論文 / 著書情報
Article / Book Information

論題(和文)	ストレージシステムにおける省電力効果検証のためのシミュレータ
Title(English)	A Simulator to Evaluate the Power Consumption and Performance of Storage Systems
著者(和文)	引 田諭之, LE Hieu Hanh, Koh Kai Hung, 横田 治夫
Authors(English)	Satoshi Hikida, Hieu Hanh Le, Koh Kai Hung, Haruo Yokota
出典(和文)	情報処理学会研究報告，，，
Citation(English)	IPSJ SIG Technical Report, , ,
発行日 / Pub. date	2010, 8
権利情報 / Copyright	

ストレージシステムにおける省電力効果検証のためのシミュレータ

引田 諭之[†1]　LE Hieu Hanh[†1]
Koh Kai Hung[†1]　横田 治夫[†1]

近年データセンター等の消費電力量の増加に伴い，ストレージシステムの省電力化が重要な課題となっている．我々はこれまでに RAPoSDA というストレージシステムの省電力化手法を提案しており，さらに消費電力量を見積もるための概算式を構築してその効果を検証してきた．概算式ではワークロードを与えた場合での省電力効果や性能に関しては評価が出来なかったため，今回我々はその問題点に対処するためにストレージシステムの省電力効果と性能を評価するシミュレータを作成した．本論文では RAPoSDA について簡単に説明し，作成したシミュレータの構成や特徴について述べる．

A Simulator to Evaluate the Power Consumption and Performance of Storage Systems

Satoshi Hikida,[†1] LE Hieu Hanh,[†1] Koh Kai Hung[†1]
and Haruo Yokota[†1]

It becomes a big issue of reducing power consumption in storage systems corresponding to the increase of the energy consumption in a data center. We have proposed a method for reducing the power consumption in storage systems named *RAPoSDA(Replica Assiste Power Saving Disk Array)*, and also roughly estimated the power cunsumption effect of it. But we could not verify the actual performance and effect of the power reduction for the given workload.To evaluate the performance and effect of the power reduction on actual worklaod, we are now developing a dedicated simulator. In this paper, we describe the configuration of simulator of the RAPoSDA.

†1 東京工業大学大学院情報理工学研究科計算工学専攻
Department of Computer Science, Craduate School of Information Science and Engineering, Tokyo Institute of Technology

1. はじめに

近年，データセンターではその消費電力量の増加が大きな問題となっている．例えば2000年から 2006 年にかけて全米のデータセンターでは消費電力は年率 19 %で増加している．[1)] その中でも，ストレージの消費電力量は 3 倍も増加しており，ストレージの省電力化は重要な課題である．

そこで我々はストレージの省電力化手法として RAPoSDA（Replica Assisted Power Saving Disk Array）を提案した[2)]．この手法ではキャッシュメモリとディスクドライブの双方でプライマリ・バックアップ構成[3)] をとり，データ配置やディスクアクセスの制御を工夫することによりストレージの省電力化を実現する．これまでに，ストレージの消費電力量を概算する概算式を構築し，その概算式を用いて消費電力量を見積もり，その有効性を示してきた．

しかし概算式では性能に関して検証することはできず，実際にワークロードを与えた場合における省電力効果も明確には示せていなかったため，ワークロードのもとでの性能や省電力効果を検証することが課題であった．そこで我々は省電力効果や性能の評価は，シミュレータを新たに作成し，そのシミュレータ上で評価を行うこととした．

作成するシミュレータに対して我々が考える要求事項は以下の通りである．

- ディスクドライブの消費電力を評価するために回転状況（回転中か停止中であるかの状態やその期間等）が把握できること
- ディスクドライブの振る舞い（read/write やシーク動作およびその遅延）をシミュレート出来ること
- 複数のディスクを組み合わせた場合でのシミュレーションが可能であること
- キャッシュメモリとの連携部分もシミュレート出来ること
- ストレージの構成変更が容易に出来ること
- 様々な種類のワークロード（人工的，トレースベース）を用いてシミュレーションが出来ること

我々が対象しているストレージは複数のディスクドライブで構成されているものを前提にしているため，各ディスクドライブの振る舞いをシミュレート出来ることは必須である．また，ディスクドライブの消費電力はディスクの回転状況に大きく影響を受けている[4),5)] の

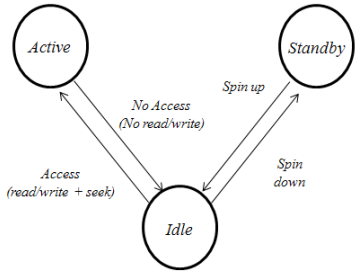

図 1 ディスクドライブの状態遷移図
Fig. 1 The state chart of a Disk drive

で，回転状況やその状況での消費電力量などもシミュレート出来なければならない．その他にも上記に列挙した要求事項を満たす必要があるが，詳細は 4 節で述べる．

なお，本論文は以下の通りの構成となっている．2 節ではディスクドライブの構成と消費電力について述べ，3 節では提案手法である RAPoSDA について説明する．4 では今回作成したシミュレーターのについて説明し，5 節で関連研究を述べて，最後の 6 節でまとめと今後の課題について述べる．

2. ディスクドライブの消費電力

ディスクドライブはデータを記録するディスク（プラッター）と呼ばれる円盤と，データの読み出し/書き込みを行うヘッドを搭載したアーム，プラッターを回転させるスピンドルモーター等の機械部品と，それらの動作を制御するコントローラーなどの制御部品とから構成されている．そのうち，消費電力に最も影響を与えるのはディスクを回転させるスピンドルモーターである[4]．

ディスクドライブは消費電力の観点から，アクティブ（Active）状態，アイドル（Idle）状態，スタンバイ（Standby）状態の 3 つの状態の状態遷移図で表すことが出来る．（図 1 参照）

表 1 より，3 状態の中で一番消費電力が大きいのはアクティブ状態であり，一番小さいのがスタンバイ状態である．この他，ディスクの回転を停止する時（Spin-down）と回転を開始するとき（Spin-up）でも一時的に大きな消費電力が必要となる．特にスピンアップ時はアクティブ状態時よりも大きな消費電力が発生することもある．このような性質から，ストレージの省電力化のためにはディスクドライブを長時間回転停止させることが必要である

表 1 ディスクドライブの状態と消費電力
Table 1 A table of status and corresponding power consumption

状態	I/O 処理	RPM	ヘッド位置	消費電力
Active	処理中	最高回転	ディスク上	大
Idle	処理なし	最高回転	ディスク上	中
Standby	処理なし	0	ディスク外	小

が，無闇に回転を停止させていてはディスクアクセスが必要になるたびにスピンアップが必要になり，消費電力はかえって従来よりも大きくなってしまうおそれがある．

そこで，ディスクのスピンダウンはどのような基準で行うべきかを判断するためにブレイクイーブン時間（break-even time）というものが用いられる．ブレイクイーブン時間とは，アイドル状態に対し，スタンバイ状態で節約できるエネルギーと，スピンアップとスピンダウンとで消費されるエネルギーの合計が等しくなる時間のことをいう．もしスタンバイ状態の期間がこのブレイクイーブン時間よりも長い場合，その分だけ省電力効果がある．

3. RAPoSDA について

RAPoSDA では，多数のディスクドライブを組み合わせたストレージシステムの省電力化を対象としている．データセンター等で実際に運用されるときは，データは冗長化されて複数のディスクに保存されているのが一般的なため，信頼性の確保も考慮している．この観点から，RAPoSDA ではキャッシュメモリとディスクドライブの双方でプライマリ・バックアップ構成をとるようにし，データ配置方法や，ディスクへの書き込みタイミングを工夫することで省電力化を実現している．

以下では，RAPoSDA の構成や動作のそれぞれににについて述べる．

3.1 構　　成

RAPoSDA は以下に示す 3 つの要素から構成される（図 2）．

- キャッシュメモリ
- キャッシュディスク
- データディスク

以下それぞれについて述べる．

3.1.1 キャッシュメモリ

キャッシュメモリには揮発性の RAM を用いることを前提としているため，信頼性を持たせるためにプライマリとバックアップの冗長構成をとる．それぞれのキャッシュメモリは

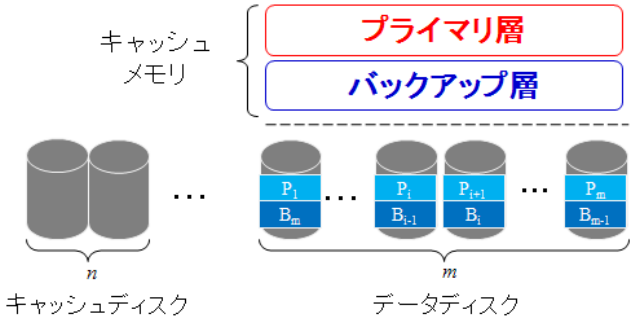

図 2 RAPoSDA の全体構成図
Fig. 2 Configuration of RAPoSDA

個別の電源系統を持ち，UPS（無停電電源装置）等で断電対策が施されているものとする．キャッシュメモリは複数で電源系統を共有する場合もあり，電源を共有しているキャッシュメモリをキャッシュ単位と呼ぶことにする．1 つのキャッシュ単位には 1 つか複数のディスクドライブが共有しており，キャッシュ単位を共有しているディスク数を N_{CU} で表す．$N_{CU} = 1$ ではキャッシュ単位とディスクは 1 対 1 で対応しており，これを単純構成と呼び，$N_{CU} \geq 2$ では複数のディスクでキャッシュ単位を共有するのでこれを複数構成と呼ぶ．

3.1.2 キャッシュディスク

データをキャッシュするためのディスクである．キャッシュメモリでは容量に限界があることと，多くのワークロードでは読み出し要求が書き込み要求よりも多いという状況から，キャッシュメモリは書き込みデータのバッファを主目的とし，キャッシュディスクを読み出し専用とする．読み出し要求に迅速に対応するキャッシュディスク数は後述するデータディスク数よりも少ない構成としており，常時回転させておき，ディスクのスピンアップに伴う応答遅延を回避する．

3.1.3 データディスク

実際のデータを格納するディスクである．キャッシュメモリ上のバッファがあふれた場合などにデータが書き込まれる．ディスクアクセスが，ある閾値時間を超えて発生しなかった場合そのディスクドライブをスピンダウンさせてスタンバイ状態に移行する．このスタンバイ状態期間がブレイクイーブン時間よりも長ければ長いほど省電力効果が得られる．

また，データディスクでも信頼性の確保のためにプライマリ・バックアップ構成をとる．

3.2 動　　作

3.2.1 書 き 込 み

書き込み処理は以下に示す通り大きく 3 つの処理に分けられる．

(1) キャッシュメモリへの書き込み

書き込みデータを d_{ij} とし，d_w を格納するディスク D_{ij} が所属しているキャッシュ単位を CU_i とする．ここで，i はキャッシュ単位の識別子で，j はキャッシュ単位に所属しているデータディスクの識別子である．

d_{ij} はまず CU_i のプライマリ層に書き込まれ，冗長性を持たせるために別のキャッシュ単位のバックアップ層へも書き込まれる．

d_{ij} を CU_i へ書き込むことによって，CU_i が最大バッファ容量を超えてしまう場合は，D_{ij} をスピンアップさせ，データディスクへの書き込み処理を行う．もし D_{ij} が回転中だった場合はそのままデータディスクへの書き込み処理を行う．

最大バッファ容量を超えなかった場合でも，D_{ij} が回転中でありかつあらかじめ設定しておいたバッファ容量の閾値を超えていた場合にも，データディスクへの書き込み処理を行う．

上記の条件判定は，プライマリ層のキャッシュ単位とバックアップ層のキャッシュ単位の両方で行う．

(2) データディスクへの書き込み

データディスクへの書き込みは，キャッシュ単位 i（CU_i）のバッファがあふれるか閾値を超えたときに発生する．キャッシュ単位は複数のデータディスクで共有しているので，データディスクへの書き込み処理は，キャッシュ単位を共有している全てのデータディスクが対象となる．ただし，バッファ中には格納すべきデータが存在していないデータディスクや，回転停止中のデータディスクについては書き込み処理は行われない．

CU_i を共有しているデータディスク D_{ij} が回転中で，かつ CU_i のバッファ中に格納するべきデータが存在していた場合，そのデータは D_{ij} に書き込まれ，バッファ中から削除される．書き込んだデータが D_{ij} のプライマリ領域のデータだった場合，同時に D_{ij} のバックアップ領域のデータもキャッシュメモリから書き込む．バックアップ領域のデータは CU_i とは異なるキャッシュ単位のバックアップ層に存在しているため，バックアップ層のバッファ容量はまだ余裕がある可能性はあるが，その場合でもデータディスクへの書き込みは行われる．書き込まれたバックアップ層のデータは，

そのバッファから削除される．書き込まれたデータがバックアップ領域のデータだった場合も同様に，プライマリ領域のデータをキャッシュメモリからデータディスクへ書き込む．

(3) **キャッシュディスクへの書き込み**
データディスクに書き込まれたデータは，次回同じデータに対する読み出し要求が発生した場合にデータディスクへのアクセスを抑制するためにキャッシュディスクへ書き込まれる．

3.2.2 読 み 出 し

読み出し動作は以下に示す通り大きく分けて 5 つの処理からなっている．

(1) **キャッシュメモリからの読み出し**
該当データがキャッシュメモリ上に存在するかを確認する．キャッシュメモリはプライマリ層とバックアップ層に分かれており，該当データが少なくともどちらかの層に存在していればデータはキャッシュメモリから読み出される．

(2) **キャッシュディスクからの読み出し**
キャッシュメモリ上に該当データが存在しなかった場合は，キャッシュディスクにデータが存在するかを確認する．もしキャッシュディスク上にデータが存在していれば，データはキャッシュディスクから読み出される．

(3) **データディスクからの読み出し**
キャッシュメモリ，キャッシュディスクに該当データが存在しなかった場合，データディスクから読み出す必要があるが，データディスクはプライマリ・バックアップ構成をとっているのでどちらのディスクから読み出すかを決定する必要がある．以下に対象データディスクを選択するパターンを示す．

- **片方のみ回転中** 回転している方のディスクから読み出す
- **両方回転中** バッファ容量の多い方のディスクから読み出す
- **両方停止中** 停止期間が長い方のディスクから読み出す

同一データディスクのプライマリ領域データとバックアップ領域データは，それぞれ異なるキャッシュ単位のプライマリ層とバックアップ層のバッファに格納されている．そのため，両者のバッファデータの量は異なる可能性があり，バッファデータの多いほうに対応するデータディスクを選ぶと，後述するデータディスクへのバッファデータの書き込み時にバッファ容量をより効率的に扱うことが出来る．

(4) **データディスクへの書き込み**
データディスクからデータを読み出すにはデータディスクが回転中か，スピンアップによって回転させた状態でなければならなず，さらにディスクアクセスが発生した時点でスピンダウンまでの閾値時間のカウントはリセットされてしまう．そこで，回転中である機会を活用して，そのデータディスクのプライマリ領域およびバックアップ領域に対応するキャッシュメモリ上のデータを書き込むようにする．このことによってキャッシュメモリのバッファに空き領域を増やすことが出来，後述するキャッシュディスクへの書き込み処理との連携によってデータディスクへのアクセス頻度を抑制することが期待できる．

(5) **キャッシュディスクへの書き込み**
書き込み処理と同様，データディスクに書き込まれたデータは次回以降の同一データに対する読み出し要求に備えてキャッシュディスクに書き込まれる．

4. シミュレータについて

4.1 目　　的

本シミュレータは，ストレージシステムの性能および消費電力を評価することを主な目的としている．また，提案手法である RAPoSDA や他の手法との比較を行えるようにするため，シミュレートするストレージシステムの構成も柔軟に変更できるように設計し，様々な構成同士での比較が容易に行えるようにする．

4.2 概　　要

複数のクライアントからの読み出し/書き込み要求を発行し，ストレージシステムが各要求に対して処理する際の消費電力や応答時間などをシミュレートする．

シミュレーションを実施する際は，最初に WorkloadGeneration コンポーネントによってワークロードを生成しておく．生成したワークロードやシミュレータ自体のパラメータファイル，ディスクドライブモデルのパラメータファイルをシミュレータの実行時に渡すことにより，ストレージシステムの構成や特性およびワークロードを動的に設定する．

シミュレーションを実施する際の入力として，いくつかのパラメータファイルがある．パラメータファイルは以下のものである．これらのうち，シミュレーションの構成を設定するパラメータファイルの内容を表 2 に示す．

- ワークロードを生成するためのパラメータファイル（4.3.1 節を参照）
- シミュレーションの構成を設定するパラメータファイル
- ディスクドライブモデルの情報を設定するパラメータファイル（4.3.3 節を参照）

表 2 シミュレータに渡すパラメータ
Table 2 Parameters of Simulator

Name	description
number of cache disks	キャッシュディスク数
number of data disks	データディスク数
number of cache memory	キャッシュメモリー数
memories per cache unit	キャッシュ単位当たりのキャッシュメモリ数
disks per cache unit	キャッシュ単位当たりのデータディスク数
init data	シミュレータに渡すワークロードが初期データかどうかを指定
block size	ディスクとの入出力処理を行う際のデータサイズ
layout policy	データ配置方法を指定
threshold of to spindown	アイドル状態からスピンダウンを開始するまでの閾値時間
threshold of memory buffer	キャッシュメモリからデータディスクへ書き込む際の閾値容量
model of cache disk	キャッシュディスクのディスクドライブモデル
model of data disk	データディスクのディスクドライブモデル
model of cache memory	キャッシュメモリの各種パラメータ

シミュレーションの結果はログ情報として DB に格納され，実行後に評価や解析のために使用する．

RAPoSDA で提案したデータの配置方法は，データ配置方法を管理するコンポーネントを用意し，そのコンポーネント上で実現している．RAPoSDA 以外の手法をシミュレートする際は，この部分のみを変更すればよく，変更が容易である．データ配置に関するコンポーネントに関しては 4.3.2 で詳しく述べる．

4.3 構　　成

シミュレータの全体構成を図 3 に示す．シミュレータを構成する主要なコンポーネントは以下の通りである．

- Workload Generation
- Data Layout Management
- Storage Devices
- Log Collection

次に個々のコンポーネントの詳細について説明する．

4.3.1 Workload Generation

ストレージに与えるワークロードを生成するコンポーネントである．WorkloadGenerator という名前の Java のクラスに表 3 に示す各種パラメータを設定ファイルを引数で渡して実行することにより，様々な種類のワークロードを生成することが出来る．WorkloadGenerator

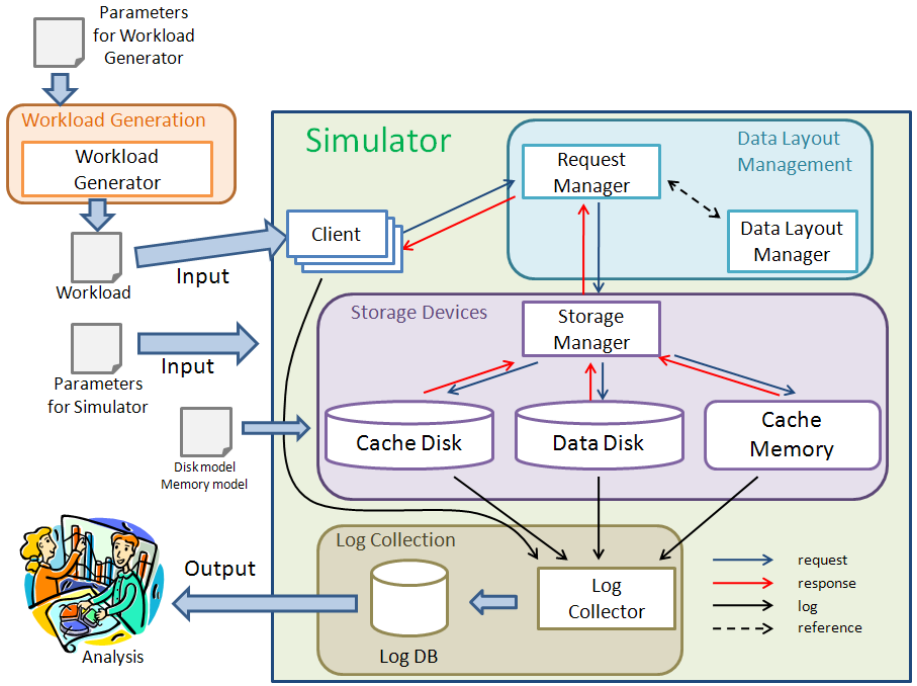

図 3 シミュレーターの全体構成
Fig. 3 Configure of the Simulator

表 3 Workload Generator に渡すパラメータ値
Table 3 Parameters of Workload Generator

Name	Description	value
workload type	ワークロードの種類	synthetic, trace
trace type	ワークロードが初期データかどうかを示す	init, workload
trace file path	トレースファイルのパス	*file path*
arrival time	到着間隔の分布	uniform, poisson
access data	アクセスされるデータの偏りパターン	uniform, zipf
read ratio	読み出し要求の割合	$N \mid 0 \leq N \leq 1$
data size	リクエストのデータサイズ	8KB ∼ 64MB

は，シミュレータが受け付けることが出来る形式のワークロードファイルを生成し，これがシミュレータの実行時にパラメータの 1 つとして渡される．

このコンポーネントによって人工的に生成したワークロードや公開されているトレースファイル等 [9] から生成したワークロードもその形式の違いに影響されず，シミュレータは統一されたインターフェースでワークロードを受け取ることができる．

4.3.2 Data Layout Management

データの配置情報や配置方法を管理するコンポーネントである．本シミュレータはブロック単位での I/O をシミュレートするが，クライアントから発行されるリクエストはブロックよりも粒度の大きいサイズのデータであるため，リクエストに対してどのデータブロックが対応しており，そのブロックはどのストレージに格納されているか（書き込み要求の場合は，どのストレージに書き込むべきか）の情報を管理する．

ディスクへのデータ配置方法はシミュレータに与えるパラメータで設定するため，シミュレータの実行毎に動的に変更できる．RAPoSDA ではディスク配置方法として Chained Declustering を採用しているため，デフォルトではこの設定となっている．その他には，単純に 1 つのディスクにデータを格納してゆき，ディスクの容量がいっぱいになったら次のディスクに格納するという構成や，ランダムで格納先ディスクを決定する方法なども設定できる．

さらに，データ配置方法を管理するモジュールは独立しているため，独自にデータ配置方法を作成することも比較的容易に行える．

また，RAPoSDA ではキャッシュメモリとデータディスク，キャッシュディスクでの連携が必要であるため，Data Layout Management ではこれらのストレージデバイス間でのデータのやりとりなども管理する．

表 4 ディスクドライブモデルのパラメータ
Table 4 Parameters of Disk Drive Model

Name	Description
capacity	ディスクの容量
platters	プラッターの枚数
rpm	ドライブの回転数（RPM）
cacheSIze	キャッシュサイズ
interface	I/O のインターフェース規格
activePower	アクティブ（読み出し/書き込み）時の消費電力
idlePower	アイドル時の消費電力
standbyPower	スタンバイ時の消費電力
spindownEnergy	スピンダウン時の消費エネルギー
spindownTime	スピンダウンにかかる時間
pinupEnergy	スピンアップ時の消費エネルギー
spinupTime	スピンアップにかかる時間

4.3.3 Storage Devices

ディスクドライブやキャッシュメモリなどの記憶装置の振る舞いをシミュレートするコンポーネントである．シミュレートする振る舞いは，ディスクアクセス時のシーク時間やデータ転送時間などの性能に関する情報と，ディスクの回転状況（回転中，回転停止の期間）や状態遷移の状況などの消費電力に関する情報である．

既存のディスクシミュレータでは性能面の振る舞いはシミュレートしても，消費電力に関連する振る舞いはシミュレートしない．また，本提案手法ではディスクドライブとキャッシュメモリの連携によってディスクドライブの省電力化を実現するので，この部分の振る舞いをシミュレートするために新規にシミュレータを作成することが必要であった．

シミュレータで評価するディスクドライブのモデルは，パラメータで指定出来るため，様々なディスクモデルにおける評価が可能である．これを実現する仕組みは，シミュレータに渡すパラメータでどのディスクドライブのモデルを使用するかを指定し，実際のディスクドライブのモデル情報は別のパラメータファイルで用意しておき，実行時に読み込む．4 に示す項目がディスクドライブモデルのパラメータである．

RAPoSDA で用いているデータディスクとキャッシュディスクには，同じディスクモデルや，別々のディスクモデルを設定することが出来る，

4.3.4 Log Collection

ログに記録する情報は以下のようなものである．

- 回転開始/停止を行ったときの判断条件

- リクエスト/レスポンスが発生した時間とその情報
- 各ディスクの状態とその期間

これらの情報は LogCollector という Java のクラスが収集し，LogCollector が必要な情報を DB に書き込む．ある程度情報を集めてから DB へ書き込むため，DB アクセスのオーバーヘッドを低減でき，シミュレータ自体の実行時間の短縮に寄与している．

ログに保存した各ディスクの状態とその期間に関する情報と，ディスクドライブのモデルで設定していた各状態での消費電力の情報を用いて，ストレージシステムの消費電力を求める．データディスク，キャッシュディスクのそれぞれで解析し，いつ，どの程度電力が消費されたかを解析する．

また，各リクエストの発生時間と応答時間の情報もログとして DB に保存しているので，この情報を用いて応答時間やスループットなども解析する．

4.4 シミュレータの動作

シミュレーターの実行手順としては，まず Workload Generator によってワークロードを生成し，生成したワークロードと，ストレージシステムの構成を設定するためのパラメータファイルを入力としてシミュレーターに渡す．そしてシミュレーションが実行され，実行結果が DB に格納される．シミュレーションの終了後に DB から各種の実行結果情報を用いて結果の解析を行う．

5. 関 連 研 究

MAID[4] はキャッシュディスクを用いた手法を提案している．データの局所性があれば非常に良い省電力効果と良い性能維持をもたらすが，データの局所性に依存し過ぎているため局所性のない負荷によってはパフォーマンスが著しく低下してしまうという問題がある．

DRPM[5] では，ディスクドライブの消費電力量はその回転数（RPM）ディスクドライブの回転数（RPM）の関数として表せることを示し，負荷に応じて動的にディスクの回転数を変更することにより，省電力化と性能の維持を実現することを提案した．しかし回転数の動的な変更には技術的な課題も多く，多段階に渡って動的に回転数を変えることができるディスクドライブは未だに実用化はされていない．

EERAID[6] は RAID コントローラーレベルでの動的な I/O スケジューリングとキャッシュ管理ポリシーによって RAID 構成ストレージの省電力化を実現する手法である．RAID 構成に特化している．

PARAID[7] は RAID 構成のストレージシステムに対する省電力化手法であり，データのストライピングのパターンを偏らせてアクセスの無いディスクを作り，そのディスクを停止させる．従来の RAID に対して性能劣化はほとんどなくある程度の省電力効果は得られるが，PARAID も RAID 構成に特化しているため，RAID 構成でないストレージに対しては適用できない．

GRAID[8] も RAID 構成ストレージの省電力化手法の一つである．GRAID では省電力化と信頼性の確保に重点をおいており，RAID10 ベースのディスクアレイを前提としている．通常のディスクドライブの他に，ログディスクというログ格納用のディスクを用いることにより，ミラーリングされたディスクペアの一方のディスクアクセスを抑制する．

本論文で紹介したシミュレータは，ストレージ構成を柔軟に設定でき，ディスクドライブのモデルも個別に設定できるため，MAID や DRPM や PARAID，GRAID 等のシミュレーションも可能である．EERAID は I/O スケジューリングやキャッシュの管理ポリシーを工夫して省電力化を実現するものであるが，我々が作成するシミュレータではキャッシュメモリとディスクドライブの間でのデータ配置等もシミュレートするため，このようなストレージの上位階層レベルでの省電力化手法のシミュレーションにも対応可能である．

6. まとめ及び今後の課題

本論文では，これまでに我々が提案してきたストレージの省電力化手法である RAPoSDA に関して，その省電力効果および性能を評価するために作成したシミュレーションの構成について述べた．シミュレーションはパラメータを設定することで様々なワークロードを生成したり，異なる構成のストレージシステムをシミュレートできることを示した．

今後の課題としては，実際のディスクストレージの消費電力とシミュレーションによる比較を行い，シミュレーション結果との差異を分析し，より現実に近いシミュレーションを実現するための課題を発見することが挙げられる．

さらにデータの多重化構成や，書き込みデータのディスク割り当ての方法などを様々に変化させた時の性能および消費電力の特性を調べ，ワークロード，データ配置方法，と省電力効果の関係性について調べる必要がある．

また，信頼性についての定量的な評価も必要であるが，これは今回作成したシミュレーターでは評価出来ないため，信頼性について検証することも今後の課題である．

謝辞 本研究の一部は，日本学術振興会科学研究費補助金基盤研究 (A)(# 22240005) お

よび文部科学省科学研究費補助金特定領域研究 (# 21013017) の助成により行われた.

参 考 文 献

1) 杉浦利之：データセンターにおける電力供給システムと省電力化 (2008).
2) 引田諭之，横田治夫：プライマリ・バックアップ構成を有効利用したストレージシステムの省電力化手法の提案 (2010).
3) Hsiao, H.-I. and DeWitt, D.J.: Chained Declustering: A New Availability Strategy for Multiprocessor Database Machines, *Proceedings of the Sixth International Conference on Data Engineering*, Washington, DC, USA, IEEE Computer Society, pp.456–465 (1990).
4) Colarelli, D. and Grunwald, D.: Massive arrays of idle disks for storage archives, *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, Los Alamitos, CA, USA, IEEE Computer Society Press, pp.1–11 (2002).
5) Gurumurthi, S., Sivasubramaniam, A., Kandemir, M. and Franke, H.: DRPM: Dynamic Speed Control for Power Management in Server Class Disks, *Computer Architecture, International Symposium on*, Vol.0, p.169 (2003).
6) Li, D. and Wang, J.: EERAID: energy efficient redundant and inexpensive disk array, *EW 11: Proceedings of the 11th workshop on ACM SIGOPS European workshop*, New York, NY, USA, ACM, p.29 (2004).
7) Weddle, C., Oldham, M., Qian, J., Wang, A.-I.A., Reiher, P. and Kuenning, G.: PARAID: A gear-shifting power-aware RAID, *Trans. Storage*, Vol. 3, No. 3, p. 13 (2007).
8) Mao, B., Feng, D., Wu, S., Zeng, L., Chen, J. and Jiang, H.: GRAID: A Green RAID Storage Architecture with Improved Energy Efficiency and Reliability, *MASCOTS*, pp.113–120 (2008).
9) HP Labs, Tools and Traces, *http://www.hpl.hp.com/research/ssp/software/*, (2006).