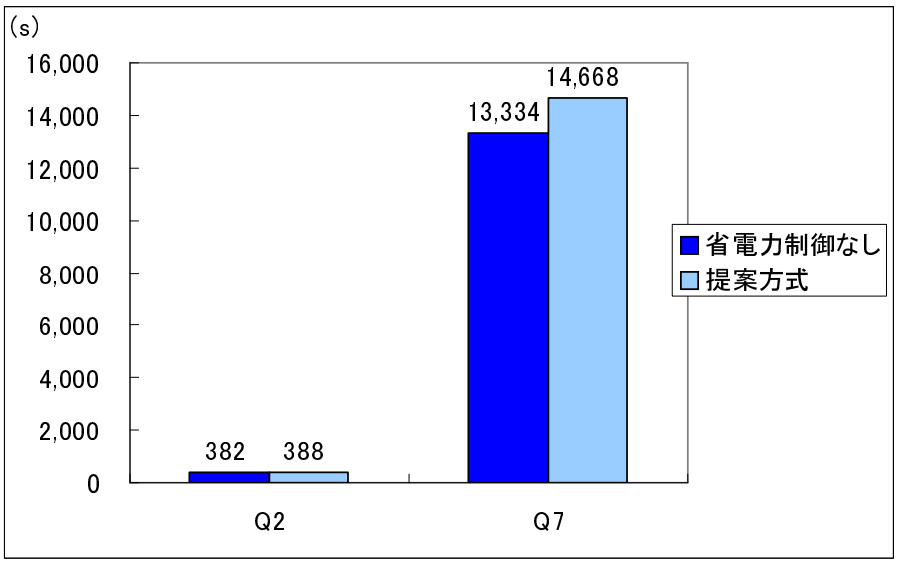

図 6.42: TPC-H のクエリ Q2, Q7 の応答時間

図から分かるように，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．またクエリ応答時間は Q2 はほぼ同等，Q7 が約 9.1%倍増加した．Q7 の応答時間が増加したのは，クエリ実行中のディスク筐体の起動待ち (約 120 秒), 及び入出力が 2 台のディスク筐体の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．

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．

TPC-C が稼動するストレージの Energy Proportion

続いて，TPC-C が稼動するストレージの Energy Proportion 性を明らかにするために，TPC-C のトランザクションスループットと消費電力の関係を調査した．調査結果を 6.43 に示す．

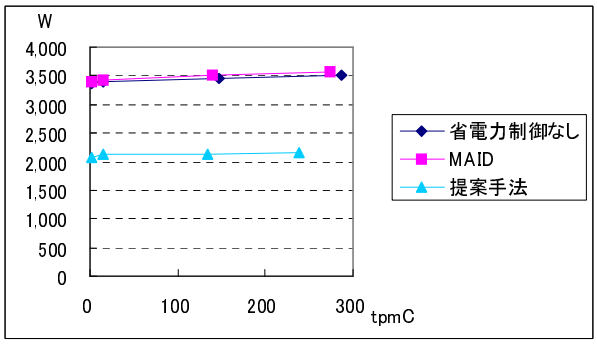

図 6.43: TPC-C のスループットとストレージ消費電力の関係

図からわかるように，省電力制御なし，MAID，提案手法とも，ストレージ消費電力はトランザクションスループットによらずほぼ一定である．これは，TPC-C ではトランザクション数が少ない場合でも入出力が発生し，Break Even Time より長い入出力間隔が発生しないためである．このことは，TPC-C などデータセンタで稼動する多くのアプリケーションではストレージ Energy Proportion 性がほとんどないことを意味している．

6.9 まとめ

本章では，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．本フレームワークは，ストレージレベルの入出力挙動とアプリケーションレベルの入出力挙動を結びつける．次に，ストレージレベルの入出力挙動と結び付けられたアプリケーションレベルの入出力挙動を 4 つの論理入出力パターンに分類し，論理入出力パターンに適した省電力手法を選択する．これによりアプリケーションの入出力挙動がストレージ省電力に利用することが可能となる．

次に，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．また，従来のストレージ省電力手法である Popular Data Concentration (PDC) 及び Dynamic Data Reorganization (DDR) と提案手法の比較を行い，提案手法が従来手法と比較してアプリケーションの性能を大きく落とすことなくストレージの消費電力を大幅に削減できることを示した．また，提案手法はアプリケーションの入出力挙動を利用するため，最悪の場合においても消費電力やスループットを省電力なしの場合と同等程度に抑えることが可能である．これらは，アプリケーション協調型の省電力手法がデータセンタの省電力に大きく貢献できることを示している．

さらに，ストレージキャッシュの割当て制御と write 遅延の併用により，消費電力の削減が困難である P3 型のデータアイテムのうち TPC-C のようにランダム write を多く発行するデータアイテムの省電力が可能であることを示した．

また，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，トレース再生によるシミュレーションと同等以上の省電力効果を得られることを確認した．これに加え TPC-C を用いてデータセンタで稼動するアプリケーションの Energy Proportion 性について調査を行い，Energy Proportion 性がないことを確認した．

近年のストレージは様々な RAID 構成を取ることが可能である．次章では RAID 構成の観点からのストレージ省電力について論じる．

第7章　大規模ストレージシステムにおける省電力を考慮したRAID構成

7.1　RAID構成とストレージ省電力

データセンタで用いられるストレージは数百台から数千台のドライブ (HDD や Solid State Disk (SSD)) を搭載し，RAID レベルやドライブ数が異なる複数の RAID を構成することができる [9]．現在のストレージでは，性能や容量効率，信頼性などの観点から，数十台のドライブを用いて RAID 5 や 6 構成を取ることが一般的に行われている．ストレージの省電力制御は，ディスク筐体単位で行われるが，RAID を構成するドライブ (以下 RAID グループと呼ぶ) の台数が増加すると，RAID グループは 1 台以上のディスク筐体に跨るようになる．RAID グループが複数のディスク筐体に跨った場合，その一部のディスク筐体のみを省電力状態に移行することはできないため，電源制御の単位が大きくなる．この結果，省電力の効率が悪化すると考えられる．ストレージの省電力を実現するためには，アプリケーションの入出力挙動に合わせた RAID レベルやドライブ数を選択する必要があると考えられる．

そこで，本章では，ストレージの省電力の観点から，異なる RAID 構成の効果を調査する．ここで，RAID 構成とは，RAID を構成するドライブ数，RAID レベル，及びドライブのメディア種別 (HDD, SSD) である．

まず，RAID グループを構成するドライブ数について検討する．4.2.2 節において示したように，ディスク筐体の電源を ON にするためには，1 分以上を要する．このため数十台のドライブから構成される RAID グループの省電力のオーバヘッドは極めて高いことが予想できる．適切なドライブ数を選択することにより，ストレージのエネルギー効率を高めることが可能になると考えられる．

次に，RAID レベルについて検討する．性能と容量効率の点から，通常 RAID 5(以下，RAID 6 も含む) が多く用いられている．しかし，RAID 5 はデータの入出力の際には RAID を構成する全てのドライブがアクセスされるため，省電力の観点からは望ましくない可能性がある．RAID 4 や 0+1 では，データの read の場合は冗長データを格納したドライブにアクセスする必要はなく，これらのドライブを省電力状態に移行することができる．また，RAID グループを省電力状態に移行すると，RAID グループ内のドライブは全て電源 OFF 状態となる．一方，HDD 自身も省電力機能を持っており，HDD の起動に要する時間は数秒である．そこで，RAID グループ単位の省電力機能が適用するには不十分な長さの入出力発行間隔に対して HDD 単位の省電力機能の適用も考える．

第三は，ドライブのメディア種別である．今日，Solid State Disks (SSD) が高性能，省電力デバイスとしてストレージで広く用いられるようになってきている．しかし，ストレージの実行時省電力の観点から SSD を用いることの省電力効果について述べた研究はほとん

どない．SSD は HDD のヘッドやアームなどに相当する機械動作部がなく，その電力や性能特性は HDD とは大きく異なる．このため，SSD から構成される RAID グループと HDD から構成される RAID グループにより，アプリケーション毎の省電力効果が異なると考えられる．

本章では，データインテンシブアプリケーションである TPC-C と TPC-H の入出力トレースを用いて，RAID 構成を変化させた場合の消費電力と性能をシミュレーションにより計算する．そして，アプリケーションの入出力挙動特性の観点から，RAID 構成がストレージの省電力に大きな影響を与えることを述べる．ストレージの省電力の観点からの RAID 構成の検討は，本論文で提案した実行時省電力フレームワークの最下層に位置付けられる (図 7.1).

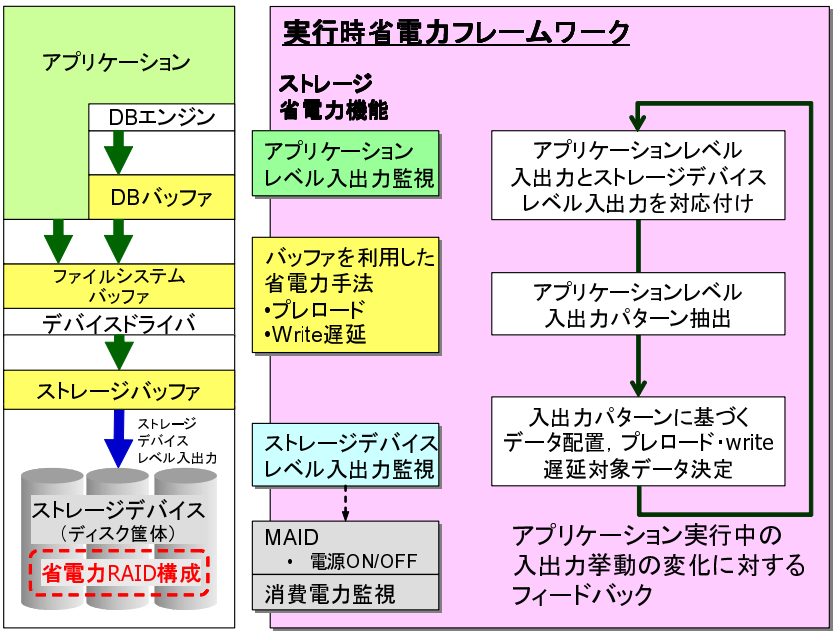

図 7.1: RAID 構成と実行時省電力フレームワーク

7.2 SSD の消費電力特性と Break Even Time

ドライブのメディア種別の省電力効果に対する影響を調べるに当り，SSD の消費電力特性の調査と Break Even Time の計算を行った．HDD 及び HDD から構成される RAID グループの電力特性は 4.2 節及び 4.2.2 節でそれぞれ述べた通りである．また，Break Even Time については 4.3 節で述べた通りである．

7.2.1 SSD の消費電力特性

表 7.1 に，Intel 社の SATA SSD ドライブ (X25-E Extreme)[46] の消費電力特性を示す．Active 時の消費電力は，IOMeter[1] を用いて queue depth 1, 64KB シーケンシャル write を実行した時の値である．

表 7.1: SSD の消費電力特性

Status	Power Consumption
Active	2.4 W
Idle (DIPM)	0.06 W

表から分かるように，Active 時の SSD の消費電力は 2.4W，Idle 時の消費電力は 0.06W である．Active 時の SSD の消費電力は HDD の消費電力の約 1/6，Idle 時は 1/25 と非常に少ないことが分かる．

7.2.2 SSD の Break Even Time

SSD，及び SSD ドライブ 15 台を用いたディスク筐体の Break Even Time と Spin up 待ち時間の計算値を表 7.2 に示す．SSD の Active 時の消費電力は HDD の約 1/6 である．そのため SSD をドライブとして用いるディスク筐体の電源部分の消費電力及び起動時間を，HDD をドライブとして用いるディスク筐体の電源部分の消費電力及び起動時間の 1/6 と想定した．なお，HDD 及び HDD を用いたディスク筐体の Spin up 待ち時間及び Break Even Time は，4.3 節に示した通りである．

表 7.2: SSD の Spin up 待ち時間と Break Even Time

Type	Power Saving Method	Spin Up Wait Time	Break Even Time
SSD	Idle	1 s	1.0 s
SSD RAID	Spin down	2 s	3.7 s
Group	Power off	9 s	5.3 s

表から分かるように，SSD の起動待ち時間は 1 秒，Break Even Time は 1.0 秒である．また，SSD を用いた RAID グループを Spin down した場合の Spin up 待ち時間は 2 秒，Break Even Time は 3.7 秒，電源 OFF した場合の Spin up 待ち時間は 9 秒，Brak Even Time は 5.3 秒であった．これらの結果から，SSD を用いた場合，Spin up 待ち時間及び Break Even TIme は HDD のそれと比較して非常に短いことが分かる．

7.3 RAID グループの省電力の可能性

本節では，RAID グループ内のドライブ数，RAID レベル，ドライブ単位の省電力機能とディスク筐体単位の省電力機能の併用，及び SSD の省電力の可能性について議論する．

7.3.1 RAID グループ内のドライブ数

今日，RAID グループは何十台ものドライブから構成される．そのような大規模な RAID グループは，何十台ものドライブを同時に電源 ON，OFF するため，省電力のオーバヘッドが高いと考えられる．逆に，数台のドライブから構成される RAID グループは，電源 ON，OFF のオーバヘッドが低いため，大規模な RAID グループよりも省電力効果が高いと考えられる．

7.3.2 RAID レベル

RAID グループを構成するドライブへの入出力の傾向は RAID レベルによって大きく異なる．例えば RAID 4 や 0+1 では，いくつかのドライブは冗長データのみを格納する．これらのドライブはデータの read の間は入出力が行われないため，これらドライブの存在は，RAID グループの省電力の可能性をさらに高めるものとなる．RAID レベルごとに，次のような実行時省電力の可能性がある:

RAID 5: RAID 5 は通常良く用いられる RAID レベルである．RAID 5 では，データとパリティは全てのドライブに均一に分散される．そのためデータの read 及び write の間は全てのドライブがアクセスされることになる．従って省電力の効果は低い．

RAID 4: RAID 4 では，冗長データは単一のパリティドライブに配置される．データの read 時はパリティドライブにアクセスする必要はないため，read only のワークロードではパリティドライブの電源を OFF にすることができる．Write の場合は全てのドライブがアクセスされるため，RAID 5 と同様省電力の可能性は低い．

RAID 0+1: RAID 0+1 では，ミラーリングによりデータの冗長性を高める．ワークロードの負荷が低くかつ read only の場合は，ミラーデータを格納したミラーディスクにアクセスする必要はない．このためミラーディスクの電源を OFF にすることにより省電力効果を高めることが可能となる．さらに，RAID 0+1 では冗長性を確保するために必要となる入出力の数が RAID 5 や 4 と比較して少ない．そのため RAID 5 より少数の RAID グループで同等の性能を達成でき，さらに省電力効果を高めることができる．

7.3.3 ドライブ単位省電力機能と RAID グループ単位省電力機能の併用

ドライブの Break Even Time は RAID グループのそれと比較して短い．さらに，RAID グループに対する read 入出力は，通常 RAID グループを構成するドライブの 1 つに対して行われるため，ドライブの入出力発行間隔は RAID グループの入出力発行間隔よりも長くなる可能性が高い．従って，入出力発行間隔が RAID グループ単位の省電力機能を活用できない程度の長さであっても，ドライブ単位の省電力機能を併用することにより，RAID グループの消費電力を削減できる可能性がある．

7.3.4 SSD の省電力の可能性

7.2 節で述べたように，Active 時の SSD の消費電力は HDD の約 1/6 であり，Spin up 待ち時間及び Break Even Time は HDD と比較して非常に短い．このことは，SSD の省電力効果の可能性が高いことを示している．

7.4 評価

7.4.1 評価条件

RAID 構成

本評価では，15 ドライブ構成の RAID グループ 8 台 (15x8) 及び 5 ドライブ構成の RAID グループ 24 台 (5x24) の消費電力と性能を比較した．単一ディスク筐体に 1 つの RAID グループが配置されていると仮定した．RAID レベルは 5, 4, 及び 0+1 をドライブ種別は HDD 及び SSD をそれぞれ比較した．

省電力機能

まず RAID グループ単位の省電力機能 (ディスク筐体の電源 OFF) を用いた場合の評価を行った．次にドライブ単位と RAID グループ単位の省電力機能を用いた場合の評価を行った．

Spin up 待ち時間と Break Even Time

15x8 構成の RAID グループの Spin up 時間及び待ち時間は 4.3 節及び表 7.2 で述べた値を用いた．また，5x24 構成の RAID グループ当りのドライブ数は 15x8 構成の RAID グループの 1/3 であるため，5x24 構成の RAID グループの Spin up 時間は 15x8 構成の RAID グループの Spin up 時間の 1/3 とした．また 5x24 構成の RAID グループの Spin up エネルギーは 15x8 構成の RAID グループの 1/9，省電力状態に移行することにより削減できる電力は 1/3 であるため，Break Even Time は 15x8 構成の RAID グループの 1/3 である．

アプリケーション設定

TPC-C 及び TPC-H を用いて RAID グループの消費電力と性能の比較を行った．表 7.3 にアプリケーション設定を示す．

表 7.3: アプリケーション設定

Application	DB Size	DB Buffer Size	Conditions
TPC-C	500GB (#of warehouse is 5000)	25GB	# of Threads: 1000 Think TIme: 0 s
TPC-H	100GB (SF=100)	5GB	Run Query 1 to 22 one by one

データ配置

DB ファイルを Hot と Cold の 2 種類に分割した．Cold DB ファイルとは当該 DB ファイルを単独で RAID グループに配置し省電力機能を適用した場合に，当該 RAID グループの消費電力を削減できるファイルである．それ以外のファイルが Hot ファイルである．Hot ファイルの合計 IOPS と RAID グループの最大 IOPS を元に Hot RAID グループ数を決定し，Hot データをこれらの RAID グループに RAID グループの負荷が分散されるように配置した．表 7.4 に各 RAID 構成における Hot RAID グループ数を示す.

表 7.4: Hot RAID グループ数

Application	# of drives in a RAID Group	RAID Level	# of Hot RAID Groups
TPC-C	15 (8 RGs)	5, 4, 0+1	7
	5 (24 RGs)	5, 4	21
		0+1	21
TPC-H	15 (8 RGs)	5, 4, 0+1	1
	5 (24 RGs)	5, 4, 0+1	1

TPC-C における 5x24 構成の RAID 0+1 の Hot RAID グループのドライブ数は 80($20 \times 4 = 80$) は，RAID 4,5 構成の場合の HDD 数 ($15 \times 7 = 105$) より少ない．これは RAID 0+1 の冗長データ生成の負荷が RAID 4,5 の冗長データ生成の負荷より低いためである．

7.4.2 シミュレーション手法

アプリケーション入出力トレース

4.2.2 節で述べたシステム上で，表 7.3 に示した設定で TPC-C 及び TPC-H を稼動させ，その入出力トレースを取得した．RAID グループ数は 8，RAID グループの構成は HDD 数 15 台，RAID レベルは 5 である (DB 用の RAID グループのみ)．ここで取得したトレースを用いて，他の RAID レベル (RAID 4, 0+1)，及びドライブ数 5 台の場合の消費電力と性能をシミュレートした．

消費電力の計算

式 7.1 を用いて HDD 15 台構成の RAID グループの消費電力を計算した．これは 4.2.2 節での計測値を用いて求めた式である．$P_{HDD_{15}}$ は HDD 15 台構成の RAID グループの消費電力，i は秒当りの入出力数である．

$$P_{HDD_{15}} = \begin{cases} -1.594 \times 10^{-5} i^2 + 0.036i + 287.5 & i \leq 2000 \text{ のとき} \\ -1.840 \times 10^{-6} i^2 + 0.094i + 285.4 & i > 2000 \text{ のとき} \end{cases} \tag{7.1}$$

SSD 15 台構成の RAID グループの消費電力 $P_{SSD_{15}}$ は，式 7.2 に示す通りである．SSD 1 台の消費電力は，表 7.1 に示した値を用いた．ここで，48 はディスク筐体のベース部分の消費電力である．SSD の消費電力は HDD の 1/6 であるため，SSD RAID グループのベース部分の消費電力を HDD RAID グループの 1/6 と仮定した．

$$P_{SSD_{15}} = \begin{cases} 2.40 \times 15 + 48 & \text{入出力数} > 0 \text{ のとき} \\ 0.06 \times 15 + 48 & \text{入出力数} = 0 \text{ のとき} \end{cases} \tag{7.2}$$

HDD 5 台及び SSD 5 台構成の RAID グループの消費電力 P_{HDD_5} 及び P_{SSD_5} はそれぞれ式 7.3, 7.4 に示す通り，HDD 15 台及び SSD 15 台の場合の 1/3 とした．

$$P_{HDD_5} = 1/3 \times P_{HDD15} \tag{7.3}$$

$$P_{SSD_5} = 1/3 \times P_{SSD15} \tag{7.4}$$

性能の計算

省電力状態のドライブや RAID グループに対する read 入出力を発行したトランザクションあるいはクエリは，当該ドライブや RAID グループが Spin up するまで待たされると仮定した．それ以外のトランザクションあるいはクエリは遅延しないと仮定した．Spin up 待ち時間は 4.3 節及び表 7.2 に示した通りである．また，RAID グループやドライブの入出力応答時間が短縮しても，ドライブや RAID グループの秒当り入出力数は変わらないと仮定した．

7.4.3 評価結果

HDD RAID グループの消費電力及び性能

図 7.2 及び 7.3 は TPC-C を実行した場合の HDD RAID グループの消費電力とトランザクションスループットを示している．本評価では，RAID グループの省電力機能 (電源 OFF) のみを用いている．また，15x8 RAID 5 構成の消費電力とトランザクションスループットは実測値である．

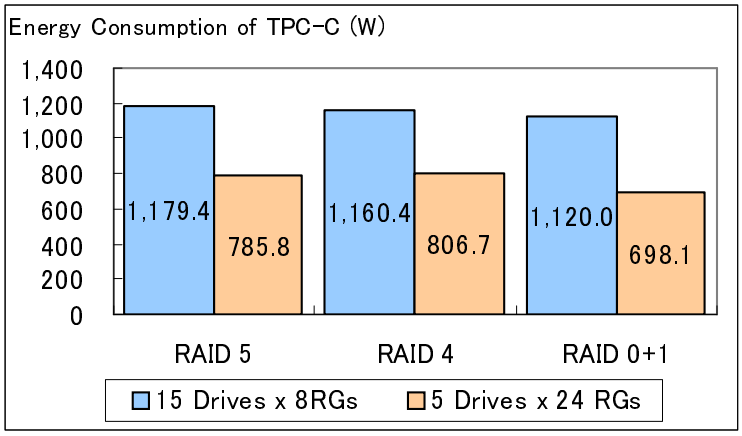

図 7.2: TPC-C 実行時の HDD RAID グループの消費電力

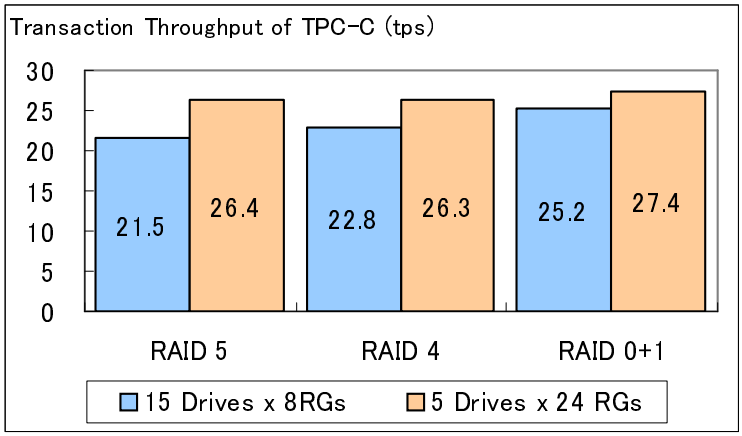

図 7.3: HDD RAID グループ上で動作する TPC-C のトランザクションスループット

図 7.2 に示すように，TPC-C では 5x24 構成の RAID グループの消費電力は 15x8 構成の RAID グループの消費電力より少ない．これは，5x24 構成では，Cold RAID グループは 3 つであるが，これら Cold RAID グループにアクセスがある場合でもそのうちの 1 台 (HDD 数 5 台) のみを Spin up するのみであったのに対し，15x8 構成では Cold RAID グループは 1 台のみであり，Cold RAID グループにアクセスがある場合は当該 Cold RAID グループ (HDD 数 15 台) を Spin up する必要があったためである．5x24 構成のトランザクションスループットは 15x8 構成より高い．これは 5x24 構成の Spin up 待ち時間が 15x8 構成より短いためである．

また，RAID 0+1 は，RAID 5 や 4 と比較して消費電力の削減幅が大きい．これは，TPC-C はランダム write を行うために RAID のパリティ生成オーバヘッドが発生するが，RAID 0+1 のパリティ生成オーバヘッドが RAID 5 や 4 と比較して小さく，Hot DB ファイルを配置する RAID グループ数を RAID 5 や 4 より少なくできたためである．

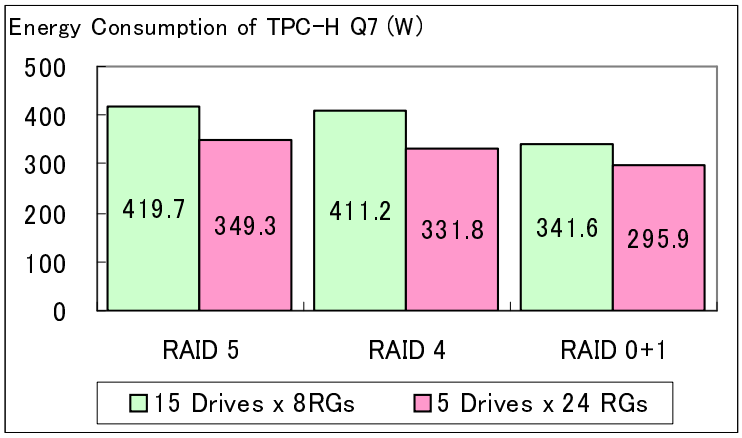

図 7.4: TPC-H クエリ 7 実行時の HDD RAID グループの消費電力

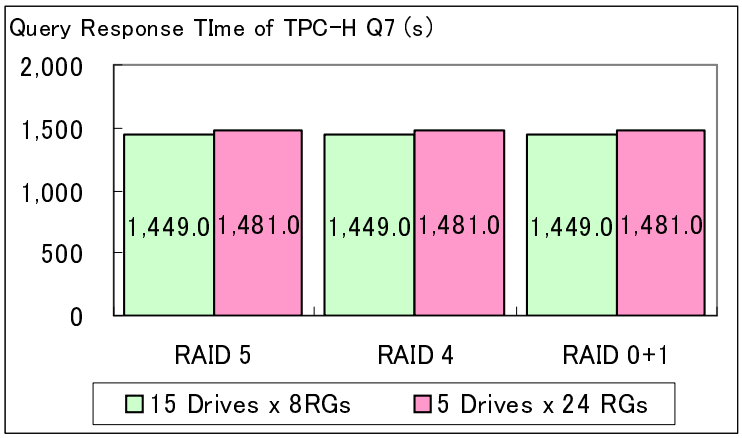

図 7.5: HDD RAID グループ上で動作する TPC-H クエリ 7 の応答時間

図 7.4 及び 7.5 は HDD RAID グループ上で TPC-H のクエリ 7 を実行した場合の RAID グループの消費電力とクエリ応答時間をそれぞれ示している．また，15x8 RAID 5 構成の消費電力とクエリレスポンスは実測値である．TPC-H では，RAID 4 及び RAID 0+1 の消費電力は RAID 5 の消費電力より小さくなっている．これは TPC-H は read 入出力のみを発行し冗長データを格納した HDD にアクセスする必要がなく，これらの HDD を省電力状態に移行することができたためである．性能の減少はほとんど見られなかった．

これらの結果は，少数の HDD から構成される RAID グループの方が，アプリケーションの省電力効果が高いことを示している．また，TPC-C ではパリティ生成オーバヘッドが小さい RAID グループの方が省電力効果が高く，TPC-H では冗長データのみを格納する HDD が多い方が省電力効果が高くなることを示している．

SSD RAID グループの消費電力及び性能

図 7.6 及び 7.7 は TPC-C を実行した場合の SSD RAID グループの消費電力とトランザクションスループットを示している．本評価では，RAID グループの省電力機能 (電源 OFF) のみを用いている．

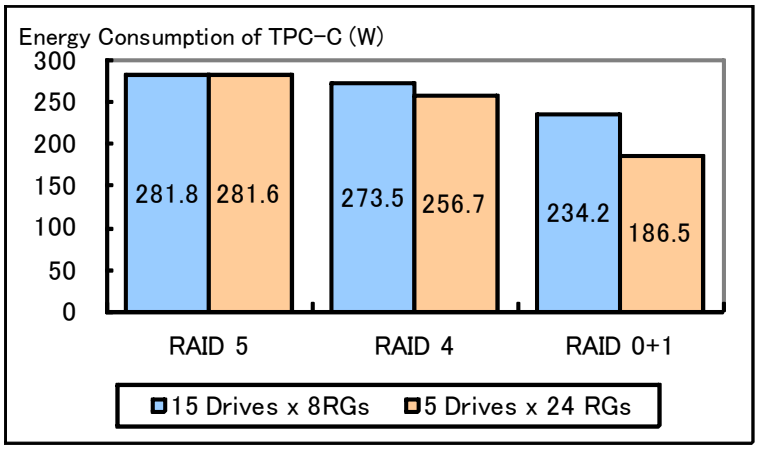

図 7.6: TPC-C 実行時の SSD RAID グループの消費電力

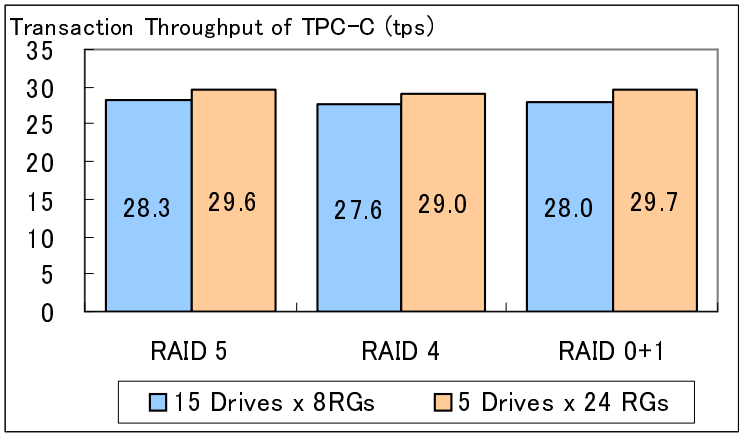

図 7.7: SSD RAID グループ上で動作する TPC-C のトランザクションスループット

図 7.6 より，RAID 5 の場合に 15x8 構成と 5x24 構成において消費電力の差がほとんどないことが分かる．これは，Idle 時や Spin up 時の SSD RAID グループの消費電力が非常に小さく，省電力状態に移行した RAID グループの消費電力の差がほとんどなかったためである．また，5x24 構成の RAID グループにおいては，HDD の場合と異なり RAID 4 構成の RAID グループ方が RAID 5 構成の RAID グループより消費電力が少ない．これは，SSD RAID グループでは Break Even Time が HDD と比較して非常に短く，write データを遅延している間，冗長データのみを格納した SSD ドライブを省電力状態に移行できたためである．

トランザクションスループットは 5x24 構成の方が良いが，これは HDD の場合と同様，RAID グループの Spin up に要する時間の差である．

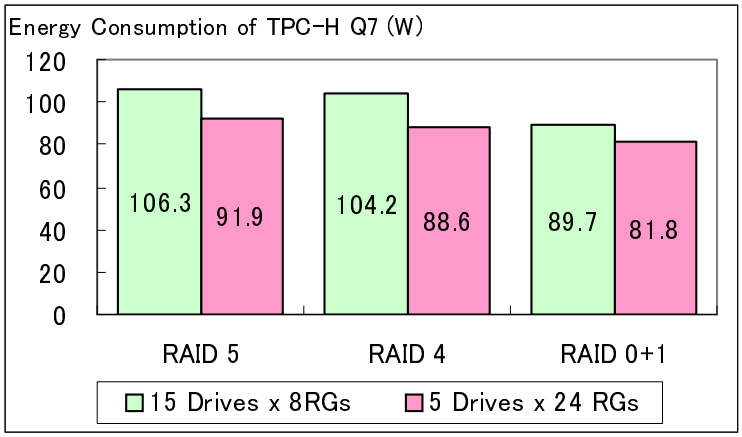

図 7.8: TPC-H クエリ 7 実行時の SSD RAID グループの消費電力

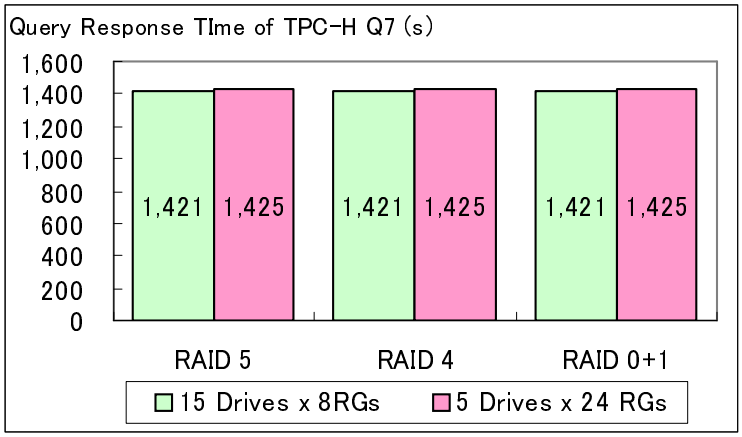

図 7.9: SSD RAID グループ上で動作する TPC-H クエリ 7 の応答時間

図 7.8 及び 7.9 は SSD RAID グループ上で TPC-H のクエリ 7 を実行した場合の RAID グループの消費電力とクエリ応答時間をそれぞれ示している．TPC-H では，何れの RAID レベルにおいても，5x24 構成の RAID グループの消費電力の方が小さくなっている．これは，5x24 構成の RAID グループの方が 15x8 構成の RAID グループより Break Even Time が短く，RAID グループに省電力機能を適用できる機会が多かったためである．また，図 7.9 より，何れの場合もクエリの応答時間は 5x24 構成の RAID グループの方が少し長いことが分かる．これは，クエリ実行中の RAID グループの Spin up 回数の差である．

ドライブ単位の省電力機能と RAID グループ単位の省電力機能の併用

SSD RAID グループについて，ドライブ単位の省電力機能と RAID グループ単位の省電力機能を併用した場合の省電力効果を調査した．図 7.10 に TPC-C を実行した場合の消費電力の削減率を示す．

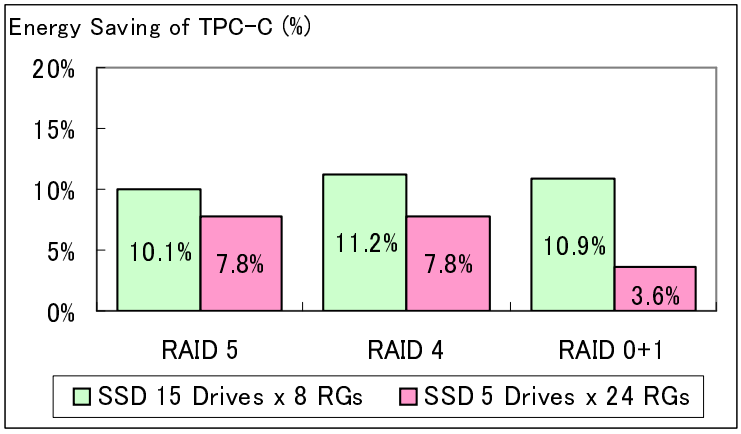

図 7.10: SSD 単位及び RAID グループ単位の省電力機能を用いた場合の消費電力削減率 (TPC-C)

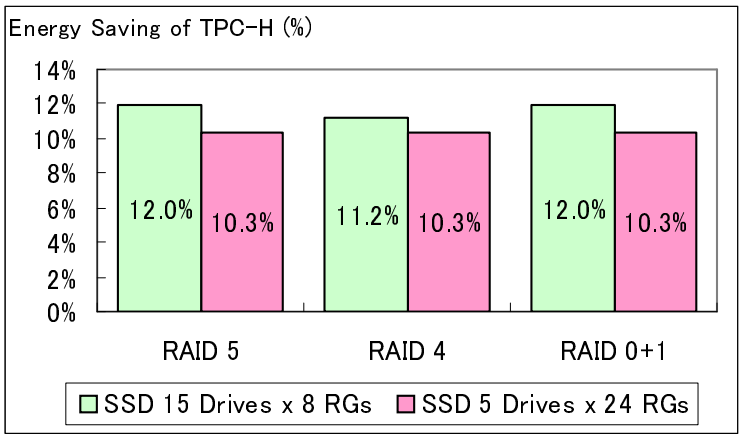

図 7.11: SSD 単位及び RAID グループ単位の省電力機能を用いた場合の消費電力削減率 (TPC-H)

図 7.10 より，SSD 単位の省電力機能を併用することにより，TPC-C を実行する 15x8 構成の RAID グループの消費電力を 10%以上，5x24 構成の RAID グループの場合でも 3%から 8%近く消費電力を削減できる可能性があることが分かる．また，図 7.11 より，SSD 単位の省電力機能を併用することにより，TPC-H を実行する 15x8 構成の RAID グループの消費電力を 11%以上，5x24 構成の RAID グループの場合でも 10%以上削減できる可能性があることが分かる．これらは，SSD の Break Even Time が SSD RAID グループの Break Even Time より短いこと，及び RAID グループ内の SSD に対する入出力が分散した結果

SSD に対する入出力発行間隔が RAID グループに対する発行間隔より長いため，省電力機能をより多く活用できたためである．

また，HDD RAID グループに対しても同様の調査を行ったが効果はなかった．これは，HDD の Break Even Time は SSD のそれと比較して非常に長く，HDD 単位の省電力機能を適用するには至らなかったためである．

7.5 まとめ

省電力可能なストレージ構成を導出すべく，構成が異なる RAID グループの省電力の可能性を，アプリケーションと RAID 構成の観点から調査した．シミュレーションの結果，少数のドライブを用いた RAID 0+1 構成がこれまで用いられている RAID 5 構成より省電力効果が高いことを明らかにした．本章で述べた定量的な評価結果は，RAID レベルと RAID を構成するドライブ数を見直すことによりストレージの消費電力を削減できる可能性があることを示している．さらに，SSD 構成の RAID グループでは，RAID グループ単位の省電力機能と個々のドライブ毎の省電力機能を併用することで，RAID グループ単位の省電力機能を用いる場合よりさらに削減できる可能性があることを示した．また，TPC-C に見られるようなランダム入出力と TPC-H に見られる一括シーケンシャル read のような入出力挙動の差異が，RAID 構成の種類毎の消費電力に大きな影響を与えていることを確認した．

本章で述べた RAID 構成は，前章までで述べてきた省電力手法と併用することが可能である．次章では，データセンタにおけるストレージの構築時の省電力も含めた階層的なデータ管理手法，及びストレージ省電力手法の実装及び大規模ストレージを用いた評価について述べる．

第8章　階層的データ管理と省電力ストレージ管理機構

8.1　階層的データ管理とストレージ省電力

近年，高いアクセス性能や短い復旧時間など高いサービスレベルが要求されるデータの管理にコストを掛け，そうではないデータの管理コストを低く抑える階層的なデータ管理が注目されている [74]．階層的データ管理の目的は，データの性能や信頼性に合わせてストレージをアクセス性能や冗長度が異なる階層に分割し，データをサービスレベルに適した階層に配置することで，増大し続けるデータの管理コストを低減することにある．

階層的なデータ管理では，ユーザが，応用処理の特性に基づきデータの管理階層を定める．このデータの管理階層をストレージの省電力に用いることにより，ユーザが設計したアプリケーションの要件をストレージの省電力に活用することが可能になる．

本節では，ユーザからの性能要求を満たしつつストレージ省電力を実現する機構を検討する．そのために，まずデータセンタにおける階層的なデータ管理と省電力運用を支援する，新たなモニタリング機構を提案する．本モニタリング機構は特に省電力という観点からストレージに着目しており，その特長は，i) ストレージ階層の性能，消費電力，温度の収集・蓄積，ii) データ毎のアクセス性能及びアクセス頻度の収集・蓄積，iii) 階層的なデータ管理のための指標の提供，iv) データの性能要件に適した格納先の検索，である．これらの特長により，管理者はストレージ階層の性能や消費電力，及び適切なストレージ階層にデータが配置されているか否かを知ることが可能となる．このモニタリング機構を利用することで階層的なデータ管理が容易に実現されデータセンタの省電力に貢献できる．さらに，データ階層を利用した実行時ストレージ省電力について述べる．アプリケーション実行前に入出力パターンを把握し省電力に利用するために，データの要件を入出力パターンとして利用する．また，ユーザからの性能要件を満たしつつ実行時ストレージ省電力を適用するために，個々のストレージ階層に実行時ストレージ省電力を適用し，さらなる消費電力の削減を試みる．

階層データ管理を利用した実行時ストレージ省電力は，データに対する要件をフレームワークの上位の情報として使用する (図 8.1).

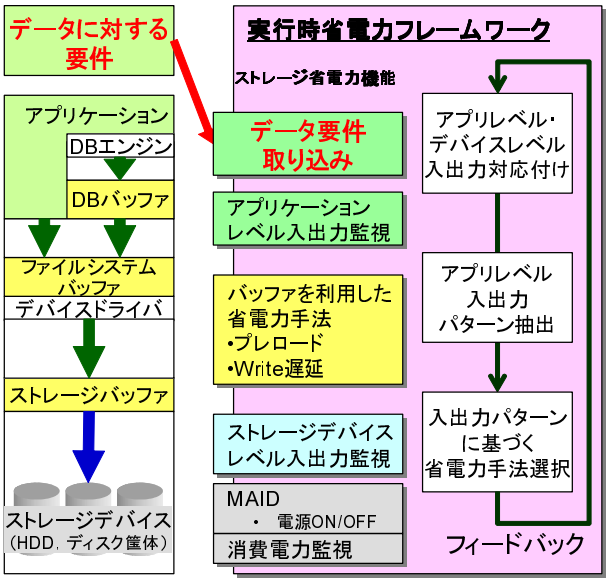

図 8.1: データに対する要件のストレージ省電力への活用

8.2 データ統合・解析システム DIAS

DIAS とは，地球規模の観測や各地域で得られたデータを収集，蓄積，統合，解析し，地球規模の環境問題や自然災害の脅威に対する危機管理に有益な情報を提供するデータ統合・解析システム [4] である．その主なアプリケーションは，海洋の気候変動の分析，ユーラシア寒冷圏の氷河の長期的な変動の明確化，地球上の天候変動と植生変動の関連の分析 [6] などである．これらのアプリケーションは，ストレージより数十 GB〜1TB のデータを読み出してサーバの主記憶に展開し，解析やシミュレーションを行う．そして結果をストレージに書き戻している．

DIAS は 3 台のサーバと約 1.6PB の容量を持つストレージ (全部で 5 台) を有する地球環境デジタルライブラリシステムであり，日々，計測データやシミュレーション結果などのデータが追加されている．運用開始は 2008 年度である．

DIAS の写真を図 8.2 に示す．サーバは (株) 日立製作所の SR16000/VL1，ストレージは同じく (株) 日立製作所の Adaptive Modular Storage 2500 である．

図 8.2: データ統合・解析システム DIAS

DIAS において使用されているストレージは，13 台から 18 台のディスク筐体 (1 ディスク筐体当り容量約 10TB) と，1 台のコントローラ筐体を有している．ディスク筐体は RAID 6(13D+2P) 構成を取る 15 台の HDD を格納している．

8.3 階層的データ管理を用いたストレージの構築

階層的なデータ管理を行うには，まず稼動するアプリケーションやユーザがデータに求める要件などを基にデータ管理階層を定める．次に各階層に求められる性能や消費電力を提供するストレージ階層を構築する．そして，データを適切なストレージ階層に配置する．

8.3.1 データ管理階層の決定

ユーザの要件からデータが満たすサービスレベルを決定し，それに基づきデータ管理を階層化する．サービスレベルにはデータのアクセス性能やアクセス頻度，データのアクセス待ち時間，最大容量などがある．従来，データ管理階層の構築には，データに求められる性能要件やデータの容量に基づき決定されている．

本研究では，データ階層構築に用いられる要件に，ストレージの省電力の指針となる新たな要件を加えることを提案する．これにより，ストレージ省電力の適用可否のヒントをユーザより得ることが可能となる．このヒント情報を利用することにより，省電力効果の高いストレージを構築することが可能となる．本研究では，省電力に関する指針として次のデータアクセスに関する要件を新たに導入する．ここで，アクセスとはファイルのオープンからクローズまでを想定する．

アクセス性能 データに要求される，単位時間当たりの入出力性能.

アクセス間隔 データに対して，どれくらいの間隔でアクセスが行われるか.

ユーザの要件がアクセス性能とアクセス間隔であり，ユーザがそれぞれを高・低，及び長・短の 2 つに分けた場合，データ管理階層は図 8.3 にに示すようになる．アクセス性能が高いデータやアクセス間隔が長いデータが電力コスト低減の対象である．アクセス性能とアクセス間隔は直交する軸であるため，低アクセス性能・長アクセス間隔と高アクセス性能・短アクセス頻度データは同一階層にある．

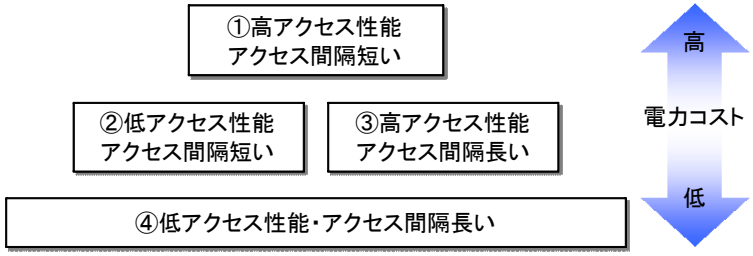

図 8.3: データ管理階層の決定

8.3.2　ストレージの階層化とデータの配置

次に，データ管理の階層に合わせてストレージの階層を決定する．アクセス性能及びアクセス間隔をサービスレベルの指標とするため，データのアクセス性能をストレージの性能に，データに対するアクセス間隔をストレージの電力制御方式にそれぞれ対応させる．図 8.3 のデータ管理階層では，ストレージの階層は，それぞれのデータ管理階層のサービスレベルを満たすよう，高性能高消費電力 (H1)，低性能高消費電力 (H2)，高性能低消費電力 (H3)，及び低性能低消費電力 (H4) の 4 つになる．ストレージ階層を構成するディスク筐体の数は，ユーザが求めるアクセス性能やデータ量を満たすように決定する．1 台のディスク筐体では十分なアクセス性能が出せない場合は，複数のディスク筐体間でデータをストライピングすることにより性能を確保する．また，ほぼ毎日アクセスされるデータを格納するストレージ階層は常時電源を ON にするなど，アクセス間隔に基づき省電力方式を決定する．

図 8.4 は図 8.3 に示す 4 つのデータ管理階層がある場合のストレージ階層 (H1〜H4) とデータ配置の例である．図 8.3 では，ユーザが求める高いアクセス性能を単一のディスク筐体で満たす高性能ストレージ，高いアクセス性能を複数のディスク筐体間でストライピングを行うことにより満たすことができ，かつディスク筐体毎の電源 ON/OFF が可能な中性能ストレージ，ディスク筐体単位の電源 ON/OFF が可能であるがユーザが求める高いアクセス性能は出せない低性能ストレージ，の 3 種類のストレージがあると仮定している．管理者はこれら 3 種類のストレージを，高性能ストレージ内の階層 (H1)，常時電源 ON である中性能ストレージ (H2)，アクセス時のみ電源 ON かつ高い性能を出すためにストライピングされた中性能ストレージ (H3)，及びアクセス時のみ電源 ON にする低性能ストレージ (H4)，の 4 階層に分割する．そして，アクセス性能が高くアクセス間隔が短いデータを H1 に，アクセス性能は低いがアクセス間隔が短いデータを H2 に，アクセス性能が高くアクセス間隔が長いデータを H3 に，アクセス性能が低く，アクセス間隔が長いデータを H4 に配置する．高性能ストレージのディスク筐体 1 台ではユーザのアクセス性能要件が満たせない場合は，高性能ストレージのディスク筐体間でもストライピングを行う．

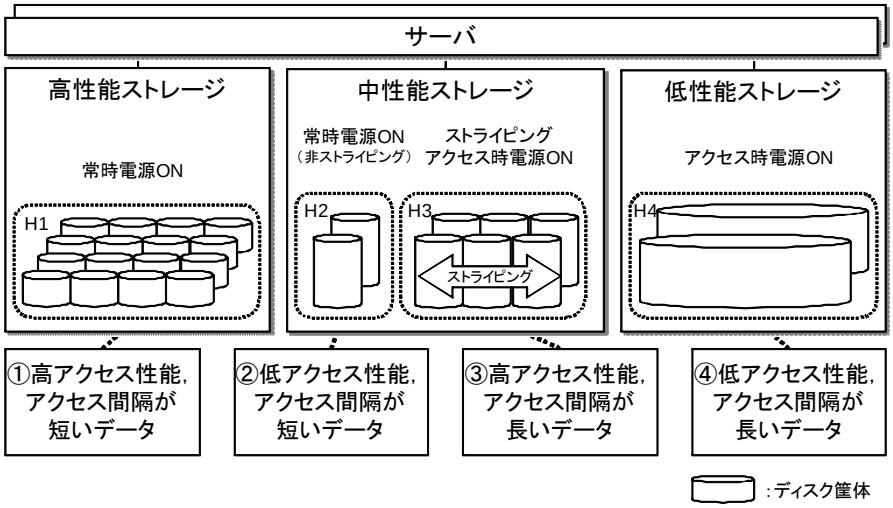

図 8.4: ストレージ階層とデータの配置

8.4 省電力モニタリング機構

8.4.1 省電力モニタリング機構の設計

階層的なデータ管理を支援するためには，ストレージ階層や個々のデータの視点で性能や消費電力を管理する必要がある．このため，本モニタリング機構は，従来のストレージ装置単位やディスク筐体単位の性能及び電力 [8] に加え，ストレージ階層毎のの性能及び消費電力，及びデータ毎のアクセス性能やアクセス頻度に関する情報を収集する．

また，DIAS では大規模なストレージ空間に対して随時データが投入される．このため，投入されるデータの容量やアクセス性能，アクセス頻度に基づき適切なストレージ階層にデータを投入することにより，データに求められる性能を維持しつつストレージの消費電力を削減できる可能性が高まる．例えば，投入されるデータが Break Even Time より短い間隔でアクセスされるデータであればそれらのデータを常時電源が投入されているストレージ階層に配置することによりディスク筐体の頻繁な電源 ON を防ぎ，消費電力を削減することが可能となる．また，単一ディスク筐体では達成できないほど高いデータ転送性能が求められるデータを，複数ディスク筐体間でデータを分散配置し並列に入出力を行うストレージ階層に格納することにより，データのアクセス性能を満たすことが可能となる．このため，本モニタリング機構は，データに求められるアクセス頻度やアクセス性能に基づき，データの配置に適したストレージ階層を検索する機能を有する．

8.4.2 省電力モニタリング機構の機能

サーバ及びストレージの性能情報，電力・温度情報の収集・蓄積 サーバ CPU のビジー率，ストレージのコントローラ筐体内プロセッサのビジー率，ディスク筐体のアクセス性能，ディスク筐体内の HDD のビジー率を一定時間間隔で収集し，性能情報管理 DB に格納する．またサーバやストレージの消費電力，及び温度を一定時間間隔で収

集し，電力・温度情報管理 DB に格納する．ストレージの消費電力及び温度は，コントローラ筐体及び個々のディスク筐体単位で収集・蓄積する．

サーバ・ストレージ性能情報及び電力・温度の可視化 サーバの CPU ビジー率，ストレージのディスク筐体毎のアクセス性能，及びコントローラ筐体のプロセッサのビジー率，ディスク筐体内の HDD のビジー率の推移及び平均値，現在値を可視化する．またサーバ，及びストレージの筐体毎の消費電力及び温度の推移，平均値，及び現在値を可視化する．

ストレージ階層の性能・消費電力情報の可視化 ストレージ階層のアクセス性能，ストレージ階層に含まれるディスク筐体内の HDD のビジー率の推移，平均値，及び現在値を可視化する．またストレージ階層に含まれるディスク筐体の合計消費電力の推移，平均値，現在値を可視化する．

データのアクセス性能とアクセス頻度の可視化 データにアクセスが行われた時の単位時間当りのアクセス性能，及び単位時間当りのデータのアクセス回数 (データ内のファイルのオープン回数の合計値) の推移，平均値，現在値を可視化する．

ストレージ階層の検索 データの性能要件を満たすストレージ階層を検索し，管理者に提示する．次の条件を満たす階層を選択する:

- 階層のアクセス性能 + 新規データのアクセス性能 < 階層が提供できる最大アクセス性能
- データのアクセス間隔が階層毎に決められた閾値以内

図 8.5 は，モニタリング機構の GUI を示している．左側はストレージの単位時間当たりアクセス性能，消費電力，及び温度の推移である．グラフ内の要素の色によってストレージ装置を区別している．右側は消費電力が高い時刻 (左側図の点線枠内) のストレージのコントローラ筐体及びディスク筐体の消費電力を示している．ストレージ装置 1 台を 2 列で表しており，小さな箱がディスク筐体を示している．箱の色が赤に近い (色が濃い) ほど筐体の消費電力が高いことを示している．白い箱は筐体の電源が OFF であることを示している．

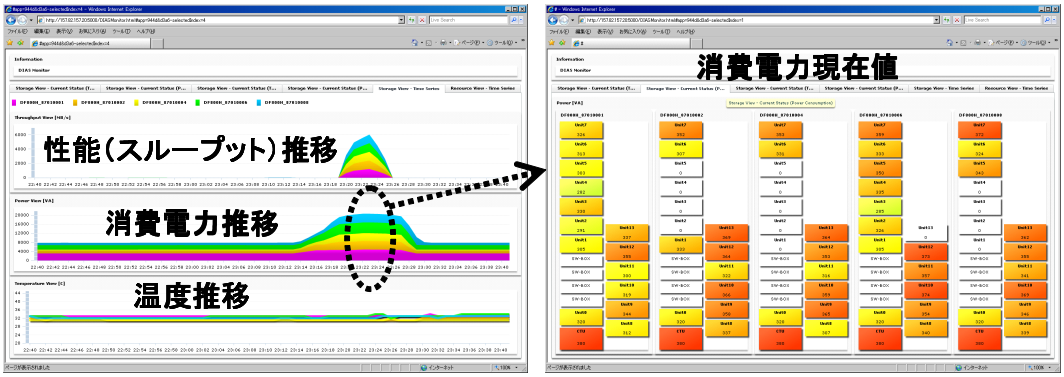

図 8.5: モニタリングシステム画面

8.5 DIAS における階層的データ管理と省電力

8.5.1 DIAS に対する階層的データ管理の適用

データ管理の階層化

8.3 章において述べた考え方に基づき DIAS ユーザの求める性能要件を満たすサービスレベルを決定した．DIAS ユーザのヒアリングに基づき作成したデータ毎のサービスレベルとその管理方針を表 1 に示す.

表 8.1: DIAS におけるデータ管理階層と管理方針

階層	データ管理階層のサービスレベル	データ管理方針
D1	データのアクセス性能が 250MB/s 以上, Break Even Time より長い Idle なし.	高い転送性能と応答性を維持．省電力は積極的に行わない.
D2	データのアクセス性能が 50MB/s 未満, Break Even Time より長い Idle なし.	高い応答性を維持. 省電力は積極的には行わない.
D3	データのアクセス性能が 100MB/s 以上, Break Even Time より長い Idle あり.	高い転送性能を維持. 省電力を優先.
D4	データのアクセス性能が 50MB/s 未満, Break Even Time より長い Idle あり.	省電力を優先.

省電力に対応した階層的データ管理を行うために必要なサービスレベルの指標として，アクセス性能，及び Break Even Time より長い Idle 時間の有無を選んだ．アクセス性能は，従来の省電力非対応のシステムにおいても見られる指標である．Break Even Time より長い Idle 時間の有無は，新たに導入した指標である．Idle 時間が Break Even Time より長いか短いかは，期間当りのアクセス回数とアクセス 1 回当りのアクセス時間長とから求められるアクセス間隔の平均値を用いる．アクセス性能は単一ディスク筐体により達成できるデータ転送速度 (50MB/s)，及び Break Even Time より長い Idle 時間の有無により，D1〜D4 の 4 種類のデータ階層を構築した.

ストレージの階層化とデータ管理階層との対応付け

データ管理階層 D1〜D4 のサービスレベルを満たすよう，DIAS のストレージをストレージ階層 H1 から H4 に分割した (図 6).

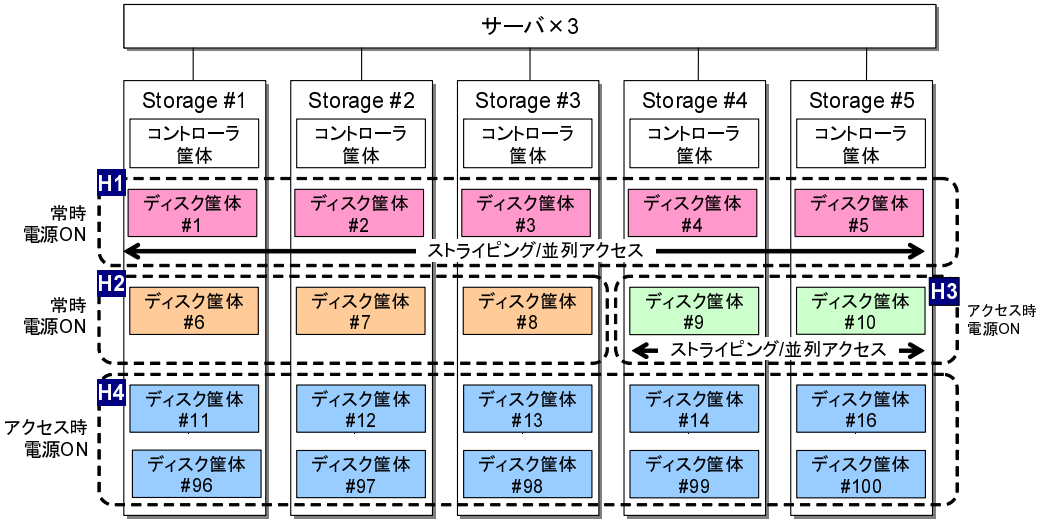

図 8.6: DIAS のストレージ階層

DIAS において用いられているストレージのディスク筐体のデータ転送性能は全て同一である．そこで，1 台のディスク筐体ではユーザの求めるアクセス性能を満たせない場合は，複数のディスク筐体間でデータをストライピングすることによりユーザの求めるアクセス性能を出せるようにした[1]．

ストレージ階層 H1 は，データ階層 D1 に対応するデータを格納する．データ階層 D1 の性能要件は，データアクセス性能 250MB/s かつ Break Even Time より長い Idle 時間がない．このため 5 台のディスク筐体間でデータをストライピングし要求されたアクセス性能を満たすようにした．また，D1 のデータは Break Even Time より長い Idle 時間がないため，電源は常時 ON とした．

ストレージ階層 H2 はデータ階層 D2 のデータを格納する．D2 のデータ量は約 21TB であるため 3 台のディスク筐体を用いた．データのアクセス性能要件は低いためストライピングは行わない．また，H2 も Break Even Time より長い Idle 時間がないデータを格納するため，常時電源 ON 運転とする．

ストレージ階層 H3 はデータ階層 D3 のデータを格納する．データアクセス性能は 50MB/s 以上でありかつ同時アクセス数が 3 以上であるため，2 台のディスク筐体間でデータのストライピングを行う．Break Even Time より長い Idle 時間があるデータを格納するため，データにアクセスがある場合のみディスク筐体の電源を ON にする．

残りのディスク筐体はデータ階層 D4 のデータを格納する (ストレージ階層 H4)．D4 のデータ量は約 700TB である．ストレージ階層 H3 の場合と同様，データにアクセスがある場合のみディスク筐体の電源を ON にする．アクセス性能は低いためストライピングは行わない．

[1]ディスク筐体 1 台当りの最大データ転送性能を 50MB/s としてストレージ階層を構築した．

8.5.2 階層的データ管理の効果

計測手法

DIAS の消費電力及び性能に関し，階層的なデータ管理の有効性を運用中のデータに基づくシミュレーションにより検討した．また提案手法の有効性を確認するため，提案手法と i) 全てのディスク筐体の電源を ON のままにする (省電力制御なし)，ii) データが入っていないディスク筐体の電源を OFF にする (現在の DIAS の運用形態; DIAS 現状)，iii) データに一日以上アクセスがない場合はディスク筐体にデータが格納されていても電源を OFF にする (電源 OFF)，iv) データのアクセス性能要件は考慮せずアクセス頻度が高いデータをできるだけ少数のディスク筐体にストライピングせずに配置しデータに一日以上アクセスがない場合にディスク筐体の電源を OFF にする (Non-SLA)，とを比較した．

階層的データ管理のストレージ階層の構築とデータの配置，及び方式 iv) のデータ配置を決定するために，DIAS から収集した 2010/4 月のデータ毎の性能情報を用いた．消費電力及び性能の計算には，2010/5 月及び 6 月分の性能情報及び消費電力情報を用いた．方式 ii) の性能及び消費電力は 2010 年 5,6 月の実測値である．それ以外の方式は，各ディスク筐体に配置されたデータ毎の秒当り I/O 数よりディスク筐体毎の秒当り I/O 数を求め，式 8.1 を用いて消費電力を計算した．式 8.1 は DIAS のストレージにおけるファイルシステムのランダムアクセスの実測値から算出した消費電力である．i はディスク筐体に対する秒当り入出力回数である．

$$P(i) = \begin{cases} -1.594 \times 10^{-5} i^2 + 0.036i + 287.5 (i \leq 2000) \\ -1.840 \times 10^{-5} i^2 + 0.094i + 287.5 (i 2000) \end{cases} \tag{8.1}$$

また，ディスク筐体の電源 ON/OFF の切り替えは，午前 0 時より 6 時間当該ディスク筐体にアクセスが行われていなければ当該ディスク筐体の電源を OFF にすると仮定した．その後もしアクセスが行われれば，その時点で電源を ON にし，翌日の午前 6 時まで電源 ON 状態が継続すると仮定した．

計測結果

図 8.7 に消費電力の比較結果を，DIAS 現状を 100%とした場合の比で示す．また，図 8.9,8.8 データ管理階層 D1〜D4 内のデータの 2010 年 5，6 月におけるアクセス待ち時間の平均値と平均アクセス性能をそれぞれ示す．

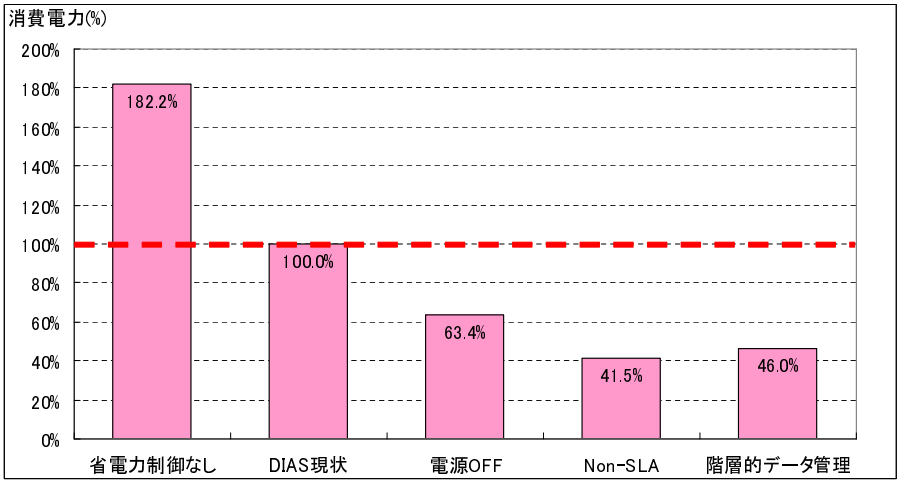

図 8.7: 消費電力比較

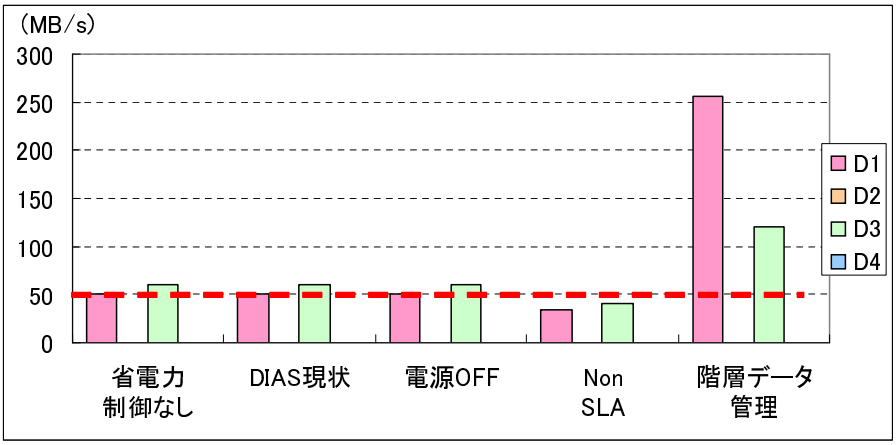

図 8.8: データ転送性能比較

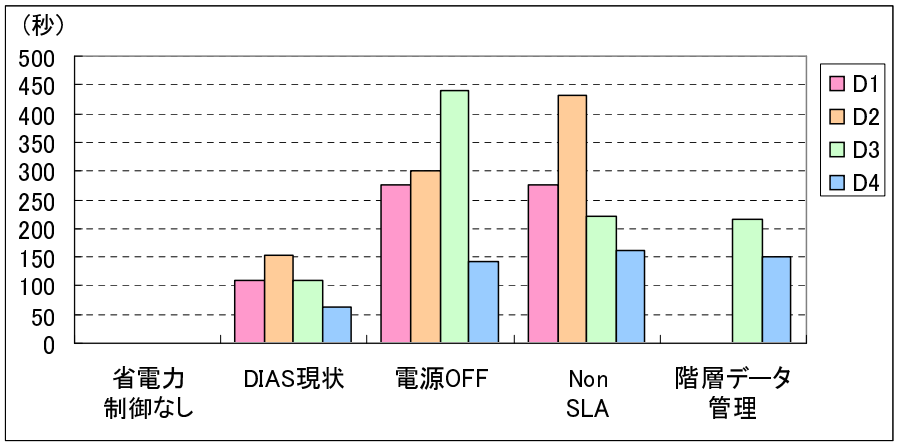

図 8.9: アクセス待ち時間比較

まず消費電力について検討する．図 8.7 より，消費電力が最も少ない制御方式は Non-SLA であることが分かる．階層的データ管理のストレージの消費電力は Non-SLA 方式よりわずかに多いが，他の方式と比較すると少ない．

次に，データ転送性能について検討する．図 8.8 より Non-SLA 方式のデータ階層 D1 及び D3 のアクセス性能はそれぞれ 38MB/s と 40MB/s であり，利用者が指定した 50MB/s を満たしていないことが分かる．これは，アクセス性能・アクセス頻度とも高いデータを単一のディスク筐体上に配置した結果，ディスク筐体でアクセス競合が発生したためである．一方，データアクセス性能はそれぞれ 257MB/s，119MB/s とサービスレベルの要件を満たしていることが分かる．

最後に，データアクセス時の待ち時間について述べる．図 8.9 より，Non-SLA では高アクセス頻度のデータ (D1, 2) に対するアクセス待ち時間は，データ階層 D1 のデータではデータ当り平均約 280 秒，D2 では 400 秒以上あることが分かる．一方，提案手法は D1, D2 の待ち時間がないことが分かる．また，現在の DIAS の運用 (DIAS 現状) でも待ち時間は少ないが，これはデータが格納されたディスク筐体は常時電源 ON 状態となっており，データアクセス時にディスク筐体の起動待ちが発生しないためである．つまり，現在の DIAS の運転ではデータが増加するにつれ消費電力が増大し，最終的には省電力制御無しの運用となる．

以上より「階層的データ管理」方式に基づく省電力化のみが利用者が求める性能要件を満たしつつ，ストレージの消費電力を削減できる可能性があることが分かる．

8.5.3 低消費電力運用支援

本モニタリング機構を用いることによりストレージのさらなる省電力運用が可能になることを示す．

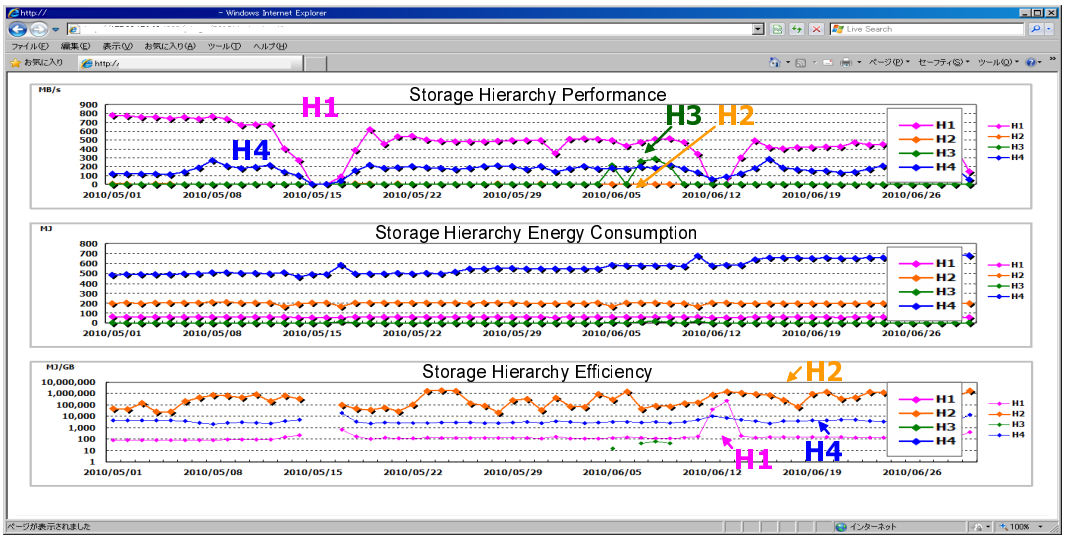

図 8.10: ストレージ階層のアクセス性能，電力，電力効率推移

図 8.10 はストレージ階層 (H1〜H4) の一日毎のアクセス性能の平均値の推移 (上段)，電力消費量の平均値の推移 (中段)，及び電力消効率 (1GB の転送に必要な消費電力量 (MJ)) の平均値の推移 (下段) を示している．ストレージ側の視点であるディスク筐体毎ではなく，管理者側の視点であるストレージ階層に基づく性能及び消費電力，電力効率を表示する．これらの機能により，管理者は，階層毎の性能や消費電力の傾向の把握，問題点の発見を容易に行うことが可能となる．具体的には，電力効率の可視化により，アクセスを行っていない，あるいはアクセス性能が低いにも関わらず高い電力を消費している階層の発見が可能となる．図 8.10 の下段を参照することにより，階層 H1，4 と比較して階層 H2 の電力効率が悪いことが分かる．

8.5.4 新規データ追加支援

DIAS には，2010/8 月から 9 月にかけて約 24.2TB のデータが追加された．このデータを対象に，新規データの追加に対して階層的なデータ管理を用いた場合と用いない場合 (DIAS 現状) の消費電力とアクセス性能を比較した．ここで，新たに追加されたデータのユーザ要件を，アクセス性能 140MB/s 以上かつ計測期間の 80%以上の期間でアクセスがある，とした．これらのデータはデータ管理階層 D1 に割当てられた．図 8.11 に比較結果を示す．

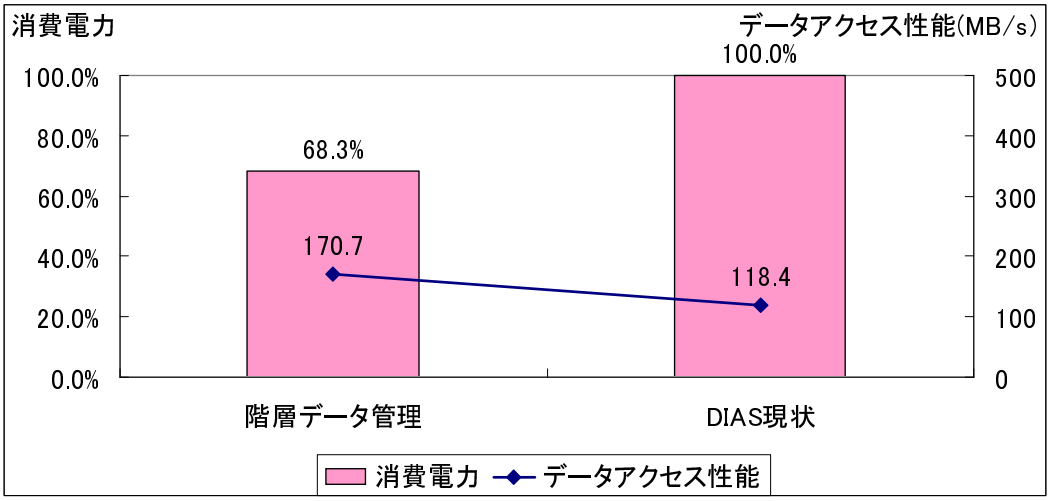

図 8.11: 新規データ追加後の消費電力とアクセス性能

図 8.11 より，階層的データ管理を用いた場合の消費電力は，DIAS 現状の消費電力の約 68.3%であるがアクセス性能は 170.7MB/s とユーザ要件を満たしている．DIAS 現状のアクセス性能は 118.4MB/s でありユーザ要件をみたさない．これは，階層的データ管理がストレージ階層 H1 の 5 台のディスク筐体 (常時電源 ON) にストライピングしてデータを入れたのに対し，DIAS 現状は新たなディスク筐体の電源を ON にし，ストライピングを行わずにデータを追加したためである．

8.6 まとめ

データセンタの省電力も考慮した階層的なデータ管理手法を提案した．階層的なデータ管理は，ユーザの要件に基づきデータを階層化し，それに合わせてストレージを性能や消費電力の異なる階層に分割しデータを配置する．また，本手法を用いてストレージの消費電力を削減する，省電力ストレージ管理機構を開発し，その評価を行った．東京大学で実際に運用されているデータ統合・解析システムから取得した I/O トレースを用いて評価を行った．この結果，省電力ストレージ管理機構を適用することにより，ユーザの要求性能を損なうことなく消費電力を現状の運用と比較し 54.0% 削減できることを示した．

第9章　結論

9.1　本論文のまとめ

デジタルデータの爆発的な増加に伴うストレージの急増により，ストレージ消費電力は今後も増加することが予想されている．このため，膨大なデジタルデータを管理するデータセンタではストレージ消費電力の削減が最重要の課題となりつつある．本論文では，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に用いるフレームワークを提案しその実装を示すと共に，ファイルサーバや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) などデータセンタで稼動するデータインテンシブアプリケーションを用いた評価を行い，高い省電力効果を達成できることを示した．

本論文では，まずアプリケーション実行時のストレージの省電力を目的とした実行時ストレージ省電力フレームワークを定義した．本フレームワークは，アプリケーション及びストレージデバイス双方の入出力挙動を収集・対応付けし，ストレージ省電力に有用な入出力パターンを抽出する．そして，抽出した入出力パターンに基づきストレージデバイスへのデータの配置やプレロード・write 遅延対象データの決定などの省電力手法を選択する．これによりアプリケーションの入出力挙動に適した省電力手法をアプリケーション実行時に容易に選択することができ，最悪の場合においても消費電力やスループットを省電力なしの場合と同等程度に抑えることが可能となる．これにより，様々なアプリケーションが稼動するストレージのアプリケーション実行時の消費電力の大幅な削減を達成することができる．

ストレージの省電力手法を検討するに当たり，本論文ではまず HDD 及びストレージの消費電力特性の計測を行い，その特性を明らかにした．この結果，HDD 及びストレージの省電力機能を用いることで HDD 及びストレージの消費電力を大きく削減できる反面，省電力状態からの Spin up には多大なエネルギーと待ち時間が必要であること，及び省電力機能を利用してストレージの消費電力を削減するには，入出力間隔が損益分岐時間より長い場合に省電力機能を適用することが重要であることを明らかにした．

次に，提案した実行時ストレージ省電力フレームワークを複数のハードディスクからなる DB サーバに適用し，その省電力を試みた．アプリケーションには入出力負荷の高い OLTP を用いた．まずハードディスク上で稼動する OLTP の入出力挙動特性を解析し，OLTP においても，表や索引等アプリケーションが認識するデータを単位として入出力挙動を解析することにより，損益分岐時間より長い入出力発行間隔が多数ありハードディスクの省電力の可能性があることを示した．既存手法の Popular Data Concentration (PDC) 及び Dynamic Data Reorganization (DDR) と比較した結果，5 台のディスク構成の場合には提案手法により数%のトランザクションスループットの低下でディスクの消費電力を 20%以上削減できることを示した．

さらに，提案した実行時ストレージ省電力フレームワークを複数のアプリケーションが稼働するストレージに適用し，その省電力を試みた．データセンタで稼動する主要なアプリケーションであるファイルサーバ，OLTP，DSS の入出力挙動を用いたストレージ省電力機構について述べた．また，アプリケーションが認識するデータ単位で入出力挙動を解析することにより，ファイルサーバ，OLTP，DSS においても損益分岐時間より長い入出力間隔をさらに増加あるいは延伸できる可能性があることを示した．またストレージレベルの入出力挙動とアプリケーションレベルの入出力挙動を結びつけるための，ストレージの省電力に適した 4 つの論理入出力パターンを定義するとともに，データの論理入出力パターンに基づきストレージ省電力手法を選択する新たなストレージの省電力手法を提案した．提案手法と従来の省電力手法である PDC 及び DDR を比較し，全てのアプリケーションについて，提案手法の電力削減率は PDC 及び DDR と同等かそれ以上であることを確認した．さらに，提案手法はアプリケーション性能の劣化を省電力なしの場合とほぼ同程度に抑えることができた．この結果は，実行時ストレージ省電力手法がデータセンタの省電力に大きく貢献できることを示している．

さらに，ストレージキャッシュの割当て制御と write 遅延の併用により，OLTP を実行するストレージの消費電力をさらに削減する手法について検討し，シミュレーションによる評価を行い，OLTP のスループットをほとんど低下させることなく，OLTP 実行中のストレージの消費電力を最大約 45%削減できることを示した．また，提案手法の実装を示すと共に提案手法を MAID 機能を持つストレージ上で動作させ，TPC-C 及び TPC-H を用いた評価を行った．この結果，TPC-C が稼動するストレージの消費電力を約 40%．TPC-H が稼働するストレージの消費電力 80%近く低減できることを確認した．

次に，近年のストレージが，ドライブの台数や RAID レベル，メディア種別などの多様な RAID 構成を取ることに着目した省電力手法の検討を行った．本研究では，省電力可能なストレージ構成を導出すべく，構成が異なる RAID グループの省電力の可能性を，アプリケーションと RAID 構成の観点から調査した．シミュレーションの結果，少数のドライブを用いた RAID0+1 構成がこれまで用いられている RAID5 構成より省電力効果が高いことを明らかにした．定量的な評価結果は，RAID レベルと RAID を構成するドライブ数を見直すことによりストレージの消費電力を削減できる可能性があることを示している．さらに，SSD 構成の RAID グループでは，RAID グループ単位の省電力機能と個々のドライブ毎の省電力機能を併用することで，RAID グループ単位の省電力機能を用いる場合よりさらに削減できる可能性があることを示した．また，TPC-C に見られるようなランダム入出力と TPC-H に見られる一括シーケンシャル read のような入出力挙動の差異により，省電力に適した RAID 構成が異なることを確認した．

最後に，データセンタの省電力も考慮した階層的なデータ管理手法を提案した．データの性能や信頼性などの要件に合わせて，ストレージをアクセス性能や冗長度が異なる階層に分割し，データをその要件に適したストレージの階層に配置することで，増大し続けるデータの管理コストを低減する．本研究では，データに対する要件をフレームワークの上位の情報として使用することにより，ユーザからの性能要求を満たしつつストレージ省電力を実現する機構を検討した．東京大学で実際に運用されているデータ統合・解析システム (DIAS) から取得した入出力トレースを用いて評価を行った結果，省電力ストレージ管理機構を適用することにより，ユーザの要求性能を損なうことなく現運用と比較して消費

電力を最大 54% 削減できることを確認した．

以上，本論文では，提案手法が，アプリケーションの入出力挙動を用いてストレージデバイスに対する入出力を把握することにより，アプリケーションの性能低下を抑えつつストレージの消費電力を大幅に削減できることを確認した．また，HDD，ストレージ，及び DIAS を用いた評価により，提案手法が様々な規模のストレージやアプリケーションに広く適用可能であることを確認した．

9.2 今後の研究課題

本研究では，アプリケーションの入出力挙動を用いた実行時ストレージ省電力フレームワークを提案した．本フレームワークは，対象とするシステム毎に省電力手法を配置するバッファを変えることにより，様々なシステム構成に対応することが可能である．本論文では，OLTP が稼動するハードディスクに対しては DB バッファ層に，ファイルサーバや OLTP，DSS が稼動するストレージに対してはストレージバッファ層に省電力手法を組み込んでいる．これらのバッファには，バッファの分割機能や電源断などの障害に対する耐障害性機能が組み込まれており，本論文でもこれらの機能を活用している．しかし，アプリケーションからストレージデバイスに至る間の主要なバッファの一つであるファイルシステムバッファには，バッファの分割機能や耐障害機能は組み込まれておらず，そのまま省電力手法を組み込んだのでは期待した省電力効果や耐障害性を得ることは難しい．今後は，本フレームワークの省電力手法のファイルシステムバッファへの組み込み方法を検討する予定である．

また，本フレームワークの研究では，データセンタで稼動する基幹的なアプリケーションであるファイルサーバや DSS，OLTP を対象に省電力手法を検討した．これらのアプリケーションは基本的に止めることはできないが，データセンタで稼動するアプリケーションには Web サービスなどのように基幹的な部分とそうではない部分の両方を併せ持つアプリケーションも稼動している．このようなアプリケーションでは多少の遅延を許すことにより更に省電力できるの可能性がある．そのようなアプリケーションが稼動するストレージの省電力手法についても検討を行う．

また，本論文では，大規模ストレージにおける省電力を考慮した RAID 構成の検討を行い，アプリケーションの入出力挙動に適した省電力 RAID 構成があることを明らかとした．本研究では RAID 構成として RAID を構成するドライブ数，メディアの種別，及びディスク筐体全体と個々のドライブ毎の省電力の併用などの観点から省電力の可能性について述べたが，さらに SSD と HDD が混在したヘテロ構成の RAID グループや，冗長ドライブが増加することによる容量効率まで含めた省電力の可能性を検討する．

さらに本論文では，大規模データに対して適用される階層データ管理において用いられるデータに対する要件を利用した省電力手法について検討している．本研究では，データに対する要件として入出力性能とデータに対するアクセス間隔を用いたデータ階層及びストレージ階層の構築手法を示した．しかし，データに対する要件はこれらの他にも応答時間や容量などの観点があり，さらにデータに対する要件以外の観点として電力消費の目標値も要件として与えられることが考えられる．今後はこれらの指標を包括的に活用する省電力手法を検討する．

謝辞

本研究を進めるにあたり，多くの方々のご指導やご協力を頂きました．ここに感謝の意を表したいと思います．

指導教官である東京大学生産技術研究所の喜連川優教授には素晴らしい研究テーマを頂き，日頃から研究生活全般に渡り熱心なご指導とご助言を頂きました．深く感謝の意を表します．

主査である国立情報学研究所の安達淳教授をはじめ，東京大学大学院情報理工学系研究科の坂井修一教授，豊田正史准教授，東京大学大学院工学系研究科の相田仁教授，東京大学大学院新領域創成科学研究科の杉本雅則准教授には，論文審査を通じて，大変有益なご指摘とご指導を賜りました．また，東京大学大学院情報理工学系研究科の浅見徹教授には，アドバイザ教員として多くの有意義な意見とご指導を賜りました．厚く御礼を申し上げます．

本研究を進めるに当っては，国家基幹技術・海洋地球観測探査システムのデータ統合・解析システム (DIAS) を使用させて頂きました．また，日立製作所 RAID システム事業部におかれましては省電力機能を強化したプログラムをご提供頂きくと共に，日立製作所ソフトウェア事業部より DBMS ソフトウェアある HiRDB を借用させて頂きました．関係各位に深く感謝致します．

喜連川研究室の中野美由紀特任准教授には日々の研究に関して非常に有益なご助言と熱心なご指導を頂きました．根本利弘准教授，合田和生特任助教には，実験環境の利用に関して多くの便宜を図って頂きました．ここに改めて深く感謝の意を表します．また，本論文の執筆の機会を与えて下さいました日立製作所横浜研究所の関係各位に深く感謝の意を表します．

最後に，常に暖かい励ましをもって研究生活を支えてくれた妻麻里と子供たちに心から感謝いたします．

参考文献

[1] Iometer. http://www.iometer.org/, 2001.

[2] Manpage of hdparm. http://linuxjm.sourceforge.jp/html/hdparm/man8/hdparm.8.html, 2005.

[3] The Rise of MAID : A New Tier in Disk Storage. Technical report, 2005.

[4] DIAS データ統合・解析システム. http://www.editoria.u-tokyo.ac.jp/dias/, 2006.

[5] Product Manual Barracuda ES Serial ATA. 2007.

[6] My Atlas and Plot Service. http://www.jamstec.go.jp/drc/maps/j/, 2010.

[7] TPC BENCHMARK C Standard Specification Revision 5.11. *Transaction Processing Performance Council*, No. February, 2010.

[8] 日立ストレージソリューション ストレージシステム稼動管理. http://www.hitachi.co.jp/products/it/storage-solutions/products/software/hsms/htm, 2010.

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

[10] TPC BENCHMARK H (Decision Support) Standard Specification Revision 2.14.2. *Transaction Processing Performance Council*, 2011.

[11] Giovanni Agosta, Marco Bessi, Eugenio Capra, Chiara Francalanci, and Politecnico Milano. Dynamic Memoization for Energy Efficiency in Financial Applications. In *Green Computing Conference and Workshops (IGCC), 2011 International*, pp. 1–8, 2011.

[12] Faraz Ahmad and T N Vijaykumar. Joint Optimization of Idle and Cooling Power in Data Centers While Maintaining Response Time. In *Proceedings of the fifteenth edition of ASPLOS on Architectural support for programming languages and operating systems*, pp. 243–256, 2010.

[13] Josep Ll. Berral, Íñigo Goiri, Ramón Nou, Ferran Julià, Jordi Guitart, Ricard Gavaldà, and Jordi Torres. Towards energy-aware scheduling in data centers using machine learning. *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking - e-Energy '10*, Vol. 2, p. 215, 2010.

[14] A.D. Brunelle. btrecord and btreplay User Guide, 2010.

[15] Enrique V. Carrera, Eduardo Pinheiro, and Ricardo Bianchini. Conserving disk energy in network servers. *Proceedings of the 17th annual international conference on Supercomputing - ICS '03*, p. 86, 2003.

[16] Doron Chen, George Goldberg, Roger Kahn, Ronen Kat, and Kalman Meth. Leveraging Disk Drive Acoustic Modes for Power Management. In *26th IEEE Symposium on Massive Storage Systens and Technologies*, No. May, 2010.

[17] Yuan Chen, Daniel Gmach, Chris Hyser, Zhikui Wang, Cullen Bash, Christopher Hoover, and Sharad Singhal. Integrated management of application performance, power and cooling in data centers. *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, pp. 615–622, 2010.

[18] Tom Clark, Feng Yang, Greg Kaine, Brian Leete, and Seiram Ranganathan. 第 2 世代インテル Centrino モバイル・テクノロジ・プラットフォームに採用された低消費電力のオーディオ及びストレージ I/O テクノロジ. *Intel Technology Journal*, Vol. 9, No. 1, pp. 1–13, 2005.

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

[20] Dennis Colarelli, Dirk Grunwald, and Michael Neufeld. The Case for Massive Arrays of Idle Disks (MAID). In *In The 2002 Conference on File and Storage Technologies*, pp. 1–6, 2002.

[21] Rajarshi Das, Gerald Tesauro, Jeffrey O Kephart, David W Levine, Charles Lefurgy, and Hoi Chan. Autonomic Multi-Agent Management of Power and Performance in Data Centers. In *7th International Conference on Autonomous Agents and Multiagent Systems*, pp. 107–114, 2008.

[22] F.M. David and C.F. Devaraj. Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management. *10th International Symposium on High Performance Computer Architecture (HPCA'04)*, pp. 118–118, 2008.

[23] 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.

[24] Fred Douglis, P. Keishnan, and Brain Breshad. Adaptive Disk Spin-down Policies for Mobile Computers. In *Proceedings of the 2nd Symposium on Mobile and Location-Independent Computing*, No. April 1995, pp. 121–137, 1995.

[25] Jon Flower. The Benefits of Balance. In *Storage Networking World Fall 2011 Conference*, 2011.

[26] Kazuhisa Fujimoto, Hirotoshi Akaike, Naoya Okada, Kenji Miura, and Hiroaki Muraoka. Power-aware Proactive Storage-tiering Management for High-speed Tiered-storage Systems. In *Proceedings of the First USENIX conference on Sustainable information technology*, 2010.

[27] Anshul Gandhi, Yuan Chen, Daniel Gmach, Martin Arlitt, and Manish Marwah. Minimizing Data Center SLA Violations and Power Consumption via Hybrid Resource Provisioning. In *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011.

[28] Lakshmi Ganesh, Hakim Weatherspoon, Mahesh Balakrishnan, and Ken Birman. Optimizing Power Consumption in Large Scale Storage Systems. In *Proceedings of the 11th USENIX workshop on Hot topics in operating systems*, 2007.

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

[30] Jody Glider. Towards Integrated Data Center Energy Management: An IBM Research Strategic Initiative. In *First USENIX Workshop on Sustainable Information Technology*, No. February, 2010.

[31] C. Gniady, a.R. Butt, and Y.C. Hu. Program counter-based prediction techniques for dynamic power management. *IEEE Transactions on Computers*, Vol. 55, No. 6, pp. 641–658, June 2006.

[32] C. Gniady and Y.C. Hu. Program Counter Based Techniques for Dynamic Power Management. In *10th International Symposium on High Performance Computer Architecture (HPCA'04)*, pp. 24–35. Ieee, 2004.

[33] Jorge Guerra, Wendy Belluomini, Joseph Glider, Karan Gupta, and Himabindu Pucha. Energy Proportionality for Storage : Impact and Feasibility. *ACM SIGOPS Operating Systems Review*, Vol. 44, No. 1, pp. 35–39, 2010.

[34] 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.

[35] S.K.S. Gupta and G. Varsamopoulos. Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 19, No. 11, pp. 1458–1472, November 2008.

[36] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: dynamic speed control for power management in server class disks. In *30th Annual International Symposium on Computer Architecture, 2003. Proceedings.*, pp. 169–179. IEEE Comput. Soc, 2003.

[37] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. Reducing disk power consumption in servers with DRPM. *IEEE Computer*, Vol. 36, No. 12, pp. 59–66, December 2003.

[38] 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.

[39] David P Helmbold, Darrell D E Long, Tracey L Sconyers, and Bruce Sherrod. Adaptive Disk Spin-Down for Mobile Computers. In *Mobile Networks and Applications*, Vol. 5, pp. 285–297, 2000.

[40] Magnus K Herrlin and Craig M Compiano. Top-Level Energy and Environmental Dashboard for Data Center Monitoring. *ASHRAE Transactions*, 2010.

[41] Hitachi. WhitePaper Hitachi Power & Acoustic Management - quietly cool. Technical Report March, 2004.

[42] Hitachi. Hard Disk Drive Specification Hitachi Deskstar P7K500 Hitachi CinemaStar P7K500. http://www.hitachigst.com/tech/techlib.nsf/techdocs/F92C2F79396264FD862573A90016BDAE/$file/Deskstar-CinemaStar_P7K500_Specifications-v1.2.pdf, 2007.

[43] Hitachi. Power and Acoustic Management. http://www.hitachigst.com/tech/techlib.nsf/techdocs/EBB67181ACB207C586256D340075B4DF/$file/WP_PowerAcoustic_25March.pdf, 2007.

[44] I. Hong and M. Potkonjak. Power optimization in disk-based real-time application specific systems. *Proceedings of International Conference on Computer Aided Design*, pp. 634–637, 1996.

[45] Wei Huang, Malcolm Allen-ware, John B Carter, Elmootazbellah Elnozahy, Hendrik Hamann, Tom Keller, Charles Lefurgy, Jian Li, Karthick Rajamani, and Juan Rubio. TAPO : Thermal-Aware Power Optimization Techniques for Servers and Data Centers. In *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011.

[46] Intel. Intel X25-E SATA Solid State Drive Uses Intel NAND flash memory Single Level. http://download.intel.com/design/flash/nand/extreme/extreme-sata-ssd-datasheet.pdf, 2009.

[47] K. Okada, N. Kojima and K. Yamashita. A Novel Drive Architecture of HDD: ” Multimode Hard Disk Drive”. In *Proceedings of the International Conference on Consumer Electronics*, pp. 7–8, 2000.

[48] Rini T Kaushik, Tarek Abdelzaher, Ryota Egashira, and Klara Nahrstedt. Predictive Data and Energy Management in GreenHDFS, 2011.

[49] H.S Kim, Dong In Shin, Young Jin Yu, Hyeonsang Eom, and H.Y. Yeom. DASCA : Data Aware Scaling Down to Provide Power Proportionality for Distributed Data Processing Frameworks. In *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011.

[50] Kyong Hoon Kim, Anton Beloglazov, and Rajkumar Buyya. Power-aware provisioning of Cloud resources for real-time services. In *Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science - MGC '09*, No. December, pp. 1–6, New York, New York, USA, 2009. ACM Press.

[51] P. Krishnan, P. M. Long, and J. S. Vitter. Adaptive Disk Spindown via Optimal Rent-to-Buy in Probabilistic Environments. Technical Report 1, January 1995.

[52] Willis Lang and Jignesh M Patel. Towards Eco-friendly Database Management Systems. In *4th Biennial Conference on Innovative Data Systems Research*, 2009.

[53] Dong Li, Hailong Cai, and Xiaoyu Yao. Exploiting redundancy to construct energy-efficient, high-performance RAIDs. In *Tech. Rep. TR-05-07-04, Computer Science and Engineering Department, University of Nebraska Lincoln*, pp. 1–20, 2005.

[54] Dong Li and Jun Wang. A Performance-oriented Energy Efficient File System. In *Proceedings of the international workshop on Storage network architecture and parallel I/Os*, 2004.

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

[56] Kester Li, Roger Kumpf, Paul Horton, and Thomas Anderson. A Quantitative Analysis of Disk Drive Power Management in Portable Computers 1 Introduction 2 Background Simulator Components. In *Proceedings of the USENIXWinter Conference*, pp. 279–291, 1994.

[57] Shen Li, Tarek Abdelzaher, and Mindi Yuan. TAPA : Temperature Aware Power Allocation in Data Center with Map-Reduce. In *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011.

[58] Y.H. Lu, E.Y. Chung, T. Simunic, G. De Micheli, and Luca Benini. Quantitative comparison of power management algorithms. In *Proceedings of the Design Automation and Test in Europe*, p. 20. Published by the IEEE Computer Society, 2000.

[59] Nagapramod Mandagere, Jim Diehl, and David Du. GreenStor: Application-Aided Energy-Efficient Storage. In *24th IEEE Conference on Mass Storage Systems and Technologies (MSST 2007)*, pp. 16–29. Ieee, September 2007.

[60] Manish Marwah, Ratnesh Sharma, Rocky Shih, Chandrakant Patel, Vaibhav Bhatia, Mohandas Mekanapurath, Rajkumar Velumani, and Sankaragopal Velayudhan. Data analysis, visualization and knowledge discovery in sustainable data centers. In *COMPUTE '09*

Proceedings of the 2nd Bangalore Annual Compute Conference, New York, New York, USA, 2009. ACM Press.

[61] John Masiewicz. ATA Power Management " SATA Device Initiated Power Management (DIPM)". http://www.t10.org/t13/docs2004/e04149r0 - DIPM Proposal.pdf, 2004.

[62] Justin Meza, Mehul a. Shah, Parthasarathy Ranganathan, Mike Fitzner, and Judson Veazey. Tracking the power in an enterprise decision support system. In *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design - ISLPED '09*, p. 261, New York, New York, USA, 2009. ACM Press.

[63] Timo Minartz and Thomas Ludwig. Managing Hardware Power Saving Modes for High Performance Computing. In *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011.

[64] Fred Moore, Aloke Guha, and D Ph. Introducing COPAN Systems' MAID Architecture, 2004.

[65] Justin Moore and Jeff Chase. Making Scheduling Cool: Temperature-Aware Workload Placement in Data Centers. In *USENIX Annual Technical Conference*, pp. 61–74, 2005.

[66] Ripal Nathuji and Karsten Schwan. VirtualPower : Coordinated Power Management in Virtualized Enterprise Systems Ripal Nathuji and Karsten Schwan. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, pp. 256–278, 2007.

[67] Ripal Nathuji and Karsten Schwan. VPM Tokens: Virtual Machine-Aware Power Budgeting in Datacenters. In *Proceedings of the 17th International Symposium on High Performance Distributed Computing*, 2008.

[68] Ekow Otoo, Doron Rotem, and Shih-Chiang Tsao. Workload-Adaptive Management of Energy-Smart Disk Storage Systems. In *IEEE International Conference on Cluster Computing and Workshops*, pp. 1–11. Ieee, 2009.

[69] 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.

[70] Ekow Otoo, Dron Rotem, and Shih-chiang Tsao. Energy Smart Management of Scientific Data. In *Proceedings of the 21st International Conference on Scientific and Statistical Database Management*, 2009.

[71] Ehsan Pakbaznia and Massoud Pedram. Minimizing data center cooling and server power costs. In *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design - ISLPED '09*, pp. 145–150, New York, New York, USA, 2009. ACM Press.

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

[73] D.A. Patterson, Garth Gibson, and R.H. Katz. A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of the 1988 ACM SIGMOD international conference on Management of data*, pp. 109–116. ACM, 1988.

[74] Paul P. Tallon, Joseph A, and S.J. Sellinger. Understanding the Dynamics of Information Management Costs. *Communications of the ACM*, Vol. 53, No. 5, p. 121, 2010.

[75] 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.

[76] Eduardo Pinheiro, Ricardo Bianchini, Enrique V. Carrera, and Taliver Heath. DYNAMIC CLUSTER RECONFIGURATION FOR POWER AND PERFORMANCE. In *Compilers and operating systems for low power*. 2003.

[77] Eduardo Pinheiro, Ricardo Bianchini, and Cezary Dubnicki. Exploiting redundancy to conserve energy in storage systems. *ACM SIGMETRICS Performance Evaluation Review*, Vol. 34, No. 1, p. 15, June 2006.

[78] J. S. Plank. The Raid-6 Liber8Tion Code. *International Journal of High Performance Computing Applications*, Vol. 23, No. 3, pp. 242–251, June 2009.

[79] M. Poess and R.O. Nambiar. Tuning Servers, Storage and Database for Power Efficient Data Warehouse. In *26th IEEE International Conf. on Data Engineering*, pp. 1006–1017, 2010.

[80] Meikel Poess and Raghunath Othayoth Nambiar. Energy Cost, The Key Challenge of Today's Data Centers: A Power Consumption Analysis of TPC-C Results. *Proceedings of the VLDB Endowment*, Vol. 1, No. 2, pp. 1229–1240, 2008.

[81] M Mustafa Rafique, Nishkam Ravi, Srihari Cadambi, Ali R Butt, and Srimat Chakradhar. Power Management for Heterogeneous Clusters : An Experimental Study. In *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011.

[82] David Reinsel and Jeff Janukowicz. Datacenter SSDs : Solid Footing for Growth. In *IDC White Paper #210290*, No. January, 2008.

[83] Mehul A Shah, Stavros Harizopoulos, and Justin Meza. Energy Efficiency : The New Holy Grail of Data Management Systems Research. In *4th Biennial Conference on Innovative Data Systems Research*, 2009.

[84] Yunfei Shang, Dan Li, and Mingwei Xu. Energy-aware Routing in Data Center Network, 2010.

[85] Bing Shi and Ankur Srivastava. Thermal and Power-Aware Task Scheduling for Hadoop Based Storage Centric Datacenters. In *Green Computing Conference and Workshops (IGCC), 2011 International*, 2010.

[86] S. W. Son, G. Chen, and M. Kandemir. Disk layout optimization for reducing energy consumption. In *Proceedings of the 19th annual international conference on Supercomputing - ICS '05*, pp. 274–283, New York, New York, USA, 2005. ACM Press.

[87] S.W. Son, M. Kandemir, and A. Choudhary. Software-Directed Disk Power Management for Scientific Applications. In *19th IEEE International Parallel and Distributed Processing Symposium*. Ieee, 2005.

[88] Mark W Storer, Kevin M Greenan, Ethan L Miller, and Kaladhar Voruganti. Pergamum : Replacing Tape with Energy Efficient , Reliable , Disk-Based Archival Storage, 2008.

[89] V. Tkachenko. tpcc-mysql.

[90] Dimitris Tsirogiannis, Stavros Harizopoulos, and Mehul a. Shah. Analyzing the Energy Efficiency of a Database Server. In *Proceedings of the 2010 international conference on Management of data*, pp. 231–242, New York, New York, USA, 2010. ACM Press.

[91] Nedeljko Vasic, Thomas Scherer, and Wolfgang Schott. Thermal-Aware Workload Scheduling for Energy Efficient Data Centers. In *Proceeding of the 7th international conference on Autonomic computing*, 2010.

[92] 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.

[93] Jun Wang, Xiaoyu Yao, and Huijun Zhu. Exploiting In-Memory and On-Disk Redundancy to Conserve Energy in Storage Systems. *IEEE Transactions on Computers*, Vol. 57, No. 6, pp. 733–747, June 2008.

[94] Jun Wang, Huijun Zhu, and Dong Li. eRAID: Conserving Energy in Conventional Disk-Based RAID System. *IEEE Transactions on Computers*, Vol. 57, No. 3, pp. 359–374, 2008.

[95] Jun Wang, Huijun Zhu, and Dong Li. eRAID: Conserving Energy in Conventional Disk-Based RAID System. *IEEE Transactions on Computers*, Vol. 57, No. 3, pp. 359–374, 2008.

[96] Zhikui Wang, Niraj Tolia, and Cullen Bash. Opportunities and challenges to unify workload, power, and cooling management in data centers. In *Proceedings of the Fifth International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks - FeBiD '10*, pp. 1–6, New York, New York, USA, 2010. ACM Press.

[97] 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.

[98] 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*, pp. 245–260, 2007.

[99] Andreas Weissel, Björn Beutel, and Frank Bellosa. Cooperative I/O - A Novel I/O Semantics for Energy-Aware Applications. In *Proceedings of the 5th symposium on Operating systems design and implementation*, 2002.

[100] Thomas Wirtz and Rong Ge. Improving MapReduce Energy Efficiency for Computation Intensive Workloads. In *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011.

[101] S W Worth. Green Storage - The Big Picture. In *Storage Networking World Spring 2010 Conference*, 2010.

[102] Tao Xie, Yao Sun, and San Diego. No More Energy-Performance Trade-Off : A New Data Placement Strategy for RAID-Structured Storage. In *The 14th Annual IEEE International Conference on High Performance Computing (HiPC 2007), Lecture Notes in Computer Science (LNCS 3834)*, pp. 35–46, 2007.

[103] H. Yada, H. Ishioka, T. Yamakoshi, Y. Onuki, Y. Shimano, M. Uchida, H. Kanno, and N. Hayashi. Head positioning servo and data channel for HDDs with multiple spindle speeds. *IEEE Transactions on Magnetics*, Vol. 36, No. 5, pp. 2213–2215, 2000.

[104] 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.

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

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

[107] Jianhui Yue, Yifeng Zhu, Zhao Cai, and Lin Lin. Energy and Thermal Aware Buffer Cache Replacement Algorithm. In *IEEE 26th Symposium on Mass Storage Systems and Technologies*, 2010.

[108] Qingbo Zhu, Zhifeng Chen, Lin Tan, Yuanyuan Zhou, Kimberly Keeton, and John Wilkes. Hibernator : Helping Disk Arrays Sleep through the Winter. In *Proceedings of the twentieth ACM symposium on Operating systems principles*, pp. 177–190, 2005.

[109] Qingbo Zhu and Yuanyuan Zhou. Power-Aware Storage Cache Management. *IEEE Transactions on Computers*, Vol. 54, No. 5, pp. 587–602, May 2005.

[110] 岡田尚也, 藤本和久, 赤池洋俊, 三浦健司, 岡村裕明. アクセス予測を利用した HPC 向け高速大容量階層ストレージの階層管理方式の予測精度向 上手法に関する検討. 第 8 回情報科学技術フォーラム, No. 2, pp. 3–4, 2009.

[111] 喜連川 優. よくわかるストレージネットワーキング. オーム社, 2011.

[112] 赤池洋俊, 藤本和久, 岡田尚也, 三浦健司, 村岡裕明. HPC 分野向け高速 ・ 大容量ストレージシステムの省電力化を図るアクセス予知階層ストレージの試作と省電力効果の検証. 第 8 回情報科学技術フォーラム, 2009.

[113] 早水悠登, 合田和生, 喜連川優. オンライントランザクション処理における Dynamic Voltage and Frequency Scaling の消費電力削減効果に関する実験的考察. 電子情報通信学会データ工学研究会, 電子情報通信学会技術研究報告, pp. 41–46, 2010.

発表文献

査読付き国際講演

1. Norifumi Nishikawa, Miyuki Nakano and Masaru Kitsuregawa, Energy Efficient Storage Management Cooperated with Large Data Intensive Applications, 28th IEEE International Conference on Data Engineering (IEEE ICDE 2012), 2012.

2. Norifumi Nishikawa, Miyuki Nakano and Masaru Kitsuregawa, Cache Effect for Power Savings of Large Storage Systems with OLTP Applications, 7th International Workshop on Databases in Networked Information Systems (DNIS 2011), LNCS 7108, pp.256-269, 2011.

3. Norifumi Nishikawa, Miyuki Nakano and Masaru Kitsuregawa, Energy aware RAID Configuration for Large Storage Systems, The First International Workshop on Energy Consumption and Reliability of Storage Systems (ERSS 2011), 2011

4. Norifumi Nishikawa, Miyuki Nakano and Masaru Kitsuregawa, Potentiality of Power Management on Database Systems with Power Saving Function of Disk Drives, The 22nd Australian Database Conference 2011, 2011 (Best Paper Award).

5. Norifumi Nishikawa, Miyuki Nakano and Masaru Kitsuregawa, Low Power Management of OLTP Applications Considering Disk Drive Power Saving Function, 21st International Conference on Database and Expert Systems Applications (DEXA 2010) Short paper, LNCS 6261, pp.241-250, 2010.

査読付き国内論文

1. 西川記史，中野美由紀，喜連川優，アプリケーション処理の I/O 挙動特性を利用したディスクの実行時省電力手法とその評価: オンライントランザクション処理における省電力効果，電子情報通信学会論文誌 学生論文特集 Vol.J95-D, No.3, 2012.

2. 西川記史，中野美由紀，喜連川優，データセンタの階層的データ管理と省電力化を支援するモニタリング機構の開発，日本データベース学会論文誌 (DBSJ Journal) Vol.9, No.3 (産業論文), 2011.

3. 西川記史，茂木和彦，河村信男，喜連川優，ストレージと DBMS の連携による I/O 性能障害の統合診断支援方式の開発と評価，情報処理学会論文誌コンピューティングシステム 1(3) (ACS 24), pp.28-40, 2008.

その他発表

1. 西川記史，中野美由紀，喜連川優，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，第 4 回データ工学と情報マネジメントに関するフォーラム，第 10 回日本データベース学会年次大会 (DEIM 2012)，2012

2. 西川記史，中野美由紀，喜連川優，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 DSS を用いた評価，第 74 回情報処理学会全国大会, 2011

3. 西川記史，中野美由紀，喜連川優，データインテンシブアプリケーションの I/O 挙動解析とストレージ省電力モデルの提案，電子情報通信学会研究会，2011

4. 西川記史，中野美由紀，喜連川優，OLTP を対象としたアプリケーション協調型大規模ストレージ省電力制御方式の提案及び評価，第 73 回情報処理学会全国大会, 2011 (大会優秀賞)

5. 西川記史，中野美由紀，喜連川優，階層的データ管理手法を用いた大規模省電力ストレージ構築方式の提案，第 3 回データ工学と情報マネジメントに関するフォーラム，第 9 回日本データベース学会年次大会 (DEIM 2011), 2011

6. 西川記史，中野美由紀，喜連川優，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，第 72 回情報処理学会全国大会，2010

7. 西川記史，中野美由紀，喜連川優，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－，第 2 回データ工学と情報マネジメントに関するフォーラム，第 8 回日本データベース学会年次大会 (DEIM 2010) , 2010