

オープンソースソフトウェア ロードマップ

Ver 1.0

2007年3月1日

日本OSS推進フォーラム

サーバー部会 ロードマップタスクフォース

<http://www.ipa.go.jp/software/open/forum/>

目次

1. はじめに	2
1.1 本資料の目的	2
1.2 本資料で扱う範囲	2
2. 策定指針	3
2.1 OSS の開発状況分析	3
2.1.1 状況分析に関する観点	3
2.1.2 OSDL DCL WG 作成ドキュメントによる分野別動向分析	3
2.1.3 技術的背景から見た OSS 開発の流れ	6
2.2 開発状況分析に基づく RMTF の方針について	9
2.3 ロードマップ	11
3. 強化策の提案	14
3.1 強化項目の DCL 報告書における位置付け	14
3.2 強化策の提案	16
3.2.1 Xen の基幹システム適用に向けた開発	16
3.2.2 メッセージマニュアル データベースの作成	17
3.2.3 Linux カーネル互換性(リグレッション)テストツール	19
3.2.4 クラスタファイルシステムの評価	21
3.2.5 Linux ディスク冗長化機能の強化	22
3.2.6 統合運用管理ツール Hinemos の強化	23
4. まとめ	26
付録	27
付録 A DCL-WG 「DCL Goals and Capabilities」	27
付録 B 関連 URL 集	34
ロードマップ TF メンバー一覧	36
商標について	37

1. はじめに

1.1 本資料の目的

2009年にオープンソースソフトウェア(以下、OSS)を市場のトップシェアにすることを目標に掲げ、日本政府・民間が協力して推進する日本OSS推進フォーラムのサーバー部会・ロードマップタスクフォース(以下、RMTF)が2006年に発足した。

このタスクフォースでは、日本のOSSに関わる主要な企業が集まり、日本のユーザからの要望をふまえながら、OSSのミッションクリティカル(以下、MC)システム分野への適用領域を拡大することを目標として、現時点でのMC領域でのOSSの課題の抽出、分析を行った。

その成果として今回、主にOSS開発者に向けて提案することを目的としたOSSサーバ技術のロードマップを策定した。また、その中のいくつかの項目について、具体的なOSS強化策を提示した。

今回、OSSサーバ技術ロードマップとOSS強化策とを示すことにより、我々が提案するOSSの方向性を日本のみならず、世界に発信し、開発者間のコミュニケーションツールとして活用してもらうことを想定している。同様な思いをもったOSS開発者が協力してOSS技術の検討・開発を行えるよう本資料を媒介として情報の共有化を図ることにより、効率的なOSS開発を促し、OSSでのMCシステム構築の早期実現を促すのが本資料の主旨である。

1.2 本資料で扱う範囲

MC領域のサーバ分野において、OSSをさらに普及させる上で現時点では実現されていないが、シェア向上に有効な機能項目を抽出した。特に、Linuxカーネルおよび基盤となるOSSにおいて、企業およびコミュニティの両者にとって、インセンティブが低いために開発に着手しにくい事項に着目する方針とした。加えて、これらのOSSの機能項目の評価を行った。

機能項目の評価にあたっては、OSS開発強化策といった開発の側面だけでなく、現場で利用していく上で必要な評価や品質向上のための方策の提案、OSSシステムの運用を簡便化、効率化するための運用策の提案、といった機能開発フェーズから運用フェーズにいたる範囲について考察を試みた。

2. 策定指針

2.1 OSS の開発状況分析

2.1.1 状況分析に関する観点

本章では、OSS の動向を二つの面から考察する。

一つ目は、OSDL(Open Source Development Labs、現 The Linux Foundation)DCL WG(Data Center Linux Working Group)作成のドキュメント「DCL Goals and Capabilities」(以下、DCL 報告書)を利用し、分野別に OSS 開発状況を開発背景及びコミュニティ状況を中心に俯瞰的に分析した。これは、カテゴリ別に開発状況を押さえることにより、RMTF で注目すべき分野を見極めることが可能になると考えたからである。

二つ目は、技術的観点として、プラットフォームのここ数年の進歩状況をベースとした開発状況を分析した。この分析により、プラットフォームの進歩と同期が取れていない部分、ハードウェアのサポートが不足している部分が明確になり、RMTF からの提言のベースになることを期待している。

2.1.2 OSDL DCL WG 作成ドキュメントによる分野別動向分析

本節では、MC 分野に関連する OSS について、その技術分野、技術達成度を探り、今後の開発の指針とする。このような目的で OSS の動向に絞って調査した文献として現在のところ広く知られているものは少なく、もっとも近いものとして OSDL の DCL WG が策定した DCL 報告書が考えられる。WG の名称には、DCL、即ち Data Center Linux が含まれているが、WG の目的とするところは DCL に留まらず、より一般的な MC 分野に Linux を適用するために必要な OSS の具体的な機能面の充足状況を該当分野の代表者を中心に、ユーザ、ISV の声を調査し、まとめたものである。

DCL WG では、DCL 報告書の策定にあたって、機能を

- Scalability
- RAS (Reliability, Availability, Serviceability)
- Performance
- Management
- Cluster
- Standards
- Security
- Usability
- Virtualization

の9つに分類し、それぞれの中小項目毎に、コミュニティの説明、進捗状況を調査している。各技術項目の状況について、技術達成度(Maturity)、コミュニティでの活動状況を調査している。この資料を参照することにより、各技術の状況が一望出来ると思われる。また、この9つの分類は、事前のマーケティング調査の結果から策定されたものであり、それを参考にすることにより、RMTF 開発項目の市場におけるポジションを明確にしてくれるものと期待している。即ち、RMTF で取り上げる開発候補項目を、DCL 報告書と比較・検討することにより、位置付け・開発の意義が見えてくると期待している。尚、DCL 報告書は、Virtualization に関する部分は 1.2(2006 年発行)、その他の部分の記述については 1.1(2005 年発行)を参照している。

付録 A-表 A.1 にある DCL 報告書で記述されている技術達成度(Maturity)を表 2.1 に示す。技術達成度は、9 つに分類されており、各々の意味は以下の通りである。

表 2.1 技術達成度(Maturity)

(a)	Investigation	該当プロジェクトは、まだ概念検討レベルである
(b)	Development	該当プロジェクトが開始された
(c)	Released	初期リリースが出された
(d)	Usable	動作する使用可能なリリースが出た
(e)	Stable	安定版リリースが出た
(f)	Integrated	該当プログラムは、開発中の Kernel の Tree に入ったか、もしくはパッチとしてディストリビューションに入った
(g)	Mainline	該当プログラムは、安定 Kernel の Tree に入ったが、まだ完全には使用出来ない。将来、ディストリビューションに入ることが期待されている
(h)	Product Available	仕様を充足し、カスタマーが使用可能なリリースが出た
(i)	Complete	複数のディストリビューションで、仕様を満足し、カスタマーが使用可能なリリースが出た

本書では、DCL 報告書に基づき、分野別に各技術項目の概要、技術達成度(Maturity)を評価した。その上で、実システムへの適用が容易でない段階(Investigation / Development / Released / Usable / Stable)にマーク(網かけ)することにより、MC 領域からみた達成度が見えるようにした(付録 A-表 A.2 - A.9)。以下、付録の表に基づいて各分野別の状況を概観し、今後の RMTF における提案項目検討のベースにしていく。

[< >内は分類項目、()内は対応する表を示している]

< Scalability > (付録 A-表 A.2)

- (1) Priority 1 の大半が Mainline(Kernel Tree に取り込まれている)か、それ以上の達成度となっている。
- (2) NUMA 関係のいくつかの項目が Product Available/Completed にとどまっており、ハイエンドのハードウェアサポートについて遅れがある。
- (3) Priority2 の項目について見ると、ストレージ台数、サイズ、ネットワーク関係、大容量メモリ関係を中心としたハイエンドのハードウェアサポートが遅れている。

< Performance > (付録 A-表 A.3)

- (1) 本カテゴリは、多くが性能測定、評価を行うための項目であるが、特にアプリケーションに近い項目が Usable レベルとなっており、今後適切な対応が必要と考えられる。
- (2) Hyperthread Technology (以下、HT) 等の新しいハードウェア対応の性能についても、遅れている。この理由としては、対象がファイルシステムからネットワーク、アプリケーションと広いこと、測定ツールが普及していないこと、それらによりきちとした性能解析が行えないため性能改善が進まないこと、などが考えられる。

< RAS > (付録 A-表 A.4)

- (1) 本カテゴリでは、かなりの項目で達成度が低い。Priority1 の項目で見ると、ダンプは Stable レベルとなっているが、トレーサーは Released レベルであり、またホットスワップはメモリとノード関係が Released レベルである。
- (2) Priority2 の項目では、ホットスワップ関係の大半の項目、メモリ消費対応、フェールオーバー関係、ファイルシステム関係が遅れていると考えられる。

< Manageability > (付録 A-表 A.5)

- (1) 主にインストールの際に必要な機能について述べているが、こちらも Usable レベルが多い。パッケージ管理の基本機能は問題無いが、リモート関係等の拡張部分では、いまだ Usable が多い。
- (2) Priority2 の中では、資産管理を中心とした機能が不十分となっている。これは、インストール機能が各ディストリビューション固有になっていること、リモートインストールがまだ一般的でないことによるものと考えられる。

< Clusters > (付録 A-表 A.6)

- (ア) Usable 以下の項目が多くある。クラスタ環境は、コミュニティでは開発環境整備が困難な面が多いことも一因と考えられる。

< Standards > (付録 A-表 A.7)

- (1) LSB 対応については、Mainline(Kernel Tree に統合済み)で、問題無いと思われる。
- (2) Maturity が低いのは CIM, SAF AFS 等であり、緊急性を有しないとされる。

< Usability > (付録 A-表 A.8)

- (1) コマンドラインインタフェースについては、各ディストリビューション間で揃ってきたが、ISV が各ディストリビューションに対応しやすいか、という観点では、まだ達成していない。

< Virtualization > (付録 A-表 A.9)

- (1) OSS だけではなく、商用プロダクトとして提供されているものを含めれば、ハードウェアデバイスに特定された機能サポートを除けば、かなりの項目で使用可能とされている。
- (2) 商用プロダクトとしてのみ提供されている部分について、OSS での機能充実が求められる。例えば、IA64 等の 64 ビット系プラットフォームのサポート、ホットプラグ、各種リソース管理、開発環境などである。

2.1.3 技術的背景から見た OSS 開発の流れ

本節では、RMTF が MC 分野における重要技術と考えたプロジェクトについて、プラットフォームの技術的な流れと今後の動向をまとめた。図 2.1 は、プラットフォームの技術的な流れを示しており、これによれば、HT を出発とした CPU のマルチコア化、インテルバーチャライゼーション・テクノロジーに始まる仮想化の流れが重点を占めている。これ以外には、メモリや PCI バス、ネットワーク等の高速化及びホットプラグの流れが、現在の技術的な流れの中心になっている。この大きく 3 つの流れに対して、RMTF が重要とみなす開発プロジェクト状況を示したものが、図 2.2 である。

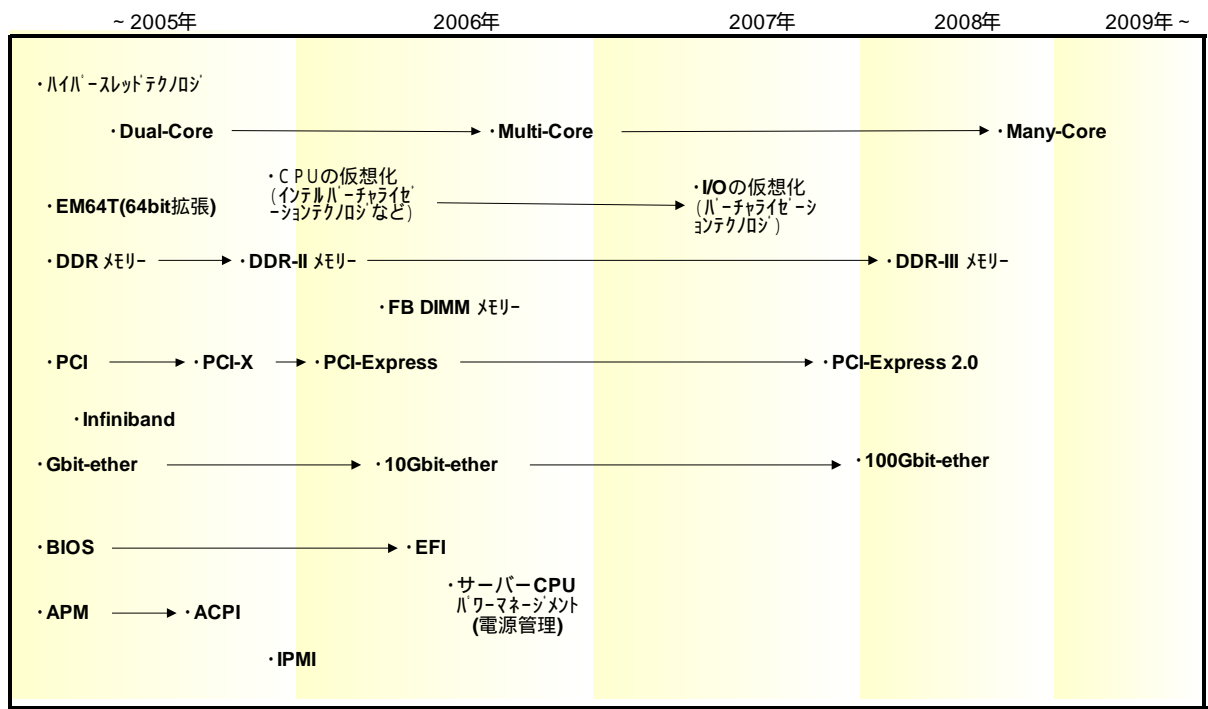


図 2.1 テクノロジー背景

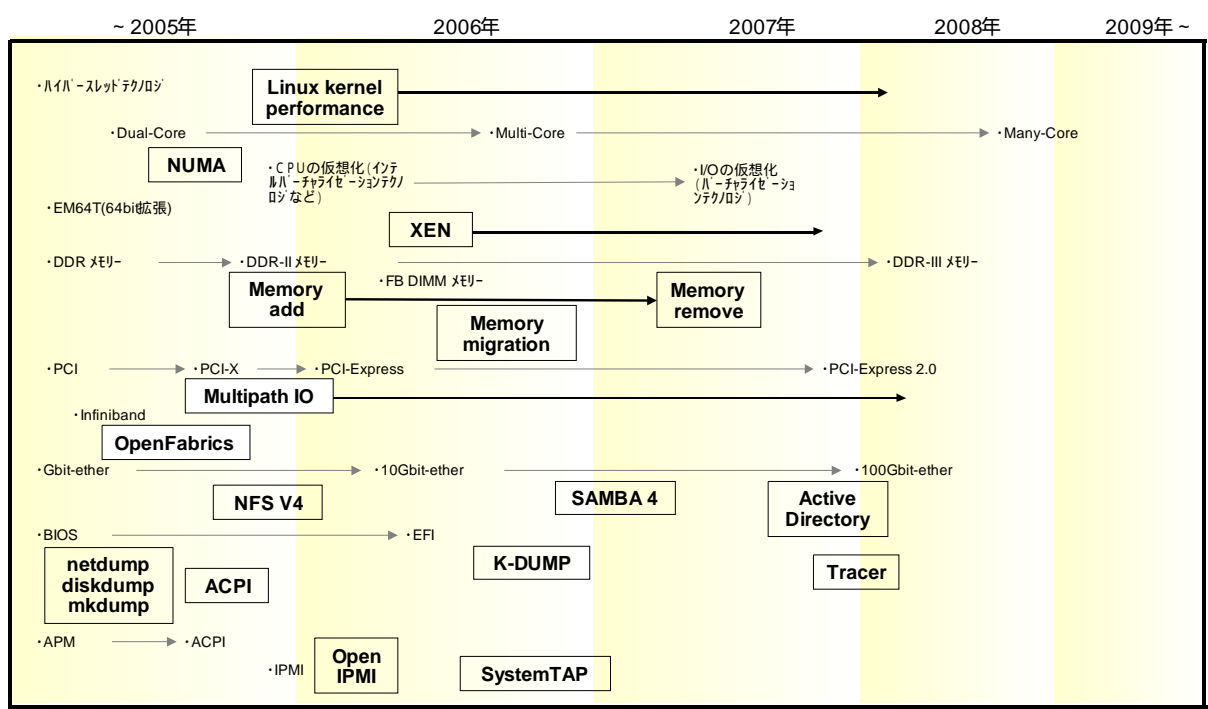


図 2.2 開発プロジェクト

(1) CPU のマルチコア化

Linux Kernel は、既に HT には対応済みであり、マルチコアについても、Scalability という観点から対応が進行中である。実際、DCL 報告書によれば、CPU は 8 Way まで対応済みである(Linux Kernel では、Multi-Core は、Multi-CPU と同等に扱われる)。またそれ以上の Scalability に関しても開発プロジェクトが進んでいる。

(2) 仮想化について

Xen を中心に、仮想化技術の対応が進んでいるが、VMI の議論も進んでおり、より汎用的な仮想化技術のサポートへ進展していくことが期待される。また、ハードウェア面においてもインテルバーチャライゼーション・テクノロジーへの対応等、着実に進んでおり、これらの新技術に対する Kernel でのサポートも進行中である。

(3) ホットプラグ

DCL 報告書には RAS に含まれている CPU、I/O 周辺機器、メモリのホットプラグが含まれる。CPU を中心として、基本的なサポートは進んできているが、I/O 系に関するサポートが途上であり、今後の注力が待たれる。また、メモリのホットプラグについても今後の開発を待つことが必要である。

(4) 高速化

DCL 報告書では、高速化への対応は Scalability に含まれているが、高速ネットワークを中心に、サポートは十分に進行していると考えられる。

(5) その他

RAS 分野では、障害解析関係のダンプ、デバッガー、トレーサー関係のツール開発が進んでいる。表にはないが、プラットフォーム毎に開発の進捗度合いが異なるため、注意が必要である。ファイルシステム強化に関しては、Ext4 関係の議論が進行中である。

2.2 開発状況分析に基づく RMTF の方針について

RMTF では、まず MC 分野で開発が必要な技術分野・項目のリストアップを行い、そのリストアップした項目から、

- (1) コミュニティの興味を引かないもの、もしくはコミュニティの体力ではなかなか実現が困難なもの
- (2) 企業として重要性・必要性は感じているが、着手しても経済的魅力が少なく、ビジネスとして成立しにくいために、投資になかなか踏み切れないもの
- (3) MC 実現には必要性が高く、実現すればコミュニティ、企業、ユーザにとって有益と考えられるもの

という観点から MC 分野で必要とされており、かつ比較的手が付いていない分野に注目するためスクリーニングを行った。その結果を表 2.2 に示す。

表 2.2 開発項目リスト

	カテゴリ	開発強化項目	概要
(1)	ファイルシステム	ローカル/クラスタファイルシステム	OSS クラスタファイルシステムにおいて、どの程度の負荷やマシン構成に耐えうるかといった基礎情報やシステム規模や用途の有効性を把握するための性能特性の評価基準設定、評価、考察
(2)	トラブルシュート	メッセージマニュアル データベース	Linux で出力されるメッセージの調査結果(意味・対応するカーネル/ソースコード・対処方法等)をメッセージマニュアルとしてデータベース化し、誰でも簡単に参照/更新できる Linux メッセージ検索システムを Web 公開
		ライブパッチ	現在実行中のプログラムやカーネルを停止することなく、実行状態のまま確実に更新するための技術の開発
		メッセージ ID	出力されるメッセージに基づくトラップ、アクション実行の構築容易性のために、ID を付与(誤り防止など)
		シスログ情報収集	複数サーバで出力されたログ情報を一箇所に収集する機能。一元管理をインフラとして提供

(3)	バッチ処理	JCL コンバータ	メインフレーム OS の JCL の基本部分をシェルに変換するツール、または変換のガイドツール
		リソース予約	ジョブ実行前に必要とされるファイル、CPU、I/O 帯域、メモリなどを予約する機能の開発
(4)	仮想化技術(Xen)	RAS 分野 デバッグ関係	仮想化技術(Xen)の基幹システムにおける RAS 機能、トラブルシューティング機能の評価。Xen のクラッシュ時のダンプ機能およびサーバの部品故障時の予防保守や縮退運用を行うリカバリ機能の開発
(5)	ドライバ関係	ホットプラグ	CPU、メモリ、メモリホットリムーブなど、ホットプラグが実現されていない機能の開発
		ドライバ互換共通 API	デバイスドライバが Kernel バージョンに依存した作りとなっており、機能追加や障害修正のための Kernel アップデートが、対応するデバイスドライバ提供の有無に左右される。Kernel とデバイスドライバが独立して開発/修正できるような共通 API を開発する
		ディスク冗長化	Linux のソフトウェア RAID 機能に対するエラー処理を含めた網羅的な評価、および品質強化
		ドライバーエラー処理ガイドライン	ハードエラー発生時の障害検出およびシステムへの通知方法/通知タイミング/ 通知内容等を共通化し、原因究明に必要なエラー情報を適切に出力できるようにする。また、適切な回復処理を行うためのガイドラインを作成し、速やかな障害回復を可能とする
(6)	データベース関係	共有ディスクのクラスタデータベース	アプリケーションの可用性、堅牢性を実現するソリューションとしての共有ディスク・クラスタデータベース管理システムの開発
(7)	リグレッションテスト	リグレッションテスト	「フレームワーク作成」と個別の「テストプログラム作成」という二つのフェーズから Linux カーネルのリグレッションテストツールを作製し、新規のカーネルのリリース毎に実行・評価

(8)	統合リソース管理 (Hinemos)	統合リソース管理	運用管理に必要な追加機能(ネットワークサービスの監視など)の開発、監視ノウハウ情報のデータベース化、基幹システムモデルにおける適用評価など
-----	-----------------------	----------	---

2.3 ロードマップ

2.2項でリストアップした開発項目を時間軸に展開すると図 2.3 のようになる。各項目は、「開発項目」、「運用改善項目」、「評価・品質向上項目」に大まかに分類される。

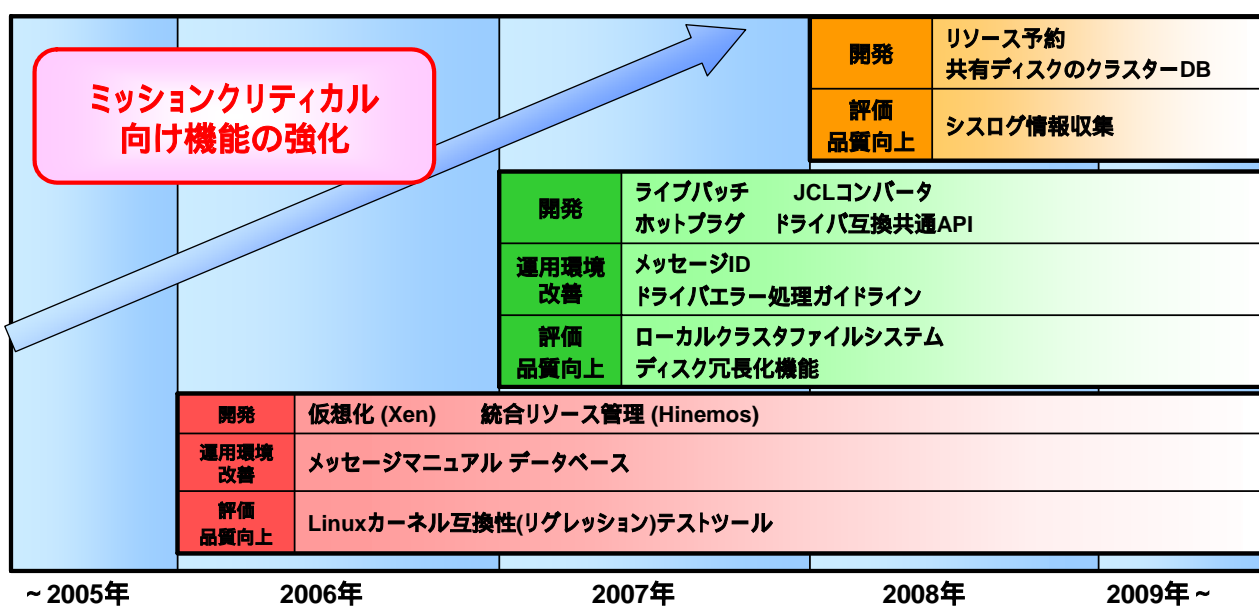


図 2.3 OSS 開発ロードマップ (RMTF 検討項目)

また、これまでに日本 OSS 推進フォーラムが提言、あるいは加盟各社が開発を進めてきた項目や、独立行政法人情報処理推進機構 (IPA) のオープンソースソフトウェア活用基盤整備事業の成果を図 2.4 に時間軸に沿ってプロットする。

また、これをふまえ、ロードマップとして各開発項目を詳細にプロットしたものが、図 2.5 である。

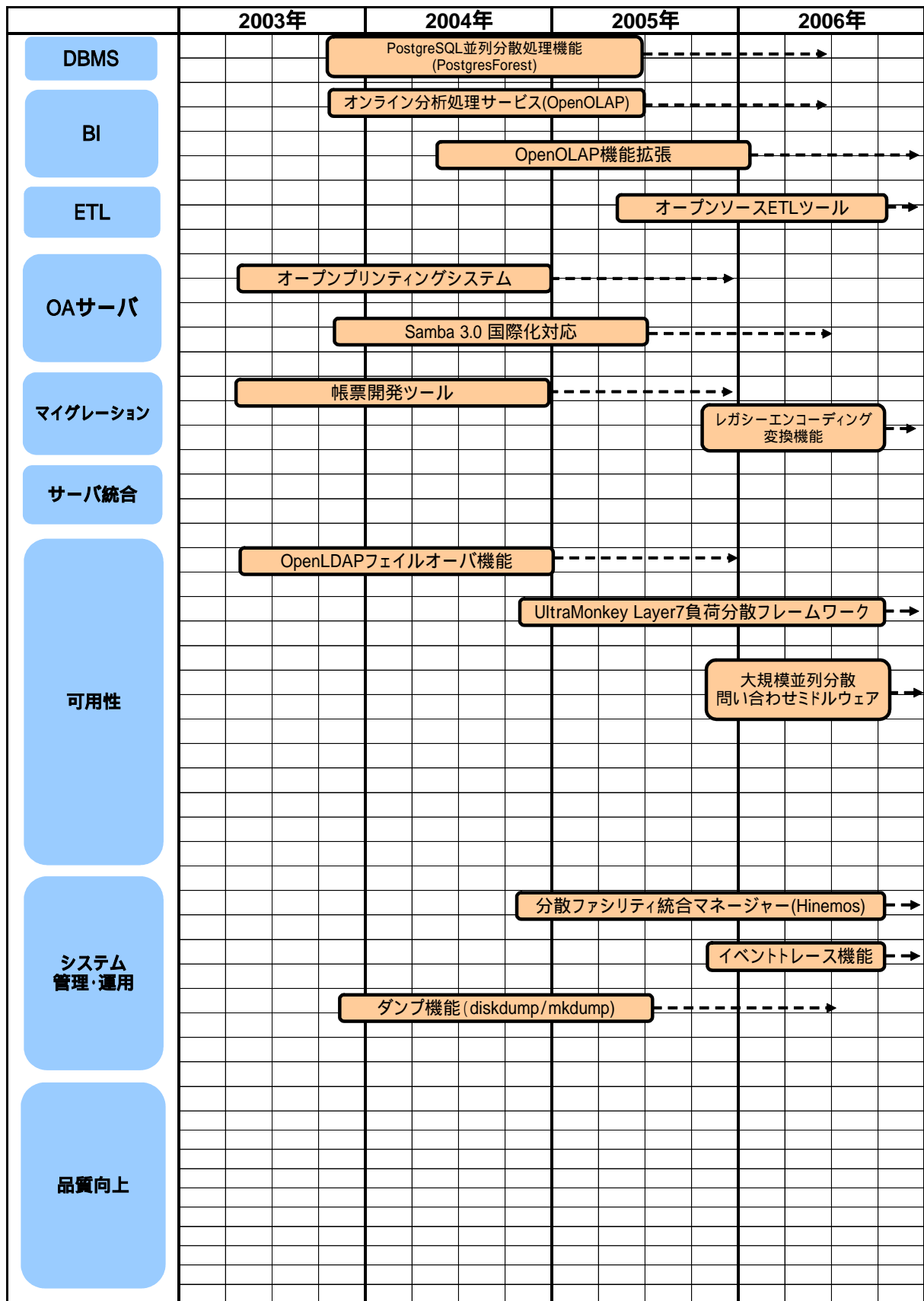


図 2.4 OSS 開発ロードマップ(着手済みの開発項目)

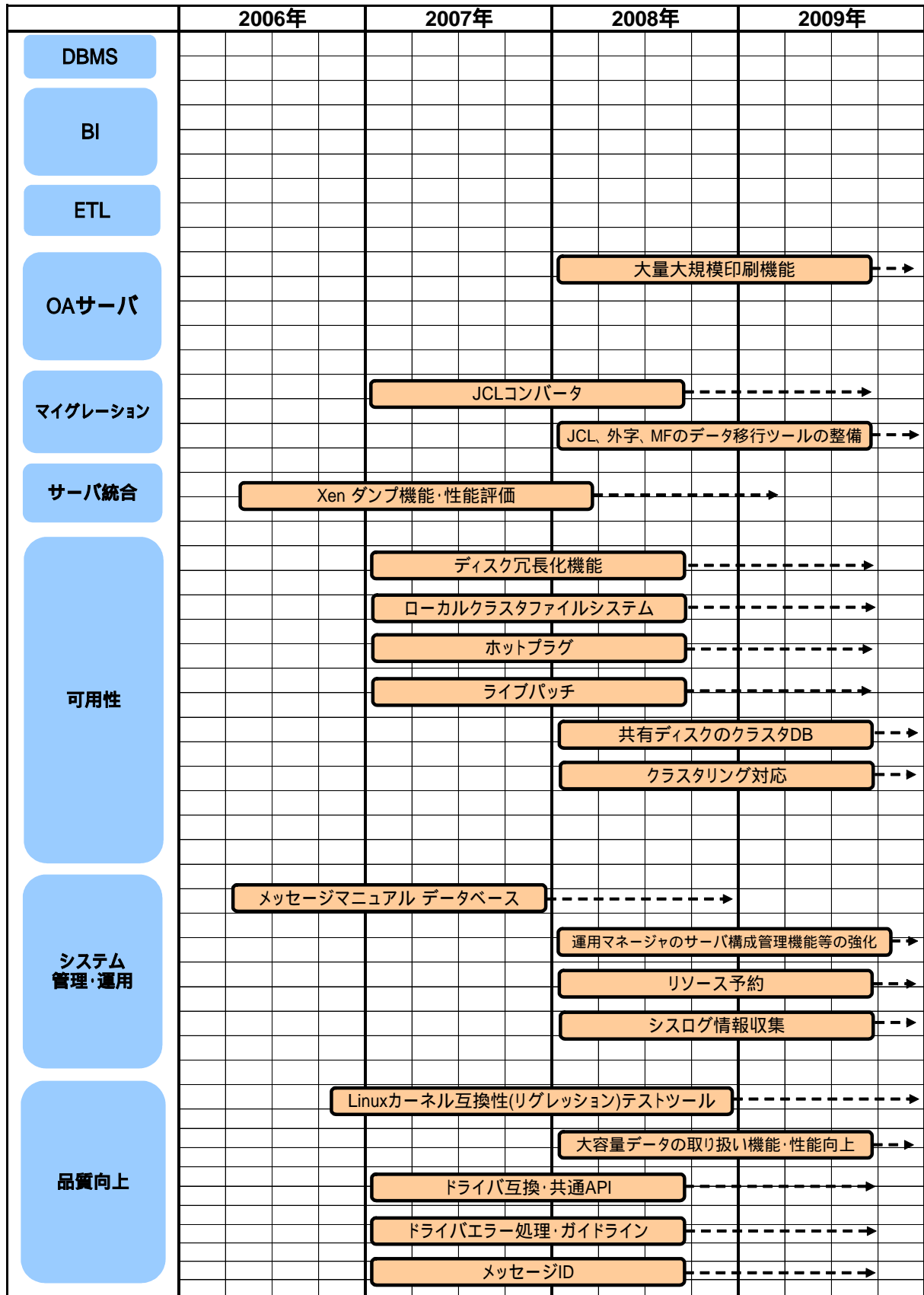


図 2.5 OSS 開発ロードマップ(RMTF 検討項目)

3. 強化策の提案

3.1 強化項目の DCL 報告書における位置付け

表 2.2 開発項目リストの開発強化項目の候補を、2.1.2 で述べた技術的動向および優先度・緊急度をもとに考慮し検討した結果、以下の 6 項目を強化項目として詳細な検討を行うことにした。

- (1) ローカル/クラスタファイルシステム
- (2) メッセージマニュアル データベース
- (3) 仮想化
- (4) ディスク冗長化
- (5) リグレッションテスト
- (6) 統合リソース管理

まず、この 6 項目の DCL 報告書における位置付けを把握するため、RMTF の検討項目と DCL 報告書の機能項目との包含関係を図 3.1 にまとめた。この包含関係を、各項目の重要性、今後の方向性の検討のベースとした。

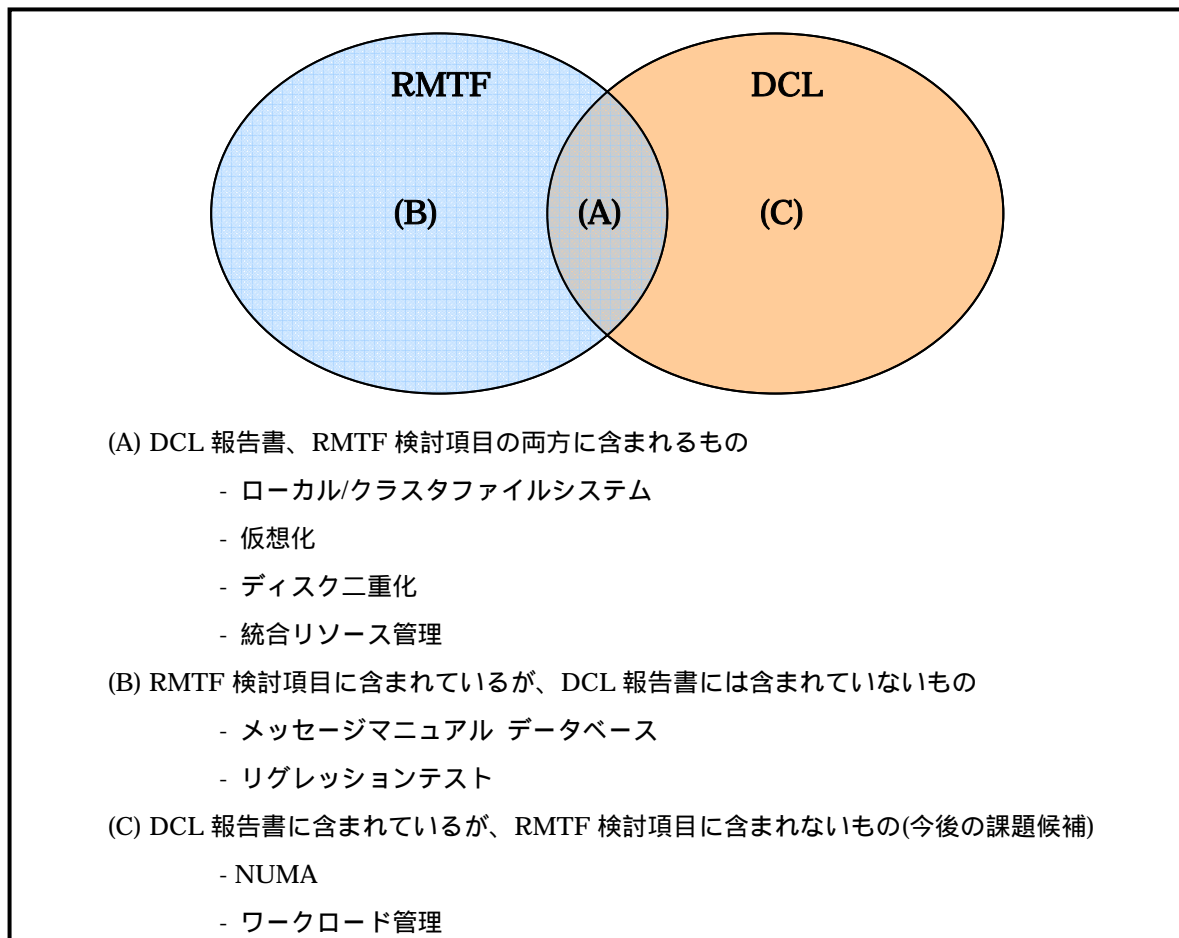


図 3.1. DCL 報告書機能項目と RMTF 検討機能項目との包含関係

2.1 章をベースに、上記の各項目について状況の評価結果を下記にまとめた。[()内は、DCL 報告書における対応する表を示している]

(1) ローカル/クラスタファイルシステム (付録 A-表 A.2 - 9)

DCL 報告書では、ローカルファイルシステム、ネットワークファイルシステム、クラスタファイルシステム等、全般に渡って項目を挙げている。また評価対象も多岐に及び、性能の面だけでなく、アトミック書き込み等の機能や信頼性運用まで幅広く取り上げている。ファイルシステムはシステムの基本であり、コミュニティ、企業の両者にとって重要である。共通の評価基準で、機能、性能、信頼性のアセスメントを継続的に行い、強化・改善をしていくことが必要である。

(2) メッセージマニュアル データベース

本項目は DCL 報告書には含まれておらず、関心が低いことが伺える。しかし、実際のシステム構築、運用中における障害解析等においては、システムが出すメッセージが最初の重要な手がかりであり、軽視することは出来ない。

本分野は、地味な作業であり、コミュニティがともすると消極的になりがちな領域でもあり、企業からしてもリソース投入の判断が難しい項目である。そのため、RMTF として開発する意味が出てくると考えられる。

(3) 仮想化 (付録 A-表 A.9)

2.1.2 でも述べたように商用プロダクトでの提供を含めた仮想化サポートはかなり進んでおり、十分使用可能なレベルにあると考えられる。しかし OSS での提供の観点から見ると、機能提供が進んでいない分野も多く、特に IA64 等の 64 ビットプラットフォームのサポート強化、ホットプラグ、各種リソース管理、開発環境における OSS 提供が重要と考えられる。

(4) ディスク二重化 (付録 A-表 A.4)

DCL 報告書にも多数項目が掲載されており、その重要度が確認出来る。DCL 報告書では、MultiPath I/O, Volume Manager, Hotswap 等、幅広い項目を網羅しており、RMTF の具体的な内容と比較検討する必要があるが、その重要性を認識しているという点では一致している。

(5) リグレッションテスト

DCL 報告書には含まれていないが、OSDL では独自にリグレッションテストスイートの開発を行っている。また OSDL のみならず、FSG (Free Standards Group、現 The Linux Foundation) のテストベッド等も存在し、その重要性が理解出来る。

(6) 統合リソース管理 (付録 A-表 A.5)

付録 A-表 A.5 によれば、インストール、リモートコンソール等の比較的初歩的な管理ツールは実装が進みつつあるが、ジョブ管理、リモート管理等の実装はまだ時間を要すると思われる。ま

た管理対象の増加に対処するため、集中的な管理等を行うことができる仕組みを併せて提供することが重要である。

3.2 強化策の提案

3.1 項での検討をふまえ、強化項目として取り上げた 6 項目について強化策の提案を下記に述べる。

3.2.1 Xen の基幹システム適用に向けた開発

(1) 必要性

サーバ仮想化ソフトウェア(以下、VM)は、サーバ統合による運用管理コストの削減手段として急速に普及してきた。今後、以下のメリットから企業システムの標準的なインフラとして普及が加速すると考えられる。

1) ハードウェアの有効活用：

1 台のサーバ上で複数の OS と業務アプリケーションを同時に稼働させて、ハードウェアを有効活用する。

2) ユーザ資産の長期保障：

新サーバへの移行時に、VM がハードウェアの非互換を吸収することにより、ユーザはソフトウェア資産(OS やその上のアプリケーション)に手を加えることなく継続利用できる。

サーバ仮想化ソフトウェアの現状としては、商用 ISV ソフトウェアである VMware 社の VMware に加え、Linux コミュニティで急速に支持を広げている OSS の Xen を Linux ディストリビュータなどが提供開始するとともに、Microsoft 社は次期 Windows サーバでの Windows Hypervisor 標準提供を表明している。

(2) 現状の問題点

Xen は開発から日が浅いことから(初版公開が 2003 年)、これまで VM 基本機能の開発に注力されてきた。このため、基幹システムへの適用で必要となる RAS 機能やトラブルシューティング機能については、機能が不足している状況である。

(3) 解決方法

Xen の基幹システム適用に向け、基幹システムモデルにて適用評価し、必要な RAS 機能、トラブルシューティング機能を開発する。

1) Xen を基幹システム(アプリケーション/データベースサーバ)に適用し、RAS 機能、トラブルシューティング機能の評価する。特に、I/O やクラスタ関連で弱点が予想されるため、重点

的に評価する。

2) Xen の弱点として、既に判明している以下の機能を開発する。なお、1) の評価で新たに判明した弱点も、今後随時強化していく。

a) ダンプ機能：

Xen がクラッシュした時に、そのメモリ内容をファイルにダンプし、解析する機能。現在の Xen はユーザシステムで障害が発生した場合、再現テストを実施した上で障害調査する必要があり、解決まで 2~3 日かかる。このダンプがあれば再現テストが不要になるので数時間での解決も可能となる。

b) サーバの部品故障時のリカバリ機能：

サーバの部品故障時に予防保守や縮退運用を行う機能。ハイエンドのサーバはハードウェア部品故障をソフトウェアに通知し、予防保守(部品交換)や縮退運用を行う仕組みがある(マシンチェックリカバリ)。現在の Xen はこのリカバリ機能を有していないため、サーバの部品故障でもシステム停止が必要になってしまう。

(4) コミュニティの状況

ケンブリッジ大学 (Xen オリジナル版の開発者) や大手プロセッサ/サーバベンダを中心に、約 200 人の開発者が性能向上や新ハードウェア対応等の機能強化を活発に続けている。RAS やトラブルシューティング機能は、コミュニティで議論され、必要性は認識されているものの、開発者不足で十分な機能が提供されていない状況である。

一方、日本のエンジニアは長年の基幹システムサポートにより RAS やトラブルシューティング機能のノウハウを持っており、これを活かして Xen の機能充実に取り組み始めている。

(5) 効果

1) Xen の適用範囲拡大：

信頼性と保守性の向上により、Xen が企業の基幹システムを含めた幅広いシステムに適用される助けとなる。

2) 基幹ソフトウェア開発者の技術力向上：

VM は OS と並び基幹ソフトウェアの中核となる技術である。この最先端である Xen のコミュニティで活動することにより、世界で通用するエンジニアの育成を見込める。

3.2.2 メッセージマニュアル データベースの作成

(1) 必要性

Linux では、メインフレームが提供しているようなメッセージマニュアルが存在しない。このため、ユーザにとってはシステム障害時の対処法が分からず、サポートサービスを提供する企業にとってはメッセージ調査のための負荷が軽減されないという悪循環をもたらしている。メッセージマニュアルを作成し公開することで、ユーザおよびサポート提供企業は以下のようなメリッ

トを享受することができる。

1) 障害発生時の迅速な対応：

障害発生時の最も一般的な初期障害切り分けは出力メッセージによる状況判断であり、障害によっては、ユーザ自身の対応による解決を期待できる。サポート提供企業もソース解析が不要になるケースが増え、迅速な対応とともにサポートコストを低く抑えることが期待できる。

2) Linux エンジニア不足への対応：

ドキュメントの充実により、運用管理者等の負担軽減を期待できる。

メッセージマニュアルは、“あれば利用したい”という形で多くの企業・ユーザが共通に欲しているドキュメントであるが、その作成には Linux ソースコードの解析、意味の把握、対処方法の調査といった膨大な作業が必要となり、開発コミュニティにとっても企業にとってもインセンティブが働かない分野である。

(2) 現状の問題点

Linux システムで障害が発生しメッセージが表示された場合、対応が必要なのか、不要なのか（重要度）がすぐに判断できず、仮に対応が必要なことが分かって、どのように対応すればよいか分からない。また、身近にノウハウを持った人間が少ないため、小さな問題でも対応に時間がかかってしまうことが多い。Web を検索することで、ある程度までの障害情報を入手することはできるが、一箇所にまとまった情報が存在しないため即座の対応がとれない。その結果、Linux の導入促進に弾みをつけられない一要因を作り出していると考えられる。

(3) 解決方法

Linux で出力されるメッセージの意味、対処方法をソースコードから調査し、マニュアルとして整備する。紙ベースでのマニュアルでは情報の陳腐化に対応しきれないため、メッセージ情報をデータベースシステム（以下、OSS Message Pedia）として開発し公開する。

OSS Message Pedia は、メッセージの意味、対処法等の情報を Web ブラウザから簡単に参照/更新でき、メッセージに対応するカーネル・ソースコードの参照も可能な Linux メッセージ検索システムである。開発成果物である OSS Message Pedia の概要を下記に示す。

- 対象カーネル/バージョンに対応したメッセージに関する調査項目を登録したデータベース
- 英文メッセージ、日本語訳（意味）、対処法、メッセージ応答後のシステム動作、メッセージ監視に有効となる情報（優先度等）等を提供

(4) コミュニティの状況

LDP (Linux Documentation Project)、JF (Japanese FAQ) Project、JM Project 等の翻訳プロジェクトが多数存在するが、メッセージマニュアルを作成しているプロジェクトは存在しない。OSS Message Pedia の普及/メンテナンスについては、バザールモデルによる展開を視野に入れ、

サポート提供企業、ディストリビュータ、Linux 開発者および一般ユーザから構成される OSS Message Pedia コミュニティの立ち上げを検討する必要がある。OSS Message Pedia を充実させていくにあたっては、ドキュメント化プロジェクトとの協調よりもむしろ Linux 開発コミュニティとの協調により、メッセージ内容自体の充実を働きかけることが重要である。

(5) 効果

OSS Message Pedia の作成および公開は、障害発生時の迅速な対応、Linux エンジニア不足問題の軽減といったことを第一の狙いとしている。副次効果として、Linux の開発コミュニティがメッセージの重要性を再認識し、その結果として Linux 自体の信頼性が向上することを期待している。メッセージマニュアル データベースを公開することで、ユーザからの Linux への信頼感が向上し、Linux の普及促進に繋がる可能性も存在する。

OSS Message Pedia がもたらす効果の大きさについては、OSS Message Pedia 自体の普及度が重要な判断指標であり、その参照/更新頻度をチェックすることで OSS Message Pedia の有効性を判断することが可能である。

3.2.3 Linux カーネル互換性(リグレッション)テストツール

(1) 必要性

Linux カーネルの互換性の保持メカニズムというものは存在しない。そのため、意図しない互換性の破壊（リグレッション）は、自動的には発見されない。通常は、コミュニティでの緩い利用を経たのちにパッチがとりこまれるので、リグレッションはパッチのリリース後にボランティアの利用によって非互換が発見されるというモデルである。意図する非互換についても、明確に文書化されているとは言いがたい。「XXX という機能を追加」という記述はあっても、それによって「YYY という非互換が発生」という記述はほとんどない。ソフトウェア開発による意図しない互換性の破壊（リグレッション）を自動的に検出するためには、リグレッションテストが必要となる。

(2) 現状の問題

現在、リグレッションテストがないために、非互換はボランティアの利用による目視、リリース後相当期間たってからの発見などによる。そのために意図しない非互換に関しては、非互換部分を元に戻すというのは不可能な場合が多い。意図する非互換（機能拡張等）についても広く通知されているとは言い難く、影響が大きい変更でも見過ごされる場合があり、後にアプリケーションの変更を余儀なくされる事がある。

開発の観点から言うと、網羅的なリグレッションテストフレームワークはコミュニティにまかせていても Just For Fun ではないため、開発は加速しない。またリグレッションテストのフレームワークの作成などは、経済的なインセンティブも低く、技術的なチャレンジも少ないので、企業からの積極的な貢献も期待しにくい。

企業にまかせていても、またコミュニティにまかせていても開発が加速しないため、公共財と

しての Linux の価値を高めるためにも取り組む必要がある。

(3) 解決方法

Linux カーネルのリグレッションテストを作成し、新規のカーネルのリリース毎に実行する。リグレッションテストというのは、次のようなメカニズムである。

- 1) テストプログラム自動実行の結果（出力）
- 2) 自動実行の仕組みと差分（期待される出力と実際の出力）の検出の仕組み
- 3) テストプログラムと期待される結果の管理（登録、変更、削除など）

テストプログラム自身と期待される結果はプログラマが作成するため、上記の「フレームワーク作成」と、個別の「テストプログラムの作成」という二つのフェーズがある。

(4) コミュニティの状況

下記サイトのようなテストに関連するコミュニティ活動が行われている。しかし、本強化策のようなリグレッションテストに関する活動は活発ではない。

- Linux Test Project
<http://ltp.sourceforge.net/>
- Software Testing Automation Framework (STAF)
<http://staf.sourceforge.net/index.php>
- Python Regression Tester
<http://pyret.sourceforge.net/>
- Auto Test (Kernel Tests)
<http://test.kernel.org/autotest/AutoTest>
- Open POSIX Test Suite
<http://posixtest.sourceforge.net/>
- Linux Standard Base (LSB)
<http://www.linuxbase.org/>

2006年7月にOttawaで開催されたLinux Symposiumでもいくつかのテストプロジェクトが紹介された。

<http://www.linuxsymposium.org/2006/speakers.php?types=TALK>

(5) 効果

リグレッションテストを実行する事により、意図しない非互換は自動的に発見される。そのため当該非互換を発生させたパッチに関しては容易に特定でき、かつ当該パッチを迅速にリジェクトできるようになる。意図する非互換についても、第三者がリグレッションテストにより発見できるため、アプリケーション開発者間における早い段階での情報共有が可能となる。

3.2.4 クラスタファイルシステムの評価

(1) 必要性

近年、高可用性が求められるフィールドでクラスタシステムの導入が増加傾向にある。中でも、Linux を使ったクラスタシステムは、様々なクラスタリングソフトウェアが提供されており、安価にシステムが構築できる点で特に注目されている。

複数のマシンをクラスタシステムとして稼働させる際、ファイルシステムの共有が必要となるケースがあるが、NFS(Network File System)では、機能、性能、スケーラビリティなどで期待した結果が得られないことが多く、近年クラスタファイルシステム（以下、CFS）が用いられている。今までは、クラスタシステムで使用する共有ディスク装置自体が高額であったため、商用のCFSを用いても初期費用には大きな影響がなかった。しかし、近年、iSCSIなどを使用した安価な共有ディスク装置が提供されるようになり、初期費用におけるCFSの価格の占める割合が大きくなりつつある。今後さらに共有ディスク装置の低価格化が予想されるため、初期投資額を抑えるためにオープンソースで開発されているCFSの導入がますます見込まれる状況である。

(2) 現状の問題点

Linux 分野においても、オープンソースで開発されているクラスタファイルシステム（以下、OSS-CFS）が複数存在する。しかし、機能、性能、スケーラビリティにおいて評価基準が存在しないため、これらが利用者の期待に対して有効なソリューションとなり得るのかどうかという点で不明な部分が多い。

共有ディスク装置の低価格化が進むことで、MC 分野以外でも共有ディスクを使ったクラスタシステムの導入が予想される。しかし、現状では OSS-CFS の期待値（ポテンシャル）が客観的に把握できておらず、商用 CFS に対する費用対効果の優位性を客観的に示すことができていない。このような状況では、今後 OSS-CFS が積極的に導入されるとは言い難い状況である。

(3) 解決方法

まず、OSS-CFS の基礎情報を把握するために、どの程度の負荷やマシン構成に耐えうるかを、以下の手順で調査する。

1) 評価基準設定

CFS を比較する際に必要な評価項目を、以下の観点で調査する。

- 機能
- 性能
- 信頼性
- スケーラビリティ

2) 評価

設定した評価項目にしたがって、OSS-CFS の評価を行う。この評価結果を元に各 OSS-CFS の性能特性を把握する。

3) 考察

各 OSS-CFS の性能特性から、どの程度の規模や用途に有効なソリューションであるかを、以下の観点でまとめる。

- ノード数
- ファイル数
- ファイルサイズ

(4) コミュニティの状況

CFS 独自のテストツールが存在するケースがあるが、動作確認用のツールであるため CFS 共通の評価用ツールとしては使えない。CFS それぞれで必要なテストツールは実装されており、各自での評価はされているが、CFS 共通での評価は現時点では行われていない。

(5) 効果

今後、ますますクラスタシステムの導入が見込まれるが、現状では OSS-CFS に関する情報不足により、Linux によるクラスタシステムの導入を見送るケースが多くでてくることが予想される。今回の提案で得られる OSS-CFS の情報で CFS の選択が適切にできれば、安価にシステムを構築できるという Linux の利点をそのままクラスタシステム構築にも適用できるので、クラスタ分野においても Linux および各種 OSS の普及が促進されると考えられる。

3.2.5 Linux ディスク冗長化機能の強化

(1) 必要性

ディスク冗長化機能として一般的には RAID (Redundant Array of Independent Disks) が知られている。これは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に継続的にアクセスすることを可能とする技術である。実現方法として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 と、OS あるいはドライバが RAID 機能を実現するソフトウェア RAID がある。ハードウェア RAID は RAID コントローラだけ価格は高くなるが、ソフトウェア RAID に比べてより高い信頼性を得ることができる。ソフトウェア RAID は、Linux カーネルに含まれて提供される md (Linux ソフトウェア RAID ドライバ) や dm (Device Mapper) と、ベンダがデバイスドライバの形で製品提供する商用のソフトウェア RAID ドライバに分類される。

(2) 現状の問題点

従来、高信頼性が求められるシステムではハードウェア RAID を用いることが一般的であった。しかし近年ではハードウェア追加無しで実現可能で、より安価なソフトウェア RAID を利用する

場合も増えてきており、基幹系システムに導入するケースでもソフトウェア RAID を使用するケースも出てきている。今までこのようなケースでは品質的により安定している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が、ベンダ製の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り、今後は md や dm を用いる事例が増加することが予想される。

また、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。しかし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り、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。

(3) 解決方法

このためには、まず Linux 上のソフトウェア RAID 機能について網羅的な評価を行うための評価プログラムを作成し、次にそれを用いてソフトウェア RAID 機能の品質状況を明確にする必要がある。評価対象は現状利用者の多い SCSI ディスク及び SATA ディスクをターゲットとする。検出された問題点については必要に応じてコミュニティへの修正提案を行い、ソフトウェア RAID 機能の品質強化を図る。この際、評価実施後に修正が円滑に取り込まれるよう、メーリングリスト等を使ってコミュニティとの連係を図るべきである。

(4) コミュニティの状況

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。しかしディスク故障時の処理やエラー処理は実行される頻度も低いため、これだけでは十分な品質を確保することは困難である。またエラー処理系も含めた網羅的な評価はかなりの作業を要するため、現状コミュニティではこのような評価は行われていない。このため故障からの修復動作中のエラー処理、多重エラー発生などの場合を含む網羅的な評価を通じた品質強化が重要であると考えられる。

(5) 効果

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安価で高い品質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。この強化策により Linux のソフトウェア RAID 機能の品質を明らかにし、今後の問題点の解決につなげることは、OSS 活用の拡大にとって大きな意味があると考えられる。

3.2.6 統合運用管理ツール Hinemos の強化

(1) 必要性

Linux を始めとした OSS の普及を促進するうえで、従来はプラットフォームやミドルウェアの

強化に重点が置かれてきた。近年は、情報システムにかかわる作業を適正に実施するための運用プロセスの効率化が注目されているが、Linux システムを管理する OSS の IT ツールは目的ごとに個別に提供される場合が多い。そこで、運用プロセスの効率化のニーズに応える統合運用管理ツールを強化する必要がある。

(2) 現状の問題点

OSS の統合運用管理ツールとして、Hinemos*が存在する。Hinemos は、各種 OSS を活用し、監視管理機能・性能管理機能・ジョブ管理機能などを統合運用環境として提供している。しかし、開発から日も浅く（初版公開が 2005 年）運用管理の基本機能が整いつつあるものの、追加機能の開発、対応プラットフォームの拡充、監視ノウハウ情報の蓄積、基幹システムモデルにおける適用評価、において基幹システムに求められる要求を満たさきれていない。

* <http://sourceforge.jp/projects/hinemos/>

(3) 解決方法

現状の問題点として挙げた項目への解決方法は以下のとおりである。

a) 追加機能の開発

運用管理で必要とされる機能は広範にわたるが、以下は優先的に進めることが望ましい項目である。

- ・ Linux の各種ネットワークサービスの監視
- ・ 各種設定を複数のサーバに対して一括して行う機能と、設定情報を収集・遠隔保守する機能
- ・ サーバの構成情報（ハードウェア、ソフトウェア等）の管理機能
- ・ OS 停止時のメンテナンス（ファームウェア経由のリモートメンテナンス）
- ・ SNMP-TRAP 情報の監視
- ・ Hinemos のマネージャ（アプリケーションサーバ）機能の HA(High Availability)クラスタ対応

b) 対応プラットフォームの拡充

情報システムは、業務アプリケーションの更改やハードウェアの保守期間などの諸事情により、1 つの情報システム内に複数の種類の環境（プラットフォーム）が混在していることが多い。Hinemos 自体は Java で開発されているため移植性は高いが、メッセージログの取得などプラットフォームに依存する部分もある。この点、運用管理対象となるプラットフォームはバージョンの相違などを加味すると Linux だけでも多岐にわたるため、試験を自動化するツールを用意するなどして動作確認の工数を極力抑えることが必要である。

c) 監視ノウハウ情報の蓄積

運用管理に必要とされるノウハウは様々であるが、特にログメッセージ監視について、監視対象のソフトウェアがどのような挙動のときにどのようなログを出力するかを把握する必要がある。この点、OSS においてはメインフレームで提供されているようなメッセージマニュアルが十分に

は整っていない。そのため、本番稼働前に異常動作させて出力されるログメッセージや、稼働開始後に出力されるログメッセージを取得して、監視設定を追加する必要がある。しかし、このようなシステム開発段階での調査は開発の負荷を増大させる。また、稼働開始後の設定追加は障害発生たびに追加していくかたちをとるため、ビジネス上の損失やリスク管理の面で問題がある。

そこで、メッセージについては、「3.2.2 メッセージマニュアル データベースの作成」と連携し、Linuxのみならず各種OSSのメッセージ情報のデータベース化が必要である。データベース化されたメッセージ情報があれば、Hinemosの監視管理機能（例えばsyslog-ng監視）のログ解析のテンプレートとして取り込むことが可能となる。また、メッセージ情報と連動する仕組みを提供できれば、システム障害発生時の出力メッセージをもとにした検索機能により調査や対応を速やかに行うことが可能となる。さらに、障害情報をメッセージマニュアル データベースにフィードバックすることにより、メッセージマニュアル データベース側もさらに精度を高めることができる。

d) 基幹システムモデルにおける適用評価

基幹システムに求められる要件は様々であるが、Hinemosの評価情報が公開されていると導入検討が容易となる。そこで、導入検討のための基礎資料を提供できることが望ましい。測定の例として、(a)大量ノード、(b)大量ログメッセージ、(c)大量ジョブ、を挙げることができる。

(4) コミュニティの状況

運用管理分野のユーザや開発者は一部の企業の部門などに限られているため、Linuxのようなバザール型での急速なコミュニティ拡大を期待しにくく、上記の強化策の項目を進めるには時間を要する状況である。

(5) 効果

Hinemosの機能追加や対応プラットフォームの拡充により、OSSのITツールの連携が容易化・強化され、自動化・効率化できる範囲が広がる。また、適用評価情報があれば、ユーザが容易かつ適切に導入判断を行うことが可能となる。さらに、ユーザが監視ノウハウ情報を用いて各種OSSのトラブルに迅速な対応ができれば、OSS全体の普及促進に繋がる可能性も存在する。加えて、こういったユーザの広がりや各種OSSとの連携が、Hinemos自体の開発コミュニティの形成・拡大に繋がることも期待したい。

4. まとめ

近年、MC システムにおける OSS 適用拡大の流れがめざましい。この流れの中で、ベンダやユーザがそれぞれ OSS の開発を進め、OSS の推進・発展に寄与している。しかし、ベンダやユーザが個々に活動を進めているため、商用ソフトウェア開発では当たり前である開発ロードマップに相当するものが OSS には存在しない。このことにより OSS で構築された IT システムの長期的な計画が立てづらいという問題が依然として残っている。

そこで今回ロードマップ TF として、日本のベンダおよびユーザからの要望をふまえながら独自に OSS 開発ロードマップのシナリオを描いてみた。あくまで、この開発ロードマップは一つの提案ではあるが、今後、日本のみならずアジアひいてはグローバルで OSS に関わる人たちの間でのコミュニケーションの活発化と MC 領域における OSS のシェア拡大につながれば幸いである。

付録

付録 A DCL-WG 「DCL Goals and Capabilities」

表 A.1 技術達成度 (Maturity) の見方

Maturity Level	Description	Range of Completion
Investigation	The project is in the concept phase	0-9%
Development	The project is started	10-39%
Released	Early releases are available	40-49%
Usable	Working, usable release are available	50-69%
Stable	The project has released a stable version	70-79%
Integrated	The code is in the development kernel tree, and/or distributions are included as patches	80-89%
Mainline	The code is in a stable kernel tree, but it might not be available complete	90-94%
Product available	A customer-available version meets specification	95-99%
Completed	A customer-available version meets specification in more than one distribution. Capabilities can be completed, yet further integration or regression testing may still be needed	100%
N/A	Not applicable	

表 A.2 Scalability 状況

Category	Capability	Description	Maturity(Edge/App/DB)	Priority	
Scalability	CPU 16way		Mainline	1	
	Disk I/O Connectivity 4096 Storage Devices		Mainline	1	
	Disk I/O Async I/O - File System		N-A/Stable/Stable	1	
	Memory - 64GB		Mainline	1	
	NUMA APIs		N-A/Product Available/Product Available	1	
	NUMA Topology		Completed	1	
	Symmetric Multi-Threading(Hyperthreading)		Stable	1	
	CPU 1 way		Completed/N-A/N-A	2	
	CPU 2 way		Completed/N-A/N-A	2	
	CPU 4 way		Completed/Completed/N-A	2	
	CPU 8 way		Completed	2	
	CPU 32 way		N-A/N-A/Stable	2	
	CPU 64 way		N-A/N-A/Stable	2	
	Network I/O 10/sec	10 connections/sec		Completed/N-A/N-A	2
	Network I/O 100/sec	100 connections/sec		Completed/N-A/Completed	2
	Network I/O 1000/sec	1000 connections/sec		Mainline/N-A/N-A	2
	Network I/O 10Mb/sec	10Mbs Throughput /sec		Completed/N-A/N-A	2
	Network I/O 100Mb/sec	100Mbs Throughput /sec		Completed	2
	Network I/O 1000Mb/sec	1000Mbs Throughput /sec		Mainline	2
	Network I/O 10Gb/sec	10Gbs Throughput /sec		Completed/N-A/N-A	2
	Network Sendfile	効率的なファイル転送		Completed/N-A/N-A	2
	Network Copyless Send/Receive	User/Kernel空間のデータコピー削減による性能向上		Development	2
	Network Scalable Poll	1000を超えるファイル数のときの"Poll/Select"		Completed	2
	Network Asynchronous I/O	汎用的な非同期I/Oの提供		Development	2
	Network Segment Offloading	Low Layer Segmentation機能を持ったNICサポート		Completed	2
	Network Checksum Offloading	TCP/UDPチェックサム機能を持ったNICサポート		Completed/N-A/N-A	2
	Network Denial of Service Protection	DOS防御機能		Development/N-A/N-A	2
	Network High Speed Routing(IPV6)	高速Routing		Development/N-A/N-A	2
	Network Better Quality of Service and	QoS向上		Integrated/Integrated/N-A	2
	Network APIs for High Speed Interconnect	AIM, RDMA and TCP offload		N-A/N-A/Development	2
	Network Support for High Speed TCP	高速TCP支援		Stable	2
	Disk I/O Connectivity 2 Storage Devices	2つのStorage接続		Completed/N-A/N-A	2
	Disk I/O Connectivity 8 Storage Devices	8つのStorage接続		Completed/N-A/N-A	2
	Disk I/O Connectivity 12 Storage Devices	12つのStorage接続		Completed/N-A/N-A	2
	Disk I/O Connectivity 256 Storage Devices	256つのStorage接続		Completed/N-A/N-A	2
	Disk I/O Connectivity 8k Storage Devices	8192のStorage接続		Stable	2
	Disk I/O Maximum File Size 160GB	160GB Storage Size		Completed/N-A/N-A	2
	Disk I/O Maximum File Size 1TB	1TB Storage Size		Completed	2
	Disk I/O Maximum File Size 16TB	16TB Storage Size		Mainline	2
	Disk I/O Maximum File Size 32TB	32TB Storage Size		N-A/Investigation/Investigation	2
	Disk I/O 625/Sec	625 I/O Storage Device Operation/Sec		Completed/Integrated/Integrated	2
	Disk I/O 5000/Sec	5000 I/O Storage Device Operation/Sec		Completed/Integrated/Integrated	2
	Disk I/O 80000/Sec	80000 I/O Storage Device Operation/Sec		Completed/Integrated/Integrated	2
	Disk I/O 160000/Sec	160000 I/O Storage Device Operation/Sec		Completed/Integrated/Integrated	2
	Disk I/O Throughput 40MB/Sec	I/O Bandwidth		Completed/N-A/N-A	2
	Disk I/O Throughput 300MB/Sec	I/O Bandwidth		Completed	2
	Disk I/O Throughput 5GB/Sec	I/O Bandwidth		Integrated	2
	Disk I/O Scalable Disk Locking	Scalable Kernel Lock		N-A/Mainline/Integrated	2
	Disk I/O ReadAhead	先読み		Completed/N-A/N-A	2
	Disk I/O Async I/O(Raw)	Raw Mode 非同期I/O		N-A/Completed/Completed	2
	Disk I/O Direct I/O-Raw	Raw Mode直接I/O		Completed/N-A/N-A	2
	Memory - 1GB			Completed/N-A/N-A	2
	Memory - 4GB			Completed/N-A/N-A	2
	Memory - 8G			Completed/Completed/N-A	2
	Memory - 16GB			Completed	2
	Memory - 256GB			Released	2
	Memory - 1TB			Released	2
	Layered Software ERP	ERP Application		N-A/Integrated/N-A	2
	Layered Software SCP	SCP Application		N-A/Integrated/N-A	2
	Layered Software CRM	CRM Application		N-A/Integrated/N-A	2
	Layered Software MRO	MRO Application		N-A/Integrated/N-A	2
	Layered Software SFA	SFA Application		N-A/Integrated/N-A	2
	Layered Software Java	Java Application		N-A/integrated/integrated	2
	Layered Software ORB	ORB Application		N-A/Integrated/Integrated	2
	Layered Software RDBMS	RDBMS Application		N-A/Integrated/Integrated	2
	I/O Interface PCI			Completed	2
	I/O Interface PCI-X			N-A/Completed/Completed	2
	I/O Interface InfiniBand			N-A/Completed/Completed	2
	Kernel Huge Number of Threads			Product Available	2
	Kernel Memory	効率的なメモリ割り当て		Completed/Product Available/Mainline	2
	Kernel CPU	効率的なスケジューリング		Completed/Product Available/Product Available	2
	Kernel I/O Interface	Block Layer I/O再構成		Completed	2
	Kernel Node	Single Nodeで動かないJob対応		Completed/Product Available/Mainline	2

表 A.3 Performance 状況

Category	Capability	Description	Maturity(Edge/App/DB)	Priority
Performance	NFS v2/v3 Performance Server/Client	NFS V2, V3 http://ietf.org/rfc/rfc.1094.txt?number=1094/	Usable/Released/Released	1
	NFS v4(Performance and Functionality)	NFS V4 http://www.linux-nfs.org	Integrated	1
	Java Performance		N-A/Usable/Integrated	1
	Port Quality		N-A/Usable/Mainline	1
	Application Performance		N-A/Usable/N-A	1
	File System Performance	Competitive performance IO Zone http://www.iozone.org/	Stable/Usable/Usable	1
	Packet Test	Packet送信性能テスト	Completed/N-A/N-A	2
	Forwarding/Forwarding Test	Forwardingテスト	Integrated/N-A/N-A	2
	Load Balancing	Load Balancing	Completed/N-A/N-A	2
	Security Keys/Second	SSL Performance	Mainline/N-A/N-A	2
	File Server	File Server性能	Completed/N-A/N-A	2
	Web Server	Web Server性能	Completed/N-A/N-A	2
	Mail Server	Mail Server性能	Completed/N-A/N-A	2
	Directory Services	DS性能	Integrated/N-A/N-A	2
	GCC Optimization	GCC性能	Integrated	2
	Large Multi Task Performance	多数のタスク実行時の性能	N-A/Integrated/Integrated	2
	Performance Feature & API	ISVが性能測定に必要とするAPI	N-A/Stable/Stable	2
	Middleware Performance - ODBC	ODBC性能	N-A/Usable/Usable	2
	Workload OLTP 4CPU less	OLTP性能	N-A/N-A/Integrated	2
	Workload OLTP 4CPU Greater	OLTP性能	N-A/N-A/Usable	2
	Workload DSS 4CPU less	Decision Support System	N-A/N-A/Integrated	2
	Workload DSS 4CPU Greater	Decision Support System	N-A/N-A/Usable	2
	Workload Ecommerce	Ecommerce性能	Product Available/Integrated/Product Available	2
	Workload Financial	Trades性能	Mainline/Integrated/Mainline	2
	Tunable Parameters	性能に影響するパラメータ調整	Stable	2
	Current Technology Implementation	新テクノロジー(Hyperthread等)サポート	Integrated	2
	Performance Measurement Infrastructure	性能測定のためのHook	Integrated	2
	Component Notification Mem/IO/Power Failure, Temperature	Allow plugging/unplugging the device	Integrated	1

表 A.4 RAS 状況

Category	Capability	Description	Maturity(Edge/App/DB)	Priority
RAS	Crash Dump		Stable	1
	Update Notification-Data Corruption	問題、障害発生、セキュリティupdateが必要であるとの知らせ	Development	1
	Debugger--Kernel		Mainline	1
	Dynamic Tracer	LTT, LKST	Released	1
	H/w Fault Prediction & Fault Location	SW should detect HW which is close to failure	Integrated	1
	S/W Fault Location Identification	OS to isolate SW failure	Investigation	1
	Live Snapshot-Kernel Level		N-A/Released/Released	1
	Performance Monitoring		Stable	1
	Hotswap(I/O bus Level PCI,PCI-X,cPCI)	Allow plugging/unplugging the device	N-A/Completed/Completed	1
	Hotswap(I/O bus Level SCSI)	Allow plugging/unplugging the device	N-A/Mainline/Mainline	1
	Hotswap(I/O bus Level PCI Express)	Allow plugging/unplugging the device	N-A/Completed/Completed	1
	Hotswap(Component Level Memory)	Allow plugging/unplugging the device	N-A/Released/Released	1
	Hotswap(Component Level CPU)	Allow plugging/unplugging the device	N-A/Product Available/Product Available	1
	Hotswap(Component Level Node)	Allow plugging/unplugging the device	N-A/Released/Released	1
	Component Notification Mem/IO/Power Failure, Temperature	Allow plugging/unplugging the device	Integrated	1
	Fast System Boot		Product Available	1
	Reliable File System Writes	Guarantee write system calls	Completed	1
	Multipath I/O	Access device over multipath	N-A/N-A/Released	1
	Shared Memory&IPC Parameter Changes without reboot		N-A/Investigation/Investigation	1
	Package Change History/Logging	RPMの履歴	Product Available	2
	Hardware Change History/Logging	HW変更履歴	Investigation	2
	Customization History/Logging	パラメータ変更履歴	Investigation	2
	Install on Alternate Target	予備Diskへのインストール	N-A/Development/Development	2
	Revert Installation/Patch Sets if Failure	失敗時の戻し	N-A/Released/Released	2
	Version and Integrity Checking	新Packageと旧Packageの依存関係チェック等	Usable	2
	Update Notification-Live Snapshot-Process Level	End Userへのセキュリティupdateの知らせ Process LevelでのSnap Shot	Product Available N-A/Investigation/Investigation	2
	Proactive System Health Monitoring	潜在エラーの検出(出来れば修正)	N-A/Investigation/Investigation	2
	Remote Serviceability	LAN外からのリモートサービス	Completed	2
	Reproducible Server Replicant - Procedural	グループ外・内でのサーバ入れ換え手続き	Completed	2
	Reproducible Server Replicant - Automated	上記の自動化	Development	2
	System Checkpoint/Server Replacement	システム全体の状態保存と復活	N-A/Investigation/Investigation	2
	Hotswap(I/O bus Level USB)	USBのホットスワップ	N-A/Mainline/Mainline	2
	Hotswap(I/O bus Level iSCSI)	iSCSIサブシステムのホットスワップ	N-A/Development/Development	2
	Hotswap(I/O bus Level infiniband)	infinibandのホットスワップ	N-A/Investigation/Investigation	2
	Hotswap(I/O bus Level IDE)	IDEのホットスワップ	N-A/Development/Development	2
	Hotswap(I/O bus Level ATA)	ATAのホットスワップ	N-A/Investigation/Investigation	2
	Hotswap(I/O bus Level Firewire)	Firewireのホットスワップ	N-A/Investigation/Investigation	2
	Hotswap(Component Level I/O)	Disk DriveとCardのスワップ	Usable	2

Fast Install	高速インストール	Product Available/Mainline/Mainline	2
Journaling File System	JFS	Product Available	2
Atomic File System Operation	システムクラッシュにおいてFSの整合性を保証するためのAtomicなOperation	Released	2
Failover - Network Service	Virtual Server and Load balance	Product Available	2
Failover - Statefull	AP Level Switch Over	N-A/Investigation/Investigation	2
Failover - Open RDBMS Dependent	OSS DBのFail over機能	N-A/Stable/N-A	2
Failover - Proprietary RDBMS Dependent	商用DBのFail over機能	N-A/Mainline/Mainline	2
Backup Solution GB Range	1GB DB LevelのBackup	N-A/Completed/Completed	2
Backup Solution TB Range	1TB DB LevelのBackup	N-A/N-A/Integrated	2
Volume Manager	Software RAID and disk replacement	Mainline	2
Replacing Modules without reboot	リポートせずに、モジュールを入れ換える	N-A/Usable/Usable	2
Graceful Handling of Memory Exhaustion	OOM時の適切な扱い	Usable	2

表 A.5 Manageability 状況

Category	Capability	Description	Maturity(Edge/App/DB)	Priority
Manageability	Common Interface for 3rd Party Integration to install tools	provide a common installation packages format for ISV	Development	1
	Software Packages Management	Cross Linux Platform Package Management	Mainline	1
	Software Packages Management-1	Localにいながら、Remote Installが可能	Mainline	1
	Software Packages Management-2	Revertが行える	Usable	1
	Software Packages Management-3	複数リモートシステムへのインストール	Usable	1
	Software Packages Management-4	内容、Versionの確認	Stable	1
	Configuration Management(Expanded to full stack)	環境の変更等の自動記録、管理	Usable	1
	Volume Manager	http://linux-hotplug.sourceforge.net	Completed/Integrated/Integrated	1
	Persistent Storage Device Naming	Device recognition and persistent device naming http://www.kernel.org/pub/linux/kernel/hotplug/udev.html	Mainline	1
	Log Monitoring/Event Notification/Agents	OS notifies the administrators via any of notification mechanism including e-mail, real time paging	Mainline	1
	Workload Management		Mainline	1
	Enhanced Process and Resource Monitoring		Stable	1
	Local Software Stack Install and Update	リモートからOSのUpdate内容を持ってきてUpdateする	Mainline	2
	Remote Multi-System Software Stack Install Update or Replication	リモートでOSを含む全てのソフトウェアのInstall/Update/Replicationを行う	Integrated	2
	Device Configuration Discovery	Boot時・動的なDevice検出	Product Available	2
	Remote Console Access	リモートコンソール	Mainline	2
	Job Management	Job管理	Stable	2
	Problem Management	問題検出ツール	Investigation/Investigation/N-A	2
	Remote Management	リモート管理	Stable/Stable/Integrated	2
	Network Management	ネットワーク管理	Mainline	2
	User Management	ユーザ管理	Stable	2
	Resource management-Asset Management	シリアル番号等を含むシステム全体の資産リソース管理	Development	2
	Resource management-Usage Tracking	システム全体の資産リソースの使用状況管理	Usable	2
	Resource management-Load Balancing and Tracking	リソースの実際の使用状況管理	Integrated/Integrated/N-A	2
	Process and Resource Monitoring	GTOP/SARを用いた状況管理	Mainline	2

表 A.6 Clusters 状況

Category	Capability	Description	Maturity(Edge/App/DB)	Priority
Clusters	Cluster-wide Persistent Device Naming	Device recognition and persistent device naming in cluster	Investigation	1
	Cluster Volume Management	Extends Volume Manager capability to the cluster	Development	1
	Cluster File System	Provide consistent file system image and service	Usable	1
	Membership		N-A/Mainline/Mainline	1
	Load Balancing--Resource Based	リソース量ベースのノード間バランス制御	Usable/Usable/N-A	1
	Single System Image--File System View	ファイルシステムの名前空間統一	Usable	1
	Administrative-User Management	User管理関係コマンド群	Usable	2
	Administrative-Software Deployment	Cluster内へのSoftware Deployment管理	Integrated	2
	Administrative-Software Upgrade	Software Upgrade管理	Integrated	2
	Administrative-Central Cluster	log管理、monitoringの中央管理	Usable	2
	Administrative-Cluster Commands	Cluster管理コマンド	Usable	2
	Group Messaging	Cluster内で、ApplicationがMessage交換出来る	N-A/Development/Development	2
	Event Notification	関係者へのevent通知機能	N-A/Integrated/Integrated	2
	Checkpoint	Checkpointでの再開、停止	N-A/Integrated/Integrated	2
	Kernel Distributed Lock Manager	共有リソース制御用ロック提供	N-A/Usable/Usable	2
	Communication	Application間での高信頼のCommunication手段	N-A/Usable/Usable	2
	Load Balancing--Connection Based	Connection状況に従ってノード間のバランス制御	Integrated/Integrated/N-A	2
	Load Balancing--Wide Area Network	WANにおけるロードバランス制御	Usable/N-A/N-A	2
	Load Balancing--Dynamic Balancing	Process Migrationを用いたバランス制御	N-A/Usable/N-A	2
	HA-Failover: Transaction Based	Transaction BaseなFailover	Integrated/Integrated/Mainline	2
	HA-Failover: Continuous	Applicationが継続可能なFailover	Usable	2
	Single System Image-Process	Cluster内でのProcess名前空間の統一	Stable	2
	Single System Image-I/O	I/O名前空間の統一	Usable	2
Single System Image-User/Group	User名前空間の統一	Usable	2	

表 A.7 Standards 状況

Category	Capability	Description	Maturity(Edge/App/DB)	Priority
Standards	LSB Compliance		Mainline	1
	CIM	Common Information Model標準 (DMTF制定)	Usable	2
	SNMP	IETF制定SNMP標準	Completed	2
	IP	Internet Protocol	Mainline	2
	IP-SEC	Internet Protocol Security	Completed	2
	IPMI	Intelligent Platform Management Interface	Mainline	2
	SAF AFS	Service Availability Forum制定Application Interface Specification	Usable	2
	SAF HPI	Hardware Platform Interface	Integrated	2
	ACPI	Advanced Configuration and Power Interface	Stable	2
	Open Printing	FSG制定Open Printing	Released	2
	I/O Interface - PCI	Peripheral Component Interconnect	Completed	2
	I/O Interface - PCI-X 1.0	PCI-X 1.0	Completed	2
	I/O Interface - InfiniBand	InfiniBand	Usable	2

表 A.8 Usability 状況

Category	Capability	Description	Maturity(Edge/App/DB)	Priority
Usability	3rd Party Software Integration		Usable	1
	Common Command Line Administration across distributions	Distributions間で共通の管理者用コマンドライン	Completed	2
	Migration Tools	他UNIXからのmigration	Usable	2

表 A.9 Virtualization 状況

Category	Capability	Description	Maturity	Priority
Virtualization	Run Application unmodified	Run Application unmodified on Virtualized Machine	Completed	1
	Application separation	異なるVirtualized Machine上のAppがseparateしていること	Completed	1
	Full Virtualization	OS変更無くVirtual化出来る	Product Available	2
	Full Virtualization+Performance	性能要件を満たすこと	Product Available	2
	Para Virtualization	OS変更有りてVirtual化出来る	Product Available	2
	32bit Linux on 32bit HW		Completed	2
	32 bit Guest on 64 bit HT		Product Available	2
	64 bit Guest on 64 bit HW		Completed	2
	64 bit Guest on 32 bit Host OS on 64 bit HW		Product Available	2
	Windows 32 bit Guest on 32 bit HW		Product Available	2
	Windows 32 bit Guest on 64 bit HW		Product Available	2
	Windows 64 bit Guest on 64 bit HW		Product Available	2
	Other Guest		Completed	2
Hardware Support	X86-64		Completed	2
	IA64		Usable	2
	IA32		Completed	2
	Power PC 64		Development	2
	Power PC 32		Investigation	2
	SMP Under 16 CPU		Completed	2
	SMP 32 CPU		Product Available	2
	SMP over 64 CPU		Investigation	2
	Multi-core SMP under 8 CPU		Completed	2
	Multi-core SMP 16-32 CPU		Product Available	2
	Multi-core SMP over 64 CPU		Investigation	2
Virtualized Guest SMP Characteristics	Plug-in Schedulers	I/O.CPUのスケジューラの適切な統合	Integrated	2
	SMP Guest on SMP Host		Completed	2
	Non-SMP Guest on SMP Host		Completed	2
	Non-SMP Guest on Non-SMP Host		Completed	2
	SMP Guest on Non-SMP Host		Completed	2
Virtualized Guest OS Drivers	Shared Drivers:Network Interface	複数の仮想OSがNetwork Interfaceをshare可能	Completed	2
	Shared Drivers:Graphics(AGP) Interface	複数の仮想OSがGraphics Interfaceをshare可能	Integrated	2
	Shared Drivers:Storage	複数の仮想OSがStorage Interfaceをshare可能	Completed	2
	Shared Drivers:Miscellaneous	複数の仮想OSがUSB等のほか Interfaceをshare可能	Usable	2
Pass-through Guest Drivers	Pass Thru:Network Interface	仮想OSがNetwork Interfaceの完全な使用が可能	Investigation	2
	Pass Thru:Graphics		Product Available	2
	Pass Thru:Storage		Development	2
	Pass Thru:Misc		Development	2
Device Specific Hardware Supported	Hardware Pass Thru:Network Interface	仮想OSがShare可能であるとともに、HWがsupportしていれば、独占可能?	Development	2
	Hardware Pass Thru:Graphics Interface		Development	2
	Hardware Pass Thru:Storage		Development	2
Hotplug	Hotplug CPU		Product Available	2
	Hotplug memory		Product Available	2
	Hotplug I/O bus		Product Available	2
Virtual Machine Management	Metering	Bandwidthの測定等	Product Available	2
	QOS	各種I/Oの性能サービスの調整	Product Available	2
	Automatic Resource Balancing	各種資源の自動バランスング	Product Available	2
	Check Point/Restart for Guest		Product Available	2
	VM Migration to another Physical Machine	Live Migration	Completed	2
	Save/Restore of a Guest		Completed	2
	CIM Support		Product Available	2
	Tie-in to Legacy Data Center Management tools	Legacy toolsのサポート	Product Available	2
	Enhanced Serviceability	複数仮想OSのサポートツール(パッチの適用とか)	Development	2
	VM Clustering		Development	2
	Virtual Network	NATと統合されたSecureな仮想Network	Product Available	2
	Remote Console	Secureな仮想コンソール	Completed	2
	Console not required	HW Consoleが無くても動作可能とする	Completed	2

Kernel Improvement	Single Kernel Binary for Paravirtualization	変更されたPara用KernelがStandaloneでも走る	Development	2
	Dynticks	Dynamic clock ticksのサポート(割り込み処理が必要などきだけ仮想OSを社行させるため)	Product Available	2
	Large Page Support in VM		Completed	2
Development	Debugger		Product Available	2
	Conversion Tools	Physicalから仮想への変換ツール	Product Available	2
	Test Suites for Validation and Regression	仮想OSのテスト	Product Available	2
	VM Aware Performance Tools	Profiler等	Product Available	2

付録 B 関連 URL 集

- ACPI
<http://www.acpi.info/>
- Auto Test (Kernel Tests)
<http://test.kernel.org/autotest/AutoTest>
- Hinemos
<http://sourceforge.jp/projects/hinemos/>
- Linux HA
<http://www.linux-ha.org/>
- Linux Kernel Mailing List
<http://lkml.org/>
- Linux Kernel Performance Project
<http://kernel-perf.sourceforge.net/>
- Linux Standard Base (LSB)
<http://www.linuxbase.org/>
- Linux Test Project
<http://ltp.sourceforge.net/>
- Mini Kernel Dump (mkdump)
<http://sourceforge.net/projects/mkdump/>
- OpenFabrics Alliance
<http://www.openib.org/>
- OpenHPI
<http://www.openhpi.org/>
- OpenIPMI
<http://openipmi.sourceforge.net/>
- Open POSIX Test Suite
<http://posixtest.sourceforge.net/>
- Python Regression Tester
<http://pyret.sourceforge.net/>
- SAF Test
<http://www.saf-test.org/>

- Software Testing Automation Framework (STAF)

<http://staf.sourceforge.net/index.php>

- SystemTap

<http://sourceware.org/systemtap/>

- The Linux Kernel Archives

<http://www.kernel.org/>

- XenSource

<http://www.xensource.com/index.html>

- X.Org Foundation

<http://x.org/>

ロードマップ TF メンバー一覧

<メンバ> (五十音順)

飯田 敏樹	ノベル株式会社
泉澤 仁	The Linux Foundation
井出ノ上 淳	日本ユニシス株式会社
宇佐美 茂男	日本ヒューレット・パカード株式会社
江後田基広	株式会社オープンソース総合研究所
大越 章司	レッドハット株式会社
熊崎 裕之	株式会社日立製作所
斉藤 雅美	ノベル株式会社
小園井 康志	株式会社オーブンドリーム
鈴木 友峰	株式会社日立製作所
高橋 秀樹	ユニアデックス株式会社
立川 江介	日本電気株式会社
谷口 剛	ターボリナックス株式会社
中島 幸雄	富士通株式会社
中村 恵夫	ターボリナックス株式会社
西川 治	株式会社 NTT データ
西田 茂雄	ミラクル・リナックス株式会社
橋本 尚	株式会社日立製作所
藤田 祐治	レッドハット株式会社
三浦 広志	株式会社 NTT データ
山崎 宥人	富士通株式会社
吉岡 弘隆	ミラクル・リナックス株式会社
吉田 正敏	富士通株式会社 (主査)

<事務局>

橋本 明彦	独立行政法人情報処理推進機構 (IPA)
-------	----------------------

所属は本資料公開時点のもの

商標について

- ・ Linux は、Linus Torvalds の米国およびその他の国における登録商標または商標です。
- ・ Windows は、米国 Microsoft Corporation. の米国およびその他の国における登録商標です。
- ・ VMware は、米国 VMware, Inc. の米国およびその他の国における登録商標または商標です。
- ・ Java およびすべての Java 関連の商標およびロゴは、米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。
- ・ その他、記載されている会社名、製品名は各社の登録商標または商標です。