

大規模仮想ディスクにおける耐間欠故障法

上原 稔†

低コストの大容量ストレージに対する要求は非常に高い。我々は、空容量を集約してこのようなストレージを構築するために、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)を開発した。VLSDでは、RAIDを拡張して高信頼な大容量ストレージを実現できる。しかし、現在の実装では停止故障にしか対応できていない。本論文では、間欠故障に対する耐故障性を実現するための方式をいくつか VLSD に導入する。1つは再試行を行う RetryDisk であり、もう1つは多数決を行う VotedRAID1 である。本論文では、これらの仕組みを述べる。また、それぞれの耐間欠故障能力を評価し、有効性を検証する。

Intermitted Fault Tolerance in Virtual Large Scale Disks

Minoru Uehara†

Recently, the demand of low cost large scale storages increases. We developed VLSD (Virtual Large Scale Disks) toolkit for constructing virtual disk based distributed storages, which aggregate free spaces of individual disks. However, current implementation of VLSD can mask only stop failure but cannot mask other kinds of failures such as intermitted failure. In this paper, we introduce two classes to VLSD in order to increase the intermitted fault tolerance. One is RetryDisk which retries to read/write at failures, another is VotedRAID1 which masks failures by majority voting. In this paper, we describe these classes in detail and evaluate their fault tolerance.

1. はじめに

インターネットの発達により多くの情報を容易に入手できるようになった。例えば、インターネットの主要なサービスである WWW では Web ページを収集し、その関連を調べることで有意な関連を発掘する Web マイニングが行われている。また、センサーネットワークの発達により多数のセンサーが日々情報を送信している。さらに、日々の活動を記録するライフログも研究されている。このような多量の情報を処理するために

は、大規模ストレージが必要である。

しかし、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。

このような分散ストレージは、通常の集中ストレージより高い信頼性が要求される。そこで、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。さらに、VLSD はファイルシステムに依存しないため、ZFS だけでなく NTFS でも利用できる。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。我々は、VLSD を用いて 8EiB ストレージを仮想的に試作した。

大規模ストレージサービスでは多量のディスクを運用する必要がある。しかし、ストレージのディスク数が増加すると信頼性が減少するという問題が生じる。一般にストレージの信頼性を増すには RAID を用いる。RAID は PC で手軽に利用できるほどコモディティ化しているが、ストレージサービスの規模は PC と比較にならない。コモディティ RAID は単一故障を前提としている。しかし、数百～数千の規模になると複数の故障が同時発生する可能性が高い。

VLSD では RAID の改良により高信頼性と高容量効率を両立する。しかし、VLSD の RAID は停止故障を前提としているため、現実のストレージ運用に生じる様々な種類の故障には十分に対応できているとは言えない。一般に、故障の種類には停止故障、遷移故障、間欠故障などがある。このうち遷移故障は修復されるまでは停止故障とみなせるため、従来の VLSD でも対応可能である。そこで、本論文では、未対応である間欠故障について議論する。

間欠故障は、偶発的に発生する故障であり、あくまでも一時的な障害である。修理せずとも復帰することがある。代表的な間欠故障はデバイスの読み取りエラーやパケットロスである。停止故障は、故障が発生するとそれが修復されるまでは故障状態が続く、その意味では予測しやすい故障である。しかし、間欠故障は予測が困難である。停止故障と同等に扱うと頻繁に不必要な修復を繰り返すことになりかねない。

我々は、このような間欠故障に対応する基本的な方式として再試行と多数決を採用し、それぞれに対応した VLSD クラスとして RetryDisk と VotedRAID1 を実装した。本論文では、これらのクラスの概要とその有効性について述べる。

本文の構成は以下の通りである。2 節で関連研究として VLSD について述べる。3 節では再試行に基づく RetryDisk の実装と、その評価について述べる。4 節では多数決に基づく VotedRAID1 の実装とその評価について述べる。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. 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 サーバを稼働させる必要がある。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。

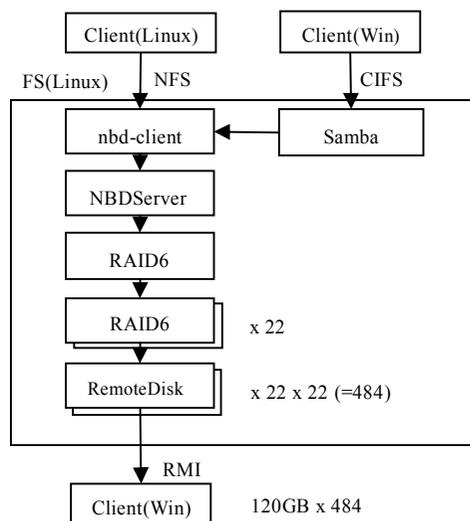


図 1 VLSD のシステム概要

Figure 1 The system overview of VLSD

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 サーバが稼働する（従来のシステムでは複数の NBD サーバを稼働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 の稼働する 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。

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/SecureRemoteServer⁶⁾

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。

WebDisk⁷⁾

Web サーバの資源を遠隔デバイスとして利用する仮想ディスク。WebDisk は、Web サーバで動作する REST 型 Web サービスにアクセスする。

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樣でなくてもよい。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 $100+120+160=380$ GB 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。一種のプロキシであるが、故障を設定すると擬似的に故障を発生させる。

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。

3. RetryDisk

耐故障性を実現するには何らかの冗長性を必要とする。再試行は時間冗長に基づく耐故障性手法である。再試行自体は新しいアイデアではない。TCP/IP や NFS でも使われている。しかし、単純な割に有効であるからこそ使われているのであり、それを避ける理由はない。ここでは、再試行により間欠故障に対する耐故障性を実現する仮想ディスクとして RetryDisk を提案する。

VLSD の RAID は停止故障にのみ対応している。なお、ここでいう停止故障には修復により回復する遷移故障も含まれる。要は、故障時に一定の状態で作動停止することを意味する。修復されるかどうかは問題としない。このような故障は行儀のよい故障である。しかし、故障は停止故障だけではなく、間欠故障も起き得る。間欠故障は、読み取りや転送のエラーなど正常な機器でも一時的に発生する。機器自体は正常であるため、受信したデータを正しいとみなしてしまう可能性がある。このようなエラーは ECC やチェックサム、ダイジェストなどで検出することができる。多くの場合、故障の再現性がないため、再試行によって正しいデータを得ることができる。

間欠故障が起きる状況は限られている。ディスクからの直接の読み書きや UDP での転送である。これら以外は OS やプロトコルにより隠ぺいされる。よって、すべてのクラスが間欠故障に対応する必要はない。VLSD では、RetryDisk を含め複数の解決法を提供するが、それをどこで適用するかは利用者の判断にゆだねる。

再試行を行うには前提として間欠故障を検出しなければならない。VLSD では、ChecksumDisk あるいは DigestDisk を用いて間欠故障を検出することができる。ChecksumDisk および DigestDisk はいずれも AttributeDisk のサブクラスである。AttributeDisk はブロック単位に任意の固定長属性を付加できる。ChecksumDisk では Checksum を、DigestDisk では MessageDigest をそれぞれ固定長属性として付加する。

ChecksumDisk と DigestDisk はいずれも読み書き時にデータのエラーを検出する。そしてフェールセーフのために例外を発生する。

RetryDisk は N 回の再試行を行う仮想ディスクである。RetryDisk は、ChecksumDisk または DigestDisk が発生したエラーを検出すると、指定された回数 N の再試行を行う。多くの間欠故障は、数回の再試行の後に正常な値を返す。しかし、いくら繰り返しても正常な値を返さない場合は、故障したとみなして、RetryDisk もエラーを返す。

間欠故障をシミュレーションするためのクラスとして IntermittentFaultDisk クラスを導入する。このクラスでは、読み書きの際に一定の確率で間欠故障を発生させる。

これらのクラスを用いて再試行のしくみを構成すると図 2 のようになる。IntermittentFaultDisk は間欠故障を擬似的に発生させる。間欠故障は ChecksumDisk で検出され、RetryDisk が再試行することでマスクされる。

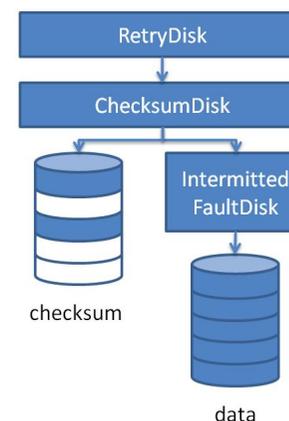


図 2 VLSD における再試行のしくみ
Figure 2 Retry Mechanism in VLSD

図中に現れるクラスの概要を要約すると以下のようになる。

AttributeDisk

ブロック単位に任意の固定長属性を付加する。データを格納するディスクと属性を

格納するディスクを個別に扱う。

ChecksumDisk

AttributeDisk のサブクラスであり、チェックサムを固定長属性とする。チェックサムは Checksum 仕様に基づく 32 ビットである。CRC だけでなく単純合計や XOR でもよい。計算が簡単であるため比較的早い。読み書きの際にチェックサムと比較して不一致ならエラーを発生する。

DigestDisk

AttributeDisk のサブクラスであり、ハッシュを固定長属性とする。ハッシュは MessageDigest 仕様に基づく。各仕様によりビット数は異なる。MD5 や SHA1 などがある。計算が複雑であるため比較的遅いが、高い確率で誤りを検出できる。読み書きの際にハッシュと比較して不一致ならエラーを発生する。

RetryDisk

読み書きでエラーを検出すると N 回の再試行を行う。N 回を超えるとエラーとなる。

IntermittedFaultDisk

読み書きの際に一定の確率で間欠故障を発生させる。

次に本方式の評価を行う。評価の基準は耐故障性と性能である。図 2 のシステムで 400 回の読み書きするベンチマークを実行した。IntermittedFaultDisk にビットエラー率 (Bit Error Rate, BER) を与え、成功率をシミュレーションした。結果を図 3 に示す。ここで、X 軸は再試行回数 N、Y 軸は回復成功率である。BER が 10^{-8} 以下は N=1 でも既に 100%であった。通常、普及品の HDD でも訂正不可能ビットエラー率(Unrecoverable Bit Error Rate, UBER)はビット当たり 10^{-14} である。よって、実用上は N=1 で十分であるといえる。さらに、BER が 10^{-5} まで増加しても 15 回程度の再試行で耐えることができる。ただし、BER が 10^{-4} 以上では N=256 でも回復できない。よって、実用上の限界は BER= 10^{-5} であるといえる。

RetryDisk の性能は再試行回数の分だけ低下する。BER= 10^{-4} でも 1 回は再試行しないと 100%とはならない。この 100%にしても 100%に近いという意味に過ぎず、完全に安全というわけではない。まして BER= 10^{-5} では 15 回もの再試行を必要とする。これらの結果から再試行方式は性能低下が大きいことが分かる。

また、再試行の度に Checksum の再計算が行われたり、参照されたりするため、さらに性能が低下する。この問題は ChecksumDisk に簡単なキャッシュを導入することで解決できる。しかし、再試行による遅延は本質的に改善できない。よって、本方式は比較的故障率が小さな場合に適用すべきである。

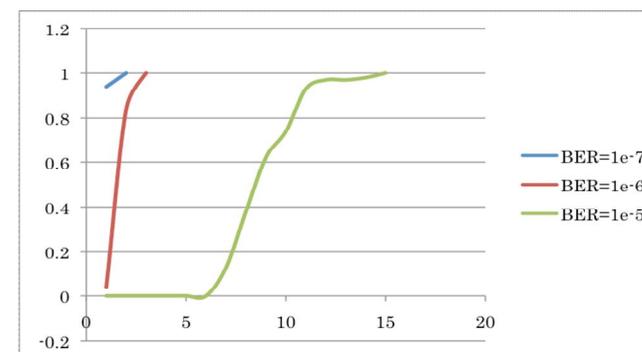


図 3 RetryDisk の再試行 N における成功率
Figure 3 The success rate of RetryDisk in N retry

4. VotedRAID1

耐故障性を実現するもう一つの冗長法は空間冗長である。代表的な空間冗長手法が NMR(N Modular Redundancy)である。

実を言えば、3 番目の冗長法である情報冗長もある。これには RAID のパリティなどが含まれる。よって、RAID で間欠故障をマスクすることも可能ではある。しかし、そのためには RetryDisk と同様に間欠故障を検出する必要がある。間欠故障が検出できれば、時間冗長でも情報冗長でも耐故障性を実現できる。しかし、逆の言い方をすれば、間欠故障が検出できなければ時間冗長も情報冗長もそのままでは適用できない。ここでは、間欠故障が検出できない場合を想定する。例えば、前節の評価ではチェックサムを信用した。しかし、データとチェックサムの両方が改ざんされている場合、エラーは検出されない可能性がある。そこで、間欠故障が検出できない前提で空間冗長を適用する。

NMR では、確率的に多数派が信頼できると仮定し、多数決により多数派を採用する。これにより故障が検出できなくても誤りを訂正できる。このように多数決を行う RAID1 を VotedRAID1 と名づけて、導入する。VotedRAID1 はすべての要素ディスクが同じ値を持つという前提から RAID1 に属する。

通常の RAID1 は多数決を行わない。性能を重視した RAID1 では、早く帰ってきた値をそのまま信頼する。しかし、この方法では誤った値が早く帰ってきててもそれを排除できない。VotedRAID1 では誤りおよび故障を排除できる。一方で、RAID1 は最後の 1 台が故障するまで頑健に稼働する。しかし、VotedRAID1 は 1 台が停止し、偶数

台になると、多数決すら行えなくなる。その意味では、停止故障に対する信頼性は必ずしも高いとは言えない。

図4に VotedRAID1 を用いたシステムの構成を示す。原則として VotedRAID1 は奇数台の要素ディスクを持つ。図では評価のためにデータディスクの前に IntermittedFaultDisk を挿入している。実際の運用では、これらの IntermittedFaultDisk は必要ない。

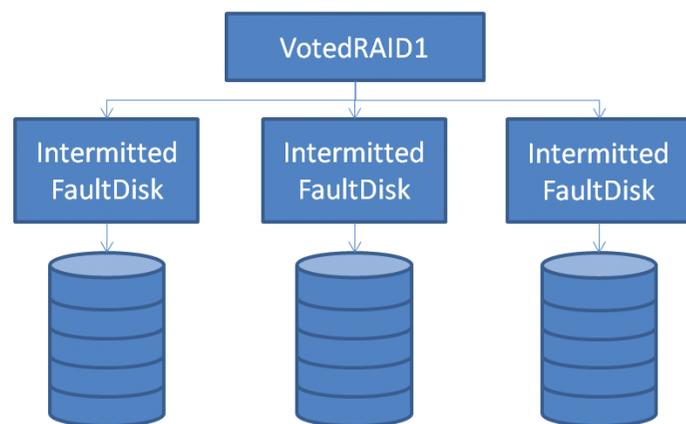


図4 VotedRAID1 の構成
Figure 4 The structure of VotedRAID1

図中に現れるクラスの概要を要約すると以下ようになる。

VotedRAID1

奇数台の要素ディスクに同じ内容を書き込み、読み取り時に多数決を行う。

次に、VotedRAID1 の耐間欠故障性を評価する。ベンチマークは RetryDisk の評価で使用したものと等しい。結果を図5に示す。

VotedRAID1 のディスク台数 N は RetryDisk の再試行回数 N に相当する。その意味では VotedRAID1 の成功率は同一条件の RetryDisk に比べると2桁ほど小さい。RetryDisk は1回でも正しいデータを受信すれば成功するが、VotedRAID1 は正しいデータを過半数受信しなければ成功しない。そのため単純な比較では VotedRAID1 の耐故障性能は RetryDisk に及ばない。ただし、VotedRAID1 はエラー検出機能がなくても利用できる利点がある。よって状況に応じて両者を使い分ける必要がある。

また、VotedRAID1 ではディスク台数 N に逆比例するように成功率が低下する。図

からは読み取りにくい、すべての BER で同様の傾向が見られた。これは再試行回数 N に比例して成功率が高まる RetryDisk と逆の傾向になる。よって、VotedRAID1 は $N=3$ で利用すべきである。

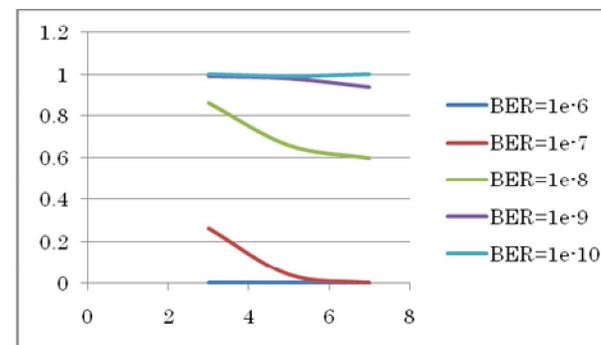


図5 N台からなる VotedRAID1 の成功率
Figure 5 The success rate of VotedRAID1 having N disks

多数決は定評のある高信頼化技術であるので、この結果は一見正しくないように見えるかもしれない。しかし、多数決の耐故障性は、修復率が低い時には決して高いとはいえない。例えば、故障率 $p=0.01$ のとき、 $N=3$ の NMR(3MR)の MTTF は 82 であり、 $N=5$ の NMR(5MR)の MTTF は 77 である。なお、MTTF の単位はシミュレーションの論理時間である。このように N を増やしてもかえって耐故障性は低下する。ただし、修復率が高いと逆に N を増やすと耐故障性は向上する。NMR は修復率が高い場合に適可能な耐故障性技術であるといえる。実際、表では非冗長化単一モジュール(SM)の MTTF は 99 であるから何もしないより悪くなっている。しかし、 $N=1$ における VotedRAID1 の成功率は 0%であったので、少なくとも多数決の効果はある。修復率が低いと故障が蓄積する。その結果、多数決で修復可能な範囲を超えてしまう。ビットエラーでもブロック単位ではエラーが蓄積される。

表1 NMR の MTTF

Table 1 MTTF of NMR

	SM	3MR	5MR
MTTF	99	82	77

ここで、VotedRAID1の多数決機能について考察する。図5ではRetryDiskと同様にブロック全体を比較して多数派ブロックを決定した。しかし、この方法ではすべてのブロックでエラーが発生すると正しいデータそのものが存在しないため決して成功しないことになる。しかし、局所的なエラー箇所を比較すれば対応する箇所同士で多数決可能であり、修復できる可能性もある。そこで、ビット単位の多数決を行った。結果を図6に示す。結果として有意な差はなかった。

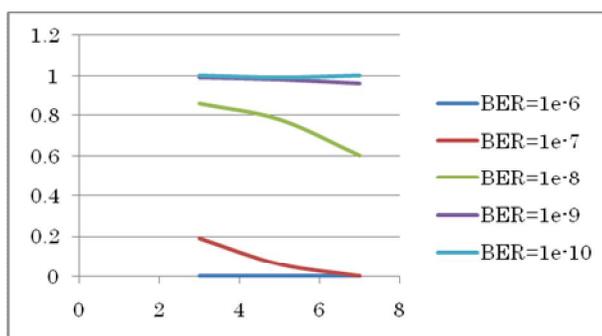


図6 ビットレベル多数決
Figure 6 Bit level voting

最後にVotedRAID1とRetryDiskの無故障時の性能を比較する。表2はベンチマークの処理時間である。RetryDiskの処理にはChecksumDiskの処理も含まれている。一方、VotedRAID1はブロックのハッシュで多数決を行っている。

表2 VotedRAID1とRetryDiskの性能比較
Table 2 Performances of VotedRAID1 and RetryDisk

	RertyDisk	VotedRAID1
処理時間[s]	1.516	1.344

5. まとめ

本論文では、ビットエラーなどによる間欠故障に対する耐故障性を向上させるために、大規模仮想ディスクに再試行に基づくRetryDiskと多数決に基づくVotedRAID1を導入し、その有効性を評価した。いずれも通常の利用範囲では十分に有効であるが、

極限的な状況ではRetryDiskの方がVotedRAID1より耐間欠故障性能に優れる。ただし、RetryDiskが機能するには故障検出が前提となり、検出不可能な状況ではVotedRAID1を使用するしかない。

通常、間欠故障はファイルシステムによって自動的に修復される。そのため、仮想ディスクが直接間欠故障に対応しなければならないことはほとんどない。しかし、物理ディスクは間欠故障に対応出来なければならない。間欠故障に対応したことで、VLSDで物理ディスクを操作する可能性が開けた。

謝辞 本研究は科研費基盤(C)「ストレージ統合型軽量クラウドの研究(22500098)」により援助されています。

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)