情報処理学会論文誌　コンピューティングシステム　Vol. 1　No. 3　28–40 (Dec. 2008)

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

西川記史[†1,†2]　茂木和彦[†1]
河村信男[†1]　喜連川優[†2]

データ量の急増にともない，ストレージ管理の容易化とストレージ容量の利用効率向上を可能とするストレージ仮想化技術が広く使われるようになっている．一方，ストレージ仮想化の進展はアプリケーションからストレージの物理リソースを隠し，I/O性能障害の原因の発見やその解決策の立案を困難なものとしている．本研究では，ストレージの主要なアプリケーションであるデータベース管理システムを対象に，ストレージ仮想化環境において，DBMSのSQLやデータがどのストレージリソースをどの程度使用しているか，その内訳を可視化する技術を開発した．本技術を用いてDBMSのI/O性能障害を診断する評価を実施した結果，ストレージのRAIDグループにおけるDBデータ配置の競合を発見でき，従来とは異なる新たな対策方法をとることが可能であること，これによりI/O性能障害の対策に要する時間を削減できることを確認した．

Development and Evaluation of Storage-DBMS Integrated Diagnosis Technology for I/O Performance Problem

Norifumi Nishikawa,[†1,†2] Kazuhiko Mogi,[†1]
Nobuo Kawamura[†1] and Masaru Kitsuregawa[†2]

Growth of data drives the spread of a storage virtualization which simplifies the management of storage and improvements of storage capacity utilization. Storage virtualization, however, hides the physical resources of storage form applications and makes difficult to solve I/O performance problem. To solve this problem, we developed an I/O performance monitor which can visualize the detail of I/O response time and I/O count of storage resource by DBMS data and SQL. We also evaluated by solving the storage performance problem of DBMS which uses virtualized storage by using this technology. As a result, we confirmed that our approach could find a DB data placement confliction in storage RAID group. We also confirmed that out approach could provide a new troubleshooting method, and reduce the I/O performance troubleshoot time.

1. はじめに

ディジタル世界に毎年追加される情報は，2006年では161エクサバイト，2010年には988エクサバイトに達すると見込まれる[1)]．このデータ量の急増は人手によるデータ管理の限界を超えているという事実を直視する必要がある．また，大量のデータを管理するスキルを要する人材はきわめて枯渇しており，管理をいかに容易化するかが，今後のデータ処理システムを利用していくうえで重要な課題となっている．とりわけデータ処理システムの主要な構成要素であるストレージの管理の容易化は重要な課題である．

ストレージの管理を容易化する技術として，アプリケーションからハードディスクなどのストレージ物理リソースを隠し，それらを論理的に提供する仮想化が知られている[2),3)]．特にSNIAテクニカルチュートリアル[2)]では，仮想化技術がストレージの信頼性，性能，および容量効率の改善に寄与することが述べられている．

一方，ストレージの仮想化はストレージ物理リソースをアプリケーションから隠すと同時に，アプリケーションとハードディスクの間の論理的な階層数を増加させる．この結果，I/O性能障害が発生した場合に，ボトルネックとなったストレージリソースとI/Oを発行したアプリケーションやそれが使用するデータとの対応関係の把握が困難になり，I/O性能障害の診断や対策が長時間化するという課題があった．

この課題を解決するため，我々はストレージの主要なアプリケーションであるDBMSを対象として，ストレージ仮想化環境におけるI/O性能障害の診断支援技術を開発した．開発した技術の特徴は，ストレージリソースの応答時間やI/O数を，DBMSのSQL文やデータごとに詳細化して可視化する点である．これにより，DBMSの個々の処理とストレージリソースとの性能上の対応関係の把握が可能となり，ストレージ仮想化環境下でのI/O性能障害の対策に要する時間を短縮することが期待できる．

本論文では，開発技術によりDBMSの処理やデータとストレージリソースとの性能上の

†1 株式会社日立製作所
Hitachi, Ltd.
†2 東京大学
The University of Tokyo

対応関係の把握が可能であることを示すとともに，開発技術を用いた評価実験により，ストレージが仮想化された環境下においても，DBMS の I/O 性能障害の対策時間の短縮ができることを確認する．

以下，2 章ではストレージの仮想化と DBMS の概要について，3 章では I/O 性能障害診断支援の関連技術について述べる．4 章では I/O 性能障害診断支援技術の設計および実装方式について説明する．さらに 5 章では本技術を用いて I/O 性能障害の診断を行い，その有効性について述べる．また，6 章では本技術を実システムに適用するにあたり考慮すべき点について述べる．

2. ストレージの仮想化

今日，ストレージの仮想化には多くの技術が使われている[2]．**図 1** は仮想化されたストレージを使用する，比較的小規模のストレージ階層および DBMS 階層を示している．

図 1 の例では，ストレージ層は，データを記憶するハードディスク，それらを複数個まとめて RAID を構成する RAID グループ，RAID グループ上に構築されホストに対して仮想的なディスク領域を提供する論理ユニット（LU），ストレージが提供する LU をホスト側でマッピングした物理ボリューム，1 つ以上の物理ボリュームをグループ化したボリュームグループ，ボリュームグループから仮想的なボリュームを切り出した論理ボリュームの 4〜6 層からなる．

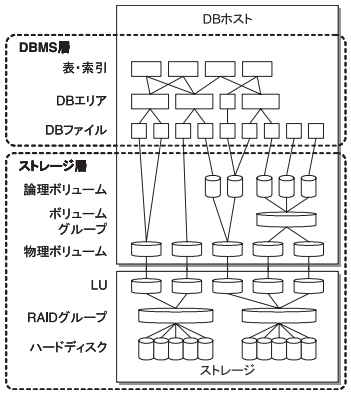

図 1　ストレージおよび DBMS 階層
Fig. 1　Storage and DBMS hierarchy.

また，主要 DBMS[4)–6)] では，DBMS 層は表・索引，それらを格納する DB エリア，および DB ファイルを構成する DB ファイルの 3 階層からなる．以下，本論文ではこれら DBMS 層内部の階層を構成する要素を DB データと呼ぶ．

このように，ストレージの仮想化により，比較的小規模な環境においても DBMS の表または索引からハードディスクまでは 7〜9 階層を経由することが分かる．また，仮想化層の中には，ボリュームグループに見られるように，下位の複数のリソースをまとめて 1 つのプールを構成し，そこから上位の層に対して論理的なリソースを提供する機能を有する層も存在する．これにより，上位層のリソースが階層のどのリソースを使用しているかを把握することが困難となっている．

なお，ここであげた例は比較的小規模なものであるが，より大規模なシステムでは階層数のさらなる増加により，上位リソースと下位リソースの対応関係の把握がさらに困難になることは想像に難くない．

3. 関 連 研 究

本研究は，ストレージ仮想化環境においてストレージリソースと DBMS の処理やデータとの対応付けを可能とすることにより，ストレージと DBMS の性能管理，特に I/O 性能障害の診断および対策を容易化することを目的としている．ストレージおよび DBMS における性能管理を容易化する技術として，ストレージ管理ソフトによるストレージリソース管理，DBMS による DBMS 性能管理，およびストレージと DBMS が連携した管理方式がある．以下それらの概要について述べる．

3.1 ストレージリソース管理

ストレージリソース管理は，OS やストレージ装置，ストレージエリアネットワーク（SAN）などから情報を収集し，構成や容量，性能に関するレポーティングや分析，プロビジョニングなど，ストレージリソースに関する管理全般を行う技術である[7)]．

特に近年，ストレージリソース管理はその管理対象をアプリケーション層にまで拡大しつつある．たとえば，HP Storage Essentials では，管理対象を DBMS のテーブルや索引などにまで拡大しており，DB のテーブルからストレージのハードディスクまでの各リソースの対応関係を収集し，それらをトポロジ形式で可視化することが可能である[8)]．また，Akkori は，OS のファイルシステムからストレージの RAID グループまでの構成情報および性能情報を収集し，それらの対応関係をトポロジ形式で表示するとともに性能値に応じて色分

け[⋆1]して表示する[9]．これにより，ストレージのハードディスクからアプリケーションが使用するデータまでの対応関係やそれらの性能を把握することが可能となる．

また，Pollack らは，ストレージを利用するアプリケーションの処理とリソースの対応関係を用いることにより，ストレージにおける性能問題の発生原因がどの処理に起因するかを抽出する技術を発表している[10]．

これらの技術はストレージのリソースやアプリケーションを管理の対象としている．このため，前述のようなプールを含む層が存在する場合，アプリケーションがプール層より下位のストレージリソースをどの程度使用しているか，およびこれらのアプリケーションが使用するデータがどのストレージリソースに配置されているかを把握することができない．

3.2　DBMS 性能管理

一方，DBMS の性能管理においては，DBMS が行う処理の挙動に関する情報を収集しそれに基づき性能を管理する研究が行われている．Dias らは，DBMS のアクティブな接続（セッションと呼ぶ）に関する詳細な挙動を定期的にサンプリングし蓄積，その情報を用いて性能障害の診断やチューニングを自動的に実施する研究について報告している[11]．特に，セッションが I/O を行っている場合は，どの DB データのどの部分に I/O を行っているかを記録しており，これにより DBMS の処理がどの DB データにアクセスするかを把握することが可能となっている．しかし，上記の技術は DBMS 層のみに閉じた技術である．

3.3　ストレージ・DBMS 連携

ストレージと DBMS との連携により，ストレージ層から DBMS 層までの end-to-end で性能管理を行う技術も見られる．DBMS 管理ツールである Oracle Enterprise Manager では，商用 DBMS の管理ツールから同じく商用のストレージ管理ツールを呼び出す仕掛けを提供している[12]．これにより，DB ファイルやサーバのボリュームとストレージデバイスとの対応関係，およびストレージのポート，コントローラ，およびディスクの性能を DBMS 管理ツール上に表示することにより，end-to-end での性能管理を実現している[12]．しかし，本技術はは 3.1 節に示した技術と同様にストレージリソースとその対応関係に基づいて性能管理を行っている．

また，Oracle と EMC によるデータベース管理ソリューションに示される技術は，性能管理技術ではないが，DB データを DBMS のデータ分割機能を用いて分割し，それらを複数の異なるストレージリソースに配置している[13]．これにより，ストレージリソースと DB データとを対応付け，ストレージ管理機能による DB データのライフサイクル管理を可能としている．しかし，本技術では，DB データの分割とストレージリソースとの対応付けやそれに基づくデータの配置を管理者が注意深く設計をする必要がある．

⋆1 閾値に基づき，正常は緑，警告は黄，エラーは赤色でリソースを表示．

4.　I/O 性能障害診断支援ツール

I/O 性能障害の診断は，DBMS の処理に影響を与えているストレージリソースの発見，およびストレージリソースに対して負荷を与えている処理とその理由の発見というステップからなる．これらのステップを効率良く支援するためには，ストレージ仮想化環境下においても DBMS の各処理が行う I/O とストレージリソースとの対応関係を把握することが不可欠である．

一方，従来の技術は，DBMS の処理，データ，およびストレージリソースごとに性能情報を収集している．DBMS の I/O 性能障害を診断・解決するためには，障害を起こした DBMS の処理が，負荷が高いストレージリソースをどの程度使用しているかを明らかにする必要がある．従来の技術ではこのような情報の取得は困難であり，ストレージの I/O 性能障害の診断や対策立案の妨げとなっていた．

そこで，我々は，ストレージ層のアドレスレベルの対応付けを用いて DBMS の個々の処理が行う I/O の I/O 先を解決，すなわちどの処理がどのストレージリソースに I/O を行ったかを明らかにすることで，ストレージ仮想化環境下における I/O 性能障害の診断を支援するツールを開発した．

我々は，本ツールの利用者が DBMS およびストレージにおいて発生する I/O 性能障害の発見，原因の推定，対策案の立案を行うスキルを有すると想定している．これは，スキルを有する利用者が診断を実施しても，前述の対応関係の把握の容易性や収集可能な情報の差が，対策案の内容および対策に要する時間に差を与えると考えたためである．

以下，まず本ツールの設計方針について述べ，次にツールの実装について述べる．

4.1　I/O 性能障害診断支援ツールの設計方針

ストレージ層のアドレスレベルの対応関係を用いて DBMS の個々の処理が行う I/O の I/O 先を解決するためには，DBMS の処理別 DB データ別の I/O 先を収集する機能，DB データとストレージリソースとのアドレスレベルの対応関係を収集し DBMS の処理ごとの I/O 先となったストレージリソースを解決する機能が必要である．

また，I/O 性能障害の診断においては，障害が発生した時点の性能情報のみではなく，システムが正常に稼動していた時点などその前後の性能情報も参照しながら多様な観点から

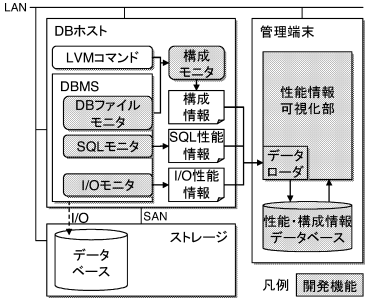

図 2　I/O 性能障害診断支援ツールの実装
Fig. 2　Implementation of Diagnostic Tool.

診断を行う．そのため本ツールでは，前述の I/O 先情報および対応関係を蓄積・検索する機能を設けることとした．

さらに，収集された性能情報は一般に膨大な量になる．このため，これらの性能情報を効率良く可視化することも必要である．そこで，本ツールの開発にあたっては，前述の I/O 性能障害の診断に適したビューを提供する可視化機能を設けることとした．

以下，各機能の実装について説明する．

4.2　I/O 性能障害診断支援ツールの実装

I/O 性能障害診断支援ツールは，**図 2** に示すように I/O モニタ，SQL モニタ，DB ファイルモニタ，構成モニタ，性能・構成情報データベース，データローダ，および性能情報可視化部の各機能からなる．以下，これら各機能について説明する．

(1)　I/O モニタ

I/O モニタは，DBMS が行った I/O を監視し，DBMS のプロセスごとに，当該プロセスが実行している SQL 文識別子，I/O を行った表および索引，DB ファイル名および DB ファイル内のオフセット，I/O サイズおよび I/O 応答時間のトレースを収集する．これにより，DBMS のプロセスが，どの SQL を実行中にどの DB データのどの部分にアクセスしたかを把握することが可能となる．また，この情報と後述のアドレスレベルのストレージ構成情報とを突き合わせることにより，DBMS がどのストレージリソースに I/O を行ったかを求めることが可能となる．I/O モニタが出力する情報を**表 1** に示す．ここで，DBMS のプロセスには，DBMS が持つバッファの非同期書き出し処理やログ管理処理などのバックグラウンドプロセスも含む．I/O モニタ機能の実装個所は DBMS 内部である．

表 1　I/O モニタが収集する情報
Table 1　Gathered information of I/O monitor.

情報名	説明
I/O 開始・終了時刻	DBMS が I/O を開始および終了した時刻（DBMS 層で計測）
DB ホスト名	DBMS が稼動するホスト名
プロセス名	I/O を発行した DBMS プロセス名
プロセス識別子	I/O を発行した DBMS プロセスの識別子
I/O 種別	Read または Write の区別
表・索引識別子	DBMS が I/O を行った表または索引の識別子（I/O 先が表または索引の場合のみ記録）
DB ファイル名	DBMS が I/O を行った DB ファイル名
オフセット	DBMS が I/O を行った I/O 先オフセット（DB ファイル内）
I/O サイズ	DBMS が行った I/O で Read または Write されたデータサイズ
応答時間	I/O 終了時刻と I/O 開始時刻の差分

表 2　SQL モニタが収集する情報
Table 2　Gathered information of SQL monitor.

情報名	説明
SQL 開始・終了時刻	DBMS が SQL 文の実行開始および終了時刻（DBMS 層で計測）
トランザクション ID	トランザクションの識別子
DB ホスト名	DBMS が稼動するホスト名
プロセス名	SQL を実行した DBMS プロセス名
プロセス識別子	SQL を実行した DBMS プロセス識別子
CPU 使用時間	SQL 文の実行要求を受け付けてから完了するまでに使用した CPU の時間
排他待ち時間	SQL 文が排他待ちとなった時間

(2)　SQL モニタ

SQL モニタは，DBMS が実施した処理の CPU 使用時間および排他待ち時間，応答時間を SQL 実行ごとに収集する．SQL モニタが収集する情報を**表 2** に示す．SQL モニタは，SQL 文開始・終了時刻，CPU 使用時間，および排他待ち時間を，表 1 に示す情報と突合せするために使用する情報である DB ホスト名，DB プロセス名およびプロセス識別子とともにトレース形式で出力する．

I/O 性能障害の診断においては，管理者はまず応答時間の悪化が I/O に起因するか否かを

表 3　ストレージ構成情報
Table 3　Storage configuration information.

情報名	説明
ホスト	DBMS が稼動するホスト
ボリューム	上記ホストのボリューム
境界オフセット	1 つの論理ボリュームが複数の物理ボリュームに対応している場合の境界
ストレージ	DB を記憶するストレージ
LUN	ホストの物理ボリュームに割り当てられたストレージの LU の番号
RAID グループ	LU が配置されている RAID グループ

表 4　DB 構成情報
Table 4　DB configuration information.

情報名	説明
ホスト	DBMS が稼動するホスト
スキーマ	DBMS が管理する DB に定義されているスキーマ
表・索引	DBMS が管理する DB に定義されている表と索引
DB エリア	表または索引を格納している DB エリア
DB ファイル	DB エリアを構成するファイル
オフセット	DB ファイル領域先頭オフセット
サイズ	DB ファイル領域サイズ
ボリューム	DB ファイルが配置されるボリューム

判断する必要がある．SQL モニタが出力する情報と I/O モニタが出力する情報を用いることで，管理者は I/O 時間の増加が性能障害の原因か否かを切り分けることが可能となる．ここで，SQL 文の実行 1 回あたり I/O が複数回発行される可能性があることから，本ツールでは I/O に関する情報と SQL に関する情報を分けて出力する仕様とした．SQL モニタ機能の実装個所は，I/O モニタと同様 DBMS 内部である．

(3)　**構成モニタおよび DB ファイルモニタ**

構成モニタおよび DB ファイルモニタは，DB ホストのボリュームからストレージの RAID グループまでの対応関係，および DB の表または索引から DB ファイルまでの対応関係を収集し，それらをストレージ構成情報および DB 構成情報として構成情報ファイルに出力する．

まず，構成モニタは，DBMS のディクショナリを参照して DB ファイルの一覧を求め，これと OS が提供する raw デバイスの定義を用いて DB ファイルが配置されている論理ボリュームまたは物理ボリュームを求める．DB ファイルが論理ボリューム上に配置されている場合，構成モニタは LVM コマンドを用いて論理ボリュームと物理ボリュームのエクステントの対応関係を取得し，それに基づき論理ボリュームの LBA のどこからどこまでが，どの物理ボリュームに記憶されているかを求める．その後，構成モニタは SCSI コマンドを用いて物理ボリュームとストレージの LU および RAID グループの対応関係を取得，これらの情報を突き合わせ，**表 3** に示す情報を作成する．

次に，構成モニタは再度 DBMS のディクショナリを参照し，表および索引，DB エリア，DB ファイルの一覧とそれらの対応関係を取得する．次に，OS が提供するシステムコールを用いて DB ファイルが論理ボリュームまたは物理ボリュームのどの論理ブロックアドレス（LBA）に配置されているかを求める．さらに，構成モニタは DB ファイルモニタを用いて DBMS 内部の管理情報を参照し，DB ファイルの領域のボリューム上の配置を計算，**表 4** に示す情報を作成する．なお，本論文で示す提案方式は raw デバイスを対象としており，DB をファイルに保存する方式は対象としていない．

ストレージ構成情報の特徴は，1 つの論理ボリュームが複数の物理ボリュームに対応する場合にその境界を示すボリューム内のオフセットを設けている点である．また，DB 構成情報の特徴は DB ファイル領域のボリューム内オフセットを有している点である．これらの情報により，DB ファイルのどの部分がどの物理ボリュームに配置されているかを把握することが可能となる．

構成モニタの起動，および構成モニタが出力する構成情報の性能・構成情報データベースへのロードは，それぞれ管理者が実施することを想定している．これは，実際のトラブルシュート作業においては，性能障害が発生したシステムが存在する場所と診断を行う場所が異なる場合が存在し，それぞれ異なる管理者が実施する場合があるためである．

(4)　**性能・構成情報データベース**

性能・構成情報データベースは，I/O モニタ機能が収集した性能情報および構成情報を蓄積するデータベースである．多様な観点からの診断を支援するため，本ツールでは関係データベースを用いて性能・構成情報データベースを実装した．

(5)　**データローダ**

データローダは，I/O モニタが収集した性能情報と構成情報を用いて DBMS のどの処理がどのストレージリソースに I/O を行ったかを解決した後，当該情報を性能情報データベースにロードする．以下，この情報を I/O 性能情報と呼ぶ．またデータローダは，SQL モニタが収集した情報，および構成情報モニタが収集した情報も性能・構成情報データベースにロードする．以下，特に前者の情報を SQL 性能情報と呼ぶ．構成情報の付加にあたり，

データローダは DB ファイルとボリュームの対応関係，およびボリュームと LU の対応関係を示すオフセット情報を用いて DBMS の I/O 先オフセット（DB ファイル内）から I/O が行われた LU および RAID グループを求める．I/O 性能情報と構成情報を突き合わせて性能・構成データベースにロードすることにより，DBMS の各処理が行った I/O とストレージリソースの対応関係を把握することが可能となる．

しかし，このままでは I/O 数に OS 層やストレージのキャッシュにヒットした I/O とヒットしない I/O が混在する．そこで，我々は個々の I/O の応答時間に着目し，ある時間より応答時間が長い I/O をキャッシュミス，すなわちハードディスクにアクセスした I/O とカウントすることで，I/O がハードディスクまで到達したか否かを切り分けることとした．

(6)　**I/O 性能情報可視化部**

I/O 性能情報可視化部は，DBMS の処理に影響を与えているストレージリソースの発見，およびストレージリソースに対して負荷を与えている処理とその理由の発見に必要な情報を提示するビューを管理者に提供する．I/O 性能情報可視化部が提供するビューは以下のとおりである．

ストレージリソース別 I/O 時間可視化　ストレージリソース別 I/O 時間可視化ビューは，DBMS が実行したトランザクションの I/O 時間について，ストレージリソースごとの応答時間を管理者に提示する．本ツールは，性能・構成情報データベースに蓄積された I/O 性能情報中の応答時間を，ストレージリソース別に集計することにより本情報を生成する．これにより，DBMS の処理に影響を与えているストレージリソースの発見を支援する．

I/O 発行元別 I/O 数可視化　I/O 発行元別 I/O 数可視化ビューは，ストレージリソースに対して I/O を発行した DBMS プロセスごとの I/O 数，およびストレージリソースに配置されている DB データごとの I/O 数を管理者に提示する．本ツールは，性能・構成情報データベースに蓄積された I/O 性能情報のレコードを，DB データ別リソース別に集計することにより本情報を求める．これにより，ストレージリソース別 I/O 時間の可視化によって発見したストレージリソースに対して負荷を与えている DB データの発見を支援する．

時系列情報の提示　時系列情報表示は，ストレージリソース別 I/O 時間や I/O 発行元別 I/O 数を時系列で管理者に提示する．これにより，上記情報の推移の把握を支援する．

5. I/O 性能障害診断支援技術の評価

本技術の有効性を確認するため，開発したツールを用いて DBMS の I/O 性能障害を診断する評価実験を実施した．

5.1 実験環境

実験環境のハードウェア構成を**図 3** に示す．本実験では，TPC-C ベンチマーク[14] 相当の DB を管理する DBMS が稼動する DB ホストと TPC-H ベンチマーク[15] 相当の DB を管理する DBMS が稼動する DB ホストが 1 台のストレージを共有し，さらにこれらの DB ホストと TPC-C 相当のベンチマークプログラムを稼動する負荷生成ホストを，LAN を用いて接続する構成とした．負荷生成ホストには DELL Power Edge（Pentium D 3 GHz × 2，メモリ 2 GB，RedHat7.2 カーネル 2.4），DB ホストには DELL Power Edge（Xeon X3210 2.13 GHz × 4，メモリ 4 GB，CentOS 4 カーネル 2.6），ストレージは（株）日立製作所製 SANRISE9500 シリーズを，DBMS には同社製の HiRDB/Single Server 07-01 改造版を使用した．

DBMS の表および索引からそれらが記憶されるストレージの RAID グループまでの対応関係を**図 4** に示す．TPC-C 用ホストには DBMS01 が稼動し，当該 DBMS が管理する DB の表および索引は，それぞれ 2 種類，合計 4 種類の物理ボリュームに格納する構成とした．

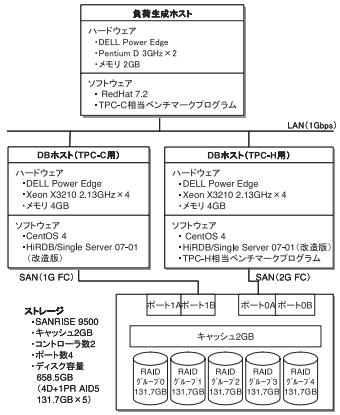

図 3　ハードウェア構成
Fig. 3　Hardware configuration.

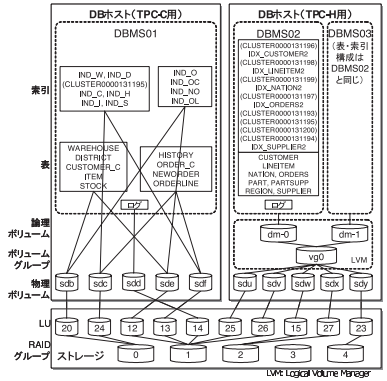

図 4　データ配置
Fig. 4　Data mapping.

また，DBMS のログをこれらの物理ボリュームとは異なる物理ボリュームに配置した．また，TPC-H 用のホストには，DBMS02 および DBMS03 の 2 個の DBMS インスタンスが稼動し，それぞれの DBMS が管理する表および索引，ログは論理ボリューム dm-0 および dm-1 にそれぞれ格納する構成とした．DBMS02 および 03 のデータは，それぞれ 1 つの DB ファイルから構成される 1 つの DB エリアに格納している．

論理ボリューム dm-0 および dm-1 は TPC-H 用 DB ホストの LVM が提供するボリュームグループ vg0 上に構築した論理ボリュームである．vg0 はさらに 5 つの物理ボリュームを連結した構成としている．これは，ボリューム割当てを簡素化するために使用するボリュームプール機能を用いた環境でも本技術が有効であることを確認するためである．また，DBMS01，DBMS02，DBMS03 はストレージの RAID グループ 1 を共有している．これは，ストレージの仮想化によりストレージのハードウェア境界が見えない状態でボリューム割当てを行った状況を模擬している．

5.2　ストレージ層における診断

まず，我々は，前節に示した実験環境を用いて，TPC-C ベンチマーク相当および TPC-H ベンチマーク相当の処理を同時に実行した場合の TPC-C ベンチマーク相当の処理のトランザクション応答時間の遅延を起点とした性能障害の診断を実施した．本実験ではまず TPC-C ベンチマーク相当の処理を開始した後，性能障害の発生を模擬するため 10 分後に DBMS02 および 03 の TPC-H ベンチマーク相当のプログラムを実行しストレージに対する I/O 負荷を増加させている．

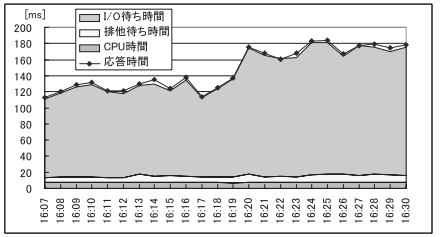

図 5　トランザクション応答時間内訳（従来方式）
Fig. 5　Transaction response (conventional).

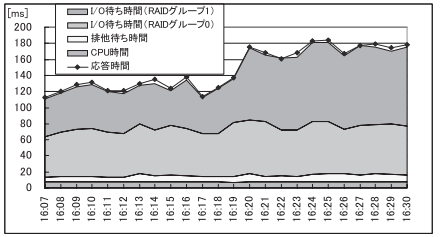

図 6　トランザクション応答時間内訳（開発方式）
Fig. 6　Transaction response (developed).

(1)　**性能障害が発生したストレージリソースの発見**

まず，従来の方式を用いた場合と開発した方式を用いた場合に，応答時間が増加したストレージリソースの発見がどのように異なってくるかについての評価を実施した．**図 5** に従来方式を用いた場合の TPC-C ベンチマーク相当の処理のトランザクション応答時間の内訳を，**図 6** に提案方式を用いた場合の同処理のトランザクション応答時間の内訳をそれぞれ示す．

従来方式では，管理者は DBMS 層およびストレージ層の情報を突き合わせて診断を実施する．図 5 は DBMS 管理ツール[16] を用いて取得したトランザクション応答時間の内訳（CPU 時間，排他待ち時間，I/O 待ち時間）である．これにより，管理者は，応答時間の増加の原因が CPU 時間の増加にあるのか，排他待ち時間の増加にあるのか，あるいは I/O 待ち時間の増加にあるのかを知ることができる．図 5 からは，16 時 18 分前後を境に I/O 応答時間が延びていることが読み取れる．

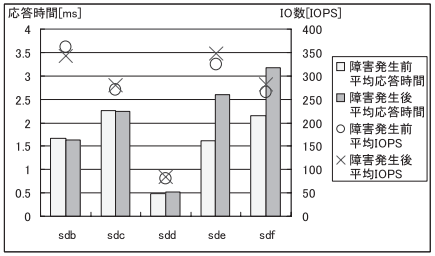

図 7　物理ボリューム別応答時間および IO 数
Fig. 7　Response time and IO count of physical volumes.

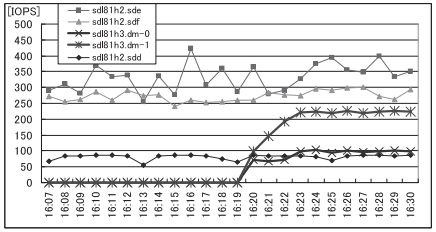

図 8　RAID グループ 1 への I/O 数（従来方式）
Fig. 8　Details of I/O for RAID Group 1 (conventional).

次に，管理者は，どのストレージリソースの応答時間が延びたのかを明らかにし，それを解消するための対策をとることになる．そこで管理者は，ストレージ層の管理ツール[17] を用いて，TPC-C 用 DB ホストの性能障害発生前後の物理ボリュームごとの I/O 数および応答時間を確認する．この結果を**図 7** に示す．物理ボリューム sde および sdf の応答時間が 60%以上増加しているにもかかわらず，同ボリュームへの I/O 数の増加は 5%程度であることから．管理者は当該ボリュームの応答時間の悪化がトランザクションにおける I/O 待ち時間増加の主原因と判断することができる．

この後管理者は，図 4 に示すデータ配置を参照し，物理ボリューム sde および sdf がともにストレージの RAID グループ 1 上に配置されていること，および RAID グループ 1 上にはさらに別の DBMS（DBMS02 および DBMS03）の DB が配置されていることを確認する．これらの結果とストレージ層で収集した RAID グループ 1 の使用率を確認することにより，管理者は RAID グループ 1 上で何らかの競合が発生したことが性能障害発生の原因ではないかと推測することが可能となる．

以上示したように，従来方式においては，ストレージの RAID グループ上の競合を発見するためには，DBMS 層，ストレージ層の性能情報と構成情報とを人手で突き合わせて判断する必要があり，ボトルネックの推定作業は非効率的であるといえる．

一方，開発方式では I/O モニタおよび SQL モニタを用いて収集した性能情報を用いることにより，CPU 時間，排他待ち時間に加え，I/O 待ち時間の内訳を，ストレージリソースごとに詳細化して可視化することが可能である（図 6）．これにより，管理者は本情報を参照するのみでどのストレージリソースの応答時間が増加したかを容易に把握することが可能となる．図 6 に示した例では，16 時 18 分前後を境に I/O 応答時間が延びた原因が，RAID グループ 1 の応答時間の増加によるものであることを容易に把握することが可能である．

(2)　ストレージリソースへの I/O が増加したボリュームの発見

次に我々は，RAID グループ 1 の応答時間増加原因の明確化における開発技術の有効性を確認するため，従来方式と開発方式について，RAID グループ 1 に対するボリュームごとの I/O 数の調査方法を比較した．従来方式では，障害診断を実施するために，ストレージ層で収集した情報および構成情報を用いている．なお，本実験では，I/O がハードディスクにまで到達したかどうかを判定する閾値として 1 ms を用いた．

従来方式を用いた場合の RAID グループ 1 に対するホストのボリュームごとの I/O 数の推定値を**図 8** に示す．ここで，ホスト sdl81h2 は TPC-C ベンチマーク相当の処理を実施した DB ホスト，sdl81h3 は TPC-H ベンチマーク相当の処理を実施した DB ホストである．DB ホスト sdl81h3 ではボリュームグループ vg0 の存在により論理ボリュームと RAID グループの対応関係の把握ができないことに注意を要する．性能障害の対策を行うためには，DBMS02 と DBMS03 のそれぞれからどの程度 I/O が発行されているかを知る必要がある．このためには，RAID グループ 1 に対する I/O 数を論理ボリューム dm-0，dm-1 ごとに把握する必要がある．しかし，従来方式では前述の理由により DBMS02 と DBMS03 のどちらの処理が RAID グループ 1 に対してどの程度の量の I/O を発行しているかを直接知ることはできない．このため，本実験では，管理者は dm-0 と dm-1 の I/O 数に比例した I/O が RAID グループ 1 に対して発行されているとの仮定の下，論理ボリュームまたは物理ボリュームごとの I/O 数を図 8 に示すように求めた．ここで，我々は，dm-0 および dm-1 が発行した I/O 数の比として TPC-H 用 DB ホストで計測された dm-0 および dm-1 の I/O 数の比を用いた．また，RAID グループ 1 に対する DBMS02 および DBMS03 からの I/O 数の算出には，TPC-H 用 DB ホストの物理ボリューム sdu および sdv の I/O 数の総和を

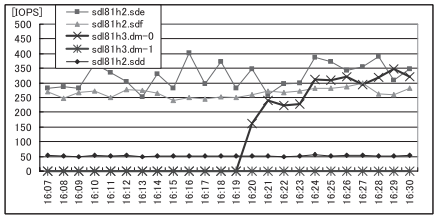

図 9　RAID グループ 1 への I/O 数（開発方式）
Fig. 9　Details of I/O for RAID Group 1 (developed).

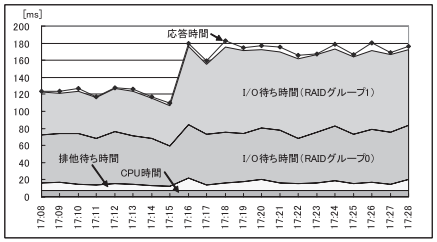

図 10　トランザクション応答時間内訳（開発方式）
Fig. 10　Details of transaction response (developed).

用いた．図 8 では，16 時 18 分頃より DB ホスト sdl81h3 の論理ボリューム dm-0 と dm-1 の I/O が増加していることが読み取れる．そのため，本実験では管理者は特に dm-1 を使用する DBMS3 のトランザクションからの I/O が増加したと推定することになる．

一方，開発方式では RAID グループに対するボリュームごとの I/O 数を直接可視化することが可能である．このため管理者は，どのボリュームからの I/O が増加したかを容易に把握することができる．開発方式を用いた場合の RAID グループ 1 への I/O 数の内訳を**図 9** に示す．

図 9 では図 8 と異なり，16 時 18 分前後を境に増加したのは DB ホスト sdl81h3 の論理ボリューム dm-0 であることが分かる．この結果は驚くべきものであり，本方式のように RAID グループに対する I/O 数の内訳を可視化できなければ，誤った判断を行う可能性があることを示している．対して本方式は，仮想化されたボリュームを使用する環境においても，対策を実施すべき DBMS を求めることが可能である．

5.3　DBMS 層と連携した診断

次に我々は，DBMS レベルの統計を用いて，同様の診断を実施した．実験環境のハードウェア構成は図 3，データ配置は図 4 とそれぞれ同一であるが，DBMS 層との連携を強調するため，本評価では DBMS01 と DBMS02 のみを起動し，DBMS03 は停止させた状態で性能障害を発生させている．本実験においても，前節の実験と同様，まず TPC-C ベンチマーク相当の処理を開始した後，10 分後に DBMS02 の TPC-H ベンチマーク相当のプログラムを実行し性能障害の発生を模擬している．

(1)　性能障害が発生したストレージリソースの発見

従来方式では前節の場合と同様，DBMS 層およびストレージ層双方の情報を用いてボトルネック個所を推定する．しかし，応答時間が悪化した RAID グループの発見は前節の例と同様非効率である．

開発技術を用いた場合の TPC-C ベンチマーク相当のトランザクションの応答時間の推移を**図 10** に示す．この図より，管理者は 17 時 16 分を境に I/O 応答時間が増加し，特に RAID グループ 1 に対する I/O 時間が増加していることを容易に把握可能である．この場合も前節同様，I/O 性能情報と SQL 性能情報を用いることにより，トランザクション応答時間における CPU 使用時間，排他待ち時間，および I/O 待ち時間のストレージリソースごとの内訳を単一画面で可視化することが可能となったためである．

(2)　ストレージリソースに対する I/O が増加した DB データの発見と I/O 数の調整

次に我々は，RAID グループ 1 の応答時間増加原因の明確化における開発技術の有効性を確認するため，従来方式と開発方式の診断方法を比較した．本実験でも，I/O がハードディスクに到達したかどうかを判定する閾値として 1 ms を用いた．

従来方式として DBMS 層の情報のみを用いた診断，ストレージ層の情報のみを用いた診断，および DBMS 層およびストレージ層それぞれにおいて取得可能な情報を組み合わせた診断を示す．

DBMS 層のみの情報を用いた場合，管理者は RAID グループ 1 に I/O を発行する可能性のある表および索引を，図 4 に示すデータ配置を参照し求める．本例では，DBMS01，DBMS02 の全表からの I/O が RAID グループ 1 に到達する可能性があることが分かる．その後管理者は表および索引ごとの I/O 数の推移を DBMS 管理ツールを用いて調査する．**図 11** は，DBMS01 および DBMS02 の表および索引ごとの I/O 数である．図より，17 時 16 分を境に，DBMS02 の表 ORDERS，LINEITEM，および索引（CLUSTER0000131197）からほぼ均等な数の I/O が発生していることが読み取れる．その後管理者は，これら 2 つの表と 1 つの索引を他の RAID グループへ移動する計画を立案，実施することにより問題の解

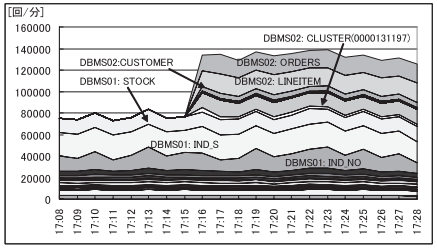

図 11　DBMS の表・索引別 I/O 数（従来方式）
Fig. 11　Number of I/O of tables and indexes (traditional).

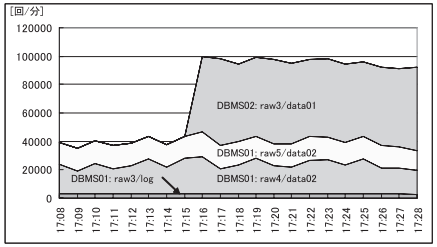

図 12　DB ファイル別 I/O 数（従来方式）
Fig. 12　Number of I/Os of DB files (traditional).

決を図る．ここで，従来方式では，表および索引単位の I/O 数の取得は可能であるが，それがどの RAID グループに対する I/O であるかを取得することはできないことに注意されたい．我々は，RAID グループ 1 に対する I/O の内訳が DBMS01，DBMS02 それぞれにおいて図 11 に示す内訳と等しいとの仮定の下，対策案を立案している．

DBMS 層のみの情報を用いた診断における他のアプローチとして，表・索引ごとの調査ではなく，表および索引を格納する DB エリアを構成する DB ファイルごとの診断も存在する．**図 12** に DB ファイルごとの I/O 数を示す．なお，DBMS01 の DB ファイルは物理ボリューム上に配置されており，かつ物理ボリュームと RAID グループとの対応関係は図 4 より把握可能であるため，図 11 と異なり図 12 は DBMS01 については RAID グループ 1 上に存在する DB ファイル raw3/log，raw4/data02，raw5/data02 のみ記載している．

図 12 より，17 時 16 分を境に，DBMS02 の DB ファイル raw3/data01 から毎分約 55,000 回の I/O が新たに発生していることが読み取れる．この結果より，管理者は DBMS02 の DB ファイル raw3/data01 を他の RAID グループへ移動する計画を立案，実施することにより問題の解決を図る．

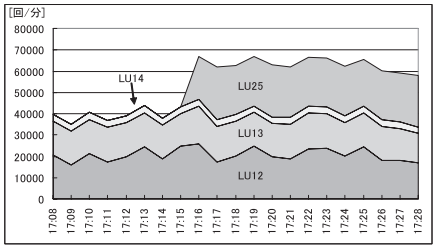

図 13　LU 別 I/O 数（RAID グループ 1）（従来方式）
Fig. 13　Number of I/Os of logical units (RAID Group 1) (traditional).

ストレージ層のみの情報を用いた場合，管理者はストレージ管理ソフト[18] を用いて，まず RAID グループ 1 上の LU ごとの I/O 数を調査し，それに基づき RAID グループの負荷均衡を図るべくボリュームの移動計画を立案・実行する．**図 13** は RAID グループ 1 上に定義された LU ごとの I/O 数の推移であるが，17 時 16 分前後を境に LU25 に対する I/O が発生していることが読み取れる．その後管理者は，この情報をもとに，LU25 上のデータの他の RAID グループ上の LU への移動を計画，実施し問題の解決を図る．

DBMS 層およびストレージ層それぞれにおいて取得可能な情報を組み合わせた診断では，表および索引の I/O 数と LU の I/O 数，または DB ファイルの I/O 数と LU の I/O 数を用いる．しかし，前者の組合せでは，表および索引に対する I/O がどの RAID グループに対して行われているかを取得することはできないため，管理者は RAID グループ 1 に対する DBMS02 からの I/O の内訳が図 11 と同じとの仮定の下，ORDERS 表，LINEITEM 表，および索引（CLUSTER000013119）を移動するか，またはストレージ層のみの情報を用いた場合と同様に LU25 上のデータを移動することにより問題の解決を図ることになる．後者の組合せにおいても．DBMS02 の DB ファイル raw3/data01 からの I/O が増加したとの判断に基づき，当該 DB ファイルまたは LU25 上のデータの移動を計画・実行することになる．

一方，開発方式では，ボリュームごとの I/O 数に加え，RAID グループ 1 に記憶されている DBMS データごとの I/O 数の可視化が可能である．ここで，DBMS データとは DB の表あるいは索引，DB エリア，DB ファイルを指す．これにより，従来方式と比較してより木目細かな対策案の立案が可能であることが期待できる．

図 14 は，RAID グループ 1 に記憶された DB の表および索引ごとの I/O 数の推移である

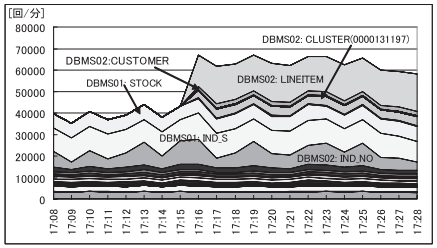

図 14　DBMS の表・索引別 I/O 数（開発方式）
Fig. 14　Number of I/O of tables and indexes (developed).

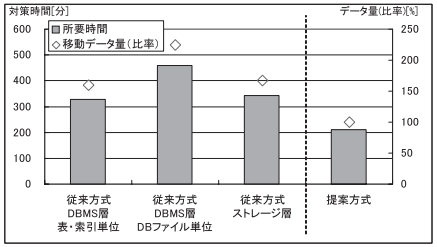

図 15　対策時間およびデータ移動量
Fig. 15　Data migration time and amount of data.

が，トランザクション応答時間が悪化した 17 時 16 分前後を境に，DBMS02 の LINEITEM 表に対して毎分約 16,000 回の I/O が，DBMS02 の ORDERS 表の索引である（CLUSTER0000131197）に毎分約 4,000 回の I/O が，CUSTOMER 表に対して毎秒分約 2,500 回の I/O が発生していることが分かる．逆に，DBMS 層のみの情報を用いた場合に計測された，ORDERS 表に対する I/O は計測されていない．提案方式を用いて RAID グループ 2 および 4 の I/O 数を確認した結果，ORDERS 表への I/O はすべて RAID グループ 2 に対して行われていることを確認できた．

これにより，管理者は，DBMS02 の LINEITEM 表および CUSTOMER 表，ORDERS 表の索引が RAID グループ 1 に配置され，なかでも特に LINEITEM 表に対する I/O が DBMS01 に影響を与えていたことを把握することができる．本情報を用いることにより，管理者は LU や DB ファイルと比較してデータ量が少ない LINEITEM 表の移動，および I/O 数がさほど多くない表または索引の移動を回避しつつ障害を解決可能である．

(3)　対策時間の比較結果

我々は，提案方式および従来方式を用いた場合のそれぞれの対策（データ移動）に要する時間を比較した．この結果を**図 15** に示す．ここで，データの移動先は未使用の RAID グループ 3 上に作成した LU 上である．計測時間に LU 作成時間は含んでいない．従来方式の表・索引単位，DB ファイル単位，および提案方式は DBMS のデータ一括移動ツールを，ストレージ層のデータ移動は対象ボリュームを lvm コマンドを用いて移動した．

図に示すように，従来方式はデータ移動に約 310 分から 460 分要したのに対し，提案方式は約 210 分とデータ移動に要する時間を約 3 割から最大 5 割まで大幅に短縮している．図 15 には提案方式のデータ移動量を 100 としたデータ移動量を示しているが，データ移動時間はほぼこのデータ移動量に比例していた．図 15 において DB ファイル単位のデータ移動に要する時間が長いのは，DBMS02 の DB ファイル raw/data01 のサイズが複数の LU にまたがるほど大きいためである．

DBMS 層の情報のみ，あるいは既存の DBMS 層の情報とストレージ層の情報を組み合わせた場合と提案方式との観測結果の差は，データへのアクセスの偏りやストレージ層におけるデータの配置の偏りの把握の可否に起因すると考えられる．従来方式は表および索引，DB ファイル，LU 単位で I/O を計測するのみで RAID グループに対する I/O の内訳を計測していない．このため，特にストレージ層でボリュームの仮想化が行われた場合は，偏りが存在してもそれを把握することは困難である．一方，提案方式は I/O の内訳を計測するため前述の偏りの把握が可能となっている．このことが，移動しなければならないデータの絞り込みを可能とし，対策時間の差を生じさせたと考える．

5.4　モニタの影響

最後に，我々は開発機能の影響を調べるため，TPC-C ベンチマーク相当の処理を用いてトランザクションのスループットおよびレスポンス時間を比較した．実験環境は図 3 および図 4 に示したとおりであるが，本比較では TPC-H 用 DB ホストは使用していない．比較の結果，スループットは 2.2%減，レスポンス時間は 5.9%増であり，実用的なレベルであると考える．

6. 考　　察

本章では，開発技術の適用にあたり考慮すべき点について考察する．

6.1　ボリュームのストライピング

開発した技術は，ストレージプール層における下位ボリュームの集約方式として，ボリュームの順次結合を前提としている．これにより，ボリュームの開始・終了アドレスと I/O 先

アドレスを比較することで，プール層の下位に位置するどのボリュームに I/O が行われたかを把握している．

一方，仮想化されたストレージにおいては，ボリュームの容量効率の向上や負荷バランスを目的として，ストライピングが用いられるケースもある．このようなケースでは，前述のようなボリュームの開始・終了アドレスを用いた方式ではプール層の下位のどのボリュームに I/O が行われたかを把握することができない．

これに対しては，アドレス変換の計算式の使用，あるいはストライピングの単位となるブロックごとのマッピング情報を用いることで，I/O 先ボリュームの解決を図ることが可能であると考える．

6.2　キャッシュの影響

本実験で用いたシステムは，OS 層やストレージ層にキャッシュを持つ．そこで，我々は，I/O がストレージのハードディスクにまで到達したか否かを判定するために応答時間を閾値として用い，本実験ではその閾値を 1 ms と設定した．この値は，我々の実験環境における経験値から求めた値であるが，より一般的なシステムに適用するためにはこの決定方法も検討する必要があると考える．たとえば，Kochut らはストレージの上位層からみた I/O の応答時間の分布を用いることでボトルネック個所を推定する手法[19] について述べているが，同様の手法を用いることにより，閾値を求めることが可能であると考える．

6.3　構成モニタと DB ファイルモニタの適用範囲

本論文で提案した構成モニタおよび DB ファイルモニタは，DB ファイルが直接論理ボリュームまたは物理ボリューム上に配置される，raw デバイスを用いる方式を対象としている．一方，小規模なシステムでは DB ファイルを OS が提供するファイル上に配置する方式も行われるが，提案方式はこのような方式には対応していない．OS が提供するファイル上に DB ファイルを配置する方式に対応するためには，DB ファイルが OS ファイルのどの LBA に配置されているか，および OS ファイルが論理ボリュームまたは物理ボリュームのどの LBA に配置されているかを解決する必要がある．

また，今回の実装では LVM や RAID コントローラの種類を限定しているため，他の実装にそのまま適用することはできない．しかし，他の種類の LVM や RAID コントローラであっても，LVM の論理ボリュームと物理ボリュームの LBA レベルの対応関係，および物理ボリュームが配置されている LU や RAID グループの情報取得が可能であれば，それらを取得する部位の変更のみで図 1 に示す構成の範囲において提案方式を適用することが可能であると考える．

7. ま と め

我々は，ストレージの主要なアプリケーションである DBMS を対象に，ストレージの仮想化層の下位に位置するストレージリソースの応答時間，およびストレージリソースに対する DBMS の処理や DB データごとの I/O の量の把握・可視化を可能とする技術を開発した．

開発した技術を用いて DBMS の性能障害の診断を実施した結果，ストレージ層における診断では，LVM 層を介した場合においても I/O 数が増加した論理ボリュームを正しく識別できることを，DBMS 層を含めた診断ではストレージの RAID グループにおける DB データ配置の競合を発見でき，従来とは異なる新たな対策方法をとることが可能であること，これによりデータの移動量および対策に要する時間を削減できることを確認した．また，本技術による性能オーバヘッドも実用的な範囲に収まるであることが確認できた．これらの結果は，本技術がストレージ仮想化環境における性能障害の診断や対策案の立案の容易化に貢献することを示していると考える．

謝辞　本研究の成果の一部は，文部科学省リーディングプロジェクト e-Society 基盤ソフトウェアの総合開発「先進的なストレージ技術」の支援により得られたものである．

参 考 文 献

1) IDC: The Expanding Digital Universe, IDC White Paper (2007). http://www.emc.com/about/destination/digital_universe/pdf/Expanding_Digital_Universe_IDC_WhitePaper_022507.pdf
2) SNIA: *Storage Virtualization, SNIA Technical Tutorial*, SNIA (2004).
3) 喜連川優：ストレージネットワーキング技術—SNIA ストレージ技術者認定プログラム準拠，オーム社 (2005).
4) IBM: IBM DB2 9 (2006). http://www-06.ibm.com/jp/software/data/db2/v9/
5) Oracle: Oracle Database 11g (2007). http://www.ora-cle.com/lang/jp/database/index.html
6) Microsoft: Microsoft SQL Server (2005). http://www.microsoft.com/japan/sql/default.mspx
7) Russell, D. and Passmore, R.E.: *Magic Quadrant for Storage Resource Management and SAN Management Software*, Gartner RAS Core Research Note G00146578 (2007).
8) HP: HP Storage Essentials Oracle Viewer — Overview & Features (2006). http://h18006.www1.hp.com/products/storage/software/e-suite/wf05-oracle.html

9) Akorri: Akorri BalancePoint Overview (2006). http://www.akorri.com/products-overview.htm
10) Pollack, K.T. and Uttamchandanim, S.M.: Genesis: A Scalable Self-Evolving Performance Management Framework for Storage Systems, *Proc. 26th IEEE International Conference on Distributed Computing Systems* (2006).
11) Dias, K., Ramacher, M., Shaft, U., Venkataramani, V. and Wood, G.: Automatic Performance Diagnosis and Tuning in Oracle, *Proc. 2005 CIDR Conference* (2005).
12) Oracle: Oracle Enterprise Manager 10g System monitoring Plug-in for EMC Symmetrix DMX System (2007). http://www.oracle.com/technology/products/oem/pdf/ds_emcsymmetrixdmx.pdf
13) EMC: Oracle+EMC データベース管理ソリューション (2007). http://www.emcinfo.jp/promo/solutions/oracle/pdfs/Oracle_Solution.pdf
14) Council, T.P.P.: *TPC BENCHMARKTM C Standard Specification Revision 5.9* (2007). http://www.tpc.org
15) Council, T.P.P.: TPC BENCHMARKTM H (Decision Support) Standard Specification Revision 2.6.1 (2007). http://www.tpc.org
16)（株）日立製作所：*HiRDB* (1994). http://www.hitachi.co.jp/Prod/comp/soft1/hirdb/
17) Musumeci, G.D.: Unix システムパフォーマンスチューニング，オライリー・ジャパン (2003).
18)（株）日立製作所：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 (1994). http://www.hitachi.co.jp/products/it/storage-solutions/products/software/hsms/htm/index.html
19) Kochut, A. and Bobroff, K.N.: G.Kar: Management Issues in Storage Area Networks: Detection and Isolation of Performance Problems, *Network Operations and Management Symposium 2004* (2004).

(平成 20 年 4 月 15 日受付)
(平成 20 年 8 月 22 日採録)

西川　記史（正会員）

1989 年神戸大学工学部計測工学科卒業．1991 年同大学大学院工学研究科計測工学専攻修士課程修了．同年（株）日立製作所入社．システム開発研究所にてストレージ管理ソフトウェアの研究開発に従事．現在同研究所主任技師，および東京大学大学院情報理工学系研究科電子情報学専攻．1998 年度情報処理学会山下記念賞受賞．

茂木　和彦（正会員）

1992 年東京大学工学部電気工学科卒業．1997 年同大学大学院工学系研究科情報工学専攻博士課程修了．博士（工学）．1998 年（株）日立製作所入社．現在同社ソフトウェア事業部にて高性能データ処理技術の研究開発に従事．

河村　信男（正会員）

1981 年愛媛県立松山工業高等学校卒業．同年（株）日立製作所入社．ソフトウェア事業部にてデータベースシステムの開発に従事．現在同事業部主管技師．

喜連川　優（副会長）

1978 年東京大学工学部電子工学科卒業．1983 年同大学大学院工学系研究科情報工学専攻博士課程修了．工学博士．同年同大学生産技術研究所第三部講師．現在，同教授．2003 年より同所戦略情報融合国際研究センター長．データベース工学，並列処理，Web マイニングに関する研究に従事．1997〜1998 年電子情報通信学会データ工学研究専門委員会委員長，1999〜2002 年 ACM SIGMOD Japan Chapter Chair，2002〜2003 年本会理事，2003 年本会フェロー．2008 年本会副会長．日本データベース学会理事，SNIA-J 顧問．IEEE TCDE Asian Coordinator，ACM SIGMOD Advisory Boad Member．文部科学省特定領域研究「情報爆発 IT 基盤」領域代表．