情報処理学会論文誌 Vol. 52 No. 2 434–445 (Feb. 2011)

3耐故障直交RAIDの提案と大規模仮想ディスクにおける構成法

上原 稔[†1] 村上 真[†1] 山際 基[†1]

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．しかし，大容量ストレージの信頼性を高めるには3以上の耐故障性を持つRAIDが必要になる．本論文では，直交RAIDに基づく3耐故障RAIDとしてMeshRAIDを提案する．MeshRAIDは3耐ディスク故障に加えて2つまでのコントローラ故障にも強い．また，我々はVLSDを用いてMeshRAIDを実装した．VLSDのクラスライブラリを組み合わせることで比較的容易にMeshRAIDを実装することができた．この方式は複雑なRAIDを実装する方式として有効である．しかし，構造的なオーバヘッドのために性能は従来方式に劣る．それゆえラピッドプロトタイピングに適した構成法であるといえる．

A Proposal of 3 FT Orthogonal RAID and Its Implementation in Virtual Large-scale Disk

Minoru Uehara,[†1] Makoto Murakami[†1] and Motoi Yamagiwa[†1]

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, in order to construct large-scale storage, more than or equal to 3 fault tolerant RAID is important. In this paper, we propose MeshRAID that is 3 fault tolerant orthogonal RAID. MeshRAID is tolerant to at most 3 disk failures and also tolerant to at most 3 controller failures except controllers in 1st or 2nd layer. Furthermore, we implement MeshRAID easily by combining several VLSD classes. This methodology is generally applied to implement complex RAID. However, the performance of new RAID implemented in this method is less than conventional method. Therefore, this method is suited for rapid prototyping.

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．たとえば，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，センサネットの発達により多数のセンサ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きわめて高価であり，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．

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，数十〜数百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．実際には，そこまで大規模でなくても故障は十分同時発生する．

大規模ストレージでは，耐故障性が重要である．しかし，普及しているRAIDの中では2耐故障であるRAID6が最も耐故障性に優れている．3耐故障RAIDはまだ普及していない．最も現実的な3耐故障RAIDはRAID55である．RAID55はRAID5を2階層に組み合わせた階層RAIDである．RAID6を2階層に組み合わせた階層RAIDであるRAID66は，さらに高い耐故障性を持つが，容量効率が低下する．100台からなるRAID55の容量効率は81%であるが，RAID66の容量効率は64%である．一般的に耐故障性（あるいは信頼性）と容量効率はトレードオフの関係にある．ある耐故障で十分な用途に対して過剰な耐故障性を実現することは，容量効率の低下，ひいてはコスト増を招く．RAID6のような2FT RAIDの$MTTF_{2FT}$は以下の式で与えられる．

†1 東洋大学総合情報学部
Faculty of Information Sciences and Arts, Toyo University

$$MTTF_{2FT} = \frac{MTTF^3}{N(N-1)(N-2)MTTR^2}$$

ここで，N はディスク台数，$MTTF$，$MTTR$ は要素ディスクの $MTTF$，$MTTR$ である．

同様に，3FT RAID の $MTTF$ は以下の式で与えられる（文献 8)，10) 参照)．

$$MTTF_{3FT} = \frac{MTTF^4}{N(N-1)(N-2)(N-3)MTTR^3}$$

要素ディスクの容量はすべて等しいものとし，仮に 160 GB とする．$MTTR$ は容量に比例するとし，5 MB/s で転送可能な I/F で接続されていると仮定すると，$MTTR \fallingdotseq 9$ [h] である．ここで，$MTTF$ を 50,000 [h] とすると，100 台からなる RAID の場合，$MTTF_{2FT} = 1.6 \times 10^6$ [h] である．これは 200 年弱であるから十分であるとすると，同程度の $MTTF$ を実現する 3FT RAID は約 300 台でも構成可能である．本論文では，3 耐故障でも足りる規模のストレージを前提とする．前述の例では，100 台を超えたとき $MTTF$ が劣化することを防ぐ効果がある．

また，階層型 RAID はディスク故障には強いが，コントローラ故障には弱い．コントローラが故障してしまうと複数のディスクが一度に故障してしまう．RAID55 でも耐コントローラ故障は 1 にすぎず，RAID66 でさえ 2 である．一般的にコントローラの $MTTF$ はディスクの $MTTF$ より長いが，絶対故障しないわけではない．我々の RAID は VLSD によるソフトウェア（SW）RAID であるため，ハードウェア（HW）RAID とは故障原因が異なる．RAID 階層の根にあたるサーバは単一故障点であるため，故障しないことを前提とするが，それゆえサーバと一体化したコントローラも故障しないと仮定されてきた．しかし，実際の運用ではサブコントローラは遠隔サーバに設置することもある．そのような場合，故障の確率は比較的高い．

本論文では，直交 RAID に基づく 3 耐故障 RAID として MeshRAID を提案する．我々は，VLSD で MeshRAID を実装した．その際，VLSD の既存クラス（コンポーネント）を活用することで短期間に実装できた．これは RAID 開発のラピッドプロトタイピングの実例として，あるいは VLSD の有効性を示す事例としても有意義である．それゆえ，本論文では実装方法についても焦点をあてる．

本文の構成は以下のとおりである．2 章で関連研究として既存の 3 耐故障 RAID について述べる．3 章では VLSD について述べる．4 章では MeshRAID の概念を述べ，5 章で VLSD による実装法について述べる．6 章でその評価を行い，最後に結論を述べる．

2. 関 連 研 究

2.1　3 耐故障 RAID

3 耐故障を実現する RAID には，RAID TP，階層型 RAID などがある．ここでは，関連研究として既存の 3 耐故障 RAID について述べる．

RAID TP は Accusys 社の開発した 3 耐故障 RAID である．3 つのパリティディスクで 3 つの故障を修復できる．RAID TP は RAID6 P+Q の延長にある．RAID5 までは排他的論理和（XOR）に基づく単一パリティのみで単一故障を修復するが，RAID6 は 2 つのパリティで 2 つの故障を修復する．原理的には，直交するパリティ系を選択できれば，系の数だけ故障を修復することができる．しかし，直交する系を見つける系統だった方法はない．RAID6 には対角パリティ方式と P+Q パリティ方式がある．垂直パリティは水平パリティに直交するが，垂直パリティが耐故障性を向上させないことは明らかである．対角パリティは，垂直パリティをずらし，対角線上にパリティグループを構成することで巧みに水平パリティと直交なパリティ系を実現する．対角パリティ方式では，いずれのパリティも XOR 演算のみで計算できる．一方，P+Q パリティ方式ではパリティ Q をガロア体で計算する必要があるため，単純な XOR 演算のみでは計算できない．同様に RAID TP も特殊な演算を必要とする．そのため Accusys は RAID TP をハードウェア RAID として提供している．ハードウェア RAID は高性能であるが，柔軟性に欠けるため，他の手法と組み合わせることが難しい．以上のことから本論文では RAID TP を直接の比較対象としない．

階層型 RAID も 3 耐以上の耐故障性を実現することができる．階層型 RAID はそれ自体が新しいクラスであるというより既存クラスの RAID を組み合わせる手法である．階層型 RAID は既存 RAID クラスを階層的に組み合わせる．たとえば，RAID0 と RAID1 を組み合わせると RAID01 あるいは RAID10 となる．RAID01 は，上位層に RAID1，下位層に RAID0 で構成される．RAID10 は逆となる．階層型 RAID では，通常の RAID より信頼性を向上させることが可能となる．たとえば，通常の RAID5 は 1 耐故障だが，RAD55（2 階層の RAID5）は 3 耐故障である．RAID55 の信頼性は RAID6 を上回る．16 台のディスクからなる RAID55 の構成を**図 1** に示す．RAID55 が 3 耐故障である証明は文献 2) に詳しい．ここでは簡易的な証明を示す．たとえば，図 1 の RAID55 において下層 RAID5 で 1〜3 個のディスクが故障する場合に分類する．下層 RAID5 で 1 個のディスクが故障しても RAID5 が耐えるため RAID55 も耐えうる．下層 RAID5 で 2 個のディスクが故障すると，その RAID5 は故障するが，残り 3 つの RAID5 は正常に動作するため RAID55 自体は正

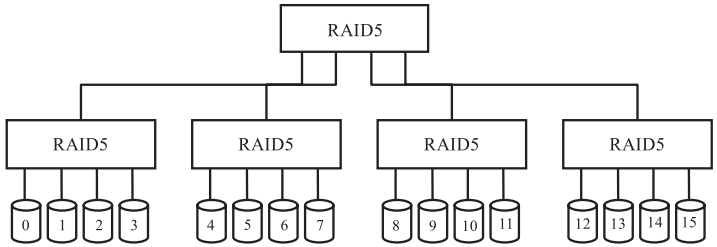

図 1　RAID55
Fig. 1　RAID55.

常に動作する．下層 RAID5 で 3 個のディスクが故障しても同様である．よって，RAID55 は 3 耐故障である．

2.2　直交 RAID

直交 RAID という語はしばしば 2 つの異なる文脈で使われる．1 つはパリティグループを配線グループと直交して構成する際に使われ，もう 1 つは既存配線グループに対して直交する新たな配線グループを構成する際に使われる．本論文における直交 RAID は後者の意味である．**図 2** に直交 RAID の例を示す．図中の四角（$H_{0\text{-}2}$，$V_{0\text{-}2}$）はコントローラを示し，SCSI などのケーブルでディスクとコントローラが配線されていることを示す．同じコントローラと結合されたディスクは同一配線グループに属する．同一配線グループ内のディスクはコントローラの故障あるいは配線の切断により同時に故障と見なされる．しかし，実際にはディスク自体が故障するわけではないので別の経路が存在すればデータにアクセスすることができる．そこで，既存配線グループと直交する形で新たな配線グループを構成する方式が直交 RAID である．

直交 RAID の概念はけっして新しくない．事実，Chen らの文献 1) においても紹介されている．ただし，配線故障により一度に大量のディスクが故障する場合に対する方法として紹介されており，積極的に 3 耐故障を目指したわけではない．配線故障はコントローラ故障と等価であり，一度に複数の配線あるいはコントローラに関連付けられた複数のディスクが故障状態と見なされる．しかし，実際にはディスク自体は故障していない．直交 RAID では，配線故障が生じても複数の経路があるためディスクにアクセスできる．配線故障において，直交 RAID は，木構造である階層 RAID より信頼性が高い．

直交 RAID が実際に製品に応用されることはあっても配線故障以外に適用された事例は筆者の知るところない．配線故障に適用する場合，単純にう回路が与えられればよい．たとえば，図 2 で H_0 が断線してもディスク 0 の内容は V_0 からアクセスできる．しかし，直交 RAID をディスク故障に応用し，ましてや 3 耐故障に対応するには，う回路を提供するだけでは十分ではない．ディスク 0 が故障したとき，その内容は H_0 グループではディスク 1，2 から，V_0 グループではディスク 3，6 から修復される．そのためには，両グループに正しくパリティを保存しておく必要がある．しかし，単なる直交 RAID ではそのための仕組みがない．本論文では，直交 RAID をディスク故障に適用し，しかも 3 耐故障を実現する点で従来方式と異なる．

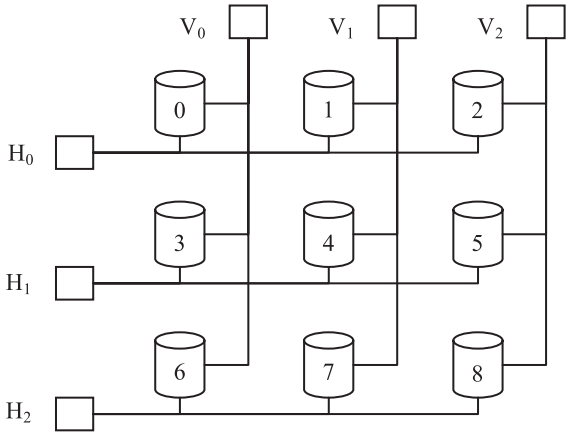

図 2　直交 RAID
Fig. 2　Orthogonal RAID.

3.　VLSD

本章では，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 は関連研究に位置づけられるが，広く一般に認知されている技術ではないため，1 章を割いて解説する．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．

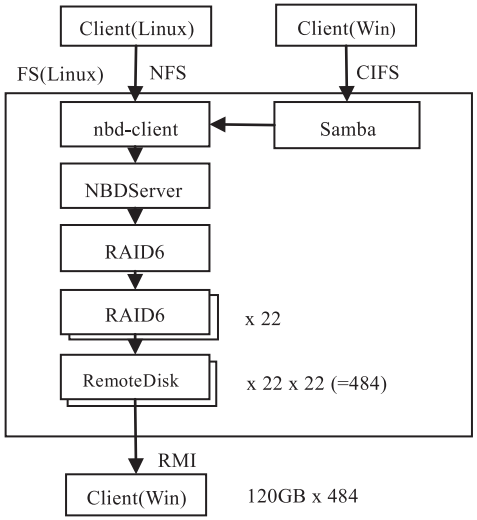

図 3　VLSD のシステム概要
Fig. 3　The system overview of VLSD.

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 4 GB の制約がある FAT32 で動作させた場合，120 GB/2 GB=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 3 に VLSD を用いて分散ストレージを構成した例を示す．クライアントは約 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 RAID66 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．8 KiB を単位とする 1 K 分木で管理する．葉ノードには 8 KiB のデータが格納される．中間ノードには 1,024 個の 64 b（8 B）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 8 EiB-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（たとえば，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．この対は認証および認可を司る．データ自身の盗聴を防ぐには CipherDisk を用いて暗号化する必要がある．また，改ざんの検出は ChecksumDisk あるいは DigestDisk を用いれば可能である．これらはそれぞれチェックサムとメッセージダイジェストを計算してデータの正当性を検証する．

WebDisk

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．たとえば，100 GB，120 GB，160 GB 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

耐故障性評価を行うためのクラス．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．たとえば，RAID6 を 2 段階で組み合わせると RAID66 を構築できる．

4. MeshRAID の提案

我々は直交 RAID で 3 耐ディスク故障を実現する．本論文では，3 耐故障を実現する直交 RAID を MeshRAID と呼ぶ．MeshRAID は 2 次元直交 RAID である．**図 4**，**図 5** は図 3 の直交 RAID と同数のディスクから構成される MeshRAID である．

図 4 はそのパリティグループを示す．直交 RAID の配線グループは配線だけでなくコントローラを共有するグループであった．MeshRAID のパリティグループも RAID コントローラを共有するグループであり，パリティグループに対応する．MeshRAID のパリティグループは直交 RAID の配線グループと同様に縦 3（V_i），横 3（H_i）の計 6 つのグループからなる．これら 6 つのグループは理論上任意の RAID クラスでありうる．しかし，論理的に矛盾する構成はとりえないので，H_i は等しく，V_i も等しい．また，H_i と V_i も等しいことが自然である．本論文では，3 耐故障に十分な XOR パリティに基づく RAID クラスとして最も単純な RAID4 を想定する．H_i，V_i のいずれのグループにおいても最大の添え字のディスクをパリティディスクと仮定する．すなわち H_0 ではディスク 2（D_2），V_0 ではディスク 6（D_6）である．

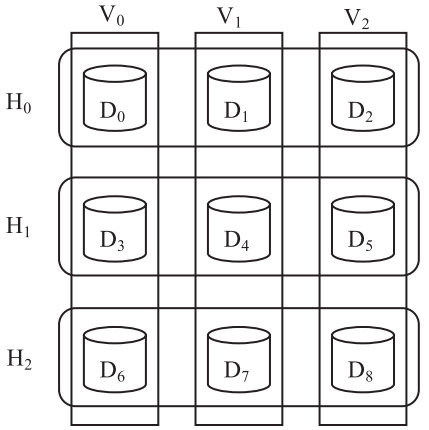

図 4　MeshRAID のパリティグループ
Fig. 4　The parity groups of MeshRAID.

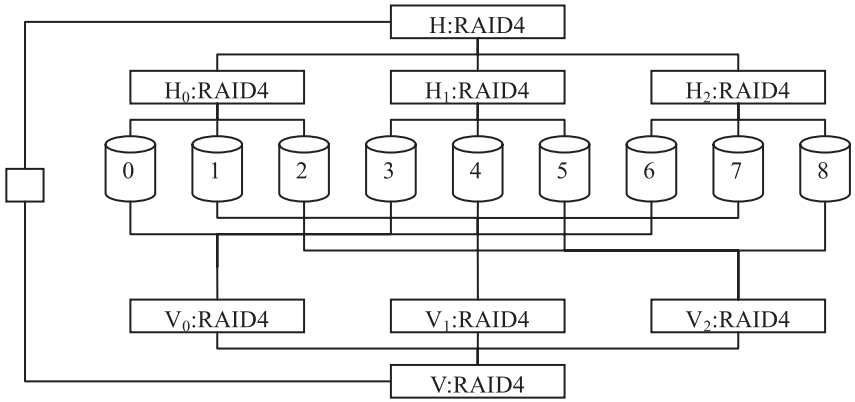

図 5　MeshRAID の概念的構成
Fig. 5　The conceptual structure of MeshRAID.

図 5 に図 4 の MeshRAID の概念的構成を表すブロック図を示す．ここで，概念的構成とは MeshRAID の原理を示すための構成であり，実際の実装を示すものではない．図 5 では縦と横のパリティグループが階層 RAID を構成する．MeshRAID は縦のグループ V_i か

らなる階層 RAID と横のグループ H_i からなる階層 RAID により二重に構成される．ここでは自然な階層構造として 2 層を想定したが，理論上は矛盾がなければ多階層にも構成できる．この二重構造において H_i，V_i はそれぞれ下位層にあたる．H_i，V_i の上位層をそれぞれ H，V とすると，これらは必ずしも下位層と同 RAID クラスである必要はない．しかし，本論文では 3 耐故障であるために必要十分な RAID4 を想定する．図から明らかなように，MeshRAID は階層 RAID に近いが，その二重構造と要素ディスクを共有する構成からユニークである．H と V を統合するコントローラは MeshRAID 固有の制御を司る．

図 5 の構成に基づき MeshRAID の 3 耐故障性を簡単に示す．RAID55 が 3 耐故障であることはすでに示したが，同様に RAID44 が 3 耐故障であることも明らかである．MeshRAID は RAID44 が縦横 2 系統あり，いずれの系統でも 3 耐故障である．ゆえに二重の RAID44 で構成される MeshRAID は 3 耐故障である．次に，二重の RAID44 が矛盾なく構成できるか否かを論じる．RAID4 ではパリティディスクがストライプに依存せず固定であるため，あるストライプについて示せば，すべてのストライプについて示したことになる．ここで，ディスク i のストライプデータを d_i とすると，$d_2 = d_0 \wedge d_1$，$d_5 = d_3 \wedge d_4$，$d_6 = d_0 \wedge d_3$，$d_7 = d_1 \wedge d_4$ である．H_2 から $d_8 = d_6 \wedge d_7 = d_0 \wedge d_1 \wedge d_3 \wedge d_4$ であり，同時に V_2 から $d_8 = d_2 \wedge d_5 = d_0 \wedge d_1 \wedge d_3 \wedge d_4$ である．ここで演算子 $\wedge$ は排他的論理和を示す．H_2 と V_2 で d_8 が一致することからパリティは無矛盾である．よって，RAID44 の MeshRAID は構成可能であり，3 耐故障となりうる．また，詳細は述べないが，この結果から，同一ストライプにおいてつねにパリティが縦横対称である RAID55 は MeshRAID を構成可能である．

次に，MeshRAID の処理について説明する．MeshRAID の主な処理は，読み取り，書き込み，修復である．読み取りは 3 以内の故障を含む MeshRAID から正しいデータを読み取る処理である．書き込みは，同 MeshRAID にデータを正しく書き込む処理である．修復は，故障ディスクを物理的に交換したのち，交換したディスクに該当するデータを復元する処理である．なお，本章で述べる処理は実装と異なるため，概念的かつ原理的な処理である．実装に基づく処理は次章で詳細に解説する．

はじめに読み取り処理について述べる．H と V の各階層 RAID に対して並行に読み取りを依頼し，早く正常に返された値を読み取り値とする．H（V）は下位 RAID H_i（V_i）に対して読み取りを依頼する．H（V）の読み取りでは，RAID4 の原理から H_i（V_i）の 1 つが故障していても，その値を他の構成要素から復元することができる．H_i（V_i）の読み取りでは，1 台以下の故障は RAID4 の原理から正常な値を返すことができる．2 台以上が故障した場合，例外を発生して異常を知らせる．以上，H（V）の動作は RAID44 の原理から 3 つまでの故障に対して正しく動作することが保障できる．この読み取りアルゴリズムによって 3 つまでの故障は隠ぺいされる．

次に，書き込み処理について述べる．ある値 r が保存された場所 x に値 w を書き込むとする．その場所に対応するパリティを p とすると，$x \leftarrow w$，$p \leftarrow p \wedge r \wedge w$ となる．この手続きは階層 RAID の各層で行われる．以下，図 5 を例にディスク i の内容を d_i とし，$d_0 = A$，$d_1 = B$，$d_3 = C$，$d_4 = D$ とすると，パリティの整合性がとれている状態では以下のようになる．

$d_0 = A$，$d_1 = B$，$d_2 = A \wedge B$
$d_3 = C$，$d_4 = D$，$d_5 = C \wedge D$
$d_6 = A \wedge C$，$d_7 = B \wedge D$，$d_8 = A \wedge B \wedge C \wedge D$

ここで，d_0 を A' に更新すると，$d_2 = A' \wedge B$，$d_6 = A' \wedge C$，$d_8 = A' \wedge B \wedge C \wedge D$ とならなければならない．下位の H_0 で d_0 を更新すると d_2 は正しく更新されるが，d_6，d_8 は更新されない．しかし，H_0 の d_0，d_1，d_2 に H_2 の d_6，d_7，d_8 がそれぞれストライプブロックとして対応するなら，上位の H で d_0 を更新すれば，d_6 も更新される．そして，H_2 で d_6 が更新されると，d_8 も更新される．よって，MeshRAID は H か V のいずれか正常に書き込める経路を用いて書き込めばよい．試しに書き込み，例外が発生しなければ正しく書き込めたと見なせる．この議論は MeshRAID の構成が縦横等しい正方であれば任意のサイズで成り立つ．

そこで，次に非正方構成の場合について検討する．たとえば，4 行 3 列の以下のような構成を考える．ここで，H_i，V_i はパリティグループを構成するディスクの集合とする．

$H_0 = \{d_0, d_1, d_2\}$，$V_0 = \{d_0, d_3, d_6, d_9\}$
$H_1 = \{d_3, d_4, d_5\}$，$V_1 = \{d_1, d_4, d_7, d_{10}\}$
$H_2 = \{d_6, d_7, d_8\}$，$V_2 = \{d_2, d_5, d_8, d_{11}\}$
$H_3 = \{d_9, d_{10}, d_{11}\}$

ディスク i の内容を d_i とし，$d_0 = A$，$d_1 = B$，$d_3 = C$，$d_4 = D$，$d_6 = E$，$d_7 = F$ とすると，パリティの整合性がとれている状態では以下のようになる．

$d_0 = A$，$d_1 = B$，$d_2 = A \wedge B$
$d_3 = C$，$d_4 = D$，$d_5 = C \wedge D$
$d_6 = E$，$d_7 = F$，$d_8 = E \wedge F$
$d_9 = A \wedge C \wedge E$，$d_{10} = B \wedge D \wedge F$，$d_{11} = A \wedge B \wedge C \wedge D \wedge E \wedge F$

この場合も前述の議論と同様に正しく更新されることが保障される．この一般化は明らか

であるため省略するが，任意の非正方構成でも MeshRAID の一貫性は維持される．

最後に，修復処理について述べる．まず，故障ディスクを特定する．読み取り，書き込み処理において異常を検出したディスクを故障とマークする．次に，マークした故障ディスクに対応する代替ディスクを用意する．用意した代替ディスクに対して，対応する故障ディスクのデータを MeshRAID から読み取り，コピーする．コピーの間，新たな書き込みは禁止する．なお，この制約は，書き込みログを記録し，コピーの間に書き込まれたデータのみコピーする処理を書き込みログがなくなるまで繰り返すことで，緩めることができる．コピーが終了し，データが復元された代替ディスクを故障ディスクと置き換えて修復を完了する．

本章のまとめとして，MeshRAID と既存方式との比較を行う．直交 RAID はディスクを格子状に配置し，冗長配線する．この意味で MeshRAID は直交 RAID である．しかし，多くの直交 RAID の事例では縦横いずれか一方のグループが RAID0 であり，3 耐故障を実現するに至らない．MeshRAID では RAID44 の構造を内包するため 3 耐故障である．学術的には，直交 RAID と階層 RAID を個別に論じた論文はあるものの，両者を組み合わせて解析した事例は少ない．本論文では，直交＋階層 RAID である MeshRAID を提案し，その特性を詳細に分析する．また，続く章で VLSD を用いて MeshRAID を比較的容易に実装する方法についても述べる．

5. MeshRAID のコンポーネント方式による実装

前章では，MeshRAID の原理を述べた．本章では，MshRAID を VLSD で実装する方法について述べる．MeshRAID の動作は比較的複雑であるため，ハードウェア RAID として実装することは（少なくとも十分な動作検証がなされるまでは）難しい．MeshRAID をソフトウェア RAID として構成するには VLSD を使うと比較的容易である．3 章で述べたように VLSD には既存 RAID クラスが網羅されている．これらを組み合わせることにより比較的容易に実装が可能である．VLSD による MeshRAID の実装には 2 種類考えられる．1 つは，図 5 のように構成されるオールインワンの実装である．この実装では，概念に忠実に全体を 1 つのクラスとして構成する．しかし，このような実装はアルゴリズムのコーディングに手間がかかり，デバッグも難しい．

そこで，本論文では，単純な働きのコンポーネントを組み合わせて MeshRAID を構成する方式を提案する．このような実装を本論文ではコンポーネント方式による実装と呼ぶ．VLSD には，多くの仮想ディスクのクラスが含まれており，それらを組み合わせることで MeshRAID のような一見複雑なクラスを実現することができる．

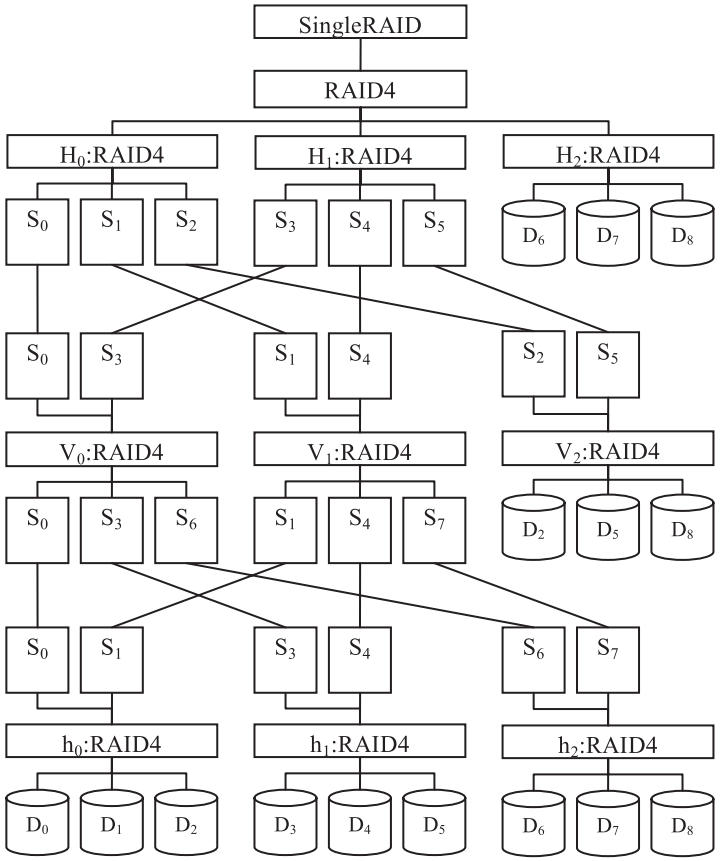

図 6　MeshRAID
Fig. 6　MeshRAID.

コンポーネント方式における MeshRAID の構成を**図 6** に示す．図 6 の四角は VLSD のクラス（名のインスタンス）を表し，円柱はディスクを表す．同名のディスクは同一の実体である．なお，説明上不要な要素は省略してある．ここで，図中で使用された VLSD クラスについて概要を述べる．

SingleRAID

SingleRAID は単一ディスクと RAID として扱うアダプタである．要素ディスクが 1 つしかない RAID0 とも考えられるが，すべての作業を要素ディスクへ委譲するプロキシとして動作する点が単なる RAID0 とは異なる．

RAID4

RAID4 は専用パリティディスクを持つ 1 耐故障 RAID である．耐故障数において RAID5 と等しいが，RAID5 と異なりパリティは専用ディスクに格納される．ここでは，最右のディスクをパリティディスクとする．ソフトウェア RAID においてはパリティを分散させる意味がほとんどない．むしろ，キャッシュを有効に作用させるためにあえて RAID4 を使うこともある．ここでは，パリティディスクを明確にするために RAID4 を用いている．

StripeDisk（図中の S）

StripeDisk は，あるストライプグループ中から指定したブロックを抽出する．これにより RAID の隠ぺいを迂回することができる．たとえば，$\{D_0, D_1, D_2\}$ からなる RAID4 に対して D_1 のみアクセスするには，ストライプグループを 3 とし，そのオフセットを 1 とすればよい．ただし，StripeDisk はあくまで RAID を介して要素ディスクにアクセスするため，パリティディスク D_2 を直接読み取ることはできない．しかし，今回の MeshRAID では，パリティディスクを StripeDisk 経由でアクセスする必要はない．図 4 では，S_i は S_i または D_i へのバイパスを意味する．

MeshRAID からの読み取りは，以下のように行われる．ディスク D_0〜D_8 のいずれも正常のとき，D_0 を読み取るとする．H_0 は S_0 から V_0 へ依頼する．V_0 も S_0 から h_0 に依頼する．最後に h_0 が D_0 を読み取る．他のディスクも同様に正しいデータを返す．

ここで，D_0 が故障したとする．D_0 は H_0 に属する．H_0 は S_0 から V_0 へ依頼し，V_0 は S_0 から h_0 に依頼する．最後に h_0 が D_0 を修復して，正しく読み取る．故障が 1 つしかなければ他のディスクも同様に正しいデータを返す．

次に，D_0，D_1 が故障したとする．D_0，D_1 は H_0 に属する．H_0 は D_0，D_1 を任意の順に読み取る．ここでは，D_0 から読み取るとする．H_0 は S_0 から V_0 に D_0 を依頼する．V_0 は S_0 から h_0 に依頼するが，h_0 は D_0 を修復できない．次に，V_0 は S_3，S_6 からそれぞれ D_3，D_6 を取得し，D_0 を修復する．次に，H_0 は S_1 から V_1 に D_1 を依頼する．V_1 には D_1 しか故障していないので，V_1 は D_1 を修復する．よって，D_0，D_1 ともに修復できる．

次に，D_0，D_1，D_3 が故障したとする．D_0，D_1 は H_0 に属する．H_0 は D_0，D_1 を任意の順に読み取る．ここでは，D_0 から読み取るとする．H_0 は S_0 から V_0 に D_0 依頼する．

表 1　D_0 に x を書き込むシーケンス
Table 1　The sequence of writing x to D_0.

H_0.read d from S_0	V_0.read d from S_0	h_0.read d from D_0
H_0.read p from S_2	V_2.read p from S_2	h_0.read p from D_2
H_0.write x to S_0	V_0.read d from S_0	h_0.read D_0
	V_0.read q from S_6	h_2.read D_6
	V_0.write x to S_0	h_0.read d from D_0
		h_0.read p from D_2
		h_0.write x to D_0
		h_0.write p^d^x to D_2
	V_0.write q^d^x to S_6	h_2.read q from D_6
		h_2.read r from D_8
		h_2.write q^d^x to D_6
		h_2.write r^d^x to D_8
H_0.write p^d^x to S_2	V_2.read p^d^x from D_2	note that d, p, q, and r is initial value of D_0, D_2, D_6, and D_8 respectively.
	V_2.read r^d^x from S_8	
	V_2.write p^d^x to D_2	
	V_2.write r^d^x to D_8	

V_0 は S_0 から h_0 に依頼するが，h_0 は D_0 を修復できない．次に，V_0 は S_3 から h_1 に D_3 を依頼する．h_1 は，D_3 しか故障していないので，D_3 を修復する．次に，V_0 は D_6 を読み取り，D_3，D_6 から D_0 を修復する．次に，H_0 は S_1 から V_1 に D_1 を依頼する．V_1 には D_1 しか故障していないので，V_1 は D_1 を修復する．よって，D_0，D_1，D_3 すべてを修復できる．

MeshRAID への書き込みは，以下のように行われる（**表 1** 参照）．表 1 の各列は MeshRAID の階層を表す．第 $i+1$ 列は第 i 列のサブシーケンスである．ここで，D_0 に d，D_2 に p，D_6 に q，D_8 に r が格納されているとする．また，ディスク D_0〜D_8 のいずれも正常であり，D_0 に x を書き込むとする．H_0 は S_0 から V_0 へ依頼する．V_0 も S_0 から h_0 に依頼する．最後に h_0 が D_0 を書き込む．ここで，パリティについて考える．h_0 が D_0 を書き込んだとき，h_0 は同時に D_2 も変更する．また，V_0 が S_0 へ書き込んだとき，V_0 は同時に S_6 も更新する．S_6 の更新は，h_2 から D_6 へ伝達される．このとき h_2 は D_8 も更新する．H_0 が S_0 に書き込んだとき，H_0 は同時に S_2 も更新する．S_2 は V_2 から D_2 を更新する．そのとき同時に S_8 も更新する．ただし，表 1 が示すとおり，D_2 および S_8 の更新は実際には何

も行わないに等しい無意味な処理である．以上の結果から D_0，D_2，D_6，D_8 はいずれも正しく更新される．この恒常的無更新は，少なくとも逐次処理ではつねに成り立つ．

ここで，D_0 が故障したとする．h_0 は D_0 を修復する．h_0 は正しい D_0 の値 d を返す．h_0 は x を D_0 に格納する代わりに D_2 に $p \wedge d \wedge x$ を保存する．ここで，D_1 の値を e とすると，$p = d \wedge e$，$p \wedge d \wedge x = e1 \wedge x$ であるから，$p \wedge e = x$ となり，正しい値を読み出せることが分かる．

次に，D_0，D_1 が故障したとする．このとき h_0 も故障する．V_0 は d を S_0 から読むことはできない．しかし，V_0 は S_3，S_6 から S_0 を修復できる．また，V_0 は S_0 に x を書き込むことはできない．しかし，V_0 は $q \wedge d \wedge x$ を S_6 に書き込むことはできる．

次に，D_0，D_1，D_3 が故障したとする．このとき h_0 も故障する．しかし，それ以外は故障しない．よって，前の場合と同様に正しく書き込むことができる．

このように書き込みにおいても 3 耐故障が実現できる．他のディスクが故障した場合についての説明を省略するが，おおむね同様である．それらは評価において厳密に検証する．以上のことから，コンポーネント方式も 3 耐故障であるといえる．

表 1 に示したとおり，パリティの一貫性は維持される．ただし，最後の処理は無意味な処理であり，純粋なオーバヘッドとなる．これはアルゴリズムの複雑さを減らしたことに対する代償である．

MeshRAID は，3 耐故障を制御するため一般的な RAID としては複雑であるが，コンポーネント方式の実装は単純である．実際，MeshRAID の主要なプログラム部分はわずか 44 行しかない．オールインワンで開発すればもっと多くの行数を必要とするが，VLSD を用いた本実装では簡潔に表現できた．

6. 評　　価

ここでは MeshRAID を耐故障性，容量効率，性能の各面から評価する．

MeshRAID が 3 耐故障であることは 4 章で述べた．MeshRAID が正しく実装されていれば 3 耐故障であることは当然である．ここでは，VLSD で実装した MeshRAID が 3 耐故障であるかどうかを検証する．VLSD には，耐故障性テストのために FaultDisk クラスが用意されている．耐故障性評価では，計 9 台の要素ディスクを用いて 9 台（3 行 3 列），12 台（4 行 3 列），15 台（5 行 3 列）の MeshRAID を構成し，その各ディスク台数から 3 とるすべての組合せにおいて故障を発生させ，故障が発現するかどうか調べた．結果を**表 2** に示す．#faults は#cases 中の故障を修復できなかった場合の数である．表 2 の結果から

表 2　MeshRAID の耐故障性
Table 2　Fault tolerance of MeshRAID.

Configuration	#cases	#faults
3 x 3	${}_9C_3 = 84$	0
4 x 3	${}_{12}C_3 = 220$	0
5 x 3	${}_{15}C_3 = 455$	0

すべての場合を修復できたことが確認できる．よって，MeshRAID の VLSD 実装は 3 耐故障である．MeshRAID の 3 耐故障性は基本的に VLSD に依存しない．しかし，実際に 3 耐故障を検証することに興味ある読者は VLSD の開発版を sourceforge.jp [13] から入手できる．ただし，メンバ登録が必要である．

末端のディスクレベルの耐故障性の観点からは RAID55 と MeshRAID に有意の差はない．しかし，RAID コントローラ（VLSD の RAID5 クラスに相当）の故障まで考慮すると差が出る．RAID55 には上位と下位の 2 層の RAID コントローラがあり，上位の RAID コントローラ 1 つか，あるいは下位の RAID コントローラが 2 つ故障すると全体が故障してしまう．MeshRAID には 4 層の RAID コントローラが存在し，最上位の 1 つか，あるいは第 2 層（図 4 の H_i 層）の 2 つが故障したとき全体が故障する．つまり，第 3 層以下の RAID コントローラは 2 故障までは隠ぺいされる．これは，実際に RAID コントローラに FaultDisk を用いて故障を注入した結果，確認された．しかし，このままでは RAID55 と変わらない．RAID55 に存在しなかったコントローラの故障がマスクされるだけである．

次に，最上位を除く第 2 層の故障をマスクする方法について述べる．現在の VLSD の実装では，RAID4 は 2 故障でフェールセーフするようになっている．すなわち，データを破壊する前に故障する．それゆえ，第 2 層に 2 つ以上の故障が検出されると全体が停止してしまう．しかし，そうする代わりに最上位の RAID4 を第 2 層の正常な RAID4 に置き換えることで問題なくデータを読み書きすることができる．このことも同様にシミュレーションで確認した．ゆえに，MeshRAID では，RAID55 で救済できない RAID コントローラの故障からも回復する手段が残されている．ただし，そのためには最上位の RAID4 は通常と異なるモードで動作する必要がある．

最後に最上位の RAID4 が故障したときは全体が停止する．MeshRAID では最上位 RAID4 が単一故障点である．しかし，現在はまだ実装していないが，最上位 RAID4 を回避して直接第 2 層の正常な RAID4 にアクセスする手段を与えれば，最上位 RAID4 の故障をマスクすることも可能である．MeshRAID には複数の RAID44 が内在し，それらが正常であれ

ば正しく機能する．このように MeshRAID はコントローラ故障に対して優れた耐故障性を持つ．

ここで，階層型 RAID と MeshRAID の一般的な耐故障性について議論する．階層型 RAID では，階層を増やすほど信頼性が高まり，同時に規模も拡大する．たとえば，n 層の RAID があり，各層が m 個の RAID4/5 からなるとすると，その規模は m^n であり，耐故障性は 2^n-1 である．MeshRAID は経路が多重化された階層型 RAID であるから，原理的には階層型 RAID と等しい耐ディスク故障を持つ．したがって，各次元の大きさを m とする n 次元 MeshRAID の規模は m^n であり，耐故障性は 2^n-1 である．ただし，コントローラ故障を含むとき，MeshRAID は複数の経路を持つため，その耐故障性はつねに階層型 RAID より大きい．

次に，MeshRAID を容量効率の点から評価する．階層型 RAID では，階層が増えると容量効率が低下し，経済性が急速に劣化する．RAID の経済性は容量効率で決定される．各層のディスク数 m が十分大きいとき，n 層 RAID の容量効率は以下の式で表される．

$$\left(\frac{m-1}{m}\right)^n = \left(1-\frac{1}{m}\right)^n \approx 1-\frac{n}{m}$$

ゆえに，$m \gg n$ であることが望ましい．本研究では，複製方式との分岐点である容量効率 50%以上を目標としている．しかし，普及品 RAID では，m が小さく規模の拡大が困難である．

MeshRAID の容量効率はつねに階層 RAID と等しい．各次元の大きさを m とする n 次元 MeshRAID の容量効率も前式で与えられる．MeshRAID は階層 RAID に比べて，経路が冗長であるが，ディスク（に保存された情報）が冗長なわけではない．よって，容量効率に関しては有利でも不利でもない．

次に，MeshRAID の性能を同規模の RAID55 と比較した．ベンチマークは 400 回のランダムアクセスである．ただし，データの大きさが 1 ブロックに収まる Small Read（SR）/Small Write（SW）とデータの大きさがストライプグループ全体にわたる Large Read（LR）/Large Write（LW）に分けた．SR/SW，LR/LW は RAID の原典以来の一般的な評価法である．結果を**表 3** に示す．数値はベンチマークの結果そのものであるため，相対値として読む必要がある．評価に用いた機種は Panasonic CF-Y4（CPU Intel Pentium M 1.6 GHz，RAM 1 GB，HDD 60 GB）である．なお，いずれも VLSD による実装を比較したものであるため，一般的なハードウェア RAID とは大きく異なる．

RAID55 は最も性能が高い．これは最も構造が単純であるためである．コンポーネントを経由するほどオーバヘッドが増加し，性能は低下する．MeshRAID は階層が深いため遅い．よって，本論文で提案した仮想ディスクのコンポーネントベースの開発法は開発時間を短縮することには秀でているが，性能を重視する運用段階では実装は新たに行う必要がある．

また，**表 4**，**表 5**，**表 6** が示すように，故障数が増えても一概に遅くなるとはいえない．故障数が増えると，全体を故障と判断するため，かえって動作が簡略化できることがある．

最後に MeshRAID の特性を**表 7** にまとめた．MeshRAID は 3 耐ディスク故障であり，かつコントローラ故障にも強い．一方，コンポーネント方式による実装法では，あまり高い

表 3　故障時の 3 耐故障 RAID の性能比較
Table 3　Comparison of 3FT RAIDs in no failure.

Class	Small Read[s]	Small Write[s]	Large Read[s]	Large Write[s]
RAID55	0.4	1.4	0.4	1.3
MeshRAID	2.2	2.7	2.1	2.8

表 4　故障時の 3 耐故障 RAID の性能比較
Table 4　Comparison of 3FT RAIDs in 1 failure.

class	Small Read[s]	Small Write[s]	Large Read[s]	Large Write[s]
RAID55	0.4	1.0	0.4	1.1
MeshRAID	2.5	2.5	2.0	2.7

表 5　故障時の 3 耐故障 RAID の性能比較
Table 5　Comparison of 3FT RAIDs in 2 failures.

class	Small Read[s]	Small Write[s]	Large Read[s]	Large Write[s]
RAID55	0.4	1.3	0.4	1.1
MeshRAID	2.1	2.8	2.2	2.4

表 6　故障時の 3 耐故障 RAID の性能比較
Table 6　Comparison of 3FT RAIDs in 3 failures.

class	Small Read[s]	Small Write[s]	Large Read[s]	Large Write[s]
RAID55	0.4	1.0	0.4	1
MeshRAID	2.1	2.7	2.1	2.1

表 7　ID 方式の比較
Table 7　The features of RAID classes.

RAID 方式	コントローラ故障	3FT	性能
MeshRAID	○	○	×（本実装）
階層型 RAID	×	○	○
直交 RAID	○	×	△

性能は得られない．しかし，この実装はラピッドプロトタイピングには適しており，本実装を通じて MeshRAID の動作検証が行えたことから，MeshRAID の有効性が証明され，性能を重視した実装を開始する十分な理由が得られたことになる．

7. ま と め

本論文では，3 耐故障直交 RAID である MeshRAID を提案した．MeshRAID は同じ 3 耐故障 RAID である RAID55 よりコントローラ故障に強い．また，我々は VLSD を用いて実装した．VLSD の多様なクラスコンポーネントを組み合わせることで新たな RAID を容易に実現することができることを示した．この手法は新たに複雑な RAID を構成するうえで有効な手法であるといえる．一方，性能面では必ずしも最善の手法とはいえない．

本研究では，MeshRAID を既存コンポーネントの組合せで実装した．そのため，性能は他実装に及ばなかった．今後は，オールインワンで実装することで性能の向上を図る．また，MeshRAID は VLSD という SW RAID で実現された．同様な構成で HW RAID として実現できる保証はない．今後は，HW RAID としての実現方法も検討する．さらに，直交次元を増やしたり，RAID MP（Multiple Parity）と組み合わせたりすることで，さらに信頼性を高める予定である．

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) Chen, P.M., Lee, E.K., Gibson, G.A., Katz, R.H. and Patterson, D.A.: RAID: High-Performance, Reliable Secondary Storage, *ACM Computing Surveys*, Vol.26, No.2, pp.145–185 (1994).
2) Accusys: The RAID Architect. http://www2.accusys.com.tw/FAQDetail.aspx?Lan=en&FAQ=39
3) Baek, S.H., Kim, B.W., Joung, E.J. and Park, C.W.: Reliability and Performance of Hierarchical RAID with Multiple Controllers, *Proc. 20th Annual ACM Symposium on Principles of Distributed Computing*, pp.246–254 (2001).
4) Chai, E., Uehara, M., Mori, H. and Sato, N.: Virtual Large-Scale Disk System for PC-Room, LNCS 4658, *Network-Based Information Systems*, pp.476–485 (2007.9).
5) Chai, E., Uehara, M. and Mori, H.: A Case Study on Large-Scale Disk System concatenating Free Space, *Proc. IEEE 2nd International Conference on Innovative Computing, Information and Control* (*ICICIC2007*) (2007.9).
6) Chai, E., Uehara, M. and Mori, H.: Evaluating Performance and Fault Tolerance in a Virtual Large-Scale Disk, *Proc. 22nd International Conference on Advanced Information Networking and Applications* (*AINA2008*), pp.926–933 (2008.3).
7) Chai, E., Uehara, M. and Mori, H.: 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).
8) Uehara, M.: Security Framework in a Virtual Large-Scale Disk System, *Proc. IEEE 10th International Workshop on Multimedia Network Systems and Applications* (*MNSA2008*), pp.30–35 (2008.6)
9) Chai, E., Uehara, M., Murakami, M. and Yamagiwa, M.: Online Web Storage using Virtual Large-Scale Disks, *Proc. 3rd International Workshop on Engineering Complex Distributed Systems* (*ECDS-2009*), pp.512–517 (2009.3).
10) Matsumoto, K. and Uehara, M.: N-nary RAID: 3-resilient RAID based on an N-nary number, *Proc. 23rd International Conference on Advanced Information Networking and Applications* (*AINA2009*), pp.249–255 (2008.5).
11) Uehara, M.: Combining N-ary RAID to RAID MP, *Proc. 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).
12) 上原　稔：仮想ディスクモデリング言語に基づくストレージ管理，マルチメディア通信と分散処理ワークショップ論文集（DPSWS2009），pp.241–246 (2009.10).
13) Virtual Large Scale Disk. http://sourceforge.jp/projects/vlsd/

(平成 22 年 5 月 11 日受付)
(平成 22 年 11 月 5 日採録)

上原　　稔（正会員）

1964 年生．1993 年慶應義塾大学大学院博士課程（計算機科学専攻）単位取得退学．同年東洋大学工学部情報工学科講師．1995 年博士（工学）．1998 年東洋大学工学部情報工学科助教授．2005 年同教授，2009 年総合情報学部現在に至る．分散計算，情報検索等に興味を持つ．ACM，IEEE 各会員．

村上　　真（正会員）

1972 年生．1999 年早稲田大学大学院理工学研究科情報科学専攻修士課程修了．2002 年早稲田大学大学院理工学研究科情報科学専攻博士課程単位取得退学．同年東洋大学工学部情報工学科講師．2003 年早稲田大学大学院理工学研究科博士（情報科学）．2005 年東洋大学工学部情報工学科助教授．2009 年総合情報学部准教授．コンピュータビジョン，画像認識に関する研究に従事．ACM，IEEE，電子情報通信学会各会員．

山際　　基（正会員）

1974 年生．2008 年東洋大学大学院工学研究科情報工学専攻博士後期課程修了．博士（工学）．同年東洋大学工学部情報工学科講師．2009 年より東洋大学総合情報学部助教．超音波イメージング，ソフトコンピューティング応用研究に従事．電子情報通信学会，日本音響学会，プロジェクトマネージメント学会各会員．