2004－DBS－134　(II)　(62)
2004／7／14

自律分散型ストレージシステムの開発

江尻　革†　　太田　光彦†　　横畑　徹†

あらまし　ストレージシステムの管理コスト低減と信頼性向上を目的とした自律分散型ストレージシステムを開発した。本システムは、自律ディスクモジュールと自律分散型アーキテクチャからなる。自律ディスクモジュールは、HDD 内情報を用いて故障予測などを行い、専用チップ化によりコストダウンを図る。自律分散アーキテクチャは、ディスクとデータ・ネットワークに可変値の「優先度指標」を付与し、この指標に基づいて、データの自律移動・バックアップなどの管理機能をサーバレスで実現する。今回は、アロケート先探索、ミラー作成によるデータ多重化、ハートビートによる相互監視とミラー再生について、アーキテクチャおよび実験結果について報告する。

Development of distributed autonomous storage system

Arata EJIRI†　　Mitsuhiko OHTA†　and　Toru YOKOHATA†

Abstract　We developed the distributed autonomous storage system that aimed at the management cost decrease and the reliability improving of the storage system. This system consists of the distributed autonomous architecture with the autonomous disk module. The autonomous disk module does the fault prediction etc. by using the HDD internal report. This architecture gives "Priority index" to the disk, data, and the network, and achieves the controlling functions of the autonomous transportation and the backup, etc. without concentrated server. This time, we report on architecture and the experimental result about the allocation search, the mirror making, mutual watches by heart beat, and the mirror reproductions.

1. はじめに

ストレージシステム市場は、従来のサーバに１対１に接続するタイプ（DAS）から、複数のストレージシステムをネットワークに接続して集中管理する方向（SAN）に急速に移行している。しかし、管理すべきストレージノード数の増加につれて、集中管理サーバが必要とするリソース（メモリ容量やＣＰＵ性能）は増加し、人的コストも含めた管理コストの爆発的な増加が見込まれている。

また、コンシューマ市場においても、TV パソコンや HDD ビデオレコーダ等の普及により増加しつつある家庭内デジタルコンテンツの管理が問題になりつつある。そもそも家庭では管理者自体が不在であり、自律的な管理に対するニーズは高いと考えられる。

一方、HDD 単体に目を向けると、年率 100%の記録密度向上により、メディア、ヘッド、HDI、サーボ、信号処理系すべての部分で原理的な限界に近づいており、HDD 単体での信頼性は下がらざるを得ない。したがって、今後はシステム全体として信頼性を確保すべきであることは論を待たない。

このような背景の元、自律ディスク[1][3]やネットワークストレージ[2][4]の研究が盛んに行われている。

†㈱富士通研究所　神奈川県
†Fujitsu Laboratories Ltd.　10-1 Morinosato wakamiya, Atsugi-shi, Kanagawa, 243-0197 Japan

今回、我々は「自律分散」と「信頼性」をキーワードに、従来の集中管理の対極的な概念としての IP ベースの「自律分散型」ストレージシステムの開発を開始した。

図１に本システムの概要を示す。本システムは、構成要素である自律ディスクモジュールとモジュール相互をつなぐ自律分散型アーキテクチャからなる。自律ディスクモジュールは、HDD 内情報を用いて故障予測などインテリジェントな機能を実現するとともに、CPU や TOE(TCP Offload Engine)を含めた専用チップ化と機能の絞込みによりコストダウンを図る。また、自律分散アーキテクチャは、ディスクとデータ・ネットワークに可変値の「優先度指標」を付与し、この指標に基づいて、データの自律移動（＝自律待避）・バックアップなどの管理機能をサーバレスで実現する。これらによって、システム全体での信頼性の確保と、管理コストの低減を図る。

本報では、自律分散型アーキテクチャおよび、Linux 上に構築した実験システムにおける実験結果について報告する。

2. 自律分散型アーキテクチャ

図２に本アーキテクチャのデータ構造を、図３に機能モジュール構成を示す。前述した様に、本ストレージシステムでは集中サーバは存在せず、クライアント

ノード上のイニシエータモジュール、ターゲットモジュールから構成される。

2.1. データ構造

（1） LVOL と LUN

ユーザアプリケーションから見ると、本システムは仮想的な論理ボリューム（LVOL）に見える。ユーザは LVOL 内のセグメント先頭アドレス(LVA）とセグメントサイズにより仮想 LVOL にアクセスする。それぞれのセグメントは実体であるターゲット内論理ユニット（LUN）のユーザデータ領域に対応している。

（2） データ ID

データ ID は、イニシエータ IP アドレス＋LVA から構成され、ユーザデータの保存と同時にターゲットのメタ情報領域に格納され、データ探索等に利用される。

2.2. イニシエータ

イニシエータは以下の機能を持つ。

（1） メタ情報探索機能

イニシエータはユーザから来た LVA を LUN 情報（ターゲット IP アドレス、LUN 番号）に変換するメタ情報をメモリ内に持っている。必要なメタ情報がメモリ内に存在しない場合に、ネットワーク内のターゲットと通信を行い、メタ情報を収集する。

（2） アロケート機能

アロケート機能は、ユーザから新規の書き込みを依頼された場合に、ネットワーク内の他のターゲットのメタ情報管理機能と通信を行い、アロケート可能なターゲットを探索し、領域を確保する。

（3） データアクセス機能

データアクセス機能は、ターゲットの iSCSI Target 機能との間でユーザデータの転送を行い、ターゲットが内蔵するハードディスクへの read/write を行う。

2.3. ターゲット

ターゲットは、内部に 1 台以上のハードディスクを持つノードである。内蔵のハードディスクは複数の論理ユニット(LUN)を持つ。 LUN にはメタ情報と、ユーザデータが格納されている。

（1） メタ情報管理機能

メタ情報管理機能は、イニシエータからのメタ情報に対する要求に対し、適切な処理を行い、応答を返す。

（2） データアクセス機能

iSCSI Target 機能は、イニシエータからのユーザデータの read/write 要求を、内蔵のハードディスクに対し実行する。

（3） データ保障機能

データ保障機能は、ユーザデータの多重化とターゲット相互監視による故障検出、多重度の回復を行う。

（4） イニシエータ機能

イニシエータ機能は、ユーザデータの多重化を行う際に、イニシエータの役割を果たしてミラー先の確保、とユーザデータの転送を行う。

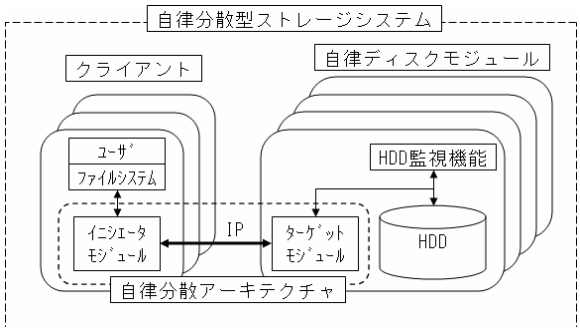

図 1　自律分散型ストレージシステム

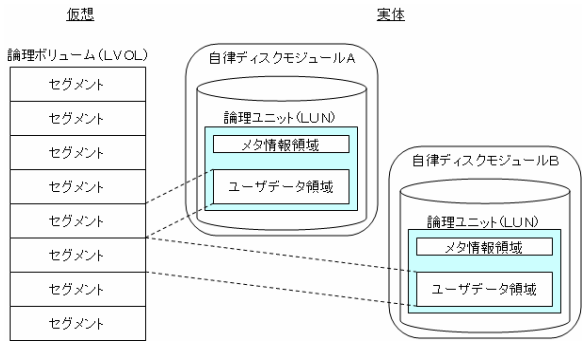

図 2　データ構造

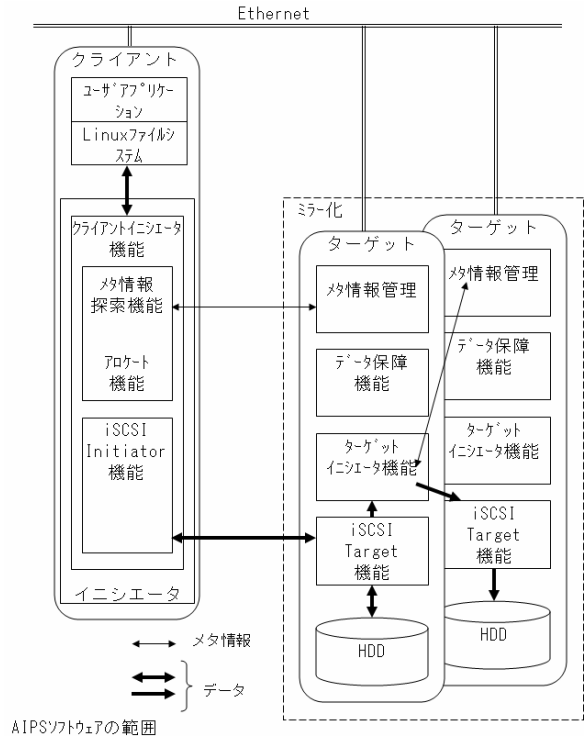

図 3　機能モジュール構成

3. 自律分散型処理の動作例

以下に、前述したイニシエータとターゲットを用いてストレージシステムの各動作を、自律分散的に行う方式について説明する。

3.1. データアクセス

イニシエータは、ユーザデータの read/write 要求を受け取り、以下の要領でターゲットにアクセスする。図 4 を参照のこと。

（1） まず、キャッシュ上のメタデータをチェックし、メタデータが残っていた場合、それにしたがって、該当ターゲットにデータ ID を確認する。
（2） ID が合致すれば、直ちに iSCSI Initiator によりターゲットに R/W 要求を送る。
（3） ID が合致しない場合、全ターゲットにデータ探索要求を出す。通信の高速性、同時発送性を重視し、データ探索は UDP Broadcast を使用する。該当するデータを持つターゲットは、応答を返す。
（4） ターゲットから応答があった場合、iSCSI Initiator によりターゲットに R/W 要求を送る。
（5） ターゲットから応答がない場合、要求が Write であれば、アロケートを行う。

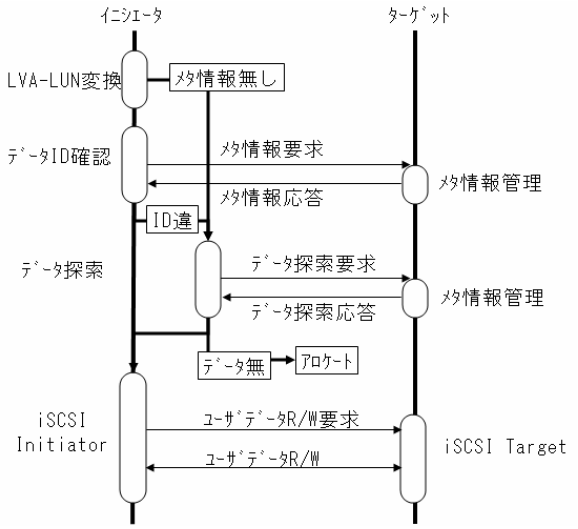

図 4　データアクセス

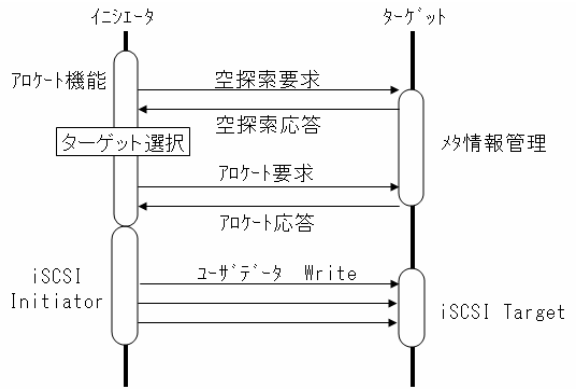

図 5　アロケート

（6） （1）の時点でメタ情報が無い場合、（3）と同様にデータ探索を行う。

3.2. アロケート

新規にデータを保存する場合や、Write 時のデータ探索においてデータが見つからない場合、以下の要領でアロケートを行う。図 5 参照。

（1） イニシエータは、ネットワーク内の全ターゲットに対し、必要とするサイズ、LUN 属性を提示して、アロケート可能なターゲットを要求する。イニシエータの要求を満足することが可能なターゲットは、自己の IP アドレスと優先度指標を添えて応答する。空探索も UDP Broadcast を使用する。
（2） イニシエータは、応答してきたターゲットの中から優先度指標を基準にして適切なターゲットを選択、選択したターゲットに対し、データ ID を通知して LUN のアロケートを要求する。
（3） アロケート成功後、iSCSI Initiator によりターゲットにデータを転送、iSCSI Target によりディスクにデータを書き込む。

3.3. ミラー作成

ターゲットは、データ保全の一環としてミラーを作成し、データの多重化を行う。図 6 参照。

ターゲット上のイニシエータ機能を用いて、イニシエータと同様にミラー用の LUN をアロケートし、iSCSI Initiator 機能を用いてミラー用ターゲットにデータを書き込む。このときミラー元とミラー先のデータ ID は等しく、プライマリ・セカンダリといった区別は無い。また、イニシエータとしてはミラー元にデータが書き込まれた時点で保存完了と判断する。

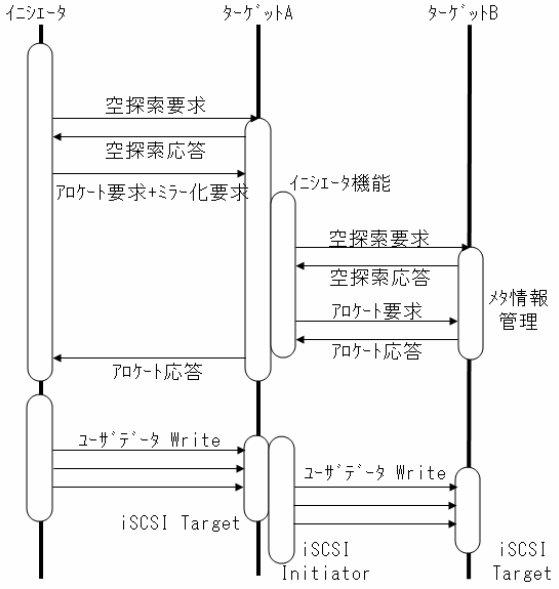

図 6　データ多重化（ミラー作成）

3.4. 相互監視

ターゲットは、データ保全の一環として以下の様に、ターゲット相互の監視を行う。図 7 参照。

各ターゲットはネットワーク内の全ターゲットに UDP Broadcast を用いてハートビート(HB)を定期的に送信する。HB には送信元の IP アドレスと送信元ターゲットが持つデータ ID の一覧が添付されている。

各ターゲットでは HB 監視スレッドが常に動いており、自分が持っているデータ ID と等しいデータ ID を持つ HB を監視し、一定時間 HB が来ない場合、該当ターゲットを故障とみなす。つまり自分のミラーを持つターゲットのみ監視し、無関係のターゲットは監視しない。

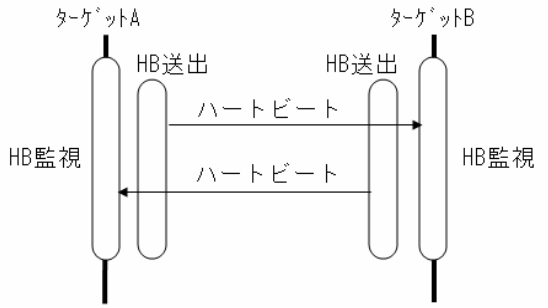

図 7　相互監視

3.5. ミラー再生

ターゲットは、以下の場合にミラーの再生を行う。

- 相互監視により自分のミラーを持つターゲットの故障を検出した場合。
- 自己監視により自分の内蔵ハードディスクの不調を検出した場合。

ミラー再生は、ミラー作成と同様に、ターゲット上のイニシエータ機能を用いてミラー用の LUN をアロケートし、iSCSI Initiator 機能を用いてミラー用ターゲットにデータを書き込む。

4. 自律分散環境でのアロケート分散

ネットワークに接続したディスク群をまとめてストレージとして利用する場合の留意点は以下である。

①故障などを考えて、データがひとつのターゲットに集中しないようにする（使用量均等）
②アクセススピードを考え、なるべく読み込み・書き込みアクセスは分散する（アクセス分散）

すべてのターゲットディスクを管理する集中サーバがある場合、容量やアクセスを均一化するスケジューリングは比較的容易である。しかし、サーバをもたない自律分散型の本システムでは、クライアントやターゲットディスク相互間のやり取りと優先度指標という数値よって均一化を実現しなければならない。このときの問題点と解決策について考える。

図 8-12 はすべて 3 台のターゲットディスクを使用して、一台または 3 台のクライアントから 1GB 単位のデータを連続的にアロケートした場合のシミュレーション結果である。

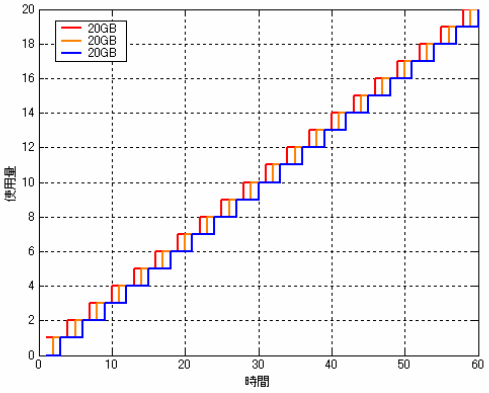

図 8　アロケート分散(Simulation)

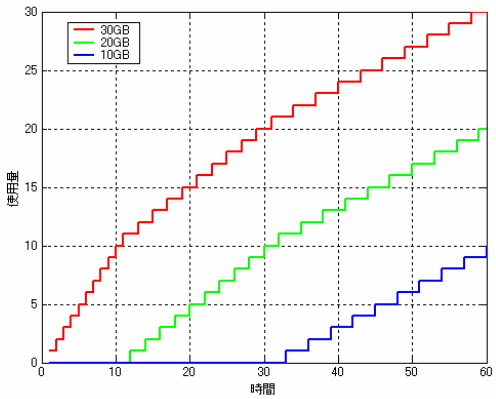

図 9　ディスク容量が異なる場合

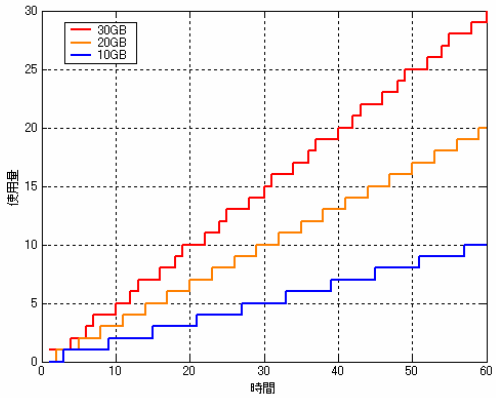

図 10 優先度＝未使用率とした場合

4.1. ディスク容量の不均一

図 8 は３台のターゲットディスクの容量がすべて 20GB とし、優先度＝空き容量、１ユーザのみのアクセス、とした場合の結果である。データの均等配置、アクセスの均等化が同時に達成されている。

図 9 は 3 台のターゲットの容量が 10GB,20GB,30GB の３種類ある場合である。優先度＝空き容量、とした場合、初期(時間 0-10 の間)に 30GB のターゲットにアクセス、データ配置が集中してしまう。

ここで、優先度＝未使用率とすると、図 10 のようにアクセスおよびデータ配置を分散することができる。

4.2. ディスクの途中追加

次に、途中から空のターゲットがネットワークに追加された場合を考える。

図 11 は３台のターゲットの容量がすべて 20GB、優先度＝未使用率として、時間＝20 に３台目のターゲットを追加した場合を示す。この場合、時間 20-30 の間は３台目のターゲットにのみアロケートが集中する．

図 12 は未使用率の大きいターゲットにアロケートが集中しないように、イニシエータがターゲットを選択する確率を(1)式にした場合の結果である（10 回の試行の平均値)。追加ターゲットへの集中がかなり改善されているのがわかる。

ターゲット i を選択する確率 Pi：

$$Pi = Ri / \Sigma(Ri) \quad \cdots(1)$$

（Ri：ターゲット i の優先度指標）

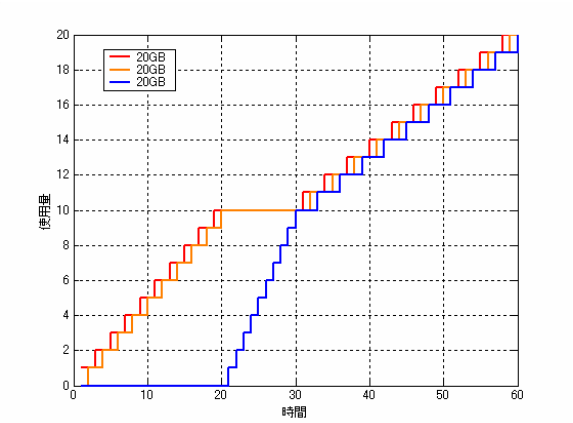

図 11　ディスクの途中追加

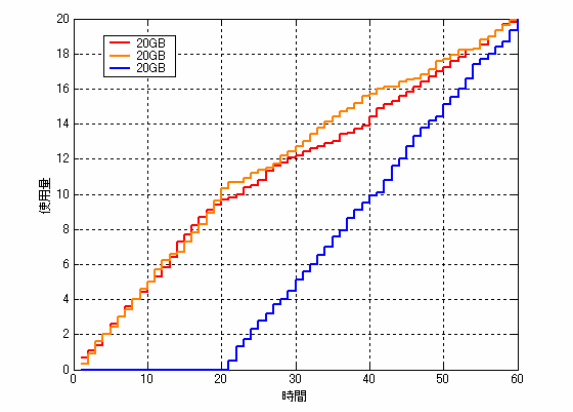

図 12 優先度に応じた確率的選択

5. 実験結果

前章までのアルゴリズムを Linux 上に実装し、実験を行った結果を以下に示す。

5.1. 機器構成

（１） 使用マシン

クライアント：PentiumIV 3GHz

ターゲット　：(CPU) Mobile-PentiumIII 933MHz
(HDD) 40GB(E-IDE/5,400rpm)

（富士通製ブレードサーバ PRIMERGY BX300）

（２） ネットワーク：1Gbps

5.2. アロケート

図 13 にクライアント１台、ターゲット 10 台の場合に、イニシエータから 50LUN のアロケートを行った実験結果を示す。（凡例は IP アドレスを示す）

ミラーも含めて１ターゲットあたり 10LUN×10 台＝合計 100LUN がランダムな順番でアロケートされているのが分かる。

今回はアロケート確認実験であるため、アロケート後 16MB のみ DataWrite しているが、実際に 30 台規模の完全 Read/Write 試験も実施し、データ化け・ネットワーク輻輳等問題ないことを確認している。

5.3. 故障検出＆ミラー再生

図 14 に 100LUN アロケート後に１台のターゲットを故障（HB 発信を停止）させた場合のミラー再生の実験結果を示す。（横軸 330 秒の時点で 179 を停止）

今回は、HB 間隔５秒で４回 HB が来ない場合に故障とみなした。HB 停止後 20 秒後にミラー再生が始まり、故障ターゲットが持っていた 10 個の LUN が、残り９台のターゲットに再生される様子が分かる。

5.4. ディスクの途中追加

図 15 に、途中で空のターゲットを追加した場合の実験結果を示す。（横軸 155 秒の時点で 184 を追加）

これは、前章で述べた確率的選択方式の結果であり、追加ターゲットへの集中が少ないことが分かる。

6. まとめ

本報の内容を以下にまとめる。

（１） 集中サーバを用いない自律分散型ストレージシステムのアーキテクチャを提案した。

（２） 提案したアーキテクチャに基づき、10 台のターゲットディスク上にてアロケート＆Write、相互監視＆ミラー再生実験を行い、本方式の有効性を

実証した。

（３） アロケート分散の手法について検討し、確率的選択方式により、途中追加したターゲットへのアロケート集中防止を確認した。

今後は、以下の開発を行う予定である。

（１） データ自律移動による負荷分散、性能改善。

（２） ターゲットの自己診断機能

（３） ハードウエア化による自律ディスクモジュールの製作

文　献

[1] H.Yokota. Autonomous Disks for Advanced Database Applications. In Proc.of International Symposium on Database Applications in Non-Traditional Environments (DANTE'99). pp.441-448, Nov.1999.

[2] 合田和生, 田村孝之, 小口正人, 喜連川優, “共有ストレージプールを用いた並列データベース処理に於けるオンデマンド資源調節, ”信学技報, DE2002-85,DC2002-12,pp.1-6,(2002)

[3] 上村哲也、熊谷幸夫、Active Network Storage のネットワーク性能の評価、電子情報通信学会論文誌、D-I Vol.J85-D-I No.9 pp.841-849,(2002)

[4] 武理一郎, ペタバイトストレージの開発とその応用について, 大規模データマネージメント関連会議合同論文集、p.13-20,(2003)

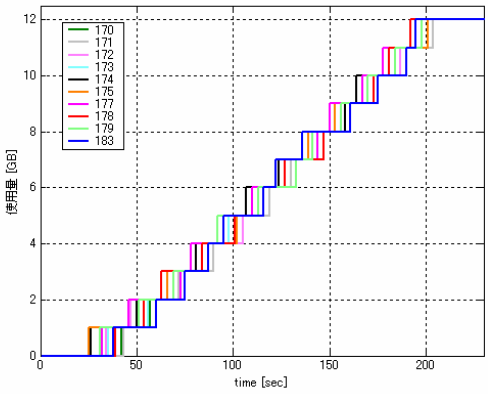

図 13　アロケート分散（実験）

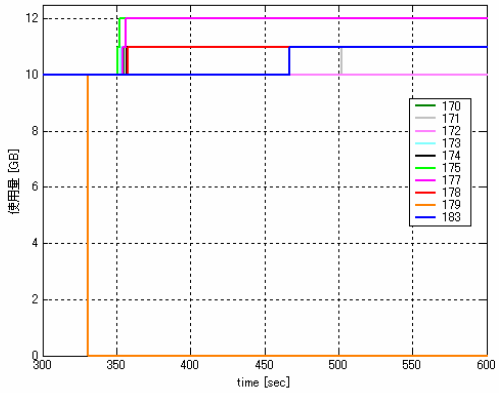

図 14　故障検出＆ミラー再生

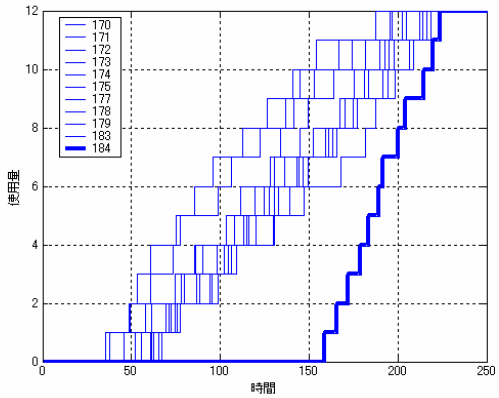

図 15　ディスク途中追加