SAN を用いた仮想化ボリュームとその自律的負荷分散機構

横山　有一[†]　　合田　和生[†]　　喜連川　優[†]
†東京大学生産技術研究所　〒153-8505　東京都目黒区駒場 4-6-1
E-mail:　†{yokoy, kgoda, kitsure}@tkl.iis.u-tokyo.ac.jp

あらまし　データウェアハウスや大規模 WWW システムなどの登場により、エンタープライズシステムにおける情報量は飛躍的に増大し、情報システムにおけるストレージの重要性は増している。ハードディスクの容量単価の価格は下落し、管理コストが大きな負担となっている。ストレージシステムの高い性能と安定した運用が求められる中、大規模ストレージシステムを人手で管理することは困難になりつつある。

本論文では、ストレージエリアネットワーク(SAN)を用いることにより、複数のストレージ装置から仮想化ボリュームを構成する機構について述べる。機構内では、変化する負荷に適応してデータ配置を変更することにより、自律的な負荷分散を行った。

キーワード　ストレージエリアネットワーク, 仮想化機構, 負荷分散機構

Autonomous Load Balancing for Virtualized Volume Using SAN

Yuichi YOKOYAMA[†]　　Kazuo GODA[†]　and　Masaru KITSUREGAWA[†]
†Institute of Industrial Science, The University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan
E-mail:　†{yokoy, kgoda, kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract　As we can see in the evolution of data warehouse and WWW, data volume in the enterprise systems has increased dramatically and the importance of storage systems has been risen. While the cost per capacity of disks in the market has decreased, management cost has taken more significance. Administrators of large-scale storage systems have difficulties to satisfy the requirements of users, high performance and reliable services. In this paper, we describe our virtualizer that organizes virtualized volumes from physical devices distributed over storage area network (SAN.) The mechanism includes autonomous load-balancing mechanism, which can change block assignment to balance workloads between devices.

Keyword　storage area network (SAN), storage virtualization, load balancing

1. はじめに

膨大なデータを保存するデータウェアハウスやマルチメディア情報を提供する大規模 WWW システムなど大量のデータを蓄積するアプリケーションの登場により、エンタープライズシステムにおけるデータ容量は急速に増加している。

システム全体のデータの増加により、容量追加のためのストレージ増設や、障害への対応、性能改善のためのチューニングなどの管理コストが大きな問題となっている。

従来サーバに従属して接続されていたストレージ装置を、ストレージエリアネットワーク（Storage Area Network: SAN）で接続し、サーバ群で共有するストレージ技術が普及してきた。その中でも、SAN に接続された複数のストレージ装置上の記憶空間を統合し、仮想的な記憶空間を構成するストレージの仮想化技術[1]が注目を集めている。

高い性能が求められるトランザクション処理システムや意思決定支援システムでは、ストレージに関する多くのパラメータをチューニングする必要がある。従来サーバの管理として一括に行われていたこれらのチューニング作業を、仮想化ストレージ装置内において自律的に行うことにより、かかるコストを大幅に削減することが期待されている。また、SAN を用いた新しいアクセス方式により、アプリケーションの性能を改善する技術も期待されている[7]。

ストレージ仮想化技術により、サーバからストレージの物理構成を隠蔽し、ストレージ管理をストレージシステム内で閉じて行うことにより、ストレージの性能を最大限に発揮するとともに、サーバ管理を軽減することにより資源をより優先性の高いタスクに集中させることが可能となる。

これら SAN を用いることによる効率的な記憶空間

の管理及び障害対策に関しては、既にいくつかの製品が登場している。しかし、SAN がシステムの性能に与える影響に関する研究は少ない。

本論文では、一般のファイルシステムやトランザクション処理システムで支配的な、小さいサイズのアクセス要求が頻繁に発行される IO 負荷に関して、SAN 上の仮想化ボリュームの性能を改善する手法を提案する。仮想化ボリューム中のブロック毎の統計情報を収集し、ブロックの物理ディスクへの配置を動的に変更することにより、物理ディスク間の負荷不均衡を解消し、IO 性能を改善する。これらの制御は、ストレージシステム内で閉じて行なわれ、仮想化ボリュームの性能を最大化することを目指す。さらに、提案手法を SAN 結合 PC クラスタ上で実装を行ない、人工的な負荷を用いて評価する。

本論文の構成は以下の通りである。2 では仮想化ボリュームを構成するストレージ仮想化機構に関して述べる。3 では、SAN 結合 PC クラスタにおけるストレージ仮想化機構の設計と実装に関して説明し、4 で自律的負荷分散機構の設計と実装に関して述べる。さらに 5 では実験結果に関して述べ、最後に 6 でまとめを行う。

2. SAN 技術とストレージ仮想化機構

SAN は従来サーバに接続されていたストレージ装置を、専用のネットワークにより接続し、サーバ群により共有する技術である。ストレージ仮想化機構は、SAN 上に接続されているストレージ資源を共有ストレージプールとみなし、その中から仮想化ボリュームを構成してサーバに提供する機構である。サーバはあたかも物理ディスク領域にアクセスするように仮想化ボリュームにアクセスできるため、ストレージの物理的な構成に依存しないボリュームアクセスが可能である。仮想化ボリュームでは、サーバからストレージの物理構成を隠蔽することにより、ストレージの中で閉じた管理をすることが可能になる。

仮想化ストレージ空間は、サーバからは無限の容量を有した記憶空間とみなすことができる。サーバは個々のストレージ装置の記憶空間にとらわれることなく必要とする領域を確保することができるため、ストレージ全体の領域を効率的に使用することができる。また、サーバ群の必要とする記憶空間を集約することにより、容量の断片化を防ぐ意義も大きい[2][3]。

また、記憶装置として不可欠な障害対策を仮想化機構内で行うことができる。SAN 構成ストレージシステムのサーバフリーバックアップでは、サーバを経由しないデータのバックアップが可能である[4]。また、

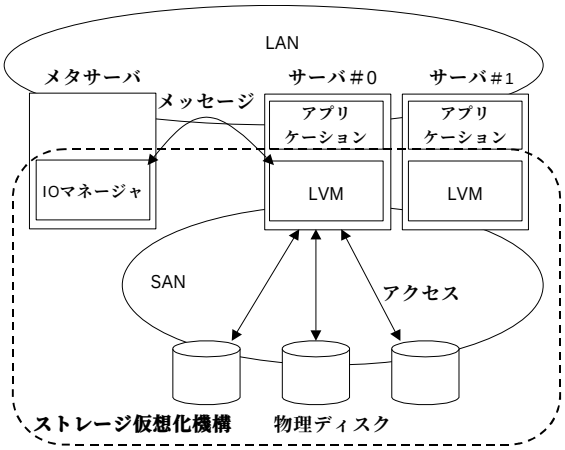


図 1: ストレージ仮想化の機構

ポイントインタイムコピーによるスナップショット生成は、データのある時点での一貫性のある断面を複製する機能を提供する[5][6]。これらの機能により、バックアップやスナップショットの管理を容易にすると共に、その負荷が通常運用へ与える影響を低く抑えることが可能となる。

図 1 に、StorageTank や SafeFILE/Global に見られるような、メタサーバ、論理ボリュームマネージャ（Logical Volume Manager: LVM）を用いた仮想化機構の構成を示す。LAN 上に、メタサーバ、アプリケーションが実行されるサーバが接続され、ストレージは SAN によって共有されている。各サーバに設けられた LVM は、メタサーバと連携する形でアプリケーションにストレージアクセスを提供する。このとき、仮想化ボリュームの論理的一貫性を保つためメタサーバが用いられる。メタサーバの LVM 管理のメッセージ等は LAN 経由で行われ、ストレージのデータへのアクセスは SAN 経由で行われる。

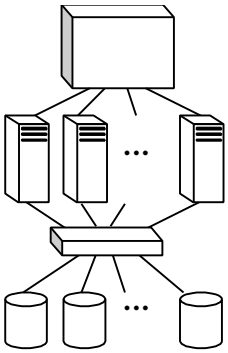
3. ストレージ仮想化機構の設計と実装

3.1. SAN 結合 PC クラスタ

本論文では、大規模 SAN システムを想定した実験システムとして、SAN 結合 PC クラスタを用いる。SAN 結合 PC クラスタは、32 台の PC サーバが Fast Ethernet によって接続されており、さらに、32 台のディスクが Fibre Channel によって接続されている（図 2）。各 PC の諸元を表 1 に示す。

表 1: PC の諸元

CPU	PentiumIII 800MHz
Memory	128MB
OS	Solaris 8(10/00)
HBA	Emulex LP8000
NIC	Intel EtherPro/100+



ファストイーサネットスイッチ
日立 Blackdiamond

32PCs
PentiumIII 800MHz
メモリ 128MB

ファイバチャネルスイッチ
Brocade SilkWorm

FC ディスク
Ciprico Fibre Store
(Seagate Cheetah 18GB)

図 2: SAN 結合 PC クラスタ

3.2. SAN 結合 PC クラスタによる実装

本論文では、ストレージ仮想化機構の実験環境として、LVM、メタサーバ、及び管理 PC を実装した（図 3)。本来 LVM は OS のカーネル空間で動作すべきであるが、実験の容易のため、ユーザ空間のライブラリとして実装した。アプリケーションはライブラリの提供する API を用いることにより仮想化ボリュームにアクセスすることが可能である。また、メタサーバは、専用の PC にデーモンプロセスとして実装している。メタサーバは、仮想化ボリュームへや物理ディスクへのアクセス統計情報を収集する。現状の実装では、書き込み操作と移送の一貫性のためのロック機構は実装していない。

さらに管理 PC を実装し、メタサーバが収集した統計情報をグラフィカルに管理者に表示することができる。

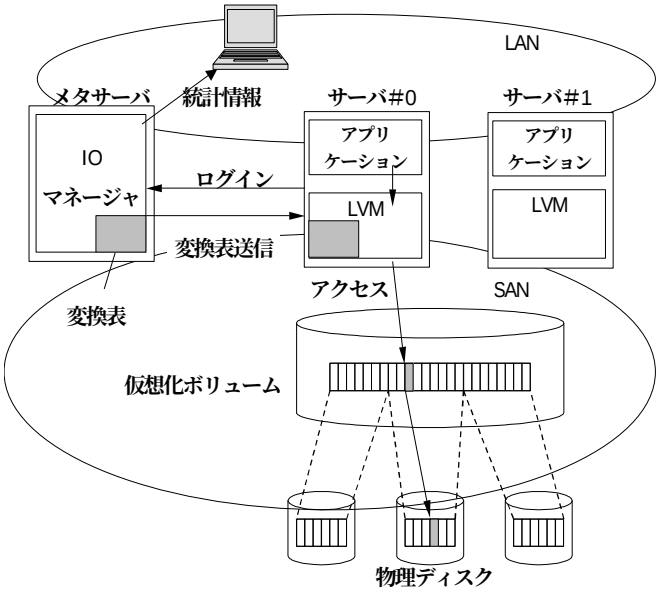


図 3: SAN 結合 PC クラスタによる実装

3.2.1. 仮想化ボリュームへのアクセス

仮想化ボリュームの記憶空間は、固定長ブロック単位で物理ディスクへ割り当てられる。割り当ては論理ブロック番号 (LBN)と物理ブロック番号(PBN)の変換表を用いて実現され、LVM は IO 要求毎に変換表を参照する。

サーバ内の LVM が起動されると、LVM はメタサーバにログインする。次にメタサーバから変換表が送信され、これ以降、LVM は仮想化ボリュームに対してアクセスが可能となる。アプリケーションから仮想化ボリュームへのアクセス要求があると、まず LVM は変換表を用いて PBN を取得し、該当の物理ブロックのアドレスを取得する。その後該当ブロックのデータの読み出しを行い、呼び出し元にデータを返すことで、仮想化ボリュームへのアクセスを完了する。

3.2.2. 統計情報の収集

仮想化ボリュームへのアクセスの際、LVM はアクセスの統計情報を記録する。統計情報は、LBN、PBN 毎の IO スループット、物理ディスクごとの IO スループットと応答時間で構成される。各 LVM によって記録された統計情報は、メタサーバによって収集され、管理 PC により表示することが可能である。

4. 自律的負荷分散機構の設計と実装

実際のアプリケーションにおいて、データへのアクセスには偏りが多いことが一般に知られている。ストレージ仮想化機構で提供される仮想化ボリュームに対しても、データのアクセスが頻繁に行われるホットな領域とアクセスがあまり行われないコールドな領域が存在するものと考えられる。仮想化ボリュームに割り当てられた一部の物理ディスク空間にホットな領域が集中すると、アクセスが集中した物理ディスクがボトルネックになり、仮想化ボリューム全体の性能が劣化する。

空間的な偏りを持つ負荷を物理ディスク間で均一にすることにより仮想化ボリューム全体の性能を改善するため、アクセスの多いホットな物理ディスクからアクセスの少ないコールドな物理ディスクへデータブロックの移送を行う必要がある。また、時間的に変化する負荷に対して自律的にデータの移送を行い、負荷を分散することにより仮想化ボリュームの性能を改善することが可能になる。

図 4 に自律的負荷分散機構の構成を示す。

負荷分散処理は、仮想化ボリューム内でメタサーバ

内の移送プラン作成器と移送器によって自律的に行われる。メタサーバ内の IO マネージャによって定期的に収集された統計情報を元に、移送プラン作成器はデータ移送の必要性を判断し、データ移送プランを作成する。作成されたデータ移送プランは移送器によって実行され、データの移送と変換表の更新が行われる。これらの自律的負荷分散機構により、負荷の空間的な偏りや時間的な変動に追従して動的に物理ディスク間の負荷が均一化され、ストレージの性能を最大限に発揮することが可能となる。

4.1. データ移送プランの作成

定期的に収集される統計情報のうち、物理ディスク間の IO スループットの差の最大値が閾値を超えるとき、移送プラン作成器は移送プラン作成を開始する。閾値は、物理ディスク毎の IO スループットの平均に対する割合とする。データ移送プランは以下の手順で作成される。

1. 最もホットなディスクを i、最もコールドなディスクを j と決定する。
2. ディスク i からディスク j へのデータ移送プランを追加する。
3. 移送分を統計情報に加減算し、再評価する。

1-3 の移送プラン追加は、移送ブロックがなくなるか一定回繰り返した後に終了する。2 のデータ移送プランへの追加は、以下の手順で行われる。

2.1 ディスク i の最もホットなブロックをディスク j の空き領域に移送することを提案する。
2.2 移送分を統計情報に加減算し、ディスク i, j のどちらかが平均値に到達した場合、ディスク i から j への移送プラン追加を終了する。
2.3 提案されたブロックを移送プランに追加する。

2.1-2.3 のデータ移送プラン追加手順は、規定回数繰り返される。

作成されたデータ移送プランは、移送元物理ディスク番号・移送元 PBN、移送先物理ディスク番号・移送先 PBN の集合からなり、移送器に入力される。

4.2. 移送器によるデータ移送

移送器はデータ移送プランにより、以下の手順でデータの移送を行う。

1. データ移送プランより先頭の移送プランを取得する。
2. 移送元 PBN のデータを移送先 PBN に複製する。
3. 変換表の更新要求を各 LVM に送信し、更新完了通知を受信する。

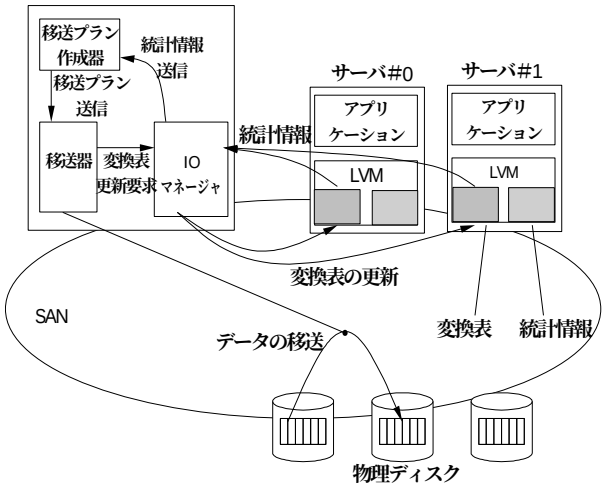


図 4: 自律的負荷分散機構の構成

1-3 のデータ移送は、データ移送プランが終了するまで繰り返される。これにより、データの移送と変換表の更新はブロック毎に逐一行われるため、移送処理の完了を待つことなくブロック移送の効果が順次反映される。

データ移送はアクセスが集中している物理ディスクからのデータ読み出しを伴うので、システム全体の性能の劣化を招く。システムの急激な性能劣化を回避するため、ウェイトの挿入により移送器はデータの移送速度を制限することが可能である。

5. SAN 結合 PC クラスタによる実験

5.1. 基本性能測定

実装した仮想化機構のスケーラビリティを確認するために、3.2 に示される SAN 結合 PC クラスタ上での仮想化機構を用いて基本性能測定を行った。仮想化ボリューム全域において一様アクセスの負荷をかけ、物理ディスク数と仮想化ボリュームの IO スループットと応答時間の関係を測定した。このとき、サーバは 1-4 台、物理ディスクは 1-4 台とし、サーバと物理ディスクは同数とした。仮想化ボリュームには各物理ディスクの先頭から 5GB を順に割当て、ブロックサイズは 8MB とした。負荷のアクセスはサイズ 512byte の一様ランダムアクセスとし、サーバ 1 台あたり 40 個の負荷プロセスを起動した。

実験結果を図 5 に示す。図中、実線は IO スループット、点線は応答時間の測定結果を示している。図より、サーバ・ディスク数に比例して IO スループットが増加していることがわかる。また、そのときの応答時間は、ほぼ変化しないことがわかる。

これらの結果から、実装した仮想化機構はサーバ・ディスク数に対してスケールすることがわかる。

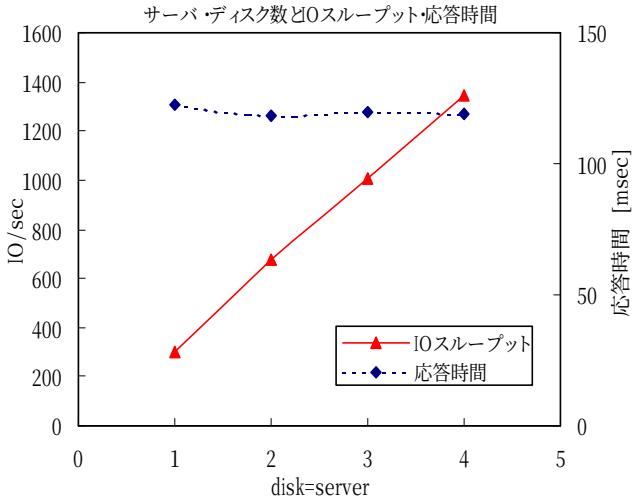


図 5:仮想化機構のスケーラビリティ

5.2. 自律的負荷分散機構の評価

SAN 結合 PC クラスタ上に実装された仮想化機構において、IO 負荷の空間的偏り、時間的変動に対しての自律的負荷分散機構の評価を行う。

自律的負荷分散機構の評価実験において、4 台のサーバと 4 台の物理ディスクを用意し、物理ディスクの先頭の 7GB を実験用の空間として使用した。このうち先頭から 5GB を順に仮想化ボリュームに割り当て、残りの 2GB は空き領域とした。このため、仮想化領域は 20GB となる。ブロックサイズは 8MB とした。負荷分散開始の閾値は 20%とし、制御間隔は 10 分とした。また、負荷のアクセスサイズは 512byte とした。

5.2.1. 負荷の空間的偏りに対する評価

仮想化ボリュームを構成する物理ディスクへのアクセスに偏りがあるときの自律的負荷分散機構の評価を行った。

仮想化ボリュームに対して、表 2 に示すような複数の負荷プロセスを起動した。各負荷プロセスがアクセスするアドレスは、確率分布によって決定される。表中、Uniform は一様分布を示す。Tri(a, b)は以下の式で与えられるような、頂点の位置が a、底辺の幅が b の三角分布を持つ確率分布を表す。

$$prof\ (x) = \frac{2}{a} - \frac{4}{a^2} \bullet [x - b]$$

ただし、$[x - b] \geq \frac{a}{2}$ では $prof\ (x) = 0$ である。

このときの定常状態における LBN 毎の IO スループットを図 6 に示す。

負荷を与えたのち 10 分、30 分経過後にそれぞれデータ移送が行われた。各移送により行われたデータブロックの移動を表 3 に示す。このときの物理ディスク毎の IO スループットと応答時間の時間変化を図 8 に示す。また、データ移送にかかる時間、データ移送時の性能劣化、データ移送後の仮想化ボリューム全体の IO スループットの向上を表 4 に示す。2 回のデータ移送の結果、合計 260 ブロックのデータの移送が行われ、仮想化ボリュームの IO スループットは 70.0%向上した。データ移送を行う前（10 分後）、2 度目のデータ移送後（40 分後）の各物理ディスク IO スループットを図 7 に示す。データ移送により、各物理ディスクに負荷が分散され、全体の性能向上に寄与していることがわかる。

この結果、仮想化ボリューム内の物理ディスクへのブロック割り当てを動的に変更することにより、空間的な負荷の偏りを解消することが、仮想化ボリュームの性能の向上に有益であることが示された。

表 2: 負荷の分布とプロセス数

確率分布	IO/sec	プロセス数
Uniform	10	12
Tri (100, 40)	2.5	3
Tri (200, 40)	2	3
Tri (300, 600)	100	6
Tri (700, 400)	5	3
Tri (700, 100)	2.5	3

表 3: データ移送ブロック

1 回目（10 分後）		2 回目（30 分後）	
Disk#	ブロック数.	Disk#	ブロック数.
0→1	29	0→1	29
0→2	92	0→2	9
0→3	91	0→3	10
合計	212	合計	48
移送時間	958sec	移送時間	245sec

表 4: データ移送時の性能劣化と移送後の性能向上

	1 回目	2 回目	合計
IO スループットの劣化	3.6%	4.7%	-------
IO スループットの向上	53.4%	10.7%	70.0%

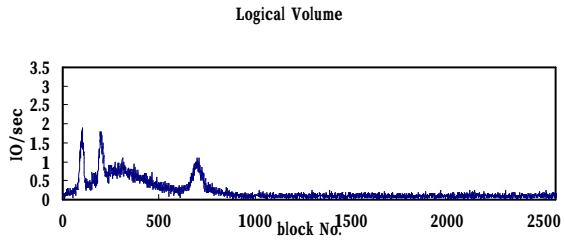
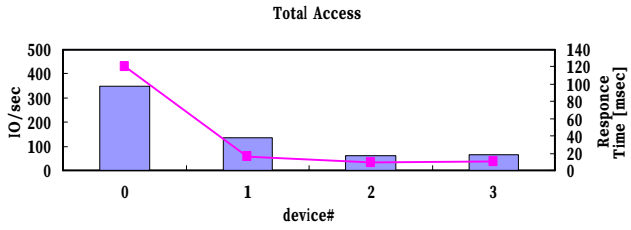
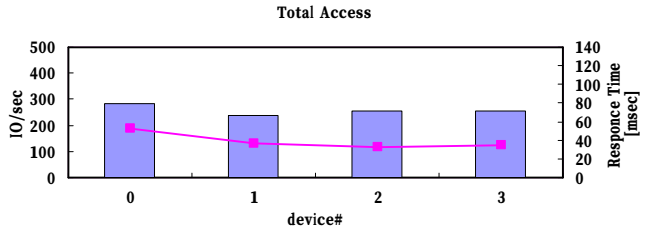


図 6: 仮想化ボリュームの IO スループット

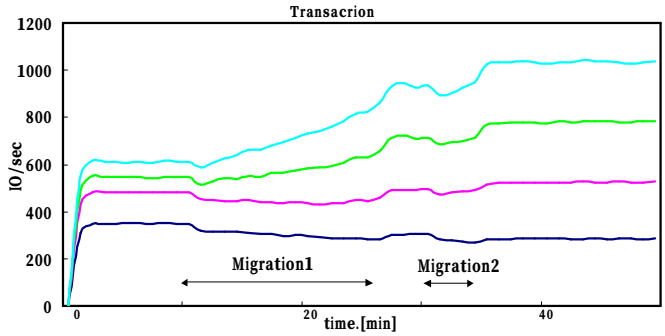


a) データ移送前

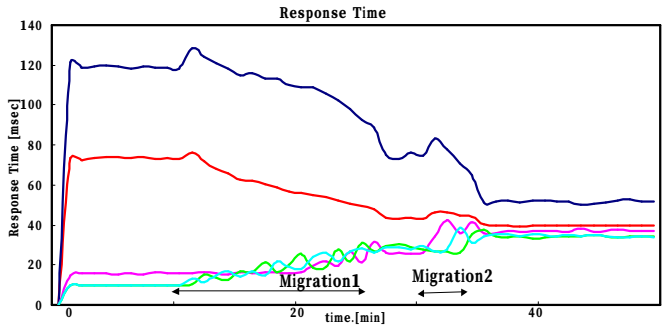


b) データ移送後

図 7: 物理ディスクの IO スループット



a) IO スループット



b) 応答時間

図 8: 物理ディスクの IO スループットと応答時間の時間的変化

上記の実験は仮想化ボリュームに対して充分な IO 負荷を与えているが、実際のシステムでは全体の性能に対して余裕のある IO 負荷での運用が多い。仮想化ボリューム全体を構成する物理ディスクの一部に処理能力を超える IO 負荷が集中している場合、自律的負荷分散機構にて負荷を分散することによる応答時間改善の評価実験を行った。

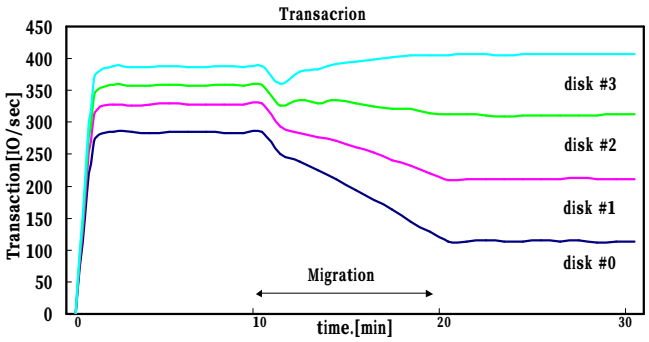
各サーバに起動した負荷プロセスを表 5 に示す。負荷を与えた 10 分後にデータの移送が行われた。移送ブロック数、データ移送時の性能劣化とデータ移送後の性能向上を表 6 に示す。このときの物理ディスク毎の IO スループットと応答時間の時間変化を図 9 に示す。

データ移送により、各物理ディスクに負荷が分散され、応答時間の短縮に寄与していることがわかる。

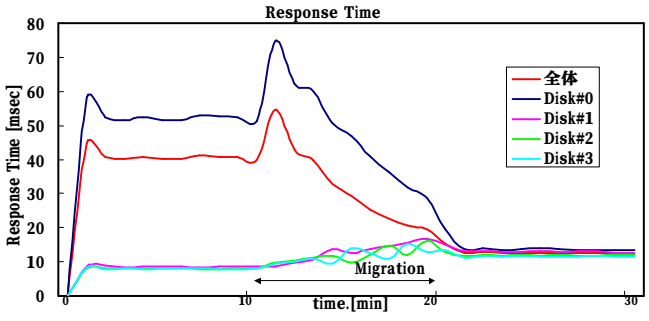
この結果、仮想化ボリューム内の物理ディスクへのブロック割り当てを動的に変更することにより負荷不均衡を解消することが、仮想化ボリュームの性能の向上に有益であることが示された。

表 5: 負荷の分布とプロセス数

確率分布	IO/sec	プロセス数
Uniform	10	12
Tri (100, 40)	2	3
Tri (200, 40)	1.25	3
Tri (300, 600)	10	6
Tri (700, 400)	1	3
Tri (700, 100)	0.5	3



a) IO スループット



b) 応答時間

図 9: 物理ディスクの IO スループットと応答時間の時間的変化

次に移送器のウェイト挿入による性能劣化と移送時間の関係を測定した。

表 5 の負荷においてデータ移送の際の仮想化ボリュームの応答時間増加の最大値及び移送時間を、移送器のウェイトを 0 から 3000msec まで変化させて計測した。ウェイトは移送器によりブロックが移送されるごとに挿入される。

測定の結果を図 10 に示す。図中、点線が移送時間、実線が応答時間の性能劣化である。図より、移送器に挿入するウェイトを増加させると性能劣化はより小さく抑えることができ、移送にかかる時間が増加することが確認できる。

表 6: データ移送ブロック

1 回目（10 分後）	
Disk#	ブロック数.
0→1	76
0→2	86
0→3	86
合計	248
移送時間	568sec
応答時間の増加	32.6%
応答時間の短縮	70.5%

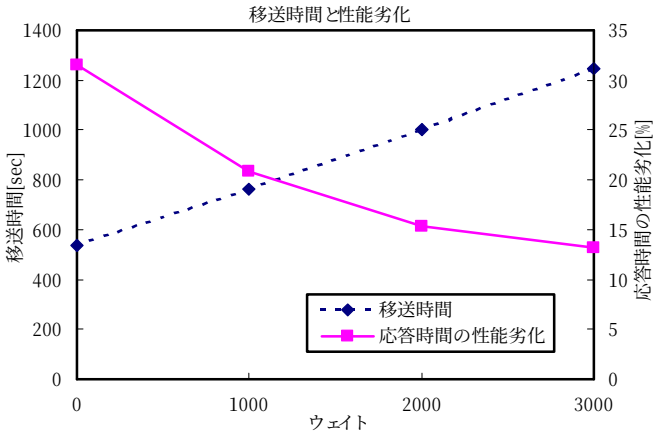


図 10: ウェイト挿入時の移送時間と性能劣化

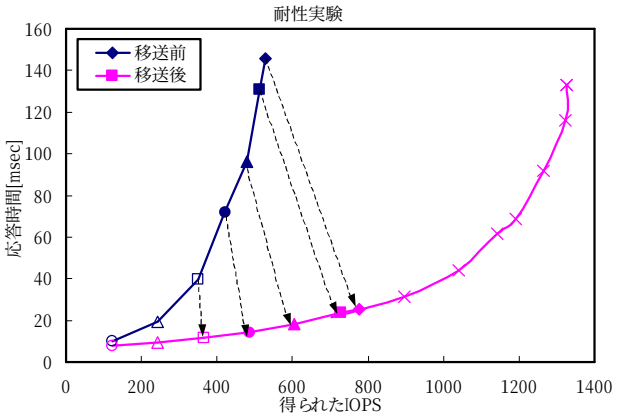


図 11: IO スループットと応答時間

また、表 5 の負荷プロセスと同じ割合で負荷プロセスを各サーバに 10-70 個起動したときの移送前後の IO スループット、応答時間の関係を測定した。

測定結果を図 11 に示す。

図中、グラフの数字はサーバごとのプロセス数、理想曲線は仮想化ボリューム全体に対してランダムアクセスを行ったときの値である。

図より、空間的に偏っている負荷を均衡化することにより応答時間を短縮できることがわかる。また、システムに対して負荷が過大になると、IO スループットも改善することがわかる。

5.2.2. 負荷の時間的偏りに対する評価

銀行の業務システムのように、通常業務時間内でのデータへのアクセスと、業務終了時間後のバックアップなど、ストレージへのデータアクセスの偏りは時間的に変化することが予想される。このように、物理ディスクへのアクセスの偏りに時間的な変化があるとき、自律的負荷分散機構により物理ディスク間の負荷の均衡化を自律的に保つことにより仮想化ボリュームの性能を最大化することを実験により評価した。

仮想化ボリュームに対して、表 2 に示すような複数の負荷プロセスを起動し、50 分後に表 7 に示される負荷プロセスに変化させた。

データの移送は表 8 に示すように負荷変化前に 2 回、変化後に 2 回行われた。物理ディスク毎の IO スループットと応答時間の時間変化を図 12 に示す。負荷変動前の 2 回のデータ移送で IO スループットと応答時間が改善していることがわかる。しかし、50 分後の IO 負荷の変化で、物理ディスク間での負荷の偏りが生じるため、IO 負荷、応答時間共に悪化している。その後、60 分後と 80 分後に 2 度のデータ移送が行われ、再度 IO スループットと応答時間が改善している。

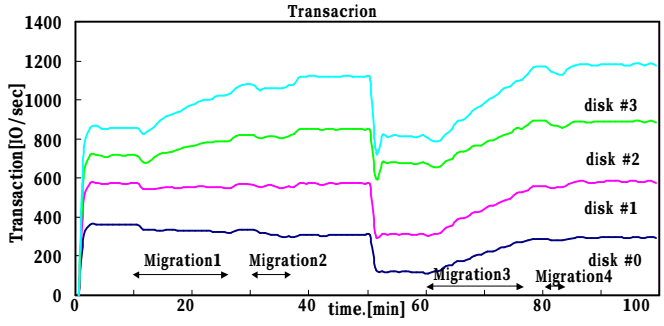
この結果、仮想化ボリュームへの負荷変動に対して、自律的に負荷を分散させることにより、仮想化ボリュームの性能を改善できることが示された。

表 7: 変化後の負荷の分布とプロセス数

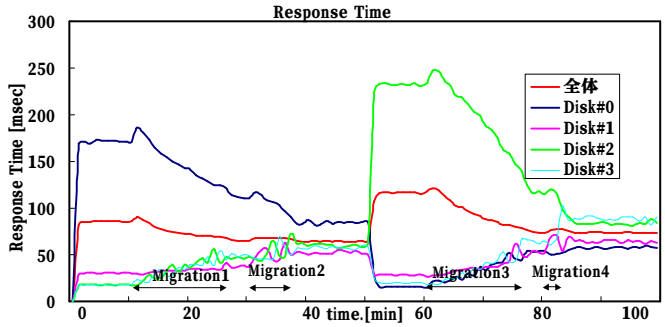
確率分布	IO/sec	プロセス数
Uniform	10	12
Tri (100, 40)	2	3
Tri (200, 40)	1.25	3
Tri (300, 600)	10	6
Tri (700, 400)	1	3
Tri (700, 100)	0.5	3

表 8: データ移送時間とブロック数

移送	開始時間	移送時間	移送ブロック数
1 回目	10min	975sec	150
2 回目	30min	402sec	52
3 回目	60min	986sec	141
4 回目	80min	208sec	25



a) IO スループット



b) 応答時間

図 12: 物理ディスクの IO スループットと応答時間の時間的変化

6. 結論

本論文では、SAN 環境におけるストレージ仮想化機構を設計し、SAN 結合 PC クラスタ上での実装を行った。また、小さいサイズのアクセス要求が支配的なシステムを対象とした自律的負荷分散機構の設計・実装を行った。評価実験としてストレージ仮想化機構の基本性能を測定し、サーバ・ディスク数に対して性能がスケールすることを確認した。さらに自律的負荷分散機構として、ブロックのデータの移送により物理ディスク間の IO 負荷の空間的な偏りを解消することにより、仮想化ボリュームの性能向上を確認した。また、時間的な IO 負荷の変動に対して自律的に性能を改善することを確認した。

今後は、LVM の書き込みに関して、一貫性保持のためメタサーバのロック機能を実装し、実験により評価する予定である。

文　献

[1] Charles T. Clark. The Virtualization of Storage. White Papers, TidalWire http://www.tidalwire.com/, 2001.

[2] 富士通, SafeFILE/Global - 次世代のファイルシステム -, Technical White Paper

[3] SAN Symphony. DataCore Software, http://www.datacore.com/

[4] VERITAS. NetBackup. http://www.veritas.com/. Technical report

[5] Hitachi. ShadowImage. http://www.hds.com/pddf/shadowimageR6.pdf. 2001.

[6] EMC Corporation. EMC TimeFinder Product Description Guide. 1998.

[7] K. Goda, T. TAMURA, M. Ouchi, and M. Kitsuregawa, "Run-time Load Balancing System on SAN-connected PC Cluster for Dynamic Injection of CPU and Disk Resource --- A Case Study of Data Mining Application ---," DEXA2002, France, September 2002