内部統制のためのバックアップシステム

上原 稔[†], 犬竹 義洋[†], 友野 敬大[†]

内部統制において事業継続性は重要な課題である。それを保障するにはディザスタリカバリが必要である。ディザスタリカバリには遠隔バックアップが適する。しかし、遠隔バックアップには、通信のボトルネック、セキュリティ、コストなど多くの問題がある。さらに、内部統制のために任意の時間のイメージを復元できなければならない。本論文では、これらの課題を VLSD (Virtual Large Scale Disks)ツールキットを用いて解決する。本システムは、VLSD のクラスを組み合わせて構成されている。本システムでは、PrimaryBackupDisk によって効率よく差分バックアップを行い、CipherDisk によって通信データを暗号化し、RemoteDisk によって転送する。遠隔側では、UnionDisk によって差分を結合し、任意の時間イメージへアクセス可能とする。

A Backup System for Internal Control

Minoru Uehara[†], Yoshihiro Inutake[†], Akihiro Tomono[†]

Business continuation is very important for internal control. Disaster recovery plan is required to guarantee business continuation. Remote backup is suited for disaster recovery. However, conventional remote backup system has several issues such as communication bottleneck, security, cost and so on. In addition, it must recover the image snapshot taken at any time. In this paper, we propose a backup system which can solve these issues using VLSD (Virtual Large Scale Disks) toolkit. VLSD is a toolkit for constructing large scale disks by gathering free resources. Our system has been developed with VLSD classes. For example, PrimaryBackupDisk generates differential backup image efficiently. CipherDisk encrypts communication data. RemoteDisk communicate with remote disk server. In a remote side, UnionDisk concatenates differential disk images and then provides accessing to the image of any time.

1. はじめに

近年、内部統制の重要さが高まってきている。内部統制とは、業務の有効性と効率性、財務報告の信頼性、関連法規の遵守という目的の達成に関して合理的な保証を提供することを意図した、事業体に属する人々によって遂行される一つのプロセスとして定義される。内部統制ではセキュリティに焦点が当てられた研究が活発であるが、本来の定義からいえばセキュリティは副次的な問題に過ぎない。むしろ、どのように業務を継続するか、あるいは改善するかという方が重要である。本論文では主として事業継続性の観点からバックアップについて議論する。

今日では、企業の業務は情報システムによって支えられている。情報システムは、事業に欠かせないインフラとなっている。そのため、情報システム自身の継続性が事業の継続性と直結している。情報システムのインフラは IaaS などのクラウドサービスを用いることで短期間に回復可能となってきた。しかし、事業に必要なデータが失われると、インフラがあっても事業を継続することはできない。よって、情報システムのバックアップシステムの整備が不可欠である。業務中断が発生しないために復旧計画を策定し、そのための体制を整える必要がある。

事業が中断すると、企業の目標達成に大きな影響を及ぼす。例えば、災害などでデータが消失すれば、事業活動が行えなくなり、利益目標達成できず、顧客に大きな迷惑をかけることになる。企業が災害などで被害を受けても、企業として存続できるよう備えることは、当然必要であり、例え業務が中断しても可能な限り短期間で再開することが望まれる。そのためには、システムのバックデータを遠隔地に保存することが重要になる。一か所に集中したクラウドを用いることもできないためクラウド間連携も重要である。

しかし、遠隔地にバックアップを保存するには、相応のコストがかかる。例えば、バックアップデータのバックアップ媒体コピー作業やバックアップ媒体のなど人件費、保存場所のコストなどのコストがある。

既存のストレージアプライアンス製品は極めて高価であり、低コストかつ大容量のストレージを望む市場に応えることができない。実際、組織内で使用される PC には多くの空容量があり、それらを集約すれば大容量の大規模ストレージを低コストで実現することができるかもしれない。

我々はディスクレベル分散型ストレージを構築するためのツールキット VLSD (Virtual Large Scale Disks)を開発している 2)3)4)5)6)7)。VLSD は 100% pure Java で記述され、プラットフォームに依存しない。多様なクラスを組み合わせることで、プラットフォームの限界を越える大規模ストレージを仮想的に実現することができる。また、遊休資源を集約することで、低コストで大容量のストレージを構築することができる。我々は、VLSD を用いて 500 台の PC からなる 60TiB ストレージを試作した。

本研究では、VLSD(Virtual Large-Scale Disk)を用いた遠隔ミラーリングによるバックアップの方法を提案する。本研究によって、内部統制で求められているバックアップのためのコスト低減を実現できる。

本文の構成は以下の通りである。2 節で関連研究について述べる。3 節では VLSD について述べる。4 節では、バックアップシステムの設計と実装について述べる。5 節では、その評価を行う。最後に結論を述べる。

[†]東洋大学 総合情報学部
Faculty of Information Sciences and Arts, Toyo University.

2. 関連研究

近年，米国企業に留まらず、日本企業でも不祥事が多発し、内部統制の必要性が急速に高まっている。内部統制とは、COSO(Committee of Sponsoring Organizations of the Treadway Commission)によれば、『(1) 業務の有効性・効率性、(2) 財務報告の信頼性、(3) 関連する法規の遵守の 3 つの目的の達成に関して、合理的な保証を提供することを意図した、会社の取締役会、経営者およびその他の従業員によって遂行されるプロセスであり、相互に関連する要素、すなわち、統制環境、リスクの評価、統制活動，情報と伝達，モニタリングから構成される』と定義されている。本論では、これを実現するための一連の仕組みを内部統制システムとする。

内部統制の目的を達成するためにはログによる監視が必要である。内部統制のためには通常の運用より詳細なログを必要とするため、大量のログを安全に管理するログ管理システムが必要になる。このシステムは半永久的に保管されたログから内部統制に必要な情報を抽出する。セキュリティより、むしろ経営の改善を目的として利用される。

しかし、このようなシステムを用いても企業経営におけるリスクが完全になくなるわけではない。内部統制では、事業継続性が求められる。そして、事業継続のためにはリスク管理が必要となる。特に、自然災害に対するリスク管理は重要である。自然災害は予測が困難であるが、無視もできない。災害を管理するには、災害があっても事業を継続できるような災害復旧計画（ディザスタリカバリ）がなければならない。ここで、ディザスタリカバリとは、災害などで被害を受けた際のシステム復旧・修復をすることを指す。また、そのための備えとなる機器やシステム、災害などの状況下でもシステム停止を免れるための緊急時対応などが含まれる。

一般的なディザスタリカバリの戦略は遠隔バックアップである。事業に必要なデータを失うと事業を継続できない。そのような事態を防ぐためにバックアップを行う。しかし、バックアップしたデータを近くに置くと、大規模災害で全滅する可能性がある。そこで、遠隔地にバックアップデータを保管し、大規模な災害が生じてもデータの損失を回避する必要がある。

一般的にバックアップにもいくつかの方式がある。階層の順に、データベースレベル、ファイルレベル、ディスクレベルなどである。データベースレベルのバックアップでは､すべてのデータがデータベースに保管されていることが前提となる｡しかし、実際には EUC(End User Computing)により非定形文書、すなわち Word や Excel などの Office 文書が広く使用されている。ファイルレベルのバックアップは一般的でかつ効率がよい。しかし、バックアップに長い時間を要する。ファイルレベルのバックアップとして近年注目されているのは Apple 社の Time Machine である。任意の時間に戻ることができるが、最初のバックアップ時間は非常に長い。ディスクレベルのバックアップでは dump などがあるが､RAID1 もバックアップとして利用できる。ただし､RAID1 では常に最新版のコピーを行うため Time Machine のような世代管理ができない。内部統制のためには、RAID1 のように（通常業務に影響しないほど）高速で、かつ世代管理ができるバックアップシステムが必要である。

バックアップのメディアについては、テープがよく利用されている。しかし、HDD のコストが下がっていることと、内部統制では常に過去のデータをアクセスできなければならないため、テープの代わりに安価な HDD を利用することも考慮しなければならない。もっとも合理的な方法は、余剰資源を用いてバックアップを保管することである。我々はディスクレベル分散型ストレージを構築するためのツールキット VLSD (Virtual Large Scale Disks)を開発している。また、これを用いて余剰資源から 60TiB のストレージを構築した。同様の手法を用いれば、バックアップのメディアはほとんど必要ない。

基本的にバックアップはコピーに等しいため、非常に大きな容量を必要とする。そこで、容量を減らす工夫が重要となる。差分バックアップは有効である。また、ストレージ内には同内容のデータが複数保存されていることが少なくない。そうした重複データの保存を回避して、ストレージ利用の効率化を図る技術が重複排除(deduplication)である。重複排除には差分バックアップと同様もしくはそれ以上の効果がある。

重複排除を実現する方式には、アプリケーションレベル、ファイルシステムレベルとブロックレベルがある。アプリケーションレベルでは、同じ内容のファイルを検索する。しかし、発見した重複ファイルをそのまま残すか、シンボリックリンクに変換するかなどの対応は用途によるため、一概に決められない。ファイルシステムレベルでは、いかなる用途に対してもファイルシステムが一貫したビューを提供するため、このような混乱はない。しかし、広く普及しているファイルシステムを改変することは容易でない。ブロックレベルでは、ファイルシステムに依存しない重複排除が可能である。例えば､ZFS はブロックレベルの重複排除をサポートする予定である。また、Time Machine ではディスクイメージをブロック（バンドルという）単位に分割し、重複するブロックをハードリンクにして、重複を排除している。このように重複排除が OS またはファイルシステムの機能として提供されるようになれば、物理資源を効率よく利用することができる。

3. VLSD

本節では大規模ストレージ構築のための VLSD(Virtual Large Scale Disk)ツールキットについて述べる。VLSD は大規模ストレージ構築のためのツールキットであり､Java

によるソフトウェア RAID 実装と NBD 実装を含む。VLSD は 100% pure Java であり、Java が動作するプラットフォームの上なら VLSD も動作する。そのため Windows や Linux が混在する環境に適している。

VLSD を用いると OS に制約されることなく NBD デバイスと RAID を自由に組み合わせることができる。最低限必要な NBD デバイスはファイルサーバの 1 つである。

Linux の nbd-server コマンドや Windows の nbdsrvr コマンドは単一ファイルを仮想ディスクとして公開する。そのため 4GB の制約がある FAT32 で動作させた場合、120GB/2GB=60 プロセスの NBD サーバを稼動させる必要がある。VLSD は複数のファイルを単一の JBOD にまとめて公開することができる。

ただし、VLSD の NBD サーバを用いた場合、ポート数の制約がある。ディスクを利用している最中はコネクションを維持するため NBD デバイスごとにポートを 1 つ消費する。ポート数はデバイス数より大きいため余裕があるが、その資源は無限ではない。数千台までは直接構成可能であるが、それを超える場合は間接的に、階層的に構成する必要がある。また、意図的に負荷を分散するために階層化することもある。この問題を解消するためにポート数に制限されない RMI を用いたディスクサーバも用意した。

図 1 に VLSD を用いて分散ストレージを構成した例を示す。クライアントは 500 台存在し、その OS は Linux または Windows である。それらはそれぞれ NFS、CIFS で 1 台のファイルサーバと通信する。クライアントは同時に NBD サーバでもある。各クライアントでは空き容量を束ねた 1 つの NBD サーバが稼動する（従来のシステムでは複数の NBD サーバを稼動させなければならない場合があった）。ファイルサーバは Samba の稼動する Linux マシンである。ファイルサーバでは、クライアントの分だけ NBDDisk（後述）を作成し、22 の NBDDisk から 1 つずつ合計 22 の RAID6 を作成し、最後に 22 の RAID6 から 1 つの RAID6 を作成する。この RAID0File を NBD サーバで公開し、自分自身の NBD デバイスで参照する。

VLSD ツールキットには以下のクラスが含まれる。

Disk

すべての仮想ディスクのインターフェースを規定する。

FileDisk

単一ファイルによる固定容量ディスク。論理的な容量と物理的な容量は正確に一致する。java.io.RandomAccessFile により実装される。

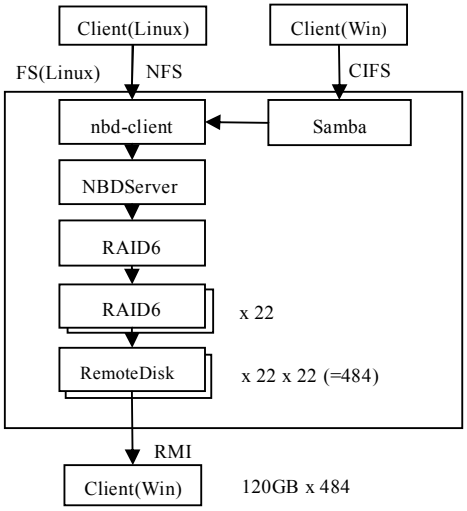

図 1　VLSD のシステム概要
Figure 1　The system overview of VLSD

VariableDisk

単一ディスクにより容量可変ディスクを作成するラッパー。8KiB を単位とする 1K 分木で管理する。葉ノードには 8KiB のデータが格納される。中間ノードには 1024 個の 64b(8B)ポインタが格納される。ノードは必要に応じて割り当てられる。６階層で 8EiB-1 まで拡張できる。データ以外の管理情報が保存されるため物理的な容量は 0.1% 増加する。容量可変ディスクを実現するため、Disk インターフェースには容量を追加する API が定義されている。

NBDDisk/NBDServer

NBD デバイスのクライアント。NBDServer と NBD プロトコルで通信する。その他の NBD サーバ実装（例えば、nbdsrvr）とも通信できる。

RemoteDisk/RemoteServer

遠隔デバイスのクライアント。RMI プロトコルで通信する。RemoteDisk に対応するサーバは DiskServer である。

SecureRemoteDisk/SecureRemoteServer

アクセスキーによる安全な遠隔デバイスのクライアント。RMI プロトコルで通信する。SecureRemoteDisk に対応するサーバは SecureDiskServer である。

CipherDisk

ディスクの内容を暗号化する。任意の暗号を利用できる。

JBOD

複数のディスクを直列に連結したディスク。冗長性がなく、容量増のために用いられる。各ディスクの容量は一様でなくてもよい。ストライピングを行わないため容量は単純に総和となる。例えば、100GB, 120GB, 160GB を連結すると 100+120+160=380GB になる。JBOD に対して連続的に逐次アクセスすると特定の部分ディスクに負荷が集中する。

RAID*n* (*n*=0,1,4,5,6)

各 RAID クラスの実装。RAID0 は HW RAID と異なり、JBOD と有意な差はない。RAID4, 5 は 1 耐故障である。RAID5 は HW RAID と異なり、RAID4 との有意な差はない。RAID6 は 2 耐故障である。P+Q 方式を採用している。

FaultDisk

耐故障性評価をおこなうためのクラス。一種のプロキシーであるが、故障を設定すると擬似的に故障を発生させる。

UnionDisk

読み取り専用と読み書き可能な２つのディスクを組み合わせて、差分を作成する。

これらのクラスは自由に組み合わせることができる。例えば、RAID6 を 2 段階で組み合わせると RAID66 を構築できる。

4. バックアップシステム

前節までの議論から、内部統制のためのバックアップシステムの要件は以下のように定めることができる。

- 遠隔バックアップ
- 世代管理可能であること
- ニアライン（遅くてもアクセス可能）であること
- 重複排除されること

本節では、上記要件を満たすバックアップシステムを設計する。

まず、大規模災害が生じても全滅しない遠隔地に少なくとも 1 つの事業所があると仮定する。また、それら事業所の余剰資源はバックアップを保管するのに十分な容量があるとする。そこで、本社からそれら事業所へ遠隔バックアップを行うこととする。VLSD には遠隔資源をアクセスするための RemoteDisk がある。

遠隔アクセスは局所アクセスに比べて非常に遅い。特にディザスタリカバリのためには遠い場所を選択する必要があるため、インターネットを介する必要がある。すると、そのスループットは極端に遅くなる。

このような極端な性能差のディスクを RAID1 で構成すると全体の性能は遅い方に抑制されてしまう（この性質は実装に依存する）。そこで、遠隔ディスクの前段にキャッシュを挿入し、遅延を隠ぺいする必要がある。しかし、キャッシュはミスすると効果がない。バックアップ周期を 1 日とした場合、一日分の読み書き量をすべてメモリにキャッシュすることは困難である。

そこで、我々は RAID1 の代わりに PrimaryBackupDisk を用いる。PrimaryBackupDisk は、通常時はプライマリディスクのみにアクセスし、同期時にプライマリからバックアップへコピーを行う。同期は任意の時点で可能であるが、原則としてクローズ時に行う。変更のあったブロックのみバックアップへコピーされる。図 2 に PrimaryBackupDisk の構造を示す。局所ディスクをプライマリとし、遠隔ディスクをバックアップとする。

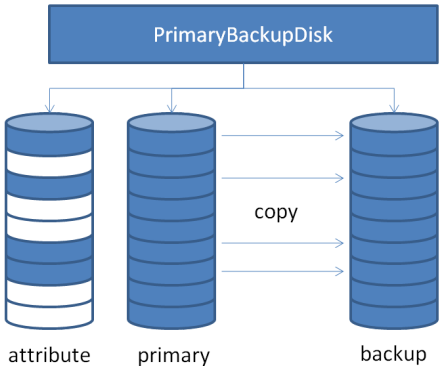

図 2　PrimaryBackupDisk

Figure 2. PrimaryBackupDisk

バックアップデータはビジネス上必要なデータであり、機密情報が含まれる。よって、第三者に盗聴あるいは改ざんされないように保護されなければならない。VLSD のセキュリティ機能には仮想ディスク全体を暗号化する CipherDisk が含まれる。これを PrimaryBackupDisk と RemoteDisk の間に挿入することで、安全な通信が可能となる。また、複数の遠隔ディスクに分散し、それらを RAID で構成することで改ざんを修正することができる。遠隔ディスクが多数の場合は RAID1 で複製する。少数の場合は RAID6 などを用いてもよい。ある場所に送信されるブロックが不連続になるため盗聴が困難となる。CipherDisk の前に RAID を配置してもよい。複数の異なる鍵を用いることができる。

重複排除が ZFS で可能であるならば、バックアップシステムの要件からはずすことができる。その場合でも差分バックアップは必要である。

以上の議論から、ローカル側のシステム構成は図 3 のようになる。

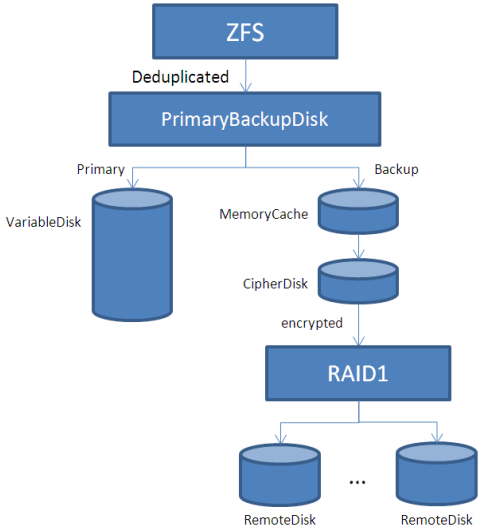

図 3　ローカルの構成
Figure 3. local configuration

次に、遠隔側のシステム構成について述べる。遠隔側では世代管理が課題となる。

VLSD では、ファイルレベルの世代管理はできない。ブロックレベルの世代管理を行うには２つの方法がある。１つはブロック単位で書き込み時間を記録することである。しかし、この方法は書き込み時間を記録する領域を必要とするため資源の利用効率が低いことと、同じファイルに属するブロックの書き込み時間にはほとんど差がないことから、無駄の多い方式である。もう一つはある時刻でデバイスを切り替える方式である。デバイス自身がバックアップの時刻を表す。この方法は無駄がないが、デバイスを切り替えたタイミングでしか復旧できない。しかし、OS のサポートなしに可能な方式としては現実的である。本論文では、後者の方式を採用する。

図 4 に VLSD を用いた世代管理システムの構造を示す。VLSD では、UnionDisk を用いて差分を保存することができる。この図では UnionDisk の右子ノードを読み取りのみ可能なディスク、左子ノードを読み書き可能なディスクとしている。左子ノードを差分ディスクと呼ぶ。原則として毎日、差分を求める。この差分はリンクされ、ディスクのリストとなる。i 番要素は d+1-i 日以前のデータである。このようなしくみによって Time Machine のような自由な時間旅行を行うことができる。

データを無矛盾でバックアップするには、すべての読み書きを正常に終了させる必要がある。そのため、バックアップ時にいったんディスクを close/unmount し、オフラインにする。そして、新たな差分ディスクと UnionDisk を追加して、再度 open/mount する。差分ディスクは VariableDisk であり、実際に書き込まれたデータ量しか資源を消費しない。差分ディスクの論理的なサイズは全体に等しい。

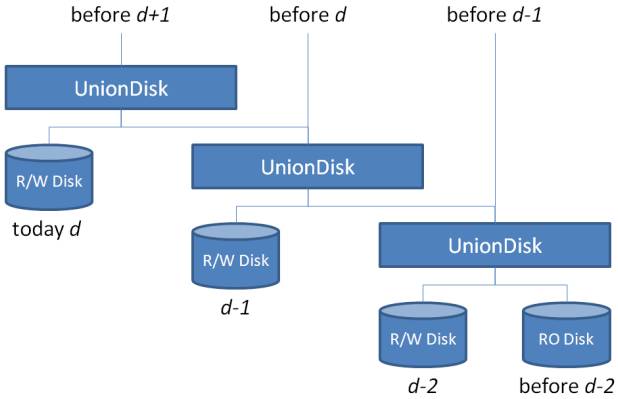

図 4　バックアップの世代管理
Figure 4. The version control of backup

任意の世代のディスクイメージは擬似的な Read Only デバイスとして提供される。ここで書き込みを許可すると改ざんの危険性がある。また、UnionDisk は共有されるため一貫した書き込みを行うことができない。よって、読み取りのみ許可されなければならない。

このような方式はニアラインの要件を満たす。この構造は線形リストであるため、過去のデータほどアクセスに要する時間は長くなる。よって、適切な間隔でキャッシュを挿入するか、あるいはマージすることが望ましい。後者の場合、戻ることのできる時間の選択肢が少なくなる。

5. 評価

ここでは、バックアップシステムを容量効率と性能の面から評価する。

はじめに本学を例にバックアップに必要な容量の概算を示す。保存する対象とするログは内部統制上必要な要素に限る。本大学の場合、40GB のサーバ 20 台、110GB のデータベース 1 台のフルバックアップを行っている。それらの容量を単純に合計すると約 1TB/日である。月ごとにフルバックアップを保存しておくとすると 1 年で約 12TB となり、5 年では約 60TB となる。これに先に述べた操作ログの総容量を足すと、必要となるストレージの総計は約 63TB となる。我々は既に 60TiB のストレージを試作しているので、63TB のバックアップデータを収容することは問題ではない。

次に、提案システムにおける容量の試算を行う。まず、40GB HDD を持つサーバの 1 日分の差分は多くとも 4GB であろう。ここで、仮に 1 日に 10%更新されると仮定する。この前提が正しければ、あるサーバでは、プライマリディスクは 40GB、バックアップは 4GB である。プライマリディスクの容量は変化しない。しかし、バックアップディスクの容量は単調に増加する。その増分は、論理容量で 40GB/日、物理容量で 4GB/日である。これを単位として世代管理が行われる。毎日バックアップを続けると 1 月で物理容量 120GB に達する。また、1 年で物理容量 1.44TB、5 年で物理容量 7.2TB になる。20 台では物理容量 144TB になる。データベースサーバを加えると 164TB になる。これは我々が VLSD で試作したストレージの高々2 倍の規模である。よって、十分構築可能な規模である。

次に、性能について評価する。

ネットワークのスループットを 100Mbps とし、1 か所の遠隔地に転送する。転送量は毎日 98GB(=4*20+11)である。よって、約 2 時間で転送できる。これはネットワークがボトルネックとなる場合の所要時間である。

次に、VLSD がボトルネックとなる場合の所要時間を評価する。提案システムは 100GB を約 6700 秒、すなわち約 2 時間で転送できる。この結果から、一晩で十分バックアップを行うことが可能であるといえる。

現在の実装では、ネットワークとサーバの処理能力はほぼ等しい。よって、ネットワークが混雑すればネットワークがボトルネックとなる。ネットワークの実効性能が 50Mbps になっても 4 時間で転送できるので、実際の運用ではあまり問題にならないと考えられる。ネットワークの実効性能がさらに低下した場合は、複数の遠隔地に並行にバックアップする必要がある。

6. まとめ

本論文では、内部統制のためのバックアップシステムを提案し、VLSD を用いて構築した。このシステムは、稼働システムに与える負荷を抑え、効率よく差分を作成し、任意の時間に戻ることができる。しかも、従来システムに比べて安価に実現できる。本論文では、提案システムが十分実用的な性能を持つことを示した。

本論文ではバックアップシステムを仮想的に構成したが、今後は実機を用いて構成し、性能等を評価したい。

参考文献

1) Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy H. Katz, and David A. Patterson: “RAID: High-Performance, Reliable Secondary Storage,” ACM Computing Surveys, Vol. 26, No. 2, pp.145-185, June 1994

2) Erianto Chai, Minoru Uehara, Hideki Mori, Nobuyoshi Sato: "Virtual Large-Scale Disk System for PC-Room", LNCS 4658, Network-Based Information Systems, pp.476-485, (2007.9.3-4)

3) Erianto Chai, Minoru Uehara, Hideki Mori: "A Case Study on Large-Scale Disk System concatenating Free Space", In Proceedings on IEEE 2nd International Conference on Innovative Computing, Information and Control(ICICIC2007) (2007.9.5-7)

4) Erianto Chai, Minoru Uehara, Hideki Mori: "Evaluating Performance and Fault Tolerance in a Virtual Large-Scale Disk", In Proceedings of 22nd International Conference on Advanced Information Networking and Applications(AINA2008), pp.926-933, (2008.3.28)

5) Erianto Chai, Minoru Uehara, Hideki Mori: "Case Study on the Recovery of a Virtual Large-Scale Disk", Springer LNCS Volume 5186/2008 Network-Based Information Systems(NBIS2008), pp.149-158(2008.8.21)

6) Minoru Uehara: "Security Framework in a Virtual Large-Scale Disk System", In Proc. of IEEE 10th International Workshop on Multimedia Network Systems and Applications(MNSA2008), pp.30-35, (2008.6.20)

7) Erianto Chai, Minoru Uehara, Makoto Murakami, Motoi Yamagiwa: "Online Web Storage using Virtual Large-Scale Disks", In Proc. of the Third International Workshop on Engineering Complex Distributed Systems (ECDS-2009), pp.512-517, (2009.3.16-19)

8) Katsuyoshi Matsumoto, Minoru Uehara: "N-nary RAID: 3-resilient RAID based on an N-nary number", In Proceedings of 23rd International Conference on Advanced Information Networking and Applications(AINA2009), pp.249-255, (2008.5.26)

9) Minoru Uehara: "Combining N-ary RAID to RAID MP", In Proc. of 1st International Workshop on Information Technology for Innovative Services(ITIS2009) in conjunction with 2009 International Conference on Network-Based Information Systems(NBiS2009), pp.451-456, (2009.8.19-21)