2007－DPS－133 (12)
2007／11／27

大規模仮想ディスクにおける修復に関する考察

チャイ エリアント　　上原 稔　　森 秀樹
東洋大学大学院工学研究科情報システム専攻

今日、大量のデータが氾濫している。そのデータを蓄積することが１つの課題になっている。現在の普及品の HDD は安価であるにもかかわらず、ストレージシステムは高価である。そこで、我々は安価な普及品とソフトウェアだけで大規模ストレージを構築するために VLSD(Virtual Large-Scale Disk)ツールキットを開発した。また、これを用いて PC の空き容量を集積し、大規模ストレージを試作した。しかし、そのストレージの信頼性は MTTR に依存する。本論文では、試作ストレージの MTTR を測定し、実用性を考察する。

A Case Study on Recovery of Virtual Large-Scale Disk

Erianto Chai　　Minoru Uehara　　Hideki Mori
Toyo University Graduate School, Dept. of Open Information System

Today, massive data is flooding. One of major issues is to store massive data. Recently, although commodity HDD is very cheep, appliance storage system is very expensive. So, we developed VLSD (Virtual Large-Scale Disk) toolkit in order to construct a large-scale storage using only cheep commodity hardware and software. And we also developed a prototype of large-scale storage system by collecting free disk spaces of PCs using VLSD. However, the reliability of its storage depends on MTTR. In this paper, we evaluate MTTR of our prototype and then discuss the efficiency.

1　はじめに

近年、ストレージサービスに対する要求が高まっている。

従来のアプリケーションでは、データを共有するためにデータベースを用いていた。しかし、近年 Web サービスが普及するにつれ、SOA ベースのアプリケーションが増えてきた。これらのアプリケーションでは、異なるベンダーから提供されるサービスを組み合わせるマッシュアップが行われる。このようなアプリケーションでは、インターフェースを Web サービスで統合するため、ストレージサービスを利用する必要がある。ストレージの Web サービスは Amazon などが提供している。

YouTube やニコニコ動画などの動画投稿サイトはマスコミュニケーションを補完するジャーナリズムとして定着すると予想される。USA では、大統領選にも利用されている。このようなサイトでは、多くの動画を保管するために大容量のストレージを必要としている。

ライフログでは、人のあらゆる活動を記録する。そのためには一人当たり数 TB 以上の容量を必要とする。

企業は内部統制のためにログを保管する必要がある。企業の所有する全 PC から集められたログは膨大である。これを効率的に管理するため、情報ライフサイクル管理に基づく 3 階層ストレージが使用される。

このようなストレージサービスに対する要求は HDD 技術の進歩を促した。その結果、安価で大容量なディスクが入手可能となった。しかし、アプライアンス系のファイルサーバで使用している HDD はその普及品より 10 倍高価である。このように現在のストレージのコストは適切でない。ストレージが高価な理由は専用ハードウェアにある。普及品のハードウェアと専用ソフトウェアによって問題を解決できる。

我々は、大規模ストレージを構築するためのツールキット VLSD(Virtual Large-Scale Disk)を開発した。VLSD は 100 % pure Java であるため、プラットフォームに依存しない。我々は VLSD を用いて 500 台の PC からそれぞれ空き容量 170GB を集め 70TB のストレージを構築するシステムを試作した。このシステムでは RAID66(2 階層の RAID6)を用いて十分な MTTF を実現している。

ストレージ全体の信頼性は MTTR にも依存する。ディスクが多くなるほど容量は大きくなるが、MTTF は減少する。それゆえ MTTR を最小化することが必要である。MTTR を最小化するには、故障ディスクをスペアに交換する時間（交換時間）とデータを修復する時間（修復時間）をそれぞれ短縮する必要がある。交換時間はホットスペアが存在するとき無視できる。

本論文では、VLSD を用いた試作ストレージにおいて修復時間を測定し、システム全体の信頼性を評価する。

本論文の構成は以下の通りである。第 2 章では関連研究について述べる。第 3 章では VLSD について述べる。第 4 章では VLSD における修復方法について述べる。第 5 章では評価する。第 6 章では得られた知見を考察する。最後に結論を述べる。

2 関連研究

2.1 RAID

今大規模ストレージの信頼性を上げるために RAID（Redundant Arrays of Inexpensive Disks）[1][2]を用いる。RAID とは、記憶すべきデータと障害回復のための冗長データを複数のハードディスクドライブに分散して格納することで、パフォーマンス（性能）とフォルトトレラント（耐障害）性を同時に確保するための技法である。冗長データの種類と各ディスクドライブへのばらまき方によって、RAID は 0 から 6 までのレベルがある。

RAID0 は冗長性なしのストライピングである。容量を拡大することができる。RAID1 はミラーリングである。容量はまったく増えない。RAID2 は、各ビットをディスクに分散させ、ECC でデータ誤りを訂正する方式である。RAID2 はパリティ方式に比べて優位性が少ないためほとんど使われない。RAID3 はパリティによって誤りを訂正し、ビットまたはバイト単位でストライピングする。通常、パリティは専用ディスクに保存する。RAID4 はブロック単位でストライピングする点が RAID3 と異なる。RAID3 同様パリティは専用ディスクに記録される。パリティ専用ディスクを用いる方式はそれがボトルネックとなる。RAID5 はパリティをディスクに分散して保存する。ボトルネックが存在しないため性能が高い。RAID 5 システムでは、オペレーターは時々駆動中のドライブを引き間違って、2 つのドライブが同時に失敗するようなことがある。そこで、RAID 5 より信頼性が高い RAID 6[5]が採用される。RAID6 は 2 重障害に対応することができる。つまり、同時に 2 台のディスクが故障してもデータを失うことなく回復できる。RAID6 ではパリティが 2 二つある（P パリティと Q パリティと名づけ）。P パリティは RAID5 と同じくすべてのディスクのブロックデータの XOR から生成される。RAID6 で使われる Q パリティは P パリティの計算より複雑である。Q パリティの生成には Galois Field 演算によって求められる。GF（Galois Field）は代数学では有限体と呼ぶが、計算機関連の分野では、ガロア体またはガロア域とも呼ぶ。GF は有限数の要素を含んでいる値のセットである。2^8 個要素の GF は GF(2^8)と表示され、整数 0 から 2^{8-1} までの要素がある。RAID6 で 2 台のディスクが故障した時に、4 つのケースが考えられる。P と Q ブロックが故障、P とデータブロックが故障、Q とデータブロックが故障、2 つのデータブロックが故障である。

RAID は実装によりソフトウェア(SW)方式とハードウェア(HW)方式に分類される。現在、主として用いられているのは HW RAID である。CPU の負荷のため SW RAID は HW RAID より性能が低いと考えられてきた。しかし、最近の CPU は HW RAID コントローラより高い性能を持つ。ファイルサーバのように CPU が RAID 演算に専念できるマシンでは SW RAID も有効である。また、ネットワークを越えて RAID を構成する場合、SW RAID は唯一の解である。

2.2 RAID の信頼性

RAID の信頼性ではシステムが故障するまでの平均時間（MTTF）の計算がよく用いられる。文献[1]によると、RAID1 の MTTF は以下の式で表せる。

$$\frac{MTTF_{disk}^2}{N \times MTTR_{disk}}$$

RAID1 では N=2 である。RAID1 を一般化すると以下のようになる。

$$\frac{MTTF_{disk}^{N}}{N! \times MTTR_{disk}^{N-1}}$$

RAID4、RAID5 の MTTF は以下のようになる。

$$\frac{MTTF_{disk}^2}{N \times (G-1) \times MTTR_{disk}}$$

$MTTF_{disk}$ は一台ディスクの MTTF、N はディスクアレイのディスク数、G はパリティディスク含まないグループ内のデータディスク数、$MTTR_{disk}$ は一台ディスクの MTTR（故障したから復旧にかかる時間の平均）である。RAID6 の MTTF は以下のようになる。

$$\frac{MTTF_{disk}^3}{N \times (G-1) \times (G-2) \times MTTR_{disk}^2}$$

シングル RAID を多重にすること（階層型 RAID）で性能や信頼性を上げることができる。例えば RAID0 を上位層にすると、性能が上がる。そして、下位層に RAID1、RAID5、または RAID6 にすると信頼性が高められる。本研究では信頼性を重視するために RAID55 や RAID66 が用いられる。RAID55 の MTTF は以下の式で表せる。

$$\frac{MTTF_{RAID5\_disk}^2}{N \times (G-1) \times MTTR_{RAID5\_disk}}$$

RAID66 の MTTF は以下の式で表せる。

$$\frac{MTTF^3_{RAID6\_disk}}{N\times(G-1)\times(G-2)\times MTTR^2_{RAID6\_disk}}$$

$MTTF_{RAID5\_disk}$ は RAID5 ディスクの MTTF で、$MTTF_{RAID6\_disk}$ は RAID6 ディスクの MTTF である。

信頼性には MTTR の影響も大きい。MTTR が大きくなると MTTF が小さくなる。ゆえに MTTF を大きくするため、つまりシステムが故障するまでの平均時間を大きくするのに MTTR を小さくする必要がある。

3 VLSD

VLSD[3]は大規模ストレージを構築するための 100% pure Java ツールキットである。ここでは、VLSD[3]およびそれを用いたストレージの概要について述べる。

3.1 VLSD のクラス

ここで、VLSD ツールキットのクラスについて説明する。

- NBDServer

NBD サーバのクラス。NBD サーバはクライアントで動作し、空き容量を仮想ディスクファイルとしてストレージシステムに提供する。クライアントの OS は Windows または Linux である。NBD サーバは Java で実装されているためプラットフォームに依存しない。Linux と Windows の両方で動作する。また、クライアントには複数のディスクが接続されていたり、FAT32 が使われていたりすることがある。FAT32 ではそのサイズが 2GB 以上のファイルを作成できない。これらのような場合、120GB の仮想ディスクを単一ファイルとして作成することはできないので、後述の RAID0 または JBOD と組み合わせることで複数のファイルを束ねて仮想ディスクを実現することができる。

- DiskServer

ディスクサーバのインターフェースである。

- DiskServerImpl

ディスクサーバのインプリメンテーションであって、RMI による遠隔ディスクを提供する。

- Disk

仮想ディスクが備えるべきインターフェースである。

- AbstractDisk

抽象的な仮想ディスクのクラスであって、下位クラスで使う定数やメソッドを定める。

- DiskArray

複数ディスクからなるディスク・ラッパーの基底クラスで、簡単な RAID1 を実装している。

- RAID

RAID の基底クラスで、簡単な RAID1 の実装を DiskArray から引き継いでいる。

- SingleDisk

単一ディスクからなるディスク・ラッパーの基底クラスである。

- PagedDisk

ページ単位でアクセスするディスクで、任意のディスクのラッパーとなる。ラップされたディスクはページ単位でしか read/write されない。ページ単位の端数は無視される。

- VariableDisk

可変容量ディスクであって、事前に資源を割り当てず、必要に応じて動的に資源を確保する。作成時に指定したサイズを超えて資源を使用することはないが、論理的なディスクサイズより大きくなることがある。

- RemoteDisk

ディスクサーバをアクセスする遠隔ディスクである。

- RAID0

RAID0 仮想ディスクのクラス。RAID0 は容量を増やすために使われる。後述の JBOD とはストライピングを行う点が異なる。若干性能はよいが、容量は最小サイズのディスクに合わされる。例えば、100GB, 120GB, 160GB を連結しても 100GB×3 にしかならない。純粋に容量増を目的とする場合 JBOD を用いたほうがよい。逆に、RAID0 は性能向上が期待できる場合がある。一部のファイルシステムでは i-node を管理するスーパーブロックが集中して配置される。このようなファイルシステムでは、規模が大きくなると特定のディスクにアクセスが集中する。このような場合、ストライピングは負荷を分散する効果がある。RAID0 はバイト単位でストライピングする。

- RAID5

RAID5 仮想ディスクのクラス。パリティを各ディスクに分散して格納する。パリティを格納するディスクはブロックごとに異なる。

- RAID6

RAID6 の実装で、GF テーブルの生成、ブロック単位ストライピングと分散パリティが行われる。

- JBOD

JBOD 仮想ディスクのクラス。RAID0 と同様に冗長性がなく、容量増のために用いられる。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 380GB になる。RAID0 には負荷分散の効果があると述べたが、JBOD には負荷を集中させる効果がある。ある程度の規模まではキャッシュが有効に働くため、RAID0 より性能がよくなる可能性がある。

3.2 VLSD を用いた大規模ストレージの試作

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。

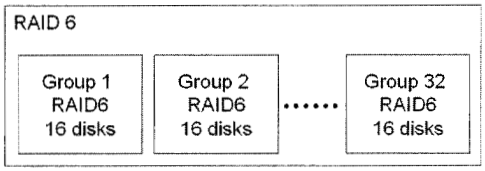

図 1　RAID66 の構成図

本研究ではディスクレベルで空き容量を連結して 1 つの 70TB ストレージを試作した。システムはディスクレベルなので、部分ディスクサイズを越えるファイルの保存も可能である。本研究は 512 台のディスク（1 ディスク=170GB）を 32 グループにして RAID66 を構築する(図 1 に示す)。

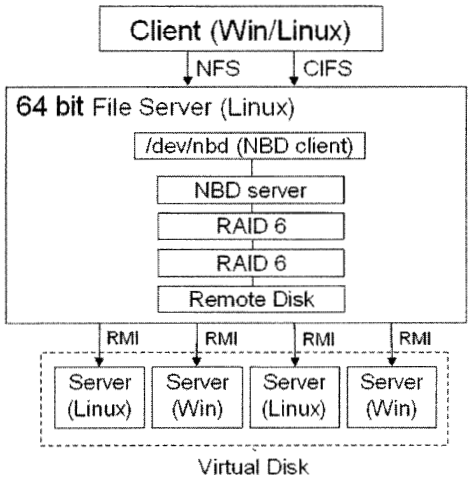

図 2　システム構成図

図 2 に示すように本システムは 64 ビットファイルサーバとディスクサーバがある。ディスクサーバの Linux や Windows などの OS からなる仮想ディスクはディスクの読み込みや書き込みを Java の RMI で機能を提供する。ファイルサーバの方は用意されたディスクに接続して RAID66 を構築する。RAID66 とは、2 階層の RAID6 である。NBD Server はその RAID66 を利用して NBD Client からのアクセスを待つ。そして、NBD Client の起動をした後、XFS でフォーマットする。Windows のクライアントは Samba を介してそのファイルサーバをアクセスする（Linux の場合は NFS）。

NBD プロトコルはセキュリティに欠けるため、ネットワーク上で運用するのは危険である。しかし、我々の方式では 1 台のサーバ内のプロセス間通信として NBD を用いているため安全に運用できる。実際の C/S 間通信はセキュリティを考慮した RMI に基づくプロトコルで実現される。

4　修復

冗長な RAID では、故障が発生しても動作し続けることができる。RAID5 は 1 台のディスクに耐える。RAID6 は 2 台のディスク故障に耐える。RAID から冗長性が失われた状態を縮退モードという。しかし、故障時の性能は正常時より低下する可能性がある。我々は文献[4]にて故障時の VLSD の性能を評価した。その結果、速度低下はあるものの使用可能なレベルであることが確認できた。

2.1 節で述べたように RAID の MTTF はディスクの MTTF と MTTR に依存する。我々の大規模ストレージは階層型 RAID に基づくため、同様にその MTTF はディスクの MTTF と MTTR に依存する。

MTTR は、故障してから再び正常に使用できるまでの時間として定義される。縮退モードでは、RAID は停止しないが、もう 1 台故障すると停止する。RAID を稼動し続けるためには、もう 1 台故障する前に故障したディスクを修復する必要がある。

ディスクを修復する方法は故障の原因に依存する。故障の原因がディスクの故障ではない場合、例えば PC のシャットダウンの場合、PC を再起動すれば容易にディスクを回復できる。一般にディスクの修復に要する時間より PC が再起動する時間の方が短い。特に大容量ディスクでは、この特徴が顕著である。ただし、故障していた期間の差分を起動後に適用する必要がある。

次に、故障の原因がディスクそのものの故障である場合、直ちにホットスペアを割り当て、修復を開始する。ホットスペアには大域ホットスペアと局所ホットスペアがある。大域ホットスペアはシステム全体で空き容量を管理する。例えば、500 台の PC からなる試作では、実際に 484(=22*22)台の PC しか使わない。よって、16 の大域ホットスペアを確保できる。局所ホットスペアは同一 PC あるいは同一 RAID で空き容量を管理する。大域ホットスペアは容量が大きいが、ネットワーク遅延のために遅い。局所ホットスペアは、容量は小さいが速い。本研究では、大域ホットスペアを仮定する。

修復は以下のように行われる。故障したディスクをスペアディスクに入れ換えた後、RAID内で故障したディスクの内容だけを読み取り、単純に書き戻す。これにより、無故障ディスクからデータを復元し、そのデータを故障したディスクに書き込むことができる。VLSD ではRAID クラスが故障ディスクに対応するブロックを順次アクセスすることで復元する。

データ復元の時、修復中のディスクに読み書きがある。ここで３つの場合に分けて考える。修復済みブロックへの読み書き、修復中ブロックへの読み書き、未修復ブロックへの読み書きである。

スペアディスクの修復したブロックに読み書きがある場合、読み書きを許可することができる。それは修復したブロックにデータの復元が完了した領域だからである。修復中ブロックに読み書きがある場合、読み書きを修復するまで遅延させる必要がある。未修復ブロックに読み書きがある場合、読み書きを行わないで故障したブロックとみなす。データはパリティから復元できる。

データの復元方法は RAID レベルによって異なる。RAID1 でのディスク修復はただのディスクコピーの作業が行われている。RAID4 とRAID5 の修復では故障したディスク以外のディスクのストライプブロックデータを読み込んで、排他的論理和を計算すればデータの復元ができる。RAID6 の修復では 1 台ディスク故障の場合と２台ディスク故障の場合に分けて考える。1 台ディスク故障の場合、P ブロックまたはデータブロックの故障時は RAID5 と同様な作業が行われるが、Q ブロックの故障時はすべてのデータブロックを読み込んで、Q ブロックを生成する。２台ディスク故障時の修復は 2 章に述べたように４つのケースに分ける。P ブロックと Q ブロックが故障時は P ブロックと Q ブロックを生成すればよい。P ブロックとデータブロックが故障時は、無故障データブロックと Q ブロックから故障したデータブロックを復元してから P ブロックを復元する。Q ブロックとデータブロックが故障時は、故障していないデータブロックと P ブロックから故障したデータブロックを復元してから Q ブロックを復元する。２つのデータブロックが故障時は、P ブロックと無故障データブロックから排他的論理和によってできた式と、Q ブロックと無故障データブロックから GF 掛け算によってできた式で連立方程式を解くことによって、２つのデータブロックを復元することができる。

このアルゴリズムは容易に並列化できる。ブロックは競合しないため複数のスレッドで異なるブロックを復元すれば、修復処理を短縮できる。短縮の度合いは実行環境及び実装に依存する。

5　評価

ここでは、修復時間を測定し、RAID 全体の信頼性を評価する。

5.1 実験環境

本評価は、AMD Athlon(tm) 64 X2 Dual Core 3800+、メモリ 2GB、Windows XP Professional x64 で行われた。

5.2 HDD の MTTF

目標とする大規模ストレージでは、対費用効果の高い 320GB の HDD を使用する。表 1 に代表的な製品の MTTF を示す。その平均は 85 万時間である。よって、平均的な HDD を 500 台使用した場合、85 万/500=1700[h]≒70[日]に 1 台の割合で故障する可能性がある。

表 1 HDD 製品の MTTF

メーカー	製品	MTTF[10^3h]
Seagate	ST3320620AS	700
IBM	HDT725032VLA360	1000
平均		850

5.3 修復時間の評価

ここでは、RAID5 と RAID6 の修復時間を評価した結果について述べる。

4 台の仮想ディスクで構築された RAID5 とRAID6 の修復時間を図 3 に示す。

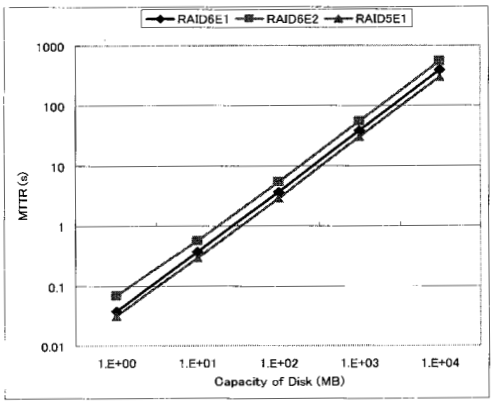

図 3 RAID5 と RAID6 の修復時間

RAID5E1、RAID6E1 はディスクが 1 台故障時の RAID5、RAID6 で、RAID6E2 はディスクが 2 台故障時の RAID6 である。グラフに示すようにRAID6E1、RAID6E2 と RAID5E1 のディスク容量は MTTR に比例する。修復時間は RAID6E2 > RAID6E1 > RAID5E1 の順になる。

5.4 並列修復による MTTR

修復作業は並列処理できる。もし N 並列で修復する場合、ブロック番号を mod N で分類か、あるいは 1/N で分類すればよい。

N が必要以上に大きくても効果は期待できない。N=1、2、4 の場合、表 2 のような結果が得られた。

この実験ではデュアルコアの場合、スレッド数が 2 のときが最適である。これは CPU より I/O がボトルネックとなっているためと考えられる。

表 2 並列修復の評価

N	処理時間[h]
1	8.47
2	7.12
4	7.97

5.5 信頼性の評価

測定した MTTF、MTTR を用いて階層 RAID 全体の MTTF を計算する。RAID1, 5, 6 の組み合わせからなる階層 RAID を考える。

表 3 2 階層 RAID の MTTF と容量効率

RAID level	MTTF[h]	容量効率[%]
RAID11	4.25×10^{2463}	0.2
RAID15	1.7×10^{219}	4.3
RAID16	2.2×10^{328}	4.1
RAID51	4.19×10^{182}	4.3
RAID55	5×10^{15}	91.1
RAID56	1.1×10^{23}	86.8
RAID61	4.78×10^{272}	4.1
RAID65	7.8×10^{23}	86.8
RAID66	2.2×10^{35}	82.6

ここで、ディスク総数を 484、下位層の台数を 22、上位層の台数を 22 にしたときの結果を表 3 に示す。なお、RAID3, 4 は RAID5 と等しいため、省略する。また、RAID1 は N 台に同じ内容をコピーするものとした。

RAID1 を用いると容量効率は極端に低下するので RAID1 を使うべきではない。容量が大きくて MTTF が高いのは RAID66 が最もよいが、RAID6 は普及したばかりで比較的高価である。

6 まとめ

本論文では、VLSD の修復機能と性能について述べた。VLSD は PC の空き容量を用いて大容量ストレージを構築するツールキットである。今回の評価で、負荷のない条件下では単一の 170GB ディスクを 7.12 時間で修復可能であることがわかった。これは 500 台の PC からなる RAID55 の 75TB 試作システムにおける MTBF が最悪 5×10^{15} 時間であることから、十分実用可能であるといえる。

今回の評価は、そのサイズが物理容量である FileDisk を用いて行われた。FileDisk のサイズは実行時に変化しない。つまり最初から大きな容量を必要とする。しかし、実際の運用では、必要に応じてサイズを変更したほうが良い。このような可変容量ディスクは VariableDisk で実現される。しかし、VariableDisk は未使用領域を含むためブロックが連続していない。今後は、不連続なブロックを繰り返すしくみを導入する必要がある。

また、ディスク交換における割り当てポリシーの評価を行う必要がある。例えば、実行時の性能を優先するなら局所ホットスペアがよい。あるいは、交換時間の短縮を優先するなら大局ホットスペアがよい。それぞれの性能を評価する必要がある。

今回、1 台のマシンで評価した。しかし、実際のシステムは複数マシンで構成される。複数のマシンでネットワーク経由で使用した場合、性能はどう変化するのか調査する必要がある。

謝辞

本研究は科研費基盤 (C)「PC グリッドによる高信頼・高効率な分散仮想ストレージの研究 (19500066)」により援助されています。

参考文献

[1] David A. Patterson, Garth Gibson, and Randy H. Katz: "A Case for Redundant Arrays of Inexpensive Disks (RAID)", ACM SIGMOD, 1988

[2] 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

[3] Chai Erianto, 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)

[4] Chai Erianto, Minoru Uehara, Hideki Mori: "Performance Evaluation at Failure in a Large-Scale Virtual Disk", DPSWS2007 (2007.10)

[5] Intelligent RAID 6 Theory Overview and Implementation, http://download.intel.com/design/storage/papers/30812202.pdf