

2007年度
オープンソースソフトウェア活用基盤整備事業
第I期 テーマ型（調査）

「Linux ディスク冗長化機能の適用評価と
最適な適用方法の調査」
— 調査報告書 —

2008年 1月

独立行政法人 情報処理推進機構

目次

第 1 章 はじめに	1
1. 本調査の目的	1
1.1. 背景	1
1.2. 目的	1
2. 本調査の方針	2
2.1. 評価対象とする RAID 機能の概要	2
2.1.1 md (Multiple Devices)	2
2.1.2 DM (Device Mapper)	3
2.2. 評価手法	4
2.3. 評価環境	7
2.3.1 ハードウェアプラットフォーム	7
2.3.2 OS 環境	8
3. 用語定義	10
第 2 章 ディスク故障パターン	13
1. ディスクエラー処理の概要	13
1.1. SCSI ディスクのエラー	13
1.1.1 ソフトウェアレベルでのエラー処理	13
1.1.2 ハードウェアレベルでのエラー処理	15
1.2. SATA ディスクのエラー	16
1.2.1 ソフトウェアレベルでのエラー処理	16
1.2.2 ハードウェアレベルでのエラー処理	17
2. 現実に発生するディスク故障パターンの調査結果	17
2.1. SCSI ディスクのエラー	18
2.2. SATA ディスクのエラー	18
2.3. ハードウェアレベルでのエラー処理の設定	19
2.4. タイムアウトが発生するケース	19
3. 模擬故障内容	20
3.1. 模擬故障パターン	20
3.2. 模擬故障の動作説明	20

4. まとめ	23
第3章 故障模擬ライブラリ	25
1. Linux の SCSI I/O 処理	25
1.1. Linux の SCSI I/O 処理概要	26
1.1.1 SCSI 処理の流れ	26
1.1.2 SCSI 中間層のエラーハンドラ	27
1.1.3 cmd/sense の判断	28
1.1.4 SCSI 中間層の done オペレーション	31
1.1.5 scsi_io_completion	31
1.2. Linux の SCSI I/O に関するデータ構造について	32
1.2.1 scsi_cmnd 構造体	32
1.2.2 request 構造体	32
1.2.3 SCSI command block	33
1.2.4 SCSI Sense data	33
2. SystemTap 模擬故障フック	34
2.1. SystemTap による模擬故障を実現するための考察	34
2.1.1 SCSI コマンドがエラー応答する場合の擬似	34
2.1.2 SCSI コマンドの応答が無い場合の擬似 (擬似 timeout 発生)	34
2.2. SystemTap 模擬故障フック実装方法について	35
2.2.1 SCSI コマンドがエラーを応答する場合の擬似	35
2.2.2 SCSI コマンドの応答が無い場合の擬似 (擬似 timeout 発生)	36
2.2.3 故障パターン種類別実装方法について	37
2.2.4 擬似的に発生させる故障種別について	37
第4章 機能評価	39
1. 機能評価方針	39
2. ソフトウェア RAID の操作手順	39
3. 機能評価結果	45
3.1. SATA デバイスの機能評価結果	45
3.2. SCSI デバイスの機能評価結果	47
4. 機能評価結果の分析	48
4.1. SATA デバイスの機能評価結果の分析	48

4.2.	SCSI デバイスの機能評価結果の分析	53
5.	機能評価の追加評価結果	56
5.1.	SATA デバイスの機能評価の追加評価結果	56
5.2.	SCSI デバイスの機能評価の追加評価結果	56
6.	機能評価の追加評価結果の分析	57
6.1.	SATA デバイスの機能評価の追加評価結果の分析	57
6.2.	SCSI デバイスの機能評価の追加評価結果の分析	58
7.	まとめ	58
 第 5 章 性能評価		 61
1.	性能評価方針	61
1.1.	性能評価内容	61
1.2.	性能評価環境	62
1.3.	性能評価項目	62
1.3.1	性能測定プログラムによる I/O 処理性能測定	62
1.3.2	復旧処理時間測定	62
2.	性能測定プログラムによる評価結果	62
2.1.	通常運用時の RAID 未構築時と各 RAID レベルの性能測定結果	62
2.2.	縮退状態での各 RAID レベルの通常運用時の性能測定結果	67
2.3.	復旧処理中の各 RAID レベルの通常運用時の性能測定結果	70
2.4.	復旧処理時間の測定結果	73
3.	まとめ	75
 第 6 章 品質評価		 76
1.	品質評価方針	76
1.1.	品質評価内容	76
1.2.	品質評価項目	76
1.3.	故障ディスク交換評価	77
1.3.1	評価内容について	77
1.3.2	各品質評価項目における確認内容	78
1.3.3	評価手順	78
2.	品質評価結果	78

2.1.	SATA デバイスの品質評価結果	79
2.2.	SCSI デバイスの品質評価結果	85
3.	品質評価結果の分析	90
3.1.	RHEL4.5 カーネル上の品質結果の分析	90
3.2.	コミュニティカーネル上の品質結果の分析	92
4.	まとめ	93
第7章 結論		95
1.	機能評価	95
2.	性能評価	95
3.	品質評価	96
4.	まとめと今後の予定	97
4.1.	ソフトウェア RAID の利用者の視点	97
4.2.	開発コミュニティの視点	97
参考文献		99
付録A RAID の概要		100

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Red Hat は米国およびその他の国における Red Hat, Inc.の登録商標または商標です。

MIRACLE LINUX は、ミラクル・リナックス株式会社が権利を有する商標です。

Novell は米国および日本における Novell, Inc.の登録商標です。

SUSE は、Novell, Inc.の一部門である Novell SUSE LINUX Products GmbH の登録商標です。

その他記載されている会社名および商品名は各社の登録商標または商標です。

第1章 はじめに

本章ではまずこの調査の背景、目的について触れた後、本調査の方針について説明する。次に評価対象となる Linux のソフトウェア RAID 機能について説明した後、本報告書中で用いる用語の定義を行う。

1. 本調査の目的

1.1. 背景

近年ハードウェアの追加無しに実現可能であり、より安価なソフトウェア RAID (Redundant Array of Independent Disks) を利用する機会が増えている。さらに基幹系システムでもソフトウェア RAID を導入するケースも出てきている。今までこのようなシステムでは専用のハードウェア (RAID コントローラ) を使用するハードウェア RAID や、品質的により安定しているベンダ製の商用のソフトウェア RAID ドライバが用いられるケースが多かった。しかしハードウェア RAID は RAID コントローラ分だけ価格が高くなり、また商用ドライバはソースコードが公開されていないため、ユーザが利用しているカーネルバージョンに必ずしも対応していないことや、新しいカーネルバージョンへの対応には長期間要するという問題があった。このため、今後は Linux カーネルに含まれて提供されるオープンソースソフトウェア (OSS) の md (Multiple Devices) や DM (Device Mapper) を用いる事例が増加することが予想される。

RAID 機能では個々のディスクが故障した場合でも安定して処理が継続できる必要があるため、エラー処理が重要となる。一般的にエラー処理は通常処理に比べて実行頻度が低いため、体系的にテストを実施しない限り品質を高く維持することは困難である。md、DM においてもコミュニティでも使用中に発見された問題については個別にバグ修正が行われているが、設計時の抜けやコーディングミス等がそのまま残されている場合が多い。

ユーザがあえてソフトウェア RAID 構成を選択するのは、ストレージに対して高い信頼性と可用性を求めている場合である。このようなケースでのソフトウェアの不具合による処理の中断や、ファイル中のデータが失われるような障害が発生すると、ビジネスや社会に対してインパクトがあるだけでなく、OSS に対する信頼を失わせることにもなる。

1.2. 目的

Linux のソフトウェア RAID 機能の品質を明らかにし、今後の品質向上につなげることは、OSS 活用の拡大にとって大きな意味がある。今回の調査では、md の各 RAID レベル及び DM の dm-mirror の機能、性能、品質を明らかにし、問題点及び復旧手順を明確化することを目的とする。調査結果は Linux のソフトウェア RAID 開発のコミュニティと、ソフトウェア RAID 機能の利用を考えるユーザに以下の利益をもたらす [図 1]。この結果、システム障害の未然防止や、ダウンタイムが短縮でき、ソフトウェア RAID を利用したシステムでの可用性向上が期待できる。

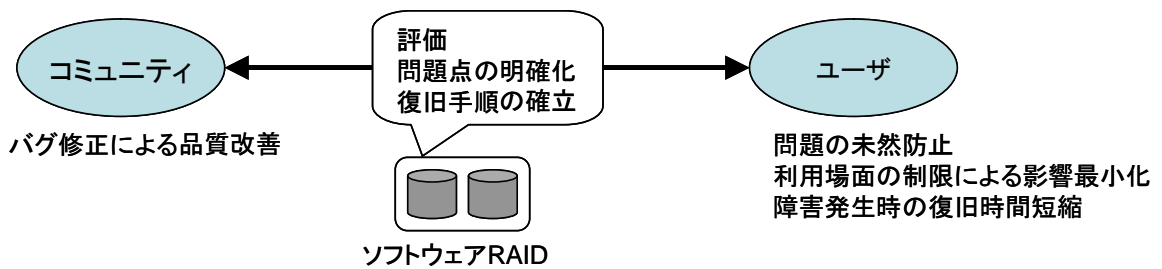


図 1 : Linux ソフトウェア RAID 機能評価の利点

- 今回の調査で発見された致命的な問題をコミュニティに報告するとともに、開発した評価プログラムを公開することで、Linux のソフトウェア RAID 機能の改善を図ることができる。
- Linux のソフトウェア RAID の品質状況を明らかにすることで、利用者が信頼性の高い機能や RAID レベルを選択することで未然に問題発生リスクを低減することができる。また性能的な要件が高い用途では、縮退運転や復旧処理中を含めたソフトウェア RAID の性能情報を参考に性能問題を防ぐように設計することが可能となる。
- ソフトウェア RAID 機能で発見された問題が修正されるまでの間、各装置の利用用途に応じて運用を制限することでシステムのコストを低減しつつ、個々の障害のシステムへの影響を最小限に抑えることが可能となる。
- 復旧までの手順を明確化し作業時間を短縮することで縮退運転に陥っている時間を短くすることができ、致命的な多重障害の可能性を低減する。また複数のディスクでたまたまリードやライトエラーが重なった場合の復旧手順を明確化することで、そのような状態からの復旧時間を短縮し、ダウンタイムを短くする。

2. 本調査の方針

本調査で実施する評価は、Linux のソフトウェア RAID 機能に関する機能評価、性能評価、品質評価の三つに大きく分類される。本節では調査対象となる RAID 機能の概要について説明した後、機能評価、性能評価、品質評価で実施する評価項目について説明する。また最後に以降の評価で使用した評価環境について説明する。

2.1. 評価対象とする RAID 機能の概要

評価対象となる Linux のソフトウェア RAID 機能としては md (Multiple Devices) と、DM (Device Mapper) の二つがある (RAID レベルに関する説明については付録 A を参照のこと)。

2.1.1 md (Multiple Devices)

md は Neil Brown や Ingo Molner 等によってメンテナンスされている Linux 上のソフトウェア RAID 用のドライバである。md はソフトウェア RAID 以外の機能として、Fibre-Channel 等のマルチパス機能などごく限られたものしか持たない反面、ソフトウェア RAID 機能については、RAID0 (ストライ

ピング)、RAID1 (ミラーリング)、RAID4 (Data Guarding)、RAID5 (Distributed Data Guarding) 、RAID6 (Advanced Data Guarding)、RAID10 (RAID1+RAID0)の豊富な機能を持つ。

今回の評価では、冗長化機能の無い RAID0 と RAID4 を除いた、RAID1、RAID5、RAID6、RAID10 を評価対象とした。RAID0 は冗長化機能を持たないため、また RAID4 は書込みの際にパリティディスクにアクセスが集中するという問題がありほとんど使用されることがないため評価対象外とした。

md の開発の歴史は古く、1997 年に RAID0 と RAID1 の機能が Linux の 2.0 カーネル向けに出されており、その直後に RAID4 と RAID5 の機能も追加されている。その後は 2004 年に RAID6 と RAID10 の機能が Linux の 2.6 カーネルで取り込まれている。開発コミュニティでは linux-raid@vger.kernel.org のメーリングリスト上で議論が行われている。最近のメーリングリスト上のトラフィックの推移を図 2 に示すが、継続的に開発が続いていることがわかる。

なお、RHEL4.5 のカーネルでは上記 RAID レベルの機能は全て持っているものの、Red Hat から公開されている「システム管理ガイド」[参考文献(p.99)の 1] の「II. ファイルシステム」の「11 章 ソフトウェア RAID の設定」では、RAID レベルとして RAID0、RAID1、RAID5 のいずれかを設定するように指定している。

アレイの認識及び障害発生時の構成情報の更新は md ドライバによって行われる。アレイの作成、ディスク追加・削除、情報表示等の運用管理は mdadm によって行われる。

2.1.2 DM (Device Mapper)

DM は Alasdair Kergon がメンテナをしているドライバであり、ボリューム管理のための複数のカーネルモジュールから構成されている。DM はソフトウェア RAID 以外に、Fibre-Channel 等のマルチパス機能、スナップショット機能、暗号化機能など様々な機能を含んでいるが、ソフトウェア RAID 機能に限れば、RAID0 (ストライピング)機能を含んだ基本モジュールと、RAID1 (ミラーリング)機能を提供する dm-mirror の二つのモジュールしか提供されていない。今回の評価では dm-mirror のみが評価対象となる。

DM の RAID1 機能は 2004 年に Linux の 2.6 カーネルになって取り込まれており、歴史は比較的浅い。開発コミュニティでは dm-devel@redhat.com のメーリングリスト上で議論が行われている。最近のメーリングリスト上のトラフィックの推移を図 2 に示す。トラフィック中には RAID 以外の議論に関するメールも含まれているため、RAID 機能に限ると、コミュニティのサイズは md に比べ小さい。当初 DM が Linux の 2.6 カーネルで登場した時点では、ソフトウェア RAID 機能も DM に統一されると言われたこともあり期待が高かったが、現在でもまだ md と DM は共存したままの状態である。

アレイの認識及び障害発生時の構成情報の更新、アレイの作成、ディスクの追加・削除、情報表示等の運用管理を行う方式としては、dmraid を用いる場合と、LVM2 を用いる場合の 2 通りの方法に大別される。

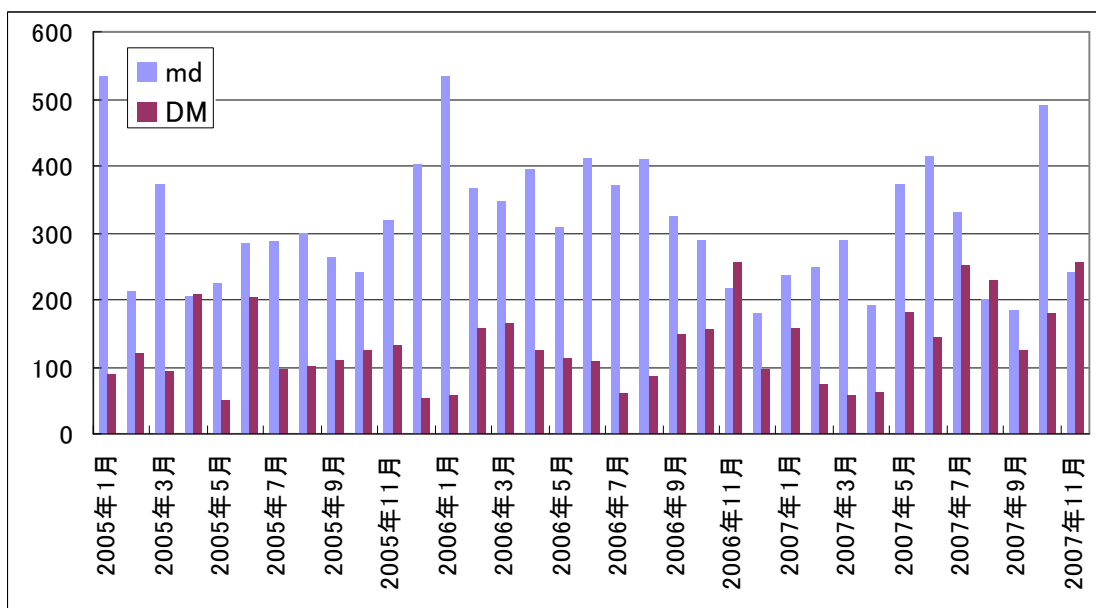


図 2：最近の md と DM のコミュニティメーリングリストのトラフィック

dmraid コマンドを用いる方式は、デバイスドライバベンダが提供するソフトウェア RAID 機能のうち、RAID アレイの作成や管理についてはデバイスドライバベンダ提供のツールを使用し、デバイスドライバを DM のドライバで置き換えるものである。dmraid コマンドは RAID の認識及び DM の設定を行う。現在 dmraid でサポートされている RAID アレイタイプは以下のものがある。

- Highpoint HPT37X
- Highpoint HPT45X
- Intel Software RAID
- LSI Logic MegaRAID
- NVidia NForce
- Promise FastTrack
- Silicon Image Medley
- VIA Software RAID

RAID アレイの作成や削除は、RAID アレイタイプ毎にデバイスドライバベンダが提供する独自のツールを用いて行う。RAID アレイ作成後は dmraid コマンドを用いて Linux の RAID アレイ認識や管理作業を行う。調査を行った時点で、dmraid コマンドには障害発生時などに構成情報を更新する機能は備わっていなかった。

LVM2 を用いる場合は、RAID アレイを構成するディスクをまとめたグループから RAID1 の機能を持つ論理ボリュームを構成して利用する。ボリュームの管理には lvm2 パッケージ中に含まれるコマンドを利用する。

2.2. 評価手法

機能評価、性能評価と品質評価に関しては評価用のテストプログラム(ソフトウェア RAID 評価

プログラム)を開発した後、評価を実施した。各評価項目は、ソフトウェア RAID の種類や利用可能な RAID レベルそれぞれについて実施した。

(1) 機能評価

機能評価では Linux のソフトウェア RAID 機能を利用、管理するに当たって必要な基本機能が正しく動作しているか評価する。具体的には、RAID アレイ上でファイル操作の基本システムコールが問題なく動作するか確認する機能評価用テストプログラムを開発する。ソフトウェア RAID を使用した場合も、既存ファイルシステムをそのまま利用しているため、ファイル操作の全システムコールを再評価する必要はない。

また md や DM に関するドキュメントやインターネット上に公開されている情報から各 RAID ボリュームの設定手順・復旧手順・管理機能を調査する。実機により md の RAID1、RAID5、RAID6、RAID10、及び DM の dm-mirror のそれぞれについて、SCSI、SATA それぞれのハードウェアを用いて以下の表 1 のテスト項目が可能か評価する。

表 1：機能評価における評価項目

項番	分類	機能評価テスト項目
1	設定	指定した RAID レベルで RAID アレイが作成できる
2		RAID アレイを削除できる
3		RAID アレイへのディスクの追加・削除ができる
4		指定したデバイスに対し不良マークがつけられる
5		RAID アレイのスーパーブロック内容の表示・更新・0クリアができる
6		RAID アレイのモード(読み書き、読み取り専用)切り替えができる
7		RAID アレイ上でのイベント検出時に管理者に通知する機能を持つ
8		RAID アレイ上でのイベント検出時に特定のプログラムが実行できる
9		RAID アレイのチェックが行える
10		ファイルシステムとして運用している RAID アレイ上で各種システムコールが問題なく動作する
11		RAID アレイを用い大容量ファイルシステムが作成できる
12		RAID アレイ上に巨大ファイルが作成できる
13	復旧	スペアディスク切り替え時に自動的に復旧処理を開始する。スペアディスク切替え後に自動的に復旧処理が実行されない場合は、復旧処理を手動で実行するための手順を明確化する
14		復旧処理の進捗状況・完了したことが確認できる
15		hotplug 機能が使える。hotplug が使用できない場合は故障ディスクをシステムから切り離し後、新しいディスクを接続し、RAID アレイに組込むための手順を明確化する

16		hotplug でディスク追加時に自動的に復旧処理を開始する。hotplug でディスク追加後に自動的に復旧処理が実行されない場合は、復旧処理を手動で実行するための手順を明確化する
17		縮退状態でブート可能であること
18		復旧処理中にブート可能であること
19		復旧処理中にリポートした場合に自動的に復旧処理を再開する。リブート後に自動的に復旧処理が再開されない場合は、復旧処理を手動で再開するための手順を明確化する
20	管理機能	管理者が故障ディスクを特定するのをサポートする
21		RAID の構成情報をバックアップ・リストアする機能を持つ

(2) 性能評価

性能評価では、性能評価用のプログラムを開発し、ソフトウェア RAID の通常運用時の性能低下、復旧処理中のオーバーヘッド、及び復旧動作にかかる時間を評価する。具体的には、アクセスサイズを変えながらブロックデバイス及びファイルシステムの読み書きにかかる時間を測定し、RAID のアクセス性能を計算する性能評価用テストプログラムを開発する。性能評価用テストプログラムは、md の RAID1、RAID5、RAID6、RAID10、及び DM の dm-mirror に対応可能なものとする。性能評価用テストプログラムを利用して SCSI、SATA それぞれのハードウェアを用いて以下の表 2 示すように、通常運用時や縮退状態での性能低下、及び復旧処理中のオーバーヘッドや復旧動作にかかる時間などを評価する。

表 2：性能評価における評価項目

項番	分類	性能評価テスト項目
1	通常運用	RAID なしの時の性能を測定する
2		各 RAID レベルの通常運用時に RAID アレイの性能を測定する
3	縮退状態	各 RAID レベルの縮退運転時に RAID アレイの性能を測定する
4	復旧処理	各 RAID レベルの復旧処理中に RAID アレイの性能を測定する
5		無負荷状態での各 RAID レベルの復旧時間を測定する
6		負荷状態での各 RAID レベルの復旧時間を測定する

(3) 品質評価

品質評価ではディスク故障発生時に呼ばれるエラー処理が正しく行われるかを中心に評価を行うために、Linux カーネル内に種々のディスク故障の結果、発生するエラーを模擬的に発生させるライブラリ(故障模擬ライブラリ)を開発する。このために最初に、文献等の調査を行い発生するエラーをリストアップした後で、サポート部門等へのヒアリングを行い、ディスク故障で返される条件について評価を行い、Linux カーネル内で Fault Injection すべき故障の種類、ソースコード上の位置を決定した後、故障模擬ライブラリの実装、開発を

行う。

これと平行して品質評価用テストプログラムを開発し、故障模擬ライブラリと組み合わせて品質評価を実施する。具体的には、md の RAID1、RAID5、RAID6、RAID10、及び DM の dm-mirror のそれぞれについて、品質評価用テストプログラムを用いて、運用中、縮退中、及び復旧中に故障を発生させることにより、ソフトウェア RAID の持つ冗長性でカバーできる範囲内での故障に対してソフトウェア RAID としてサービスを継続して提供できること、及び冗長性でカバーできない多重故障等が発生した時にパニックやデータ化けなどの障害を発生せずに正しくエラーを返しシステムとして運用継続できることを、表 3 にあげた項目により評価する。また合わせて、デバイスドライバを含めたシステムとしての品質評価を行うために、動作中に物理的にディスクを抜く操作を行う評価を行う。

表 3：品質評価における評価項目

項番	分類	品質評価テスト項目
1	運用中故障	スペアディスク付き環境でディスク1台に故障が発生した時、故障が検出でき、RAID アレイ及びシステムの運用に問題がない
2		スペアディスク無しの環境でディスク1台に故障が発生した時、故障が検出でき、RAID アレイ及びシステムの運用に問題がない
3	縮退中故障	ディスク2台に故障が発生した時、故障が検出でき、RAID アレイが運用できなくなるが、システムの運用に問題がない
4	復旧中故障	復旧処理中のスペアディスクに対してディスク故障が発生したときに、RAID アレイ及びシステムの運用に問題がない
5		復旧処理中のアクティブディスク1台に故障が発生した時、故障が検出でき、RAID アレイが運用できなくなるが、システムの運用に問題がない
6	故障ディスク交換	故障したディスクを挿抜した時、RAID アレイ及びシステムの運用に問題がない

2.3. 評価環境

以下では評価に使用したハードウェアプラットフォームと OS バージョンについて説明する。評価で使用したファイルシステムは最も一般的に用いられている ext3 ファイルシステムを利用した。

2.3.1 ハードウェアプラットフォーム

評価はホットプラグをサポートする SCSI、及び SATA2 の二種類のデバイスを用いて実施した。具体的には以下のハードウェア構成を使用した。

- SCSI プラットフォーム
Express5800/120Lh(Xeon 3.4GHz×2, メモリ 4GB, SCSI 36GB×6)
- SATA プラットフォーム
Super Server 6025B-T(Xeon 3.4GHz, メモリ 2GB, SATA 80GB×6)

SATA プラットフォームでは、Intel Software RAID が利用でき、ハードウェアに付属していた Intel Matrix Storage Manager というツールを用いて RAID1 構成を構築することができる。このため DM (dmraid)の評価としては、Intel Software RAID により構築した RAID アレイを用いて、SATA 環境上でのみ実施した。SCSI デバイスでは dmraid で使用できるソフトウェア RAID が構築できなかったため、dmraid による評価は行わなかった。

2.3.2 OS 環境

評価対象のカーネルは、基本的には評価開始時点でのコミュニティ最新版カーネルと Red Hat 社の Red Hat Enterprise Linux 4.5 (以降 RHEL4.5)を利用した。一部評価項目に関しては Novell 社の SUSE Linux Enterprise Server 10 SP1 (以降 SLES10SP1)、ミラクル・リナックス社の MIRACLE LINUX V4.0 SP2 (以降 ML4SP2)のディストリビューションに含まれるカーネル上でも評価を行った。評価で使用した各カーネルの具体的なバージョンを以下に示す。

- コミュニティカーネル : linux-2.6.22.6
- RHEL4.5 : 2.6.9-55.EL.smp
- SLES10SP1 : 2.6.16.46-0.12-smp
- ML4SP2 : 2.6.9-42.7AXsmp

また各 OS 環境において、RAID アレイを管理するために使用したツールを含むパッケージの情報を以下に示す。

コミュニティカーネル、及び RHEL4.5

mdadm Version 1.12.0
dmraid Version 1.0.0.rc14
lvm2 Version 2.02.21

SLES10SP1

mdadm Version 2.6
dmraid Version 1.0.0.rc13
lvm2 Version 2.02.17

ML4SP2

mdadm Version 1.11.0
dmraid Version 1.0.0.rc11
lvm2 Version 2.02.06

以降の本報告書は以下の構成からなる。

2 章では品質評価で模擬故障として用いるディスク故障のパターンを決めるために行った、ディスク故障に関する調査結果について報告し、3 章では 2 章の故障パターンを発生させられる故障模擬ライブラリについて説明する。

4 章ではソフトウェア RAID が設定、復旧と管理に必要な機能を備えているか、インターネット上に公開されているドキュメントから調査し、正しく動作していることを実機上での確認を行った。

5章ではソフトウェア RAID の通常運用中、縮退時、復旧処理中の時にベンチマークプログラムを用いて行った性能評価結果について報告する。

6章では 3章で説明した故障模擬ライブラリとテストプログラムを用いて、Linux のソフトウェア RAID ドライバのエラー処理系の品質に関する調査結果について報告する。

7章では 4章から 6章の調査結果をまとめた後、現在の Linux のソフトウェア RAID 機能を適用するための注意点と、今後ソフトウェア RAID 機能を改善して行くために取り組むべき活動についてまとめる。

3. 用語定義

用語	意味
サブライズリムーブ	動作中の機器で構成要素を、通常運用の状態のまま準備をせずに着脱する。hotplugの場合は事前にコマンドを実行して着脱に備える場合も許している点が異なる。
代替セクタ割り当て	ハードディスクにはあらかじめ故障に備えて媒体に予備の領域が確保されており、故障などで特定の部分にアクセスできなくなった場合、故障したセクタの代わりに用いる代替セクタをこの予備の領域から割り当てる。
パーティション	OSがファイルシステム等に利用するために、一台のディスクを複数の部分領域に分割した管理単位。
パーティションテーブル	パーティションの開始位置や大きさなどの管理情報が格納されたディスク上のテーブル。
パリティデータ	RAID5などの誤り訂正符号を用いて耐故障性を向上させる RAID レベルにおいて、故障発生時に元データを復元するために付加する冗長部分のデータ
CDB	Command Descriptor Block の略。SCSI デバイスに対して実行する SCSI コマンドの情報を保持するメモリブロック。
DM	Device Mapper の略。Linux のソフトウェア RAID 機能を実現する方式の一つ。DM のソフトウェア RAID 機能は RAID0 と RAID1 の二つしかないが、ソフトウェア RAID 以外の機能として Fibre-Channel 等のマルチパス機能、スナップショット機能、暗号化機能等様々な機能を提供している。
dm-mirror	DM で RAID1 を実現するためのモジュールの名前。
dmraid	ドライバベンダ固有の RAID アレイ構成を認識して、ベンダ固有のデバイスドライバの代わりに dm-mirror を用いて運用可能とするコマンドとそれを含むパッケージの名前
ext3 ファイルシステム	Linux で一般的に使用されているジャーナリング機能を持つファイルシステム。
Fault Injection	本来は故障が発生していない装置で、あたかも故障が発生したかのように見せる事象をハードウェアあるいはソフトウェアで発生させる技術。エラー処理系の評価等に用いられる。
HBA	Host Bus Adapter の略。SCSI デバイスを接続するためのハードウェア。
hotplug	動作中の機器で構成要素を着脱すること。
Logical Volume Management	ディスク上の複数のパーティションをまとめて、一つのディスク領域として見せる技術。複数のパーティションを結合やストライピングにより大きな領域として扱ったり、同じデータを複数のパーティションに格納してミラーリングに使用したりできる。

LVM2	Linux 上で Logical Volume Management 機能を提供するツール。2.4 カーネルのころにあった LVM と機能的には互換性を持ちながら、2.6 カーネル中の DM を利用して実現されている。
md	multiple devices の略。Linux のソフトウェア RAID 機能を実現する方式の一つ。md はソフトウェア RAID 機能として、RAID0、RAID1、RAID5、RAID6、RAID10 等の豊富な機能を持つ反面、ソフトウェア RAID 以外の機能は Fibre-Channel 等のマルチパス機能などごく限られたものしか持たない。
mdadm	md ドライバが使用する RAID アレイの作成、ディスク追加・削除、情報表示等の運用管理を行うためのコマンドと、それを含むパッケージの名前
RAID	Redundant Array of Independent Disks の略。冗長性を持たせたディスクを用いて、個々のディスクが故障してもディスク上のファイルを継続的にアクセスすることを可能とする技術。
RAID0	ストライピング (複数のディスク上にデータを分散して格納することでディスクスループットを向上させる) を実現する RAID 機能の一つ。データに冗長性を持たせていないため、RAID を構成するディスクが一つでも故障を起こすとボリューム全体がアクセスできなくなる。
RAID1	ミラーリング (複数のディスク上に同一の内容を格納する) を実現する RAID 機能の一つ。
RAID4	一つのディスクが故障してもデータが復元できるように冗長性を持たせた情報を複数のディスクに格納する機能を実現する RAID 機能の一つ。RAID5 と異なりパリティデータが一つのディスクに集まっているため、write 時にパリティディスクに負荷集中が起きやすい。
RAID5	一つのディスクが故障してもデータが復元できるように冗長性を持たせた情報を複数のディスクに格納する機能を実現する RAID 機能の一つ。RAID4 でのパリティディスクへの負荷集中の問題を解決するために、パリティデータを全ディスクで分散して格納する。
RAID6	二つのディスクが故障してもデータが復元できるように冗長性を持たせた情報を複数のディスクに格納する機能を実現する RAID 機能の一つ。
RAID10	RAID1 と RAID0 を組合わせて、性能向上と耐故障性向上の両方を実現する RAID 機能の一つ。
RHEL	Red Hat Enterprise Linux の略。Red Hat 社が提供しているディストリビューション。
SATA2	コンピュータにディスクを接続する Serial ATA 規格の一つ。最初の SATA の規格に比べて通信速度が引き上げられたとともに、エラー処理等が強化された。
SCSI	Small Computer System Interface の略。コンピュータにディスクを接続する規格の一つ。

SCSI 中間層	Linux カーネル内の SCSI I/O に関する共通処理部分。現在では IDE/SCSI/SATA/SAS HDD のデバイスドライバなど、SCSI に限らず様々なデバイスが SCSI 中間層を利用している。
SCSI ローレベルドライバ	Linux カーネルに於いて、SCSI 中間層を利用してデバイスを制御するデバイスドライバ。
SLES	SUSE Linux Enterprise Server の略。Novell 社が提供しているディストリビューション。
sense key	SCSI でエラーが発生した時に、エラー発生要因を示すデータ。
SystemTap	動作中の Linux 2.6 カーネル内部の情報を収集・解析するツールの名前。カーネル内の任意のポイントにフックをしかけて指定した処理を実行することができる。実行する処理内部でカーネル内部の状態を書き換えることで Fault Injection に利用することが可能である。
write ペナルティ	パリティデータを用いる RAID レベルで、データ更新時に、更新前のデータとパリティデータを読み出し、更新パリティデータを作成後に書き込むことにより発生する余分なアクセス。

第2章 ディスク故障パターン

本章では、ソフトウェア RAID 機能が耐故障性を向上させるために、対処すべきディスク故障パターンについて調査する。最初に SCSI と SATA の規格から OS に見えるエラーパターンを調査し、次にその中から実際の製品で発生した故障パターンをヒアリングによって絞り込み、その結果ディスク故障を 8 つのパターンに整理した。この結果を用いて第 4 章で説明する故障模擬ライブラリを開発し、それを用いて第 6 章の品質評価でソフトウェア RAID 機能の品質評価を行った。

1. ディスクエラー処理の概要

1.1. SCSI ディスクのエラー

SCSI ディスクのエラー処理を決める要因は、OS 内のローレベルドライバや SCSI 中間層などのソフトウェアレベルでのエラー処理と、ハードウェアレベルでのエラー検出やエラー発生時の動作に大きく分類することができる。

1.1.1 ソフトウェアレベルでのエラー処理

SCSI エラーが発生した時に Linux カーネル内で参照する情報を表 4 に示す。以降、それぞれの情報について説明する。

表 4 : SCSI エラー発生時に OS が参照する情報

情報元	取得方法
status code	CDB から判断する
sense key	sense data から判断する
asc/ascq	sense data から判断する
message code	HBA のレジスタ (chip/driver 依存) から判断する

CDB: Command Descriptor Block の略。SCSI デバイスに対して実行する SCSI コマンドの情報を保持するメモリブロック。

HBA: Host Bus Adapter の略。SCSI デバイスを接続するためのハードウェア。

(1) status code

status code は SCSI コマンドの実行状態を表す。status code はそれぞれのコマンドの CDB 中に保持されており、ハードウェアにより適宜値が書き換えられる。SCSI architecture model (SAM)仕様 [参考文献(p.99)の 6]で定義された status code の値を表 5 に示す。Linux の SCSI ドライバはまず status の値を見て、コマンドの実行結果が正常かエラーかを判断する (status code の詳細については SAM 仕様「5 章 status」の項を参照)。

表中の BUSY は SCSI コントローラで規定された時間内にデバイスから応答が返って来なかった状態を示す。ただし SCSI 中間層が行うコマンドのタイムアウト処理は、一定時間経過してもコマンドが完了していないことを OS 内のタイマを使用して検出するものであり、SCSI

コマンドの BUSY とは独立したものである。

表 5 : SCSI architecture model(SAM)で定義された status の値

SAM 値	意味	備考
0x00	GOOD	正常終了
0x02	CHECK CONDITION	エラー発生
0x04	CONDITION MET	
0x08	BUSY	コマンドが BUSY 状態である場合。SCSI デバイスが HBA に規定時間内に応答を返さなかった場合等に発生する
0x10	obsolete	
0x14	obsolete	
0x18	RESERVATION CONFLICT	既存の reservation に違反した場合
0x22	obsolete	
0x28	TASK SET FULL	task set を作るリソースが足りない場合
0x30	ACA_ACTIVE	
0x40	TASK_ABORTED	hard reset 等でコマンドがアボートした場合
other	reserved	

表 6 : SCSI primary command(SPC)[参考文献(p.99)の 5] で定義された sense key の値

Sense key の値	意味
0x00	NO_SENSE
0x01	RECOVERED_ERROR
0x02	NOT_READY
0x03	MEDIUM_ERROR
0x04	HARDWARE_ERROR
0x05	ILLEGAL_REQUEST
0x06	UNIT_ATTENTION
0x07	DATA_PROTECT
0x08	BLANK_CHECK
0x09	vendor specific (Linux として global な対応はなし)
0x0a	COPY_ABORTED
0x0b	ABORTED_COMMAND
0x0d	VOLUME_OVERFLOW
0x0e	MISCOMPARE

(2) sense key

SCSI でエラーが発生すると、SCSI コマンドを発行する際に指定したバッファに、エラー原因を切り分けるための情報(sense key)が格納される。SCSI primary command (SPC)仕様 [参考文献(p.99)の 5]で定義された sense key の値を表 6 に示す。

(3) Additional Sense Code (ASC) / Additional Sense Code Qualifier (ASCQ)

ASC/ASCQ は、sense key に加えてエラー発生要因の詳細情報をわたすために使用される。詳細は SPC の 4 章に記載されている。Linux カーネル内ではヘッダファイルにおいて define されていないものの、内部では sense key と合わせて使っている箇所もあるため、エラー発生時には適切な値に設定する必要がある。

(4) message code

message code は HBA 上のチップのレジスタから取得する情報であり、今回の目的であるディスクの模擬故障を発生させる際には直接関係しないため、詳しい説明は割愛する。

1.1.2 ハードウェアレベルでのエラー処理

ハードウェアレベルのエラー処理の動作を制御する設定項目で、本評価に影響を与える設定項目としては、デバイス上でエラー検知した場合の動作を制御する Read-Write Error Recovery mode page 中の設定項目がある。Read-Write Error Recovery mode page を用いて、エラーリカバリ発生時の動作を指定することができる項目を表 7 に示す。詳細については SCSI block command 仕様[参考文献(p.99)の 4] 6 章「Read-Write Error Recovery mode page」の項参照。

表 7 : SCSI のエラーリカバリ発生時の動作の制御

bit	意味
AWRE	1 の場合、write 時にリカバリ可能なエラーが発生した場合に自動的に、代替セクタの割当てが発生する。ERR、PER、DTE、DCR に従ったエラーの報告は再割当てが完了した後に発生する。
ARRE	1 の場合、read 時にリカバリ可能なエラーが発生した場合に、代替セクタの割当てが自動的に発生する。ERR、PRE、DTE、DCE に従ったエラーの報告は再割当てが完了した後に発生する。read 時のリカバリ可能なエラー発生検出は、ECC の利用や、retry で正しいデータが読めたことで判断する。
EPR	1 の場合、簡易リカバリになる。
PER	1 の場合、リカバリ可能なエラー検出を報告する。
DTE	1 の場合、リカバリ可能なエラーを検出すると、データバッファの通信を中断する。
DCR	1 の場合、ECC によるエラーリカバリを実施しない

注) ERR、PER、DTE、DCR の組合せによってエラーリカバリ動作の有無、データ転送の有無、エラー報告の有無を制御する。

表 8 : SCSI のデバイスビジー時の動作の制御

bit	意味
RAC	1 の場合、busy が check condition で返る
BUSY TIMEOUT PERIOD	コマンドが BUSY を返すまでの期間を指定することができる (100ms 単位)。

1.2. SATA ディスクのエラー

現在の Linux カーネルの SATA デバイスドライバは、SCSI 中間層の下に位置するローレベルドライバの一つとして実装されている。このため SATA ディスクのエラー処理を決める要因も SCSI ディスクの場合と同様に、OS 内のローレベルドライバと SCSI 中間層のソフトウェアレベルでのエラー処理と、ハードウェアレベルでのエラー検出やエラー発生時の動作に大きく分類される。

1.2.1 ソフトウェアレベルでのエラー処理

SATA でエラーが発生したときに Linux カーネル内で参照する情報を表 9 に示す。以下、それぞれの情報について説明する。

表 9 : SATA エラー発生時に OS が参照する情報

情報元	取得方法
Status register	Command block register から判断する
Error register	Command block register から判断する
SStatus	Status and control register から判断する
SError	Status and control register から判断する

(1) Status register

Status register の内容から SATA コマンドでエラー発生したことを確認できる。ATA7 仕様 [参考文献(p.99)の 2] で規定された Status レジスタの値を表 10 に示す。

表 10 : ATA 仕様で規定された Status register の値

値	意味	備考
0x00	ERR	have an error
0x08	DRQ	data request i/o
0x20	DF	device fault
0x40	DRDY	device ready
0x80	BSY	BSY status bit

(2) Error register

エラーが発生した場合は、Error register の内容から発生したエラーの原因を切り分けることができる。ATA7 仕様[参考文献(p.99)の 2] で規定された read / write 時の Error register の値を表 11 に示す。Linux では Error register の値を元に対応する SCSI の sense code を計算し、上位にある SCSI 中間層は生成された sense code の値を元にエラー処理を行う作りになっている。

表 11 : ATA 仕様で規定されたの Error register の値

値	意味	備考
0x02	NM	no media present
0x04	ABRT	command aborted
0x08	MCR	media change request
0x10	IDNF	ID not found
0x20	MC	media changed
0x40	UNC / WP	uncorrectable media error (read) write protect (write)
0x80	ICRC	interface CRC error

(3) SStatus / SError

SStatus と SError の二つは Serial ATA の通信状態やエラーを表すレジスタであり、今回の目的であるディスクの模擬故障を発生させる際には直接関係しないため、詳しい説明は割愛する。

1.2.2 ハードウェアレベルでのエラー処理

SATA ではブロック再割当てなどの訂正可能エラーのリカバリ処理はベンダ固有であり、ハードウェアやファームウェア内に閉じた形で実装されている。検出されたエラーに関する情報収集を行うための SMART command は提供されているが、エラー発生時の動作を制御する方法は一般的には存在しない。またタイムアウト処理に関しても、ソフトウェアで検出する必要がある（詳細は SATA2:Extension to SerialATA1.0a 仕様[参考文献(p.99)の 3] 4 章「Defect Management」の項参照）。

2. 現実に発生するディスク故障パターンの調査結果

SCSI と SATA のエラー情報を元に、ディスクアクセスで発生するエラーの種類と、その原因として考えられる故障のパターンについてヒアリングを行った。ヒアリング対象は日本電気株式会社内の PC サーバ用のディスクドライブの製品担当部門、HBA のドライブ担当部門、及び Linux サポート製品のリリース前評価を実施している部門である。

まずディスクドライブで故障が発生した時の動作は、ハードウェアからエラーが返されるケース

と、エラーが返されないケースに大きく分けることができる。

2.1. SCSI ディスクのエラー

ディスクドライブが故障した場合、エラーを発生させたコマンドの status code は必ず CHECK_CONDITION が返される。ヒアリングした部門で SCSI 及び SATA ディスクに関して遭遇した過去の障害事例において発生した sense key の値と、その時の故障原因をヒアリングした結果を表 12 にまとめる。ほとんどの場合、MEDIUM_ERROR と HARDWARE_ERROR の二種類のエラーとなる。

表 12 : SCSI エラー発生時の sense key の値と原因となった故障

sense key	Sense key
RECOVERED_ERROR	仕様上はディスクドライブでリカバリが行われた時に発生するものであるが、2.3 で説明するように、リカバリは OS に通知されない設定で使われているため、返されることはない。
MEDIUM_ERROR	ディスクドライブの媒体上の不良セクタによりリカバリ不能なエラーが発生した。
HARDWARE_ERROR	シーク時にディスクドライブの指定トラック位置を示すマークが検出できず、シークエラーが発生するケースが多いが、その他のディスクドライブ上のハードウェア故障により発生する場合もある。通常は固定的な故障となる。
ILLEGAL_REQUEST	不正な CDB が検出された場合やその SCSI デバイスではサポートされていないコマンドを受けた時に発生する。バグや設定ミス以外では発生しない。
UNIT_ATTENTION	新しいディスクを接続した時に、ディスクが接続されたことを通知するために返されることがある。故障では返らない。
DATA_PROTECT	書き込み不可となっているドライブに書き込もうとした場合に発生する。故障では発生しない。

2.2. SATA ディスクのエラー

ディスクドライブが故障した場合、Status register の ERR ビットがセットされた上で、Error register の対応するビットがセットされる。ただし表 13 の説明にあるように、現在ハードディスクの故障によりエラーが発生するケースは、ほとんど UNC ビットがセットされた場合となる。

表 13 : SATA エラー発生時の Error の値と原因となった故障

sense key	Sense key
IDNF がセットされる場合	以前はメディア上にセクタ番号を記録していたため、要求されたセクタ番号がメディア上に見つからなかった場合に IDNF がセットされる場合があったが、最近ではメディア上の位置でセクタ番号を表しているため、IDNF は発生しない。
UNC がセットされる場合	ディスクドライブ上のリカバリ不能なエラーが発生した。故障時はほとんどこのエラーが発生する。
ICRC がセットされる場合	DMA 等の転送中の CRC エラーが発生した。

2.3. ハードウェアレベルでのエラー処理の設定

1.1.2 で説明したように、SCSI インタフェースではエラーが発生した時の動作を制御することができるが、現在の Linux に含まれるデバイスドライバではその初期化処理は行っていないため、デフォルトの値のまま用いられている。PC サーバ用に出荷されているディスクの現在の設定では、ディスク装置上で発生したリカバリ可能なエラーに関しては、OS に報告しない設定となっている。

一方 SATA ではもともとリカバリ可能なエラーを OS に報告する機能はないため、以下の動作は SCSI と SATA 共通の動作となる。

ディスク上で発生したメディアエラーが検出された場合のディスクドライブ側の動作をまとめると以下ようになる。

(1) リカバリ可能な read エラー

リトライや ECC による修復でデータが読めた場合は、サーバには修復されたデータがそのまま返される。この際、代替セクタ割当てが行われる場合もある。

(2) リカバリ不能な read エラー

read アクセスに対してエラーを返すが、ディスク装置側でエラーの発生したセクタ番号を覚えておき、後で write アクセスが来たときに代替セクタが割り当てられ、以降は正常に read / write できるようにするデバイスも多い。

(3) write エラー

メディアの不良箇所にデータを書き込んでも、エラーが検出されず write が正常終了するケースが多い。この場合書かれたデータを read しようとした時にエラーが発生する。ただし可能性として write がエラーとして返る場合も想定すべきである。

2.4. タイムアウトが発生するケース

タイムアウトによりエラーが発生するケースとしては、ディスクドライブ側の一時的な過負荷で応答が間に合わなくなる場合や、ディスクドライブ中のコントローラやその上で動作するファームウェアの問題により応答がなくなるケースがある。しかしディスクドライブからの応答が無い場合、返ってきたステータスに従ってエラーの種類や原因を分類することはできない。

そこで、模擬故障を発生させる上でも、一時的なエラーであるか固定的なエラーであるかという

症状のみに着目して分類する。

3. 模擬故障内容

3.1. 模擬故障パターン

ディスクの故障パターンを元に、まず模擬故障として発生すべき故障を一時的なエラー応答、固定的なエラー応答、無応答の場合それぞれについて分類を行う。品質評価の際には、以下で説明する8種類の模擬故障を投入する。

- 一時的なエラー応答

転送中のエラーや、ディスクドライブ上の一時的な故障によりエラーが発生した場合が該当する。read アクセスで発生する場合と、write アクセスで発生する場合の2種類が存在する。

- 固定的なエラー応答

2.3.(2)にある通り、メディアエラーが発生した場合は write アクセスにより訂正可能な read エラーとして表すことができる。これ以外のケースとしてはそれぞれ、read アクセスのみ、write アクセスのみ、read と write アクセスの両方で固定故障が発生する場合の合計4種類が存在する(現実的には write アクセスのみでエラーが発生するパターンは考えにくい、ここでは評価パターンとして数えている)。

- 無応答の場合

過負荷などにより一時的に無応答になるケースと、ディスクドライブ上のコントローラの故障等が原因で固定的に無応答になるケースがある。前者は read アクセスで発生する場合と、write アクセスで発生する場合があるが、後者は全てのアクセス種別で無応答となるため、合計3種類存在する。

また故障を発生させる対象としては、テストプログラムでアクセスする特定のセクタを含む read 及び write SCSI コマンドだけでエラーを発生させるように限定した。これは複数のブロックをエラー発生対象として選んでも、エラー処理としては、アクセスしたそれぞれのブロックに対するエラー処理が多重に動くだけとなり、それぞれのエラー処理が正しく動いていれば全体としても正しく動作すると考えられるためである。

3.2. 模擬故障の動作説明

以下では故障模擬ライブラリで発生する8つの故障パターンについて、故障模擬ライブラリで実現すべき動作を説明する。

説明では、状態遷移図を用いて、各状態での read や write アクセスが発生した時の動作を示す。状態遷移図では、各列が状態番号を示し、各行がアクセス種別(read/write)を示す。アクセスの結果(OK や NG)と次にどの状態に遷移するかを矢印で示す。いずれの場合も状態1が初期状態となる。

(1) read 不可エラー

read のみ失敗する。write コマンドは成功しているものの、実際はエラーを検出できずに書き込まれたデータが失われる場合を想定している。

(凡例：出力(アクセス結果)/次状態)

イベント \ 状態	S ₁
read	エラー/S ₁
write	正常終了/S ₁

模擬故障投入後、対象となるブロックに対する read アクセスは失敗するが、write アクセスは成功する。固定故障であり状態は一つしか存在しない。

(2) read/write 不可エラー

read/write 共に失敗する。ディスクの重大故障を想定している。

(凡例：出力(アクセス結果)/次状態)

イベント \ 状態	S ₁
read	エラー/S ₁
write	エラー/S ₁

模擬故障投入後、対象となるブロックに対する read/write アクセスが共に失敗する。固定故障であり状態は一つしか存在しない。

(3) write によって訂正可能な read エラー

ディスクメディアのエラーが原因となって read エラーが発生したもの。同じところに write すると、代替セクタが割り当てられるため、正しいデータを書き込むことでデータが修復する。

(凡例：出力(アクセス結果)/次状態)

イベント \ 状態	S ₁	S ₂
read	エラー/S ₁	正常終了/S ₂
write	正常終了/S ₂	正常終了/S ₂

模擬故障投入直後、対象となるブロックに read アクセスが発生した場合 read エラーを起こす。read エラーは対象ブロックに write が発生するまで続くが、write アクセスがあると、その後の read アクセスは成功する。初期の read アクセス時にエラーが発生する場合は状態 S₁、一度 write アクセスが発生した場合が状態 S₂ で、それぞれの read/write アクセスの結果と次にどの状態に移るかが示されている。

(4) read 単発エラー

read の偶発的なエラーに対応。後続の read/write は正常に動作する。

(凡例：出力(アクセス結果)／次状態)

状態 \ イベント	S ₁	S ₂
read	エラー／S ₂	正常終了／S ₂
write	正常終了／S ₁	正常終了／S ₂

模擬故障投入直後、対象となるブロックに初めて read アクセスが発生した場合エラーとなるが、write アクセスの場合は成功となる。一度でも read アクセスが発生した後の read/write アクセスはいずれも成功する。初期の一度も read アクセスが発生していない場合が状態 S₁、一度でも read アクセスが発生した場合が状態 S₂ で、それぞれの read/write アクセスの結果と次にどの状態に移るかが示されている。

(5) write 単発エラー

write の偶発的なエラー。後続の read/write は正常に動作する。

(凡例：出力(アクセス結果)／次状態)

状態 \ イベント	S ₁	S ₂
read	正常終了／S ₁	正常終了／S ₂
write	エラー／S ₂	正常終了／S ₂

模擬故障投入直後、対象となるブロックに初めて write アクセスが発生した場合エラーとなるが、write アクセスの場合は成功となる。一度でも write アクセスが発生した後の read/write アクセスはいずれも成功する。初期の一度も write アクセスが発生していない場合が状態 S₁、一度でも write アクセスが発生した場合が状態 S₂ で、それぞれの read/write アクセスの結果と次にどの状態に移るかが示されている。

(6) read 単発無応答

read アクセスに対して、上位に応答がしばらく返らないまま一定時間後にタイムアウトを返すケース。アクセス過多などで一時的に応答できない場合を想定している。SCSI 層より上位には 5 分間のタイムアウトに見えるが、SCSI 内部では 1 分間のリトライを 5 回繰り返している。

(凡例：出力(アクセス結果)／次状態)

状態 \ イベント	S ₁	S ₂
read	エラー(タイムアウト)／S ₂	正常終了／S ₂
write	正常終了／S ₁	正常終了／S ₂

模擬故障投入直後、対象となるブロックに初めて read アクセスが発生した場合応答を返さないが、write アクセスの場合は成功となる。一度でも read アクセスが発生した後の

read/write アクセスはいずれも成功する。初期の一度も read アクセスが発生していない場合が状態 S_1 、一度でも read アクセスが発生した場合が状態 S_2 で、それぞれの read/write アクセスの結果と次にどの状態に移るかが示されている。

(7) write 単発無応答

write アクセスに対して、上位に応答がしばらく返らないまま、一定時間後にタイムアウトを返すケース。アクセス過多などで一時的に응答ができない場合を想定している。SCSI 層より上位には 5 分間のタイムアウトに見えるが、SCSI 内部では 1 分間のリトライを 5 回繰り返している。

(凡例：出力(アクセス結果)/次状態)

状態 \ イベント	S_1	S_2
read	正常終了/ S_1	正常終了/ S_2
write	エラー(タイムアウト)/ S_2	正常終了/ S_2

模擬故障投入直後、対象となるブロックに初めて write アクセスが発生した場合応答を返さないが、read アクセスの場合は成功となる。一度でも write アクセスが発生した後の read/write アクセスはいずれも成功する。初期の一度も write アクセスが発生していない場合が状態 S_1 、一度でも write アクセスが発生した場合が状態 S_2 で、それぞれの read/write アクセスの結果と次にどの状態に移るかが示されている。

(8) read/write 無応答

read/write 共に永続的に応答が無い場合。ディスク上のコントローラのハードウェアやソフトウェア故障が原因で HBA から応答が無い場合を想定している。

(凡例：出力(アクセス結果)/次状態)

状態 \ イベント	S_1
read	エラー(タイムアウト)/ S_1
write	エラー(タイムアウト)/ S_1

模擬故障投入後、対象となるブロックに対する read/write アクセスが共に応答を返さない。固定故障であり状態は一つしか存在しない。

4. まとめ

本章ではディスクドライブで発生する故障パターンについて調査を行い、それに基づいて模擬故障として投入する故障パターンを決定した。

まず SCSI、及び SATA の規格書をもとに、Linux カーネルに上がるデバイスドライバのエラーをリストアップし、その中から現実の故障で発生し得るエラーとその原因についてヒアリングを行った。また同時に、ディスク上でメディアエラーが検出された時のディスクドライブの動作についても説明した。

これを元に故障模擬ライブラリで実現すべき故障パターンとして、一時的なエラー応答 2 種類、

固定的なエラー応答 3 種類、無応答によるタイムアウト 3 種類で、合計 8 種類を決定した。合わせて故障模擬ライブラリ開発のための入力として、各故障パターンでのエラー発生時の動作を read / write アクセス毎の状態変化として表した。

今までの Linux カーネルのディスク I/O で模擬故障を発生させる取り組みでは、ディスクへの read / write アクセスの結果を確率的にエラーとするものであった。このため Linux カーネルのディスク I/O に関する模擬故障を発生させるために、実際に発生する故障パターンを調査する取り組み自体が新しいと考えられる。

ここで選定した故障パターンについては、規格上返る可能性のあるエラーの中から、製品利用した局面において発生するエラーを抽出したものであるため、現実に発生する故障パターンを十分網羅していると考ええる。模擬故障発生の実現方法については次章で説明する。

第3章 故障模擬ライブラリ

本章では、前章で説明したディスク故障を発生させる故障模擬ライブラリについて説明する。故障模擬ライブラリは Linux の SystemTap の機能を利用して実装されている。SystemTap とは、Linux 内の比較的任意の関数に対して動的にフックを投入する仕組みであり、これにより、関数の引数変更、戻り値変更、メモリ書き換え、レジスタ書き換え、各種情報採取などの操作ができる。故障模擬ライブラリは、これら SystemTap の機能を利用して、SCSI 中間層で適宜エラーが発生したように見せかけ、それを上位層に伝える。結果、SCSI より上位に位置するソフトウェア RAID のドライバに対して、RAID で故障が発生したと認識させることができる。

最初に SystemTap による模擬故障発生のためのフック投入ポイントを決めるために実施した、Linux カーネル SCSI I/O 関連の調査結果を説明する。次に調査結果を基にどのように故障模擬ライブラリを実装したかについて説明する。

1. Linux の SCSI I/O 処理

Linux の SCSI I/O に関するモジュールレベルでのブロック図を図 3 に示す。

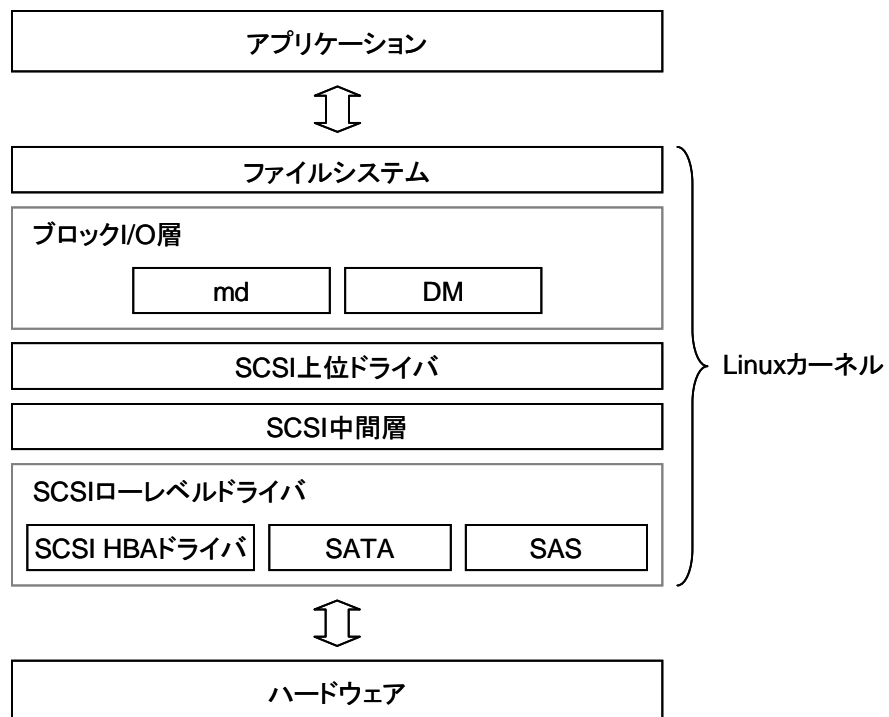


図 3 : Linux の SCSI I/O 関連のブロック図

アプリケーションからディスクへの書き込み要求を出す例を用いて、このブロック図を説明する。アプリケーションからの書き込み要求は Linux カーネルが受け取り、ファイルシステム層でアクセスする場所とアクセスするサイズが判断され、ブロック I/O 層に伝わる。ブロック I/O とは、I/O アクセスをまとまった単位で処理するタイプの I/O を意味する。ブロック I/O 層では、いつ、どのような

領域にどれだけの I/O を発行するか判断し、書込み要求を SCSI 層に伝える。md や DM 等のソフトウェア RAID を使用している場合は、ブロック I/O 層での処理がソフトウェア RAID ドライバにわたり処理される。ブロック I/O 層から伝えられた I/O 要求は、SCSI ディスクドライバなどの SCSI 上位ドライバで SCSI コマンドに変換され、SCSI 中間層を経由して SCSI ローレベルドライバにわたされる。SCSI ローレベルドライバは SCSI 中間層を利用してハードウェアとやり取りをするドライバで、主に SCSI HBA 用のドライバなどハードウェア固有のドライバからなる。SCSI ローレベルドライバにわたった SCSI コマンドは実際に HBA を経由して SCSI ディスクに送られる。SCSI 中間層は、このような様々な種類の SCSI ローレベルドライバの差分を吸収し、SCSI コマンド要求を統一的に扱い、上位層と下位層を中継する役割を持つ。

1.1. Linux の SCSI I/O 処理概要

Linux はブロック I/O 処理で SCSI を始め、SATA、SAS、USB mass-storage、Legacy IDE 等のデバイスを利用する際に、SCSI 中間層を経由する。これら SCSI 中間層を利用するデバイスのドライバは、SCSI 中間層から見ると一つの SCSI 用ローレベルドライバとして実装されている。つまり、SCSI 中間層から見ると SATA、SAS 等は、それぞれ一つの SCSI デバイスの一種として扱われる。

このため、SCSI 中間層の適切な箇所に模擬故障を投入することができれば、デバイス種別やドライバ種別に依存しない共通部分でカーネルに模擬故障投入用の処理をさせることができ、汎用性が期待できる。

1.1.1 SCSI 処理の流れ

SCSI コマンド発行後の処理概要を図 4 に示す。ここでは SCSI 中間層を使う場合共通となる `scsi_softirq_done` 以降で模擬故障を投入することを考察する。

エラー処理には以下の三つの場合が存在する。

(1) リトライ(`scsi_queue_insert`)

SCSI コマンドを再度実行する。

(2) 返ってきたエラーを上位に返す (`scsi_finish_command`)

ステータスの更新等をした後、SCSI 中間層の `done` を呼ぶ。エラーがエラーハンドラで回復できなかった場合は、`scsi_command` にエラーが記録され判定される。

(3) エラーハンドラのカーネルスレッド起動 (`scsi_eh_scmd_add`)

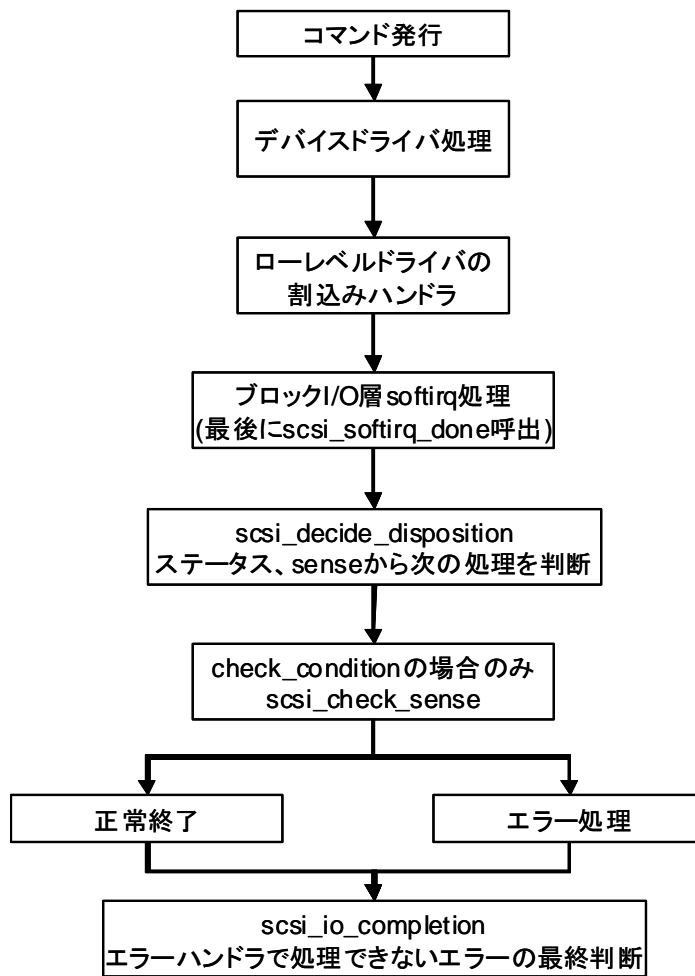


図 4 : Linux の SCSI 処理の流れ

1.1.2 SCSI 中間層のエラーハンドラ

SCSI 中間層のエラーハンドラは、エラーハンドラのキューに登録された scsi_cmnd に対して、カーネルスレッドがタイムアウト等のエラー処理を行う。scsi_cmnd をエラーハンドラのキューに登録する関数 scsi_eh_scmnd_add は、SATA や SAS を含め SCSI 中間層を使う全てのドライバが通るパスから呼び出されている。エラーハンドラスレッドの動作概要を図 5 に示す。

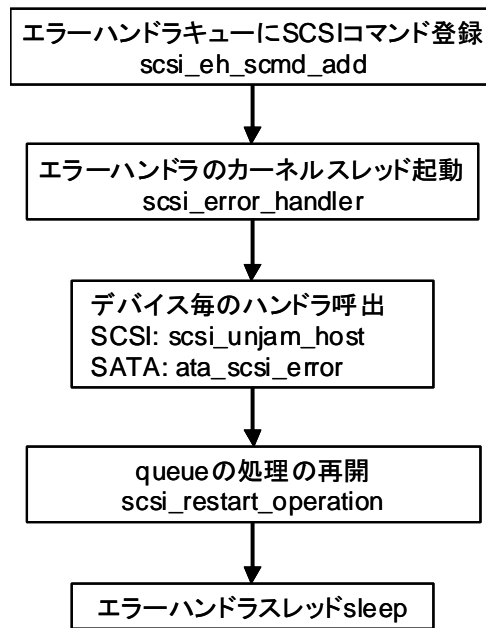


図 5 : SCSI 中間層のエラーハンドラ動作概要

例として SCSI の場合のデバイス毎のハンドラの動作を説明する。SCSI デバイスのハンドラ (scsi_unjam_host)ではエラーの程度に応じて、以下の処理を順次実行する。

1. コマンドのアボート
2. ユニットのスタート・ストップ発行
3. デバイスのリセット
4. バスリセット
5. ホストのリセット
6. デバイスを切り離す (デバイスがオフラインになるためリトライ不可)
7. リトライ可能であれば SCSI コマンドを再度キューイングする。そうでなければ後処理をして上位にエラーを返す

1.1.3 cmd/sense の判断

(1) scsi_decide_disposition() [scsi_error.c]

SCSI 中間層の関数。通常 scsi_softirq_done から呼ばれ、SCSI 中間層の done オペレーション(後述)を呼ぶか、エラーハンドラのスレッドで処理させるかに分かれる。SCSI コマンドの戻り値(host、msg、status)および必要に応じて sense (response code、sense key)で判断する。

(注意) ローレベルドライバが host_byte でタイムアウト(DID_TIMEOUT)を返す場合もある。これはエラーハンドラも返しうるが、エラーハンドラが正常動作する限りは通常返らない。

ローレベルドライバから scsi_cmnd 内のメンバ result に返される[図 6]。

driver (31~24)	host (23~16)	msg (15~8)	status (7~1)	
----------------	--------------	------------	--------------	--

図 6 : scsi_cmnd 構造体メンバ result の値

scsi_decide_disposition はステータスの値により以下のような返り値を取る。

host	msg	status	driver	返り値	備考
DID_PASSTHROUGH	*	*	*	SUCCESS	IBM power 用
DID_NO_CONNECT	*	*	*	SUCCESS	
DID_BAD_TARGET	*	*	*	SUCCESS	
DID_ABORT	*	*	*	SUCCESS	
DID_SOFT_ERROR	*	*	*	NEEDS_RETRY	但し retry count 上限まで
DID_IMM_RETRY	*	*	*	NEEDS_RETRY	
DID_REQUEUE	*	*	*	ADD_TO_MLQUEUE	mpt と aic7*xx でのみ使用
DID_ERROR	*	*	*	NEEDS_RETRY	但し retry count 上限まで
DID_BUS_BUSY	*	*	*	NEEDS_RETRY	但し retry count 上限まで
DID_PARITY	*	*	*	NEEDS_RETRY	但し retry count 上限まで
DID_TIME_OUT	*	*	*	FAILED	INQUIRY や test unit ready 中な ら SUCCESS
DID_RESET	*	*	*	SUCCESS	
DID_OK	10	*	*	FAILED	msg byte は 0 か 10 以外は未確認
それ以外	*	*	*	FAILED	
DID_OK	0	QUEUE_FULL	*	ADD_TO_MLQUEUE	
DID_OK	0	BUSY	*	ADD_TO_MLQUEUE	
DID_OK	0	GOOD	*	SUCCESS	
DID_OK	0	COMMAND_terminated	*	SUCCESS	期待値不明
DID_OK	0	TASK_ABORTED	*	SUCCESS	期待値不明
DID_OK	0	CONDITION_GOOD	*	SUCCESS	期待値不明

DID_OK	0	INTERMEDIATE_GOOD	*	SUCCESS	期待値不明
DID_OK	0	INTERMEDIATE_C_GOOD	*	SUCCESS	期待値不明
DID_OK	0	ACA_ACTIVE	*	SUCCESS	期待値不明
DID_OK	0	RESERVATION_CONFLICT	*	SUCCESS	期待値不明
DID_OK	0	CHECK_CONDITION	*	scsi_check_sense の戻り値	
DID_OK	0	それ以外	*	FAILED	

(2) scsi_check_sense の戻り値

scsi_check_sense は sensekey、ASC、及び ASCQ の値により、以下の値を返す。

response	sensekey	ASC	ASCQ	result	備考
70 or 72	NO_SENSE	*	*	SUCCESS	
70 or 72	RECOVERED_ERROR	*	*	SUCCESS	
70 or 72	ABORTED_COMMAND	*	*	NEEDS_RETRY	
70 or 72	NOT_READY	(4, 2) 以外		SUCCESS	
70 or 72	NOT_READY	4	2	FAILURE	LU not ready, initializing command required
70 or 72	UNIT_ATTENTION	(4, 2) 以外		SUCCESS	
70 or 72	UNIT_ATTENTION	4	2	FAILURE	scmd->device->allow_restart != 0 の場合
70 or 72	COPY_ABORTED	*	*	SUCCESS	未サポート
70 or 72	VOLUME_OVERFLOW	*	*	SUCCESS	未サポート
70 or 72	MISCOMPARE	*	*	SUCCESS	未サポート
70 or 72	MEDIUM_ERROR	11,13,14	*	SUCCESS	
70 or 72	MEDIUM_ERROR	11,13,14 以外	*	NEEDS_RETRY	
70 or 72	HARDWARE_ERROR	*	*	SUCCESS	但し hwerr をリトライする設定なら NEEDS_RETRY
70 or 72	ILLEGAL_REQUEST	*	*	SUCCESS	未サポート
70 or 72	BLANK_CHECK	*	*	SUCCESS	未サポート
70 or 72	DATA_PROTECT	*	*	SUCCESS	未サポート
70 or 72	上記以外	*	*	SUCCESS	未サポート

1.1.4 SCSI 中間層の done オペレーション

任意の SCSI コマンドに対して、コマンド発行のための初期化の段階で、コマンド発行後に SCSI 中間層で呼ばれる done オペレーションが設定される。特に、通常の read/write の場合、done オペレーションとして、sd_rw_intr 関数が指定される。done オペレーションが呼ばれた時点で既に一度 sense の内容が検証されているので、ここでまだエラーが残っているのは 下位層でエラーが処理できなかった場合である。scsi_cmnd の戻り値(status)および sense(sense key)で状態を判断する。

ここでの主な作業は正常に処理できたサイズ(good_bytes)を判別することであり、sense の値に従った戻り値を決定することではないため、status が 0(正常終了)の場合は request_bufflen のサイズ全て完了とみなす。

response	sensekey	result
70	MEDIUM_ERROR	sense data から処理が正常に完了した量を判断する。
70	HARDWARE_ERROR	
70	NO_SENSE	以降のレイヤでは正常終了の場合と同様に扱うため、status を 0、完了サイズを request_bufflen のサイズに書き換える。
70	RECOVERED_ERROR	
70	ILLEGAL_REQUEST	特殊なデバイス用のフックと思われる。
70	上記以外	特に何もしない。つまり status == 0 なら上記の通り処理サイズ = request_bufflen, status != 0 なら処理サイズ = 0

1.1.5 scsi_io_completion

SCSI 中間層で呼び出され、scsi_cmnd の戻り値(status)及び sense(sense key)を使って状態を判断する。途中の scsi_end_request では正常完了した量を判断し、正常系のリトライ処理を行うか、リクエストの完了でリターンする。ここに来るまでに正常系、エラーハンドラ、上位層を通っており、処理可能なエラーは全てハンドリングされていたはずなので、この時点で残っているエラーは、全ての層でハンドリングできなかったことを意味する。

response	sensekey	result
70	UNIT_ATTENTION	removable であればデバイス状態を更新してリトライ有りで scsi_end_request を呼ぶ。それ以外は scsi_requeue_command() を呼ぶ
70	ILLEGAL_REQUEST	6 バイトコマンドを使用していた場合は、デバイスに use_10_for_ms フラグを設定し scsi_request_command を呼ぶ
70	NOT_READY	(asc, ascq) = (4, 1 or 4 or 5 or 6 or 7 or 8 or 9)の場合、scsi_requeue_command を呼ぶ。

		それ以外はリトライ有りですcsi_end_request を呼ぶ
70	VOLUME_OVERFLOW	リトライ有りですcsi_end_request を呼ぶ

また、host_byte が DID_RESET の場合は scsi_requeue_command を呼ぶ他、上記以外の result, sensekey の場合は scsi_end_request が呼ばれるが、これも result != 0 ならばリトライしない。

1.2. Linux の SCSI I/O に関するデータ構造について

以下に Linux SCSI I/O に関連するデータ構造を示す。主に故障模擬ライブラリが利用するメンバについて抜粋して説明する。

1.2.1 scsi_cmnd 構造体

scsi_cmnd 構造体は SCSI 中間層から発行される SCSI コマンドを管理するための大元となるデータ構造である。各 SCSI コマンド発行に対して、scsi_cmnd 構造体が1つ割り当てられる。内部に SCSI デバイスに発行する SCSI コマンド本体(CDB)や、エラー情報を格納する sense buffer のための領域が設けられている。コマンド発行からコマンド完了まで、実行中の1つの SCSI コマンドに関する情報のみ扱うが、コマンド発行後に scsi_cmnd 構造体自身は再利用される。

型	名前	説明
struct scsi_device	scsi_device	コマンド発行対象となるデバイスの情報
void *	done	コマンド完了時に SCSI 中間層から呼ばれる done オペレーション
int	retries	SCSI 中間層に於けるこのコマンドのリトライ回数
int	allowed	SCSI 中間層に於けるリトライ許容値
int	timeout_per_command	この SCSI コマンドに定義されたタイムアウト値
unsigned char	cmnd[16]	SCSI コマンド本体
struct request *	request	この SCSI コマンドの元となる I/O リクエストに関する構造体
unsigned char	sense_buffer[96]	sense data を格納するための領域
int	result	SCSI コマンドの成功可否を判定する

1.2.2 request 構造体

request 構造体は Linux の block I/O リクエスト要求を管理する構造体である。内部に block

I/Oリクエストの最小単位である、bio 構造体のリストを含む。これは SCSI より上位となる block レイヤで定義される。故障模擬ライブラリでは、SystemTap のフックを主に SCSI 中間層で投入しており、block レイヤとの関連性は少ないため、説明は最低限にとどめ、詳細説明は割愛する。

型	名前	説明
request_queue_t *q	q	この request 構造体が所属する request queue
sector_t	sector	処理開始位置情報。対象が SCSI である場合、この情報が SCSI コマンドの LBA に設定される。
struct bio *	bio	この request に含まれる bio 構造体
struct gendisk *	rq_disk	リクエスト発行対象となるデバイスの情報

1.2.3 SCSI command block

SCSI command block(CDB)はデバイスに発行する SCSI コマンドの命令制御部分にあたる。CDB のサイズは発行するコマンドの種類によって異なるが、いずれも先頭1バイトがコマンド種別となる。Read/Write 系コマンドは指定できる LBA のサイズに従って read(6) / write(6) , read(10)/write(10), read(16)/write(16), read(32)/write(32)が SCSI 仕様で定義されている。括弧内の数字が CDB のサイズを示している。前述の scsi_cmd 構造体 cmd メンバのサイズからわかるように、Linux では read(16)/write(16)までしかサポートされていない。Read/Write コマンドでは CDB 内にアクセス開始 LBA とアクセスサイズが記述してある。記述フォーマットはコマンドによって異なる。

1.2.4 SCSI Sense data

SCSI コマンド発行に対して、正常終了しなかった際、SCSI デバイスは付加情報として sense data を返す場合がある。Linux 上では、SCSI ローレベルドライバが sense data を受け取り、scsi_cmd 構造体の該当箇所に格納される。sense_data は複数のフォーマットがあるのだが、故障模擬ライブラリではそのなかの fixed format を用いて、擬似的な故障発生時のデータを作成する。

オフセット	名前	説明
バイト 0 ビット 7	Valid	Sense data が有効であることを示す。
バイト 2 ビット 0 - 3	Sense key	発生したエラーの大まかな分類を示す。詳細は 2 章を参照。

バイト 1 2	AdditionalSense Code(ASC)	Sense code に対する細かなエラー種別を示す。
バイト 1 3	AdditionalSenseCodeQualifier(ASCQ)	Sense code, ASC に対するさらに細かなエラー種別を示す。

2. SystemTap 模擬故障フック

2.1. SystemTap による模擬故障を実現するための考察

Linux の SCSI I/O 処理を踏まえて、SCSI コマンド発行に対して模擬故障を発生させるために SystemTap で SCSI I/O 処理の何を擬似的に書き換える必要があるのかについて考察する。

故障模擬の種類は大別すると、エラー応答を模擬する場合と、無応答を模擬する場合の 2 通りがある。以降では、SystemTap スクリプトで SCSI 中間層、具体的には I/O 操作が SCSI コマンド発行という形にたどり着いた段階で、SCSI コマンドの結果を操作することを目指し、大別する 2 つの模擬故障について、詳しく分析する。

2.1.1 SCSI コマンドがエラー応答する場合の擬似

コマンドがエラー応答する動作を模擬する際に求められる動作は、コマンドの戻り値を何らかの手法でエラーした旨を上位に通知すること、及び、実際にコマンド発行が成功していないことである。

SCSI コマンド発行の戻り値は 2 種類から成り、Linux 内部の戻り値、および SCSI 仕様としての戻り値(status, sense code, sense key, asc, ascq)がある。これらの値を適宜書き換えて、上位に伝える必要がある。特に、Linux に於ける SATA コマンド発行は、上述の通り SCSI コマンド発行の延長という形になっている。戻り値についても、SATA コマンドから得られる戻り値を SATA ドライバが上述の(status, sense code, sense key, asc, ascq)に変換することにより、結局 SCSI 中間層には SCSI コマンド発行の戻り値として見えるようになっている。

SCSI コマンド発行完了後、戻り値を上位層に伝えるのは割り込み context である。割り込み context を擬似的に再現するのは極めて困難であるため、コマンドは実際に発行するが、戻り値を適宜書き換えてコマンド発行の結果エラーが返った場合を模擬し、上位に伝えるという方法が妥当と考えられる。

また、コマンドを発行するならば、結果としてデータ転送が実際に発生しうるが、期待値通りのデータ転送を発生させないようにするには、発行するコマンドのデータ転送サイズを 0 に書き換えてコマンドを発行すればよい。

2.1.2 SCSI コマンドの応答が無い場合の擬似 (擬似 timeout 発生)

SCSI コマンドの応答を返さない動作を模擬する際に求められる動作は、I/O を発行した上位層は I/O 発行要求自身は成功した状態であること、及び、実際に上位に I/O の結果が返らないことである。

また、任意の SCSI コマンドはコマンド発行の際にタイムアウト値を設定し、設定値以上の時間が経過して無応答であった場合にタイムアウト専用処理が動作するが、模擬する場合には機構が正常に機能することが求められる。

SCSI 中間層から SCSI コマンドを下位ドライバ/デバイスに発行するためには下位ドライバの `queuecommand` メソッドを呼ぶ必要がある。ここで、`queuecommand` を実際に発行しなかったにも関わらず、上位に `queuecommand` は発行したように見せかけることができたとすれば、それは上記の条件を満たしている。つまり、実際に SCSI 中間層より下位が原因でコマンドが応答を返さない場合の擬似となる。

2.2. SystemTap 模擬故障フック実装方法について

以下では先に説明した模擬故障発生のために必要な操作の実装方法について概要を説明する。

2.2.1 SCSI コマンドがエラーを応答する場合の擬似

(1) コマンド発行前

投入場所:

```
scsi_dispatch_cmd(struct scsi_cmnd *scmd)
```

説明:

任意の SCSI コマンドはこの関数を経由して SCSI ローレベルドライバの `queuecommand` オペレーションを呼び出し、コマンドを発行する。なお、前述の通り、SATA は一つの SCSI ローレベルドライバとみなせるため、この部分は SCSI/SATA で共通である。ここで、`scsi_dispatch_cmd` 関数の先頭に SystemTap によるフックを投入し、あらゆる SCSI コマンド発行をトレースして、その引数から、模擬故障投入対象となる SCSI コマンドを発行しようとしているか否かを判断する。もし、模擬故障を投入すべき SCSI コマンドを発行しようとしている場合、引数からたどれる、SCSI コマンドブロックの中の、コマンド処理サイズを0に書き換えて `queuecommand` オペレーションに渡す。これにより、コマンドそのものを発行するが、実際に DMA 転送はコマンドが意図した様に発生せず、バッファにはゴミが残る。また、発行後の処理で使用するために、書き換えを実施した `scsi_cmnd` 構造体へのポインタを保存する。

(2) コマンド発行後

投入場所:

```
scsi_decide_disposition(struct scsi_cmnd *scmd)
```

説明:

前述の通り、SCSI コマンド発行後、SCSI 中間層に於いて SCSI コマンドの戻り値は最初に `scsi_decide_disposition` で評価される。この関数が呼ばれた時点で、SCSI ローレベルドライバは `scsi_cmnd` 構造体内の `sense_buffer` に `sense` 情報を格納済みである。

そこで、`scsi_decide_disposition` 関数先頭に SystemTap のフックを投入し、コマンド発行前に保存しておいたターゲットとなる `scsi_cmnd` 構造体へのポインタと同じ `scsi_cmnd` 構造体を処理する場合を検索する。ターゲットとなる `scsi_cmnd` 構造体が見つかった場合、`sense_buffer` の内容を適宜書き換えることによって、あたかもコマンドの戻り値でエラーを検

出したかのように見せかける。この関数で設定したコマンド発行後ステータスと sense_buffer の内容が以降の SCSI 中間層及び上位層に伝わる値の源であるため、この関数内で書き換えれば、以降は矛盾無く故障が発生したと見せかけることができる。

2.2.2 SCSI コマンドの応答が無い場合の擬似 (擬似 timeout 発生)

(1) コマンド発行前

投入場所:

```
scsi_dispatch_cmd(struct scsi_cmnd *scmd)
```

説明:

任意の SCSI コマンド発行の際、各コマンドにタイマーが設定され、コマンドが完了するとタイマーが削除される。前述の通り、任意の SCSI コマンド発行の際に scsi_dispatch_cmd 関数を通るのだが、この関数から queuecommand メソッドで実際にコマンドを発行する前に、タイマーが有効か否かを判定する。SCSI コマンドは何らかの理由で正常に完了しなかった場合に再発行されうるが、以前に発行した SCSI コマンドが本当に完了していないことを確認する必要があるため、それをタイマーが有効であるか否かで判断している。この仕組みを利用し、scsi_dispatch_cmd 関数で模擬故障投入対象となる SCSI コマンドを認識した場合、そのコマンドに対するタイマーが無効である様に見せかけるため、scsi_dispatch_cmd から呼ばれる scsi_delete_timer 関数の結果を書き換える。これにより、scsi_dispatch_cmd タイマーが無効、つまりコマンドは既に完了済みと認識し、queuecommand オペレーションを実行せずに、処理を終える。上位層から見ると、コマンド発行完了に見えるため、それ以上何もせずにコマンド完了を待つ。しかし、実際は queuecommand オペレーションを発行していないため、SCSI コマンドは SCSI ローレベルドライバに発行されておらず、コマンド結果が返ることは無い。また、コマンド発行時に設定されたタイマーは有効であるため、そのうち実際にコマンドタイムアウトが発生する。また、後続の擬似無応答で使用するために、書き換えを実施した scsi_cmnd 構造体へのポインタを保存する。

(2) コマンド発行後

投入場所:

```
scsi_dispatch_cmd(struct scsi_cmnd *scmd)
```

説明:

任意の SCSI コマンドは、リトライ回数制限値と現在のリトライカウントを保持している。一般的に、SCSI コマンド発行後、正常終了しなかった場合、SCSI 中間層のエラーハンドラでそのコマンドを処理する場合がある。一旦エラーハンドラに入ったコマンドは、最終的にコマンド再発行がかかるか、あきらめてエラーを上位層に伝えるかの2択である。ここで、再発行は最大で各コマンドのリトライ回数制限値だけ発行されうる。SCSI コマンドタイムアウトはエラーハンドラで処理されるタイプの故障である。そのため、最大でリトライ回数制限値の回数だけ、コマンドが再発行されうる。逆の見方をすると、リトライ回数制限値以上の回数

のタイムアウトが発生しない限り、コマンドタイムアウトという結果が上位層に伝わらない。従って、ソフトウェア RAID のコマンドタイムアウト動作に関する評価に実施するためには、SCSI 中間層でリトライ処理を完結させてはならず、リトライ制限値に達するまで、模擬タイムアウトを発生させ続ける必要がある。そこで、`scsi_dispatch_cmd` 関数に於いて、擬似無応答を発生させている途中の `scsi_cmd` 構造体を検出した場合、そのコマンドの現在のリトライ回数を確認して、制限回数に達していない場合は、再びタイマーが無効であるように見せかけて、`queuecommand` オペレーションを妨害し、コマンドタイムアウトを発生させる。SCSI 中間層から上位層にコマンドタイムアウトが伝わるまでには、各コマンドのタイムアウト値 * 制限回数 だけ時間を要することになる。

2.2.3 故障パターン種類別実装方法について

コマンドがエラーを返す場合の模擬、無応答になる場合の模擬の基本的な実装方法は 2.2.1、2.2.2 の通りだが、第 2 章 3.1 で説明した通り、故障パターンは 8通り存在する。大きく分けて一時的なエラーと固定的なエラーの 2通りが存在するが、これは `scsi_dispatch_cmd` 関数内でエラー発生回数フラグを大域変数で持ち回り、2度目以降の模擬故障を発生させるか否かを判断する。また、各パターンに於ける故障発生契機は、`scsi_cmnd` 構造体から取得できる SCSI コマンド種別、処理開始セクタ、処理サイズから判断できる。

2.2.4 擬似的に発生させる故障種別について

2.2.1 で述べた通り、応答を返す故障を擬似的に発生させるために、故障模擬ライブラリは `sense_data` 構造体を書き換えて、擬似的に故障が発生したように見せかける。以下では書き換える値について説明する。

発生させる故障種別について

第 2 章 2 で説明した通り、調査の結果、SCSI/SATA ディスク共に SCSI ローレベルドライバより上位に見える故障は媒体不良が一般的であることが判明した。そこで、故障模擬ライブラリでは媒体不良を意味する `sense data` を作成する。ここで、2 節の冒頭で述べた通り、SATA は SCSI 中間層を利用するローレベルドライバの層と見做すことができるため、SATA ディスクの媒体不良も SATA ドライバが何らかの形で `sense data` に変換されて、SCSI 中間層に伝わる。故障模擬ライブラリとして `sense data` を一本化するため、SATA ドライバで検出された媒体不良が `sense data` に変換された値を SATA/SCSI 共通の `sense data` として採用する。

設定値について

SCSI 中間層のエラー処理として利用される値は SCSI のステータスコード、Sense key、ASC、ASCQ である。これらの値を適切な値に書き換える。詳細は以下の通り。

変更データ	値	説明
scsi_cmnd.result	2	フォーマットは 1.1.3 項参照。媒体不良なので SCSI HBA の故障は無いとみなし、 scsi status code のみ 2 (check condition)を設定する。
scsi_cmnd.sense_buffer[2]	3	Sense key。フォーマットは 1.2.4 項参照。媒体不良なので、3(medium error)を設定する。
scsi_cmnd.sense_buffer[12]	11	ASC。フォーマットは 1.2.4 項参照。11 は sense key=3 の場合、Unrecovered read error を意味する
scsi_cmnd.sense_buffer[13]	4	ASCQ。フォーマットは 1.2.4 項参照。4 は sense key=3, ASC=11 の場合に auto reallocate failed を意味する。

第4章 機能評価

本章では Linux のソフトウェア RAID 機能について最適な適用方法を調査するために、md と DM に設定・復旧・管理のために必要な機能がサポートされているか調査した結果と、それらの機能が正常に動作するか評価により検証した結果について報告する。

1. 機能評価方針

機能評価項目は以下の 3 種類に大別される。

- 設定手順
- 復旧手順
- 管理手順

各手順について、以下の環境を使用して網羅的に評価を実施した。表 14 は、ディスク種別とソフトウェア RAID 種別ごとの評価環境を一覧にしたものである。

RHEL4.5 と linux-2.6.22.6 で機能評価を実施し、障害が発生した項目については、SLES10SP1 と ML4SP2 を用いて、同様の手順で追加評価を実施した。

表 14：機能評価環境

		ソフトウェア RAID 種別	
		md	DM
ディスク種別	SATA デバイス	mdadm を用いて RAID1、RAID5、RAID6、RAID10 構築	Intel Matrix Storage Manager を用いて RAID1 で dmraid 構築と、LVM2 の RAID アレイ構築機能を用いて RAID1 構築
	SCSI デバイス	mdadm を用いて RAID1、RAID5、RAID6、RAID10 構築	LVM2 の RAID アレイ構築機能を用いて RAID1 構築

注：使用したカーネルは、RHEL4.5 とコミュニティ最新版カーネル linux-2.6.22.6 (以下 linux-2.6.22.6) の 2 種類である。コミュニティ最新版カーネルは、機能評価を開始した 2007/9/10 時点のものである。

2. ソフトウェア RAID の操作手順

最初に、インターネット上に公開されている情報等から、md、DM (dmraid)、DM(LVM2)のそれぞれの場合について、各機能評価項目のサポートの有無と、サポートされている場合はその操作手順について調査を行った。機能評価項目としては、第 1 章 2.2 の表 1 で説明した評価項目に加えて、性能評価時に不具合を検出した縮退状態での I/O 処理の項目を加えた 22 項目の評価を実施した。

表 15 で md、DM (dmraid)、DM (LVM2)それぞれに対して、機能評価項目毎に、その操作手

順に関する調査結果の概要を報告する。具体的な手順については「ソフトウェア RAID 設定手順書」を参照のこと。また本手順を使用した実機での評価結果を次節で報告するが、その際使用した構成情報や条件についてもここで併せて説明する。

なお、DM に関しては、操作手順に関するまとまった説明を発見することができておらず、DM の手順をまとめた資料としては、本報告が始めてであると思われる。

調査結果を見ると、DM (dmraid)や DM (LVM2)では未サポートの機能が少なからずあるのに対して、md では一通りの機能はそろっていることがわかる。しかし、md の機能の多くは mdadm コマンドによる操作や、/proc/mdstat ファイルの内容確認を必要としており、md に関するある程度の知識を持っていることを前提としている。ディスク故障が発生した場合の復旧についても、故障ディスク特定や復旧を行うために Linux コマンドを操作する必要がある。ハードウェア RAID 製品で見られるような、ディスクの LED の状態を見て物理的に差し替えるだけの操作と比べると複雑であり、初心者が使いこなすのは難しいと思われる。

表15 mdとDMIにおける機能評価各項目の操作手順

		md	DM (dmraid)	DM (LVM2)	備考(実機上の機能評価方法)
(1)	指定したRAIDレベルでRAIDアレイを作成する				
	RAIDアレイの作成(スペアディスク有)	mdadmコマンドの“-C”オプションを使用	未サポート	未サポート	アクティブデバイス数を4、スペアディスク数を1として、RAIDアレイを作成する
	RAIDアレイの作成(スペアディスク無)	mdadmコマンドの“-C”オプションを使用	デバイスドライバベンダ提供のツールを利用利用	pvcreate、vgcreate、lvcreate等のコマンドを利用	アクティブデバイス数を4として、RAIDアレイを作成する
	RAIDアレイの参照	/proc/mdstatファイルの内容を確認	dmraidコマンドの“-S”オプションを使用	lvdisplay -vvvを使用	
(2)	RAIDアレイを削除する				
		mdadmの“-S”オプションを使用	デバイスドライバベンダ提供のツールを利用	lvremove、vgremove、pvremove等のコマンドを使用	
(3)	RAIDアレイにディスクを追加・削除する				
	RAIDアレイへのディスク追加	mdadmコマンドの“-a”オプションを使用	未サポート。RAIDアレイを削除してから再構築する	未サポート。RAIDアレイを削除してから再構築する	
	RAIDアレイからのディスク削除	mdadmコマンドの“-r”オプションを使用	未サポート。RAIDアレイを削除してから再構築する	lvconvertコマンドの“-m”オプションを使用	
(4)	指定したデバイスに対し不良マークを付ける				
		mdadmコマンドの“-f”オプションを使用	未サポート	未サポート	
(5)	RAIDアレイのスーパーブロック内容の表示・更新・0クリアができる				
	RAIDアレイのスーパーブロックの内容表示	mdadmコマンドの“-E”オプションを使用	未サポート	vgcfgbackupコマンドを使用	
	RAIDアレイのスーパーブロックの内容更新	mdadmコマンドの“-A”オプションを使用	未サポート	vgcfgrestore -nを使用	
	RAIDアレイのスーパーブロックの0クリア	mdadmコマンドの“-–zero-superblock”オプションを使用	未サポート	vgreduceコマンドを使用	

(凡例 — : 操作手順を必要としない)

(6)	RAIDアレイのモード(読み書き、読み取り専用)切り替えができる				
	RAIDアレイの書き込み制限(読み取り専用)	mdadmコマンドの“-o”オプションを使用	未サポート	lvchangeの“-p r”オプションを使用	
	RAIDアレイの書き込み制限(読み書き可能)	mdadmコマンドの“-w”オプションを使用	未サポート	lvchangeの“-p rw”オプションを使用	
(7)	RAIDアレイ上でのイベント検出時に管理者に通知する機能を持つ				
	RAIDアレイ上でのイベント検出(メール送信)	mdadmコマンドの“-F”オプションを使用	未サポート	未サポート	RAIDアレイのアクティブデバイスに障害を起こし、メールを受信することを確認
(8)	RAIDアレイ上でのイベント検出時に特定のプログラムが実行できる				
	RAIDアレイ上でのイベント検出(プログラム実行)	mdadmコマンドの“-p”オプションを使用	未サポート	未サポート	RAIDアレイのアクティブデバイスに障害を起こし、指定したプログラムが実行されることを確認
(9)	RAIDアレイのチェックが行える				
	ディスク内容の整合性チェックが行える	未サポート	未サポート	未サポート	
(10)	ファイルシステムとして運用しているRAIDアレイ上で各種システムコールが問題なく動作する				
	openシステムコールの動作確認	—	—	—	RAIDアレイ上にext3ファイルシステムを作成・マウントし、その上でファイル操作の基本システムコールであるopen、close、read、write、lseek、fstatの動作確認を、機能評価テストプログラムを用いて実施する
	closeシステムコールの動作確認	—	—	—	
	readシステムコールの動作確認	—	—	—	
	writeシステムコールの動作確認	—	—	—	
	lseekシステムコールの動作確認	—	—	—	
	fstatシステムコールの動作確認	—	—	—	

(凡例 ー: 操作手順を必要としない)

(11)	RAIDアレイを用いて大容量ファイルシステムが作成できる				RAIDアレイ上に評価環境で構成しうる最大サイズのext3ファイルシステムを作成する
(12)	RAIDアレイ上に巨大ファイルが作成できる				上記評価で作成したRAIDアレイ上のext3の大容量ファイルシステムで、ファイルシステムの容量の9割程度となる巨大ファイルを作成する
(13)	縮退状態でのI/O処理 (機能評価の追加評価項目)				アクティブデバイス数が4のRAIDアレイを縮退状態で作成し、性能評価用テストプログラムを利用してI/O処理を実施する。実行パターンはファイルサイズ固定でI/Oサイズを変えた測定と、I/Oサイズ固定でファイルサイズを変えた測定
(14)	ディスク切り替え時に自動的に復旧処理を開始する				アクティブデバイス数が4のRAIDアレイを作成し、アクティブデバイスに故障を発生させた後、切り替え処理を行う
	ディスク切り替え後に自動的に復旧処理が実行される	スペアディスクが存在していれば、ディスク故障でRAIDアレイが縮退すると自動的に切り替えが行われる	未サポート	未サポート	
	手動で復旧処理が実行できる	mdadmコマンドの“-a”オプションを使用	未サポート	未サポート	
(15)	復旧処理の申告状況・完了したことが確認できる				アレイの復旧が完了したことを確認する
		/proc/mdstatファイルの内容を確認	未サポート	復旧の進捗状況はlvsコマンドで確認可能	

(16)	hotplug機能が使える				
(17)	hotplugでディスク追加時に復旧処理を開始する				
	hotplugでディスク追加後に自動的に復旧処理が実行される	未サポート	未サポート	未サポート	
	復旧処理を手動で実行するための手段がある	madadmコマンドの“-a”オプションを使用	未サポート	未サポート	hotplugしたディスクを縮退状態のRAIDアレイに手動で追加する
(18)	縮退状態でブート可能である				
		/etc/mdadm.confにバックアップされた構成情報に従い、縮退状態のまま起動する	縮退状態でリブートした場合は、縮退状態のまま起動する	縮退状態でリブートした場合は、縮退状態のまま起動する	アクティブデバイス数が4のRAIDアレイを作成し、縮退させ、縮退状態のままリブートする
(19)	復旧処理中にブート可能である				
(20)	復旧処理中にリブートした場合に自動的に復旧処理を再開する。				
		/etc/mdadm.confにバックアップされた構成情報に従い、リブート後に復旧処理の最初から開始される	未サポート	未サポート	アクティブデバイス数が4のRAIDアレイを作成し、縮退後、復旧処理を開始してから、リブートする
(21)	管理者が故障ディスクを特定するのをサポートする				
		/proc/mdstatファイルの内容から故障ディスクの特定ができる	dmraidコマンドの“-r”オプションを使用し、現在のRAIDアレイに残っているディスクがわかる	dmsetupコマンドにより実行中のDMドライバの状態を確認できる。LVM2の構成情報が変更された場合はlvsコマンドで撤去したディスクを特定できる	RAIDアレイの故障ディスクを確認する
(22)	RAIDの構成情報をバックアップ・リストアする機能を持つ				
	RAIDの構成情報をバックアップする	/etc/mdadm --detail --scanによりRAID構成情報をバックアップできる	dmraidコマンドの“-rD”オプションを使用する	vgcfgbackupコマンドによりバックアップ可能	RAIDアレイ作成時の構成情報をファイルにバックアップする
	構成情報バックアップからのRAIDアレイのリストア	ディスクにRAIDアレイ構成時の情報が残っていれば、mdadmの“-A”オプションでリストアできる	バックアップした構成情報をddコマンドにより書き戻す	vgcfgresotreコマンドによりリストア可能	RAIDアレイを削除した後、構成情報のバックアップからリストアする

3. 機能評価結果

機能評価を実施した結果をまとめる。

3.1. SATA デバイスの機能評価結果

SATA デバイスの機能評価結果は、表 16 のとおりである。

表 16 : SATA デバイス機能評価結果一覧

(凡例 ○ : 問題なし × : 問題あり - : 未サポートの機能または評価対象外)

		RHEL4.5						linux-2.6.22.6					
		md				DM		md				DM	
		RAI D 1	RAI D 5	RAI D 6	RAI D10	dmraid	LV M2	RAI D 1	RAI D 5	RAI D 6	RAI D10	dmraid	LV M2
(1)	RAID アレイの作成(スベアディスク有)	○	○	○	○	-	-	○	○	○	○	-	-
	RAID アレイの作成(スベアディスク無)	○	○	○	○	○	○	○	○	○	○	○	○
	RAID アレイの参照	○	○	○	○	○	○	○	○	○	○	○	○
(2)	RAID アレイの削除	○	○	○	○	○	○	○	○	○	○	○	○
(3)	RAID アレイへのディスク追加	○	○	○	○	-	-	○	○	○	○	-	-
	RAID アレイからのディスク削除	○	○	○	○	-	○	○	○	○	○	-	○
(4)	指定デバイスへの不良マーク付加	○	○	○	○	-	-	○	○	○	○	-	-
(5)	RAID アレイのスーパーブロックの内容表示	○	○	○	○	-	○	○	○	○	○	-	○
	RAID アレイのスーパーブロックの内容更新	○	○	○	○	-	○	○	○	○	○	-	○
	RAID アレイのスーパーブロックの 0 クリア	○	○	○	○	-	○	○	○	○	○	-	○
(6)	RAID アレイの書き込み制限(読み取り専用)	○	○	○	○	-	○	○	○	○	○	-	○
	RAID アレイの書き込み制限(読み書き可能)	○	○	○	○	-	○	○	○	○	○	-	○
(7)	RAID アレイ上でのイベント検出(メール送信)	○	○	○	○	-	-	○	○	○	○	-	-
(8)	RAID アレイ上でのイベント検出(プログラム実行)	○	○	○	○	-	-	○	○	○	○	-	-
(9)	RAID アレイのチェック	-	-	-	-	-	-	-	-	-	-	-	-
(10)	RAID アレイ上でのシステムコール動作確認(open0)	○	○	○	○	○	○	○	○	○	○	○	○
	RAID アレイ上でのシステムコール動作確認(close0)	○	○	○	○	○	○	○	○	○	○	○	○

	RAID アレイ上でのシステムコール動作確認(read0)	○	○	○	○	○	○	○	○	○	○	○	○
	RAID アレイ上でのシステムコール動作確認(write0)	○	○	○	○	○	○	○	○	○	○	○	○
	RAID アレイ上でのシステムコール動作確認(lseek0)	○	○	○	○	○	○	○	○	○	○	○	○
	RAID アレイ上でのシステムコール動作確認(fstat0)	○	○	○	○	○	○	○	○	○	○	○	○
(11)	RAID アレイ上での大容量ファイルシステム作成	○	○	○	○	○	○	○	○	○	○	○	○
(12)	RAID アレイ上での巨大ファイル作成	○	○	○	○	○	○	○	○	○	○	○	○
(13)	縮退状態での I/O 処理	○	○	× ※1	○	—	× ※2	○	○	○	○	—	× ※3
(14)	自動復旧	○	○	○	○	—	—	○	○	○	○	—	—
	手動復旧	○	○	○	× ※4	—	—	○	○	○	○	—	—
(15)	復旧進捗状況確認	○	○	○	○	—	○	○	○	○	○	—	○
(16)	hotplug 後自動復旧開始	—	—	—	—	—	—	—	—	—	—	—	—
(17)	hotplug 後手動復旧処理	○	○	○	○	—	—	○	○	○	○	—	—
(18)	縮退状態でのリポート	○	○	○	○	× ※5	○	○	○	○	○	○	○
(19)	復旧処理中のリポート	○	○	○	○	—	—	○	○	○	○	—	—
(20)	故障ディスク特定	○	○	○	○	○	—	○	○	○	○	○	—
(22)	構成情報バックアップ	○	○	○	○	○	○	○	○	○	○	○	○
	構成情報バックアップからの RAID アレイのリストア	○	○	○	○	— ※6	○	○	○	○	○	— ※6	○

※1…データの不整合(書き込みデータと読み込みデータの不一致)が発生。詳細は、4.1.

(3)障害①、障害②参照。

※2…ディスク抜き取り後に OS ストール発生。詳細は、4.1.(5)障害①参照。

※3…入出力エラーとファイルシステムが作成できない障害が発生。詳細は、4.1.(5)障害②参照。

※4…RAID を構成するディスクのパーティションテーブルが破壊される障害が発生。詳細は、4.1.(4)参照。

※5…再起動時に kernel panic 発生。詳細は、4.1.(6)障害①参照。

※6…RAID の構成情報を削除する機能に Intel Matrix RAID ドライバの不具合があり、構成情報がない状態を実現できないため、評価対象外。

3.2. SCSI デバイスの機能評価結果

SCSI デバイスの機能評価結果は、表 17 のとおりである。

表 17 : SCSI デバイス機能評価結果一覧

(凡例 ○ : 問題なし × : 問題あり - : 未サポートの機能または評価対象外)

		RHEL4.5					linux-2.6.22.6				
		md				DM	md				DM
		RAID 1	RAID 5	RAID 6	RAID 10	LVM2	RAID 1	RAID 5	RAID 6	RAID 10	LVM2
(1)	RAID アレイの作成(スベアディスク有)	○	○	○	○	-	○	○	○	○	-
	RAID アレイの作成(スベアディスク無)	○	○	○	○	○	○	○	○	○	○
	RAID アレイの参照	○	○	○	○	○	○	○	○	○	○
(2)	RAID アレイの削除	○	○	○	○	○	○	○	○	○	○
(3)	RAID アレイへのディスク追加	○	○	○	○	-	○	○	○	○	-
	RAID アレイからのディスク削除	○	○	○	○	○	○	○	○	○	○
(4)	指定デバイスへの不良マーク付加	○	○	○	○	-	○	○	○	○	-
(5)	RAID アレイのスーパーブロックの内容表示	○	○	○	○	○	○	○	○	○	○
	RAID アレイのスーパーブロックの内容更新	○	○	○	○	○	○	○	○	○	○
	RAID アレイのスーパーブロックの 0 クリア	○	○	○	○	○	○	○	○	○	○
(6)	RAID アレイの書き込み制限(読み取り専用)	○	○	○	○	○	○	○	○	○	○
	RAID アレイの書き込み制限(読み書き可能)	○	○	○	○	○	○	○	○	○	○
(7)	RAID アレイ上でのイベント検出(メール送信)	○	○	○	○	-	○	○	○	○	-
(8)	RAID アレイ上でのイベント検出(プログラム実行)	○	○	○	○	-	○	○	○	○	-
(9)	RAID アレイのチェック	-	-	-	-	-	-	-	-	-	-
(10)	RAID アレイ上でのシステムコール動作確認(open0)	○	○	○	○	○	○	○	○	○	○
	RAID アレイ上でのシステムコール動作確認(close0)	○	○	○	○	○	○	○	○	○	○
	RAID アレイ上でのシステムコール動作確認(read0)	○	○	○	○	○	○	○	○	○	○

	RAID アレイ上でのシステムコール動作確認(write0)	○	○	○	○	○	○	○	○	○	○
	RAID アレイ上でのシステムコール動作確認(lseek0)	○	○	○	○	○	○	○	○	○	○
	RAID アレイ上でのシステムコール動作確認(fstat0)	○	○	○	○	○	○	○	○	○	○
(11)	RAID アレイ上での大容量ファイルシステム作成	○	○	○	○	○	○	○	○	○	○
(12)	RAID アレイ上での巨大ファイル作成	○	○	○	○	○	○	○	○	○	○
(13)	縮退状態での I/O 処理	○	○	× ※1	○	○	○	○	○	○	○
(14)	自動復旧	○	○	○	○	—	○	○	○	○	—
	手動復旧	○	○	○	× ※2	— ※3	○	○	○	○	— ※3
(15)	復旧進捗状況確認	○	○	○	○	○	○	○	○	○	○
(16) (17)	hotplug 後自動復旧開始	—	—	—	—	—	—	—	—	—	—
	hotplug 後手動復旧処理	○	○	○	○	—	○	○	○	○	—
(18)	縮退状態でのリブート	○	○	○	○	○	○	○	○	○	○
(19) (20)	復旧処理中のリブート	○	○	○	○	—	○	○	○	○	—
(21)	故障ディスク特定	○	○	○	○	—	○	○	○	○	—
(22)	構成情報バックアップ	○	○	○	○	○	○	○	○	○	○
	構成情報バックアップからの RAID アレイのリストア	○	○	○	○	○	○	○	○	○	○

※1…データの不整合(書き込みデータと読み込みデータの不一致)が発生。詳細は、4.2.

(3) 障害①、障害②参照。

※2…RAID を構成するディスクのパーティションテーブルが破壊される障害が発生。詳細は、4.2.(4)参照。

※3…論理ボリュームに対して冗長性を維持したままディスク追加する機能が未サポート。詳細は、4.2.(5)参照。

4. 機能評価結果の分析

前節の機能評価結果から分析した概要をまとめる。

4.1. SATA デバイスの機能評価結果の分析

(1) md (RAID1)

機能評価項目は、全て成功した。

(2) md (RAID5)

機能評価項目は、全て成功した。

(3) md (RAID6)

障害①

設定手順の縮退状態での I/O 処理において、RHEL4.5 で以下の手順を実行した際に、I/O 対象がファイルシステム上のファイルの場合にデータの不整合が発生した(表 16※1 参照)。また linux-2.6.22.6 では、障害が発生せず正常に処理が完了することを確認した。

1. アクティブデバイス数を4とし、アレイに含まれるデバイスの1つを縮退状態にして、RAID アレイを作成
2. 作成した RAID アレイに対して性能評価用テストプログラムを利用し、ブロックデバイス及びファイルシステム上のファイルに対して I/O 処理を実施
3. O_DIRECT でファイルを open し、I/O サイズを 32M バイトに固定してシーケンシャル I/O を実施したところ、ファイルサイズが 1G バイトの場合にデータの不整合(書き込みデータと読み込みデータの不一致)が発生したことを確認

障害②

設定手順の縮退状態での I/O 処理において、RHEL4.5 で障害①と同様の手順を実行した際に、I/O 対象がブロックデバイスとファイルシステム上のファイルの両方の場合にデータの不整合が発生した(表 16※1 参照)。O_DIRECT でファイルを open し、ファイルサイズを 512M バイトに固定してシーケンシャル I/O を実施したところ、I/O サイズが 64K バイト、128K バイト、256K バイト、512K バイト、1M バイトの場合にデータの不整合(書き込みデータと読み込みデータの不一致)が発生することを確認した。また linux-2.6.22.6 では障害が発生せず、正常に処理が完了することを確認した。

(4) md (RAID10)

障害

復旧手順の手動復旧において、RHEL4.5 で、一度 RAID6 を構築してその RAID アレイを崩した直後に、以下の手順を実行した際に障害が発生した(表 16※4 参照)。linux-2.6.22.6 では障害が発生せず、正常に処理が完了することを確認した。

1. 以下のコマンドを実行し、アクティブデバイス数を4、スペアディスク数を1として、RAID10 を構築

```
# mdadm -C /dev/md0 -R -l 10 -n 4 -x 1 /dev/sd[bcdef]1
```
2. "fdisk -l" でパーティションテーブルを確認
3. /dev/sdc と/dev/sde のパーティションテーブルが破壊され、/dev/sdc と/dev/sde それぞれの先頭 512 バイトが 0 で初期化されていることを確認

(5) DM (LVM2)

未サポート

以下の各項目については、未サポートの機能である。

- ・ RAID アレイへのディスク追加
- ・ 指定デバイスへの不良マーク付加
- ・ RAID アレイ上でのイベント検出(メール送信)
- ・ RAID アレイ上でのイベント検出(プログラム実行)

- ・ RAID アレイのチェック
- ・ 自動復旧
- ・ hotplug 機能
- ・ 故障ディスク特定

障害①

設定手順の縮退状態での I/O 処理において、RHEL4.5 で以下の手順を実行した際に OS のストールが発生した(表 16※2 参照)。また linux-2.6.22.6 では OS ストールは発生しないことを確認した。

1. 以下のコマンドを実行し、オリジナルディスクとミラーディスク用にディスク4つ、ログ取得用にディスク1つの合計5つの物理ボリュームを用いて、ボリュームグループ、論理ボリュームを作成

```
# /usr/sbin/pvcreate /dev/sd[bcdef]1
# /usr/sbin/vgcreate VolGroup10 /dev/sd[bcdef]1
# /usr/sbin/lvcreate -m3 -L 2G VolGroup10
```

2. 以下のコマンドを実行し、論理ボリュームを構成するハードディスク/dev/sdd を SCSI デバイスのリストから削除

```
# echo "scsi remove-single-device 3 0 0 0" > /proc/scsi/scsi
```

3. 対象ハードディスクの物理的な抜き取りその後しばらくすると、OS がストールする。

障害②

設定手順の縮退状態での I/O 処理において、linux-2.6.22.6 で以下の手順を実行した際に I/O エラーが発生した (表 16※3 参照)。

1. 以下のコマンドを実行し、オリジナルディスクとミラーディスク用にディスク4つ、ログ取得用にディスク1つの合計5つの物理ボリュームを用いて、ボリュームグループ、論理ボリュームを作成

```
# /usr/sbin/pvcreate /dev/sd[bcdef]1
# /usr/sbin/vgcreate VolGroup10 /dev/sd[bcdef]1
# /usr/sbin/lvcreate -m3 -L 2G VolGroup10
```

2. 以下のコマンドを実行し、論理ボリュームを構成するハードディスク/dev/sde を SCSI デバイスのリストから削除

```
# echo "scsi remove-single-device 4 0 0 0" > /proc/scsi/scsi
```

3. 対象ハードディスクの物理的な抜き取り
4. 作成した RAID アレイに対して性能評価用テストプログラムを利用し、ブロックデバイス及びファイルシステム上のファイルに対して I/O 処理を実施

上記手順を実行した際、以下の 3 つの障害が発生した。

- ・ I/O 対象がブロックデバイスの場合に、O_DIRECT でファイルを open し I/O サイズを固定してシーケンシャル I/O を実施した場合と、ファイルサイズを固定してシーケンシャル I/O を実施した場合の両方で、全てのサイズで入出力エ

ラーが発生することを確認した。この時/var/log/messages にはメッセージが出力されない。また、システムコールの read 処理及び write 処理において、EIO が返されることを確認した。

- ・ I/O 対象がブロックデバイスの場合に、オープンモード指定なしと O_SYNC でファイルを open し、I/O サイズを 32M バイトに固定してシーケンシャル I/O を実施したところ、ファイルサイズが 1G バイトの場合に入出力エラーが発生したことを確認した。この時/var/log/messages に出力されたメッセージは、下記の通りである。また、システムコールの read 処理及び write 処理において EIO が返されることを確認した。

```
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 0
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 1
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 2
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 3
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 4
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 5
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 6
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 7
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 8
Oct 19 18:24:11 ipads1 kernel: Buffer I/O error on device dm-5, logical block 9
Oct 19 18:24:11 ipads1 kernel: scsi 4:0:0:0: rejecting I/O to dead device
Oct 19 18:24:33 ipads1 last message repeated 2635 times
Oct 19 18:24:33 ipads1 kernel: printk: 55 messages suppressed.
Oct 19 18:24:33 ipads1 kernel: Buffer I/O error on device dm-5, logical block 0
Oct 19 18:24:33 ipads1 kernel: Buffer I/O error on device dm-5, logical block 1
Oct 19 18:24:33 ipads1 kernel: Buffer I/O error on device dm-5, logical block 2
Oct 19 18:24:33 ipads1 kernel: Buffer I/O error on device dm-5, logical block 3
Oct 19 18:24:33 ipads1 kernel: scsi 4:0:0:0: rejecting I/O to dead device
```

- ・ I/O 対象がファイルシステム上のファイルの場合は、ファイルシステムの作成で Warning が出力され、マウントポイント/ipa にマウント出来ないという障害が発生した。ハードディスク抜き取り対象が/dev/sdb、/dev/sdd、/dev/sdf の場合は、この障害は発生しないことを確認した。また/dev/sdc はログ取得用ディスクのため未実施である。上記 3 つの障害は、RHEL4.5 では発生せず、正常に処理が完了することを確認した。実行した手順は以下の通りである。

```
# /sbin/mke2fs -j /dev/VolGroup10/lvol0
```

```
# mount -t ext3 /dev/VolGroup10/lvol0 /ipa
```

各コマンド実行時に出力されるメッセージは下記の通りである。

【mke2fs コマンドが出力したメッセージ】

```
Warning: could not read block 0: Attempt to read block from filesystem resulted
in short read
```

```
Warning: could not erase sector 0: Attempt to write block from filesystem result
```

```

ed in short write
Writing inode tables:
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information:
Warning, had trouble writing out superblocks.done
This filesystem will be automatically checked every 33 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
【mountコマンドが出力したメッセージ】
mount: 間違ったファイルシステムタイプ、不正なオプション、/dev/VolGroup10/lvol10 の
スーパーブロックが不正、或いはファイルシステムのマウントが多すぎます
【mountコマンドの/var/log/messages出力】
Oct 19 19:45:52 ipads1 kernel: EXT3-fs: unable to read superblock

```

(6) DM (dmraid)

未サポート

以下の各項目については、未サポートの機能である。

- ・ RAID アレイへのディスク追加
- ・ RAID アレイからのディスク削除
- ・ 指定デバイスへの不良マーク付加
- ・ RAID アレイのスーパーブロックの内容表示
- ・ RAID アレイのスーパーブロックの内容更新
- ・ RAID アレイのスーパーブロックの 0 クリア
- ・ RAID アレイの書き込み制限(読み取り専用)
- ・ RAID アレイの書き込み制限(読み書き可能)
- ・ RAID アレイ上でのイベント検出(メール送信)
- ・ RAID アレイ上でのイベント検出(プログラム実行)
- ・ RAID アレイのチェック
- ・ 自動復旧
- ・ 手動復旧
- ・ 復旧進捗状況確認
- ・ hotplug 機能
- ・ 復旧処理中のリポート

障害①

復旧手順の縮退状態でのリポートにおいて、RHEL4.5 で以下の手順を実行した際にカーネルがパニックする障害が発生した(表 16※5)。障害が発生しない場合に期待される動作は、縮退状態のままマシンが起動されることである。linux-2.6.22.6 では障害が発生せず、正常に処理が完了することを確認した。

1. /dev/sdb、/dev/sdc の 2 つのハードディスクを用いて RAID アレイを構築
2. 以下のコマンドを実行し、/dev/sdc を SCSI デバイスのリストから削除

```
# echo "scsi remove-single-device 2 0 0 0" > /proc/scsi/scsi
```

3. 対象ハードディスクの物理的な抜き取り
4. 縮退状態のまま、リブートするとカーネルパニックが発生し、起動が失敗する。カーネルパニック発生時に出力されるメッセージは、下記の通りである。

Kernel panic - not syncing: Fatal exception

4.2. SCSI デバイスの機能評価結果の分析

(1) md (RAID1)

機能評価項目は、全て成功した。

(2) md (RAID5)

機能評価項目は、全て成功した。

(3) md (RAID6)

障害①

設定手順の縮退状態での I/O 処理において、RHEL4.5 でデータの不整合が発生した(表 17※1 参照)。実行した手順は、4.1.(3)障害①の場合と同様である。I/O 対象がファイルシステム上のファイルの場合に、O_DIRECT でファイルを open し、I/O サイズを 32M バイトに固定してシーケンシャル I/O を実施したところ、ファイルサイズが 1G バイトの場合にデータの不整合(書き込みデータと読み込みデータの不一致)が発生することを確認した。linux-2.6.22.6 では障害が発生せず、正常に処理が完了することを確認した。

障害②

設定手順の縮退状態での I/O 処理において、RHEL4.5 で障害が発生した(表 17※1 参照)。実行した手順は、4.1.(3)障害①の場合と同様である。I/O 対象がブロックデバイスの場合とファイルシステム上のファイルの場合の両方で、O_DIRECT でファイルを open し、ファイルサイズを 512M バイトに固定してシーケンシャル I/O を実施したところ、I/O サイズが 64K バイト、128K バイト、256K バイト、512K バイト、1M バイトの場合にデータの不整合(書き込みデータと読み込みデータの不一致)が発生することを確認した。linux-2.6.22.6 では障害が発生せず、正常に処理が完了することを確認した。

(4) md (RAID10)

障害

復旧手順の手動復旧において、RHEL4.5 で障害が発生した(表 17※2 参照)。実行した手順は、4.1.(4)の障害発生時と同様であるが、SCSI デバイスでは RAID6 の構築に関係なく障害が発生した。ただし、RAID10 を構築する際に /パーティションを含む/dev/sda ディスクの/dev/sda5 デバイスを含んだため、/boot が書き換えられてしまい、再起動を試みると Linux が起動できなくなった。linux-2.6.22.6 では障害が発生せず、正常に処理が完了することを確認した。

(5) DM (LVM2)

未サポート

以下の各項目については、未サポートの機能である。

- ・ RAID アレイへのディスク追加
- ・ 指定デバイスへの不良マーク付加
- ・ RAID アレイ上でのイベント検出(メール送信)
- ・ RAID アレイ上でのイベント検出(プログラム実行)
- ・ RAID アレイのチェック
- ・ 自動復旧
- ・ hotplug 機能
- ・ 故障ディスク特定

また復旧手順の手動復旧において、RHEL4.5 と linux-2.6.22.6 の両方で、以下の手順を実行した際に未サポートの機能を確認した(表 17※3 参照)。

1. オリジナルディスクとミラーディスク用に /dev/sdb1 と /dev/sdc1 と /dev/sdd1、ログ取得用に /dev/sde1 の合計4つの物理ボリュームを用いて、ボリュームグループ、論理ボリュームを構築
2. 以下のコマンドを実行し、論理ボリュームを構成する /dev/sdd1 を SCSI デバイスのリストから削除
echo "scsi remove-single-device 0 0 3 0" > /proc/scsi/scsi
3. vgreduce コマンドを実行し、ボリュームグループの減少を LVM2 へ通知。手順3を実行するとコマンドラインにメッセージが出力される。このとき出力されるメッセージは下記の通りである。

```
/dev/dm-3: read failed after 0 of 4096 at 2147418112: 入力/出力エラーです
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
/dev/dm-3: read failed after 0 of 4096 at 2147418112: 入力/出力エラーです
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
Couldn't find all physical volumes for volume group VolGroup10.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
Couldn't find all physical volumes for volume group VolGroup10.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
Couldn't find all physical volumes for volume group VolGroup10.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
Couldn't find all physical volumes for volume group VolGroup10.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
Couldn't find all physical volumes for volume group VolGroup10.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
Couldn't find all physical volumes for volume group VolGroup10.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
```

```

Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
Couldn't find device with uuid 'Src91p-wGGW-8HLS-aj0j-Kzx7-pFr1-50MEPg'.
/dev/dm-3: read failed after 0 of 4096 at 0: 入力/出力エラーです
WARNING: Bad device removed from mirror volume, VolGroup10/lvol0
Wrote out consistent volume group VolGroup10

```

4. 対象ハードディスクの物理的な抜き取り
5. 別のハードディスクを物理的な指し込み
6. 以下のコマンドを実行し、対象のハードディスクを SCSI デバイスのリストに追加
echo "scsi add-single-device 0 0 3 0" > /proc/scsi/scsi
7. 追加したハードディスクで物理ボリュームを作成
8. 物理ボリュームを作成済みのボリュームグループに追加
9. 論理ボリュームを再作成すると、コマンドラインに機能が未サポートという旨のメッセージが出力される。出力されるメッセージは以下の通りである。機能がサポートされている場合に期待される動作は、論理ボリュームを再作成できることである。

Adding mirror images is not supported yet.

ただし、代替の手段として以下のように一度ミラーボリュームを解除してから再度ミラーボリュームを構築することで、追加可能である。

1. 一時的にリニアに構成変更 (lvconvert -m 0)
2. リニア構成を 3 ミラー構成に変更 (lvconvert -m 2)

5. 機能評価の追加評価結果

機能評価において障害が発生した評価項目のみ、同様の手順で SLES10SP1 と ML4SP2 での追加評価を実施した。その結果をまとめる。

5.1. SATA デバイスの機能評価の追加評価結果

SATA デバイスの機能評価の追加評価結果は、表 18 のとおりである。

表 18 : SATA デバイス機能評価の追加評価結果一覧

(凡例 ○ : 問題なし × : 問題あり - : 未サポートの機能または評価対象外)

	SLES10SP1						ML4SP2					
	md				DM		md				DM	
	RAI D 1	RAI D 5	RAI D 6	RAI D10	dmraid	LV M2	RAI D 1	RAI D 5	RAI D 6	RAI D10	dmraid	LV M2
縮退状態での I/O 処理	-	-	○	-	-	× ※2	-	-	× ※1	-	-	× ※2
手動復旧	-	-	-	○	-	-	-	-	-	× ※3	-	-
縮退状態でのリポート	-	-	-	-	○	○	-	-	-	-	○	○

※1…データの不整合(書き込みデータと読み込みデータの不一致)が発生。詳細は、6.1. (1)障害①、障害②参照。

※2…入出力エラーとファイルシステムが作成できない障害が発生。RHEL4.5 で発生したディスク抜き取り後の OS ストールは発生せず。詳細は、6.1. (3)参照。

※3…RAID を構成するディスクのパーティションテーブルが破壊される障害が発生。詳細は、6.1. (2)参照。

5.2. SCSI デバイスの機能評価の追加評価結果

SCSI デバイスの機能評価の追加評価結果は、表 19 のとおりである。

表 19 : SCSI デバイス機能評価の追加評価結果一覧

(凡例 ○ : 問題なし × : 問題あり - : 未サポートの機能または評価対象外)

	SLES10SP1					ML4SP2				
	md				DM	md				DM
	RAID 1	RAID 5	RAID 6	RAID 10	LVM2	RAID 1	RAID 5	RAID 6	RAID 10	LVM2
縮退状態での I/O 処理	-	-	○	-	-	-	-	× ※1	-	-
手動復旧	-	-	-	○	-	-	-	-	× ※2	-

※1…データの不整合(書き込みデータと読み込みデータの不一致)が発生。詳細は、6.2. (1)障害①、障害②参照。

※2…RAID を構成するディスクのパーティションテーブルが破壊される障害が発生。詳細は、6.2. (2)参照。

6. 機能評価の追加評価結果の分析

5の機能評価の追加評価結果から分析した概要をまとめる。

6.1. SATA デバイスの機能評価の追加評価結果の分析

(1) md (RAID6)

障害①

設定手順の縮退状態での I/O 処理の追加評価において、ML4SP2 で障害が発生した(表 18※1 参照)。実行した手順は、4.1.(3)の障害①と同様である。I/O 対象がファイルシステム上のファイルの場合、O_DIRECT でファイルを open し、I/O サイズを 32M バイトに固定してシーケンシャル I/O を実施したところ、ファイルサイズが 1G バイトの場合にデータの不整合(書き込みデータと読み込みデータの不一致)が発生することを確認した。SLES10SP1 では、障害が発生せず正常に処理が完了することを確認した。

障害②

設定手順の縮退状態での I/O 処理の追加評価において、ML4SP2 で障害が発生した(表 18※1 参照)。実行した手順は、上記の障害①と同様である。4.1.(3)障害②とは異なり、I/O 対象がファイルシステム上のファイルの場合のみ、O_DIRECT でファイルを open し、ファイルサイズを 512M バイトに固定してシーケンシャル I/O を実施したところ、I/O サイズが 64K バイト、128K バイト、256K バイト、512K バイト、1M バイトの場合にデータの不整合(書き込みデータと読み込みデータの不一致)が発生することを確認した。I/O 対象がブロックデバイスの場合、障害が発生せず正常に処理が完了することを確認した。また、SLES10SP1 では障害が発生せず、正常に処理が完了することを確認した。

(2) md (RAID10)

障害

復旧手順の手動復旧の追加評価において、ML4SP2 で障害が発生した(表 18※3 参照)。一度 RAID6 を構築してその RAID アレイを崩した直後に、4.1.(4)と同様の手順を実行した。/dev/sdc と /dev/sde のパーティションテーブルが破壊され、/dev/sdc と /dev/sde それぞれの先頭 512 バイトが 0 で初期化されていることを確認した。SLES10SP1 では、障害が発生せず正常に処理が完了することを確認した。

(3) DM (LVM2)

障害

設定手順の縮退状態での I/O 処理の追加評価において、SLES10SP1 と ML4SP2 の両方で障害が発生した(表 18※2 参照)。実行した手順は、4.1.(5)の障害②と同様である。SLES10SP1 では、4.1.(5)の障害②で発生した入出力エラーとファイルシステムが作成できない障害全てが発生することを確認した。出力されるメッセージも同様である。ML4SP2 では、I/O 対象がブロックデバイスの場合に O_DIRECT でファイルを open し、ファイルサイズを固定してシーケンシャル I/O を実施した場合に、全てのサイズで入出力エラーが発生することを確認した。これ以外の障害は発生せず、正常に処理が完了することを確認した。また、SLES10SP1 と ML4SP2 の両方で、RHEL4.5 において発生したディスク抜き取り後の

OS ストールは発生しないことを確認した。

6.2. SCSI デバイスの機能評価の追加評価結果の分析

(1) md (RAID6)

障害①

設定手順の縮退状態での I/O 処理の追加評価において、ML4SP2 で障害が発生した(表 19※1 参照)。実行した手順は、6.1.(1)の障害①と同様である。I/O 対象がファイルシステム上のファイルの場合に、O_DIRECT でファイルを開き、I/O サイズを 32M バイトに固定してシーケンシャル I/O を実施したところ、ファイルサイズが 1G バイトの場合にデータの不整合(書き込みデータと読み込みデータの不一致)が発生することを確認した。SLES10SP1 では障害が発生せず、正常に処理が完了することを確認した。

障害②

設定手順の縮退状態での I/O 処理の追加評価において、ML4SP2 で障害が発生した(表 19※1 参照)。実行した手順は、6.1.(1)の障害①と同様である。I/O 対象がブロックデバイスとファイルシステム上のファイルの両方の場合に、O_DIRECT でファイルを開き、ファイルサイズを 512M バイトに固定してシーケンシャル I/O を実施したところ、I/O サイズが 64K バイト、128K バイト、256K バイト、512K バイト、1M バイトの場合にデータの不整合(書き込みデータと読み込みデータの不一致)が発生することを確認した。SLES10SP1 では障害が発生せず、正常に処理が完了することを確認した。

(2) md (RAID10)

障害

復旧手順の手動復旧の追加評価において、ML4SP2 で障害が発生した(表 19※2 参照)。実行した手順は、6.1.(2)と同様であるが、SCSI デバイスでは RAID6 の構築に関係なく障害が発生した。/dev/sdc と /dev/sde のパーティションテーブルが破壊され、/dev/sdc と /dev/sde それぞれの先頭 512 バイトが 0 で初期化されていることを確認した。SLES10SP1 では障害が発生せず、正常に処理が完了することを確認した。

7. まとめ

機能評価の分析結果をまとめる。

機能評価項目では、RAID アレイの設定、復旧、運用管理を行うためのコマンドについてインターネット上に公開されている情報などから調査を行い、md、DM (dmraid)、DM (LVM2)のそれぞれの場合について、各機能のサポートの有無と、サポートされている場合については操作手順の調査を行った。各操作に関する詳細な手順は「ソフトウェア RAID 設定手順書」に説明し、本報告書では操作方法に対する調査結果の概要についてだけ報告した。

DM については、設定手順をまとめて説明した資料を発見することはできていないため、本調査の成果である設定手順書は DM を利用するユーザにとっては極めて有益な情報とな

ると考える。

また、調査した機能が正しく使用できるか、実機上で確認を行った。RAID アレイの動作確認についても、ファイル操作の基本となるシステムコールを評価している。また、ファイルシステム上のファイルやブロックデバイスに対して、I/O サイズやファイルサイズ、ファイル open 時の指定オプションを変えながら、網羅的に評価を実施した。

SATA デバイスでの機能評価項目数と障害件数を表 20 に、SCSI デバイスでの機能評価項目数と障害件数を表 21 にそれぞれ示す。

表 20 : SATA デバイス機能評価項目数と障害件数

(障害件数 / 評価項目数)

	md				DM	
	RAID1	RAID5	RAID6	RAID10	dmraid	LVM2
RHEL4.5	0 / 32	0 / 32	1 / 32	1 / 32	1 / 14	1 / 22
linux-2.6.22.6	0 / 32	0 / 32	0 / 32	0 / 32	0 / 14	1 / 22

表 21 : SCSI デバイス機能評価項目数と障害件数

(障害件数 / 評価項目数)

	md				DM
	RAID1	RAID5	RAID6	RAID10	LVM2
RHEL4.5	0 / 32	0 / 32	1 / 32	1 / 32	0 / 22
linux-2.6.22.6	0 / 32	0 / 32	0 / 32	0 / 32	0 / 22

- md の RAID1、RAID5 については、SATA デバイスと SCSI デバイスの両方で、RHEL4.5、linux-2.6.22.6 とともに障害が発生せず機能評価項目が成功した。上の表の通り、各々316 項目ずつ評価を実施し障害が発生しなかったことから、機能上は問題無いものと考えられる。
- md の RAID6 については、SATA デバイスと SCSI デバイスの両方で、RHEL4.5 において、縮退状態での I/O 処理で障害が発生した。障害が発生した縮退状態での I/O 処理について、SLES10SP1 と ML4SP2 を用いて同様の手順で再度評価を行ったところ、ML4SP2 でのみ障害が発生した。linux-2.6.22.6 と SLES10SP1 では同様の障害が発生せず、正常に処理を実施できることを確認した。linux-2.6.22.6 と SLES10SP1 は、RHEL4.5 と ML4SP2 よりもカーネルバージョンが新しいことから、この障害は解決されたものと考えられる。また、他に障害が検出されなかったことから、コミュニティ最新版カーネルでは機能上問題は無いものと考えられる。
- md の RAID10 については、SATA デバイスと SCSI デバイスの両方で、RHEL4.5 において、手動復旧の評価で障害が発生した。障害が発生した手動復旧の評価については、SLES10SP1 と ML4SP2 を用いて同様の手順で再度評価を行ったところ、ML4SP2 でのみ障害が発生した。linux-2.6.22.6 と SLES10SP1 では同様の障害が発生せず、正常に処理を実施できることを確認した。md の RAID6 の障害の場合と同様に、linux-2.6.22.6 と SLES10SP1 は、RHEL4.5 と ML4SP2 よりもカーネルバージョンが新しいことから、この障害は解決されたものと考えられる。また、他に障害が検出されなかったことから、コミュニティ最新版カーネルで

は機能上問題は無いものと考えられる。

- DM の dmraid については、RHEL4.5 において、縮退状態でのリブートの評価で障害が発生した。障害が発生した縮退状態でのリブートについては、SLES10SP1 と ML4SP2 を用いて、同様の手順で再度評価を行った。linux-2.6.22.6 と SLES10SP1、ML4SP2 のいずれの OS でも障害が発生しなかったことから、RHEL4.5 の縮退状態でのリブートは品質が不十分である。
- DM の LVM2 については、RHEL4.5 において、縮退状態での I/O 処理で障害が発生した。この障害は、linux-2.6.22.6 でも発生することを確認した。障害が発生した縮退状態での I/O 処理については、SLES10SP1 と ML4SP2 を用いて、同様の手順で再度評価を行い、両方の OS で障害が発生することを確認した。縮退状態での I/O 処理については、さまざまな条件で障害が発生しており、品質に問題があると言える。DM の LVM2 における縮退状態での I/O 処理は、品質が不十分である。

第5章 性能評価

本章では md と DM 上で、性能評価用のテストプログラムを用いて、通常運用中、縮退状態、復旧処理中の三つの場合について行った性能測定と、各 RAID レベルで無負荷時と負荷をかけた時の復旧時間の測定を行った結果について報告する。

1. 性能評価方針

ソフトウェア RAID の様々な状態での性能を検証するため、以下の項目について性能測定と考察を行った。

- 通常運用の RAID に対する I/O 性能について RAID 構築なしと比較する
- 縮退状態での I/O 性能について通常運用時と比較する
- 復旧処理中の I/O 性能について通常運用時と比較する
- 復旧処理に要する時間を測定する

1.1. 性能評価内容

性能評価の運用種別は以下の 3 種類に大別される。

- (1) 通常運用：RAID が正常に運用している状態
 - 各 RAID レベルの通常運用時に RAID の性能を測定する。
- (2) 縮退状態：RAID からディスクの 1 つを抜き取った状態
 - 各 RAID レベルの縮退状態で RAID の性能を測定する。
- (3) 復旧処理中：縮退している RAID をスペアディスクで復旧処理中の状態
 - 各 RAID レベルの復旧処理中に RAID の性能を測定する。
 - 無負荷状態での各 RAID レベルの復旧時間を測定する。
 - 負荷状態での各 RAID レベルの復旧時間を測定する。

それぞれの運用種別について、表 22 に示す性能評価環境を使用して評価を実施した。

表 22：性能評価環境

		ソフトウェア	
		md	DM
種別 ディスク	SATA デバイス	mdadm を用いて RAID1、RAID5、RAID6、RAID10 構築	LVM2 の RAID 構築機能を用いて RAID1 構築
	SCSI デバイス	mdadm を用いて RAID1、RAID5、RAID6、RAID10 構築	LVM2 の RAID 構築機能を用いて RAID1 構築

使用したカーネルは、Red Hat Enterprise Linux 4 Update 5 (以下 RHEL4.5)とコミュニティ最新版カーネル linux-2.6.22.6 (以下 linux-2.6.22.6)の 2 種類である。linux-2.6.22.6 版カーネルは、性能評価を開始した 2007/9/10 時点のものである。

1.2. 性能評価環境

性能評価は、コミュニティ最新カーネル、及び RHEL4.5 上で実施した。

1.3. 性能評価項目

表 22 の環境を使用して、通常運用、縮退状態、復旧処理中の性能評価、および復旧処理時間測定を実施した。ここでは、具体的な評価方法を説明する。

1.3.1 性能測定プログラムによる I/O 処理性能測定

性能測定では、性能評価用テストプログラムを使用して、通常運用、縮退状態、復旧処理中のそれぞれの状態でブロックデバイスおよびファイルシステムのファイルに対してシーケンシャルに read/write を行い、read/write の性能(Mbyte/sec)の測定を実施した。

ファイルサイズ、I/O サイズ、オープンモードについては以下の通りである。

- ファイルサイズ：512MByte 固定
- I/O サイズ：64K、128K、256K、512K、1M、2M、4M、8M、16M、32M、64M、128M、256Mbyte
- オープンモード：指定なし、O_DIRECT、O_SYNC

1.3.2 復旧処理時間測定

復旧処理時間測定では、スペアディスクを含む RAID を構築し、RAID ディスクのひとつに不良マークを付与して復旧処理状態にし、復旧処理が完了する時間の測定を実施した。

ここでは、無負荷時、負荷時の 2 パターンの測定を実施した。

2. 性能測定プログラムによる評価結果

read/write 性能を測定するための性能評価用テストプログラムを開発し、それを用いて性能測定した結果を示す。ここでは代表的な測定項目に関して、その結果を掲載し分析を行う。

2.1. 通常運用時の RAID 未構築時と各 RAID レベルの性能測定結果

SATA ディスク上での RAID なしの時の性能測定結果と、各 RAID レベルでの通常運用時のブロックデバイスアクセス性能測定を図 7 に、ファイルアクセス性能測定結果を図 8 に示す。また SCSI ディスク上での RAID なしの時の性能測定結果と、各 RAID レベルの通常運用時のブロックデバイスアクセス性能結果を図 9 に、ファイルアクセス性能測定結果を図 10 に示す。キャッシュヒット時ではなく、ディスクアクセス時の性能を測定するため、ここでは O_DIRECT を指定してアクセスを行った結果を示す。

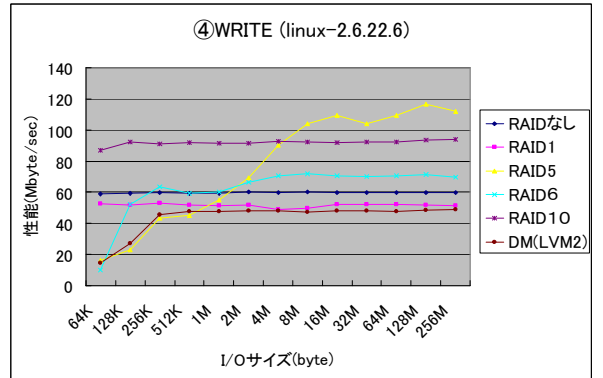
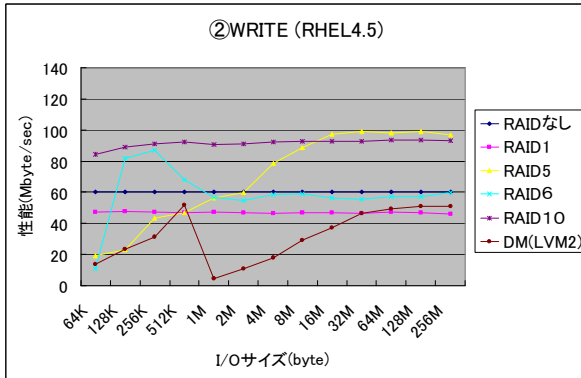
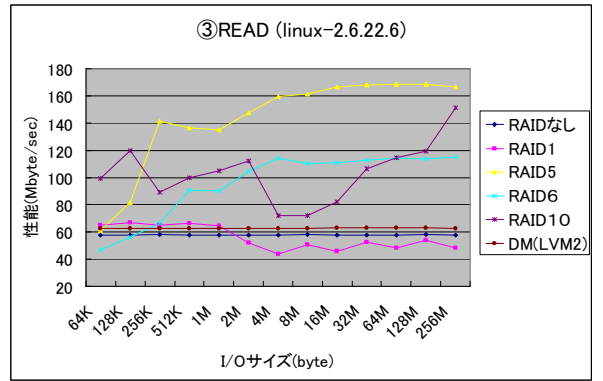
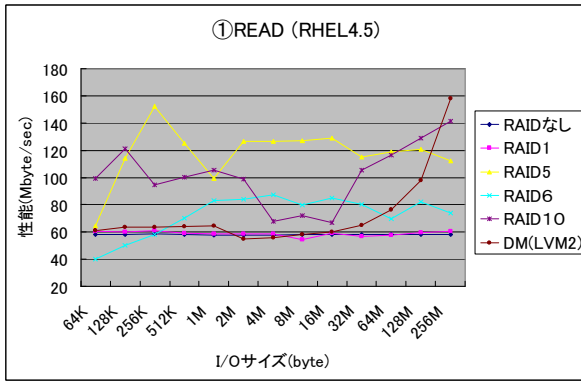


図 7：通常運用時のブロックデバイスアクセス性能測定結果 (SATA)

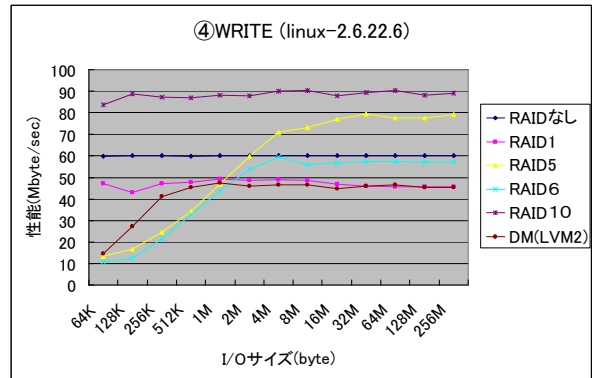
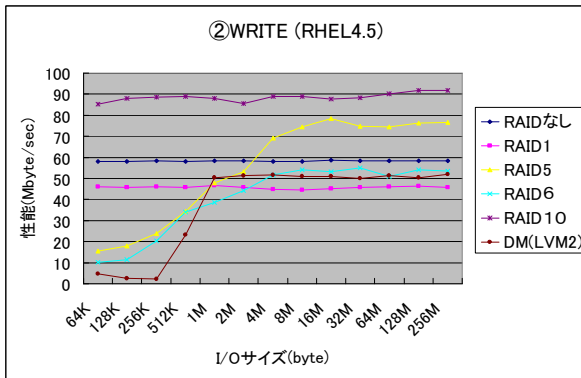
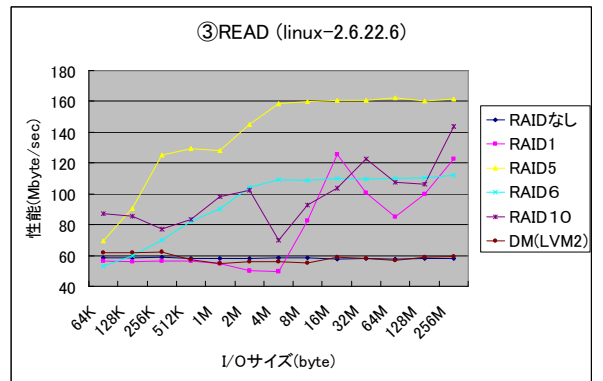
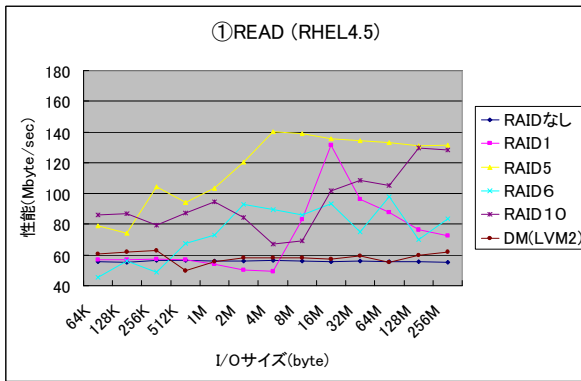


図 8：通常運用時のファイルアクセス性能測定結果 (SATA)

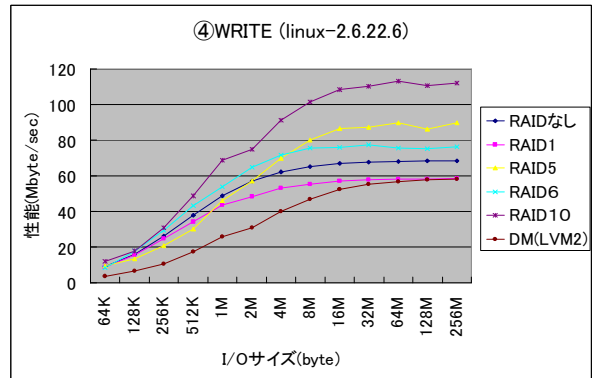
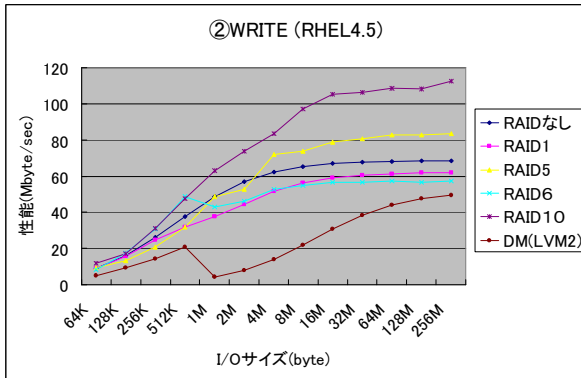
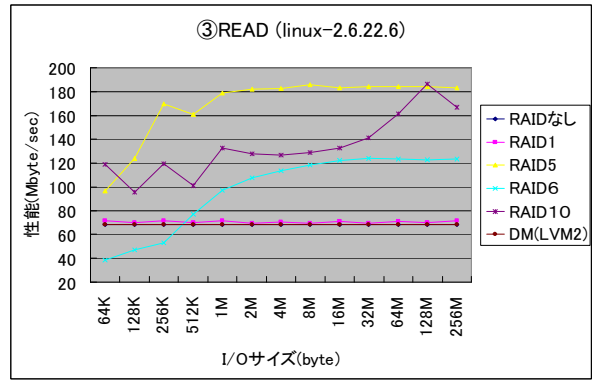
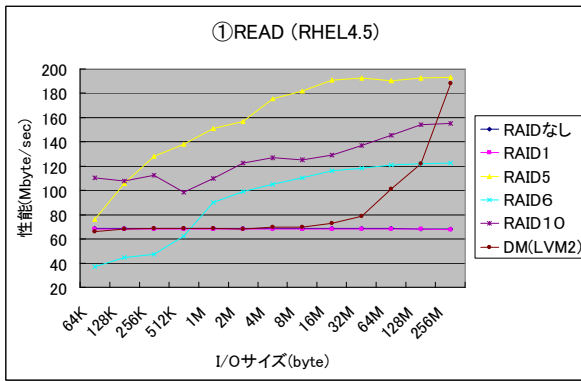


図 9 : 通常運用時のブロックデバイスアクセス性能測定結果 (SCSI)

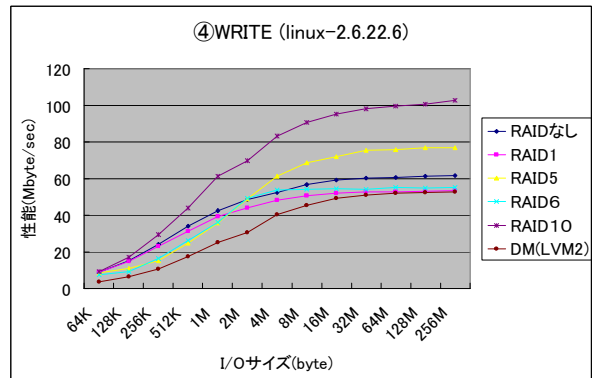
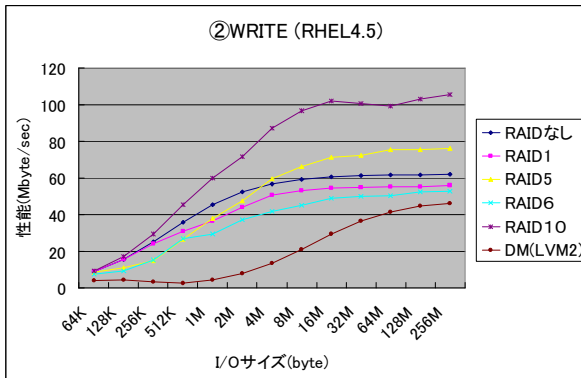
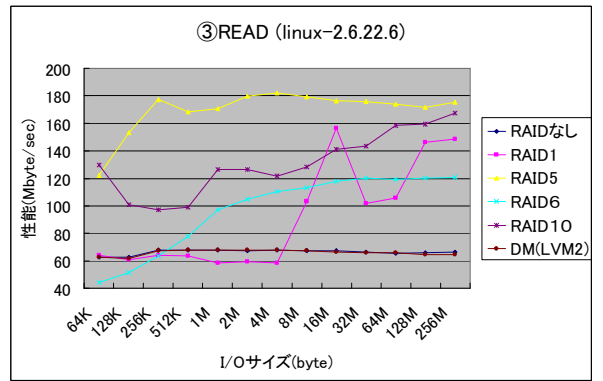
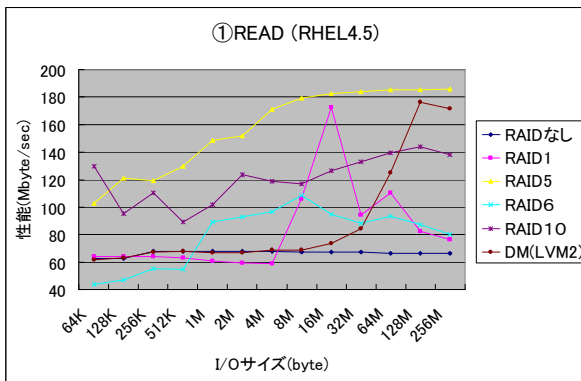


図 10 : 通常運用時のファイルアクセス性能測定結果 (SCSI)

(1) md (RAID1)の性能

ブロックデバイスアクセス

read では RAID なしの時と同等の性能が出ている。

write では RAID なしの時より若干悪い性能となっているが、これは RAID1 では write 時に同一のデータを複数のディスクに書き出すためのオーバーヘッドが現れているためと考えられる。

ファイルアクセス

read では I/O サイズ 8MB 以下では RAID なしの時と同等の性能がでている。一方、今回の測定では I/O サイズ 16MB 以上で性能向上が見られるが、その原因は不明である。

write では RAID なしの時より若干悪い性能となっているが、これは RAID1 では write 時に同一のデータを複数のディスクに書き出すためのオーバーヘッドが現れているためと考えられる。

(2) md (RAID5)の性能

ブロックデバイスアクセス

read では RAID なしの時と比べ、ほぼ同等から最大 3 倍弱の性能向上が得られている。RAID5 では複数のディスクにストライピングしてデータを格納しているため、read のスループットが向上しているためである。

write では I/O サイズが小さい時は RAID なしの時と比べて性能が低下するが、これは write ペナルティにより性能が低下しているためである。一方、I/O サイズが大きい時は 15%～30%の性能向上が得られるが、これは write によりパリティデータを含むブロック全体の書き換えが発生しているものの、複数のディスクにストライピングしてデータを格納するため、write のスループットが向上しているためである。

ファイルアクセス

read では RAID なしの時と比べ、ほぼ同等から最大 3 倍弱の性能向上が得られている。RAID5 では複数のディスクにストライピングしてデータを格納しているため、read のスループットが向上しているためである。

write では I/O サイズが小さい時は RAID なしの時と比べて性能が低下するが、これは write ペナルティにより性能が低下しているためである。一方、I/O サイズが大きい時は 15%～30%の性能向上が得られるが、これは write によりパリティデータを含むブロック全体の書き換えが発生しているものの、複数のディスクにストライピングしてデータを格納するため、write のスループットが向上しているためである。

(3) md (RAID6)の性能

ブロックデバイスアクセス

read では I/O サイズが小さい時は RAID なしの時と比べて性能低下が見られるものの、最大 1.5 倍までの性能向上が得られている。RAID6 では複数のディスクにストライピングしてデータを格納しているため、read のスループットが向上するためである。

write では I/O サイズが小さい時は RAID なしの時と比べて性能が低下するが、これは

write ペナルティにより性能が低下しているためである。一方、I/O サイズが大きい時は若干の性能向上が得られるが、これは write によりパリティデータを含むブロック全体の書き換えが発生しているものの、複数のディスクにストライピングしてデータを格納するため、write のスループットが向上しているためである。

ファイルアクセス

read では I/O サイズが小さい時は RAID なしの時と比べて性能低下が見られるものの、最大 2 倍弱の性能向上が得られている。RAID6 では複数のディスクにストライピングしてデータを格納しているため、read のスループットが向上するためである。

write では I/O サイズが小さい時は RAID なしの時と比べて性能が低下するが、これは write ペナルティにより性能が低下しているためである。一方 I/O サイズが大きくなるに従い性能低下は改善する。これは複数のディスクにストライピングしてデータを格納するため write のスループットが向上するものの、RAID5 と比べるとパリティデータの量が多く、オーバヘッドが大きいためと考えられる。

(4) md (RAID10)の性能

ブロックデバイスアクセス

read では最大 2.5 倍までの性能向上が得られる。これは RAID10 では 2 台のディスクにストライピングしてデータを格納するためにスループットが向上するため考えられる。

write では最大 1.5 倍までの性能向上が得られる。これは RAID10 では 2 台のディスクにストライピングしてデータを格納するためにスループットが向上するため考えられる。

ファイルアクセス

read では最大 2 倍強の性能向上が得られる。これは RAID10 では 2 台のディスクにストライピングしてデータを格納するためにスループットが向上するため考えられる。

write では最大 1.5 倍強の性能向上が得られる。これは RAID10 では 2 台のディスクにストライピングしてデータを格納するためにスループットが向上するため考えられる。

(5) DM (LVM2)

ブロックデバイスアクセス

read では機能的にほぼ同等な md (RAID1)と同様の傾向を示すはずであり、I/O サイズが小さい範囲ではほぼ同等の性能が測定されている。ただし、RHEL4.5 上の測定では I/O サイズが 4MB を超えたあたりから性能の上昇が見られる。RHEL4.5 上の DM (LVM2)は性能評価プログラムの読み込んだデータの整合性チェックはパスしているものの、動作上のセマンティクスで一部正確でない可能性がある。ただしこの現象はコミュニティカーネルでは見られないため、その後修正があったものと考えられる。

write も機能的にほぼ同等な md (RAID1)と同様の傾向を示すはずであるが、RHEL4.5 上では I/O サイズ 1MB 以上で大きな性能低下がみられる。この現象はコミュニティカーネルでは見られないため、修正があったものと考えられる。

ファイルアクセス

read では機能的にほぼ同等な md (RAID1)と同様の傾向を示しているが、RHEL4.5 上の

SCSI ディスクでの測定結果においてブロックデバイスアクセスの場合と同様に、I/O サイズが4MBを超えたあたりから性能の上昇が見られる。RHEL4.5 上の DM (LVM2)は性能評価プログラムの読み込んだデータの整合性チェックはパスしているものの、動作上のセマンティクスで一部正確でない可能性がある。ただしこの現象はコミュニティカーネルでは見られないため、その後修正があったものと考えられる。

write も機能的にはほぼ同等な md (RAID1)と同様の傾向を示すはずであるが、小～中 I/O サイズの領域で RHEL4.5 に比べるとコミュニティカーネルの方が性能は改善しているものの、md (RAID1)と比較すると性能が低い傾向がある。DM に比べ以前から開発の進んでいた md の最適化が進んでいるためと考えられる。

2.2. 縮退状態での各 RAID レベルの通常運用時の性能測定結果

SATA ディスク上の各 RAID レベルの縮退状態でのブロックデバイスアクセス性能測定を図 11 に、ファイルアクセス性能測定結果を図 12 に示す。また SCSI ディスク上の各 RAID レベルの縮退状態でのブロックデバイスアクセス性能結果を図 13 に、ファイルアクセス性能測定結果を図 14 に示す。ディスクアクセス時の性能を測定するため、ここでは O_DIRECT を指定してアクセスを行った結果を示す。比較のために RAID なしの時の性能測定結果も合わせて示す。

(1) md (RAID1)

ブロックデバイス及びファイルアクセスに対する read、write それぞれのケースにおいて、RHEL4.5 とコミュニティカーネルのどちらの場合でも、通常運用時の結果とほぼ同様の結果が得られている。

(2) md (RAID5)

ほぼ全てのケースにおいて通常運用時の結果とほぼ同様の結果が得られている。コミュニティカーネルの read アクセスの場合と、SCSI 上の RHEL4.5 のブロックデバイスに対する read アクセスの場合のみ性能低下が観測されているが、原因は不明である。

(3) md (RAID6)

機能評価で説明した通り、RHEL4.5 の md (RAID6)は縮退状態では正常動作していないため、性能測定は行わなかった。

(4) md (RAID10)

ほぼ全てのケースにおいて通常運用時の結果とほぼ同様の結果が得られている。ただし SCSI 上の RHEL4.5 の場合のみ縮退時に read、write アクセス性能の低下が観測されているが、原因は不明である。RHEL4.5 では RAID10 構築時にパーティションが壊れる障害が発生しており正しく測定できていない可能性がある。

(5) DM (LVM2)

測定できた項目に関しては、通常運用時の結果とほぼ同様の結果が得られている。

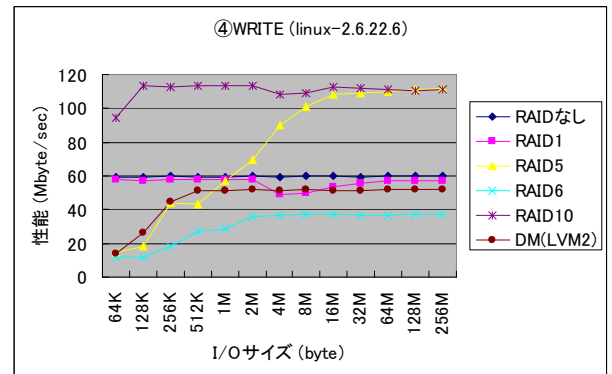
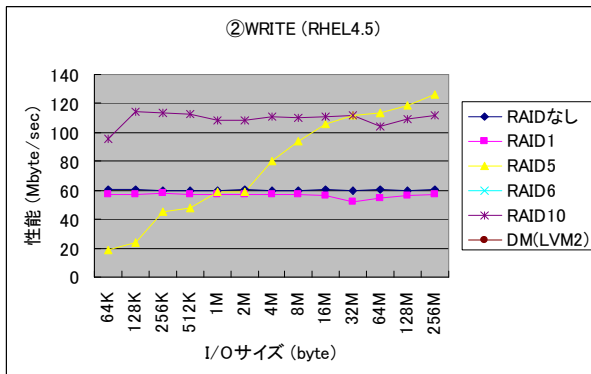
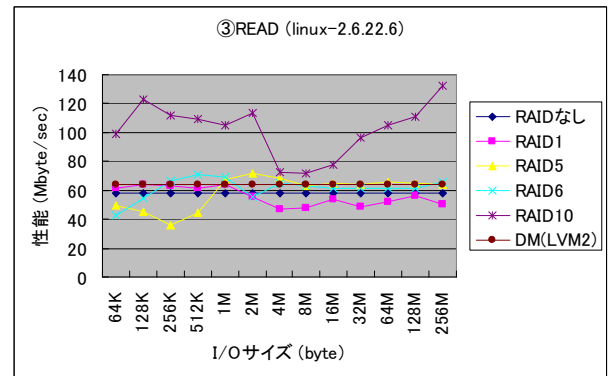
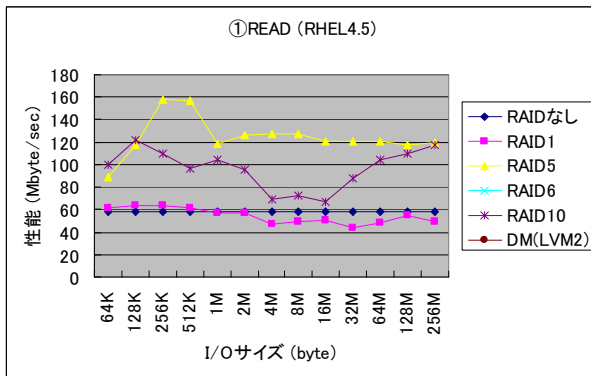


図 11：縮退状態でのブロックデバイスアクセス性能測定結果 (SATA)

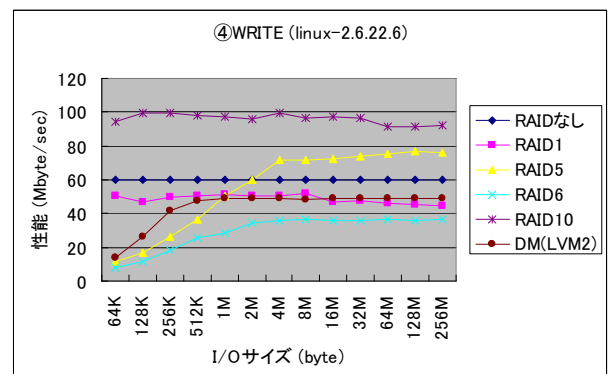
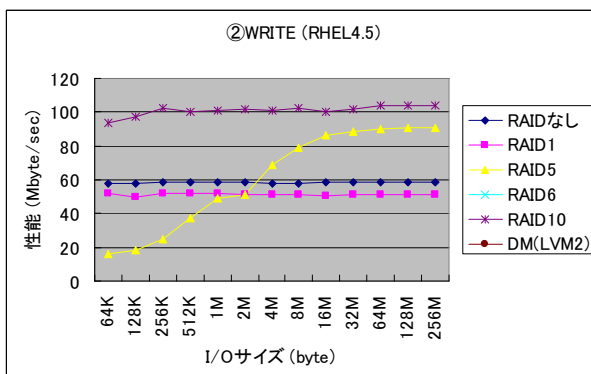
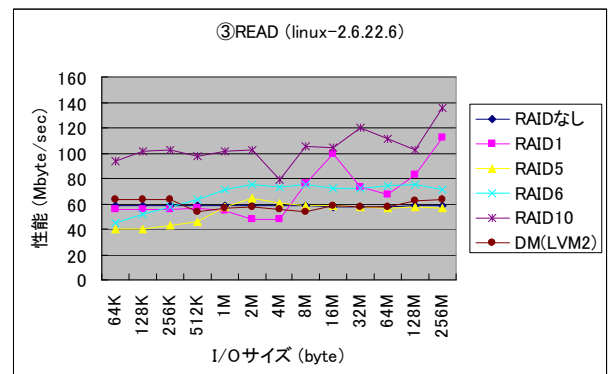
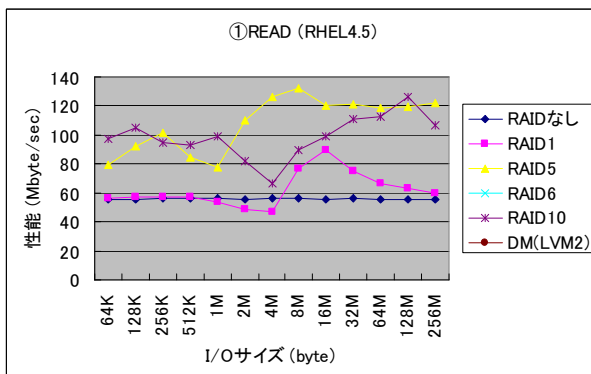


図 12：縮退状態でのファイルアクセス性能測定結果 (SATA)

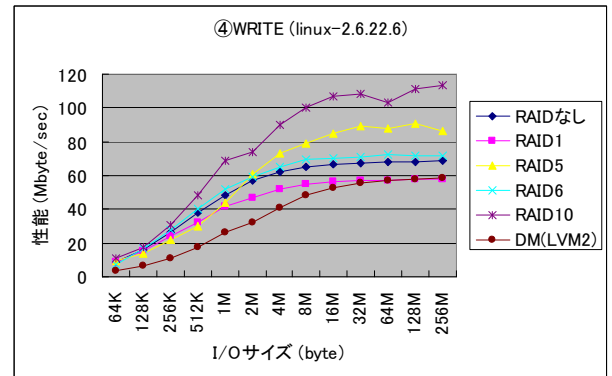
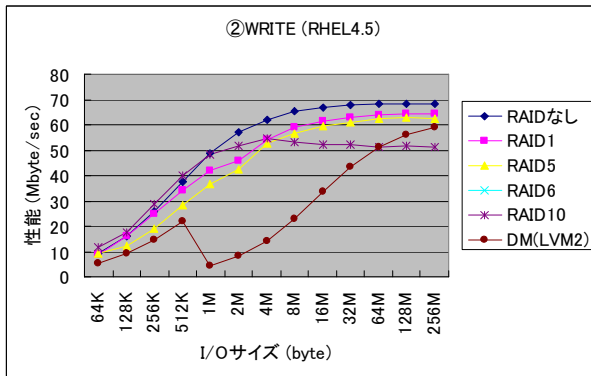
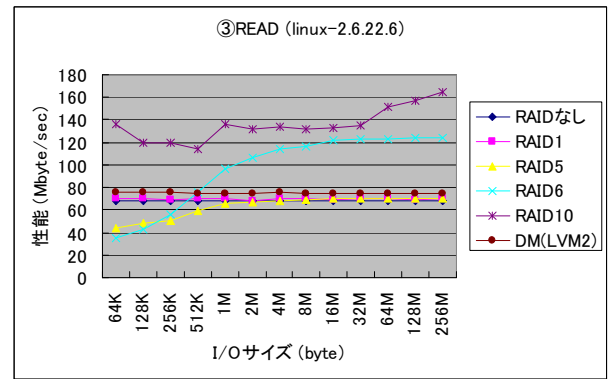
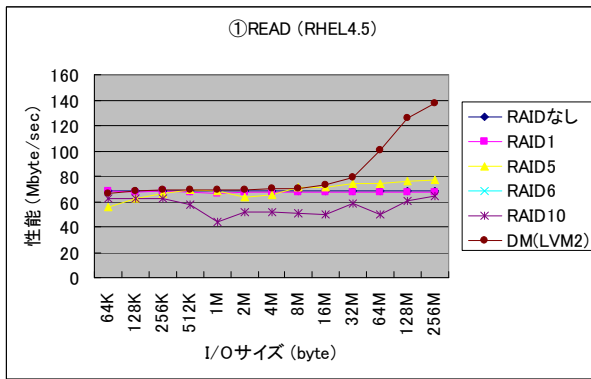


図 13 : 縮退状態でのブロックデバイスアクセス性能測定結果 (SCSI)

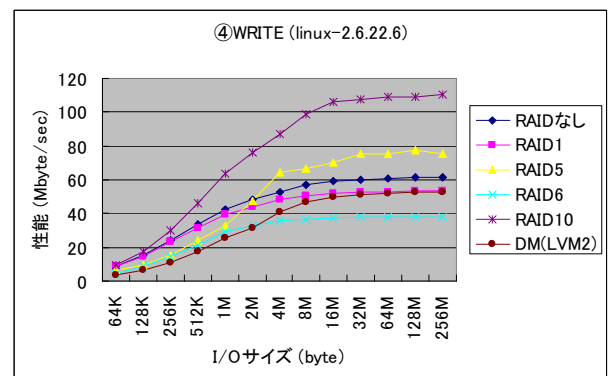
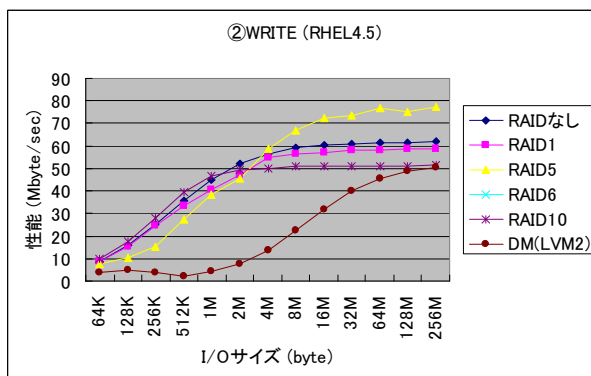
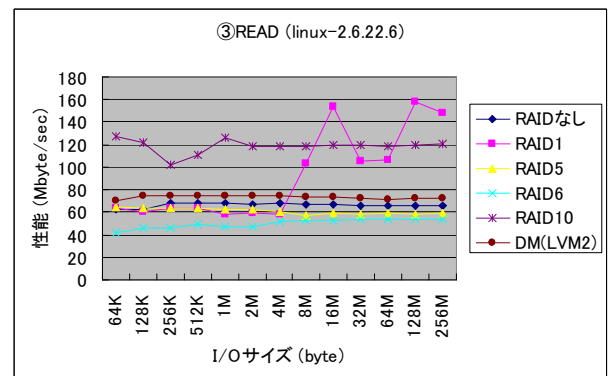
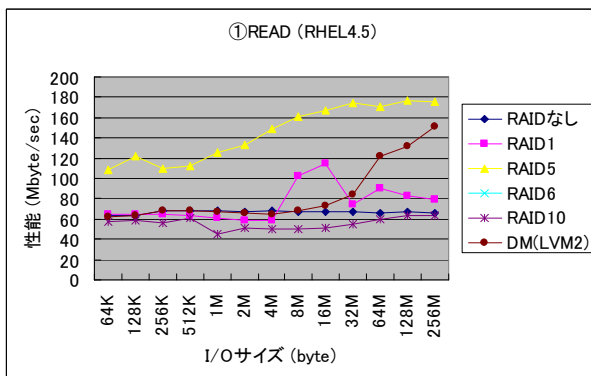


図 14 : 縮退状態でのファイルアクセス性能測定結果 (SCSI)

2.3. 復旧処理中の各 RAID レベルの通常運用時の性能測定結果

SATA ディスク上の各 RAID レベルの縮退状態でのブロックデバイスアクセス性能測定を図 15 に、ファイルアクセス性能測定結果を図 16 に示す。また SCSI ディスク上の各 RAID レベルの縮退状態でのブロックデバイスアクセス性能結果を図 17 に、ファイルアクセス性能測定結果を図 18 に示す。ディスクアクセス時の性能を測定するため、ここでは O_DIRECT を指定してアクセスを行った結果を示す。比較のために RAID なしの時の性能測定結果も合わせて示す。

(1) md (RAID1)

ブロックデバイス及びファイルアクセスに対する read、write それぞれのケースにおいて、RHEL4.5 とコミュニティカーネルのどちらの場合でも、通常運用時の結果とほぼ同様の結果が得られている。

(2) md (RAID5)

コミュニティカーネルでは多くの測定結果において read 性能の低下が見られるが、SCSI 上のファイルアクセスの評価結果のみ性能低下が認められない。

(3) md (RAID6)

RAID5 同様、コミュニティカーネルでは多くの測定結果において read 性能の低下が見られるが、SCSI 上のファイルアクセスの評価結果のみ性能低下が認められない。

(4) md (RAID10)

ブロックデバイス及びファイルアクセスに対する read、write それぞれのケースにおいて、RHEL4.5 とコミュニティカーネルのどちらの場合でも、通常運用時の結果とほぼ同様の結果が得られている。

(5) DM (LVM2)

機能評価で説明した通り、DM (LVM2)は復旧処理をサポートしておらず、復旧するためには RAID を再構築する必要があるため本評価項目は実施しなかった。

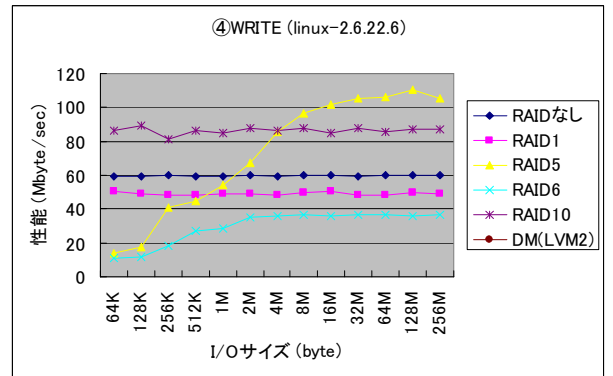
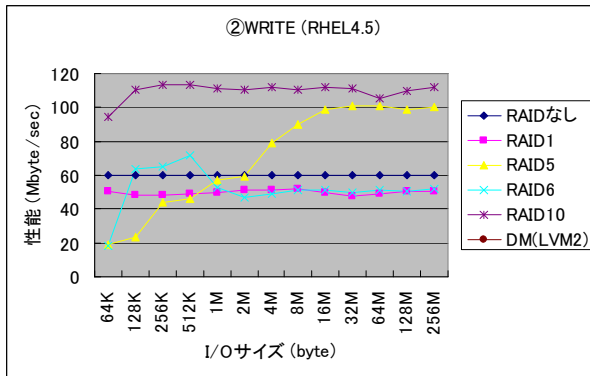
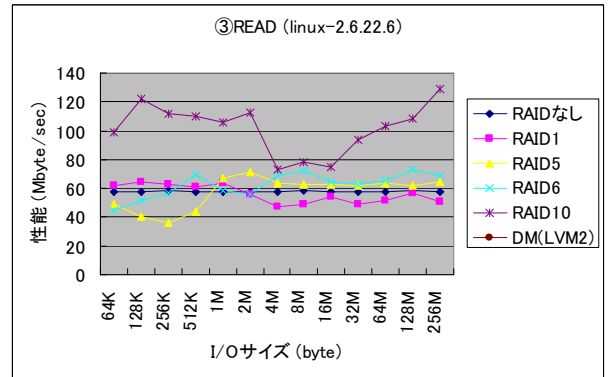
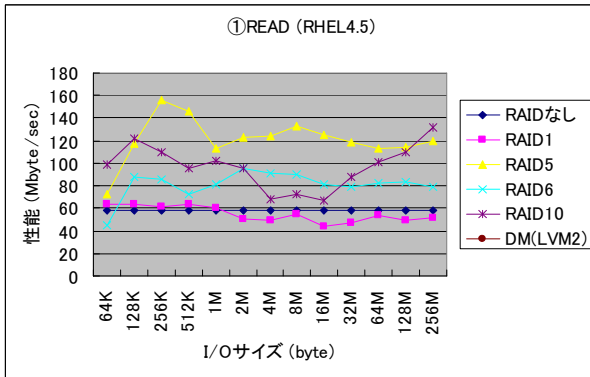


図 15 : 復旧処理中のブロックデバイスアクセス性能測定結果 (SATA)

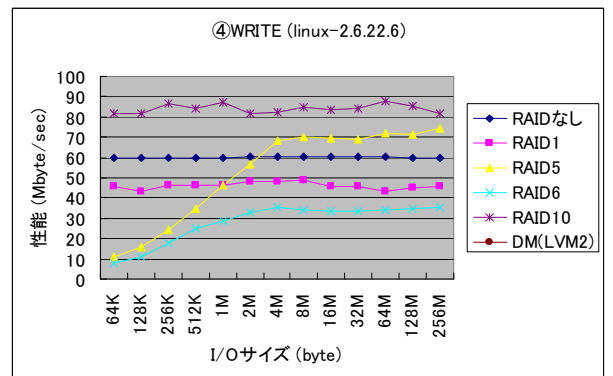
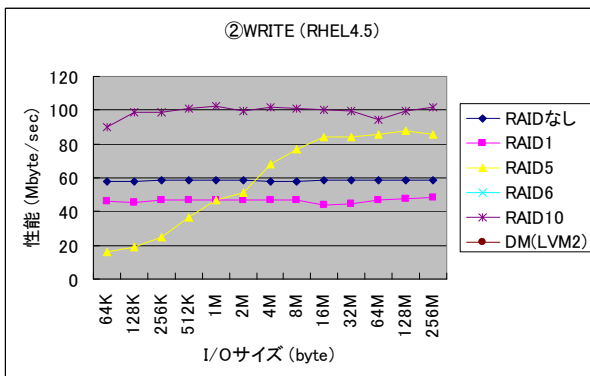
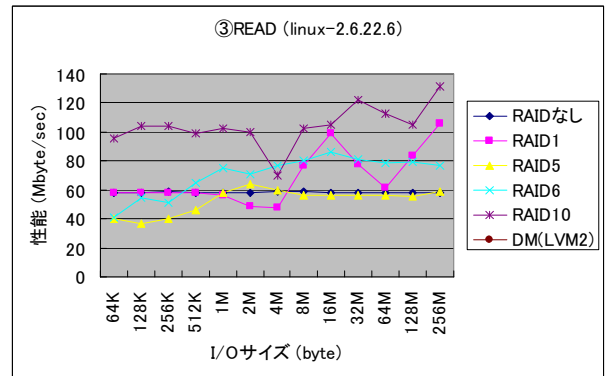
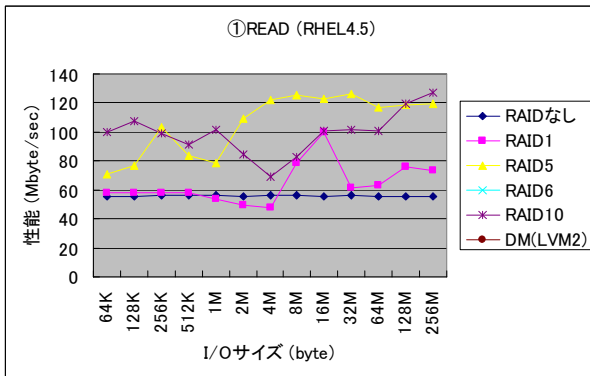


図 16 : 復旧処理中のファイルアクセス性能測定結果 (SATA)

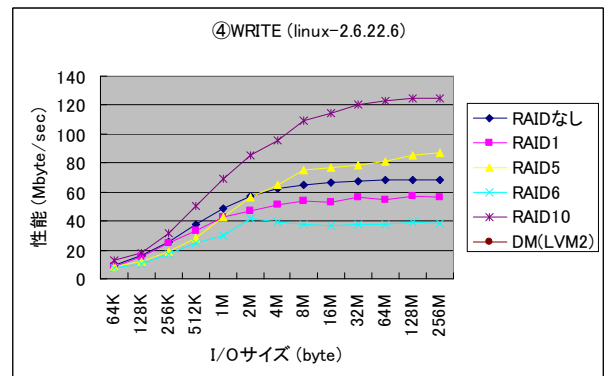
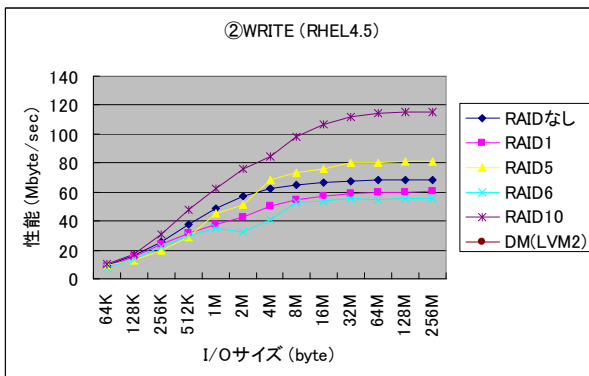
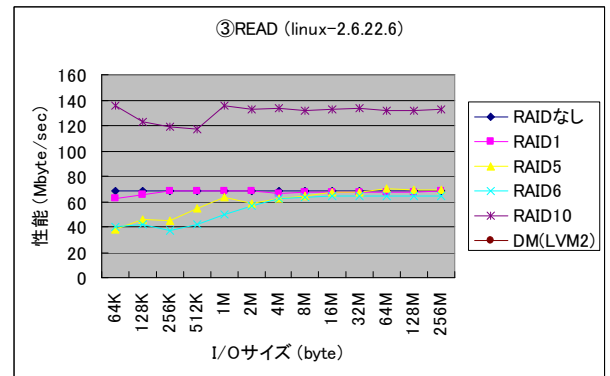
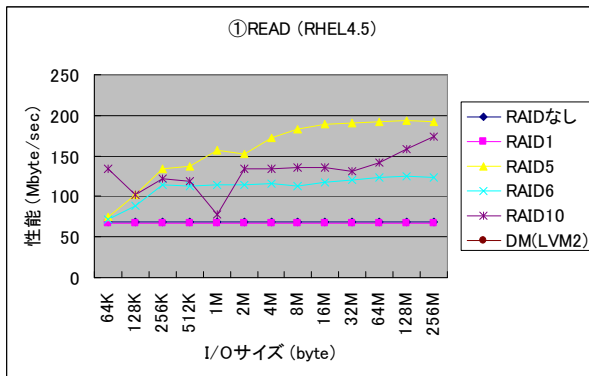


図 17：復旧処理中のブロックデバイスアクセス性能測定結果 (SCSI)

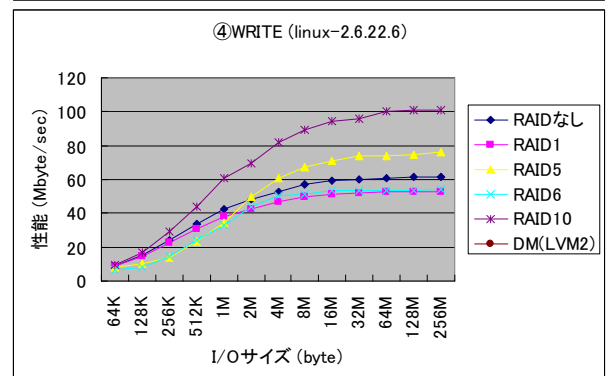
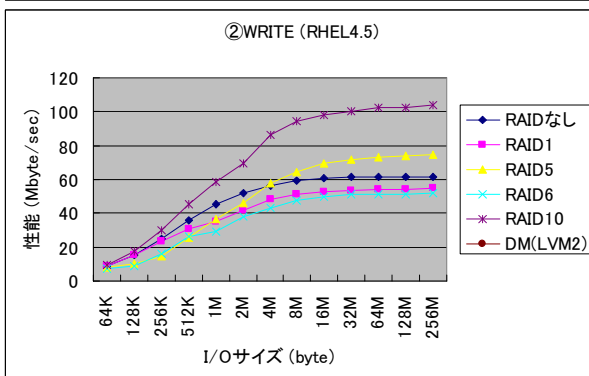
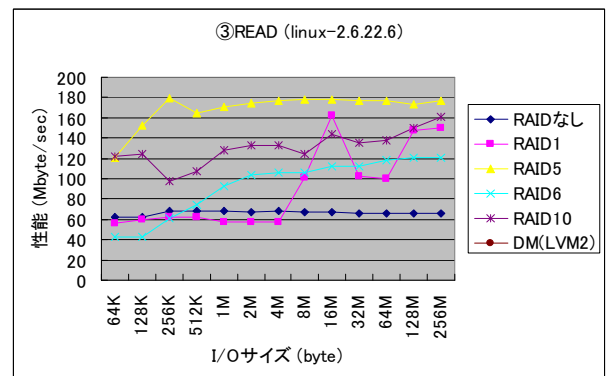
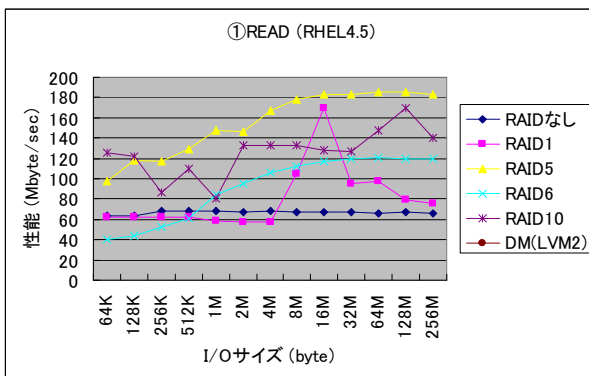


図 18：復旧処理中のファイルアクセス性能測定結果 (SCSI)

2.4. 復旧処理時間の測定結果

SATA 上での各 RAID レベルに対する無負荷時、及び負荷をかけた時の復旧処理時間の測定結果を図 19 と図 20 に示す。RHEL4.5 とコミュニティカーネルを比較すると、RAID5 の無負荷時の復旧処理時間のみが増加しているが、これは負荷をかけても増加せず、復旧処理にかかる時間自体は最新のコミュニティカーネルでは極めて安定した結果が得られている。

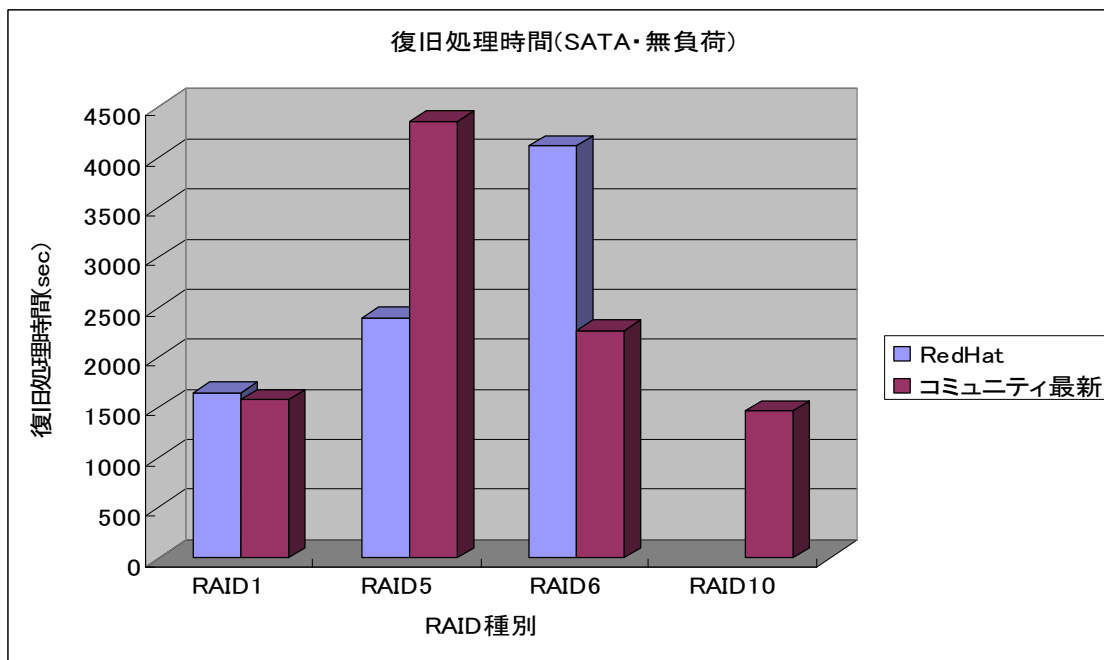


図 19：無負荷時の復旧処理時間 (SATA)

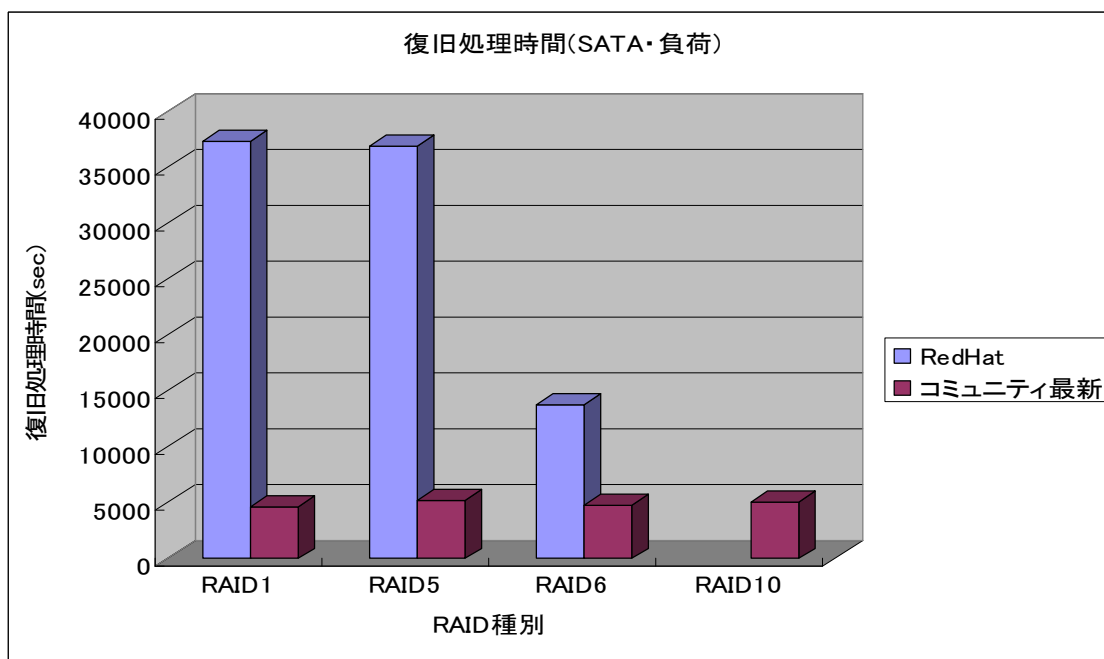


図 20：無負荷時の復旧処理時間 (SCSI)

また SCSI 上での各 RAID レベルに対する無負荷時、及び負荷をかけた時の復旧処理時間の測定結果を図 21 と図 22 に示す。RHEL4.5 とコミュニティカーネルを比較すると、全ての場合においてコミュニティカーネルの方が短い時間となっており、Linux カーネルの進化に伴い性能は向上している。しかしコミュニティカーネル上での RAID1 と RAID10 の負荷時の復旧処理時間は、無負荷時と比べると 20 倍以上かかっており、復旧時間は運用時の負荷に依存している。

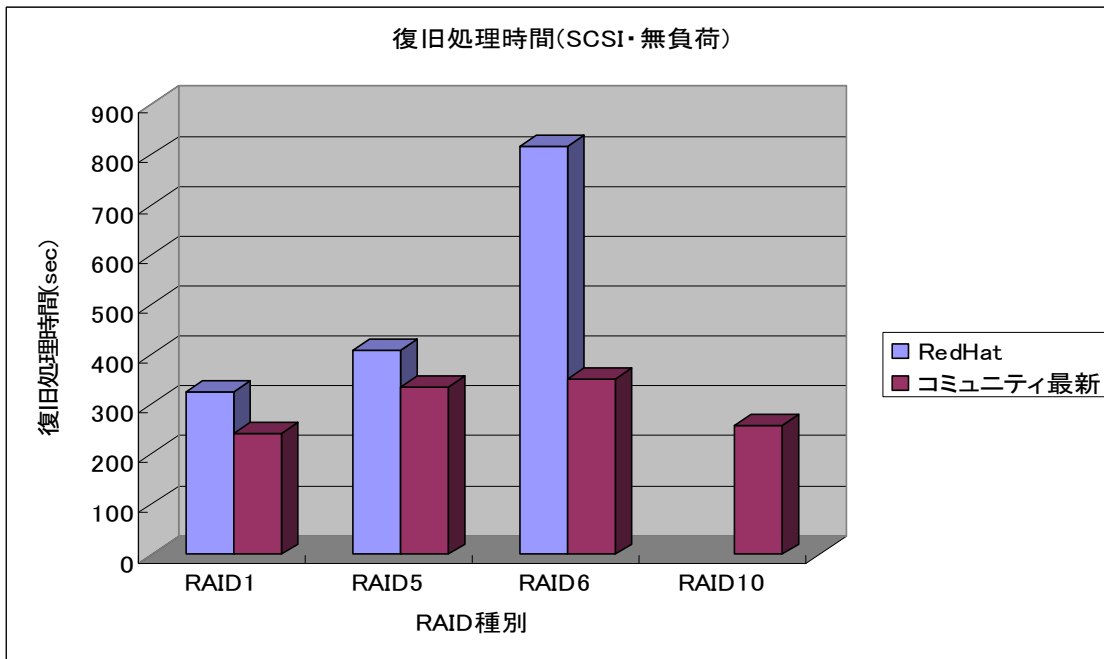


図 21：負荷時の復旧処理時間 (SATA)

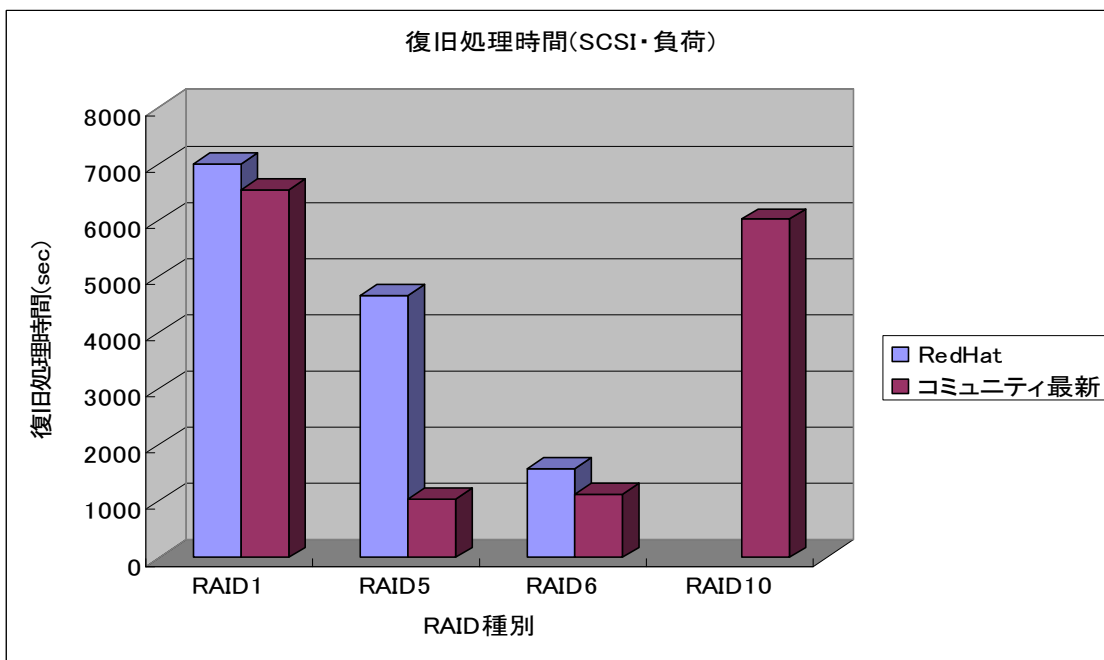


図 22：負荷時の復旧処理時間 (SCSI)

3. まとめ

性能評価の分析結果をまとめる。

通常運用中、縮退状態、復旧処理中の三つの場合についての性能測定結果からは、Linux のソフトウェア RAID 機能については、各 RAID レベルに応じて、以下のような一般的に言われている性能上の特性が確認できた。

- md の RAID1 や DM (LVM2)では RAID なしの場合と比べると、同等の read 性能が得られているが、write 性能はミラーリングのオーバーヘッドのためわずかに低下していることが確認できた。
- md の RAID5 と RAID6 では RAID なしの場合と比べると、I/O サイズが小さいときの write 性能は write ペナルティにより性能低下が見られるものの、それ以外の read 性能と I/O サイズが大きいときの write 性能では、複数のディスクに分散してデータを格納するため性能向上が確認できた。
- md の RAID10 では RAID なしの場合と比べると、複数のディスクに分散してデータを格納する影響で read 性能と write 性能の両方でスループットの向上が確認できた。

大きな I/O サイズの read / write を行う用途では RAID5 でスループットの向上が期待できる。また RAID なしの時と比較して、I/O サイズによらず性能向上が見られた RAID10 は、性能面からは有望であると考えられる。md の RAID1 や DM (LVM2)は write では多少のオーバーヘッドがあるものの、read の性能は RAID なしの時とほぼ変わらないため、read 主体の用途では性能的な問題は気にする必要がないと考える。

RHEL4.5 カーネルと最新のコミュニティカーネルを比較すると、全体的には通常運用時、縮退状態、復旧処理中の性能はおおむね向上しており、現在のコミュニティ開発でも、性能的な問題は改善されてゆく素地はあるものと予想される。

また各 RAID レベルでの無負荷時と負荷をかけた時の復旧時間測定から、以下のことが判明した。

- 多くの RAID 構成で、縮退時や復旧処理中の性能として通常運用時とほぼ同程度の性能が得られており、RAID の復旧処理に伴うオーバーヘッドで通常業務に大きな性能インパクトがあることは少ないと思われる。
- RHEL4.5 カーネルと比べ最新のコミュニティカーネルでは、復旧処理にかかる時間は少し改善しており、最適化が進んでいると考えられる。負荷をかけた時の復旧処理時間は最新コミュニティカーネルの結果が RHEL4.5 カーネルの結果を上回っているものの、無負荷時の復旧処理時間と比べると 20 倍以上遅くなっている場合もある。縮退した RAID アレイの復旧処理を運用中に行う場合は、運用時の負荷を考慮して復旧処理にかかる時間を見積る必要がある。

第6章 品質評価

本章では Linux のソフトウェア RAID 機能について、md と DM の、通常運用中、縮退状態、復旧処理中のそれぞれの場合に模擬故障を発生させることでエラー処理部分の評価を行った結果を示す。また故障ディスクの交換を行う場合を想定して、hotplug を実際に行い、デバイスドライバまで含めた OS として正しく動作するか評価を行った。

1. 品質評価方針

Linux のソフトウェア RAID (md および DM) は、RAID を構築する個々のディスクが故障した場合でも安定して運用を継続するための機能を有している。しかし、コミュニティベースの開発では上記機能が正しく動作するかを体系的に評価し品質を維持するための作業が不足していると考えられる。本調査では、故障模擬ライブラリにより様々な状態でのディスク故障を擬似的に発生させてディスク I/O の動作を検証することにより、ソフトウェア RAID の品質を評価した。

1.1. 品質評価内容

品質評価項目は以下の 4 種類に大別される。

- 運用中故障
- 縮退中故障
- 復旧中故障
- 故障ディスク交換

各項目について、以下の環境を使用して網羅的に評価を実施した。表 23 は、ソフトウェア RAID 種別ごとの評価環境を一覧にしたものである。

使用したカーネルは、RHEL4.5 とコミュニティ最新版カーネル linux-2.6.22.6 (以下 linux-2.6.22.6) の 2 種類である。コミュニティ最新版カーネルは、品質評価を開始した 2007/9/10 時点のものである。使用したディスクは SATA と SCSI の 2 種類である。

表 23 : ソフトウェア RAID 種別ごとの評価環境

md	DM
mdadm を用いて RAID1、RAID5、RAID6、RAID10 構築	LVM2 の RAID アレイ構築機能を用いて RAID1 構築

1.2. 品質評価項目

(1)～(5)に列挙されている全ての項目に対して、故障模擬ライブラリを用いて以下の 8 種類の故障を発生させてエラー処理系の品質を評価した。(6)ではディスクのホットプラグを行うことで評価を実施した。

- read 不可エラー
- read/write 不可エラー

- write によって訂正可能な read エラー
- read 単発エラー
- write 単発エラー
- read 単発無応答
- write 単発無応答
- read/write 無応答

運用中故障時の評価項目

- (1) スペアディスク付き環境でディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイ及びシステムの運用に問題がないことを確認。
- (2) スペアディスク無しの環境でディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイ及びシステムの運用に問題がないことを確認。

縮退中故障時の評価項目

- (3) ディスク 2 台に故障が発生した時、故障が検出でき、RAID アレイが運用できなくなるが、システムの運用に問題がないことを確認。

復旧中故障時の評価項目

- (4) 復旧処理中のスペアディスクに対してディスク故障が発生したときに、RAID アレイ及びシステムの運用に問題がないことを確認。
- (5) 復旧処理中のアクティブディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイが運用できなくなるが、システムの運用に問題がないことを確認。

故障ディスク交換

- (6) 故障したディスクを挿抜した時、RAID アレイ及びシステムの運用に問題がないことを確認。詳細については次節で述べる。

1.3. 故障ディスク交換評価

1.3.1 評価内容について

故障が発生してディスクを交換する場合を想定して、故障により切り離されたディスクを実際に交換する操作に対する評価を実施した。

評価パターンは以下の通りとする。

- 評価対象のカーネルは RHE4.5 及び RHEL4.5 上で動作する linux-2.6.22.6 カーネルとする。
- 評価対象のソフトウェア RAID は md (RAID1)、md (RAID5)、md (RAID6)、md (RAID10)とする。LVM2 は故障ディスク交換が機能的にサポートされていないため、評価対象外とする。
- デバイスは SCSI ディスク、SATA ディスクを用いて評価する。
- read エラーが発生した場合と write エラーが発生した場合について評価する。

なお、実際の運用において、ディスクのサプライズリムーブの後でも運用継続を可能とするためには、専用のハードウェアやドライバが必要でそれらの品質に大きく依存する機能である。そのため、ここで評価を行うのは、ディスクのサプライズリムーブを行った場合ではなく、手順を踏

んだディスクの hotplug を行った場合とする。

1.3.2 各品質評価項目における確認内容

各評価パターンにおいて確認する点は以下の通りとする。

継続運用

ディスクを手順通り交換した際に RAID アレイおよびシステム全体が継続動作すること。

構成情報

ディスクを交換した際に、システムに認識されるディスクの情報が適宜更新されること。

RAID アレイの交換動作の記録

システム運用ログに RAID アレイを構成するディスクが交換された事象が記録されること。

交換後のディスクの動作確認

組込まれたディスクに対して実際に読み書きができること。システムを再起動した後にそのディスクが再び正しく認識され、動作すること。

1.3.3 評価手順

運用中の RAID アレイに故障が発生した場合に実施すべき hotplug 手順を踏んでディスク交換を実施する。特に運用中の故障発生ならびにディスク交換であることを想定して、あらかじめ対象となる RAID アレイに負荷を与える。

具体的には以下の手順となる。

1. 対象となる RAID アレイを構築する。
2. 擬似的な高負荷環境を作るために、対象となる RAID アレイ上で読み書きを繰り返すテストスクリプトを実施する。
3. 故障模擬ライブラリを用いて対象の RAID アレイでディスク切り離しとなる模擬故障を発生させる。
4. 対象の RAID アレイから故障の発生したディスクを切り離す。
5. /proc/scsi/scsi インタフェースを用いて故障の発生したディスクを OS から切り離す。
6. 実際に筐体からディスクを抜き取り、保守部品と交換する。
7. /proc/scsi/scsi インタフェースを用いて交換したディスクを OS に組込む。
8. 対象の RAID アレイに交換したディスクを組込む。
9. 組込んだディスクに対して読み書きを実施する。特に組込んだディスクへの読み書きを確認するため、RAID アレイの冗長性を意図的に落とす。
10. OS を再起動し、交換ディスクへの読み書きを確認する。

2. 品質評価結果

品質評価を実施した結果をまとめる。以降 TP とは品質評価用テストプログラムのことである。また、LVM2 において評価対象外となっている項目は、LVM2 がスペアディスクをサポートしていないため実施できない項目である。品質評価での期待動作としては、縮退中故障と、復旧中故障(アクティブディスク故障)は、故障を検出して呼出元にエラーが返ることが期待され、その他の場合は運用が継続されることが期待される。

2.1. SATA デバイスの品質評価結果

SATA デバイスの品質評価結果は、以下のとおりである。

- (1) スペアディスク付き環境でディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイ及びシステムの運用に問題がない
 期待動作としては、エラーを検出し故障があるディスクを切り離して、スペアディスクに切替えることである。ただし、一時的なエラーや訂正可能なエラーについてはリトライや復旧処理をすることで、故障ディスクを切り離さずに運用を継続しても問題ない。

表 24 : SATA デバイス品質評価結果一覧 (スペアディスク有りディスク 1 台故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
 ■ パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2
read 不可エラー	○	○	○	○	-	○	○	○	○	-
read/write 不可エラー	○	○	○	○	-	○	○	○	○	-
writeによって訂正可能な read エラー	○	○	○	○	-	○ ※1	○ ※1	○ ※1	○ ※1	-
read 単発エラー	○ ※2	○ ※2	○ ※2	○ ※2	-	○ ※2	○ ※1	○ ※1	○ ※2	-
write 単発エラー	○ ※2	○ ※2	○ ※2	○ ※2	-	○	○	○	○	-
read 単発無応答	○	○	○	○	-	○ ※2	○ ※1	○ ※1	○ ※2	-
write 単発無応答	○	○	○	○	-	○	○	○	○	-
read/write 無応答	○	○	○	○	-	○	○	○	○	-

※1…md がデータを write することで read エラーを訂正。

※2…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題ない。

(2) スペアディスク無しで環境でディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイ及びシステムの運用に問題がない
 期待動作としては、エラーを検出し故障があるディスクを切り離して、RAID アレイを縮退状態にして運用を継続することである。ただし、一時的なエラーや訂正可能なエラーについてはリトライや復旧処理をすることで、故障ディスクを切り離さずに運用を継続することも考えられる。

表 25 : SATA デバイス品質評価結果一覧 (スペアディスク無しディスク 1 台故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
 ■ パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2
read 不可エラー	○	○	○	○	○ ※4	○	○	○	○	☆ ※1
read/write 不可エラー	○	○	○	○	○ ※4	○	○	○	○	☆ ※1
write によって訂正可能な read エラー	○	○	○	○	○ ※4	○ ※2	○ ※2	○ ※2	○ ※2	☆ ※1
read 単発エラー	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	○ ※2	○ ※2	○ ※3	○ ※3
write 単発エラー	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	○	○	○	○	☆ ※1
read 単発無応答	○	○	○	○	○ ※4	○ ※3	○ ※2	○ ※2	○ ※3	○ ※3
write 単発無応答	○	○	○	○	○ ※4	○	○	○	○	☆ ※1
read/write 無応答	○	○	○	○	○ ※4	○	○	○	○	☆ ※1

※1…残りディスク上からデータが読み出せる場合でも、ディスクエラーがそのまま呼出し元に返される場合がある。この場合でも RAID アレイとして縮退処理などは行われない。

※2…md がデータを write することで read エラーを訂正。

※3…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題ない。

※4…以降リブートするまで故障ディスクに対して write は行われるが、read アクセスは行かなくなる。構成情報は変更されない。

(3) 縮退状態で故障が発生した時、故障が検出でき、RAID アレイが運用継続できなくなるが、システムの運用に問題がない。

期待動作としては、アクセス要求を行ったアプリケーションにエラーを返し、システムとしてパニックやストールなどの障害を起こさずに、問題のある RAID アレイだけがエラーが発生していることがわかる状態に遷移することである。ただし、一時的なエラーについてはリトライすることで、そのまま運用を継続しても問題ない。

表 26 : SATA デバイス品質評価結果一覧 (縮退状態での故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
 ■ パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D1	RAI D5	RAI D6	RAID 10	LVM2	RAI D1	RAI D5	RAI D6	RAID 10	LVM2
read 不可エラー	◆ ※1	☆	☆	☆	☆	☆	● ※2	● ※2	☆	☆
read/write 不可 エラー	◆ ※1	☆	☆	☆	☆	☆	☆	● ※2	☆	☆
writeによって訂正可 能な read エラー	◆ ※1	☆	☆	☆	☆	☆	☆	● ※2	☆	☆
read 単発エラー	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	☆	☆	○ ※3	○ ※3
write 単発エラー	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	☆	☆	☆	☆	☆
read 単発無応答	◆ ※1	☆	☆	☆	☆	☆	☆	● ※2	○ ※3	○ ※3
write 単発無応答	◆ ※1	☆	☆	☆	☆	☆	☆	☆	☆	☆
read/write 無応答	◆ ※1	☆	☆	☆	☆	☆	☆	● ※2	☆	☆

※1…ログ上から read アクセスの無限ループが発生していることを確認。ただし 1~10 分後に SystemTap のフックをすり抜けて実行が継続されてしまう場合がある。

※2…read アクセスでエラーが発生した場合でも、エラーを返さずに残っていたデータを返す場合がある。この時、故障ディスクの切り離しは行われずに、RAID アレイはそのまま使われ続ける。故障情報はログ上からのみ取得が可能。

※3…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題なし。

- (4) 復旧中のスペアディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイの復旧処理は中断されるが、システムの運用に問題がない
- 期待動作としては、エラーを検出し故障があるディスクへの復旧処理を中断して、RAID アレイを縮退状態のまま運用を継続することである。ただし、一時的なエラーや訂正可能なエラーについてはリトライや復旧処理をすることで、そのまま運用を継続することも考えられる。

表 27 : SATA デバイス品質評価結果一覧 (復旧中のスペアディスク故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
 ■ : パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D1	RAI D5	RAI D6	RAID 10	LVM2	RAI D1	RAI D5	RAI D6	RAID 10	LVM2
read 不可エラー	— ※1	— ※1	— ※1	— ※1	—	— ※1	— ※1	— ※1	— ※1	—
read/write 不可エラー	○	○	○	○	—	○	○	○	○	—
write によって訂正可能な read エラー	— ※1	— ※1	— ※1	— ※1	—	— ※1	— ※1	— ※1	— ※1	—
read 単発エラー	— ※1	— ※1	— ※1	— ※1	—	— ※1	— ※1	— ※1	— ※1	—
write 単発エラー	○ ※2	○ ※2	○ ※2	○ ※2	—	○	○	○	○	—
read 単発無応答	— ※1	— ※1	— ※1	— ※1	—	— ※1	— ※1	— ※1	— ※1	—
write 単発無応答	○	○	○	○	—	○	○	○	○	—
read/write 無応答	○	○	○	○	—	○	○	○	○	—

※1…復旧中はスペアディスクに対して、アクティブ側のデータを書き出す write アクセスしか行われないため、実装上 read アクセスは発生しない。

※2…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題なし。

- (5) 復旧処理中のアクティブディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイが運用できなくなるが、システムの運用に問題がない
 期待動作としては、アクセス要求を行ったアプリケーションにエラーを返し、システムとしてパニックやストールなどの障害を起さずに、問題のある RAID アレイだけがエラーが発生していることがわかる状態に遷移することである。ただし、一時的なエラーについてはリトライすることで、そのまま運用を継続することも考えられる。

表 28 : SATA デバイス品質評価結果一覧 (復旧中のアクティブディスク故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
 ■ : パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D1	RAI D5	RAI D6	RAID 10	LVM2	RAI D1	RAI D5	RAI D6	RAID 10	LVM2
read 不可エラー	◆ ※1	☆	☆	■ ※2	-	☆ ※3	☆	☆	☆	-
read/write 不可 エラー	◆ ※1	☆	☆	■ ※2	-	☆ ※3	☆	☆	☆	-
write によって訂正可 能な read エラー	◆ ※1	☆	☆	■ ※2	-	☆ ※3	☆	● ※4	☆	-
read 単発エラー	○ ※5	○ ※5	○ ※5	○ ※5	-	○ ※5	☆	☆	○ ※5	-
write 単発エラー	○ ※5	○ ※5	○ ※5	○ ※5	-	☆ ※6	☆	☆	☆	-
read 単発無応答	◆ ※1	☆	☆	■ ※2	-	○ ※5	● ※4	☆	◆ ※7	-
write 単発無応答	◆ ※1	☆	☆	■ ※2	-	☆ ※6	☆	☆	☆	-
read/write 無応答	◆ ※1	☆	☆	■ ※2	-	☆ ※3	☆	☆	◆ ※7	-

※1…ログ上から read アクセスの無限ループが発生していることを確認。ただし 1~10 分後に SystemTap のフックをすり抜けて実行が継続されてしまう場合がある。

※2…md0_resync プロセスでパニックが発生。

※3…エラーは返るが、ログ上から resync の無限ループが発生していることを確認。

※4…read アクセスでエラーが発生した場合でも、エラーを返さずに残っていたデータを返す場合がある。この時、故障ディスクの切り離しは行われずに、RAID アレイはそのまま使われ続ける。故障情報はログ上からのみ取得が可能。

※5…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題なし。

※6…エラーは返るが、resync 動作は継続し write されたデータは失われる

※7…コミュニティカーネルの RAID10 ストール障害。

(6) 故障したディスクを挿抜した時、RAID アレイ及びシステムの運用に問題がないことを確認

SATA ディスクに対する故障ディスク交換の評価結果を表 29 に示す。いずれのパターンにおいても故障ディスクの交換に成功することが確認された。

表 29 : SATA ディスクに対する故障ディスク交換の評価結果

(凡例 ○ : 問題なし × : 障害発生 - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2
add/remove 操作、 継続運用	○	○	○	○	-	○	○	○	○	-
構成情報更新	○	○	○	○	-	○	○	○	○	-
RAID アレイの交換 動作の記録	○	○	○	○	-	○	○	○	○	-
交換後のディスクの 動作確認	○	○	○	○	-	○	○	○	○	-

ただし、2.2.(6)で説明するように、ディスクのサプライズリムーブを行った場合、切り離れたディスク以外での入出力エラーや、ディスクドライブのハード的な故障が発生する可能性があるため、安全のためには一旦電源を落としてからディスク交換を行った方が確実と言える。

2.2. SCSI デバイスの品質評価結果

SCSI デバイスの品質評価結果は、以下のとおりである。

(1) スペアディスク付き環境でディスク 1 台に故障が発生した時、故障が検出でき、

RAID アレイ及びシステムの運用に問題がない

期待動作としては、エラーを検出し故障があるディスクを切り離して、スペアディスクに切替えることである。ただし、一時的なエラーや訂正可能なエラーについてはリトライや復旧処理をすることで、故障ディスクを切り離さずに運用を継続しても問題ない。

表 30 : SCSI デバイス品質評価結果一覧 (スペアディスク有りディスク 1 台故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
■ パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DMr	md				DM
	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2
read 不可エラー	○	○	○	○	-	○	○	○	○	-
read/write 不可 エラー	○	○	○	○	-	○	○	○	○	-
writeによって訂正可 能な read エラー	○	○	○	○	-	※1	※1	※1	※1	-
read 単発エラー	○ ※2	○ ※2	○ ※2	○ ※2	-	○ ※2	○ ※1	○ ※1	○ ※2	-
write 単発エラー	○ ※2	○ ※2	○ ※2	○ ※2	-	○	○	○	○	-
read 単発無応答	○	○	○	○	-	○ ※2	○ ※1	○ ※1	○ ※2	-
write 単発無応答	○	○	○	○	-	○	○	○	○	-
read/write 無応答	○	○	○	○	-	○	○	○	○	-

※1…md がデータを write することで read エラーを訂正。

※2…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題ない。

(2) スペアディスク無しでディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイ及びシステムの運用に問題がない
 期待動作としては、エラーを検出し故障があるディスクを切り離して、RAID アレイを縮退状態にして運用を継続することである。ただし、一時的なエラーや訂正可能なエラーについてはリトライや復旧処理をすることで、故障ディスクを切り離さずに運用を継続することも考えられる。

表 31 : SCSI デバイス品質評価結果一覧 (スペアディスク無しディスク 1 台故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
 ■ パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2
read 不可エラー	○	○	○	○	○ ※5	○	○	○	○	☆ ※1
read/write 不可エラー	○	○	○	○	○ ※5	○	○	○	○	☆ ※1
write によって訂正可能な read エラー	○	○	○	○	○ ※5	○ ※2	○ ※2	○ ※2	○ ※2	☆ ※1
read 単発エラー	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	○ ※2	○ ※2	○ ※3	○ ※3
write 単発エラー	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	○	○	○	○	☆ ※1
read 単発無応答	○	○	○	○	○ ※5	○ ※3	○ ※2	○ ※2	○ ※3	○ ※3
write 単発無応答	○	○	○	○	○ ※5	○	○	○	○	☆ ※1
read/write 無応答	○	○	○	○	○ ※5	◆ ※4	○	○	○	☆ ※1

- ※1…残りディスク上からデータが読み出せる場合でも、ディスクエラーがそのまま呼出し元に返される場合がある。この場合でも RAID アレイとして縮退処理などは行われない。
- ※2…md がデータを write することで read エラーを訂正。
- ※3…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題ない。
- ※4…コミュニティカーネルの RAID1 ストール障害。
- ※5…以降リブートするまで故障ディスクに対して write は行われるが、read アクセスは行かなくなる。構成情報は変更されない。

(3) 縮退状態で故障が発生した時、故障が検出でき、RAID アレイが運用継続できなくなるが、システムの運用に問題がない。

期待動作としては、アクセス要求を行ったアプリケーションにエラーを返し、システムとしてパニックやストールなどの障害を起こさずに、問題のある RAID アレイだけがエラーが発生していることがわかる状態に遷移することである。ただし、一時的なエラーについてはリトライすることで、そのまま運用を継続しても問題ない。

表 32 : SCSI デバイス品質評価結果一覧 (縮退状態での故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
 ■ パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D1	RAI D5	RAI D6	RAID 10	LVM2	RAI D1	RAI D5	RAI D6	RAID 10	LVM2
read 不可エラー	◆ ※1	☆	☆	☆	☆	☆	● ※2	☆	☆	☆
read/write 不可 エラー	◆ ※1	☆	☆	☆	☆	☆	● ※2	☆	☆	☆
write によって訂正可 能な read エラー	◆ ※1	☆	☆	☆	☆	☆	● ※2	● ※2	☆	☆
read 単発エラー	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	☆	☆	○ ※3	○ ※3
write 単発エラー	○ ※3	○ ※3	○ ※3	○ ※3	○ ※3	☆	☆	☆	☆	☆
read 単発無応答	◆ ※1	☆	☆	☆	☆	☆	● ※2	☆	○ ※3	○ ※3
write 単発無応答	◆ ※1	☆	☆	☆	☆	☆	☆	☆	☆	☆
read/write 無応答	◆ ※1	☆	☆	☆	☆	☆	● ※2	● ※2	☆	☆

※1…ログ上から read アクセスの無限ループが発生していることを確認。ただし 1~10 分後に SystemTap のフックをすり抜けて実行が継続されてしまう場合がある。

※2…read アクセスでエラーが発生した場合でも、エラーを返さずに残っていたデータを返す場合がある。この時、故障ディスクの切り離しは行われずに、RAID アレイはそのまま使われ続ける。故障情報はログ上からのみ取得が可能。

※3…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題なし。

(4) 復旧中のスペアディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイの復旧処理は中断されるが、システムの運用に問題がない
 期待動作としては、エラーを検出し故障があるディスクへの復旧処理を中断して、RAID アレイを縮退状態のまま運用を継続することである。ただし、一時的なエラーや訂正可能なエラーについてはリトライや復旧処理をすることで、そのまま運用を継続することも考えられる。

表 33 : SCSI デバイス品質評価結果一覧 (復旧中のスペアディスク故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
 ■ パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D1	RAI D5	RAI D6	RAID 10	LVM2	RAI D1	RAI D5	RAI D6	RAID 10	LVM2
read 不可エラー	- ※1	- ※1	- ※1	- ※1	-	- ※1	- ※1	- ※1	- ※1	-
read/write 不可エラー	○	○	○	○	-	○	○	○	○	-
write によって訂正可能な read エラー	- ※1	- ※1	- ※1	- ※1	-	- ※1	- ※1	- ※1	- ※1	-
read 単発エラー	- ※1	- ※1	- ※1	- ※1	-	- ※1	- ※1	- ※1	- ※1	-
write 単発エラー	○ ※2	○ ※2	○ ※2	○ ※2	-	○	○	○	○	-
read 単発無応答	- ※1	- ※1	- ※1	- ※1	-	- ※1	- ※1	- ※1	- ※1	-
write 単発無応答	○	○	○	○	-	○	○	○	○	-
read/write 無応答	○	○	○	○	-	○	○	○	○	-

※1…復旧中はスペアディスクに対して、アクティブ側のデータを書き出す write アクセスしか行われないため、実装上 read アクセスは発生しない。

※2…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題なし。

- (5) 復旧処理中のアクティブディスク 1 台に故障が発生した時、故障が検出でき、RAID アレイが運用できなくなるが、システムの運用に問題がない
 期待動作としては、アクセス要求を行ったアプリケーションにエラーを返し、システムとしてパニックやストールなどの障害を起こさずに、問題のある RAID アレイだけがエラーが発生していることがわかる状態に遷移することである。ただし、一時的なエラーについてはリトライすることで、そのまま運用を継続することも考えられる。

表 34 : SCSI デバイス品質評価結果一覧 (復旧中のアクティブディスク故障)

(凡例 ○ : アクセス成功 ☆ : アクセスエラー ● : 故障検出されず ◆ : ストール
 ■ : パニック - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D1	RAI D5	RAI D6	RAID 10	LVM2	RAI D1	RAI D5	RAI D6	RAID 10	LVM2
read 不可エラー	◆ ※1	☆	☆	■ ※2	-	☆ ※3	☆	☆	☆	-
read/write 不可 エラー	◆ ※1	☆	☆	■ ※2	-	☆ ※3	☆	☆	☆	-
write によって訂正可 能な read エラー	◆ ※1	☆	☆	■ ※2	-	☆ ※3	☆	☆	☆	-
read 単発エラー	○ ※4	○ ※4	○ ※4	○ ※4	-	○ ※4	☆	☆	○ ※4	-
write 単発エラー	○ ※4	○ ※4	○ ※4	○ ※4	-	☆ ※5	☆	☆	☆	-
read 単発無応答	◆ ※1	☆	☆	■ ※2	-	○ ※4	● ※6	● ※6	◆ ※7	-
write 単発無応答	◆ ※1	☆	☆	■ ※2	-	☆ ※5	☆	☆	☆	-
read/write 無応答	◆ ※1	☆	☆	■ ※2	-	☆ ※3	☆	● ※6	◆ ※7	-

※1…ログ上から read アクセスの無限ループが発生していることを確認。ただし 1~10 分後に SystemTap のフックをすり抜けて実行が継続されてしまう場合がある。

※2…md0_resync プロセスでパニックが発生。

※3…エラーは返るが、ログ上から resync の無限ループが発生していることを確認。

※4…故障検知時のリトライで成功するためエラーを検出しない。これは正常な動作であるため問題なし。

※5…エラーは返るが、resync 動作は継続し write されたデータは失われる

※6…read アクセスでエラーが発生した場合でも、エラーを返さずに残っていたデータを返す場合がある。この時、故障ディスクの切り離しは行われずに、RAID アレイはそのまま使われ続ける。故障情報はログ上からのみ取得が可能。

※7…コミュニティカーネルの RAID10 ストール障害。

(6) 故障したディスクを挿抜した時、RAID アレイ及びシステムの運用に問題がないことを確認

SCSI ディスクに対する評価結果を表 35 に示す。いずれのパターンにおいても故障ディスクの交換に成功することが確認された。

表 35 : SCSI ディスクに対する故障ディスク交換の評価結果

(凡例 ○ : 問題なし × : 障害発生 - : 評価対象外)

	RHEL4.5					linux-2.6.22.6				
	md				DM	md				DM
	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2	RAI D 1	RAI D 5	RAI D 6	RAID 10	LVM2
add/remove 操作、 継続運用	○	○	○	○	-	○	○	○	○	-
構成情報更新	○	○	○	○	-	○	○	○	○	-
RAID アレイの交換 動作の記録	○	○	○	○	-	○	○	○	○	-
交換後のディスクの 動作確認	○	○	○	○	-	○	○	○	○	-

今回の評価結果を見る限り、md の RAID ドライバの機能に限定すると故障ディスクの hotplug を実施できる品質にあるといえる。

しかし一般的にディスクの hotplug を安定して行うためには、ソフトウェア RAID ドライバの品質だけではなく、ハードウェアと HBA ドライバの機能・品質に依存する部分がある。今回、評価を行った環境においても、ディスク切り離し手順をふまずにサブライズリムーブを行ったところ、切り離れたディスク以外での入出力エラーや、ディスクドライブのハードウェア的な故障が発生する場合があった。ディスクのサブライズリムーブを行ったこととの因果関係は不明であるが、電氣的にはディスクを hotplug できるハードウェア環境でも、安全のためには一旦電源を落としてからディスク交換を行った方が確実である。

3. 品質評価結果の分析

前節の品質評価結果を分析した概要をまとめる。

3.1. RHEL4.5 カーネル上の品質結果の分析

(1) md (RAID1)

障害①

RHEL4.5 の md (RAID1)では縮退中に媒体不良による read エラーが発生すると、必ずリトライが発生する作りとなっている。一時的なエラーはこれにより救済できるが、固定故障が発生している場合は、問題のある read をずっと繰り返すことになる。

(2) md (RAID5)

特に問題は検出されていない。

(3) md (RAID6)

特に問題は検出されていない。

(4) md (RAID10)

障害①

RHEL4.5 の md (RAID10)において、復旧中にアクティブディスクで故障が発生した場合、md0_resync プロセスでパニックが発生する。この件については md の不具合であると考えられるが、原因は判明していない。/var/log/messages には次のように表示される。

```
Dec 2 19:31:00 ipads1 kernel: kernel BUG at drivers/md/raid10.c:1446!
Dec 2 19:31:00 ipads1 kernel: invalid operand: 0000 [#1]
Dec 2 19:31:00 ipads1 kernel: SMP
Dec 2 19:31:00 ipads1 kernel: Modules linked in: stap_88218f307fab240fdafb6df44
bbfacf9_27177(U) raid10 parport_pc lp parport autofs4 i2c_dev i2c_core sunrpc dm
_mirror dm_mod button battery (U) ac md5 ipv6 uhci_hcd ehci_hcd hw_random e1000
floppy ext3 jbd raid1 ata_piix ahci libata sd_mod scsi_mod
Dec 2 19:31:00 ipads1 kernel: CPU: 1
Dec 2 19:31:00 ipads1 kernel: EIP: 0060:[<f89c315a>] Tainted: PF VLI
Dec 2 19:31:00 ipads1 kernel: EFLAGS: 00010246 (2.6.9-55.ELsmp)
Dec 2 19:31:00 ipads1 kernel: EIP is at sync_request+0x322/0x7de [raid10]
Dec 2 19:31:00 ipads1 kernel: eax: f5191d80 ebx: f7f2a400 ecx: 0000000c e
dx: 00000002
Dec 2 19:31:00 ipads1 kernel: esi: f5191900 edi: f4df03a0 ebp: f4df0380 e
sp: f4308e2c
Dec 2 19:31:00 ipads1 kernel: ds: 007b es: 007b ss: 0068
Dec 2 19:31:00 ipads1 kernel: Process md0_resync (pid: 5322, threadinfo=f430800
0 task=f4a1e170)
Dec 2 19:31:00 ipads1 kernel: Stack: 00000000 c0413828 00000000 00000000 f4308e
60 c012a016 f4308e60 00000001
Dec 2 19:31:00 ipads1 kernel: 00000002 00000000 00000000 00000000 000000
00 00000046 f7dcb3f8 c0225581
Dec 2 19:31:00 ipads1 kernel: 00250280 00000000 00000000 00250200 000000
00 f7f2a400 f89c58a0 f7f2a400
Dec 2 19:31:00 ipads1 kernel: Call Trace:
Dec 2 19:31:00 ipads1 kernel: [<c012a016>] del_timer_sync+0x7a/0x9c
Dec 2 19:31:00 ipads1 kernel: [<c0225581>] __generic_unplug_device+0x14/0x2d
Dec 2 19:31:00 ipads1 kernel: [<c0274800>] md_do_sync+0x3f1/0x84f
Dec 2 19:31:00 ipads1 kernel: [<c011cd42>] recalc_task_prio+0x128/0x133
Dec 2 19:31:00 ipads1 kernel: [<c011cdd5>] activate_task+0x88/0x95
Dec 2 19:31:00 ipads1 kernel: [<c0273886>] md_thread+0x13b/0x168
Dec 2 19:31:00 ipads1 kernel: [<c012052d>] autoremove_wake_function+0x0/0x2d
Dec 2 19:31:00 ipads1 kernel: [<c02d5dfe>] ret_from_fork+0x6/0x14
Dec 2 19:31:00 ipads1 kernel: [<c012052d>] autoremove_wake_function+0x0/0x2d
Dec 2 19:31:00 ipads1 kernel: [<c027374b>] md_thread+0x0/0x168
Dec 2 19:31:00 ipads1 kernel: [<c01041f5>] kernel_thread_helper+0x5/0xb
Dec 2 19:31:00 ipads1 kernel: Code: 0c 8b 5c 24 1c 89 5d 34 8b 44 24 24 89 45 4
4 eb 11 ff 44 24 20 83 c7 10 39 54 24 20 0f 8c 02 ff ff ff 8b 54 24 20 3b 56 20
75 08 <0f> 0b a6 05 6d 3b 9c f8 ff 44 24 24 83 04 24 0c 8b 4c 24 24 3b
Dec 2 19:31:00 ipads1 kernel: <0>Fatal exception: panic in 5 seconds
```

(5) DM (LVM2)

以降リブートするまで故障ディスクに対して write は行われるが、read アクセスは行かなくなる。メンテナの方針により read エラーでは自動的に再構成を行わないためであるが、構成情報は変更されないため、ログを調べないと故障ディスクの特定ができない。また再起動すると故障ディスクが再組み込みされるため、もし write でエラーが発生し、リブート後にディスクがアクセスできるようになると、データ不正を起こす可能性がある。

3.2. コミュニティカーネル上の品質結果の分析

(1) md (RAID1)

障害①

コミュニティカーネルの md (RAID1)において、read/write 無応答が発生した場合にシステムがストールする。無応答が発生している最中に、pdflush からの write がスケジューリングされた時に発生している。無応答のコマンドがタイムアウトで戻った時に、md-raid1 カーネルスレッドのエラーハンドリングが、pdflush の write 要求作成とぶつかると、pdflush のロックが原因で raid1 のカーネルスレッドが待ち合わせ状態になるが、pdflush は raid1 のカーネルスレッドが動いていないと処理を終えられないため、デッドロックが発生すると考えられる。問題検出は SCSI ディスクで無応答の模擬故障を発生させた場合に検出しているが、理論上は SCSI、SATA 両方のディスクで、無応答、単発エラーのどちらが発生した場合でも再現しうると考えられる。コミュニティカーネルのバグが疑われ、コミュニティに報告予定である。

障害②

復旧処理中にアクティブディスクでディスクエラーが発生した場合に、故障を検知してアプリケーションにはエラーを返すものの、resync の進み具合により処理が中断されずに RAID アレイを利用可能な状態のままとなり、最終的に resync が完了した時点で通常運用状態に復旧してしまう可能性がある。特に write エラーが発生した場合は、アプリケーションが同期的な書き込みを行っていないければ、システムのディスクキャッシュの書き戻し処理でエラーが発生するだけであり、アプリケーションは直接エラーを認識することは無く、ログを確認しなければ、ディスクエラーがあったことに気付かないまま運用が継続される恐れもある。復旧処理中ではない、縮退動作中の場合はエラーを検出して処理していることを考えると、開発者はエラーを処理する意図はあると考えられるため、コミュニティに報告予定である。

(2) md (RAID5)

read アクセスでエラーが発生した場合でも、エラーを返さずに残っていたデータを返す場合がある。この時、故障ディスクの切り離しは行われずに、RAID アレイはそのまま使われ続ける。故障情報はログ上からのみ取得が可能である。意図的にエラーを救済しているとも考えられるが、セクタエラーでの代替セクタ割当てのための write を行っていないことや、RAID5 ドライバは他の故障ではすぐ切り離し処理を行っていることを考えると、エラー処理もれの可能性もある。

(3) md (RAID6)

read アクセスでエラーが発生した場合でも、エラーを返さずに残っていたデータを返す場合がある。この時、故障ディスクの切り離しは行われずに、RAIDアレイはそのまま使われ続ける。故障情報はログ上からのみ取得が可能である。意図的にエラーを救済しているとも考えられるが、セクタエラーでの代替セクタ割当てのための write を行っていないことや、RAID5 ドライバは他の故障ではすぐ切り離し処理を行っていることを考えると、エラー処理もれの可能性もある。

(4) md (RAID10)

障害①

コミュニティカーネルの md (RAID10)において、read/write 無応答が発生した場合にシステムがストールする。現在まだ調査中であり、発生条件や原因などは判明していない。

(5) DM (LVM2)

障害①

ディスクで read エラーが発生した場合、RAID アレイに冗長性が残っており、残りディスクからデータが読み出せる場合でも、ディスクエラーがそのまま呼出し元に返される場合がある。この場合でも RAID アレイとして縮退処理などは行われない。直接的には read アクセスが片側のディスクからしか行われておらず、エラーが発生してももう片側にリトライが発生していないため起きてるように見える。RHEL4 では修正されている問題がコミュニティカーネルでは残されたままになっていることを検出したものであり、バグとしてコミュニティに認知されているが修正が完了していない状態である [参考文献(p.99)の 7]。write の処理も同様に、冗長性が十分に機能していない。

4. まとめ

品質評価では、実際に発生しうるディスクの故障のパターン、故障発生時の RAID の状態の全組合せについて評価を実施しており、RAIDアレイの網羅的な品質評価となっていると考えられる。品質評価結果に関して、SATA と SCSI の両環境の結果をまとめたものを表 36 に示す。

表 36 : 品質評価項目数と障害件数

(障害件数 / 評価項目数)

		md				DM
		RAID1	RAID5	RAID6	RAID10	LVM2
RHEL4.5	運用中故障	0 / 32	0 / 32	0 / 32	0 / 32	0 / 16
	縮退中故障	12 / 16	0 / 16	0 / 16	0 / 16	0 / 16
	復旧中故障	12 / 32	0 / 32	0 / 32	12 / 32	—
linux-2.6.22.6	運用中故障	1 / 32	0 / 32	0 / 32	0 / 32	12 / 16
	縮退中故障	0 / 16	6 / 16	7 / 16	0 / 16	0 / 16
	復旧中故障	8 / 16	2 / 16	3 / 16	4 / 16	—

今回の品質評価を通じて、以下のことが判明した。

- 障害を分析した結果、md あるいは DM のバグと考えられる障害を、RHEL4.5 カーネルで 2 種類、最新のコミュニティカーネルで 4 種類検出したが、RHEL4.5 と最新コミュニティカーネルで検出された問題は、全く別々のものであった。
- コミュニティの最新版カーネルの md では、最も一般的であるメディアエラーが発生した場合、冗長系からデータを復元して書き戻すことで、ディスクの代替セクタ割当てを行わせるよう、エラー処理が強化されていた。RHEL4.5 の場合、メディアエラーが発生しても、他のディスク故障の場合と同様に故障ディスクの切り離し処理が行われる。
- DM のソフトウェア RAID 機能に対しては、RHEL4.5 で修正されているバグが、最新のコミュニティカーネルではまだ取り込まれておらず、カーネルの進化に伴い品質が必ずしも向上していない場合があることが判明した。
- RHEL4.5 上の各 RAID 機能については、通常運用中故障に対するエラー処理で特に問題は検出されなかった。RHEL4 上で RAID の運用を行っている場合は、冗長性がある状態で動かしていれば故障によるエラーが正しく処理されて故障ディスクの切り離しが行われることが確認された。一方、縮退中や復旧中にエラーが発生した場合、一部でエラーが正しく処理できずにストールやパニックが発生するケースがあった。
- md の RAID ドライバの機能に限定すれば故障ディスクの hotplug を実施できる品質にあるが、ハードウェアや HBA ドライバの機能・品質に依存する部分があるため、安全のためには一旦電源を落としてからディスク交換を行うのが確実である。

縮退した状態で、さらに故障が発生すると、データを復旧することは極めて困難となる。このため、縮退が発生した時点で、速やかにシステムを止めて故障ディスクの交換ができる運用を行うか、冗長性を高めて縮退状態になりにくい運用を行うべきである。さらに今回の調査では、縮退状態でディスク故障が発生すると、エラーが正しく処理できずにパニックやストールが発生しシステム全体が停止する場合や、エラーが正しくアプリケーションプログラムに戻らないケースが確認された。縮退状態になった場合は、ディスク故障が発生していないか、ログを十分確認する必要があることが明らかになった。

本調査の結果、コミュニティカーネルでは、従来からあるバグの修正や、エラー処理系の強化が行われているものの、より複雑になったエラー処理や、発生頻度の少ないエラーに関しては新たなバグが作りこまれており、発見されたバグの数は RHEL4.5 より多い結果となった。この原因としてはコミュニティにおける評価が弱く、エラー処理系が正しく動いているか十分検証されないまま機能強化やバグ修正が行われているためと推測される。バグの作りこみを防止しない限り、品質評価を行って個別のバグ修正を行っても次々と新しいバグが作りこまれていつまでも品質が安定しない。このため本調査活動の一環として開発した模擬故障を用いた評価の枠組みを、コミュニティに認知してもらい、開発者に利用してもらうことでエラー処理系の強化を図って行くことが極めて重要であると考え。これを実現するために、故障模擬ライブラリ、及び品質評価プログラムのコミュニティ提案を行い、認知してもらう努力を継続的に行う予定である。

第7章 結論

Linux のソフトウェア RAID 機能に関する機能評価、性能評価、及び品質評価を行った結果をまとめる。ソフトウェア RAID 機能の利用者の視点からは、md の RAID1、RAID5、RAID6、RAID10、ならびに DM の RAID 機能に関する評価結果をまとめた上で、利用上の注意点を説明する。開発コミュニティの視点からは、今回開発した評価プログラムをコミュニティに提案し、今後の Linux ソフトウェア RAID の改善に向けた活動計画について述べる。

1. 機能評価

機能評価では、Linux のソフトウェア RAID 機能に対して、RAID アレイの設定・復旧・管理のために必要な機能がサポートされているか調査した。

まずインターネット上に公開されている情報等から、md、DM (dmraid)、DM (LVM2)のそれぞれについて、各機能評価項目のサポートの有無と、サポートされている場合は、その操作手順について調査を行った。調査結果は「ソフトウェア RAID 設定手順書」としてまとめた。今まで DM の操作方法やサポート範囲について、まとめて説明されたドキュメントは無かったため、DM を利用しようとするユーザにとっては今回作成した設定手順書は有用であると考える。

機能調査結果を見ると、DM (dmraid)と DM (LVM2)では md が持つ機能の、それぞれ 50%以上と、30%以上の機能が未サポートであり、先行する md には追いついていないことが確認された。DM では今後の機能強化も重要であると考える。

一方、md の機能の多くは mdadm コマンドによる操作や、/proc/mdstat ファイルの内容確認を必要としており、利用するためには mdadm コマンドの多種多様なオプションや、mdstat の内容に関する知識を持っていることを前提としている。故障ディスクが発生した場合の復旧についても、故障ディスク特定や復旧を行うために Linux コマンドを操作する必要がある。ハードウェア RAID 製品で見られるような、ディスクの LED の状態を見て物理的に差し替えるだけの操作と比べると複雑であり、初心者が使いこなすのは難しいと思われる。今後さらなる操作性や保守性の改善が必要と考える。

さらに調査した手順について実機上で、機能が正しく動作するかの確認を行った。

2.6 カーネルで導入された DM に比べると、以前から存在する md の方が機能的に充実している。RHEL4.5 カーネル上では md の RAID1、RAID5 は機能的に問題はなかったが、RAID6、RAID10 は評価中に縮退時や復旧時に問題が見られたが、最新のコミュニティカーネルではこのような問題は見られなかった。RAID6 と RAID10 は Linux の 2.6 カーネルから導入された機能であり、2.6 の初期のカーネルでは問題があったが、最新カーネルでは修正されていると推測される。

2. 性能評価

性能評価では Linux のソフトウェア RAID 機能に対して、通常運用時、縮退状態、復旧処理中

の三つの場合についての性能測定と、各 RAID レベルでの無負荷時と負荷をかけた時の復旧時間測定を行った。

その結果、大きな I/O サイズの read / write を行う用途では RAID5 を使用することで、RAID を使わない場合に比べスループットの向上が期待できることが確認できた。また RAID10 は、RAID を使わない場合と比較して、I/O サイズによらず性能向上がみられ、性能面からは有望であると考えられる。md の RAID1 や DM (LVM2) は write では多少のオーバーヘッドがあるものの、read の性能は RAID なしの時とほぼ変わらないため、read 主体の用途では性能的な問題とはならないと考える。今後アプリケーションを使った評価により確認する必要がある。

RHEL4.5 カーネルに比べて最新のコミュニティカーネルでは、全体的には通常運用時、縮退状態、復旧処理中の性能は少し改善しており、現在のコミュニティ開発でも、性能を意識して開発が進められていることがわかる。

各 RAID レベルでの復旧時間測定結果において、多くの RAID 構成で、縮退時や復旧時の性能として通常運用時とほぼ同程度の性能が得られており、RAID の復旧に伴うオーバーヘッドで通常業務に大きな性能インパクトがあることは少ない。

一方、負荷をかけた時の復旧処理時間は最新コミュニティカーネルの結果が RHEL4.5 カーネルの結果を上回っているものの、無負荷時の復旧処理時間と比べると 20 倍以上遅くなっている場合もある。縮退した RAID アレイの復旧を運用中に行う場合は、運用時の負荷を考慮して復旧作業を計画する必要がある。

3. 品質評価

品質評価では、通常運用中、縮退状態、復旧処理中のそれぞれの場合に模擬故障を発生させることでエラー処理部分の評価を行った。また故障ディスクの交換を行う場合を想定して、hotplug を実際に行い、デバイスドライバまで含めた OS として正しく動作するか評価を行った。

模擬故障評価の結果、md あるいは DM のバグと考えられる障害を、RHEL4.5 カーネルで 2 種類、最新のコミュニティカーネルでは 4 種類検出した。エラー処理部分についてはさらなる品質向上が必要と考える。

コミュニティの最新版の md では、最も一般的であるメディアエラーが発生した場合、冗長系からデータを復元して書き戻すことで、ディスクの代替セクタ割当てを行わせるよう、エラー処理が強化されていた。一方では DM のソフトウェア RAID 機能に対しては、RHEL4.5 では修正されているバグが、最新のコミュニティカーネルでもまだ取り込まれておらず、カーネルの進化に伴い品質が必ずしも向上していない場合があることが判明した。

RHEL4.5 上の各 RAID 機能については、通常運用中故障に対するエラー処理で特に問題は検出されなかった。RHEL4.5 上で RAID の運用を行っている場合は、冗長性がある状態で動かしていれば故障によるエラーが正しく処理されて故障ディスクの切り離しが行われることが確認された。一方、縮退中や復旧中にエラーが発生した場合、一部でエラーが正しく処理できずにストールやパニックが発生するケースがあった。

4. まとめと今後の予定

以下に、本調査の結果をソフトウェア RAID の利用者の視点と、開発コミュニティの視点でまとめる。

4.1. ソフトウェア RAID の利用者の視点

機能評価、及び品質評価の結果を表にまとめたものを表 37 に示す。

表 37：機能評価と品質評価結果まとめ

機能評価 ○:問題なし、×:基本機能に問題あり、△:基本機能以外に未サポート項目あり

品質評価 ○:問題なし、×:通常運用状態で問題あり、△:縮退状態のエラー処理に問題

—:対象外

	RHEL4.5						linux-2.6.22.6					
	md				DM		md				DM	
	RAI D 1	RAI D 5	RAI D 6	RAI D10	dmraid	LVM2	RAI D 1	RAI D 5	RAI D 6	RAI D10	dmrai d	LVM2
機能評価	○	○	×	×	△	△	○	○	○	○	△	△
品質評価	△	○	○	△	—	○	×	△	△	△	—	×

また今回の調査で Linux のソフトウェア RAID 機能の設定・復旧・運用方法に関してまとめた手順書を作成した。これにより復旧作業の効率化によるダウンタイムの短縮が期待される。

縮退した状態で、さらに故障が発生すると、データを復旧することは極めて困難となる。このため、縮退が発生した時点で、速やかにシステムを止めて故障ディスクの交換ができる運用を行うか、冗長性を高めて縮退状態になりにくい運用を行うべきである。さらに今回の調査では、縮退状態でディスク故障が発生すると、エラーが正しく処理できずにパニックやストールが発生しシステム全体が停止する場合や、エラーが正しくアプリケーションプログラムに戻らないケースが確認された。縮退状態になった場合は、ディスク故障が発生していないか、ログを確認する必要があることが明らかになった。

Linux のソフトウェア RAID 機能のうち、コミュニティカーネルの md の RAID1 と DM (LVM2) については、通常運用中のディスク故障でも、ケースによってはエラーが正しく処理できない場合があることが判明した。最新カーネル上でこれらの機能を利用する場合は、問題が修正されるまではサーバ障害が起きた場合の対策を検討しておく必要がある。

現在最も広く使われていると思われる、RHEL4 上の md の RAID1 及び RAID5 に関しては、通常運用時のエラー処理では問題が検出されなかった。一方、md の RAID6 と RAID10 に関しては、RHEL4 カーネル上の機能評価で問題が検出されており、RAID6 と RAID10 を利用する場合は新しいコミュニティカーネルをベースとしたディストリビューションのものを利用する必要がある。

Linux の 2.6 カーネルで新規に追加された DM は、ソフトウェア RAID の機能については、未サポートの機能が多く、まだ先行する md には追いついていないことが確認できた。

4.2. 開発コミュニティの視点

コミュニティカーネルでは、従来からあるバグの修正や、エラー処理系の強化が行われている

ものの、より複雑になったエラー処理や、発生頻度の少ないエラーに関しては新たなバグが作りこまれており、発見されたバグの数は RHEL4.5 より多い結果となった。この原因としてはコミュニティにおける評価が弱く、エラー処理系が正しく動いているか十分検証されないまま機能強化やバグ修正が行われているためと推測される。

バグの作りこみを防止しない限り、品質評価を行って個別のバグ修正を行っても次々と新しいバグが作りこまれていつまでも品質が安定しない。このため本調査活動の一環として開発した模擬故障を用いた評価の枠組みを、コミュニティに認知してもらい、開発者に利用してもらうことが極めて重要であると考え。Linux のソフトウェア RAID 機能を改善して行くために、今回開発した故障模擬ライブラリ、及び品質評価プログラムをコミュニティに提案する予定である。

しかし開発した評価プログラムや評価結果を大きな塊としてメーリングリストに投稿しても、受け入れられ難い。そこで、今回発見した具体的な問題点を成果として示しつつ、最初に模擬故障を利用したエラー処理系の評価手法をまず認知してもらい、次にそれを利用した個々の問題点を解決して行くことを計画している。今後も上記活動を通じ、コミュニティのメンバとして Linux のソフトウェア RAID 機能を改善する努力を継続的に行う予定である。

参考文献

- [1] 「Red Hat Enterprise Linux 4 システム管理ガイド」、
<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/ja>
- [2] AT Attachment with Packet Interface – 7, Volume 1 – Register Delivered Command Set, Logical Register Set (ATA/ATAPI-7 V1), T13 working draft, Revision 4
- [3] Serial ATA II: Extensions to Serial ATA 1.0a, The Serial ATA International Organization
- [4] SCSI Block Commands – 3 (SBC-3), T10 working draft, Revision 8a
- [5] SCSI Primary Commands – 4 (SPC-4), T10 working draft, Revision 10
- [6] SCSI Architecture Model – 4, T10 working draft, Revision 10
- [7] <http://www.kernel.org/pub/linux/kernel/people/agk/patches/2.6/editing/dm-raid1-handle-read-failures.patch>

付録 A RAID の概要

Linux カーネルが提供するソフトウェア RAID 機能は、現在のところ、RAID0、RAID1、RAID4、RAID5、RAID6、RAID10 のレベルがある。この内 RAID0 はデータを複数のディスクにストライピング格納することで、ディスク容量やスループットを向上する技術であるため、耐故障性を向上させる目的では、RAID1、RAID4、RAID5、RAID6、RAID10 のレベルがある。以下では、各 RAID レベルについて、データ格納の方法を簡単に説明する。

なお RAID4 ではパリティディスクへのアクセス集中という問題があり、ほぼ同等の機能を持つ上この問題のない RAID5 で代用できることから、あえて RAID4 を選択して使用する場合はないと判断し、評価対象には含めなかった。

(1) RAID1

図 23 のとおり、データを複数台のディスクに同時に書き込む。このため、「ミラーリング」とも呼ばれる。アクセス速度の向上はないが、ディスク故障に起因するデータの損失やシステムの停止が発生しない。一方のディスクが故障した場合、もう一方のディスクに自動的に切り替わりデータが処理されるため、動作は継続される。但し、ディスクの容量全体のうち、半分しか実際には使用できないため、容量の利用効率は低い。

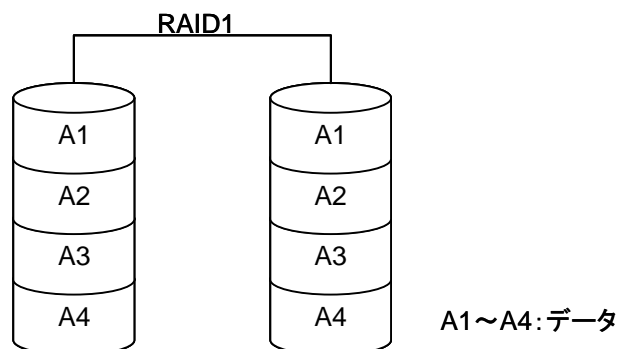


図 23 : RAID1 構成

(2) RAID4

RAID4 は図 24 のように誤り訂正符号データ(パリティデータ)により、データを再生成する機能を持たせるため、耐故障性は向上する。しかしデータ書き込み時にパリティディスクに負荷が集中しやすいという欠点を持つ一方、後述の RAID5 が同等の冗長性を持ちつつ、負荷集中の問題を解決しているため、実際上は RAID4 が使われることはあまりなく、代わりに RAID5 が用いられる。

このため、今回の評価でも RAID4 に関しては対象外とする。

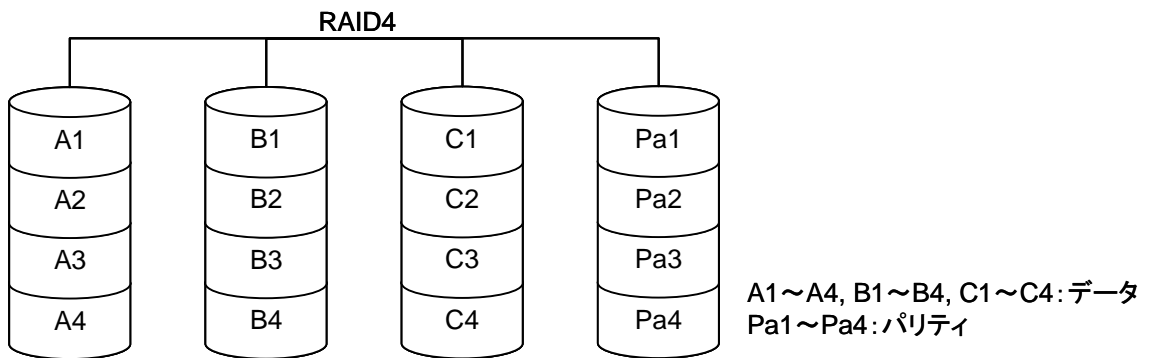


図 24 : RAID4 構成

(3) RAID5

誤り訂正符号データ(パリティデータ)により、データを再生成する機能を持たせるため、耐故障性は向上する。図 25 のとおり、データはブロック単位に分割し、パリティデータを全てのディスクに分散して配置する。但し、2 台以上のディスクが同時に故障すると、回復することができない。

データ更新時には必ず更新前のデータとパリティデータを読み出し、更新パリティデータを作成後書き込むといった余分なアクセスが必要になる。これを「write ペナルティ」という。更新するパリティデータは異なるディスクに配置されているため、write 処理が多重で発行された場合も同時に実行することが可能である。これにより、パリティ専用ディスクのみに負荷が集中することを防いでいる。

パリティの保存に必要なのは、全ディスク台数に関係なくディスク 1 台分の容量となる。このため、ディスク台数が多いほど容量の利用効率も向上する。

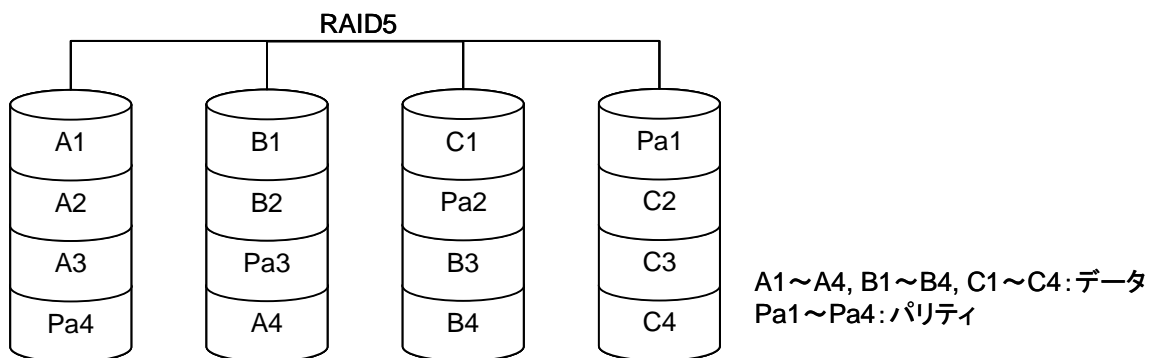


図 25 : RAID5 構成

(4) RAID6

図 26 のとおり、2 種の誤り訂正符号データ(パリティデータ)を異なるディスクに配置すること(ダブルパリティ)により、同一 RAID グループ内の 2 台のディスク故障まで救済することが可能であるため、RAID5 と比較して耐故障性は向上する。

RAID5 と同様に「write ペナルティ」が必要となるが、更新するパリティデータは異なるディスクに配置されているため、write 処理が多重で発行された場合でも同時に実行することができる。

パリティ用に2台分のディスク容量を必要とするため、RAID5と比較してディスクの利用効率は劣る。

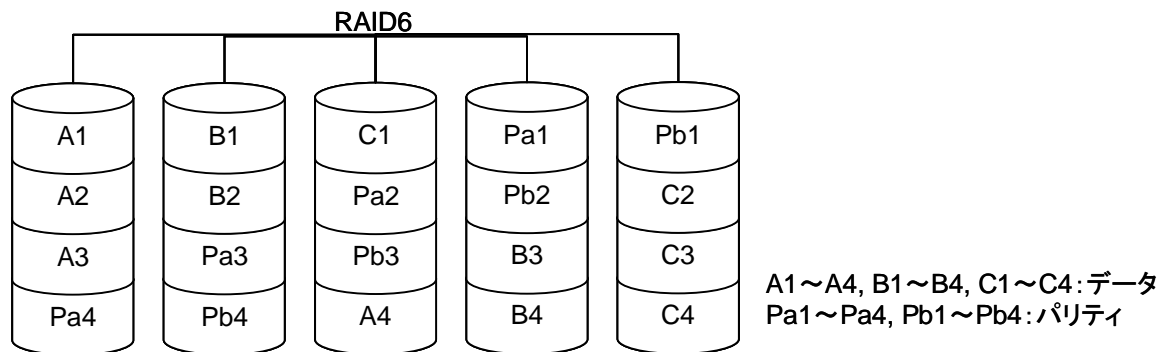


図 26 : RAID6 構成

(5) RAID10

RAID10は図27のとおり、RAID0とRAID1の組み合わせを1つのアレイに構成することで、RAID1によるデータ二重化と、RAID0の高速化を合わせて実現している。信頼性とI/O性能の高い記憶装置となるが、コストはかかる。

RAID0は、分散されたデータに同時に並行してアクセスできるためアクセスが高速となるが、誤り訂正符号データ(パリティデータ)を持たないため、故障時にデータを再生成する機能はない。ディスクが1台でも故障するとデータの読み書きが不可能となり、ディスクの台数が増えると故障率も高くなる。

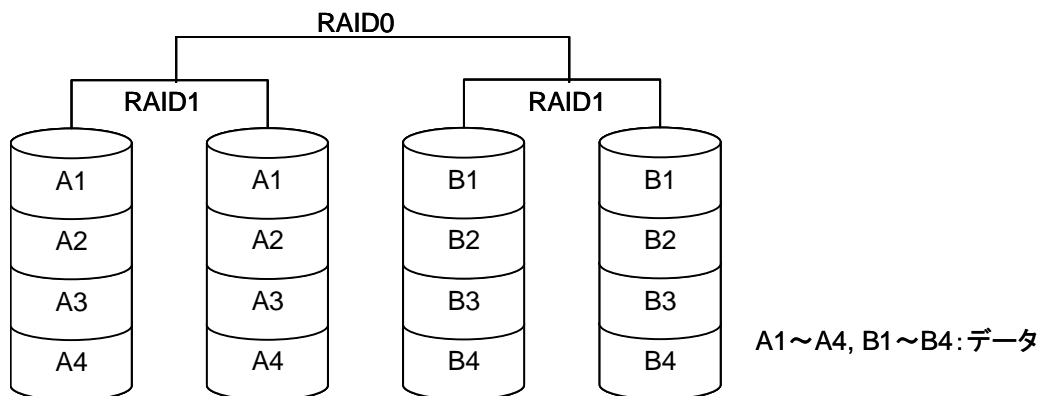


図 27 : RAID10 構成