

## アクセス履歴とデータライフサイクル情報を組み合わせた ストレージ管理機構

小林 大<sup>†1,†2</sup> 田口 亮<sup>†3</sup> 横田 治夫<sup>†4,†1</sup>

情報爆発時代の高度なデータ管理要求を実現するため、ストレージシステムは大規模化、複雑化しており、その管理コストの上昇が問題となっている。特にデータを扱う利用者、アプリケーション、システム管理者が、要求仕様を記述する際、システムの内部動作の記述を強制されるのは非常にコストが高い。一方で、ストレージシステムが自律動作を行う上で、利用履歴のみから性能要求を満たすための自律管理に必要な情報を推定・取得することは常に困難である。

本稿では、近年注目されているデータのライフサイクル情報に着目し、ユーザあるいはアプリケーションソフトウェアからメタデータとして提供される、コンテンツに関するライフサイクル情報と、ストレージシステム側で取得している利用履歴を元に、ストレージ QoS が可能な自律ストレージシステムについて考察する。また、本稿では明示的ライフサイクル情報を、要求性能維持のための複製データ配置管理に応用したコンセプト適用事例により有用性を明確化する。

### Storage Management Architecture based-on Explicit Lifecycle Information and Access History

DAI KOBAYASHI,<sup>†1,†2</sup> RYO TAGUCHI<sup>†3</sup> and HARUO YOKOTA<sup>†4,†1</sup>

Neglecting increasing storage management cost becomes hindering stable development of the information society in information explosion era. In particular, the cost to describe complex storage management policy is burden to the users and administrators because current policy must include storage system-level behavior, while there is just so much the systems can do using only access pattern analysis. We focus on lifecycle information of each content attached by users or applications for information lifecycle management. In this paper, we consider storage QoS management using content lifecycle information generated by combining application-given abstract lifecycle information in the metadata of each content with access trend information in storage management modules. We illustrate the efficiency of this concept with an example that is replica location management using lifecycle information for keep the quality of access latency.

#### 1. はじめに

近年、ストレージ上のデータ量の爆発的な増加に伴い、効率的なデータ管理に対する要求が高まっている。

スループット、レイテンシ等の性能保障やデータ信頼性・可用性の維持のため様々な管理機能を併用し、ユーザから求められるサービス品質 (QoS) を保証する必要がある。現在の大規模なストレージシステムでは、実際のコンテンツ管理動作を、“ポリシー”という記述体系によりストレージ管理ルールとし、システムに与えることで、管理機能を起動する [1] が、多様な管理機能を把握しポリシーを記述することは難しく、ストレージ管理コスト上昇の大きな要因となっている。

管理コスト軽減のため、ストレージシステムに対し計算資源を付加し、自律的な管理を行う自律ストレージシステムが様々な提案されている [2-4]。自律的な管理の精度を向上させるため、システム内資源の利用傾向から自律管理動作を起動させる仕組みが様々な提案されている [5-8]。しかし、利用傾向から逸脱したアクセ

†1 東京工業大学大学院情報理工学専攻  
Department of Computer Science, Graduate School of  
Information Science and Engineering, Tokyo Institute  
of Technology

†2 日本学術振興会特別研究員 DC  
Research Fellow (DC), Japan Society for the Promotion  
of Science

†3 日本放送協会放送技術研究所  
Science and Technical Research Laboratories, Japan  
Broadcasting Corporation

†4 東京工業大学 学術国際情報センター  
Global Scientific Information and Computing Center,  
Tokyo Institute of Technology

スのあった場合の QoS 管理は依然難しい問題である。

一方、近年注目されているのが情報ライフサイクルマネジメント (Information Lifecycle Management:ILM) [9] である。これは、データが生成されてから、活用され、破棄される一連の流れの中での利用状態の変化に着目し、機密情報や法的制限を受けたデータの保存・破棄や、各データの利用頻度の変化を利用したストレージ管理を行うものである。現在の ILM は専用 API を利用したユーザからのストレージへのポリシー記述として実現されている [10]。

また、我々は以前より、多様なコンテンツごとの粒度の細かい情報ライフサイクル管理を行うための基盤として、ECA ルールによって分散ストレージのデータ管理を記述するアーキテクチャを提案している [11]。これまで、多量に付与された管理ルールを効率よく起動、処理するルール評価システムを実現しているが、多種多量のルールを効率よく記述する方法はまだまだ重要な課題である。

そこで我々は、データ利用側から与えられるライフサイクル情報をストレージシステムが収集し、利用傾向情報と併せることで、コストが低く精度が高いストレージ管理ルールの生成を実現することを考える。

本稿では、アプリケーションからのライフサイクル情報収集方式として、コンテンツ登録時にコンテンツ管理ルールとして、生成・利用・破棄といったデータの各利用状態への遷移タイミングを用いることを提案する。そして、与えられたコンテンツ管理ルールとストレージシステム側で保持している各コンテンツの利用傾向と併せ各コンテンツのライフサイクル情報として統合することで、ストレージ管理ルールを発行する枠組みを提案する。

また、その適用例として負荷分散のための複製データ生成破棄管理を考える。生成・利用・破棄の各状態への遷移タイミングがコンテンツ管理ルールとしてアプリケーション側からストレージシステムへ与えられた場合に、応答性能・格納容量共に効率の良い負荷分散管理が実現できることを実験により示す。

本稿の構成を以下に示す。2. で、ストレージ管理に関する関連研究を示す。3. で従来のストレージ管理機構の問題点と、我々が考えるストレージ管理機構のコンセプトを述べ、4. で具体的なストレージ管理への適用例を示す。また、適用例の一部である並列ストレージの複製管理への応用について、シミュレーションプログラムを用いた実験の結果を 5. 示す。最後に 6. でまとめと今後の課題について述べる。

## 2. 関連研究

SAN ベースの分散ストレージシステムに対しポリシー実行アーキテクチャを実現した研究・実装として IBM の STEP がある [9,12]。しかし、依然としてストレージシステム間のデータ移動等のシステム構成に依存したポリシー記述が強いられる。

また我々の提案している自律ディスク [2] はアクティブデータベースにおいて用いられる ECA ルールを記述することにより、柔軟な自律管理を提供する。特に細粒度のライフサイクル管理を実現するため、各コンテンツごとに ECA ルールを付与することを考え多量のルールを高速に処理するルール実効システムを実現している [11]。

アクセス傾向や利用履歴などのストレージ利用情報から行う行うストレージ管理動作として、キャッシュアルゴリズムにおける効率の良い追い出し [5] やプリフェッチ [6,13]、データマイグレーションでは QoS を実現する速度制御 [7,8] 等が研究されている。

## 3. 従来の問題点とアプローチ

本節では、従来のストレージ管理機能発行の枠組みに対し我々の考えるストレージ管理機能発行の枠組みを述べ、比較を行う。

### 3.1 従来のストレージ管理

従来のストレージ管理の枠組みを図 1(a) に示す。

ユーザ、アプリケーションが利用するデータに対しサービス品質要件を要求したい場合、自身あるいはその代行となるシステム管理者が、利用するデータに求められるコンテンツ管理を実現するようなストレージ管理ルールを記述する必要がある。一方システム自身は格納データへのアクセスリクエストを監視し解析することで、ユーザから求められるコンテンツの性質とは独立に、個別にストレージ管理ルールを発行している。

ストレージ管理、特に QoS 制御を行う上で重要となるのがアクセス傾向の変化である。各アクセスに対して効率の良い資源配分を行うために、あらかじめ将来のワークロードを予測する必要がある。現在はデータ利用傾向に対しさまざまな解析手段を試みることで予測を行っているが、急激な傾向の変化への対応や、格納されるコンテンツの利用傾向多様化への対応は不十分である。

### 3.2 明示的情報と利用傾向との統合によるストレージ管理

従来の管理機構に対し、我々のアプローチを以下に

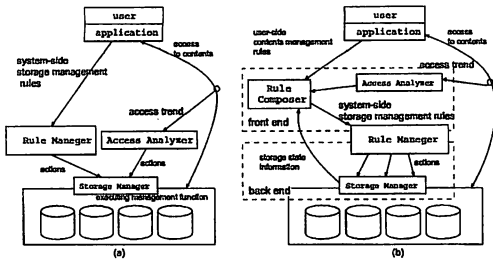


図 1 (a) 従来のストレージ管理 (b) 利用履歴とアプリケーションからの情報を統合したストレージ管理のコンセプト

示す。

従来のデータ利用者から直接ストレージ管理ルールを記述する複雑性を隠蔽するため、従来の管理機構をバックエンドとし、より抽象的な情報(コンテンツ管理ルール)からストレージ管理ルールを生成するフロントエンドを挿入したのが我々の考えるストレージ管理の枠組みである(図 1(b))。

コンセプトの根底にある仮定は(1)「データの利用方法はアプリケーションやユーザが良く知っている」こと、(2)「データの格納方法はストレージシステムが良く知っている」である。そこで、ユーザやアプリケーションから、データの利用に関するコンテンツ管理ルールを入力させ、利用傾向解析によりその情報を洗練した後、現在のシステム状況に応じた、具体的なストレージ管理動作を含むストレージ管理ルールを発行する。ストレージシステムは次の利用状態に適した構成へと自身を組み替え可能となる。

### 3.3 ライフサイクル情報を利用したコンテンツ管理ルール

近年注目されているのが情報ライフサイクルを利用した機密情報や法的制限を受けたデータの保存・破棄やストレージ管理である。ライフサイクルとはデータが生成されてから、活用され、破棄される一連の流れの中での利用状態の変化であり、これは将来のワークロード変化に関して、利用傾向解析とは直交した情報が得られる。

そこで、各コンテンツごとの明示的なライフサイクル情報を、コンテンツ管理ルールとして格納データに添付し利用することで、コンテンツごとの特徴を活かした細粒度のコンテンツライフサイクル管理 [11] を実現するストレージ管理ルールを生成可能となる。

## 4. 適用例

本節では、前節において述べた我々の目指すストレージ管理機構の実現によりストレージ管理がどのよ

うに変化するか、具体事例を上げ議論を行う。

例 1 では、階層構成ストレージの管理記述の変化をあげる。例 2, 例 3 ではアプリケーションが付与でき、コンテンツ管理ルールとして利用可能なライフサイクル情報をあげる。例 4 では、ライフサイクル情報とアクセス解析結果を利用したストレージ管理ルール生成について例を示す。

### 4.1 例 1: ストレージ階層構成管理のためのルール記述

高速ストレージと低速ストレージを用いた階層構成(HSM)は、システム性能向上と構築コスト低減を両立できるため良く使われる。階層構成における重要なストレージ管理として、高速ストレージへ配置するデータの選択がある。

従来型のストレージ管理機構では、ユーザや管理者は高速ストレージの容量や速度を考慮しつつ、自身の利用するデータが要求する性能を満たすように「ストレージ間のデータ移動」をポリシーとして記述している [9]。一方で、ストレージ自身は LRU や ARC [5] のような、コンテンツ利用傾向を利用したアルゴリズムにより、データの利用者からの要求とは関係なく、高速デバイスから低速デバイスへのデータ移動を行う。

これに対し、我々の目指すストレージ管理機構では、ユーザやアプリケーションは「7月13日早朝から最低 30Mbps 保障」といった、利用に関する情報のみを含むコンテンツ管理ルールを与える。ストレージシステムは、コンテンツ管理ルールと、高速ストレージの速度、ドライブ数、他の格納コンテンツの利用情報といったストレージシステム状態に関する情報から、具体的なストレージ管理ルールである「7/13 4:30AM に該当コンテンツを高速ドライブ 2 台に分散してプリフェッチ」を生成し実行する。

ユーザやアプリケーションはストレージシステムの構成や要素性能等に関する知識を要することなく自身の要求をストレージシステムに伝えられる。

### 4.2 例 2: web 上のニュースページ構成ファイル

ユーザから付与可能なコンテンツ管理ルールとして利用可能なライフサイクル情報を考える。速報性のある WEB ページは、あらかじめ記者のデスクトップ上で作成・編集され(作成フェーズ)、その後、監督者らの閲覧・修正(校正)を経て外部公開サーバ(公開フェーズ)からアクセス可能な位置へ配置される。サーバ上では数日の利用期間を経た後、あまり利用されなくなる(保存フェーズ)、後にニュースサービスが終了するまでサーバから利用可能な位置に保管される。

作成フェーズでは頻繁で細かい READ・WRITE,

校正フェーズではメール転送のため少量のファイル単位の READ・WRITE, 外部公開時点で大量の READ アクセスを一定の QoS を保障しながら経た後, 保存フェーズではアーカイブストレージへの移動が促される。以上の情報はユーザやアプリケーションからは容易に得られる情報であるがストレージ側で取得できればワークロード情報や複製管理へと応用できる。

#### 4.3 例 3:電子メール

同様に, アプリケーションから指定可能な, ライフサイクル情報を考える。各個人宛の電子メールは, ほとんどは一度受け取って読まれたら (保管フェーズ) 再利用されないが, 再確認が必要なメールについてはマーキング (再利用フェーズ) が行われたり, メーリングリストからのメールはサブフォルダに階層保存され一定期間ごとにまとめて読まれる。このような同様の種類のデータでもアプリケーションから付与されるメタデータにより利用傾向は大きく異なる。解析的に利用傾向をモデル化する必要なく, アプリケーションからのライフサイクル情報の付与により, その利用傾向を把握することが可能となる。

#### 4.4 例 4: 複製を用いた応答性能保障

与えられたライフサイクル情報を利用した効率の良いストレージ管理規則の生成を考える。ここでは複製を用いた応答性能保障における複製データの生成・破棄を起動するストレージ管理規則を考える。

複数のディスクノードで構成されたネットワークストレージシステムにおいて, 一部のストレージノードへアクセス負荷が集中することは性能低下を招き, 応答性能保障を難しくする [14]。よって, システム構成ストレージノード間での負荷均衡化が重要である。データの複製を複数のノードに配置する (レプリケーション) ことでアクセスリクエストをノード間に分散することが出来る。

##### 4.4.1 これまでの問題

しかし, ネットワークストレージでは各ストレージノードは半導体メモリによるキャッシュを備えていることが多く, レプリケーションによって同一のデータが複数のノード上のキャッシュに残ってしまいキャッシュヒット率を低下させるため, 単純なレプリケーションは期待される性能維持効果を得られない [15]。そのためシステム構成に関する知識に乏しいアプリケーションらが直接ストレージ管理規則を生成するのは難しい。

我々の以前の研究結果 [16] より, 複製による負荷均衡化では低・中頻度利用データへのアクセスを複製間で回送するのが, 各ノード上のキャッシュ利用効率

を最大化するためには望ましいとの結果が得られている。しかし, 多くの利用例では低・中頻度利用データは格納データの大部分を占める [17] ため, 信頼性に必要な冗長性を越えた数の複製を常時作成することはディスク領域を大きく圧迫するため, 好ましくない。

複製データの生成・破棄において重要なのは, (1) 必要な時点のみ (2) 低・中頻度利用データの複製を, 作成することである。

##### 4.4.2 ライフサイクルに基づく生成破棄管理

本節では, コンテンツ管理規則として与えられるライフサイクル情報より (1) 必要な時点を取得し, 与えられたライフサイクルと各コンテンツの利用頻度情報から, 相対的な利用頻度を推測することで複製生成・破棄を行う手法について考察する。

例 2,3 のようにユーザ・アプリケーションは自身の扱うデータの利用率上昇タイミング及び利用率収束タイミングに関する知識を備えていると仮定する。ユーザ・アプリケーションよりデータ生成時に, コンテンツ管理規則として, 利用率上昇および利用率収束を予測した情報 (ライフサイクル情報) を格納する。ストレージシステムはアクセス要求に対しデータを提供すると同時に利用傾向を監視し, メタデータあるいは管理機構内に保存する。そして, 以下のようなストレージ管理規則を生成する。

##### 4.4.3 複製生成

あるコンテンツ A について, ライフサイクル情報として利用開始時点として与えられたタイミングで, ストレージシステムは A に関するアクセス傾向を取り出し, A の利用頻度クラスを決定する。決定された頻度が低または中であつた場合に, ストレージシステムは A の複製を他のディスクに作成するストレージ管理規則を発行する。

##### 4.4.4 複製破棄

A に関する利用終了時点として与えられたタイミングで, 上記と同様にストレージシステムは A に関するアクセス傾向を取り出す。過去のアクセスピークと比較し, 現在の利用頻度が十分収束していると確認できた場合, ストレージシステムは A の複製を破棄するストレージ管理規則を発行する。

## 5. 実験

前節により述べた例のうち, 複製管理手法の挙動についてネットワーク接続された HDD 動作を模擬するシミュレーションプログラム上に実装しその結果を観察した。

## 5.1 実験概要

実験では、シミュレーションプログラム上に構成されたシステムに対し、web アクセスを模したアクセスパターンの READ リクエストを投入し、その最大レスポンスタイム（レイテンシー）を記録する。

測定対象とするシステムは、レプリケーションを行わないシステム (normal)、利用頻度に依らず全てのデータに対しライフサイクル情報を元にレプリケーションを行うシステム (rep\_all)、そして、4.4.2 で述べたように各コンテンツの利用頻度とライフサイクル情報を併用し、低・中頻度利用データのみを対象としたストレージ管理ルールを発行するシステム (ilm+access) の三種類とした。

ここで、本来はユーザまたはアプリケーションによって付与されるライフサイクル情報については投入するワークロードから事前に以下のように機械的に作成したものをを用いた。各コンテンツごとに 1 シミュレーション時間ごとのアクセス数から最もアクセスが集中するピーク負荷を算出した。つづいて、ピーク負荷値に対して 1/5 のトラフィックを記録する最初の時間を利用開始タイミング、1/2 のトラフィックを記録する最後の時間を利用終了タイミングとした。なお、ここで定義したライフサイクル情報を変化させた場合の実験を 5.5 に記す。

また、中頻度利用アクセスは、利用開始タイミングの 1 シミュレーション時間あたりの読み出しリクエストが 20 以下のものとした。これを満たすファイルは格納ファイル数の 97%をしめる。

以上を異なる頻度のワークロード下で行い、それぞれの実験における最大レイテンシーと、増加ファイル数を評価の尺度とした。

## 5.2 実験環境

待ち行列を利用したイベントドリブンのストレージシミュレーションプログラムを構築し実験に用いた。ディスクによるサービス時間については [18] を基にした表 1 にしめすパラメータと前回アクセス時のヘッド位置により算出した。シミュレーション内のシステム構成を表に示す。またノードネットワーク性能は 300Mbps とし、クライアント側ネットワーク性能は無制限とした。

シミュレーションで用いるシステム負荷は、FIFA WorldCup98 Official WEB サイト [19] のアクセスログから抽出した。詳細を表 1 に示す。シミュレーション実行時間削減のためリクエスト数を 1/80 へ無作為に削減している。また高いシステム負荷を実現するために、ファイルサイズについて加工をしている。ワー

表 1 シミュレーション構成設定

simulation parameter	value
Rotational Speed (RPM)	7200
# of surfaces	4
# of sectors per cylinder	2520 - 5184
# of zone	29
Full stroke seek time (msec)	14.7
Single track seek time (msec)	0.8
Head switch time (msec)	1.4
buffer size	2048KB
system component	#
disk node	4
client	2
workload parameter	value
time span (days)	2 (day 24 to 25)
# of request	約 200,000
read:write	10:0
total file size	約 2GB

クロード傾向の詳細については [20] を参考にされたい。異なる負荷のワークロードを実現するため、シミュレーション時間中のアクセスリクエストの発行間隔を実際のログより縮小したワークロードにより実験を行った。この縮小度合いをワークロードの負荷大小を示す指標とする。

## 5.3 複製配置による性能への影響の測定

実験結果を図 2 に示す。横軸は負荷の大きさを表す指標の値である。縦軸は、レプリケーションを行わない場合の最大レイテンシタイムを 1 とした場合の相対最大レイテンシタイムである。

図よりライフサイクル情報と利用頻度情報を用いたレプリケーション (ilm+access) が多くの場合で最大レイテンシタイムを削減できていることがわかる。これは、最大レイテンシを記録する大きなアクセスの発生時に、他のアクセスをその他のディスクへ分散しており、かつ高頻度利用データの複製によりキャッシュ利用率を大きく低下させることがないためである。

## 5.4 格納ファイルサイズの増加

利用開始タイミングを各コンテンツに関して（あらかじめワークロード情報から計算して得た）ピーク時の 1/5 のトラフィックが初めて観測された時間、利用終了タイミングをピーク時の半分のトラフィックが最後に観測された時間としたとき、ファイルの頻度を考慮せずに複製を作成した場合 (rep\_all) で最大で 53%格納データ量が増加した。

一方、利用開始タイミングの利用頻度が 20 [request/hour] 以下、2 [request/hour] 以上のコンテンツを中・低頻度コンテンツであると定義したとき、時間

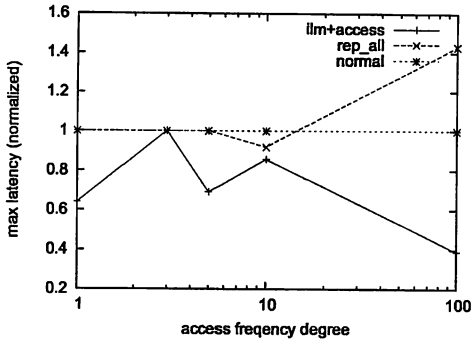


図2 最大レイテンシタイム (レプリケーションなしを1とした)

に寄らず常に複製データを配置した時、最大で66%格納データ量が増加した。

上記のライフサイクルタイミングと頻度を両方考慮し複製生成破棄を行った場合 (ilm+access)、格納データ量は最大で38%増加した。

5.3とあわせ、利用状況とライフサイクルを併用することで、性能と格納効率のいずれについても効率がよいレプリケーションが実行可能であることが得られた。

### 5.5 ライフサイクル情報精度の性能への影響

本実験ではライフサイクル情報はあらかじめ得られた正しいワークロードから算出しているが、現実の情報は予測できない誤りが生じることが多いと考えられる。ここでは、5.2で述べたタイミング定義に用いたピーク時と比較したトラフィック割合の値を変化させ、その傾向を調べた。

まず利用終了タイミングを固定し、利用開始タイミングについて変化させた結果の最大レイテンシタイムを図3(左)に表す。ここで、横軸の値が小さいほど早く複製が作成されている。図より、今回のワークロードに関してはピーク時の1/2を示すタイミングよりも十分早く複製が作成されていればタイミングの差異に依らず負荷分散による性能向上を実現できている。

同様に利用開始タイミングを固定し、利用終了タイミングについて変化させた結果の最大レイテンシタイムを図(右)に表す。横軸の値が小さいほど破棄タイミングが遅い場合を示す。終了タイミングは、遅ければ遅いほど最大レイテンシを低く抑えられることがわかる。しかし、複製削除タイミングを遅くすることはディスク利用効率の悪化を招く。

### 5.6 考察

以上のように、適切なライフサイクル情報と、アクセス頻度情報を併用することで、より効率のよいレプリケーションを実現することが可能となった。

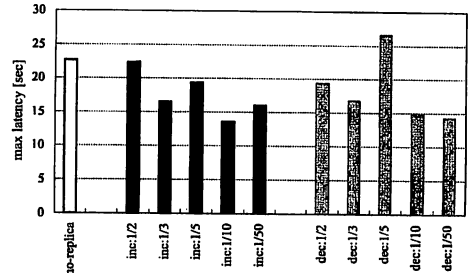


図3 利用開始タイミング情報を変化させたときの最大レイテンシタイムへの影響

今回はコンテンツ間のアクセス量の差やコンテンツ間の関係等の情報は利用していないが、これらについても [6] 等のようにマイニングによって得られる情報と、ユーザやアプリケーションから得られる、WEBにおけるhtmlファイルと構成画像ファイルの関係や、メールファイルとアドレス帳ファイルの関係といった明示的に得られる関係情報を統合することによる他のコンテンツのライフサイクル情報を併用した管理などより高効率のストレージ管理が実現可能となることが十分考えら得る。

## 6. まとめ

本稿では、ストレージ管理コスト削減を目指し、ユーザやアプリケーションから与えられたライフサイクル情報とデータ利用傾向情報を用いたストレージ管理機構について論じた。ユーザ・アプリケーション側とストレージシステム側でそれぞれ高い精度で得られるデータのライフサイクル情報を統合しストレージ管理に利用することで、高い精度のストレージ管理がより低コストで実現できることが考えら得る。また本稿では、適用例として複製データを用いた負荷分散管理に対しライフサイクル情報を利用した複製生成破棄制御を行うことでより性能の良いストレージシステムが実現できることを述べた。

今後は、今回述べたコンセプトを具体的に実現するストレージ管理ルール生成アルゴリズムを構築することが重要である。本稿で示したライフサイクルによるストレージ管理ルール生成は特に利用傾向解析では難しいQoS制御に適していると考えている。よって、大量のデータ移動を伴うことで、一時的な性能低下を引き起こす、データマイグレーションによる負荷均衡化手法のタイミング制御への適用によるQoSがあげられる。また、与えられるライフサイクル情報は非常に曖昧であり現実の挙動との誤差はもちろん大きい。そのずれを、アクセス利用傾向解析により補正すること

で、より高度なストレージ管理が実現できると考えている。また、それらを実装したストレージ管理ルール・コンパイラの実現が課題としてあげられる。

謝辞 本研究の一部は、独立行政法人科学技術振興機構戦略的創造研究推進事業 CREST、情報ストレージ研究推進機構 (SRC)、文部科学省科学研究費補助金特定領域研究 (18049026) および東京工業大学 21 世紀 COE プログラム「大規模知識資源の体系化と活用基盤構築」の助成により行なわれた。

#### 参 考 文 献

- 1) Ashton, L.L., Baker, E.A., Bariska, A.J., Dawson, E.M., Ferziger, R.L., Kissinger, S.M., Menendez, T.A., Shyam, S., Strickland, J.P., Thompson, D.K., Wilcock, G.R. and Wood, M.W.: Two decades of policy-based storage management for the IBM mainframe computer., *IBM Systems Journal*, Vol.42, No.2, pp. 302-321 (2003).
- 2) Yokota, H.: Autonomous Disks for Advanced Database Applications, *Proc. of International Symposium on Database Applications in Non-Traditional Environments (DANTE'99)*, pp. 441-448 (1999).
- 3) Frølund, S., Merchant, A., Saito, Y., Spence, S. and Veitch, A.: FAB: enterprise storage systems on a shoestring, *HOTOS 2003*, Kauai, HI (2003).
- 4) Ganger, G.R., Strunk, J.D. and Klosterman, A.J.: Self\* Storage: Brick-based Storage with Automated Administration., Technical Report CMU-CS-03-178, Carnegie Mellon University (2003).
- 5) Megiddo, N. and Modha, D. S.: ARC: A self-tuning, low overhead Replacement cache, *USENIX File and Storage Technologies Conference (FAST'03)*, San Francisco, CA (2003).
- 6) Li, Z., Chen, Z., Srinivasan, S.M. and Zhou, Y.: C-Miner: Mining Block Correlations in Storage Systems, *the 3rd USENIX Conference on File and Storage Technologies*, pp.173 - 186 (2004).
- 7) Dasgupta, K., Ghosal, S., Jain, R., Sharma, U. and Verma, A.: QoS Mig: Adaptive Rate-Controlled Migration of Bulk Data in Storage Systems., *the 21st International Conference on Data Engineering (ICDE2005)*, pp. 816-827 (2005).
- 8) Lu, C., Alvarez, G.A. and Wilkes, J.: Aque-duct: online data migration with performance guarantees, *Conference on File and Storage Technologies (FAST'02)*, Monterey, CA, pp. 219-230 (2002).
- 9) Beigi, M., Devarakonda, M.V., Jain, R., Kaplan, M., Pease, D., Rubas, J., Ghosal, S., Jain, R., Sharma, U. and Verma, A.: Policy-based information lifecycle management in a large-scale file system., *Sixth IEEE International Workshop on Policies for Distributed Systems and Networks*, pp.139-148 (2005).
- 10) Corpration, E.: EMC Celerra FileMover Functionality., Technical Report.H1685, EMC Corpration (2005).
- 11) Ota, K., Kobayashi, D., Kobayashi, T., Taguchi, R. and Yokota, H.: Treatment of Rules in Individual Metadata of Flexible Contents Management., *International Special Workshop on Databases For Next Generation Researchers (SWOD 2006) in conjunction with ICDE 2006* (2006).
- 12) Verma, A., Pease, D., Sharma, U., Kaplan, M., Rubas, J., Jain, R., Devarakonda, M.V. and Beigi, M.: An Architecture for Lifecycle Management in Very Large File Systems., *22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST'05)*, pp.160-168 (2005).
- 13) Kroeger, T. M. and Long, D. D.E.: Predicting Future File-System Actions From Prior Events., *USENIX Annual Technical Conference*, San Diego, CA, pp.319-328 (1996).
- 14) Simitci, H.: *Storage Network Performance Analysis*, Wiley Technology Publishing (2003).
- 15) Lumb, C.R., Golding, R. and Ganger, G.R.: DSPTF: Decentralized Request Distribution in Brickbased Storage Systems, *Proceedings of ASPLOS'04*, Boston, MA (2004).
- 16) Kobayashi, D., Watanabe, A., Taguchi, R., Uehara, T. and Yokota, H.: An Efficient Access Forwarding Method Based On Caches On Storage Nodes., *International Special Workshop on Databases For Next Generation Researchers (SWOD 2005) In Memoriam of Prof. Kambayashi*, pp.188-191 (2005).
- 17) Hsu, W.W. and Smith, A.J.: Characteristics of I/O traffic in personal computer and server workloads, *IBM Systems Journal*, Vol.42, No.2, pp.347-372 (2003).
- 18) Technologies, H. G.S.: *Deskstar T7K250 Hard Disk Drive Specification*, <http://www.hitachigst.com>, ver. 1.7 edition (2006).
- 19) Laboratory., L. B. N.: The Internet Traffic Archive, <http://ita.ee.lbl.gov/>.
- 20) Arlitt, M. and Jin, T.: Workload Characterization of the 1998 World Cup Web Site., Technical Report:HPL-1999-35R1, Hewlett-Packard

Laboratories (1999).