博士学位論文

高可用性サーバシステムを構築するための サーバ管理システムに関する研究

2018 年 9 月

東京工芸大学大学院工学研究科

電子情報工学専攻

北村光芳

目次

用語集

[アルファベット順]

・ARP テーブル：

IP アドレスと MAC アドレスの関係を示す情報.

・Base server：　本研究で定義（3 章，4 章）

各サービスグループに所属する仮想サーバを省電力状態において，集約する実サーバ.

・CentOS：

Red Hat Enterprise Linux の商標や商用パッケージなどを取り除いた OS のこと.

・DBSS（Dynamic Backup Server System：動的バックアップサーバシステム）：

本研究で提案（2 章，4 章）

1 台の管理対象サーバ（実サーバ）の管理を行うシステムで，管理対象サーバが故障した場合，仮想サーバによってそのサーバの機能を復旧する.

・DNS（Domain Name System）：

インターネットの重要な基盤技術の 1 つで，ドメイン名と IP アドレスの対応付けやメールの宛先ホストを指示するためのシステム.

・Executer システム：　本研究で提案（5 章）

管理サーバにインストールされる Executer プログラムによって管理プログラムの実行や停止を行うシステム.

・Expect 言語：

対話形式で行う処理を自動化でき，タイムアウト機能やサーバへのアクセスに対する返答メッセージに応じた処理が可能なプログラム言語.

・FTP（File Transfer Protocol）：

ネットワークでファイルの転送を行うための通信プロトコル.

・FTTH（Fiber To The Home）：

光ファイバによる家庭向けデータ通信サービスのこと.

・HTTP（HyperText Transfer Protocol）：

Web ページを記述するための言語で書かれた情報をやりとりする場合に使用される通信手順のこと.

・HUB：

コンピュータやプリンタなどを接続する集線装置.

・IoT（Internet of Things）：

モノ（センサ機器，車，電子機器など）がネットワークを通じてサーバやクラウドサービスに接続され，相互に情報交換をする仕組み.

・KVM（Kernel-based Virtual Machine）：

Linux カーネル 2.6.20 以降に標準実装されている仮想化環境を利用するためのソフトウエア.

・Live migration：

動作中の仮想サーバを停止させることなく別の実サーバに移動して処理を継続させる機能.

・Load class：　本研究で定義（3 章，4 章）

実測した負荷値に応じて分類する値.

・LTE（Long Term Evolution）：

携帯電話でブロードバンド並の高速通信が可能になるデータ通信規格のこと.

・MAC（Media Access Control）アドレス：

LAN カードなどのネットワーク機器のハードウエアに一意に割り当てられる物理アドレスのこと.

・MSBS（Multiple Server Backup System：複合サーババックアップシステム）：

本研究で提案（2 章，4 章）

管理対象となる複数種類の実サーバを 1 台の管理サーバ（実サーバ）で管理を行うシステム.

・MTBF（平均故障間隔）：

故障から次の故障までの平均的な間隔を表す数値.

・NFS（Network File System）：

ネットワークを介してファイルを共有するサービスやソフトウエアと，その通信プロトコルのこと.

・NIC（Network Interface Card）：

複数のコンピュータを相互接続するための LAN カードのこと.

・P2P 方式サーバ管理システム：　本研究で提案（5 章）

特定の管理サーバを必要とせず，管理対象サーバの台数に依存しないサーバ管理システム.

・Peer-to-Peer（P2P）：

ネットワークに接続されたコンピュータ同士が端末装置として通信を行う方式で，自律分散型のネットワークモデルであり，サーバまたはクライアントの立場に固定されない.

・PSS（Power-saving Server System：省電力サーバシステム）：

本研究で提案（3 章，4 章）

クライアントにサービスを提供するサービスグループごとの仮想サーバの負荷を実測する．低負荷と判断されたサービスグループの仮想サーバを特定の実サーバに集約し，不要となる実サーバをサスペンド状態にすることで省電力化を実現するシステム.

・RSG（Real Server Group）：　本研究で定義（3 章，4 章）

仮想サーバ群を起動する実サーバ群.

・SIF（System InterFace：システムインタフェース）方式：　本研究で提案（4 章）

独立して動作可能な制御プログラムを複合動作させる上で，それぞれのプログラムの仲介を行うことで制御プログラムの複雑化を防ぐ方式.

・SMB（Server Message Block）：

ネットワークを通じてコンピュータ間でファイルやプリンタなどを共有するための通信手順のこと.

・SMTP（Simple Mail Transfer Protocol）：

E メールを送信するための通信手順のこと.

・TCP（Transmission Control Protocol）：

インターネットにおいて標準的に利用されており，信頼性の高い通信を実現するためのプロトコル.

・TPUCELS（Tokyo Polytechnic University Computer Education Laboratory System）：

東京工芸大学の情報教育や研究を支えるため，情報処理教育研究センターによって構築された PC 演習室システム.

・TSS（Time Sharing System）：

CPU を時分割して複数の処理を並行するのと同等の効果を上げる方式.

・VSG（Virtual Server Group）：　本研究で定義（3 章，4 章）

クライアントにサービスを提供する仮想サーバ群.

・WOL（Wake on LAN）：

ネットワーク経由でコンピュータやネットワーク機器の電源を投入する仕組み.

［五十音順］

・異種システム混合処理方式：　本研究で提案（3 章）

　独立して動作可能な制御プログラムを交互に動作させ，それらの性能を実現する方式.

・オーバヘッド：

　コンピュータで処理を行う場合，その処理を行うために必要となる間接的な処理や

　手続きなど，余分に費やされる処理時間のこと.

・仮想 IP アドレス：

　複数の機器で共有して利用可能な一時的な IP アドレスのこと.

・仮想化ソフトウエア：

　仮想コンピュータを動作可能とするソフトウエアで，その構成に応じて Host OS 型

　または Hypervisor 型などの種類がある.

・仮想サーバ：

　仮想的なコンピュータのことで, 1 台の実サーバを複数台の仮想サーバに分割して利用

　する．それぞれの仮想サーバでは OS やアプリケーションを実行させることができ，

　あたかも独立したコンピュータのように使用することができる.

・稼働率：

　ある期間におけるシステムの全運転時間に対する稼働時間の割合.

・管理の優先順位：　本研究で提案（5 章）

　1 台の管理対象サーバを 2 台の管理サーバで監視するため，管理対象サーバ故障時に

　管理の優先権を明確化する順位.

・クライアント・サーバシステム：

　コンピュータをサーバとクライアントに分け役割分担をして運用する仕組みのこと.

・計算グリッド：

インターネットなどの広域なネットワーク上にある計算資源（CPU などの計算能力やハードディスクなどの情報格納領域）を結びつけ，ひとつの複合したコンピュータシステムとしてサービスを提供する仕組み.

・高可用性：

誤動作や故障が発生した場合，最小限のサービス中断を生じるが主要なサービスは継続できる性質.

・コールドスタンバイ：

サーバなど，同じ構成の予備マシンを電源 OFF の状態で待機させておく障害対策の手法.

・サスペンド：

コンピュータの一部の機能を停止して省電力モードで待機させること．通常の電源断とは異なり，次回起動が高速になる.

・サーバ・クライアント方式：　本研究で定義（5 章）

特定の管理サーバが複数の管理対象サーバの管理を行う方式.

・サービスグループ：　本研究で定義（3 章，4 章）

同様のサービスを提供する仮想サーバ群.

・死活監視：

コンピュータやシステムが動作しているか，外部から継続的に監視することで，停止しているか否かを判断する監視.

・システムインタフェース方式（SIF：System InterFace）：　本研究で提案（4 章）

独立して動作可能な制御プログラムを複合動作させる上で，それぞれのプログラムの仲介を行うことで制御プログラムの複雑化を防ぐ方式.

・省電力サーバシステム（PSS：Power-saving Server System）：

本研究で提案（3 章，4 章）

クライアントにサービスを提供するサービスグループごとの仮想サーバの負荷を実測する．低負荷と判断されたサービスグループの仮想サーバを特定の実サーバに集約し，不要となる実サーバをサスペンド状態にすることで省電力化を実現するシステム．

・冗長化構成：

設備や装置を複数用意し，一部が故障しても運用を継続可能とする構成．

・性能監視：

サーバの CPU，メモリやディスクなどのリソース使用量を監視することで，障害が発生する状態を迅速に検出することが可能となり，情報システムのサービス停止時間を大幅に短縮できる．

・耐障害性：

障害が発生しても，障害の影響範囲を局所的に抑え，サービスを止めることなく運用を続け，迅速に回復できる性質．

・動的管理拡張方式：　本研究で提案（5 章）

複数の管理対象サーバ故障時に対応可能な方式で，管理サーバは故障検出と同時に新たな管理対象サーバの管理を行う．

・動的バックアップサーバシステム（DBSS：Dynamic Backup Server System）：

本研究で提案（2 章，4 章）

1 台の管理対象サーバ（実サーバ）の管理を行うシステムで，管理対象サーバが故障した場合，仮想サーバによってそのサーバの機能を復旧する．

・トラヒック：

通信回線やネットワーク上で送受信される信号およびデータの量や密度のことを示す．

・ネットワークコーディング：

複数のパケットを混ぜ合わせ，効率良くデータを送信する技術のこと.

・バックアップサーバ：　本研究で定義（3 章，4 章）

故障したサーバの機能を復旧するための仮想サーバ.

・非選択値：　本研究で定義（5 章）

仮想サーバを起動する管理サーバの選択に使用する値で，その値が大きい場合，

仮想サーバの起動には適さないことを示す.

・フェイルオーバクラスタシステム：

サーバを運用系，待機系に分けて二重化して相互に監視を行い，運用系に異常が

発生した場合，速やかに待機系に引き継ぎ，サーバの稼働を継続するシステム.

・複合サーババックアップシステム（MSBS：Multiple Server Backup System）：

本研究で提案（2 章，4 章）

管理対象となる複数種類の実サーバを 1 台の管理サーバ（実サーバ）で管理を行う

システム.

・ブリッジ接続：

異なるネットワークセグメント同士をソフトウエア処理によって接続すること.

・プロトコル：

コンピュータ同士が通信をする場合の手順や規約.

・ホットスタンバイ：

稼働機に対して予備機を，通常時から起動した状態にしておく形態.

・マルチ OS ブートシステム：

複数種類の OS がインストールされたコンピュータで，その起動時に起動すべき OS を

選択することが可能となるシステム.

・モード分割方式：　本研究で提案（各章）

サーバ管理用のソフトウエアシステムをモードとし，複数のモードにより 1 つのシステムを構築する．管理対象サーバの OS に依存するモードのみを変更することにより，複数種類の OS の管理対象サーバの管理が可能となる．

・リソース：

計算資源（CPU，メモリやハードディスクの容量など）の総称.

・ロバスト性：

環境の変化といった外乱の影響によって変化することを阻止する内的な性質.

・ロードバランサ：

サーバにかかる負荷を平等に振り分けるための装置のこと．これによって停止状態を防ぐこともできる．（UltraMonkey-L7：コンピュータ上でロードバランサの動作を可能とするソフトウエア）

・ロードバランサクラスタシステム：

複数のサーバに対して負荷分散をすることでパフォーマンスを向上するとともに，それぞれのサーバの可用性を高めるシステム.

・割り込み処理：

コンピュータが通常の流れでプログラム処理を実行している所へ，割り込んで強制的に別のプログラムを実行させること.

第 1 章

序論

1.1 本研究の背景

サーバシステムは，1960 年代までは集中処理が主流でありオフィスコンピュータやメインフレームがコンピュータ処理の大半を占め，クライアントは最低限の画面制御を行っていた．1970 年代から 1990 年代にかけてコンピュータの性能が飛躍的に向上し，価格は下がる傾向となり，サーバを目的別に分割して設置する分散処理に移行していった．また，クライアントにおいてはワークステーションやパーソナルコンピュータなどの性能が高まり，クライアント・サーバシステムが普及した．1992 年 9 月に日本において最初のホームページサーバが設置され，1990 年代後半から 2000 年代にインターネットが普及した．また，管理コスト等の低減を図るためにサーバをデータセンタなどに集約する企業等が増加している．データセンタではコンピュータのリソースを効率よく使用するためにサーバ仮想化技術を採用したシステムが一般的となっている．2010 年代ではバックボーンネットワークの高速化が更に進んだことにより，クラウドコンピューティングが幅広く活用されるようになった．現代社会においてインターネットは重要な社会基盤の一部となっており，今後もその重要性が増すことは明らかである．現在ではタブレットやスマートフォンといった高性能な移動体端末の普及が加速するとともに，光通信技術の発展によるバックボーンネットワークの大容量化と Fiber To The Home（FTTH）や Long Term Evolution（LTE）などのアクセスネットワークの高速化および高効率化が実現されている．これらを背景としてインターネットサービスは益々身近な存在となり，我々の生活にとってなくてはならないサービスの 1 つとなっている．そのサービスを安定した環境で提供するためにはネッ

トワークおよびサービスを提供するサーバの役割が非常に重要となり，それらの最適化設計，構築および運用管理も更に重要となる．

ネットワークやサーバシステムの発展は，我々の生活にとって高い利便性を提供している一方，様々な危険性も増大させている．例えばデータの改ざんや漏洩などがあげられる．また，様々な情報を提供している Web システム，銀行の窓口業務や鉄道などの予約業務などはトラブル等によってサービスの中断が発生した場合，社会的影響が非常に大きい．そのため，サーバシステムの信頼性の強化が重要な課題となり，システムが障害なく稼働する信頼性，使用したい時にいつでも使用できる可用性，トラブル発生時における障害の検出，診断や切り離しなどにおける再構築の容易さである保守性，トラブルによるデータ破壊や消去などが発生したとしても修復可能となる保全性およびシステムに対する不正アクセスなどを防ぐ安全性を考慮する必要がある．

ネットワークにおいては種々の故障検出や管理手法に関して報告されており，大規模ネットワークにおいて検査すべきノードの効率的な発見方法[1]，ネットワークにおける通信遅延や通信異常の効率的な監視方式[2],[3]，インターネットにおけるバックボーンネットワークのリンク障害の特性調査を行い，その 70%は単一リンクに関連した障害であることを分析[4]した報告がされている．また，光ネットワークにおける障害検出や効率的な障害位置の特定[5]，リモートノードのリモート再構成を用いた保守方式の提案[6]，ネットワーク管理において，冗長化されているネットワークを使用して高速な故障回復の手法[7]や通信故障を動的に対処するため，パケット伝送を良好に保った上で故障ノードを回避する光学的バイパス方式を実現[8]している．

また，ネットワークにおける高速化に関しても多くの報告がなされており，ファイアウォールにおけるパケットフィルタリング時間を最小化するパケットフィルタリング動的最適化手法の採用[9]，高速でスケーラブルなコンテンツ配信システムの構築に関して分散型

ネットワークの設計原理[10]を示している．また，40Gbps と 100Gbps イーサネットにおける両方の規格化に関して両方を総括した柔軟性のあるアーキテクチャの検討[11]や100Gbps イーサネットにおける運用上の信頼度を増すための検討[12]がなされている．さらに，インターネット経由で大規模コンテンツ配信を高速化するための Bit Torrent 型ネットワークの性能解析[13]，インターネットにおけるデータ転送量の急速な増大により，高速化やスイッチング装置のスケーラビリティの点で電気的なパケット交換から光パケット交換への移行が必要であることを示し，スケーラブルな光パケットスイッチの開発[14]や高速，高遅延ネットワークにおける高効率な TCP におけるネットワーク帯域の利用に関する検討[15]が行われている．

さらに，クライアント・サーバシステムにおいて，処理の効率化に関して様々な検討がなされており，マルチメディア通信用にネットワークの状態，メモリの性能や CPU 負荷にもとづいた円滑なマルチメディアストリーミングの実現[16]，有線および無線ネットワーク上でのプロキシサーバ経由または経由なしの状態におけるマルチキャスト通信のスループット性能を分析[17]，クライアント・サーバシステムでよく見られるマルチレベルキャッシュ階層における効果的なバッファキャッシュ管理プロトコルの提案[18]やセッション制御系における耐障害性は冗長化やクラスタ化の導入で実現可能だが，サーバ数の増加を招く．この問題を解決するためのサーバ選択ポリシーを提案し，解析モデルを開発[19]している．また，モバイルコンピューティング環境における新しいキャッシュ無効化技法を提案し，従来の方式に比べ効率の良いキャッシュ無効化の実現[20]や分散処理において広く使用されている負荷分散の問題に対し，全体的な平均応答時間の改善方法[21]を示している．さらに，クラウドコンピューティングシステムにおけるマルチリソース割当てのためのシステムモデルを提案し，リソースの利用率の改善に関してシミュレーションでの調査[22]，動的モバイル無線ネットワークにおける局所的リソース共有の構築方法に関する提案[23]がな

されている．

ネットワークやサーバシステムを構築する上で高速化，信頼性に加え省電力についても十分に検討することは非常に重要であり，種々の省電力化方式が報告されている．Wireless local area network の利用増加に向けて省電力アクセスポイントの設計手法の提案[24]，イーサネットインタフェースの使用時と Idle 状態においてモード切り替えを行う方法を提案し，実験的に電力削減効果の調査[25]や 1Gbps イーサネットは 100Mbps より 4W 消費電力が高く，10Gbps においては 10 から 20W の消費電力増加が見込まれるので，イーサネットにおけるエネルギーの削減を行うため，適応リンク率の採用による省電力化の提案[26]をしている．また，モバイルネットワークに特化した省電力手法[27],[28]，サーバにおける平均応答時間を確保した上で，電力スリープによる平均電力消費を最小化する方式の検討[29]，物理スイッチで構成されたデータセンタのネットワーク上に低消費電力で性能要件を満たす仮想ネットワークの構築[30]が検討されている．大規模データセンタでは数千台から数万台のサーバが配置されており，サーバ間に十分な通信帯域を確保する必要があるため，低消費電力で十分な通信帯域を実現するには光パケットスイッチを用いることが考えられる．しかし，光パケットスイッチは帯域が大きいため 1 台の故障における影響が大きい．そこで，故障が発生しても十分な帯域が確保可能な光パケットスイッチを用いたデータセンタネットワークの構造を提案[31]している．さらに，光ネットワークにおいて使用するネットワークリソースが最小となる組み合わせを計算し，使用しない光インタフェースの電源を切ることで省電力化の実現[32]やネットワーク全体のトラヒックを特定リンクに集約し，不要なインタフェースの電源を切ることでネットワークの省電力化を行うための高速トポロジ計算手法の提案[33],[34]がなされている．加えて，サーバシステムにおける様々な報告もされている．マルチコアサーバにおける CPU のエネルギー効率の改善を最大化する検討[35]，データセンタにおける仮想サーバの台数を最小化することにより省電力化の実現[36]

などが報告されている.

一方，サーバシステムの設備投資においては，サービスを提供することによって得られる利益を超えない範囲で行うのが一般的である．設備投資において重要となるのが，ユーザにサービスを安定して提供するために，サーバ等の冗長化を含む耐障害性の検討である．また，省電力化を含む運用コストの削減も重要な課題となる．一般的なサーバシステムの構成を例として構築上の問題点を示す. 図 1.1 に東京工芸大学の情報教育や研究を支えるため, 情報処理教育研究センターによって構築された PC 演習室システム (Tokyo Polytechnic University Computer Education Laboratory System: TPUCELS）の構成図を示す．これは筆者が情報処理教育研究センターに所属していた時期に設計, 構築したシステムで, 2007 年 9 月から 2010 年 8 月まで実稼働していた構成である. すべてのサーバとクライアントは 1000BASE-T と 100BASE-TX のネットワークによってメインルータに接続され, サーバシステムは 12 種類，計 24 台から構成されている．本システムは学籍番号末尾の奇数偶数に応じて使用する File サーバを選択する仕組みとなっており，2 台でユーザデータの管理を行う．Filedc サーバはデータの配布や回収に利用する．Backup サーバは File サーバに保存されているユーザデータのバックアップをハードディスクに行う．SMS サーバは Windows update やウイルスソフトのパターンファイルを管理，PXE サーバはクライアントにおける起動方法の制御を行う．Proxy サーバはクライアントの代理として学外の Web サーバへのアクセスおよび Web データのキャッシュ管理を行う．Remote desktop サーバは TPUCELS の使用を外部のクライアントに許可する．Mail，Web，Login，Account，Print サーバは一般的なサーバより説明は省く. TPUCELS は同機能サーバを複数台設置するグループと単体サーバのグループを持つ．すべてのサーバの中で教育研究に密接に関連しているサーバには，信頼性向上のため冗長化構成としている．直接的に関連していないサーバには導入コストや運用コストといった経済的理由から冗長化を行っていない．ただ

し，これら単体で接続しているサーバが故障した場合，TPUCELS からそのサーバの機能が失われ，その結果，教育や研究において何らかの影響を与えることは否定できない.

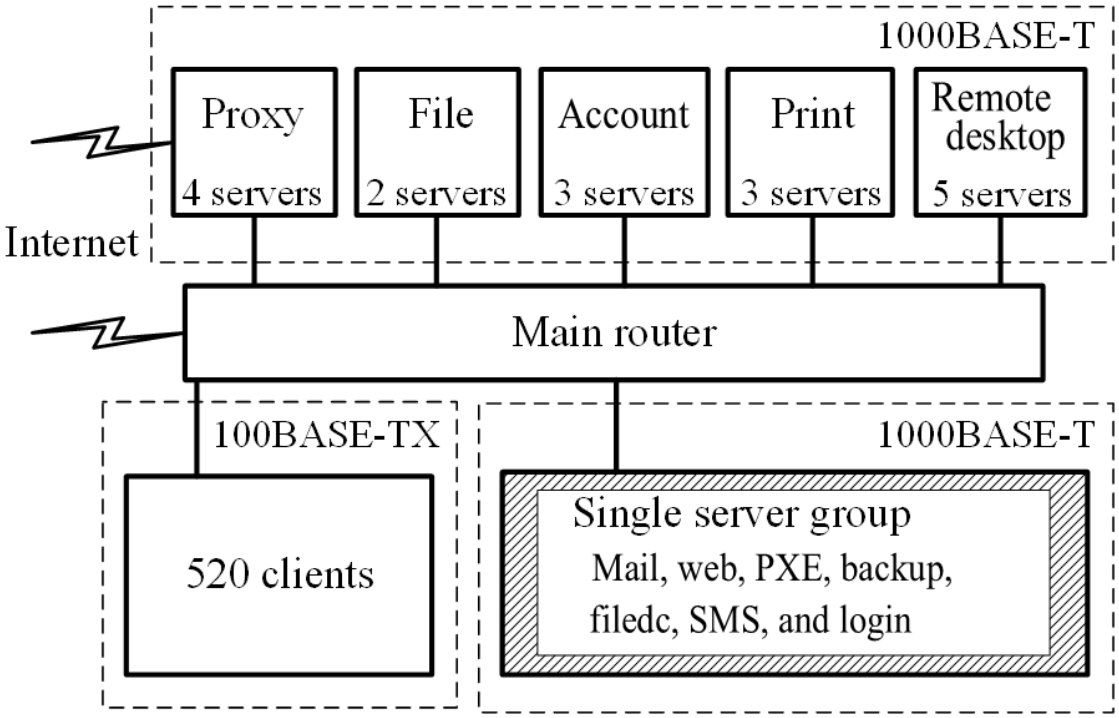


図 1.1　TPUCELS の構成

米 EMC 社が 2012 年 6 月 15 日に発表している災害復旧に関する調査報告によれば，日本企業の 50%が，過去 1 年間にデータ損失またはシステムダウンを経験したと回答しており，データ損失やシステムダウンの主原因のひとつとしてハードウエア障害が 55%にも達している[37]．システムダウンの原因に関しては以下の 4 つに分けられる[38].

・ソフトウエアの不具合

・性能や容量不足

・設定や操作ミス

・不慮の事故

ソフトウエアの不具合としてはアプリケーションソフトウエアのバグがその殆どを占め，OS やミドルウエアの不具合も含まれる．不慮の事故としてはハードウエア故障，停電や災害などによるものが考えられる．性能や容量不足，設定や操作ミスに関しては人為的ミスおよびシステム構成の設計ミスやシステムのサイジングミスが考えられる．最適なシステム設計やシステムサイズであったとしても，人為的な操作ミスや設定ミスなどでシステム障害は発生してしまう．人為的ミスを防止するには，すべての場面において 2 人で確認しながらオペレーションすることが望ましいが，多くの現場ではコストの問題から困難となる．人がオペレーションする上で，ミスは当然想定されるものでありシステム的な防止策というのは困難である．また，システム構成の設計において，各種サーバやネットワーク機器の仕様，負荷分散方法，バックアップの方法，障害時の対処方法やサーバのサイジングなどは，エンジニアの経験やノウハウに頼ることが多い．サーバサイジングとは，サービスを提供する場合に想定されるシステムの負荷を処理可能なサーバの性能や台数を用意することおよびシステムの運用開始後に負荷の増大に応じてサーバ性能を増強することである．適切なサーバサイジングを行うには，同時接続ユーザ数や CPU の使用率，ネットワークのスループットなど，システムに与えられる負荷の平均値や最大値を適切に調査して見積もることが重要であり，複数台のサーバを利用する場合には適切な負荷分散の考慮が必要となる．負荷が増大したときにシステムが停止しないよう，処理のピーク時に合わせたサーバサイジングを行うことが多く，一般にシステム負荷の平均値はピーク値よりも大幅に低いため，必要以上に大きなサイジングをしてしまう傾向があり，システム構築に関して無駄な投資金額が発生する．ただし，最終的にはコストとのトレードオフによって決定されることから，システムのサイジングミスを誘発する可能性が生じる．システム障害を避けたければ，潤沢な予算でシステム構築をするしかない．

以上から，サービス提供ができなくなる事態の発生頻度を少なくする高可用性サーバシステムの構築を低コストで実現する方法を確立することの必要性が必然的に導き出される．

そこで，本論文において，サーバシステムを安定稼働するために必要となるサーバ管理システムに関し，低コストの環境下で高可用性を実現する方式の確立に重点をおき，システムの構造や機能の検討を行うとともに実験システムを構築し，その動作検証および分析を行う．高可用性サーバシステムを構築する上で，サーバ管理システムの稼働状況における高信頼性，消費電力やコストについて技術的な検討を行うことは非常に重要な研究であると確信している．

先行研究としては以下があげられる．サーバシステムの構築に関してサーバ故障を考慮し，通信機器，サーバや周辺機器を多重化して信頼性の向上を図るウォームスタンバイシステムやホットスタンバイシステムが一般的に採用されている．近年では同じシステムを複数用意して負荷分散クラスタ構成を構築し，耐障害性および分散処理による性能向上も実現している．

サーバトラブルに関してハードウエア故障とソフトウエア故障の両面を考慮し，サービス中断時間を大幅に短縮する方式の検討がなされている[39]．また，サーバ管理システムにおいて低コストと省電力を実現するためにサーバ仮想化技術を使用し，１台の実サーバで複数台の実サーバを管理するシステムが検討されている．このシステムでは管理プログラムの複雑化を防ぐために，管理対象サーバ毎に１つの管理プログラムを実行する形式を導入し，それぞれの管理プログラムは独立して動作する方式[40]を採用している．さらに，サーバの動的な追加や削除において，追加サーバのデータ引継ぎによる負荷の抑制やデータの再配置および再冗長化による影響を最小化することにより，サーバの通常処理に影響を与えない高可用サーバクラスタシステムにおける動的構成変更方式を提案[41]している．

近年では，クラウドコンピューティングの普及により大規模なデータセンタが出現して

いる．データセンタの規模が大きくなるにつれて，サーバの故障も多くなることが予想され，サービス提供者における利益の減少が懸念される．それを改善するためにサーバの故障状態と修復状態を考慮した管理モデルを提案し，シミュレーションによりその改善を示している[42]. また, データセンタにおける高可用性仮想サーバシステムの構成方法として, 仮想サーバ故障時にそのサーバの全メモリ状態を瞬時にコピーしてホットスペアノードを起動するために，ネットワークコーディングを用いて解決する方法を提案[43]している．サーバシステムにおいて信頼性の高いネットワーク構築は必須となる．そのため，通信経路が故障により切断したとしても一定距離以内にあるサーバ数を維持して通信の継続を図るネットワーク設計[44]や省電力性と耐故障性の両立を可能とする仮想ネットワークを提供する手法が提案[45]されている．

しかしながら，これらの先行研究では，高可用性に関して焦点をあてているものが多く，また，消費電力に関しては前述したとおり，ネットワークに関する報告は[24]-[34]に比べ，サーバシステムに関する報告は[35],[36]に留まっている．また，データセンタにおける仮想サーバの配置やCPUなどの機器に注目した報告が見受けられるが高可用性を考慮していない．さらに，高可用性サーバシステムの構築に関する報告において，初期費用や運用コストに関する考慮は殆どされていない．

1.2 本研究の目的

高可用性サーバシステムはインターネットサービスを含む各種コンピュータサービスを安定して提供するために必須となるシステムである．本研究の目的は高可用性サーバシステムを構築する上で，サーバ管理システムの稼働状況における高信頼性，消費電力やコスト（初期費用や運用コスト）について技術的な検討を行い，その構築技術を向上することである．

高可用性サーバシステムを実現するには，システムにおける冗長化構成の構築，トラブル発生時の復旧手段の確立や災害対策システムを講じる必要がある．以下に高可用性サーバシステムの代表的な構成を示す．コールドスタンバイ構成は，実稼働しているサーバと同様の構成であるバックアップ機を用意しておき，障害発生時にバックアップ機で業務を継続させる方式で，1 台のみの運用時よりも可用性は高まるが，電源を入れることや差分データを取り込むなどの作業が発生するため，若干のサービス提供停止時間が発生する．ホットスタンバイ構成は，同様の構成をしたサーバを 2 台以上用意し，1 台が使用不可能となった場合に自動的に系の切り替えを実施し，サービスの提供を継続させる構成である．コールドスタンバイ構成よりもサービスの提供が停止する時間が短いため，可用性は更に高まるが，待機させているサーバに対する運用管理や保守も必要になってくることから運用コストは 2 倍以上となる．負荷分散構成は，トラヒックの分散や負荷軽減などを実施する．その構成は，常に同様の機能を提供する 2 台以上のサーバを稼働させるためコストは高額となるが，障害発生時は正常なサーバのみで縮退運用を実施するため，サービスの停止時間はほぼゼロとなる．災害対策構成としては，実稼働しているデータセンタとは別の地域にデータセンタを配置し，天変地異などによって実稼働しているデータセンタでの運用が継続できなくなった場合でも，ユーザへのサービス提供を継続できるようにしている．クラスタリングは計算に特化したシステムで，多数の比較的安価なコンピュータを特別なソフトウエアやハードウエアを使用して，あたかも 1 つの大きなコンピュータとして利用できるように構成することで，1 台のコンピュータが動作しなくなった場合でも他がその作業を引き継ぐことが可能となるため，コンピュータにおいて高い信頼性や性能を低コストで得る手段として利用される．

サーバシステムの高可用性を実現する上で，本研究では最初に効率的なサーバトラブルの検出方法およびトラブルを起こしたサーバの機能を継続することに関して検討を行う．

特に，その機能を継続する上で低コストな環境を実現することが重要となる．そこで，それらの条件を満たしたサーバ機能の継続方法の１つとしてサーバ仮想化技術を採用したシステムの検討を行い，その妥当性も検討する．

冗長化されていないサーバが提供しているサービスは，そのサーバが故障すると提供できなくなる．本研究では，低コストで構築可能なサーバ機能バックアップシステムを開発し，その提供できなくなる時間の短縮を検討課題としている．付録の F1 や F2 に示すようにハードディスクの年間故障率や MTBF（平均故障間隔）から簡易的にハードウエアの故障する確率が算出でき，また，付録の F3 に示すようにサーバの故障率と平均サーバ故障台数の関係から故障サーバの復旧時間が短い場合，平均故障台数を抑制できることを示している．図 1.2 に冗長化されていない一般的なサーバシステムと比較し，サーバ機能のバックアップシステムを保有した改善システムによる稼働率の向上について示す．

(a)は冗長化されていない一般的なシステムの状態を示し，T_i は故障しているサーバが存在しないサーバシステムの稼働時間，t_iは故障サーバの再起動時間や故障部分の修理に要する時間を示す．システムのための稼働率は式(1.1)によって示される．

$$\text{Availability rate} = \frac{\text{Average of } T_i}{\text{Average of } T_i + \text{Average of } t_i} \tag{1.1}$$

(b)はサーバ機能のバックアップシステムを保有した改善システムの状態を示し，T_i と t_i は(a)と同様とする．T_{ei}は改善システムによって追加された稼働時間を示し，t_{bi}はサーバ故障後，改善システムによってサーバ機能の復旧を行うためにバックアップシステムが起動するまでの時間を示す．改善システムにより向上した稼働率は式(1.2)となる．

$$\text{Improved availability rate} = \frac{\text{Average of } T_i + T_{ei}}{\text{Average of } T_i + \text{Average of } t_i} \tag{1.2}$$

実際のシステムにおいてサーバが故障すると，一般的にそのサーバの復旧には長い時間を要する．しかし，改善システムはトラブルを起こしたサーバの機能を継続することが可能となるため，サーバシステムの稼働率は向上される．

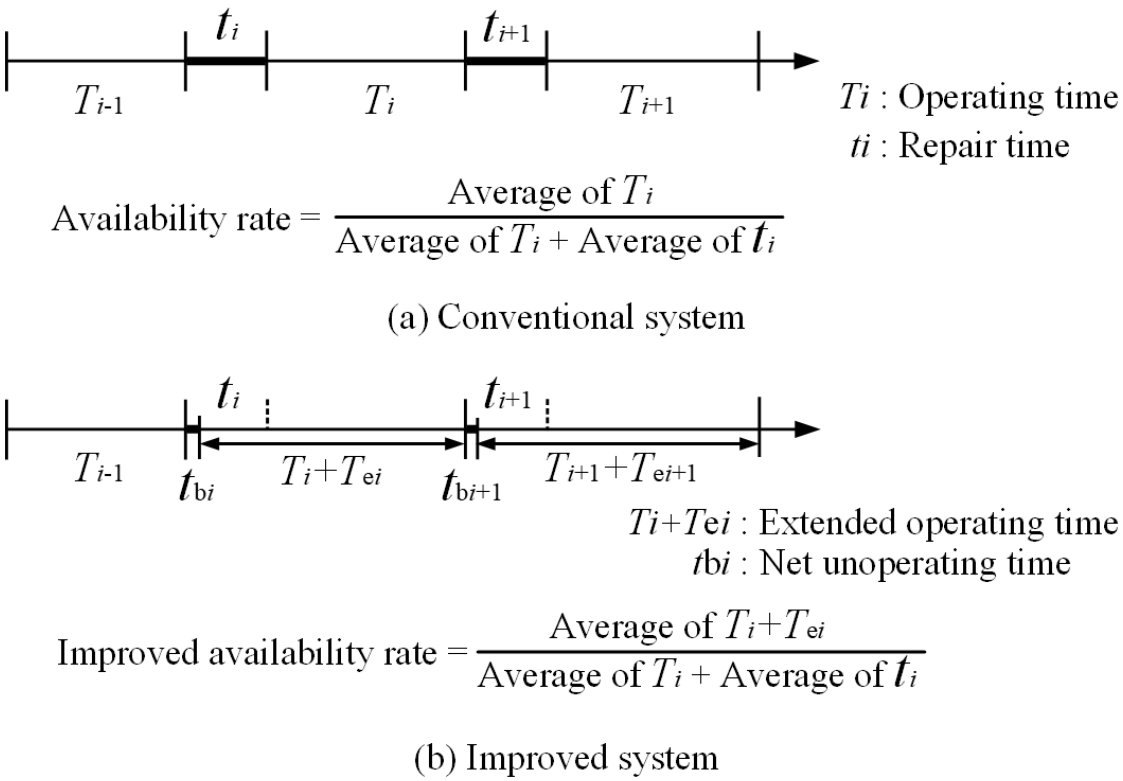


図 1.2　サーバシステムにおける稼働率の向上

次に，省電力かつ高可用性サーバシステムを構築する上で，それらの制御プログラムを簡単化するために，独立して稼働可能な省電力サーバシステムと高可用性サーバシステムの制御プログラムを交互に動作させる異種システム混合処理方式を提案し，それを実現するために必要となる条件や稼働状況について検討する．また，それらの制御プログラムを簡単化する別の方法として，省電力サーバシステムと高可用性サーバシステムの 2 つのシステムを仲介するシステムを稼働させ，それらの複合動作を実現する方式について提案し，検討を行う．さらに，ネットワークシステムにおいては，クライアント・サーバ方式に比べ Peer-to-Peer（P2P）方式がサーバの耐障害性に優れることから，P2P 方式の特徴を導

入したサーバ管理システムを提案し，その方式を採用する上での課題点や稼働状況について検討を行う．これらの技術を追求し，高可用性サーバシステムの構築技術を確立する．

1.3 本論文の概要

以上のような観点のもとに，本研究における低コストで高可用性サーバシステムを構築するためのサーバ管理システムに関する論文の構成を図 1.3 に示す. 本論文は 7 章により構成されている．本論文で提案（開発）および導入するすべてのシステムは，2 章で示すサーバ管理システムが基本となっており，3 章では 2 章を発展させ，省電力システムとの混合処理を実現している．また，3 章で示す省電力システムおよび 2 章で示すサーバ管理システムを発展させ，それらの複合動作を実現したシステムを 4 章に示す．5 章では 2〜4 章で示す特定の管理サーバを必要とするシステムとは異なる新たなサーバ管理システムについて示す．以下に各章の概要を示す.

第 2 章では低コストで高可用性サーバシステムを構築可能な複合サーババックアップシステム（Multiple Server Backup System: MSBS）を提案する．MSBS は動的バックアップサーバシステム（Dynamic Backup Server System: DBSS）を基本とし，それを多重起動することで 1 台の実サーバで複数種類のサーバ機能をバックアップ可能とする．DBSS の構築に関しては，様々なサーバシステムへの適合性を向上するため，モード分割方式を提案し，採用する．ここでは，Mail サーバを管理対象とした DBSS の評価実験および Mail サーバと Web サーバを管理対象とした MSBS の評価実験を行い, 複数種類のトラブルに対して MSBS が対処可能であることを確認する．さらに，DBSS は管理対象サーバ故障時の対処として仮想サーバを使用するため，その機能を評価することも重要となる．評価に関しては，ネットワークアクセスによる応答時間で評価する．2.2 節では MSBS の基本となる DBSS における管理対象サーバの異常検出方法やそのサーバ機能を復旧するための対処

方法を示す．2.3 節では MSBS を導入した実験システムを構築し，そのシステムの仕様および特性を示すとともに，サービス提供プログラムおよびネットワークのトラブルを再現する実験を行い，MSBS がトラブルの状況に応じたサーバ機能およびサーバ自体の復旧が可能であることを示す．また，復旧時間の予測も行う．2.4 節では DBSS において仮想サーバの機能が非常に重要な要素となることから，仮想サーバの性能調査システムの構築およびその仕様を示すとともに，それによる仮想サーバの適用性を評価する．

第 3 章において，複数のシステムを交互に動作させ，それらの機能を実現する異種システム混合処理方式を提案し，本方式を採用した実験用の省電力かつ高可用性サーバシステムを開発して稼働実験を行う．本システムは，高可用性および負荷分散を行う上で一般的に使用されるロードバランサを導入したサーバシステムを基本とする．ここで，省電力サーバシステム（Power-saving Server System: PSS）や高可用性サーバシステムはそれぞれ独立して動作可能なシステムであり，本提案方式により，それらを交互に動作させ，省電力かつ高可用性なサーバシステムを構築する．本章では，本方式を採用した実験システムを構築し，その構成を示すとともにサービス提供用の仮想サーバまたはそのサーバを起動している実サーバに対してトラブルを再現させる実験を行う．通常のサーバ稼働時と省電力化されたサーバ稼働時において，本システムは複数種類のトラブルに対し，自動的にその状況にあった対処を行うことが可能であること，また，その対処に必要となる時間を示す．3.2 節では，提案システムの全体構成および異種システム混合処理方式や構成ファイルに関して示す．3.3 節では，PSS の特徴やサーバ負荷の実測および最適なサーバ状態の判定方法を示す．3.4 節では，高可用性サーバシステムにおけるサーバ機能復旧方法の詳細およびサーバシステムが通常状態や省電力状態において，仮想サーバや実サーバにおける故障への対処方法を記述する．3.5 節では，本提案システムを採用した実験システムを構築し，その構成や仕様を示すとともにハードウエア故障を含むサーバトラブルを再現し，提案シ

ステムがサービス提供プログラムおよびネットワークのトラブルに対処可能であることを示す.

第 4 章では，省電力かつ高可用性サーバシステムの構築において，独立して動作可能である，MSBS と PSS の複合動作方式を提案する．本章での MSBS と PSS にはシステム環境に応じた変更を加えている．通常，2 種類の管理プログラムを複合動作させる場合，両方のプログラムにおいて誤動作を防止するためにそれらのプログラムは複雑となる．ここでは, MSBS を構成する DBSS が PSS の構成ファイルを調査および編集可能にすることで複合動作を実現する．そのため，DBSS のプログラム構成が複雑になることを防ぐために，2 つのシステムを仲介するシステムインタフェース（System InterFace: SIF）を提案して導入する．DBSS にファイルアクセスと編集機能を追加するのではなく，SIF にその機能を追加することにより，DBSS のプログラム構成がシンプルな状態を維持できることを示す．また，提案方式を導入した実験サーバシステムを構築し，通常状態または省電力状態におけるサーバ構成において，サービス提供プログラムおよびネットワークのトラブルを再現する実験を行い，サーバ機能の復旧時間を実測する．本提案方式を導入した省電力かつ高可用性サーバシステムは，それらのトラブルが発生した場合でも動作を継続可能であり，サーバ障害に対して迅速に復旧可能であることを示す．4.2 節では提案システムの構成概要を示し, 本章で使用する MSBS と PSS の動作概要およびその詳細を示す. 4.3 節では MSBS と PSS の複合動作における問題点を示し，その解決方法として 2 つのシステムの仲介を行う SIF の詳細を記述する．4.4 節では本方式を採用した実験システムを構築し，その構成および仕様を示すとともに，通常状態や省電力状態において，管理対象サーバにサービス提供プログラムおよびネットワークのトラブルを再現する実験を行い，提案システムがそれらのトラブルに対処可能であることを示す.

第 5 章では，ネットワークシステムのデータ取得において，クライアント・サーバ方式

に比べサーバの耐障害性に優れる P2P 方式の特徴を取り入れ，管理対象サーバの台数に依存しないサーバ管理システムを提案する．本提案システムは故障したサーバの機能を復旧するために予備の実サーバではなく仮想サーバを使用する．本提案システムを構築するため，P2P 方式で考慮する必要がある管理プログラムの分散による管理の複雑化を防止する方式や復旧処理における誤動作を防ぐための管理の優先順位および複数サーバの同時故障に対処するための動的管理拡張方式を提案し，導入している．また，本方式を採用した実験システムを構築し，その構成を示すとともにサービス提供プログラムやネットワークのトラブルを再現する実験を行い，管理対象サーバの機能および管理対象サーバ自体の復旧に要する時間を測定し，それらの予測も行う．さらに，複数台の管理対象サーバに前述のトラブルを再現させた場合においても，サーバ機能の復旧が可能であることを示す．5.2 節では提案方式の構成概要および管理プログラムの分散による管理の複雑化を防ぐ Executer システムの動作手順，サーバ管理に必要となる構成ファイルや管理対象サーバ群の負荷状態を考慮した仮想サーバの起動方式について示す．5.3 節では管理対象サーバを監視する上での管理プログラムの処理手順，管理における優先順位の必要性やそのサーバ故障時の対処方法および動的管理拡張方式の特徴を記述する．5.4 節では本提案方式を採用した実験システムの構成およびその仕様を示し，管理対象サーバにサービス提供プログラムおよびネットワークのトラブルを再現する実験を行い，提案システムがそれらのトラブルに対処可能であることを示すとともにその復旧時間の実測および予測も行う．

第 6 章の考察では，2～5 章で示したシステムの想定規模やその拡大方法および総合評価を考察し，最後に第 7 章の結論では，本研究の成果をまとめるとともに，今後に残された課題について言及する．

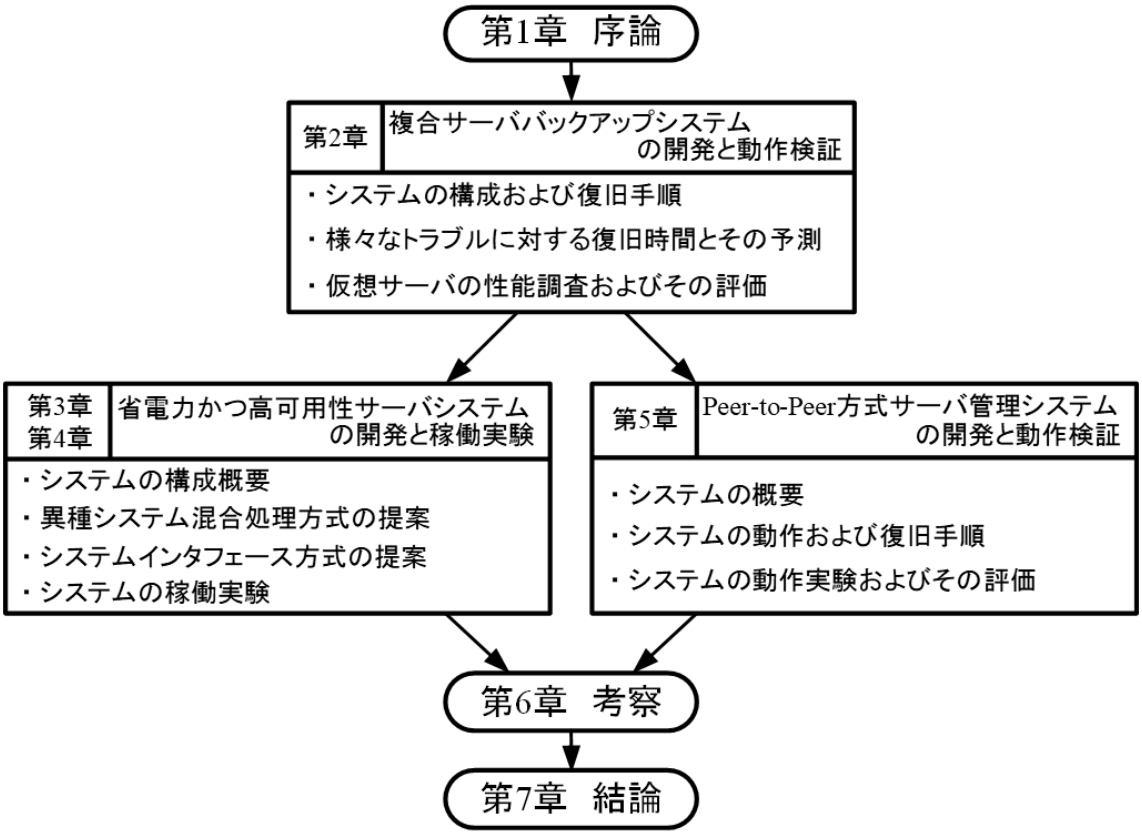

第1章　序論
第2章
複合サーババックアップシステム
の開発と動作検証
・システムの構成および復旧手順
・様々なトラブルに対する復旧時間とその予測
・仮想サーバの性能調査およびその評価
第3章
第4章
省電力かつ高可用性サーバシステム
の開発と稼働実験
・システムの構成概要
・異種システム混合処理方式の提案
・システムインタフェース方式の提案
・システムの稼働実験
第5章
Peer-to-Peer方式サーバ管理システム
の開発と動作検証
・システムの概要
・システムの動作および復旧手順
・システムの動作実験およびその評価
第6章　考察
第7章　結論

図 1.3　本論文の構成

第 2 章

低コストで高可用性サーバシステムを構築可能な MSBS の開発と動作検証

2.1 まえがき

近年，社会の情報化が急速に進展し，インターネットは世界規模のインフラストラクチャとして重要な社会基盤の 1 つとなっている．IP ネットワークにデータ通信と音声通信を統合した次世代ネットワークの商用サービスが開始され，更に多様なサービスの増加が期待される．このような高度なネットワークシステムを構築していくためには，基礎理論から実験，実装や運用までの様々な技術的問題を解決していかなければならない[46]．このため，ネットワークにより相互接続された情報処理システムの基本単位であるサーバシステムの最適化設計および実装は，インターネットシステムの適切な構築法とともに重要な課題であると考えられる[47]-[49]．今後，インターネットシステムは更に社会基盤としての必要な機能が追加され，複雑化することが想定される．そのため，そのサービス対処の高速化および信頼性の高いシステム構築が重要となる[50],[51]．これを実現するには，更に対障害用機器を含む IT 機器の台数増加が予想され，導入コストや使用電力の増加という問題が懸念される．経済産業省の報告では IT 機器による電力消費量が 2025 年には 2006 年に比べると 9 倍になると推計している[52]．これまで，ネットワークシステムの省電力化に関する重要な研究がなされている．次世代通信網を想定したナノデバイスの適用[53]，省電力アクセスポイントの設計[24]や，光ネットワーク機器への制御[54]および LAN スイッチなどのネットワーク機器に関する検討[55],[56]などがある．また，ネットワークシステムを構成するサーバシステムに関しても様々な検討がなされている．分散サーバ設置による転送ト

ラヒックの低減[57], データセンタにおけるサーバ群とネットワーク機器を管理する省電力管理サーバの導入[58]や，クラウドシステムにおける分散情報通信処理アーキテクチャ[59]などの報告がなされている．

しかしながら，ルータやスイッチング HUB などのネットワーク機器とサーバが IT 機器の約半分を占めていることを考慮すると，サーバシステムの検討が少なく，また，サーバシステムにおいてはデータセンタ等に特化した検討が多いように見受けられる．また，サーバシステムの基本単位であるクライアント・サーバシステムにおける検討の報告は非常に少ない．特にクライアント・サーバシステムにおける経済性，高信頼性や省電力の検討を進めることは重要で，それを実現することは次世代ネットワークシステムの構築に役立つ．

そこで, 本章では 1 台の実サーバで, 複数台の実サーバ機能をバックアップ可能な MSBS を提案する．MSBS は DBSS を基本とし，それを多重起動する．DBSS はサーバ機能のバックアップに予備の実サーバではなく仮想サーバを使用することからサーバ管理システムの低コスト化を実現でき，間接的にその消費電力も低減可能である．また，DBSS は 1 台の実サーバのみを管理することから，その制御プログラムは単純な形式となる．DBSS は管理対象サーバを監視し，異常を検出した場合に状況に応じた対処を自動で行う．もし，管理対象サーバの異常を検出した場合には，管理対象サーバと同設定の仮想サーバをバックアップサーバとして起動し，そのサーバ機能を補完する．また，それと同時に管理対象サーバ自体の状態を調査し，復旧した場合には元の状態に戻す．

2.2 MSBS の構成および復旧手順

MSBS はサーバ管理者がサーバトラブル時において一般的に行う復旧作業を自動化している．図 2.1 では MSBS の管理対象サーバとして Mail サーバを例とし，その構成を示す．

Mail サーバに対するサーバ機能の復旧作業は，管理サーバの MSBS と File サーバとの共同作業によって実現される．MSBS の構築における重要なポイントは管理対象サーバにおけるサービス提供プログラムの設定ファイルとそのプログラムが使用するユーザデータの保存場所である．もし，管理対象サーバ上に設定ファイルやユーザデータを保存している状態でサーバが故障すると，MSBS は必要なデータを使用することができない．そのため，MSBS は，Network File System（NFS）を使用して File サーバのディスク上にそれらのデータを保存することが構築条件となる．

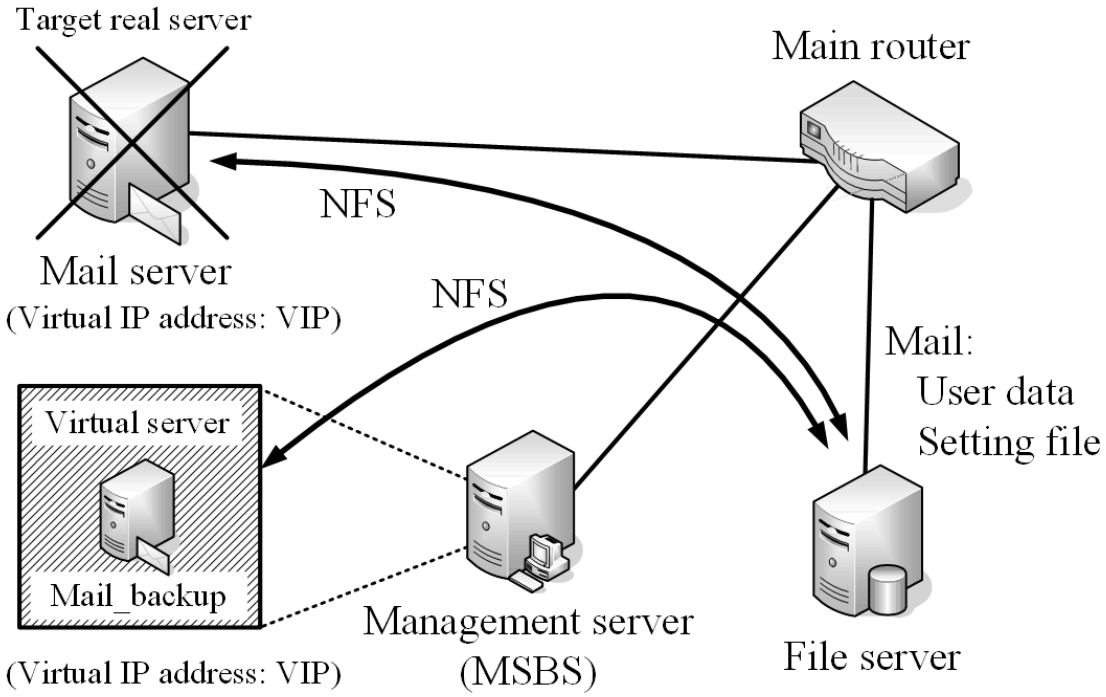


図 2.1　MSBS による Mail サーバ機能の復旧

Mail サーバが故障した場合，管理サーバの MSBS はその故障を検出し，故障した Mail サーバと同様のサービス提供機能を持つ仮想サーバである Mail_backup をバックアップサーバとして起動する．Mail サーバに設定していた仮想 IP アドレス VIP をその仮想サーバ

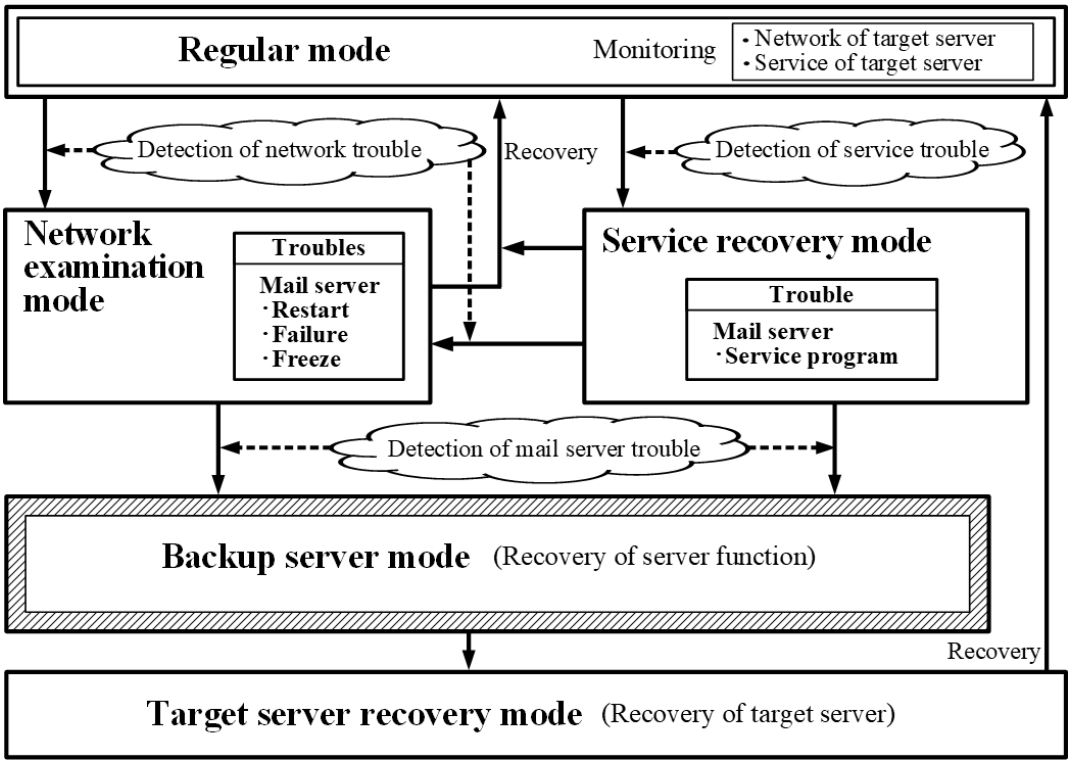
に設定し, その仮想サーバは NFS を使用して Mail サーバのデータを使用する. このため, クライアントを利用しているユーザに対して Mail サーバの故障による影響は最小限に抑えることができ，ユーザはそのサービスを継続して受けることができる．仮想サーバは仮想サーバ用システムデータである VM 構成ファイルと VM ディスクイメージファイルから構成される．もし，Mail サーバ以外のサーバを管理する場合には，そのサーバの機能を有する仮想サーバ用システムデータを作成し，そのサーバを管理するための DBSS を実行することで可能となる．そのため，MSBS は予備のサーバとして実サーバの代わりに仮想サーバを使用することで冗長化サーバシステムを構築できることから，低コストで高可用性サーバシステムを構成することが可能となる．本システムでは前提として Mail サーバの修理時間中のみ Mail_backup がその役割を実行するため，Mail_backup は管理対象外としている．

2.2.1 モード分割方式

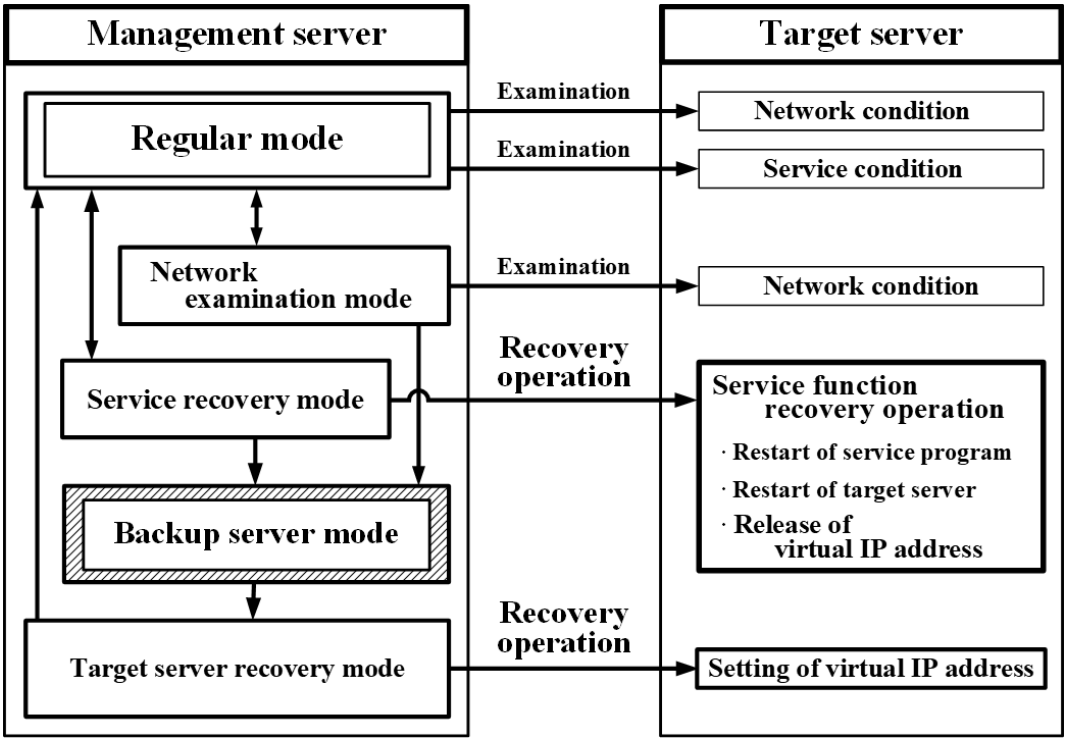
MSBS を構成する DBSS における管理とは，管理対象サーバの監視および異常を検出した場合にその対処を実施することや管理対象サーバの機能を復旧するためにそのサーバ機能を持つ仮想サーバを起動することを示す．DBSS が使用する管理プログラムではモード分割方式を採用している．ここでのモードとはサーバ管理用のソフトウエアシステムで，それぞれのモードはお互いに独立しており，複数のモードを組み合わせて 1 つのシステムを構成することができる．それぞれのモードには必要情報を提供することで動作可能とし，様々なシステムに導入する場合を考慮してシステム制御設計を簡単にしている．特に，本方式は様々な OS の管理対象サーバを管理することを考慮しており，管理対象サーバの OS に依存するモード部分の変更のみで動作可能としている．

図 2.2 にモード分割方式による DBSS の構成を示す．DBSS は Regular mode，Service

recovery mode，Network examination mode，Backup server mode と Target server recovery mode の 5 つのモードから構成され，(a)に各モードの機能を示す．Regular mode は一定間隔毎に管理対象サーバのネットワークとサービス提供状態の調査を行う．Regular mode において管理対象サーバのサービスにトラブルを検出した場合，管理プログラムの制御状態を Service recovery mode に移行する．ここでは，トラブルの状況に応じたサービスの復旧作業を行う．サービスが復旧した場合には Regular mode に戻り，復旧不可能と判断した場合には，管理プログラムの制御状態を Backup server mode に移行する．また，Regular mode または Service recovery mode において管理対象サーバのネットワークに異常を検出した場合には，その制御状態を Network examination mode に移行する．ここでは，ネットワークの状況を再調査し，ネットワークが正常の場合には Regular mode に戻り，異常と判断した場合には，その制御状態を Backup server mode に移行する．Backup server mode では，管理対象サーバの機能を復旧する．その後，管理対象サーバの復旧状態に応じた処理を行うために Target server recovery mode にその制御状態を移行する．管理対象サーバが復旧した場合にはクライアントからのアクセスを管理対象サーバに戻し，制御状態を Regular mode に戻す．(b)に各モードにおける管理対象サーバとの関連を示す．Regular mode は管理対象サーバのネットワークおよびサービスの提供状態を調査，Network examination mode は管理対象サーバのネットワークに関して再調査を行い，Service recovery mode は管理対象サーバのサービス機能を復旧するための処理を行う．復旧処理としては，状況に応じて，サービスを提供しているプログラムの再起動や管理対象サーバの再起動を行う．もし，復旧できない場合には管理対象サーバに対して仮想 IP アドレスの設定解除を行う．Backup server mode は管理対象サーバへのアクセスは行わない．Target server recovery mode は管理対象サーバが復旧した場合，仮想 IP アドレスの設定を行う．



(a) 各モードの機能



(b) 管理対象サーバとの関連

図 2.2　モード分割方式による DBSS の構成

ここで，図 2.2 に示すように，Service recovery mode と Target server recovery mode は管理対象サーバの OS に依存した処理が必要となるため, このモードの制御プログラムを編集することにより，様々な種類の OS を採用している管理対象サーバを制御可能となる．

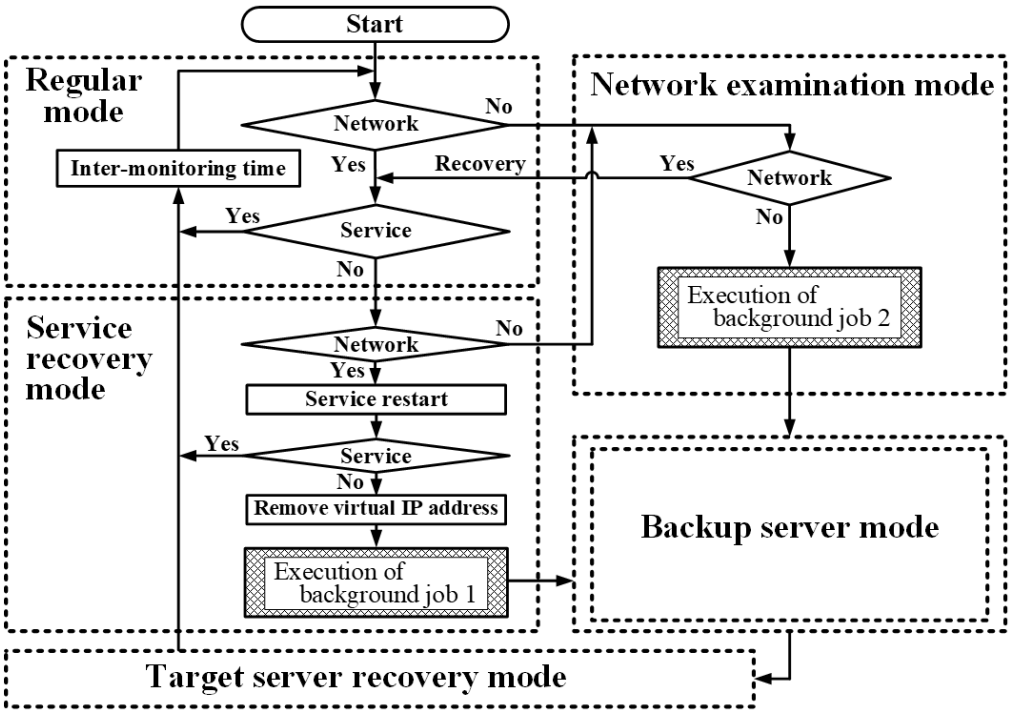
2.2.2 モード処理

図 2.3 に管理プログラムの動作フローチャートを示す．この管理プログラムには，管理対象サーバを復旧するために時間を要する処理が含まれる．そのため，通常処理はフォアグラウンドジョブとして実行し，その時間を要する処理はバックグラウンドジョブとして実行する．

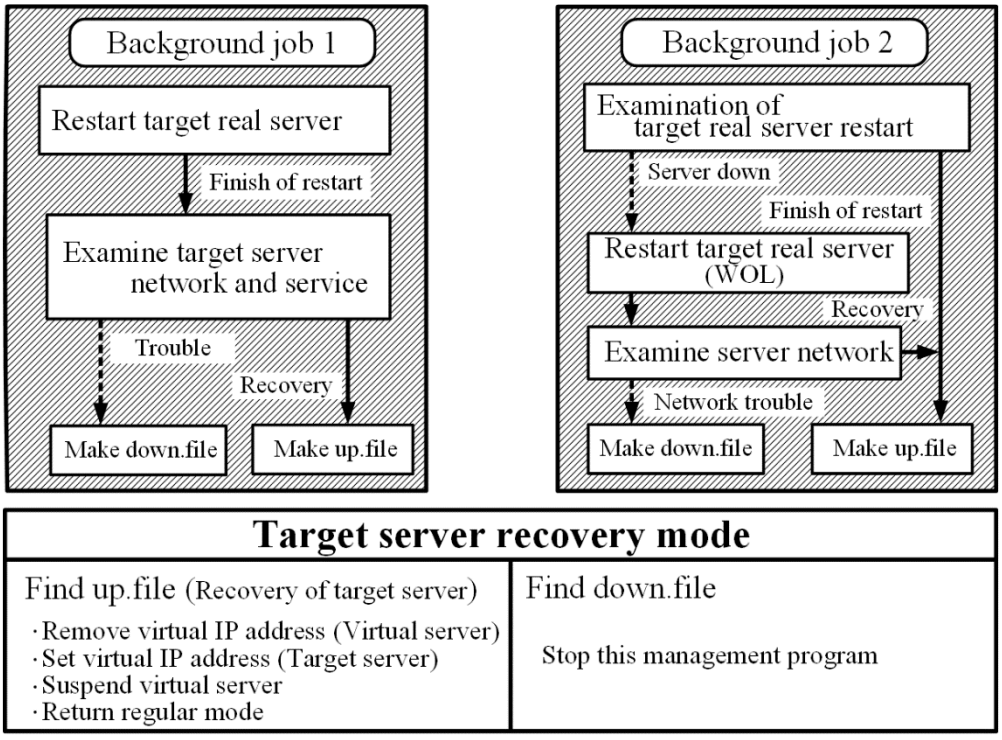
(a)に Regular mode，Service recovery mode と Network examination mode のフローチャートを示す．Regular mode では管理対象サーバのネットワークおよびサービスの提供状態を監視する. 本システムではネットワーク監視に UNIX の ping コマンドを使用している．ここでは Mail サーバを管理対象としているため，管理対象サービスは Simple Mail Transfer Protocol（SMTP）とする．サービスの提供状態に異常が検出されない場合には，監視間隔時間毎に監視を継続し，もし，サービス提供状態の異常を検出した場合には，管理プログラムの制御状態を Service recovery mode に移行する．ここでは最初に管理対象サーバのネットワークを調査し，ネットワークに異常がある場合には，その制御状態を Network examination mode に移行する．ネットワークが正常であれば異常を検出したサービス提供プログラムの再起動を行う．その後，サービス提供状態の調査を行い，正常であれば制御状態を Regular mode に戻す．サービスの状態が改善されない場合，管理対象サーバの仮想 IP アドレスを解除し，Background job1 をバックグラウンドジョブとして実行後，その制御状態を Backup server mode に移行する．ここで，Background job1 は管理対

象サーバのサービスを復旧するための処理で，この処理には時間を要するためバックグラウンドジョブとして実行している．Regular mode にて管理対象サーバのネットワークに異常を検出した場合には，その制御状態を Network examination mode に移行する．ここでは最初に管理対象サーバのネットワークを再調査し, 正常状態であれば制御状態を Regular mode に戻す．異常状態であれば，Background job2 をバックグラウンドジョブとして実行後，その制御状態を Backup server mode に移行する．ここで，Background job2 では管理対象サーバにおけるネットワークの調査や復旧処理を行う．この処理には時間を要するためバックグラウンドジョブとして実行する．

(b)にバックグラウンドジョブの役目と Target server recovery mode の詳細な処理を示す．Background job1 ではサービス提供状態を復旧させるため管理対象サーバを再起動させ，再起動終了後，そのサーバのネットワークおよびサービス提供状態の調査を行う．サービス機能が復旧した場合には up.file を，復旧しない場合には down.file を作成して終了する．Background job2 では管理対象サーバが再起動状態であるか調査を行う．再起動が終了した場合には up.file を作成する．再起動状態ではなく停止状態と判断した場合には，管理対象サーバに対してそのサーバの MAC アドレスを使用して Wake on LAN（WOL）を実行し，起動を試みる．その後，そのサーバのネットワークを調査し，復旧した場合には up.file を，復旧しない場合には down.file を作成して終了する．Target server recovery mode では一定間隔時間毎にバックグラウンドジョブの結果である up.file または down.file の存在を確認する．up.file を検出した場合，管理対象サーバが復旧したと判断し，サーバ機能復旧のため仮想サーバに設定していた仮想 IP アドレスを解除後，管理対象サーバに仮想 IP アドレスを設定する．その後，仮想サーバをサスペンド状態に変更後，Regular mode に制御状態を戻す．down.file を検出した場合，管理対象サーバが復旧不可能と判断し，この管理プログラムを停止する．



(a) Regular mode，Service recovery mode と Network examination mode の処理



(b) バックグラウンドジョブの役目と Target server recovery mode の詳細な処理

図 2.3　管理プログラムの動作フローチャート

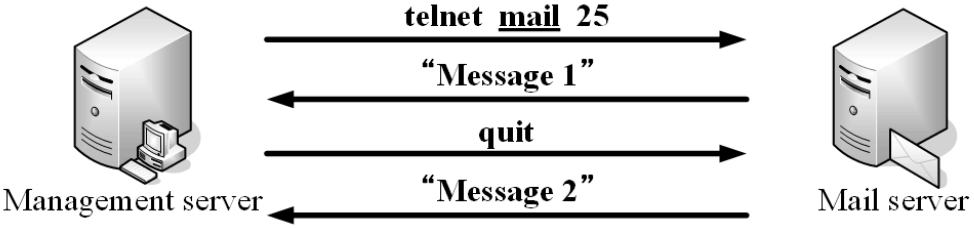
図 2.4 に DBSS におけるサービスの調査方法を示す. サービスの調査には Expect プログラム言語を使用する．Expect 言語は対話形式で行う処理を自動化でき，タイムアウト機能やサーバへのアクセスに対する返答メッセージに応じた処理が可能となるため，確実なサービス提供状態の調査が可能となる.

(a)では Mail サーバが管理対象サーバで，管理サーバは SMTP に関して監視を行う．“mail”は Mail サーバのホスト名を示す. telnet コマンドで Mail サーバの 25 番ポートにアクセスを行うと，そのサーバの SMTP に対応したサービス提供プログラムから“Message1”が返答される．そのメッセージが正常であれば Mail サーバに切断を伝える“quit”を送信する．ここで，“¥r”は Enter キーを押す操作が行われることを意味する．その後，“Message2”が Mail サーバから返答されると“OK”を DBSS に返答する．“Message1”または“Message2”が正常でなければ“ERROR”を，また，Mail サーバからの“Message1”または“Message2”がタイムアウト時間内(この図では 3 秒と設定)に返答されない場合には“NG”を DBSS に返答する．DBSS は“OK”が返答されると，サービスが正常に動作していると判断し，“ERROR”や“NG”が返答されると異常と判断する.

(b)では Web サーバが管理対象サーバで，管理サーバは HyperText Transfer Protocol（HTTP）に関して監視を行う．“web”は Web サーバのホスト名を示す．telnet コマンドで Web サーバの 80 番ポートにアクセスを行うと，そのサーバの HTTP に対応したサービス提供プログラムから“Message3”が返答される．そのメッセージが正常であれば Web サーバに index.html の存在確認を行うため，図に示す“HEAD /index.html HTTP/1.1¥rHost:web¥rConnection:close¥r¥r”を送信する. その後，“Message4”が Web サーバから返答されると“OK”を DBSS に返答する.

```
set timeout 3
spawn telnet mail 25
set ret 0
expect {
   "Message 1" { send "quit¥r" ; set ret 1}
   timeout { send_user NG¥n ; exit }
}
if { $ret == 0 } { send_user ERROR¥n ; exit }
expect {
   "Message 2" { send_user OK¥n ; exit }
   timeout { send_user NG¥n ; exit }
}
send_user ERROR¥n
```

Service examination program



(a) Mail サービス

```
set timeout 3
spawn telnet web 80
set ret 0
expect {
   "Message 3" { send "HEAD /index.html HTTP/1.1¥rHost:web¥rConnection:close¥r¥r" ; set ret 1}
   timeout { send_user NG¥n ; exit }
}
if { $ret == 0 } { send_user ERROR¥n ; exit }
expect {
   "Message 4" { send_user OK¥n ; exit }
   timeout { send_user NG¥n ; exit }
}
send_user ERROR¥n
```

Service examination program

(b) Web サービス

図 2.4　Expect プログラム言語を使用したサービス調査

“Message3”または“Message4”が正常でなければ“ERROR”を，また，Web サーバからの“Message3”または“Message4”がタイムアウト時間内（この図では 3 秒と設定）に返答されない場合には“NG”を DBSS に返答する．DBSS は“OK”が返答されると，サービスが正常に動作していると判断し，“ERROR”や“NG”が返答されると異常と判断する．

また，対象となるサービス提供プログラムの返答メッセージおよびポート番号を変更することで，SMTP や HTTP 以外のサービス調査も可能となる．

2.2.3 サーバ機能の復旧処理

図 2.5 に Backup server mode におけるサーバ機能の復旧処理について示す．このモードでは(a)～(d)の処理を行い，管理対象サーバの機能を復旧する．(a)では管理対象サーバと同様のサービスを提供可能な仮想サーバをバックアップサーバとして起動する．その起動完了を確認後，クライアントからのアクセスを故障中の管理対象サーバから仮想サーバに変更するため，(b)に示す仮想 IP アドレスの設定を行う．設定には UNIX の ip コマンドを使用する．この状態では，ルータやスイッチング HUB などの通信機器内における ARP テーブルの仮想 IP アドレスに対応する MAC アドレスが故障した管理対象サーバのアドレスとなっているため，(c)に示す仮想 IP アドレスにおける ARP テーブルの変更通知をブロードキャスト通信で行う．ここで，ARP テーブルとは IP アドレスと MAC アドレスの関係を示す情報で，その変更には UNIX の arping コマンドを使用している．(d)に示すように仮想サーバが File サーバとの NFS 接続により，管理対象サーバにおけるサービス提供プログラムの設定ファイルやユーザデータを取得し，クライアントへのサービス提供が継続可能となる．

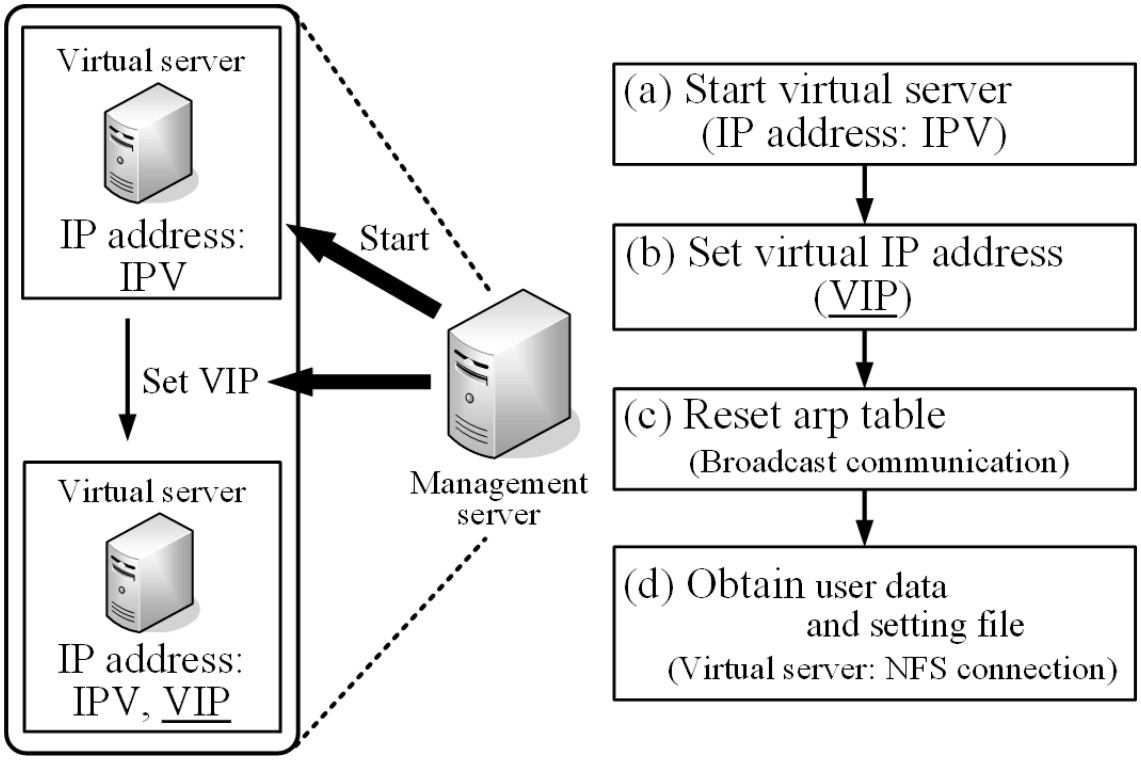


図 2.5　サーバ機能の復旧処理

2.3　様々なトラブルに対する復旧時間およびその予測

2.3.1　MSBS 実験システム

図 2.6 に MSBS 実験システムの構成を示す．実験システムは MSBS の機能を有する管理サーバが 1 台，NFS のサーバ機能を有する File サーバが 1 台，管理対象の Mail サーバと Web サーバが 1 台ずつ，クライアントが 1 台，100BASE-TX のスイッチング HUB が 1 台と 1000BASE-T と 100BASE-TX の機能を有するルータが 1 台から構成される．管理対象の Mail サーバと Web サーバは File サーバと NFS 接続している. Mail サーバと Web サーバの設定ファイルやユーザデータは故障時の対処を考慮し，NFS 接続を通じて File サーバに記録している.

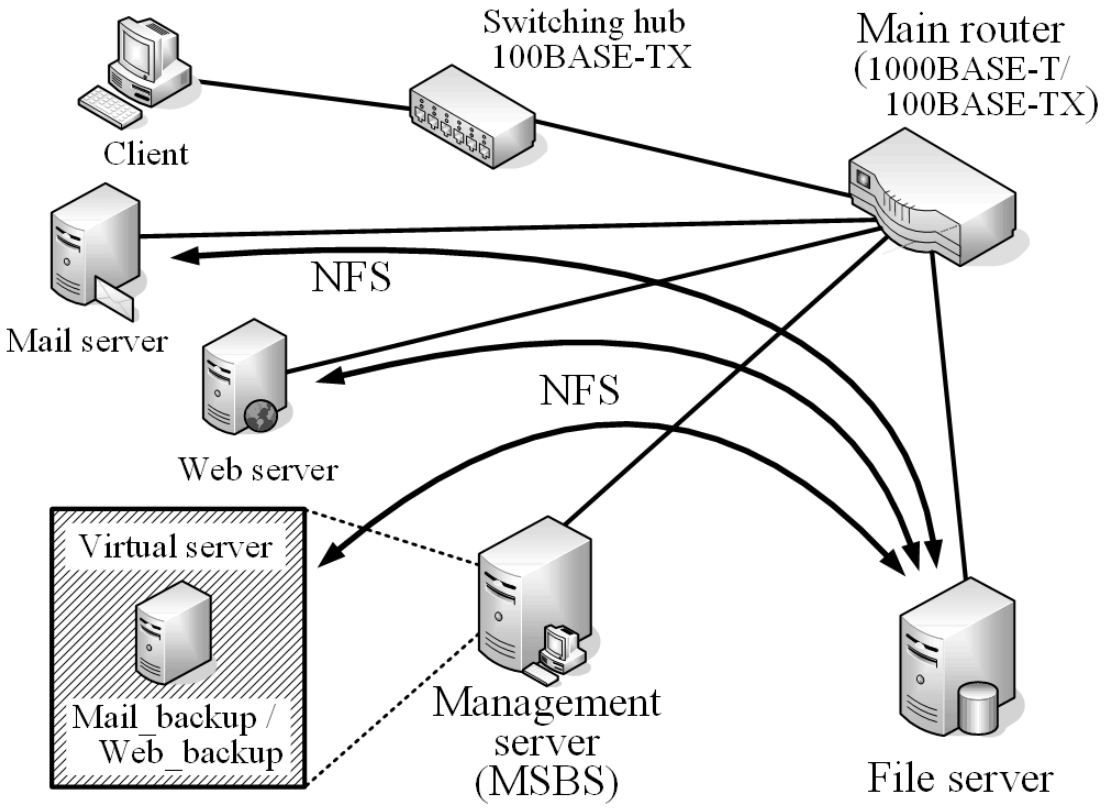


図 2.6　MSBS 実験システムの構成

管理サーバでは，Mail と Web サーバの機能を有する仮想サーバ（Mail_backup，Web_backup）を 2 台起動可能としており，その仮想サーバは管理対象サーバの設定ファイルやユーザデータを使用するため File サーバとの NFS 接続を可能としている．実験システムのネットワーク環境としては，一般的な構成としてサーバ間のネットワーク転送速度に対してクライアントからのネットワーク転送速度を 10 分の 1 としている．管理サーバに設定した MSBS は管理対象サーバである Mail と Web サーバを定期的に監視し，異常を検出した場合に，その管理対象サーバの機能を有する仮想サーバを起動し，そのサーバ機能の復旧を行う．

表 2.1 に MSBS 実験システムにおける各種実サーバおよび仮想サーバの仕様，基礎資料として各サーバの特性およびシステム動作時の設定時間を示す．MSBS をインストールした管理サーバの CPU は Intel Core i7-3770，仮想サーバの起動を考慮して物理メモリは 8,192MB, 仮想化ソフトウエアには世界的に有名な VMware Player と仮想サーバの高速起動およびウインドウを表示しない状態での稼働を可能とするため VIX API を採用する．管理対象サーバである Mail と Web サーバの CPU は Intel Core 2 Duo，物理メモリは 2,048MB，Mail サーバのサービス提供プログラムには postfix を，Web サーバには httpd を採用している．仮想サーバは，CPU 数を 1，メモリを 2,048MB と設定し，管理対象サーバと同様のサービス提供プログラムを稼働可能としている．各サーバの OS には，サーバ用の OS として採用されることが多い CentOS の 64bit 版をインストールする．

基礎資料として各サーバの特性を示す. これらは実測を 10 回行った結果の平均値とする．管理対象サーバの起動時間を Trs とし，その平均時間は 36.3 秒，再起動時間を Trr とし，47.4 秒となった．仮想サーバにおいては，通常の起動時間が 30.4 秒で，サスペンド状態から起動する時間を Tvs とし，2.1 秒となった．この結果より，DBSS ではサーバ機能復旧時における仮想サーバの起動時間を短縮するため，仮想サーバは常にサスペンド状態のまま待機させる．ここで, 仮想サーバをプログラムによってサスペンド状態にするには VIX API のインストールが必要となる．

また，本システムでは管理対象サーバの監視間隔時間を Tim とし，10.0 秒，管理対象サーバのネットワークが正常時におけるネットワーク調査時間を Tha とし，1.0 秒，管理対象サーバのネットワークが異常時におけるネットワーク調査時間を Thd とし，2.0 秒，サービス提供プログラムや管理対象サーバの再起動後における再調査のための待ち時間を Tsw とし，2.0 秒，管理対象サーバが再起動状態か判断する最大時間を Trc とし，Trr の約 1.3 倍の 60.0 秒と設定している．

表 2.1　MSBS 実験システムの仕様，特性および設定時間

Specifications	Management server (MSBS)	Target real server (Mail / Web)
CPU	Intel Core i7-3770 3.40 GHz (TB: 3.90 GHz)	Intel Core 2 Duo E8500 3.16 GHz
Memory	8,192 MB	2,048 MB
Hard disk	SATA 2 (7,200 rpm)	
System software	VMware Player 6.0.6	Mail: postfix
	VIX API 1.13.6	Web: httpd
OS	CentOS 6.7 (64-bit)	

Specifications	Virtual server
CPU	1
Memory	2,048 MB
System software	Mail: postfix / Web: httpd
OS	CentOS 6.7 (64-bit)

Characteristics	Measured time (s)	
Start time of real server	Trs	36.3
Restart time of real server	Trr	47.4
Start time of virtual server		30.4
Time to activate suspended virtual server	Tvs	2.1

Setting parameters		
Monitoring interval time (s)	Tim	10.0
Network examination time (s)	Tna	1.0
Network examination time (Disconnection) (s)	Tnd	2.0
Wait time for service reexamination (s)	Tsw	2.0
Maximum judgment time for restart (s)	Trc	60.0

2.3.2　実験結果とその評価

ここでは，DBSS および MSBS の性能に関して示す．最初に DBSS の結果を示し，次に MSBS の結果を示す．

DBSS の性能に関しては，管理対象サーバを Mail サーバとし，そのサーバに対してサービス提供プログラムおよびネットワークに関するトラブルを再現した場合の DBSS による管理対象サーバの機能およびサーバ自体の復旧時間を示す．監視間隔時間を 10 秒とし，その開始時から 5 秒経過時に Mail サーバに対してサービス提供プログラムやネットワークに関するトラブルを再現する実験を行い，それらの対処に要する復旧時間を実測する．サービスに関するトラブルは，Mail サーバのサービス提供プログラムを停止し，ネットワーク

に関するトラブルは, Mail サーバを再起動またはシャットダウンすることにより再現する．

表 2.2 にサービス提供プログラムおよびネットワークのトラブルに対する DBSS の復旧時間およびその予測時間を示す．表に示すように，トラブルは St1，St2，Nt1 と Nt2 の 4 種類を想定し，Mail サーバにおけるサービス提供プログラムを停止し，それの再起動で復旧可能な場合を St1，その再起動では復旧が困難であり，Mail サーバを再起動することで復旧可能な場合を St2，Mail サーバを意図的に再起動した場合を Nt1，Mail サーバをシャットダウンした場合を Nt2 とする．ここで，Nt1 はサーバの稼働状態が異常な場合にサーバ管理者がリセットボタンを押すなど，そのサーバを再起動することを想定している．実測時間に関して，サーバ機能の復旧時間は仮想サーバによりサーバ機能を復旧するまでの時間，管理対象サーバの復旧時間は Mail サーバが正常状態に復旧するまでの時間，クライアントにおけるアクセス切断時間はクライアントから 1 秒間隔でサーバに対してアクセスを行い，それが切断されてから復旧するまでの時間を示す．サーバ機能の復旧時間および管理対象サーバの復旧時間は付録の F4 で示す本システムが記録している Log ファイルからの測定方法を採用している．復旧時間において括弧で囲まれた数値は予測値を示し，表 2.1 で示した実測時間や設定時間から算出する．また，括弧で囲まれていない数値は実測値を示す．それぞれの実測時間は実験を 10 回行った結果の平均値とする．

サーバ機能の復旧時間において，St2 のトラブルに対する復旧時間は Ts2f，Nt1 のトラブルに対する復旧時間は Th1f，Nt2 のトラブルに対する復旧時間は Th2f とする．St2 のトラブル時におけるモードの流れは Regular mode → Service recovery mode → Backup server mode → Target server recovery mode → Regular mode となり，Ts2f はサービスの異常を検出後，Service recovery mode において管理対象サーバのネットワーク確認時間 Tha，サービス提供プログラム再起動後のサービス再調査を行うまでの待ち時間 Tsw および Backup server mode における仮想サーバの起動時間 Tvs が主な構成要素となり，予測

値は式(2.1)で示される．

$$T\text{s2f}\ [\text{s}] = T\text{na} + T\text{sw} + T\text{vs} \tag{2.1}$$

Ts2f の実測値は 5.5 秒となり予測値は 5.1 秒となった．Nt1 と Nt2 のトラブル時におけるモードの流れは Regular mode → Network examination mode → Backup server mode → Target server recovery mode → Regular mode となり，Th1f と Th2f はネットワーク異常を検出後，Network examination mode においてネットワーク接続調査時間 Thd および Backup server mode における仮想サーバの起動時間 Tvs が主な構成要素となり，予測値は式(2.2)で示される．

$$T\text{n1f}\ [\text{s}] = T\text{n2f} = T\text{nd} + T\text{vs} \tag{2.2}$$

Th1f の実測値は 4.7 秒，Th2f の実測値は 4.8 秒となり，ともに予測値は 4.1 秒となった．サーバ機能の復旧時間においては，実測時間と予測時間はほぼ同等となっており，いずれのトラブルにおいても仮想サーバの起動時間が大きく影響しており約 5 秒となった．

管理対象サーバの復旧時間において，St1 のトラブルに対する復旧時間は Ts1t，St2 のトラブルに対する復旧時間は Ts2t，Nt1 のトラブルに対する復旧時間は Th1t，Nt2 のトラブルに対する復旧時間は Th2t とする．St1 のトラブル時におけるモードの流れは Regular mode → Service recovery mode → Regular mode となり，Ts1t はサービスの異常を検出後，Service recovery mode において管理対象サーバのネットワーク確認時間 Tha とサービス提供プログラム再起動後にサービス再調査を行うまでの待ち時間 Tsw が主な構成要素となり，予測値は式(2.3)で示される．

$$T\text{s1t}\ [\text{s}] = T\text{na} + T\text{sw} \tag{2.3}$$

Ts1t の実測値は 3.4 秒，予測値は 3.0 秒となった．St2 のトラブルにおいては Mail サーバを再起動することにより，そのサーバのサービス提供状態を復旧することから再起動時間の Trr が含まれ，予測値は式(2.4)で示される．

$$T\mathrm{s2t}\ [\mathrm{s}] = T\mathrm{na} + T\mathrm{sw} + T\mathrm{rr} + T\mathrm{sw} \tag{2.4}$$

Ts2t の実測値は 54.4 秒となり予測値は 52.4 秒となった．Nt1 のトラブルにおいては監視間隔時間が 5 秒経過時に Mail サーバを再起動させることから，ネットワークトラブルを検出するまでに Tim/2＋Thd の時間を要しているため，予測値は式(2.5)で示される．

$$T\mathrm{n1t}\ [\mathrm{s}] = T\mathrm{rr} + T\mathrm{sw} - (\frac{T\mathrm{im}}{2} + T\mathrm{nd}) \tag{2.5}$$

Th1t の実測値は 44.5 秒となり予測値は 42.4 秒となった．Nt2 のトラブルにおいては Mail サーバが再起動状態か判断する最大時間 Trc とそのサーバの起動時間 Trs が含まれ，予測値は式(2.6)で示される．

$$T\mathrm{n2t}\ [\mathrm{s}] = T\mathrm{nd} + T\mathrm{rc} + T\mathrm{rs} + T\mathrm{sw} \tag{2.6}$$

Th2t の実測値は 102.6 秒となり予測値は 100.3 秒となった．ここで，Ts2t，Th1t と Th2t は実測値と予測値に差が見られるが，この差は予測値の構成要素に含まれていない 1 秒未満で終了する要素の合計が影響していると考えられる．St2 ではサーバを再起動し，そのサービス機能を復旧することからサーバ再起動時間 Trr が，Nt2 では Mail サーバが再起動状態ではないことを確認後，そのサーバを起動することから，再起動状態か判断する最大時間 Trc とサーバの起動時間 Trs の影響が大きくなっていることがわかる．

クライアントからのアクセス切断時間において，St1 のトラブルに対する切断時間は Ts1c，St2 のトラブルに対する切断時間は Ts2c，Nt1 のトラブルに対する切断時間は Th1c，Nt2 のトラブルに対する切断時間は Th2c とする．監視間隔時間内に各種トラブルを再現することから，それぞれの切断時間には監視間隔時間（Tim/2±Tim/2）を含み，予測値は以下の式で示される．

$$T\mathrm{s1c}\ [\mathrm{s}] = T\mathrm{na} + T\mathrm{s1t} + \frac{T\mathrm{im}}{2} \pm \frac{T\mathrm{im}}{2} \tag{2.7}$$

$$T\mathrm{s2c}\ [\mathrm{s}] = T\mathrm{na} + T\mathrm{s2f} + \frac{T\mathrm{im}}{2} \pm \frac{T\mathrm{im}}{2} \tag{2.8}$$

$$T\mathrm{n1c}\ [\mathrm{s}] = T\mathrm{n2c} = T\mathrm{nd} + T\mathrm{n1f} + \frac{T\mathrm{im}}{2} \pm \frac{T\mathrm{im}}{2} \tag{2.9}$$

T_{s1c} の実測値は 9.4 秒となり予測値は 9.0±5.0 秒，T_{s2c} の実測値は 11.5 秒となり予測値は 11.1±5.0 秒，T_{h1c} の実測値は 11.6 秒となり予測値は 11.1±5.0 秒，T_{h2c} の実測値は 11.7 秒となり予測値は 11.1±5.0 秒となった. いずれのトラブルにおけるクライアントからのアクセス切断時間の実測値は予測時間の範囲に含まれていることがわかる.

表 2.2　様々なトラブルに対する DBSS の復旧時間の実測値および予測値

Target server: Mail　Value : Measured time (s) (Value): Predicted time (s)							
Trouble list		**Recovery time of server function**		**Recovery time for target server**		**Access disconnection time (Client)**	
Service program	**Recovery by service restart (St1)**			T_{s1t}	3.4 (3.0)	T_{s1c}	9.4 (9.0±5.0)
	Recovery by server restart (St2)	T_{s2f}	5.5 (5.1)	T_{s2t}	54.4 (52.4)	T_{s2c}	11.5 (11.1±5.0)
Network	**Intentional restart of target server (Nt1)**	T_{n1f}	4.7 (4.1)	T_{n1t}	44.5 (42.4)	T_{n1c}	11.6 (11.1±5.0)
	Shutdown of target server (Nt2)	T_{n2f}	4.8 (4.1)	T_{n2t}	102.6 (100.3)	T_{n2c}	11.7 (11.1±5.0)

ここで，図 1.2 で示した稼働率による評価を表 2.3 に示す．図 1.1 で示した TPUCELS において，コストの関係上，冗長化を行っていないサーバでトラブルが発生した場合，その故障サーバ自体の復旧が求められる．そこで，管理対象サーバ自体を復旧する方式を従来方式として比較を行う. 従来方式としての t_iには表 2.2 に示す管理対象サーバの復旧時間を，改善システムとしての t_{bi}にはサーバ機能の復旧時間を使用する．T_iには管理対象サー

バの復旧時間における最大値を使用し，その値での従来方式の稼働率（Conventional sysytem）を 0.5 とする．また，提案方式による稼働率の改善（Improvement）を式(2.10)で算出する．

{(提案方式の稼働率－従来方式の稼働率) / 従来方式の稼働率 }×100　　(2.10)

St2 のトラブルの場合, 従来方式の稼働率は 0.65, 提案方式の稼働率（Proposed system）は 0.96 となり 47.69%の改善がみられる．また，Nt1 のトラブルの場合，従来方式の稼働率は 0.70，提案方式の稼働率は 0.97 となり 38.57%の改善がみられる．さらに，Nt2 のトラブルの場合，従来方式の稼働率は 0.50，提案方式の稼働率は 0.98 となり 96.00%の改善がみられている．以上の結果から，提案方式は従来方式に比べ稼働率の面において非常に高い優位性が認められる．

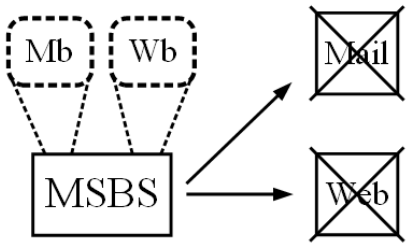
表 2.3　提案方式による稼働率の改善

St2: Service program trouble (Recovery by server restart)
Nt1: Intentional restart of target server
Nt2: Shutdown of target server

		Conventional system	Proposed system	Improvement (%)
Availability rate	St2	0.65	0.96	47.69
	Nt1	0.70	0.97	38.57
	Nt2	0.50	0.98	96.00
	Average	0.62	0.97	56.45

次に MSBS の性能に関しては, 管理対象サーバを Mail サーバと Web サーバの 2 台とし，それらのサーバに対してサービス提供プログラムおよびネットワークに関するトラブルを

2 台同時に再現した場合の MSBS による管理対象サーバの機能およびサーバ自体の復旧時間を示す．MSBS において Mail サーバ用および Web サーバ用に DBSS を動作させ，それぞれの DBSS は独立して管理対象サーバの管理を行う. DBSS では監視間隔時間を 10 秒とし，管理対象サーバに対して監視間隔時間が 5 秒経過時にサービス提供プログラムやネットワークに関するトラブルを再現する実験を行い，それらの対処に要する復旧時間を実測する. 図 2.7 に複数の管理対象サーバが故障した場合の MSBS による復旧時間を示す. St2, Nt1 と Nt2 のトラブルは前述した内容と同様で，ここでは 2 台の管理対象サーバに対してそれぞれのトラブルを組み合わせ，9 通りの実験を行う．括弧で囲まれていない数値はサーバ機能の復旧時間を示し，括弧で囲まれている数値は管理対象サーバの復旧時間を示す.



St2 : Service program trouble (Recovery by server restart)
Nt1 : Intentional restart of target server
Nt2 : Shutdown of target server

Value	: Recovery time of server function
(Value)	: Recovery time for target server

Measured time (s)		Web server trouble		
		St2	Nt1	Nt2
Mail server trouble	St2	5.8 (54.5) / 5.8 (54.4)	4.9 (44.7) / 5.7 (54.6)	5.0 (103.3) / 5.8 (54.7)
	Nt1	5.9 (54.7) / 4.8 (44.8)	4.7 (44.5) / 4.7 (44.6)	4.9 (104.2) / 4.7 (44.5)
	Nt2	5.7 (54.4) / 4.8 (102.7)	4.8 (44.7) / 4.9 (103.1)	5.2 (104.9) / 5.0 (105.3)

図 2.7　様々なトラブルに対する MSBS の復旧時間

Mail サーバと Web サーバに St2 のトラブルを同時に再現した場合, サーバ機能の復旧時間はともに 5.8 秒となり，管理対象サーバの復旧時間では Mail サーバが 54.4 秒，Web サーバが 54.5 秒となった．表 2.2 の結果とほぼ同等の結果となったが，サーバ機能の復旧時間に関しては多少の遅れが見られる．Mail サーバと Web サーバに Nt1 のトラブルを同時に再現した場合，サーバ機能の復旧時間はともに 4.7 秒となり，管理対象サーバの復旧時間では Mail サーバが 44.6 秒，Web サーバが 44.5 秒となった．表 2.2 の結果とほぼ同等の結果となった．Mail サーバと Web サーバに Nt2 のトラブルを同時に再現した場合，サーバ機能の復旧時間は Mail サーバが 5.0 秒，Web サーバが 5.2 秒となり，管理対象サーバの復旧時間では Mail サーバが 105.3 秒，Web サーバが 104.9 秒となった．表 2.2 の結果より，サーバ機能の復旧時間および管理対象サーバの復旧時間ともに多少の遅れが見られる．全体的にそれぞれの復旧時間は表 2.2 に示した結果とほぼ同等の値となっている．これは，それぞれの DBSS が独立で管理対象サーバを管理していることから，サーバ機能の復旧作業において，それぞれの DBSS が同時に仮想サーバの起動処理を行うことが殆どなかったため，動作の遅れが生じなかったものと考えられる．

表 2.4 に冗長化構成を構築する上で管理対象サーバの台数に応じた予備用実サーバの台数を示す．付録の F2 に示すようにサーバの MTBF を 100,000 時間として計算を行うと，管理対象サーバ数が 40 台の場合，1 年間で約 4 台が故障する．ここで，実験システムの管理サーバは，最大で同時に 4 台の仮想サーバを起動できるリソースとなっている．そのため，管理対象サーバは 40 台までに設計することが望ましい．提案方式を“MSBS”とし，従来方式としては図 1.1 で示した TPUCELS において，コストの関係上，冗長化を行っていないサーバで構成されるサーバシステムを“CS”として示す．CS は管理対象サーバの台数に応じて予備用実サーバの台数が比例して増加する．MSBS は予備のサーバとして仮想

サーバを使用するため，管理対象サーバの台数に応じて予備用実サーバの台数は増加しない．ただし，MSBS を導入している管理サーバが故障した場合，管理対象サーバの管理ができなくなるため，その管理サーバは冗長化する必要がある．そのため，MSBS は管理対象サーバの台数に関係なく 2 台の状態を維持している．ここで，予備用実サーバの台数に比例して初期費用は高くなり，関連して消費電力や保守料金などの運用コストも台数に比例して高くなる．例えば，MSBS を導入する管理サーバを 2 台で 100 万円，予備用実サーバを 1 台 30 万円とし，管理対象サーバが 40 台の場合には，MSBS は初期費用だけで 1,100 万円のコストを抑えることができ，さらに，台数に比例した運用コストも抑えることができる．以上の結果から，MSBS は従来方式に比べコスト面においても優位性が認められる．

表 2.4　予備用実サーバの台数比較

CS: Conventional system

	The number of target servers ($n \leqq 40$)					
	1	**2**	**3**	**・・・**	***n*-1**	***n***
CS	1	2	3	・・・	n-1	n
MSBS	2	2	2	・・・	2	2

2.4　仮想サーバの性能調査およびその評価

一般的にサーバはユーザからのアクセスに対してデータの提供，受け入れや CPU 処理を行う．そこで，仮想サーバの性能調査として，クライアントからのダウンロードアクセス，アップロードアクセスと CPU 処理要求に対する応答時間を調査する．

2.4.1 仮想サーバ性能調査システムの構成およびその仕様

MSBS は複数の DBSS によって構成される．そのため，DBSS のシステム性能を把握することは非常に重要である．DBSS では管理対象サーバの機能を復旧するために仮想サーバを使用することから，図 2.8 に示す仮想サーバ性能調査システムを構築し，仮想サーバの性能や特性を調査する．このシステムは，論理的クライアントが 2 台，実サーバが 3 台，1000BASE-T のスイッチング HUB が 1 台と 100BASE-TX のスイッチング HUB が 1 台から構成される．それぞれの実サーバでは 2 種類のサーバ仮想化ソフトを採用している．2 台の論理的クライアントは調査システムで接続を変更することによって区別され，実際のクライアントは 1 台である．Client A 接続はクライアントとサーバを 1000BASE-T のスイッチング HUB で直接接続し，Client B 接続は 100BASE-TX のスイッチング HUB 経由でサーバに接続することを意味する．本調査システムでは，仮想サーバの特性を調査することが目的となるため，2 種類のサーバ仮想化ソフトを採用し，それらの動作傾向を調査する．そのため，2 種類のサーバ仮想化ソフトの性能に関し，実サーバのハードウエアによる影響を少なくするためにマルチ OS ブートシステムを採用している．それぞれの OS には Host OS 型または Hypervisor 型のサーバ仮想化ソフトがインストールされ，それぞれの実サーバから仮想サーバを起動可能とする．

仮想サーバは内部時刻にずれが生じる問題を持っている[60]．そのため，本章ではクライアントからネットワークアクセスを行い，その応答時間による評価方法を採用し，ここではその方式をネットワークアクセス方式と呼ぶ．ネットワークアクセスに関して，ダウンロードはクライアントがサーバからデータを取得することを意味し，アップロードはクライアントからサーバにデータを転送することを意味する．ダウンロード時間，アップロード時間と CPU 処理時間の実測には UNIX の time コマンドを使用する．また，ダウンロード時間を測定する実験においてディスクキャッシュ対策のため，サーバ上のダウンロード

対象データは，同サイズデータを複数作成している．アップロード時間の測定では，クライアントからのアップロードデータをメモリディスク上に作成することでクライアントでの処理時間を最小にしている．

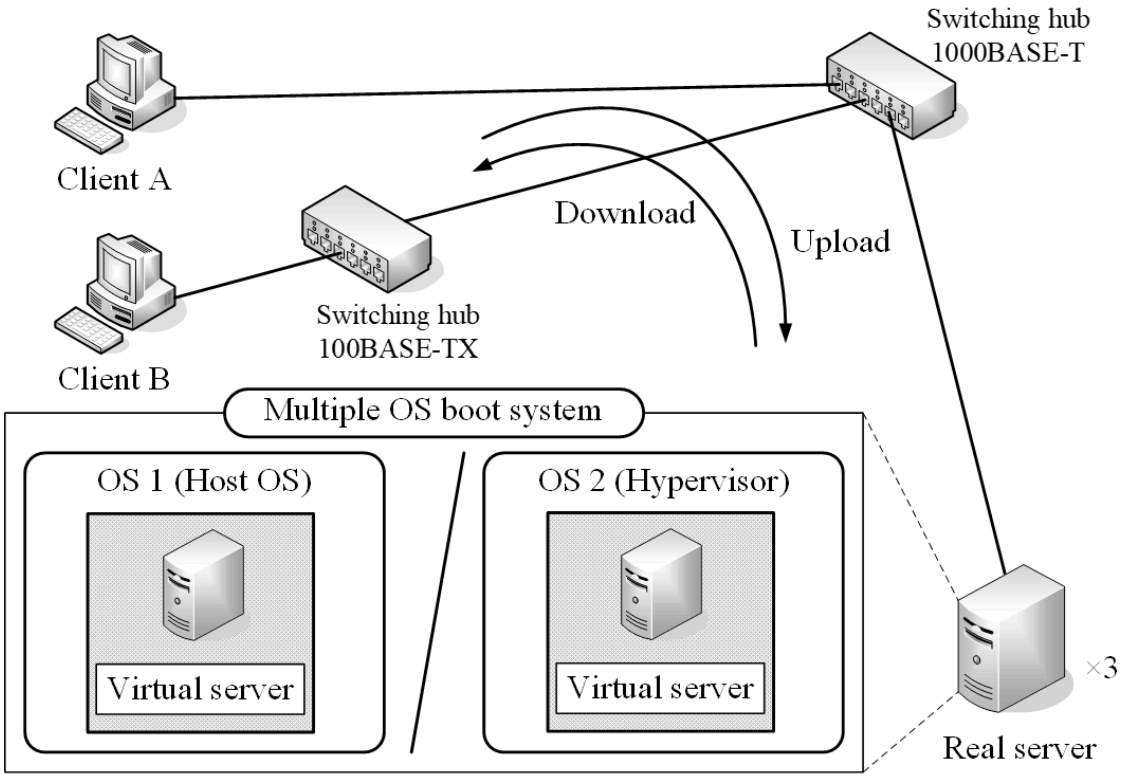


図 2.8　仮想サーバ性能調査システムの構成

表 2.5 に仮想サーバ性能調査システムにおいて使用する実サーバと仮想サーバの仕様を示す．仕様 A，B と C の 3 タイプは異なる CPU を搭載している．各実サーバにはサーバ用の OS として採用されることが多い CentOS の 64bit 版をインストールし，物理メモリは 8,192MB としている．また，実サーバはマルチ OS ブートシステムを採用しており，それぞれの OS における仮想化ソフトウエアとして，１つは世界的に有名である VMware Player と VIX API，もう１つは Linux で標準サポートしている Kernel-based Virtual

Machine（KVM）を採用する．以降，VMware Player を Host OS 型，KVM を Hypervisor 型とする．各実サーバは CUI 環境で動作させるため，仮想サーバはウインドウを出力しない状態で起動する．サービス提供プログラムとして，FTP アクセスを対処するシステムソフトウエアである vsftpd，Web アクセスを対処する httpd と Windows ネットワークにおけるファイル共有サービスのプロトコルである Server Message Block（SMB）を使用可能としている．仮想サーバに設定する CPU 数は実サーバの各仕様に応じて設定値が異なり，仕様 A では 1～8, 仕様 B では 1～4 となり仕様 C では 1～2 とする. これは実サーバの CPU コア数およびハイパースレッディング機能により決定される. メモリは 1,024MB を設定し，サービス提供プログラムや OS に関しては実サーバと同様としている．

表 2.5　仮想サーバ性能調査システムにおける調査対象サーバの仕様

Specifications	Specification A	Specification B	Specification C
CPU	Intel Core i7-2600 3.40 GHz (TB: 3.80 GHz)	Intel Core 2 Quad Q9650 3.00 GHz	Intel Core 2 Duo E8500 3.16 GHz
Memory	8,192 MB		
Hard disk	SATA 2 (7,200 rpm)		
System software	vsftpd, httpd, smb		
	Host OS : VMware Player 6.0.6 / VIX-API 1.13.6 ／ Hypervisor : KVM 0.12.1		
OS	CentOS 6.7 (64-bit)		

Specifications	Virtual server (Host OS / Hypervisor)
CPU	Specification A: 1~8 / Specification B: 1~4 / Specification C: 1~2
Memory	1,024 MB
System software	vsftpd, httpd, smb
OS	CentOS 6.7 (64-bit)

2.4.2　ハードウエア仕様の違いにおける仮想サーバの性能調査

ここでは，仮想サーバを起動する実サーバの仕様による性能の差を調査するため，クライアントとサーバを 1000BASE-T のスイッチング HUB で直接接続する Client A 接続で調査を行う．また，性能の比較には FTP による連続または同時タスクアクセスを使用する．ここでのタスクとは 10^8B データを 1 つダウンロードまたはアップロードを行うことを意味する．ここでの調査に使用する仮想サーバの CPU 数は 1 と設定している．

仕様 A, B と C の実サーバから起動した仮想サーバに対して 50 タスク連続ダウンロードアクセスを行った結果を図 2.9 に示す．図において，(a)に Host OS 型，(b)に Hypervisor 型の仮想サーバにおけるダウンロード性能の比較を示す．(a)と(b)の縦軸はダウンロード時間とし，横軸はタスクの番号とする．

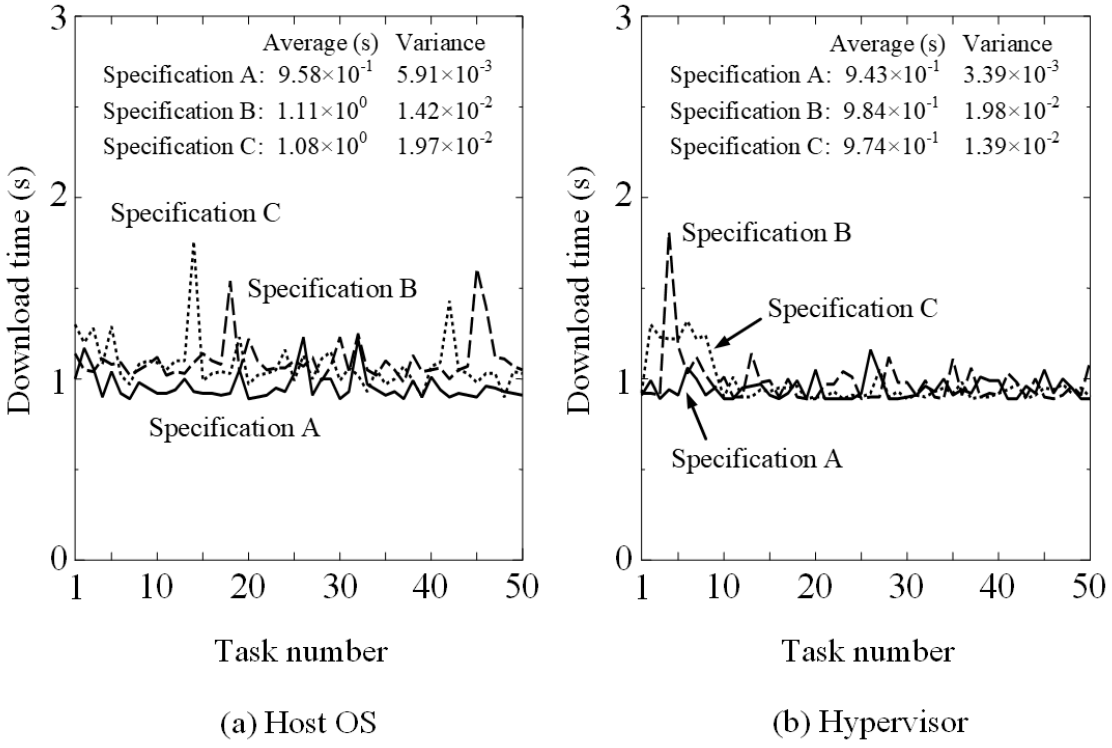


図 2.9　異仕様の実サーバから起動した仮想サーバの連続ダウンロード性能

(a)では多少の変動が見られるが, 仕様 A のダウンロード時間が仕様 B と C に比べ一番短く,仕様 A の平均値は 9.58×10^{-1} 秒となった．分散に関しても仕様 A が仕様 B と C に比べ一番低くなっている．仕様 B の平均値は 1.11×10^{0} 秒となり仕様 C は 1.08×10^{0} 秒となった．(b)においても，多少の変動が見られるが，仕様 A のダウンロード時間が仕様 B と C に比べ一番短く，仕様 A の平均値は 9.43×10^{-1} 秒となった．分散に関しても仕様 A が仕様 B と C に比べ一番低くなっている. 仕様 B の平均値は 9.84×10^{-1} 秒となり仕様 C は 9.74×10^{-1} 秒となった．以上から，Host OS および Hypervisor 型の 2 種類とも同様の傾向を示しており，仕様 A のダウンロード時間が仕様 B と C に比べ一番短くなっていることがわかる．

仕様 A, B と C の実サーバから起動した仮想サーバに対して 50 タスク連続アップロードアクセスを行った結果を図 2.10 に示す．図において，(a)に Host OS 型，(b)に Hypervisor 型の仮想サーバにおけるアップロード性能の比較を示す．(a)と(b)の縦軸はアップロード時間とし，横軸はタスクの番号とする．

(a)では多少の変動が見られるが, 仕様 A のアップロード時間が仕様 B と C に比べ一番短くなっており，仕様 A の平均値は 1.04×10^{0} 秒となった．分散に関しても仕様 A が仕様 B と C に比べ一番低くなっている. 仕様 B の平均値は 1.41×10^{0} 秒となり仕様 C は 1.38×10^{0} 秒となった．(b)は(a)に比べると変動が大きいことがわかる．仕様 A のアップロード時間が仕様 B と C に比べ一番短くなっており，仕様 A の平均値は 1.09×10^{0} 秒となった．分散に関しても仕様 A が仕様 B と C に比べ一番低くなっている. 仕様 B と仕様 C の平均値は 1.15×10^{0} 秒となった．以上から，Host OS および Hypervisor 型の 2 種類とも同様の傾向を示しており，仕様 A のアップロード時間が仕様 B と C に比べ一番短くなっていることがわかる．

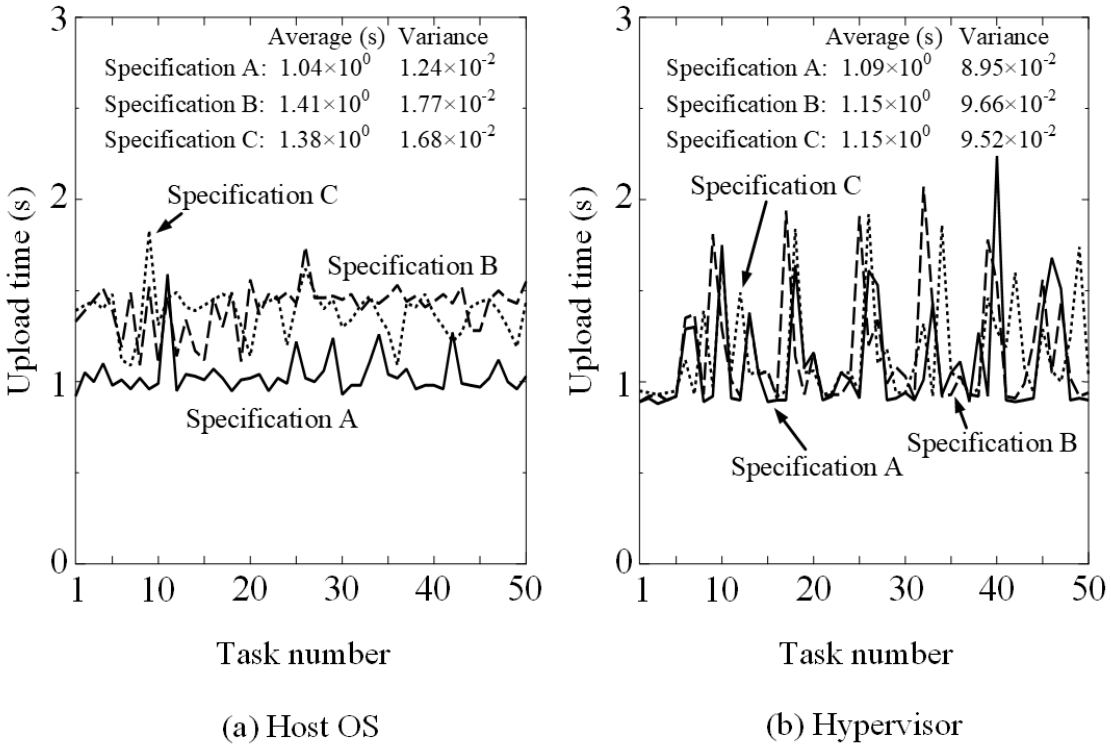


図 2.10　異仕様の実サーバから起動した仮想サーバの連続アップロード性能

仕様 A, B と C の実サーバから起動した仮想サーバに対して 50 タスク同時ダウンロードアクセスを行った結果を図 2.11 に示す．図において，(a)に Host OS 型，(b)に Hypervisor 型の仮想サーバにおけるダウンロード性能の比較を示す．(a)と(b)の縦軸は同時に実行されているタスク数とし，横軸はダウンロード時間とする．

(a)と(b)ともに，1 つ目のタスクが終了してからすべてのタスクが終了するまでの時間にあまり差がない．これはサーバが Time Sharing System（TSS）で処理していることから，このような結果になったと考えられる．また，(a)と(b)を比較すると，すべての仕様において Host OS 型のダウンロード時間が短くなっているが，(a)と(b)ともに仕様 A のダウンロード時間が一番短く，次に仕様 B，仕様 C との結果となり，同様の傾向を示している．この仕様による差に関しては CPU のコア数が影響していると考えられる．Host OS 型におけ

る仕様 A のダウンロード終了時間は 1.42×10² 秒，仕様 B のダウンロード終了時間は 1.66×10² 秒となり仕様 C のダウンロード終了時間は 2.04×10² 秒となった．また，Hypervisor 型における仕様 A のダウンロード終了時間は 1.57×10² 秒，仕様 B のダウンロード終了時間は 2.01×10² 秒となり仕様 C のダウンロード終了時間は 3.36×10² 秒となった．

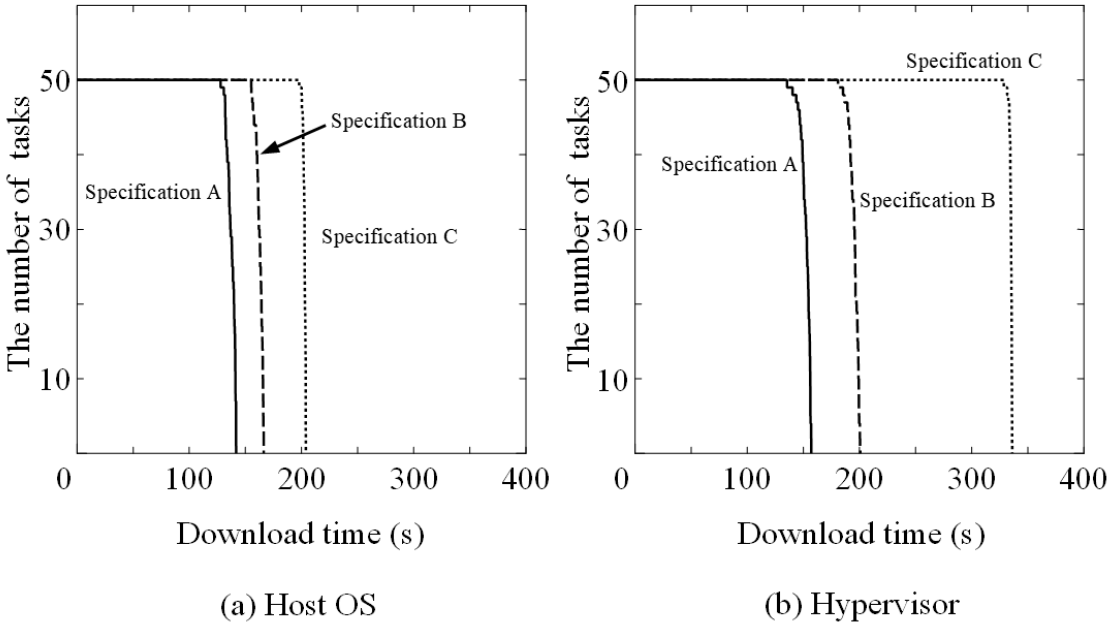


図 2.11　異仕様の実サーバから起動した仮想サーバの同時ダウンロード性能

仕様 A, B と C の実サーバから起動した仮想サーバに対して 50 タスク同時アップロードアクセスを行った結果を図 2.12 に示す．図において，(a)に Host OS 型，(b)に Hypervisor 型の仮想サーバにおけるアップロード性能の比較を示す．(a)と(b)の縦軸は同時に実行されているタスク数とし，横軸はアップロード時間とする．

図 2.11 に示した同時ダウンロードと同様に，(a)と(b)ともに 1 つ目のタスクが終了してからすべてのタスクが終了するまでの時間にあまり差がない．これはサーバが TSS で処理していることから，このような結果になったと考えられる．また，(a)と(b)はほぼ同様の傾

向を示しており，仕様 A のダウンロード時間が一番短く，次に仕様 C，仕様 B との結果となった．この仕様による差に関しては CPU のコア数ではなく，クロック周波数が影響していると考えられる．Host OS 型における仕様 A のアップロード終了時間は 4.53×10^{1} 秒，仕様 B のアップロード終了時間は 5.47×10^{1} 秒となり仕様 C のアップロード終了時間は 4.96×10^{1} 秒となった. Hypervisor 型における仕様 A のアップロード終了時間は 4.37×10^{1} 秒，仕様 B のアップロード終了時間は 4.62×10^{1} 秒となり仕様 C のアップロード終了時間は 4.52×10^{1} 秒となった.

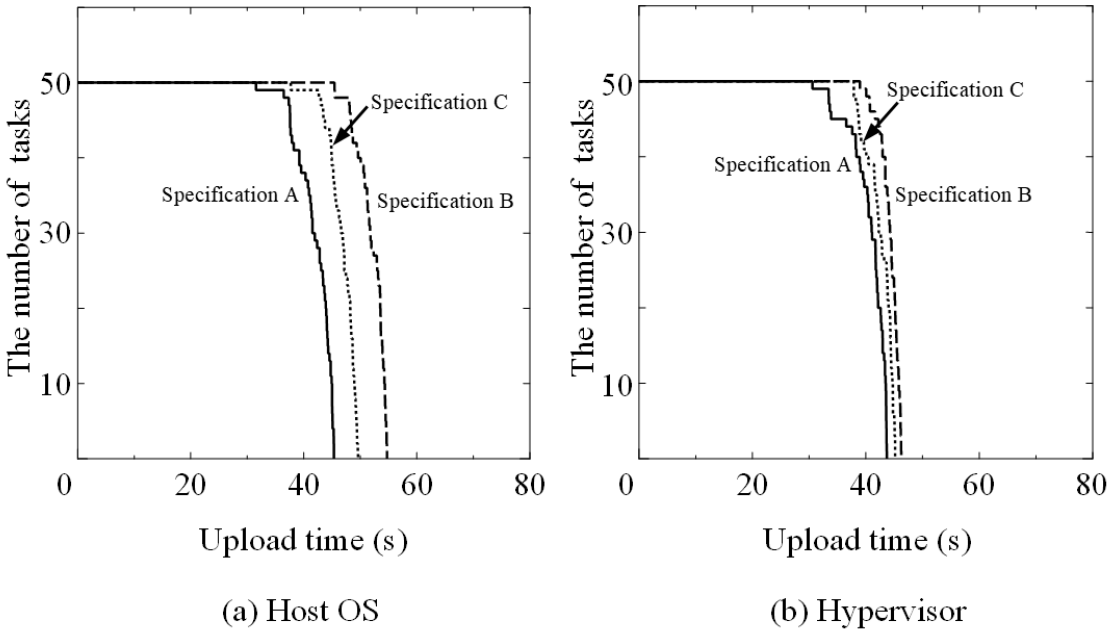


図 2.12　異仕様の実サーバから起動した仮想サーバの同時アップロード性能

図 2.11 の同時ダウンロードの終了時間と比較すると，同時アップロードの結果が非常に短くなっていることがわかる．これは，アップロードされたデータを受け付けたサーバは直ちにハードディスクへ記録せず，メモリに記録した時点でタスクが終了したとクライア

ントに対して返答するためと考えられ，10^8B データが 50 個で約 5GB となるが，実サーバの物理メモリである 8GB の範囲に収まる．

2.4.3 CPU 処理時間における性能調査

ここでは，仮想サーバを起動する実サーバに仕様 A を使用し，クライアントとサーバを 1000BASE-T のスイッチング HUB で直接接続する Client A 接続で調査を行う．上述したが，仮想サーバは内部時刻がずれる問題を持っているため，ここでもネットワークアクセス方式を使用する．本調査ではクライアントからサーバに対して CPU 処理を行うタスクを多重に実行させ，その処理時間をクライアントで実測する．CPU 処理を実行させる方法としては 10^8B のアスキー文字をランダムに発生させることにより実現し，これをタスクとする．また，本実験において仮想サーバと比較する実サーバは，仮想サーバを起動していない状態の仕様 A を使用する．図 2.13 に仮想サーバの CPU 設定値を可変した場合の CPU 処理時間の変化を示す．

(a)に Host OS 型，(b)に Hypervisor 型における仮想サーバの結果を示し，それぞれ実サーバとの比較を行う．(a)と(b)ともに実線は CPU 数を 8 と設定した仮想サーバで“VS (CPU:8)”，破線は CPU 数を 2 と設定した仮想サーバで“VS (CPU:2)”，一点鎖線は CPU 数を 1 と設定した仮想サーバで“VS (CPU:1)”と示し，実サーバは点線で示す．(a)と(b)の縦軸は CPU 処理時間とし，横軸は多重にアクセスするタスクの数とする．(a)と(b)ともに CPU 数を 8 と設定した仮想サーバと実サーバは，ほぼ同等の CPU 処理能力であることを示している．また，VS (CPU:8)と VS (CPU:2)の結果を比較すると VS (CPU:2)は約 2 倍の CPU 処理時間を要していることがわかる．ここで，仕様 A の実サーバは CPU のコア数が 4 コアでハイパースレッディングの機能を有することから，CentOS では 8 個の CPU が動作していると認識しているが，実際のコア数は 4 個のためこのような結果を示したと考

えられる．VS (CPU:8)と VS (CPU:1)の結果を比較すると VS (CPU:1)は約 4 倍の CPU 処理時間を要していることがわかる．多重アクセスするタスク数の変化に対し，Host OS および Hypervisor 型の 2 種類とも同様の傾向を示している．

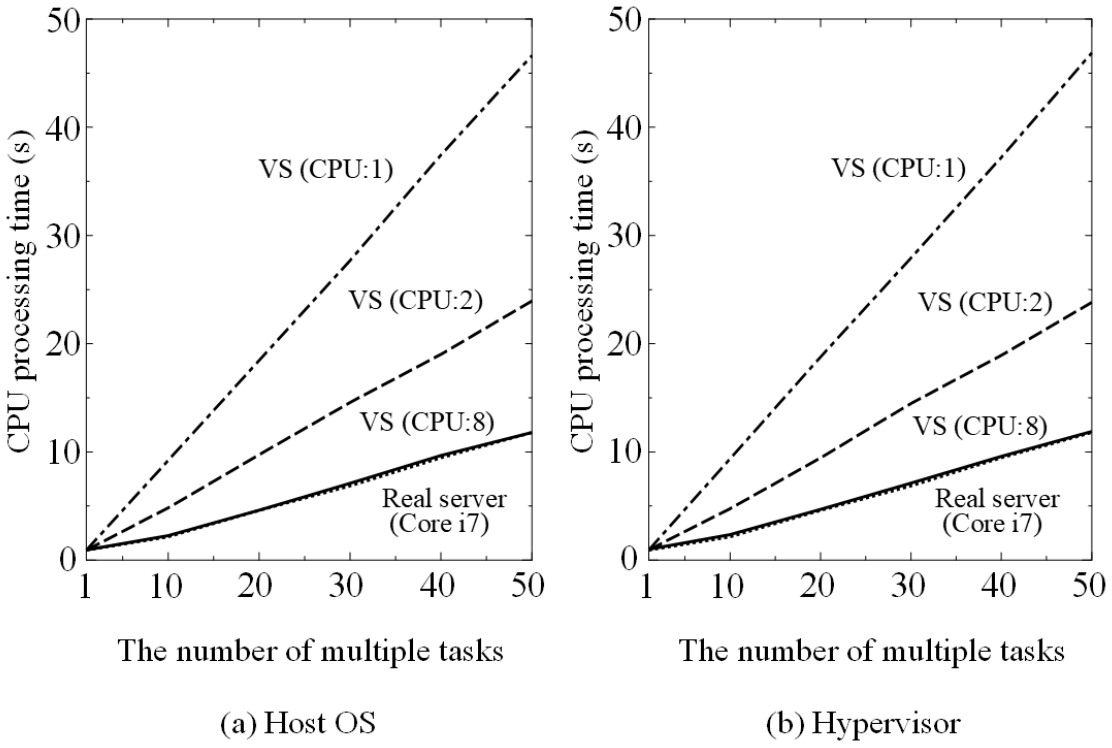


図 2.13　CPU 設定値の違いによる CPU 処理時間の比較

2.4.4　複数のプロトコルによる連続タスクアクセス調査

ここでは，実サーバと Host OS および Hypervisor 型の仮想サーバにおけるダウンロード時間およびアップロード時間の比較を行う．調査対象の仮想サーバは仕様 A の実サーバから起動する．また，本実験において仮想サーバと比較する実サーバは仮想サーバを起動していない状態の仕様 A を使用する．一般的なサーバシステムの構成としてサーバ間のネットワーク転送速度に対してクライアントからのネットワーク転送速度を 10 分の 1 程度と

しているため，100BASE-TX のスイッチング HUB 経由でサーバに接続する Client B 接続で実験を行う．ネットワークアクセス調査を行うプロトコルとして，ダウンロードでは，FTP，SMB と HTTP を使用し，アップロードでは FTP と SMB を使用する．同ファイルサイズのデータを 50 回連続ダウンロードアクセスおよび 50 回連続アップロードアクセスを行い，それらの平均値で評価を行う．

図 2.14 に複数のプロトコルによるダウンロード性能調査の結果を示す．図において，(a)に FTP，(b)に SMB，(c)に HTTP のダウンロード結果を示す．Hypervisor 型の仮想サーバを“VS (Hypervisor)”，Host OS 型の仮想サーバを“VS (Host OS)”と示し，実サーバを“Real server”と示す．(a)，(b)と(c)の縦軸はダウンロード時間とし，横軸はファイルサイズとする．

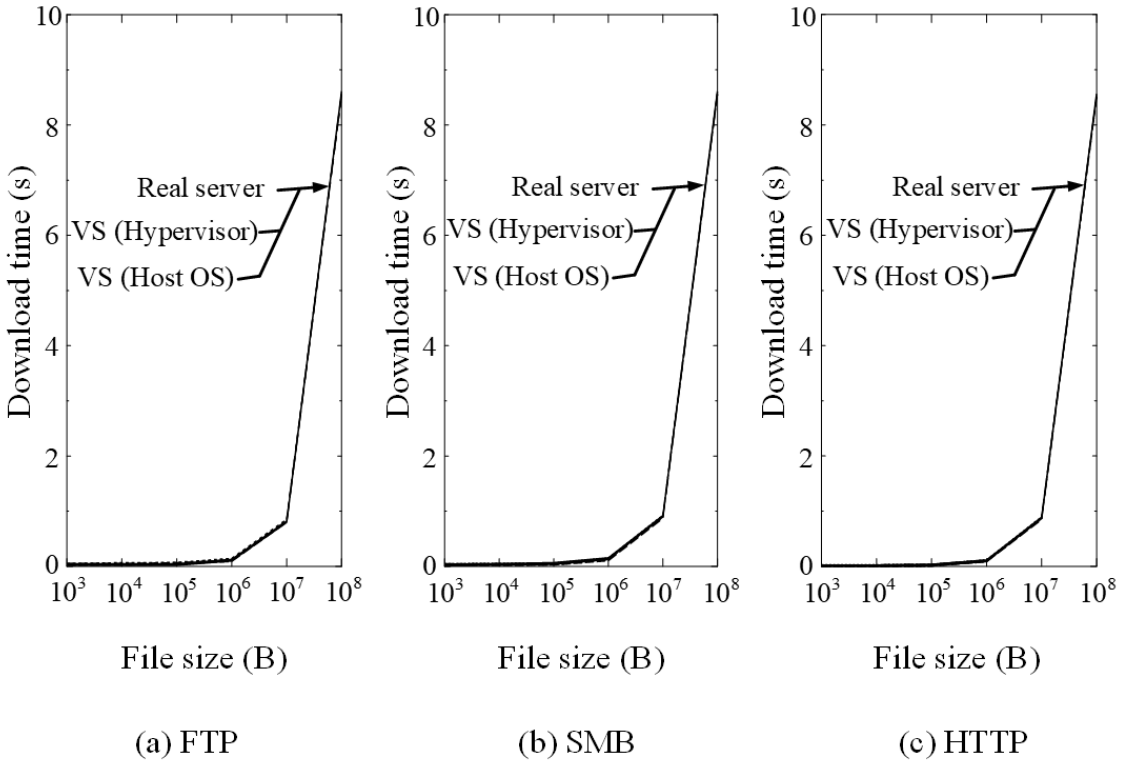


図 2.14　複数のプロトコルによるダウンロード性能調査

(a)，(b)と(c)において，いずれのプロトコルもファイルサイズの増加に対するダウンロード時間は，すべて同様の傾向を示すことがわかる．また，10^6B までの応答時間にはあまり差が見られていない．これは，各プロトコルにおいてデータ転送に要する時間の影響より，その転送におけるオーバヘッドによる時間の影響が強いためと考えられる．さらに，実サーバと 2 種類の仮想サーバにおける応答時間はいずれのファイルサイズにおいてもほぼ同等となっている．

図 2.15 に複数のプロトコルによるアップロード性能調査の結果を示す．(a)に FTP，(b)に SMB のアップロード結果を示す．Hypervisor 型の仮想サーバを“VS (Hypervisor)”，Host OS 型の仮想サーバを“VS (Host OS)”と示し，実サーバを“Real server”と示す．(a)と(b)の縦軸はアップロード時間とし，横軸はファイルサイズとする．

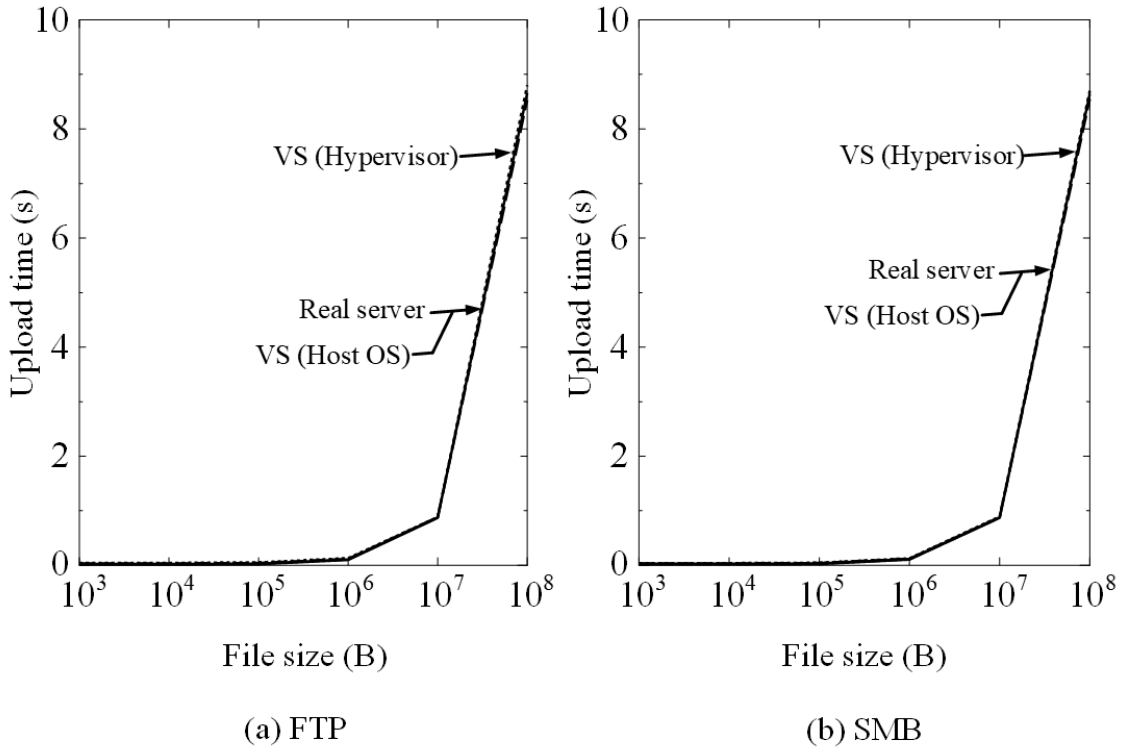


図 2.15　複数のプロトコルによるアップロード性能調査

(a)と(b)において，いずれのプロトコルもファイルサイズの増加に対するアップロード時間は，すべて同様の傾向を示すことがわかる．(a)，(b)ともに実サーバと Host OS 型の仮想サーバは同様の値を示し，Hypervisor 型の仮想サーバには若干の遅れが見られるが，10^8B において，(a)で 0.27 秒，(b)において 0.15 秒程度の差となった．また，10^6B までの応答時間にはあまり差が見られていない．これは，各プロトコルにおいてデータ転送に要する時間の影響より，その転送におけるオーバヘッドによる時間の影響が強いためと考えられる．

図 2.1 に示したように，MSBS は管理対象サーバのサーバ機能を復旧するために仮想サーバを使用する．そのため，ここでは仮想サーバの性能調査を行った．調査で使用する評価パラメータは，実際のネットワーク上で予想されるトラヒックにおいて，高負荷と想定される値を使用している．特に 10^8B データの 50 個連続ダウンロードアクセスやアップロードアクセス，また，CPU 処理時間において，10^8 回の演算を同時に 50 個実行させることはサーバに対して高負荷を生じさせることとなり，この状態において，仮想サーバは実サーバと同等の性能を示している．したがって，ここで得られた結果は，仮想サーバが実サーバの機能をバックアップすることが十分可能であることを示している．さらに，仮想サーバを起動する実サーバの仕様は管理対象サーバと同等かそれ以上が望ましいことも示している．

2.5 まとめ

MSBS の構成およびトラブルが発生したサーバの復旧手順を示した．MSBS は DBSS を複合的に起動することにより実現できることを示した．管理対象サーバの機能を復旧するために仮想サーバを使用することから，1 台の実サーバで複数の管理対象である実サーバの管理を可能とし，サーバ管理システムを低コストで構築可能であることを示した．

MSBS の基本となる DBSS の構築にはモード分割方式を提案し，採用した．モード分割方式は様々なシステムへの導入を可能としている．特に，本方式は様々な OS の管理対象サーバへの対処を考慮しており，管理対象サーバの OS に依存するモード部分のみの変更で動作が可能となる．本提案システムでは Regular mode，Service recovery mode，Network examination mode，Backup server mode と Target server recovery mode の 5 つのモードを使用し，各モードの詳細なメカニズムを示した．Service recovery mode，Network examination mode において，管理対象である実サーバの復旧処理に時間を要する部分にはバックグラウンドジョブを採用した．

また，MSBS 実験システムを構築し，構成および使用機器の仕様を示した．実サーバおよび仮想サーバの起動時間を含む特性も示し，サーバ機能の復旧時間を短縮するために，サーバ機能をバックアップするための仮想サーバは，常時，サスペンド状態で待機することとした．実験では，サーバトラブルとしてサービス提供プログラムとネットワークのトラブルを再現した．サービストラブルとしては，サービス提供プログラムの再起動により復旧する場合とその再起動では復旧せず，管理対象サーバを再起動することにより復旧する場合の 2 種類，ネットワークトラブルとしては，管理対象サーバの再起動およびシャットダウンの 2 種類を行った．実験において，それらのトラブルを管理対象サーバに対して再現し，サーバ機能の復旧時間，管理対象サーバの復旧時間とクライアントにおけるアクセス切断時間を実測した．いずれのトラブルにおいても，MSBS が対処可能であることを示すとともに，それらに対する実測値と予測値を示した．サーバ機能の復旧時間はいずれも 6 秒未満となり，クライアントでのアクセス切断時間は 12 秒未満となった．実測値と予測値はほぼ同等の値となり DBSS による管理対象サーバの機能およびそのサーバ自体の復旧時間を予測可能であることを示した．従来方式との比較では，MSBS は稼働率において非常に高い優位性があり，コスト面においても高い優位性があることを示した．

さらに，仮想サーバ性能調査システムを構築し，その構成および使用機器の仕様を示した．仮想サーバとしては Host OS 型として VMware Player，Hypervisor 型として KVM を採用し，DBSS で重要となる仮想サーバの評価を行った．性能調査システムにおいて，Host OS 型と Hypervisor 型が使用する実サーバのハードウエア環境を同様とするため，マルチ OS ブートシステムを採用した. サーバネットワークとクライアントからのアクセス速度を同様とした場合，仮想サーバを起動する実サーバの仕様によりダウンロードおよびアップロードの性能が異なることや CPU 処理に関しては仮想サーバと実サーバの処理速度が同様であることを示した．一般的なネットワーク設計である，サーバネットワークに比べクライアントからのアクセス速度が 10 分の 1 の環境において，ダウンロードおよびアップロードに要する時間が仮想サーバと実サーバは同等であることも示した．これは，MSBS を導入するサーバが管理対象サーバと同程度の仕様であれば，サーバ機能の復旧後に応答時間における遅延等の問題が発生しないことを示している．

第 3 章

異種システム混合処理方式を採用した省電力かつ高可用性サーバシステムの開発と稼働実験

3.1 まえがき

近年，高度情報化社会の発展に伴い，スマートフォンやタブレット端末などの通信機器の利用が進んでいる．今後，インターネットを利用した各種サービスは拡大を続け，その要求に応えるためには発展的な次世代ネットワークシステムが必要になると考えられる．このような高度なネットワークシステムにおいて，ユーザにより良いサービスを提供するにはこのシステムの基本単位であるクライアント・サーバシステムの最適化設計および構築が重要なポイントとなる[48].

また，今後予想されるサービスの増加に対処するためには，一般的にサーバ数の増加を避けることはできない．サーバ数が増加すると，それのハングアップや故障する可能性は増加することから，より信頼性の高いサーバシステムの構築が必要となり，対障害用機器の導入台数も増加すると考えられる．

サーバシステムに焦点をあてた障害対策において様々な観点から研究が進められており，サーバ故障とその復旧の概念を導入したマルチサーバシステムをモデル化[61]，仮想サーバのためのデバイスドライバ透過的故障検出および復旧メカニズムの提案[62]，クラウドシステムを構成するサーバ仮想化システムをソフトウエアの観点から最適化の検討[63]，データセンタにおいて複数のサーバが停止した場合でもすべてのユーザがサービスを継続できる高信頼化設計法[64]などが報告されている．サーバシステムを構成する上で，高速化，高信頼性に加え，省電力化についても十分に検討していくことが必要である[29],[65].

2.1 節で述べたように，IT デバイスにおける消費電力は 2006 年に比べ 2025 年では 9 倍に増加すると評価されている．省電力ネットワークシステムにおいては多くの重要な研究が既に行われている．無線センサネットワークにおいて分散クラスタ化と指向性アンテナの組み合わせによる省電力化の実現[66], ネットワークインタフェースにおける省電力と性能の両方に対して最適化を提案[67], 受動光ネットワークにおいてエネルギー消費を減少するためのアーキテクチャ[68]，無線センサネットワークにおける省電力デコーダの開発[69], トラヒック予測によるルータの動的制御[70]などが報告されている．加えて，サーバシステムにおける様々な報告もされている．複数のサーバ機能を 1 台の実サーバでバックアップ可能なシステムを導入することにより，低コストおよび省電力化の実現[40]，ARM プロセッサを搭載したサーバの処理性能および消費電力の検討[71], サーバ仮想化を用いてデータセンタにおけるサーバ電源の制御[72]などが報告されている．

しかしながら，IT デバイスの約半分はネットワークデバイスとサーバで構成されるが，サーバシステムにおける省電力の検討は非常に少なく，また，省電力かつ高可用性なシステムの検討は殆どされていないように見受けられる．

そこで，本章では，独立して動作可能な PSS と高可用性サーバシステムの 2 システムを交互に動作させることにより，その両面の機能を実現可能な異種システム混合処理方式を採用した省電力かつ高可用性サーバシステムを提案する．これらのシステムは独自の構成ファイルを使用しているが，共有可能な構成ファイルを使用することにより，それぞれのシステムにおいて他システムの影響による誤動作を防ぐことができる．PSS では，サービスを提供する仮想サーバの負荷を実測し，負荷が低い場合にはその仮想サーバを特定の実サーバに集約し，残りの実サーバをサスペンドすることにより省電力化を実現する．高可用性サーバシステムでは，システムの運用において，それぞれのサービスを提供する仮想サーバが冗長化構成されている通常状態や冗長化構成されていない省電力状態において，

サービスを提供している仮想サーバにトラブルが発生した場合の対処を自動的に行う．それぞれのシステムは 2 章で示したモード分割方式を採用しており，様々な状況に対応可能なモードを導入している．

3.2 省電力かつ高可用性を実現可能な提案システムの構成概要

ここでは異種システム混合処理方式を提案し，その方式を採用した省電力かつ高可用性サーバシステムを構築する．この方式は 2 つのシステムを交互に動作させ，それらのシステムから得られる機能を実現する．ここで提案する PSS と高可用性サーバシステムは独立して動作可能なシステムであり，これらを提案方式によって動作させることにより提案システムを実現する．また，提案方式はお互いのシステムでのサーバ状態を把握するために共有ファイルを使用する．このファイルを使用することにより，例えば，PSS が省電力化を実現するためにサーバをサスペンド状態に変更したとしても，高可用性サーバシステムがそのサーバを故障と判断することを避けることができる．

3.2.1 提案システムの構成

提案システムは，高可用性および負荷分散を行う上で一般的に使用されるロードバランサを導入したサーバシステムを基本としている．通常，ロードバランサは 1 台ではなく複数台設置して冗長化構成され，ロードバランシングシステムとして採用される．図 3.1 に提案システムの構成概要を示す．クライアントからのアクセスはロードバランシングシステムを経由して目的のサーバにアクセスされる．クライアントにサービスを提供するサーバには仮想サーバを使用し，ここでは Mail, Web や FTP のサービスを提供可能としている．1 種類のサービスに対して仮想サーバは冗長化構成され，そのサービス提供状態をここではサービスグループとする．クライアントからのアクセスはロードバランシングシステムに

よってサービスグループ毎の各仮想サーバにラウンドロビンアクセスされる．Virtual Server Group（VSG）には複数台の仮想サーバ，Real Server Group（RSG）にはこの VSG を起動するために複数台の実サーバが設置されている．

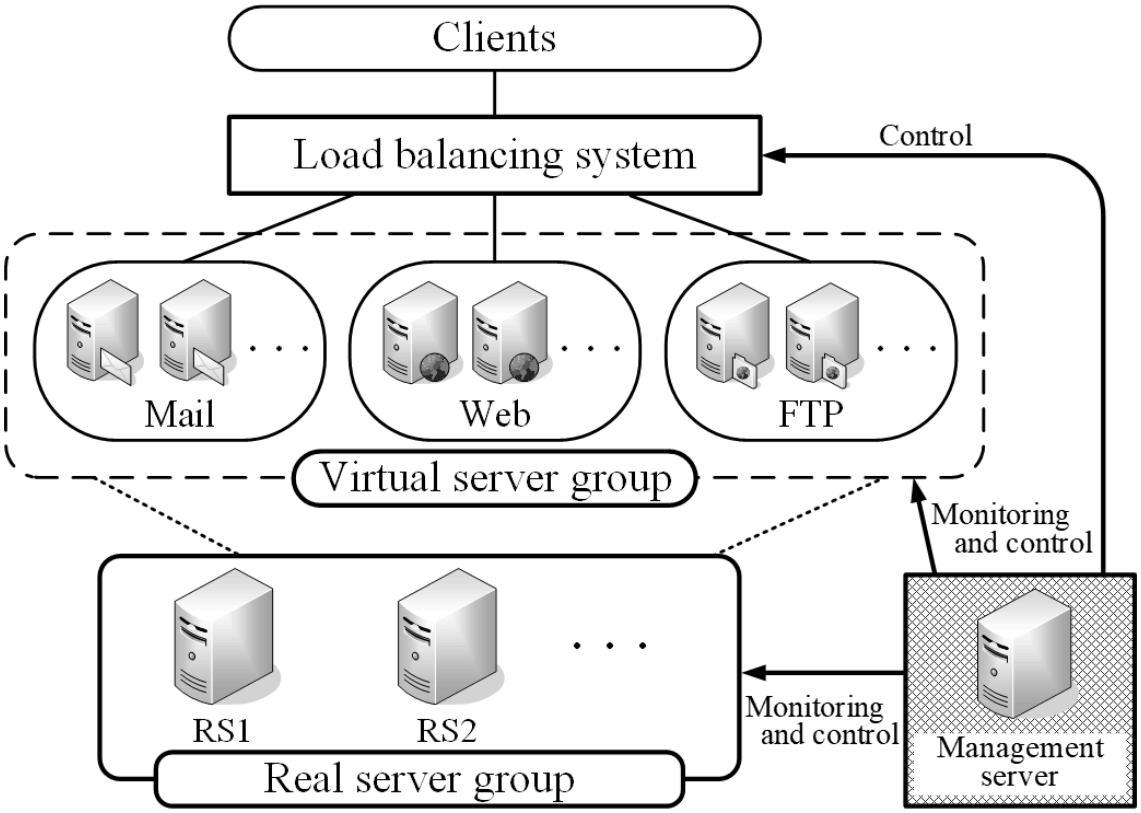


図 3.1　提案システムの基本構成

管理サーバには PSS を実現可能な省電力制御プログラムと高可用性サーバシステムを構築可能な高可用性制御プログラムがインストールされ，常に VSG と RSG の状態を監視している．状況に応じて VSG や RSG およびロードバランサの制御を行い，省電力かつ高可用性なサーバシステムの状態を維持する．省電力制御プログラムはクライアントから仮想サーバへのアクセス負荷が既定値より低くなった場合，特定の実サーバにサービスを提供

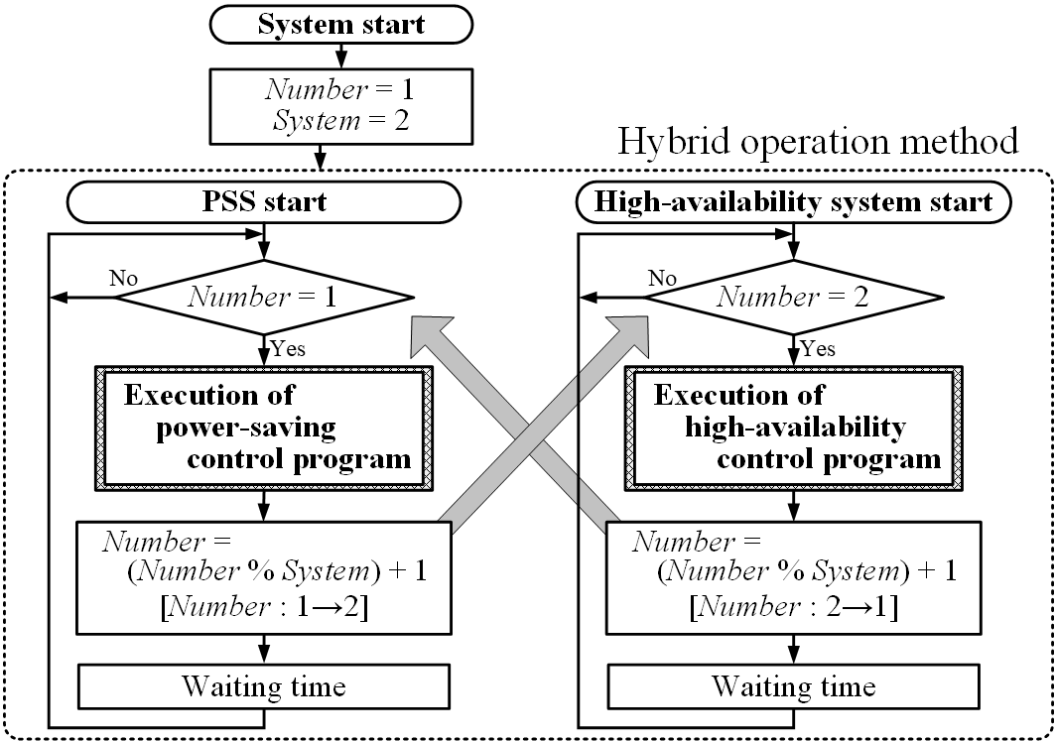
する仮想サーバを集約し，不要となった実サーバをサスペンド状態に変更することで省電力を実現する．高可用性制御プログラムはサービスを提供する仮想サーバにトラブルが発生した場合，その機能を有するバックアップ用の仮想サーバを起動することによって，トラブルを発生した仮想サーバの機能を復旧する．

3.2.2 異種システム混合処理方式

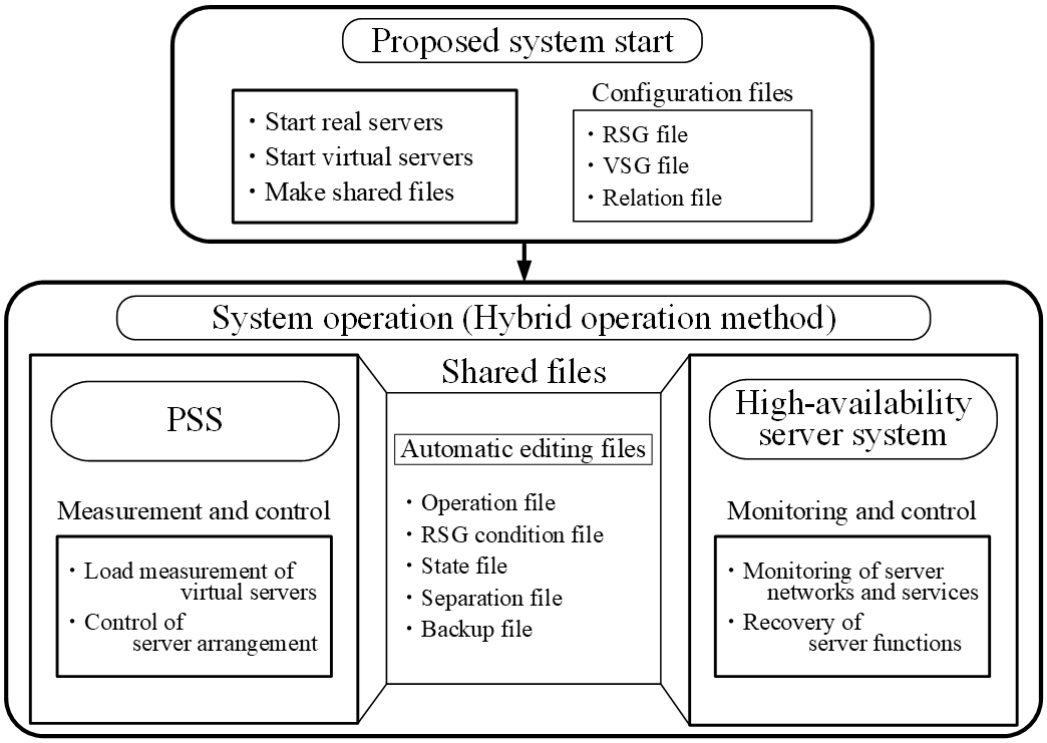
図 3.2 に異種システム混合処理方式の基本構成を示す．(a)には独立した PSS と高可用性サーバシステムの制御タイミングを示すフローチャート，(b)には提案システムを起動するために必要となる構成ファイルおよび各システムによって使用される共有ファイルを示す．(a)における“%”は余りを算出する演算子を示す．(a)において変数 *System* は起動するシステムの数を示し，変数 *Number* は実行すべきプログラムの番号を示す．システム起動時に変数 *Number* に 1，変数 *System* に 2 を代入し，PSS と高可用性サーバシステムが同時に開始される．変数 *Number* が 1 のため，省電力制御プログラムが実行され，終了すると変数 *Number* の値は式(3.1)から 2 となり PSS は待機状態となる．

$$Number = (Number \ \% \ System) + 1 \tag{3.1}$$

ここで，変数 *Number* が 2 となったため，高可用性制御プログラムが実行される．処理が終了すると，変数 *Number* の値は式(3.1)から 1 となり高可用性サーバシステムは待機状態となる．これにより 2 つの制御プログラムが交互に実行される．(b)では提案システムにおける起動と運用において必要となる構成ファイルを示しており，RSG ファイルは実サーバの詳細情報，VSG ファイルは仮想サーバの詳細情報が記録され，実サーバと仮想サーバの関連データは Relation ファイルに記録されている．これら 3 つを使用して提案システムを起動する．



(a) 省電力制御プログラムと高可用性制御プログラムの動作タイミング



(b) PSS と高可用性サーバシステムの構成ファイル

図 3.2　異種システム混合処理方式の基本構成

PSS と高可用性サーバシステムが使用する共有ファイルは提案システム起動時に自動的に作成され，その後，それらのシステムによって自動的に編集される．Operation ファイルはクライアントにサービスを提供する各仮想サーバの稼働状態, RSG condition ファイルは仮想サーバを起動する実サーバの稼働状態，State ファイルは各サービスグループの状態が記録される．PSS と高可用性サーバシステムは独立したシステムであることから，この共有ファイルによりそれぞれのシステムの状況が把握できるため，他システムの影響による誤動作を防ぐことができる．

3.2.3 構成ファイル

表 3.1 に提案システムの基本構成ファイルの詳細を示す. 提案システムはこれらのファイルの内容をもとに仮想サーバを起動して VSG を構成する．RSG ファイルはそれぞれの実サーバの仕様を記録したファイルで，サーバの物理メモリ量，IP アドレスと MAC アドレスを記録している．VSG ファイルは仮想サーバの詳細情報を記録したファイルで，所属するサービスグループ，仮想サーバ用システムデータが保存されているディレクトリパス，仮想サーバのホスト名，IP アドレス，設定するメモリ量，CPU 数と提供するサービス名が記録されている. 1 台のサーバがクライアントに対して複数のサービスを提供することが想定されるため，提供するサービスが複数の場合にはスペース区切りで複数のサービス名を記述することができる. 設定するメモリ量や CPU 数はこのファイルのデータを編集することにより，仮想サーバ起動時に，自動的にそれらの値が設定される．Relation ファイルは仮想サーバとそれを起動する実サーバおよびその仮想サーバの機能を復旧するために用意されたバックアップサーバ（仮想サーバ）との関連を示すファイルで，実サーバの IP アドレス，その実サーバから起動する仮想サーバのホスト名，その仮想サーバに対応したバッ

クアップサーバのホスト名が記録されている．もし，サービスグループに対してバックアップサーバを 1 台のみ用意する場合には，同サービスを提供する仮想サーバのバックアップサーバはすべて同じホスト名となる．ここで，提案システムではサービスを提供している仮想サーバのシステムデータが破損したとしても，バックアップサーバはそのシステムデータではなく別に用意したシステムデータを使用する方式を採用しているため，その機能を復旧することが可能となる．

表 3.1　RSG，VSG と Relation ファイルのフォーマット

RSG file

Memory (MB)	Real: IP address	MAC address
8,192	192.168.1.1	00:24:81:87:F6:AF
⋮		

VSG file

Service group	Path	Host name	Virtual: IP address	Memory (MB)	CPU	Service name
Mail	/dir/mail1/	mail1	192.168.1.51	2,048	1	smtp
FTP	/dir/ftp1/	ftp1	192.168.1.52	1,024	1	ftp
⋮						
Web	/dir/link/web/	webb	192.168.1.60	2,048	2	http

Relation file

Real: IP address	Virtual: host name	Backup: host name
192.168.1.1	mail1	mailb
192.168.1.2	ftp1	ftpb
⋮		

3.2.4　自動編集ファイル

本方式は表 3.1 に示した RSG，VSG と Relation ファイルをもとに提案システムを起動し，表 3.2 に示す自動編集ファイルを作成する．

表 3.2　自動編集ファイルのフォーマット

Operation file

Real: IP address	Path	Host name	Virtual: IP address	Service group	Service name
192.168.1.1	/dir/mail1/	mail1	192.168.1.51	Mail	smtp
192.168.1.2	/dir/ftp1/	ftp1	192.168.1.52	FTP	ftp
⋮					

RSG condition file

Real: IP address	Available memory size (MB)	Number of virtual servers
192.168.1.1	3,072	2
192.168.1.2	2,048	2
⋮		

State file

Service group	Condition	Base server Real: IP address
Mail	Moderate	192.168.1.1
FTP	Moderate	192.168.1.1
⋮		

Separation file

Real: IP address	Path	Host name	Virtual: IP address	Service group	Service name

Backup file

Virtual server host name (Trouble server)	Virtual server host name (Backup server)

Operation ファイルは起動している仮想サーバの詳細情報を記録したファイルで, 仮想サーバを起動する実サーバの IP アドレス，仮想サーバ用システムデータが保存されているディレクトリパス，仮想サーバのホスト名，IP アドレス，所属するサービスグループ，提供するサービス名が記録されている．RSG condition ファイルは実サーバの起動状況を示し，実サーバの IP アドレス，仮想サーバに割り当て可能なメモリ量とその実サーバ上で起動している仮想サーバの台数を記録している．State ファイルは状態を示すファイルで，サービスグループ名，そのグループの状態（Heavy，Moderate，Light），省電力時にサービスグループ毎の仮想サーバを集約する実サーバ(Base server)の IP アドレスが記録されている．Separation と Backup ファイルは運用時の状況に応じて必要となるファイルで，省電力時

にロードバランサからアクセスされない仮想サーバの情報は Operation ファイルから Separation ファイルに移動される．通常状態に戻る場合には，その仮想サーバの情報を Separation ファイルから Operation ファイルに戻す．また，仮想サーバのネットワークやサービスにトラブルが発生し，その機能をバックアップサーバで復旧した場合には，Backup ファイルにトラブルが発生した仮想サーバのホスト名とバックアップサーバのホスト名が記録される．

3.3 PSS の詳細構成

3.3.1 PSS のモード処理

図 3.3 に PSS における各モードの構成を示す. 本システムは, モード分割方式を採用し, (a)に示す Power-saving regular mode，Moderate mode，Light mode と Heavy mode の異なる 4 つのモードから構成される．Power-saving regular mode では，クライアントにサービスを提供する仮想サーバから構成されるサービスグループ毎の負荷を実測し，その負荷値が低い場合は Light load class，通常の場合は Moderate load class，高い場合は Heavy load class に分類する．本システムではその分類した 3 種類の Load class に応じた対処を行う．Moderate load class の場合には Moderate mode，Light load class の場合には Light mode，Heavy load class の場合には Heavy mode が対処する．ここで，Moderate mode によって構成された状態を“Moderate condition”，Light mode によって構成された状態を“Light condition”，Heavy mode によって構成された状態を“Heavy condition”とする．図に PSS の各モードにおける構成形態の一例を示す. RS1, RS2 と RS3 は実サーバを示し，A，B と C は仮想サーバが提供するサービスを示す．提案システムはシステム起動時に Moderate condition で稼働し，各サービスグループの仮想サーバは冗長化構成された状態でクライアントにサービスを提供する．

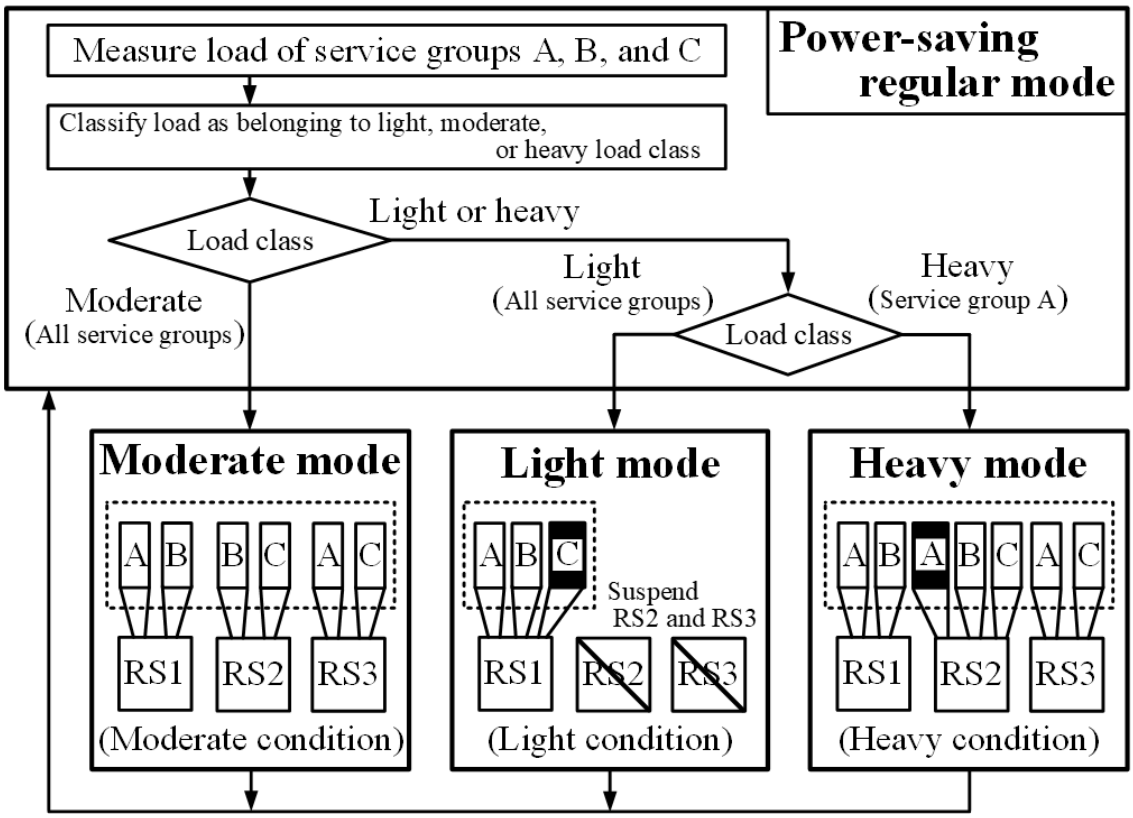
PSS では，それぞれのサービスグループの負荷を判断し，Moderate load class と判断した場合には Moderate mode を継続する．また，図では，Light mode への移行に関して，すべてのサービスグループの負荷が Light load class と判断された場合を示しており，省電力を効率的に実現するためサービス C 用の仮想サーバを RS1 上に起動し，その後，RS2 と RS3 をサスペンド状態に変更している．これにより Light condition における消費電力は Moderate condition における消費電力の約 3 分の 1 となる．次に，すべてのサービスグループの負荷が Moderate load class と判断されている Moderate mode において，サービス A を提供している仮想サーバのみが Heavy load class と判断された場合，Heavy mode に移行し，RS2 からサービス A 用の仮想サーバを起動して負荷を分散し，仮想サーバ 1 台あたりの負荷軽減を実現する．各モードにおける仮想サーバや実サーバの状態は，表 3.2 で示した Operation ファイル，RSG condition ファイル，Separation ファイルや State ファイルに反映される．

(b)にサービスグループ毎の Light condition の状態を示す．サービス A を提供している仮想サーバを V_{A1} と V_{A2}，サービス B を提供している仮想サーバを V_{B1} と V_{B2}，サービス C を提供している仮想サーバを V_{C1} と V_{C2} とし，サービス C 用のバックアップサーバを V_{Cb} とする．ここでは，すべてのサービスグループにおいて，表 3.2 で示した State ファイルの“Base server”を RS1 とする．(i)にすべてのサービスグループが Moderate condition の状態を示す．サービス A を提供する場合には実線の矢印，サービス B を提供する場合には破線の矢印，サービス C を提供する場合には点線の矢印でロードバランサからそれぞれラウンドロビンアクセスされる．各サービスグループは冗長化構成された仮想サーバによって構成されている．サービス A を提供している仮想サーバのみ Light load class と判断され，Light mode により Light condition となった状態を(ii)に示す．この状態ではクライアントからのサービス A に対する要求はロードバランサ経由で V_{A1} のみにアクセスされる．

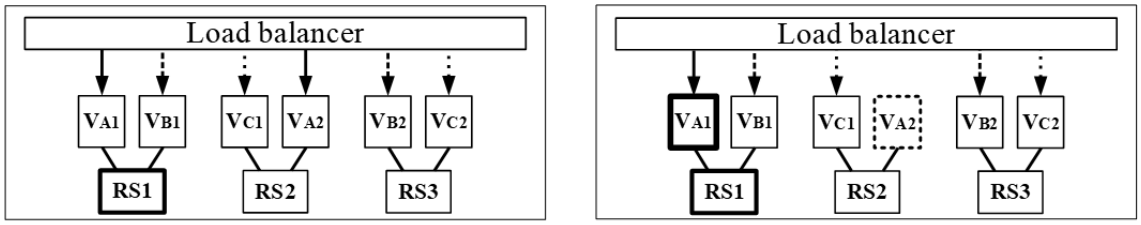
ここで，RS1 が Base server となっていることから V_{A2} は起動状態であるが，ロードバランサからアクセスされない状態となっている．サービス B を提供している仮想サーバのみ Light load class と判断され，Light mode により Light condition となった状態を(iii)に示す. この状態ではクライアントからのサービス B に対する要求はロードバランサ経由で V_{B1} のみにアクセスされる．ここで，RS1 が Base server となっていることから V_{B2} は起動状態であるが，ロードバランサからアクセスされない状態となっている．サービス C を提供している仮想サーバのみ Light load class と判断され，Light mode により Light condition となった状態を(iv)に示す．ここでは RS1 が Base server となっていることから，RS1 から V_{Cb} を起動する．クライアントからのサービス C に対する要求はロードバランサ経由で V_{Cb} のみにアクセスされる．V_{C1}，V_{C2} は起動状態であるが，ロードバランサからアクセスされない状態となる．

(c)に複数のサービスグループが Light condition となった場合を示す. サービス A とサービス B を提供している仮想サーバが Light load class と判断され, Light mode により Light condition となった場合を(i)に示す．RS2 から起動している V_{A2} と RS3 から起動している V_{B2} はロードバランサからアクセスされない状態となるが，RS2 から起動している V_{C1} と RS3 から起動している V_{C2} がロードバランサからアクセスされる状態であるため，実サーバをサスペンド状態に変更することができない．サービス A とサービス C を提供している仮想サーバが Light load class と判断され，Light mode により Light condition となった場合を(ii)に示す．RS2 から起動している V_{A2} と V_{C1} はロードバランサからアクセスされない状態となるため，RS2 をサスペンド状態に変更することができる．サービス B とサービス C を提供している仮想サーバが Light load class と判断され，Light mode により Light condition となった場合を(iii)に示す．RS3 から起動している V_{B2} と V_{C2} はロードバランサからアクセスされない状態となるため，RS3 をサスペンド状態に変更することができる．

すべてのサービスグループが Light load class と判断され，Light mode により Light condition となった場合を(iv)に示す．RS2 から起動している V_{A2} と V_{C1}，RS3 から起動している V_{B2} と V_{C2} はロードバランサからアクセスされない状態となるため，RS2 と RS3 をサスペンド状態に変更することができる．様々な状況において実サーバをサスペンドすることにより，省電力を実現することができる．



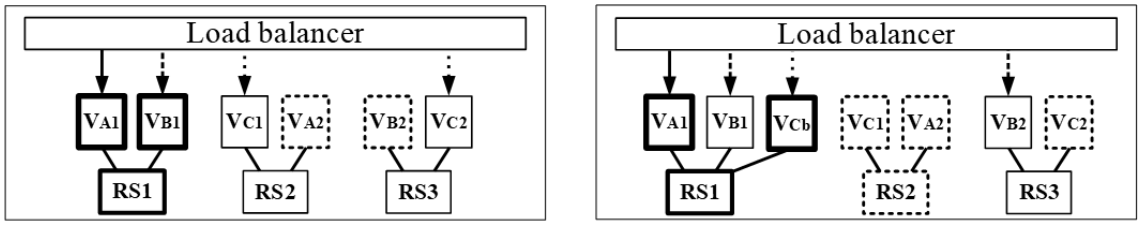
(a) 各モードの構成



(i) All service groups: Moderate condition　(ii) Service group A: Light condition

(iii) Service group B: Light condition　(iv) Service group C: Light condition

(b) サービスグループ毎の Light condition



(i) Service groups A and B: Light condition　(ii) Service groups A and C: Light condition

(iii) Service groups B and C: Light condition　(iv) All service groups: Light condition

(c) 複数のサービスグループによる Light condition

図 3.3　省電力サーバシステムの構成

3.3.2 CPU およびネットワーク負荷

PSS はクライアントにサービスを提供する仮想サーバから構成されるサービスグループ毎の負荷を実測し，その値から各モードへの移行を判断する．サーバはユーザに対して複数種類のサービスを提供することが想定できるが，一般的に要求データの送受信や演算などの CPU 処理をクライアントに対して提供することが多いと考えられる．そこで，本システムでは CPU の使用率とネットワークの転送バイト量をサーバの負荷として考え，図 3.4 に CPU およびネットワーク負荷の実測に関する詳細を示す．ここでは仮想サーバの OS に CentOS をインストールしていることを前提に述べる．CentOS では“/proc”ディレクトリ以下にシステム情報を処理するための基礎データファイルを保存している．CPU の使用状況は“/proc/stat”ファイルに累積データとして記録される．CPU 負荷の算出データとしては CPU 毎における Idle 時間の合計値を使用する．“/proc/stat”ファイルに記録されている Idle 時間は実際の Idle 時間に 100 を積算した値で示され，図の 19313 は 193.13 秒を示している．負荷の測定時刻を t_i，その時刻で測定される CPU の Idle 時間を C_i，測定間隔時間を I_t とし，仮想サーバの CPU 数を x+1 とする．CPU 負荷 L_{ci} は式(3.2)で示される．

$$L_{ci}\ [\%] = 100 - \frac{C_i - C_{i-1}}{I_t \times (x+1)} \tag{3.2}$$

ネットワークの使用状況は“/proc/net/dev”ファイルに累積データとして記録される．このデータは Network Interface Card（NIC）毎に記録されるため，ネットワーク負荷の算出データとして，ここでは図の“eth0”で示される“Receive bytes”と“Transmit bytes”を使用する. 負荷の測定時刻を t_i, その時刻で測定される“Receive bytes”と“Transmit bytes”の合計値を N_i，測定間隔時間を I_t とする．ネットワーク負荷 L_{ni} は式(3.3)で示される．

$$L_{ni}\ [\mathrm{Byte/s}] = \frac{Ni - N_{i-1}}{I_t} \tag{3.3}$$

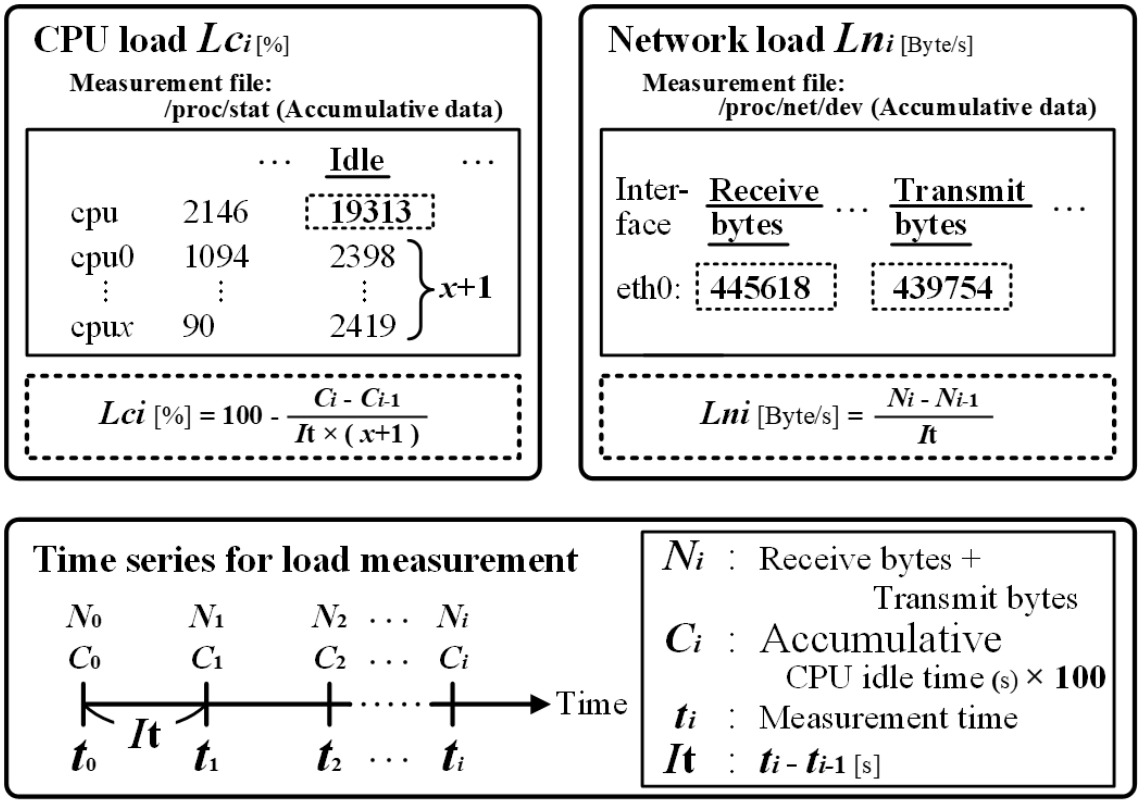


図 3.4　CPU およびネットワーク負荷

3.3.3　Load class の分類

図 3.5 に負荷を判断するためのしきい値と Load class の分類方法について示す．PSS では負荷値を Light load class，Moderate load class と Heavy load class の 3 種類に分類する．分類するためのしきい値としては，Heavy load，Moderate load1，Moderate load2 と Light load の 4 つを設定している．通常であればしきい値は 2 つで十分と考えることができるが，しきい値近辺での負荷変動による Load class 値の頻繁な変動を防ぐために 4 つとしている．図に示すように，Heavy load のしきい値以上を領域 H，Moderate load2 のしきい値以上 Heavy load のしきい値未満を領域 HM，Moderate load1 のしきい値以上 Moderate load2 のしきい値未満を領域 M，Light load のしきい値以上 Moderate load1 の

しきい値未満を領域 ML，Light load のしきい値未満を領域 L とする．(a)から(k)は負荷値の推移を示しており，黒丸が初期の状態で，矢印の先が最終の状態を示す．(a)，(d)や(j)のように同領域へ推移した場合には初期の状態が Load class と判断される．(k)のように領域 M から領域 L に負荷値が推移した場合には Light load class，(g)や(h)のように領域 L から領域 M，HM や H または領域 H から領域 M，ML や L に推移した場合には Moderate load class，(b)のように領域 M から領域 H に推移した場合には Heavy load class と判断する．ここで，(c)や(f)のように領域 HM に接している領域から領域 HM に負荷値が推移した場合および (e)や(i)のように領域 ML に接している領域から領域 ML に負荷値が推移した場合には，初期の状態が Load class として判断される．

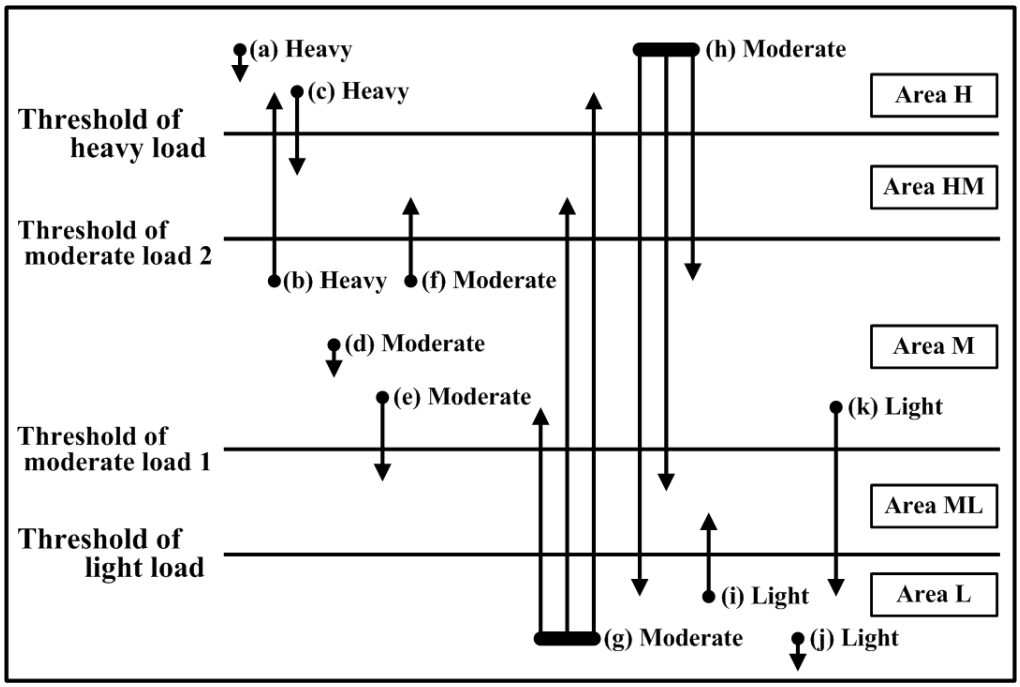


図 3.5　負荷判断のしきい値と Load class の分類

3.3.4 モード変更の判断

図 3.6 に CPU とネットワーク負荷の両方を考慮した Load class の判断と各モードへの移行について示す．図 3.4 で示したように，サービスグループ毎の負荷は CPU とネットワークの両方から構成される．そのため，図に示すように CPU における Load class とネットワークにおける Load class でマトリクスを構成し，いずれかの高い Load class を優先し，そのサービスグループの Load class として決定する．例えば，CPU 負荷が Light load class で，ネットワーク負荷が Heavy load class となった場合には，そのサービスグループの負荷は Heavy load class と判断する．逆に，CPU 負荷が Heavy load class で，ネットワーク負荷が Light load class となった場合にも，そのサービスグループの負荷は Heavy load class と判断する．

サーバシステムにおける負荷状態は様々である．例えば，日中は継続的に負荷が高く，夜間は負荷が低くなる場合や日中においても時刻によって負荷が大きく変動する場合など，提供するサービスなどによって負荷変動のパターンは異なる．そのため，負荷変動に応じた各モードへの移行に関しては，サーバシステムの使用環境を考慮して決定することが望ましい．そのため，モード毎に連続回数を設定し，同じ Load class がその回数分連続した場合にモードの変更を実行するようにしている．それにより，過剰なモードの変更によるシステム構成の変更を防ぐことができる．

PSS では，図に示すように Moderate mode の状態において，Light load class と連続 *Lc* 回判断された場合に Moderate mode から Light mode に移行しているのがわかる．Moderate mode への移行は *Mc* 回，Heavy mode へは *Hc* 回連続してその Load class として判断された場合にモード変更が行われる．

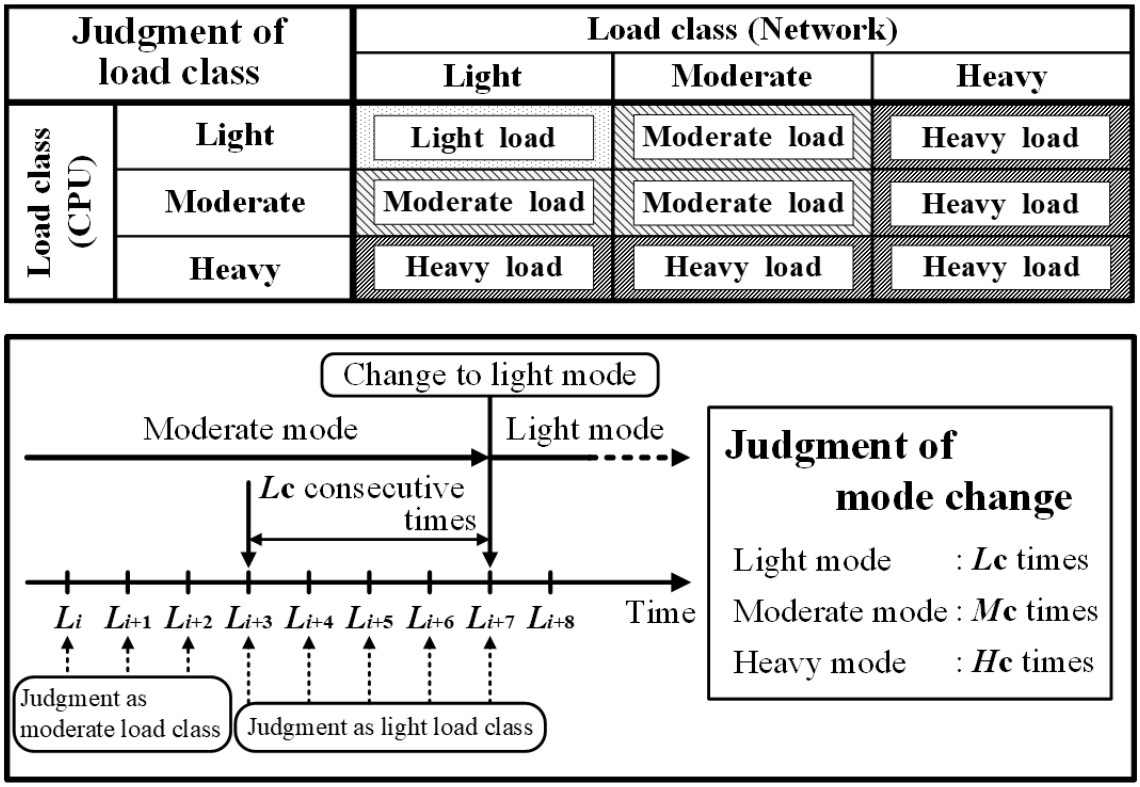


図 3.6　負荷判定とモード変更

3.4　高可用性サーバシステムの特徴

提案システムは PSS における Heavy mode，Moderate mode と Light mode によって実現される Heavy condition，Moderate condition や Light condition の状態において，高可用性を維持する必要がある．本章では 2 章で示した MSBS の方式とは異なり，1 つの制御プログラムで複数の管理対象サーバを管理する方式を採用している．

3.4.1　高可用性サーバシステムのモード処理

図 3.7 に高可用性制御プログラムにおける動作フローを示す. このプログラムは通常処理を行うフォアグラウンドジョブと管理対象サーバを復旧するために時間を要する処理を行

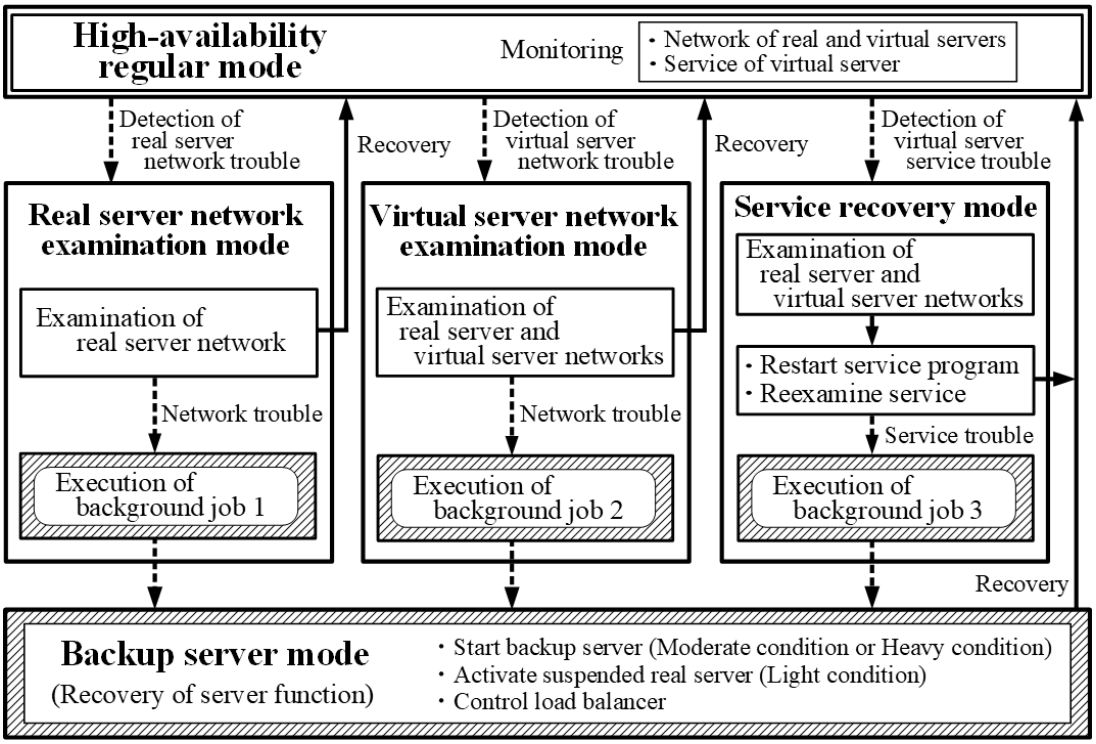
うバックグラウンドジョブの 2 種類から構成される．

(a)にモード分割方式による様々なトラブルに対処するための処理フローを示す．本高可用性サーバシステムはモード分割方式を採用しており，High-availability regular mode, Real server network examination mode，Virtual server network examination mode, Service recovery mode と Backup server mode の 5 種類から構成される. High-availability regular mode は一定間隔時間毎に表 3.2 で示した Operation ファイルをもとに RSG および VSG 内の実サーバや仮想サーバのネットワークとクライアントへのサービス提供状態を調査する．本システムではネットワークの監視に UNIX の ping コマンドを使用している．また，サービス提供状態の監視には 2 章の図 2.4 で示した Expect プログラム言語を使用している．この言語はタイムアウト機能や管理対象サーバのサービスポートにアクセスし，その返答に応じた処理を行うことができるため，クライアントへのサービス提供状態を正確に監視することが可能となる．High-availability regular mode で実サーバのネットワークに異常を検出した場合，Real server network examination mode にプログラムの制御状態が移行される．ここでは実サーバのネットワークを再調査し，ネットワークが正常であれば High-availability regular mode に戻り，異常と判断すると Background job1 をバックグラウンドジョブとして実行し，Backup server mode に移行する．High-availability regular mode で仮想サーバのネットワークに異常を検出した場合, Virtual server network examination mode にプログラムの制御状態が移行される．ここでは仮想サーバのネットワークを再調査し，ネットワークが正常であれば High-availability regular mode に戻り，異常と判断すると Background job2 をバックグラウンドジョブとして実行し, Backup server mode に移行する．High-availability regular mode で仮想サーバがクライアントに提供しているサービスに異常を検出した場合，Service recovery mode にプログラムの制御状態が移行される．ここではサービス異常を検出した仮想サーバのサービス提供プログラムを再

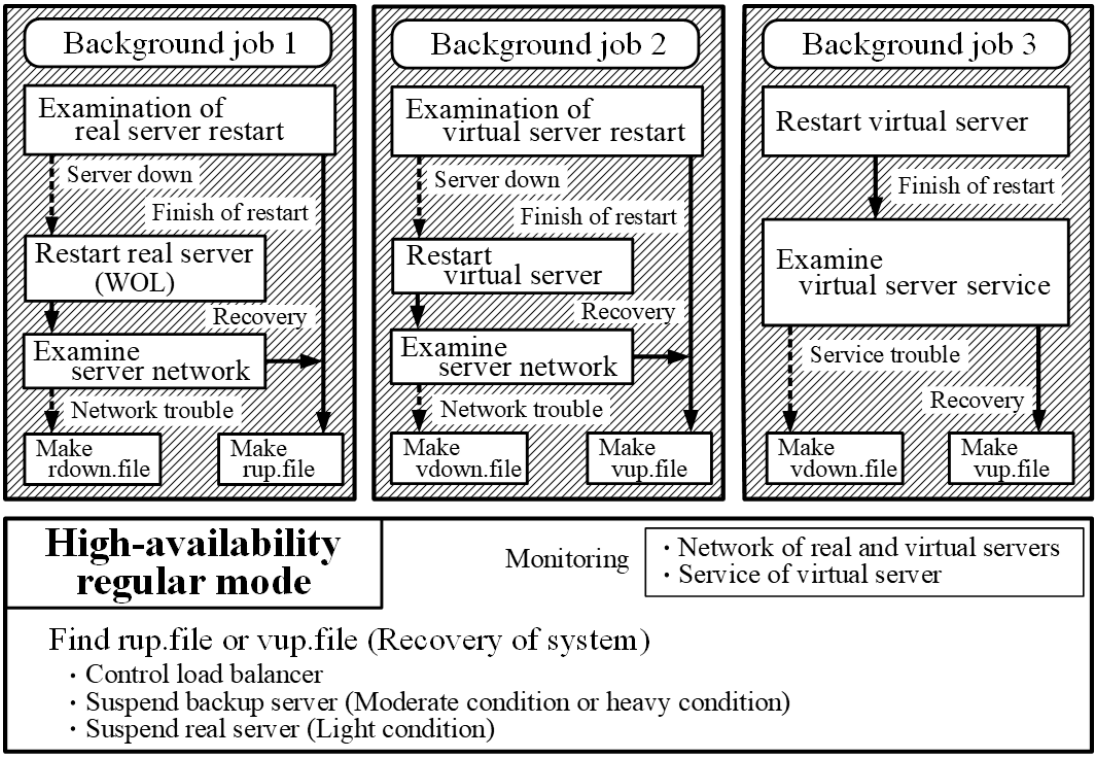
起動し，サービスの状態を再調査する．そのサービスが正常であれば High-availability regular mode に戻り，異常と判断すると Background job3 をバックグラウンドジョブとして実行し，Backup server mode に移行する．Backup server mode では表 3.2 の State ファイルをもとに異常を示しているサービスグループが，Moderate または Heavy condition の場合はバックアップサーバの起動，Light condition の場合はサスペンド状態である実サーバの起動を行い，その後，ロードバランサを制御して故障したサーバの機能を復旧する．バックアップサーバによりサーバの機能を復旧した場合には，表 3.2 の Backup ファイルにその情報が記録される．

(b)に Background job1，2 と 3 の処理および各バックグラウンドジョブの結果にもとづく High-availability regular mode における処理を示す．もし，管理対象の実サーバや仮想サーバのネットワークが異常の場合，サーバ管理者による意図的なサーバの再起動やトラブルによる停止が考えられる．このような管理対象サーバの状況を把握するには一定時間の調査が必要となり多くの時間を要する．また，仮想サーバのサービス異常を復旧するために必要となる作業も多くの時間を要する．このため，本システムではこれらの復旧作業はバックグラウンドジョブとして実行する．

Background job1 では管理対象の実サーバが再起動状態か調査し，再起動完了後，その実サーバのネットワークが正常であればその IP アドレスを記録した rup.file を作成し，再起動状態ではないと判断すると WOL を使用してそのサーバの起動処理を試みる．その後，その実サーバのネットワークを調査し，正常状態であればその実サーバの IP アドレスを記録した rup.file を，異常状態であればその IP アドレスを記録した rdown.file を作成して終了する．



(a) モード分割方式による様々なトラブル対処の処理フロー



(b) Background job1，2 と 3 の役目

図 3.7　高可用性サーバシステムの動作フロー

Background job2 では管理対象の仮想サーバが再起動状態か調査し，再起動完了後，その仮想サーバのネットワークが正常であればその IP アドレスを記録した vup.file を作成し，再起動状態ではないと判断した場合，そのサーバの起動処理を試みる．その後，その仮想サーバのネットワークを調査し，正常状態であればその IP アドレスを記録した vup.file を，異常状態であればその IP アドレスを記録した vdown.file を作成して終了する．

Background job3 ではサービスの異常を検出した仮想サーバを再起動し，再起動完了後，その仮想サーバのサービス提供状態を調査し，正常状態であればその IP アドレスを記録した vup.file を，異常状態であればその IP アドレスを記録した vdown.file を作成して終了する．

High-availability regular mode では一定間隔時間毎にバックグラウンドジョブの結果である rup.file，vup.file，rdown.file や vdown.file の存在を確認する．ここで，rup.file や vup.file を検出した場合，ロードバランサを制御してサーバシステムを通常の状態に戻す．PSS における Moderate や Heavy condition ではトラブルが発生した仮想サーバの機能を復旧するためにバックアップサーバを起動していることから，そのバックアップサーバをサスペンド状態に戻す．Light condition ではトラブルが発生した仮想サーバの機能を復旧するためにサスペンド状態の実サーバを起動していることから，その実サーバをサスペンド状態に戻す．また，rdown.file や vdown.file を検出した場合は，対象の実サーバや仮想サーバが復旧できなかったことを意味し，現状の状態のままサーバシステムを稼働する．

3.4.2 ロードバランサの制御

図 3.8 に Backup server mode におけるロードバランサにおける制御の流れを示す．ここではロードバランサ用のシステムプログラムとして UltraMonkey-L7 を例として記述する．

このモードでは管理対象サーバの機能を復旧するため，Moderate condition の場合には管理対象サーバと同様のサービスを提供可能な仮想サーバをバックアップサーバとして起動し，Light condition の場合にはサスペンド状態の実サーバ上に起動している仮想サーバを使用するため，その実サーバを稼働状態に戻す．その後，クライアントからのアクセスを故障中の管理対象サーバから，Moderate condition の場合にはバックアップサーバに変更，Light condition の場合には故障中の管理対象サーバと冗長化構成されている仮想サーバに変更するため，ロードバランサの制御を行う．図では Moderate condition の場合を例に示す．図に示すように，クライアントはロードバランサの 25 番ポート（SMTP）にアクセスすると，ロードバランサを経由して Mail サーバにアクセスされる．Mail サーバが正常状態の場合は IP アドレス“172.21.14.1”の仮想サーバにアクセスを行う．その仮想サーバにトラブルが発生した場合，このモードではバックアップサーバの起動が完了後，(a)に示すようにロードバランサ上の LB 設定ファイル内に記録されている IP アドレス“172.21.14.1”を“172.21.14.101”と変更し，(b)に示すようにロードバランサ用プログラムに対して再読み込みを実行する．これにより，図に示すようにロードバランサ経由でアクセスされる IP アドレスが“172.21.14.101”に変更されるため，クライアントからのアクセスをバックアップサーバに切り替えることができる．

ロードバランサでは，独自に LB 設定ファイル内に記録されているクライアントにサービスを提供するサーバの監視を行っており，図に示す，“checktimeout”，“checkinterval”，“retryinterval”と“checkcount”によりトラブル時の対応が決定される．“checktimeout”はネットワーク監視およびサービスポート監視で，この設定秒数未満で正常な応答が得られない場合は，監視失敗として監視失敗回数を 1 増加する．“checkinterval”はこの設定秒数間隔でサーバの監視を行う．“retryinterval”は監視失敗時に，この設定秒数間隔で監視を行う．“checkinterval”より短い秒数を設定することで，異常検出にかかる時間を短縮で

きる．“checkcount”は監視失敗回数が，この設定回数に達した場合に対象サーバを異常と判断し，クライアントからのアクセスをそのサーバに対して行わないようにする．冗長化構成のサーバに対してこれらを設定することにより，クライアントにサービスを提供するサーバに異常が発生してもクライアントへの影響を最小限またはゼロにすることができる．

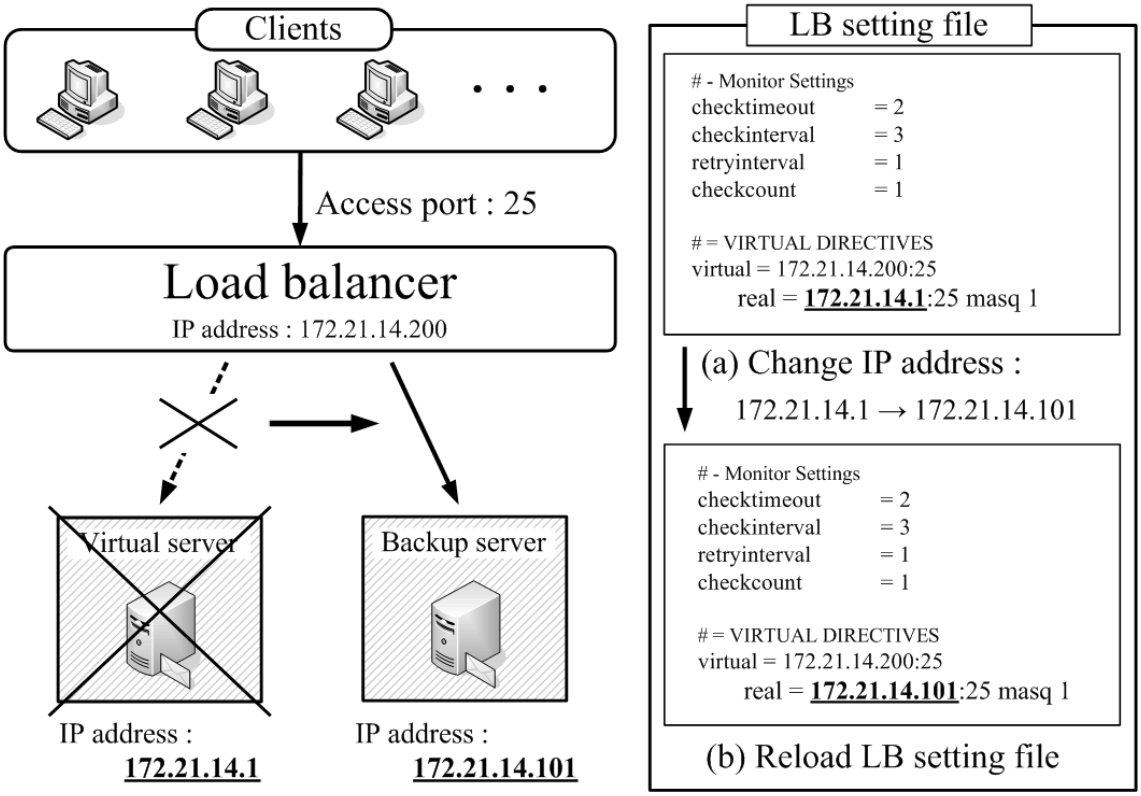


図 3.8　ロードバランサ制御の流れ

3.4.3　Moderate および Light condition におけるサーバ機能の復旧

図 3.9 に PSS によって制御された Moderate condition において，仮想サーバ，実サーバの故障時における対処方法を示す．ここではすべてのサービスグループが Moderate condition となった場合を前提としている．また，Heavy condition における対処は

Moderate condition と同様となるため省略する．M1 と M2 はサービスグループ Mail，W1 と W2 はサービスグループ Web とし，F1 と F2 はサービスグループ FTP の仮想サーバとする．サービスグループ Mail のバックアップサーバを Mb，サービスグループ Web のバックアップサーバを Wb とし，サービスグループ FTP のバックアップサーバを Fb とする．

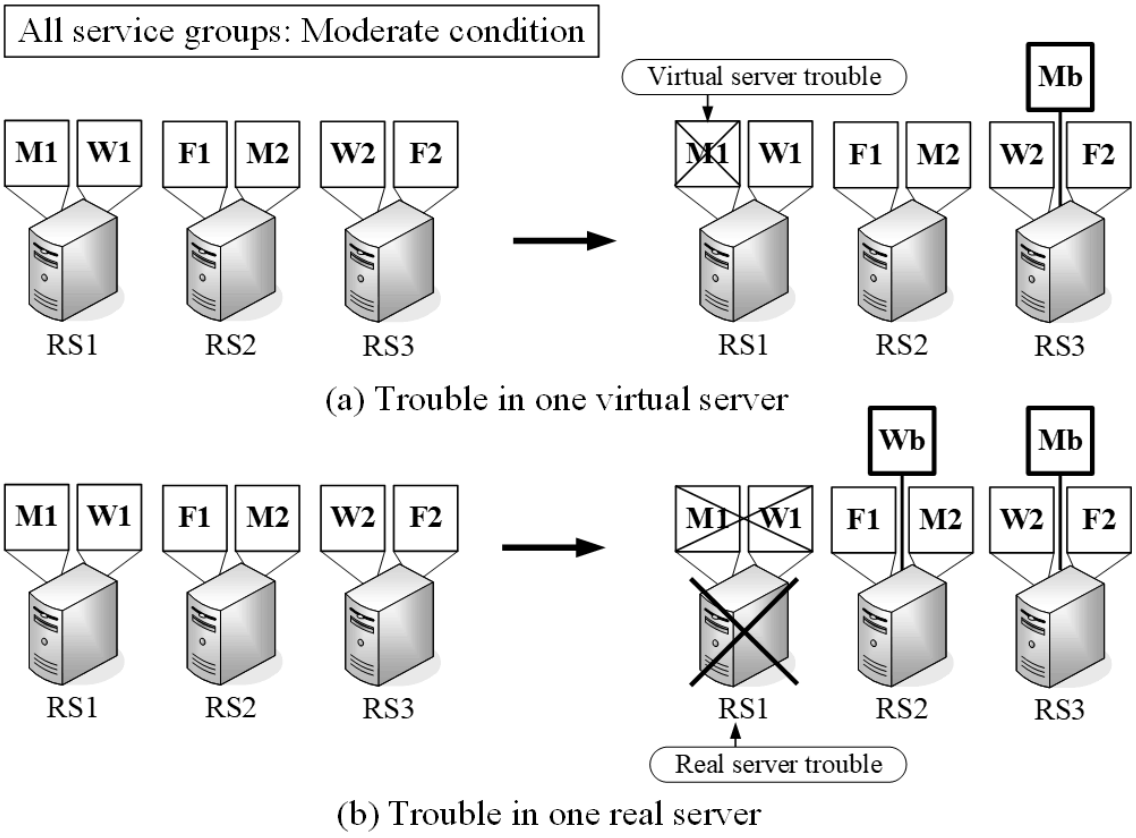


図 3.9　Moderate condition におけるサーバ機能の復旧

Moderate condition では 3 台の実サーバ RS1，RS2 と RS3 からそれぞれ 2 種類の仮想サーバが起動し，各サービスグループの仮想サーバは冗長化構成されている．(a)に仮想サーバ M1 が故障した場合の対処を示す．この場合，高可用性サーバシステムでは M1 の機能を復旧するため RS3 上に Mb を起動し，ロードバランサを制御してクライアントからの

アクセスを M1 から Mb に切り替える．ここで Mb を RS3 から起動したのは，サービスグループ Mail の冗長関係である M2 が RS2 上に起動しているためである．(b)に RS1 が故障した場合の対処を示す．RS1 上に起動している M1 と W1 も同時に故障状態となるため，M1 の機能を復旧するために RS3 上に Mb，W1 の機能を復旧するために RS2 上に Wb を起動し，ロードバランサを制御してクライアントからのアクセスを Mb と Wb に切り替える．ここで，RS2 上に Mb，RS3 上に Wb を起動したのは，冗長関係となる仮想サーバは異なる実サーバから起動した方が望ましいからである．

図 3.10 に PSS によって制御された Light condition において，仮想サーバ，実サーバの故障時における対処方法を示す．ここではすべてのサービスグループが Light condition となった場合を前提としている．Light condition では，省電力の効率を最大にするため，各サービスグループにおける仮想サーバは冗長化構成されていない. RS1 上に仮想サーバ M1, W1 と Fb が起動しており，RS2，RS3 はサスペンド状態となっている．ここで，サスペンド状態となっている実サーバ上に起動している仮想サーバは起動状態のままサスペンドされている．斜線で示される仮想サーバがクライアントからアクセス可能なことを示している．(a)に仮想サーバ M1 が故障した場合の対処を示す．この場合，高可用性サーバシステムでは M1 の機能を復旧するためサスペンド状態の RS2 を稼働状態に戻し，ロードバランサを制御してクライアントからのアクセスを M1 から M2 に切り替える．ここで，RS2 上で稼働している F1 は起動状態だがクライアントからのアクセスはされない状態となっている．(b)に RS1 が故障した場合の対処を示す．RS1 上に起動している M1，W1 と Fb も同時に故障状態となるため，この状態ではクライアントに対してサービスを提供できない．高可用性サーバシステムでは M1，W1 と Fb の機能を復旧するためサスペンド状態の RS2 と RS3 を稼働状態に戻し，ロードバランサを制御してクライアントからのアクセスを F1, M2 と W2 に切り替える．

Moderate condition における実サーバや仮想サーバの故障に関しては，サービスグループの仮想サーバが冗長化構成となっているため，クライアントへの影響はないが，Light condition における実サーバや仮想サーバの故障に関しては，クライアントに対してサーバ機能を復旧するまでの時間が影響してしまう．

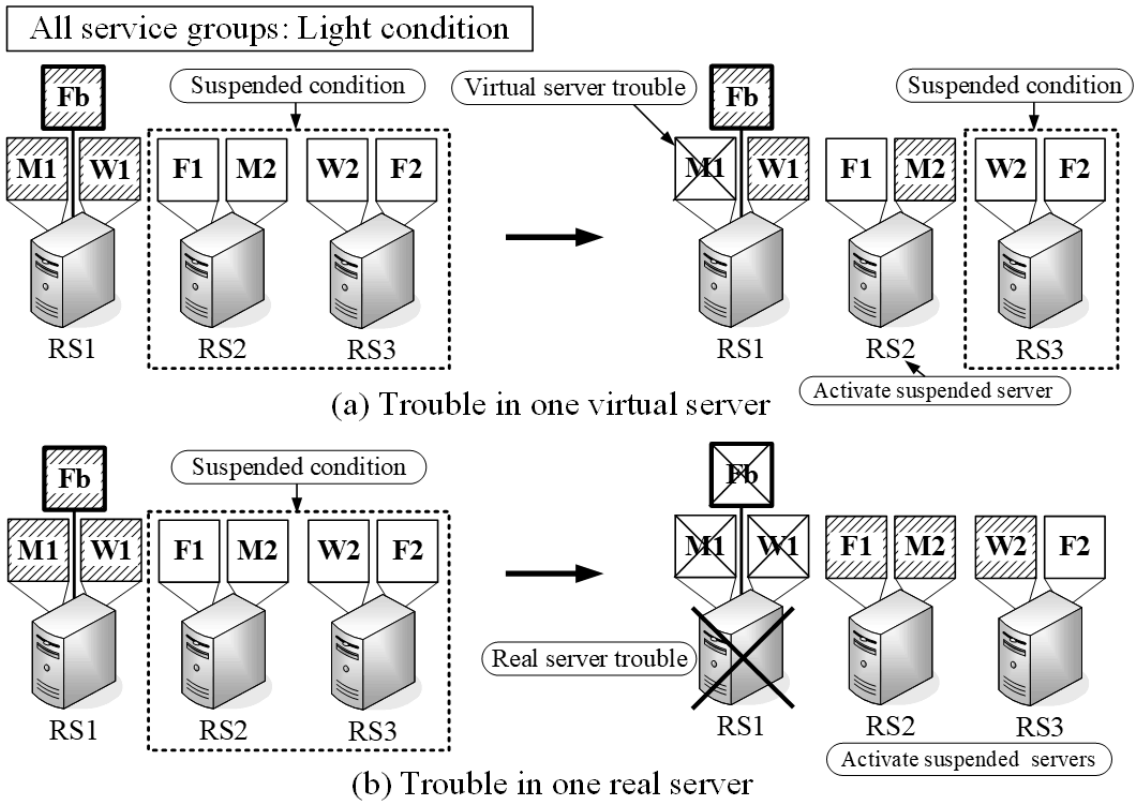


図 3.10　Light condition におけるサーバ機能の復旧

3.5　省電力かつ高可用性サーバシステムの動作検証

3.5.1　実験システムの構成および仕様

図 3.11 に実験で使用する省電力かつ高可用性サーバシステムの構成を示す．実験システムは NFS サーバの機能を有する管理サーバが 1 台，実サーバ（RS1，RS2，RS3）が 3 台，

ロードバランサが 1 台，クライアントが 1 台，1000BASE-T のスイッチング HUB が 2 台と 100BASE-TX のスイッチング HUB が 1 台から構成される．一般的にサーバ間のネットワーク転送速度に対してクライアントからの転送速度は 10 分の 1 程度となるため 100BASE-TX のスイッチング HUB を使用している．

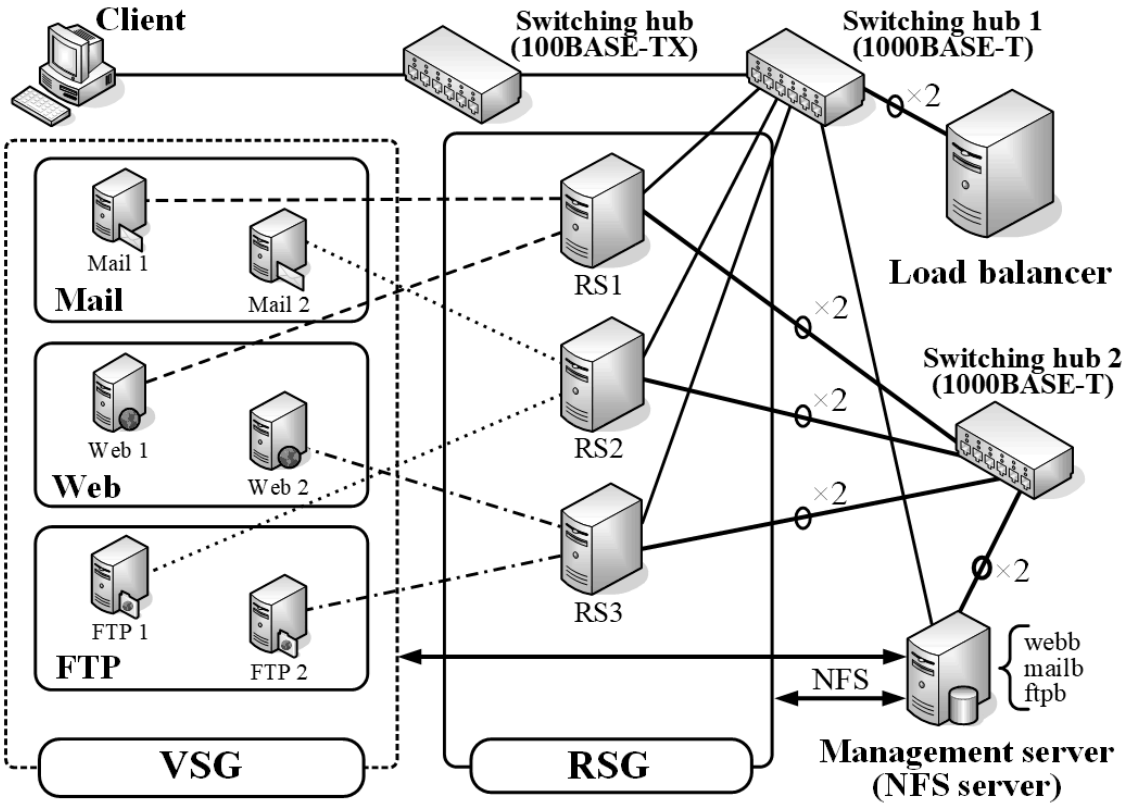


図 3.11　実験用省電力かつ高可用性サーバシステムの構成

VSG はクライアントにサービスを提供する仮想サーバ群で，RSG は VSG を起動する実サーバ群を示す．VSG にはサービスグループ Mail，Web と FTP を展開し，Mail サーバが 2 台，Web サーバが 2 台と FTP サーバが 2 台はそれぞれ冗長化構成されている．本実験システムでは Mail サーバは SMTP に対応する postfix, Web サーバは HTTP に対応する httpd,

FTP サーバは FTP に対応する vsftpd をインストールし，クライアントに対してそれぞれのサービスを提供する．仮想サーバを起動するためのシステムデータは各実サーバに通常起動用を 2 種類, バックアップサーバ用として管理サーバに 3 種類（Webb, Mailb, FTPb）保存している．冗長化構成となっている仮想サーバはクライアントに同様のサービスを提供するため，また，各実サーバは管理サーバ上に保存されているバックアップサーバ用のシステムデータを使用してバックアップサーバを起動するため，管理サーバと NFS 接続している．1000BASE-T のスイッチング HUB1 がクライアントへのサービス提供用で，1000BASE-T のスイッチング HUB2 は管理サーバと VSG や RSG との NFS 接続で使用する．

表 3.3 に実験システムにおける各実サーバおよび仮想サーバの仕様を示す．管理サーバ，実サーバとロードバランサにはサーバ用の OS として採用されることが多い CentOS の 64bit 版をインストールしている．各実サーバは同様の仕様で，CPU は Intel Core i7-960 を搭載し, 物理メモリは 8,192MB とし, 仮想化ソフトウエアとして世界的に有名な VMware Player と仮想サーバの高速起動およびウインドウを表示しない状態での稼働を可能とするため VIX API を採用する．また，各実サーバはメモリの消費を抑えるため CUI 環境で稼働する．管理サーバは，省電力および高可用性制御プログラムを動作させるとともに NFS の機能を有することから，CPU として Intel Core i7-930 を搭載し，物理メモリは 6,144MB とする. 各実サーバと管理サーバには NIC を 3 枚, ロードバランサには 2 枚搭載している．実サーバの NIC を 3 枚としたのは実験システムにおいて，クライアントからのアクセスに対してシステムが使用する通信が影響しないように考慮している．また，ロードバランサの NIC を 2 枚としたのは，クライアントとサーバの通信容量を拡張するためである．ロードバランサはソフトウエア処理方式で，UltraMonkey-L7 をインストールしている．仮想サーバは，CPU 数を 1，メモリを 1,024MB，NIC を 2 つ設定，サービス提供プログラムと

して Mail サーバには postfix，Web サーバには httpd，FTP サーバには vsftpd を採用し，OS として CentOS の 64bit 版をインストールしている．

表 3.3　実験用省電力かつ高可用性サーバシステムの仕様

Specifications	Real server (RSG)	Management server (NFS server)
CPU	Intel Core i7-960 3.20 GHz (TB: 3.46 GHz)	Intel Core i7-930 2.86 GHz (TB: 3.06 GHz)
Memory	8,192 MB	6,144 MB
Hard disk	SATA 2 (7,200 rpm)	
NIC	3	
System software	VMware Player 6.0.6 VIX-API 1.13.6	nfsd
OS	CentOS 6.7 (64-bit)	

Specifications	Virtual server (VSG)
CPU	1
Memory	1,024 MB
NIC	2
System software	postfix, httpd, vsftpd
OS	CentOS 6.7 (64-bit)

Specifications	Load balancer
CPU	Intel Core i7-3770 (3.40 GHz / TB: 3.90 GHz)
Memory	8,192 MB
NIC	2
System software	UltraMonkey-L7
OS	CentOS 6.7 (64-bit)

図 3.12 に実験システムにおけるサーバネットワークの詳細構成を示す．管理サーバと各実サーバは NIC を 3 枚搭載しており，それぞれを Ethernet0，Ethernet1 や Ethernet2 と示す. 1000BASE-T のスイッチング HUB1 と接続されている実サーバの Ethernet0 はクライアントへのサービス提供用で，スイッチング HUB2 と接続されている Ethernet1，Ethernet2 は NFS 接続やシステムの制御で使用する. 仮想サーバの Ethernet0 は実サーバの Ethernet0 とブリッジ接続し，仮想サーバの Ethernet1 は実サーバの Ethernet2 とブリッジ接続している．

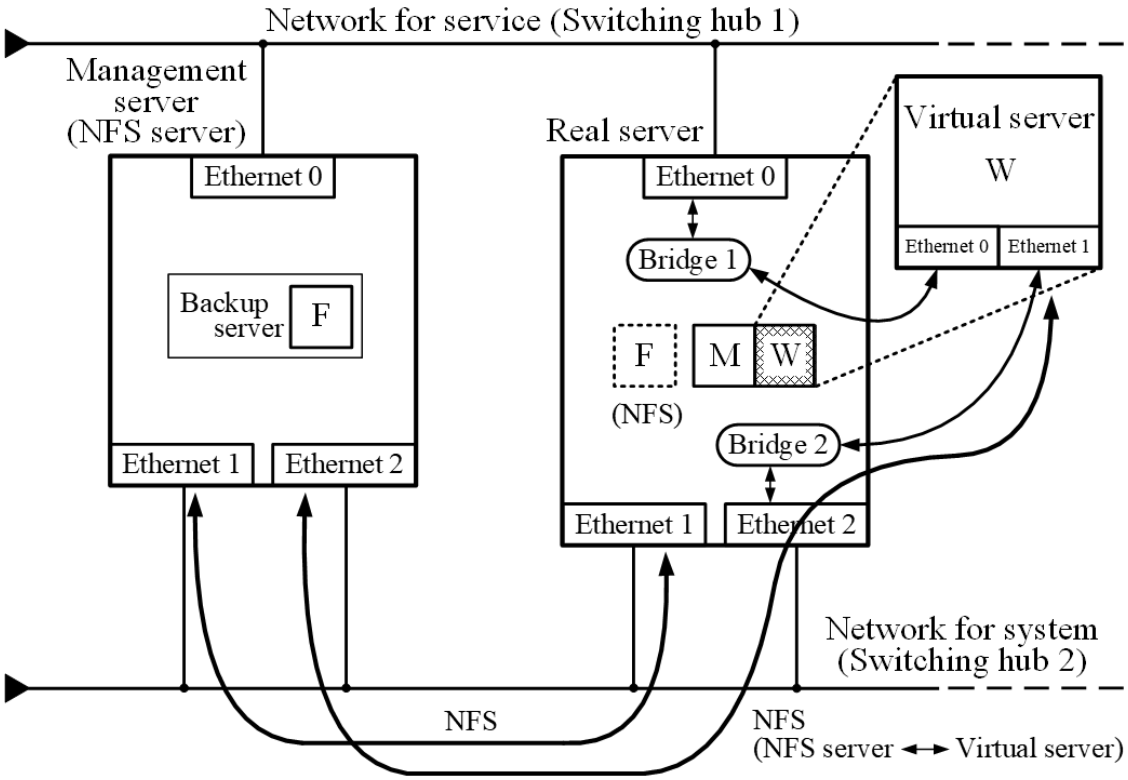


図 3.12　サーバネットワークの詳細構成

ここで，“M”，“W”と“F”は Mail，Web と FTP のサービスを提供する仮想サーバ用システムデータを示し，実サーバが通常起動する仮想サーバは“M”または“W”を使用して起動し，“F”はバックアップサーバを起動する場合に使用する．“F”は管理サーバ内に保存しているため，実サーバは NFS 接続を介して“F”を使用する．これは，いずれの実サーバでも“F”を使用したバックアップサーバの起動が可能であることを意味する．また，仮想サーバの Ethernet1 は実サーバの Ethernet2 を介して管理サーバと NFS 接続しており，サービスグループに対して冗長化構成している仮想サーバが同様のサービスを行うために用意している．ここでは PSS において，仮想サーバのサーバ負荷として Ethernet0 のデータ転送量を採用している．このため，Ethernet1 を NFS 接続やサーバ制御などに使用

するが，そのデータ転送量はサーバ負荷として積算されない.

3.5.2 実験システムの特性

表 3.4 に本実験システムの詳細特性を示す. 実験において複数種類のトラブルを再現する実サーバを RS1，仮想サーバを Mail1 とする．ここで示す時間は UNIX の time コマンドを使用して測定した値で，10 回行った実験結果の平均値とする.

RS1 の起動時間は 58.8 秒，再起動時間は 66.0 秒，サスペンド状態からの起動時間は 8.7 秒となった．RS1 から起動する仮想サーバ Mail1 の特性を“Virtual server (Local)”と示し，初回の起動は 34.7 秒，2 回目の起動は 24.0 秒となった．この時間の差は仮想サーバがソフトウエアで構成されるためディスクキャッシュが影響していると考えられる．再起動時間は 32.1 秒となった．バックアップサーバ用のシステムファイルは NFS 機能を有する管理サーバ上に保存されており，NFS 接続を通じて起動する．バックアップサーバの特性を“Backup server (NFS)”と示し，初回の起動時間は 45.8 秒，2 回目の起動時間は 36.1 秒となった．バックアップサーバも仮想サーバであることから，ディスクキャッシュが影響している．再起動時間は 39.9 秒となり，サスペンド状態からの起動時間は 9.1 秒となった．この結果より，提案システムではサーバ機能復旧時におけるバックアップサーバの起動時間を短縮するため，バックアップサーバは常にサスペンド状態のまま待機させる.

PSS における特性は，すべてのサービスグループが Moderate condition の場合，実サーバは 3 台起動しているため RSG の消費電力は 290W，すべてのサービスグループが Light condition の場合, RS2 と RS3 はサスペンド状態となり RS1 のみの起動状態となるため RSG の消費電力は 112W となり，39%の電力での運用が可能となる．電力測定に関しては，システムアートウエア社のワットアワーメータ SHW3A を使用して測定した．すべてのサービスグループが Moderate condition から Light condition に移行するためには, RS1 に FTP

のサービスを提供する仮想サーバ ftpb を起動し，ロードバランサを制御する．その後，RS2 と RS3 をサスペンド状態に変更する．すべてのサービスグループが Light condition から Moderate condition に移行するためには，サスペンド状態の RS2 と RS3 を起動し，ロードバランサを制御する．その後，RS1 に起動している仮想サーバ ftpb をサスペンド状態に変更する．状態の移行に必要となる時間は，すべてのサービスグループが Moderate condition から Light condition に移行するのに 20.7 秒, すべてのサービスグループが Light condition から Moderate condition に移行するのに 16.0 秒を要する．

表 3.4　実験用省電力かつ高可用性サーバシステムの詳細特性

Target server	**Real server:** RS1 / **Virtual server:** Mail 1				
Performances	**RS1**	**Virtual server (Local)**		**Backup server (NFS)**	
		First	Second	First	Second
Start time (s)	58.8	34.7	24.0	45.8	36.1
Restart time (s)	66.0	32.1		39.9	
Shutdown time (s)	12.5	10.4		10.3	
Time to suspend active server (s)	6.3			3.7	
Time to activate suspended server (s)	8.7			9.1	

PSS		
Condition	**Active real server (Number of active virtual servers)**	**Power consumption (W) (RSG)**
Moderate (All service groups)	RS1, 2, 3 (6)	290
Light (All service groups)	RS1 (3)	112
Condition	**Operation for condition change**	**Required time (s)**
Moderate → Light (All service groups)	Start virtual server → control load balancer → suspend two active real servers	20.7
Light → Moderate (All service groups)	Activate two suspended real servers → control load balancer → suspend virtual server	16.0

3.5.3 動作実験とその結果

実験ではサービスグループMailに所属する仮想サーバMail1のサービス提供プログラムまたはネットワークと実サーバであるRS1のネットワークに対してトラブルを再現し，提案システムにおける復旧時間およびクライアントからのアクセス状況を実測する．クライアントからのアクセスはロードバランサ経由で対象サーバにアクセスすることを意味する．各実測時間は，付録のF5に示す提案システムが出力するLogファイルからの測定方法を採用し，クライアントからのアクセス状況はクライアント上において，2章の図2.4で示したExpectプログラム言語を使用して1秒間隔で調査した結果を示す．それぞれの実測時間は実験を10回行った結果の平均値とする．表3.5にModerateまたはLight conditionにおいて，RS1やMail1にトラブルが発生した場合の復旧時間およびクライアントからのアクセス状況を示す．Heavy conditionに関してはModerate conditionと同様となるため省略する．実測時間としては，サーバ機能の復旧時間，管理対象サーバの復旧時間とクライアントからのアクセス切断時間の 3 種類を示す．サーバ機能の復旧時間とは，トラブルを発生したサーバの機能がバックアップサーバにより復旧されるまでの時間を示し，管理対象サーバの復旧時間とは，トラブルを起こした仮想サーバや実サーバが提案システムにより正常な状態に戻るまでの時間を示す．Moderate condition では各サービスグループ用の仮想サーバは冗長化されており，Light condition では冗長化されていない状態である．実験システムではHigh-availability regular modeにおける監視間隔時間を10秒とし，トラブルの再現はその開始時から5秒経過時に実行する．

サービストラブルに関しては，サービス提供プログラムpostfixの再起動で復旧する場合と復旧しない場合の 2 種類について検討する．サービス提供プログラムの再起動で復旧する場合におけるモードの移行は，High-availability regular mode → Service recovery mode (Service program restart) → High-availability regular mode となり，管理対象サー

バの復旧時間に Moderate condition で 4.3 秒，Light condition で 4.1 秒を要する．クライアントからのアクセス切断時間は Moderate condition ではサービスグループ Mail に所属する仮想サーバが冗長化されていることより 0.0 秒となり，冗長化されていない Light condition で 10.7 秒となった. サービス提供プログラムの再起動で復旧しない場合におけるモードの移行は，High-availability regular mode → Service recovery mode (Recovery by server restart) → Backup server mode → High-availability regular mode となり，サーバ機能復旧時間に Moderate condition では 13.6 秒，Light condition では 13.2 秒を要し，管理対象サーバの復旧時間に Moderate condition では 40.8 秒，Light condition では 40.1 秒を要する．クライアントからのアクセス切断時間は Moderate condition では冗長化構成より 0.0 秒となり，冗長化されていない Light condition では 21.6 秒となった．ここで，Light condition において，サービス提供プログラムの再起動で復旧しない場合におけるアクセス切断時間はサーバ機能復旧時間より約 8 秒長くなっているが，これは監視間隔時間内の 5 秒と実サーバおよび仮想サーバのネットワーク調査時間と仮想サーバのサービス調査時間経過後に復旧作業が開始されるためである.

仮想サーバのネットワークトラブルに関しては, Mail1 の意図的な再起動の場合とシャットダウンした場合の 2 種類について検討する．意図的な再起動とは，サーバの稼働状態が異常な場合にサーバ管理者がリセットボタンを押すなど，そのサーバを再起動することを想定している．Mail1 の意図的な再起動の場合におけるモードの移行は，High-availability regular mode → Virtual server network examination mode → Backup server mode → High-availability regular mode となり，サーバ機能復旧時間に Moderate condition では 13.7 秒，Light condition では 13.5 秒を要し，管理対象サーバの復旧時間に Moderate condition では 27.1 秒，Light condition では 27.4 秒を要する．クライアントからのアクセス切断時間は Moderate condition では冗長化構成より 0.0 秒となり，冗長化されていない

Light condition では 23.4 秒となった．Mail1 をシャットダウンした場合におけるモードの移行は，High-availability regular mode → Virtual server network examination mode (Restart of virtual server) → Backup server mode → High-availability regular mode となり，サーバ機能復旧時間に Moderate condition では 13.7 秒，Light condition では 13.1 秒を要し，管理対象サーバの復旧時間に Moderate condition では 80.5 秒，Light condition では 81.1 秒を要する．クライアントからのアクセス切断時間は Moderate condition では冗長化構成より 0.0 秒となり，冗長化されていない Light condition では 23.2 秒となった．ここで, Light condition におけるアクセス切断時間はサーバ機能復旧時間より約 10 秒長くなっているが，これは監視間隔時間内の 5 秒と実サーバおよび仮想サーバのネットワーク調査時間経過後に復旧作業が開始されるためである．

実サーバのネットワークトラブルに関しては，RS1 の意図的な再起動の場合とシャットダウンした場合の 2 種類について検討する．意図的な再起動の場合におけるモードの移行は，High-availability regular mode → Real server network examination mode → Backup server mode → High-availability regular mode となり，サーバ機能復旧時間に Moderate condition では 19.5 秒，Light condition では 12.3 秒を要し，管理対象サーバの復旧時間に Moderate condition では 110.3 秒，Light condition では 111.5 秒を要する．クライアントからのアクセス切断時間は Moderate condition では冗長化構成より 0.0 秒となり，冗長化されていない Light condition では 21.4 秒となった．シャットダウンした場合におけるモードの移行は，High-availability regular mode → Real server network examination mode (Restart of real server) → Backup server mode → High-availability regular mode となり，サーバ機能復旧時間に Moderate condition では 19.3 秒，Light condition では 12.1 秒を要し，管理対象サーバの復旧時間に Moderate condition では 213.6 秒，Light condition では 214.2 秒を要する．クライアントからのアクセス切断時間は Moderate condition では

冗長化構成より 0.0 秒となり，冗長化されていない Light condition では 21.1 秒となった．ここで，Light condition におけるアクセス切断時間はサーバ機能復旧時間より約 9 秒長くなっているが，これは監視間隔時間内の 5 秒と実サーバのネットワーク調査時間経過後に復旧作業が開始されるためである．

表 3.5　様々なトラブルに対する提案システムの復旧時間

Monitoring interval time: 10 s					
	Trouble list	Condition	Measured time (s)		
			Recovery of server function	Recovery of target server (Background job)	Access disconnection time (Client)
Service program	Service program stop (Recovery by service program restart)	Moderate		4.3	0.0
		Light		4.1	10.7
	Service program trouble (Recovery by server restart)	Moderate	13.6	40.8	0.0
		Light	13.2	40.1	21.6
Virtual server network	Intentional restart of Mail 1	Moderate	13.7	27.1	0.0
		Light	13.5	27.4	23.4
	Shutdown of Mail 1	Moderate	13.7	80.5	0.0
		Light	13.1	81.1	23.2
Real server network	Intentional restart of RS1	Moderate	19.5	110.3	0.0
		Light	12.3	111.5	21.4
	Shutdown of RS1	Moderate	19.3	213.6	0.0
		Light	12.1	214.2	21.1

いずれのトラブルにおいてもサーバの機能は復旧しており，Moderate および Light condition において仮想サーバのトラブルに関しては 14 秒未満，実サーバのトラブルに関しては 20 秒未満となった．管理対象サーバの復旧時間はサーバ機能の復旧時間に比べると長い時間を要していることがわかる．これは，トラブルを発生した仮想サーバや実サーバが再起動状態であるかの確認時間やシャットダウンしている場合には再起動させる処理が必要となるためである．Moderate condition では，いずれのトラブルもサービスを提供す

る仮想サーバが冗長化構成であるため，クライアントに対するサービス停止時間は 0.0 秒である．Light condition では省電力の効率を高くするため，サービスグループに対する仮想サーバが冗長化構成されていない．そのため，仮想サーバや実サーバにトラブルが発生すると最大約 24 秒のサービス停止時間が発生するが，サーバ負荷が低い状態であるため重大な問題にはならない．

提案方式は独立して動作可能な PSS と高可用性サーバシステムを交互に動作させることにより省電力かつ高可用性サーバシステムを実現する．そのため，省電力と高可用性を 1 つの制御プログラムにおいて実現する方式より，提案システムの制御プログラムは複雑にはならない．また，本実験では，PSS によって Moderate condition と Light condition を構成し，この 2 種類の状態においてクライアントにサービスを提供するサーバにネットワークまたはサービストラブルを再現した．その状態において提案システムは自動的にサーバ機能を復旧可能であることを示した．それにより，異種システム混合処理方式は省電力および高可用性サーバシステムを構築する上で実用可能であることが示された．

ここで，図 1.2 で示した稼働率による評価を表 3.6 に示す．冗長化を行っていないサーバでトラブルが発生した場合，その故障サーバ自体の復旧が求められる．そこで，管理対象サーバ自体を復旧する方式を従来方式として比較を行う．従来方式としての t_i には表 3.5 に示す管理対象サーバの復旧時間を，改善システムとしての t_{bi} にはサーバ機能の復旧時間を使用する．T_i には管理対象サーバの復旧時間における最大値を使用し，その値での従来方式の稼働率（Conventional sysytem）を 0.5 とする．また，提案方式による稼働率の改善（Improvement）を式(2.10)で算出する．

管理対象サーバが仮想サーバの場合を以下に示す．サービス提供プログラムの再起動で復旧しない場合のトラブルでは，Moderate condition において従来方式の稼働率は 0.84，提案方式の稼働率（Proposed system）は 0.95 となり 13.10%の改善，Light condition に

おいて従来方式の稼働率は 0.84，提案方式の稼働率は 0.94 となり 11.90%の改善がみられる．また，再起動した場合のトラブルでは，Moderate condition において従来方式の稼働率は 0.89，提案方式の稼働率は 0.94 となり 5.62%の改善，Light condition において従来方式の稼働率は 0.89，提案方式の稼働率は 0.94 となり 5.62%の改善がみられる．さらに，シャットダウンした場合のトラブルでは，Moderate condition において従来方式の稼働率は 0.73，提案方式の稼働率は 0.95 となり 30.14%の改善，Light condition において従来方式の稼働率は 0.73，提案方式の稼働率は 0.96 となり 31.51%の改善がみられる．

表 3.6　提案方式による稼働率の改善

Target server
VS: Virtual server
RS: Real server

Condition
MC: Moderate
LC: Light

St2: Service program trouble (Recovery by server restart)
Nt1: Intentional restart of target server
Nt2: Shutdown of target server

				Conventional system	Proposed system	Improvement (%)
Availability rate	MC	VS	St2	0.84	0.95	13.10
			Nt1	0.89	0.94	5.62
			Nt2	0.73	0.95	30.14
		RS	Nt1	0.66	0.93	40.91
			Nt2	0.50	0.95	90.00
		Average		0.72	0.94	30.56
	LC	VS	St2	0.84	0.94	11.90
			Nt1	0.89	0.94	5.62
			Nt2	0.73	0.96	31.51
		RS	Nt1	0.66	0.96	45.45
			Nt2	0.50	0.97	94.00
		Average		0.72	0.95	31.94

管理対象サーバが実サーバの場合を以下に示す．再起動した場合のトラブルでは，Moderate condition において従来方式の稼働率は 0.66，提案方式の稼働率は 0.93 となり 40.91%の改善，Light condition において従来方式の稼働率は 0.66，提案方式の稼働率は 0.96 となり 45.45%の改善がみられる．さらに，シャットダウンした場合のトラブルでは，Moderate condition において従来方式の稼働率は 0.50，提案方式の稼働率は 0.95 となり 90.00%の改善，Light condition において従来方式の稼働率は 0.50，提案方式の稼働率は 0.97 となり 94.00%の改善がみられる．提案方式は従来方式に比べ稼働率の面において非常に高い優位性が認められる．

図 3.11 に示した実験システムと同様に実サーバから起動する仮想サーバの台数を 2 台とした場合を想定し，表 3.7 に管理対象サーバ（仮想サーバ）の台数に応じた予備用実サーバの台数を示す．

表 3.7　予備用実サーバの台数比較

CS: Conventional system

	The number of target servers ($n \leqq 170$)					
	2	**4**	**6**	**・・・**	**2(n-1)**	**2n**
CS	1	2	3	・・・	n-1	n
Hybrid	2	2	2	・・・	2	2

実験システムは 1 台の実サーバおよびそれから起動している仮想サーバ 2 台が管理サーバに対してそれぞれ NFS 接続を行う．この管理サーバの仕様（OS 上で CPU コア数を 8

と認識）では付録 F7 に示すように NFS 接続を 256 台（32×8）までに設定するのが最適となるため，管理対象となる実サーバは最大で 85 台（256÷3）となり，仮想サーバは 170 台（85×2）までに設計することが望ましい．提案方式を“Hybrid”とし，バックアップサーバでのサーバ機能の復旧機能を持たない従来方式を“CS”として示す．

CS は管理対象サーバが 2 台ずつ増加すると，その予備用実サーバの台数も増加する．Hybrid は実サーバ故障時にバックアップサーバを他の実サーバ上に起動することで復旧処理を行うため，管理対象サーバの台数に応じて予備用実サーバの台数は増加しない．ただし，提案方式を導入している管理サーバが故障した場合，管理対象サーバの管理ができなくなるため，その管理サーバは冗長化する必要がある．そのため，Hybrid は管理対象サーバの台数に関係なく 2 台の状態を維持している．ここで，予備用実サーバの台数に比例して初期費用は高くなり，関連して消費電力や保守料金などの運用コストも台数に比例して高くなる．例えば，提案方式を導入する管理サーバを 2 台で 100 万円，予備用実サーバを 1 台 40 万円とし，管理対象サーバが 100 台（実サーバ 50 台）の場合には，提案方式は初期費用だけで 1,900 万円のコストを抑えることができ，さらに，台数に比例した運用コストも抑えることができる．以上の結果から，提案方式は従来方式に比べコスト面においても優位性が認められる．

3.6 まとめ

独立して動作可能な PSS と高可用性サーバシステムを交互に動作させることにより，両システムの機能を実現する異種システム混合処理方式を提案した．本方式において，2 つのシステムが編集可能な共有ファイルを使用することにより，誤動作を防ぐことが可能となることも示した．本章では，この 2 つのシステムを開発し，本方式を採用して省電力かつ高可用性サーバシステムを構築した．

PSS では，サービス提供サーバの負荷状況に応じて Moderate mode，Light mode と Heavy mode の 3 つのモードによりシステム構成の変更を行うことを示した．一般的にサーバはクライアントに対してファイルの提供や演算を行うことが多いことから，サーバ負荷としては，CPU の使用率とネットワークの転送バイト量を採用した．実測した負荷を Light load class，Moderate load class と Heavy load class の 3 種類に分類し，Moderate load class の場合は Moderate mode が処理を行い，Heavy load class の場合には Heavy mode が処理を行う．また，Light load class の場合には Light mode が処理を行い，実サーバをサスペンド状態に変更することで省電力化を実現した．すべてのサービスグループが Moderate condition の場合，サービスを提供するための仮想サーバを起動している実サーバの消費電力は合計 290W，Light condition では 112W となり，39%の電力での運用が可能となった．

高可用性サーバシステムでは，管理対象サーバの状況に応じて High-availability regular mode，Real server network examination mode，Virtual server network examination mode，Service recovery mode と Backup server mode の 5 つのモードによりトラブルが発生したサーバの対処を行い，Moderate condition と Light condition におけるサーバ稼働時の復旧方法の違いを示した．

実験システムを構築し，その構成および使用機器の仕様を示すとともに，実サーバおよび仮想サーバの起動時間を含む特性も示した．また，サーバ機能の復旧時間を短縮するために，サーバ機能を復旧するためのバックアップサーバは，常時，サスペンド状態で待機することとした．実験では，サーバトラブルとしてサービス提供プログラムとネットワークのトラブルを再現している．サービストラブルとしては，サービス提供プログラムの再起動により復旧する場合とその再起動では復旧せず，管理対象サーバを再起動することにより復旧する場合の 2 種類を，ネットワークトラブルとしては，管理対象サーバの再起動

およびシャットダウンの 2 種類を行った. すべてのサービスグループが Moderate condition のサーバ稼働時と Light condition のサーバ稼働時において，仮想サーバと実サーバに対してトラブルを再現し，サーバ機能の復旧時間，管理対象サーバの復旧時間とクライアントにおけるアクセス切断時間を実測した．仮想サーバの故障に関して，14 秒未満でサーバ機能を復旧可能となり，実サーバが故障したとしても，20 秒未満でサーバ機能を復旧可能であることを示した．また，従来方式との比較では，提案方式は稼働率において非常に高い優位性があり，コスト面においても高い優位性があることを示した.

第 4 章

MSBS と PSS の複合動作による省電力かつ高可用性サーバシステムの開発と稼働実験

4.1 まえがき

情報指向社会において，多くのセンサーや装置がインターネットに接続され，それらから得られる情報が新たな価値を生み出す Internet of Things（IoT）という概念が注目されている[73]．日本の総務省の報告では，IoT 機器は既に約 154 億個であり，2020 年までにその約 2 倍の 304 億個まで増大すると評価されている[74]．この IoT を支えるインターネットは社会経済活動の通信基盤として浸透し，社会インフラとして重要な役割を果たしている[75]. これら IoT 機器から取得するデータの保存や分析などを行うにはサーバシステムが重要な役割を果たしている．また，クラウドコンピューティングなどを支えるデータセンタにおけるサーバシステムのトラブルはクラウドプロバイダおよびそのユーザに大きな損失を与える[76]．そのため，サーバシステムの高可用性や省電力化は非常に重要な課題となる．

データセンタに焦点をあてた障害対策において多くの重要な研究が進められており，データセンタにおけるネットワーク遅延とパケット損失を推測するための Precision time protocol（精密時間プロトコル）の導入[77]，データセンタ装置の故障率を考慮した高可用性仮想インフラストラクチャ管理フレームワークの提案[78], データセンタにおけるハードウエア故障の情報収集およびその分析[79], 仮想マシン故障時にホットスペアノードを瞬時に立ち上げるためのネットワークコーディング[43], 仮想データセンタにおけるサーバやネットワークリソースの保障に関してハードウエア故障，サービス異常などの考慮[80]，障害

発生予測と自動化またはオペレータ動作を可能とする障害位置予測との統合法の提案[81]などが報告されている．

データセンタにおいてサービスを提供するシステムの基本単位はサーバシステムであり，そのシステムを構成する上で，高速化，高信頼性に加え，省電力化についても十分に検討していくことが必要である[29]．2.1 節で述べたように，IT デバイスにおける消費電力は 2006 年に比べ 2025 年では 9 倍に増加すると評価されている．そのため，ネットワークシステムの省電力化に関する種々の研究[82]-[84]が進められている．また，ネットワークシステムと比較すると非常に少ないようであるが，サーバシステムに焦点をあてた研究も進められており，サーバ移動サービスで利用される機器の消費電力を考慮したサーバ配置の決定方法の提案[85]，仮想サーバで実サーバのサーバ機能を復旧可能であることを示し，仮想サーバを使用して複数台の実サーバの機能をバックアップ可能なシステムを採用することにより，低コスト，省電力なサーバ管理システムの実現[40]や独立して動作可能な省電力サーバシステムと高可用性サーバシステムを交互に動作させる方式を採用した省電力かつ高可用性サーバシステムの開発[86]などが報告されている．

しかしながら，省電力に注目したサーバシステムの研究報告は見られるが，省電力かつ高可用性の機能を兼ねたサーバシステムに関する報告は非常に少なく，さらに，それらを実現する管理プログラムは複雑となる傾向にあるが，それを解決する方式に関する報告は殆ど見受けられない．

そこで，本論文では，省電力かつ高可用性サーバシステムの構築に関して，独立して動作可能な MSBS と PSS を複合動作させる方式を提案する．MSBS は，1 台の管理対象サーバのみを管理する DBSS を基本とし，それを多重起動することで構成される．通常，2 種類の管理プログラムを同時に動作させる場合，各制御における誤動作等を防ぐため，それぞれのプログラムは複雑となる．ここでは，それを解決するために 2 つのシステムの仲介

を行う SIF を提案し，導入する．PSS における管理プログラムは無編集とし，複合動作を実現するために DBSS のみ編集を行う．その編集に関して，SIF を導入することにより DBSS のプログラム構成がシンプルな構造を維持できることを示す．本方式を採用した実験システムを構築し，省電力かつ高可用性サーバシステムの性能を評価する．評価に関しては PSS によって構成される通常状態と省電力状態のサーバ構成において，管理対象サーバにサービス提供プログラムおよびネットワークに関するトラブルを再現し，サーバ機能の復旧状態を調査する．

4.2 提案システムの構成概要および MSBS と PSS の特徴

MSBS は 2 章で示したシステムで，高可用性サーバシステムの構築を実現する．この MSBS は実サーバを管理対象とし，管理対象サーバ故障時にはバックアップサーバ（仮想サーバ）を起動して，その故障サーバの機能を復旧する．ただし，そのバックアップサーバに関しては管理を行わない．また，管理サーバ上に起動したバックアップサーバは，故障サーバが復旧するまでその状態を維持する．本章で使用する MSBS には以下に示す変更を行っている．管理対象サーバを仮想サーバとし，故障サーバの機能を復旧するために起動するバックアップサーバに関しても管理を行う．また，仮想サーバシステムでは複数台の管理対象サーバにおける同時故障が予想される．そこで，管理サーバの負荷を軽減するために，管理サーバ上に起動したバックアップサーバは管理対象サーバを起動している実サーバへ Live migration による移動を行う．

PSS は 3 章で示したシステムで，省電力サーバシステムの構築を実現する．この PSS は同じサービスを提供する仮想サーバから構成されるサービスグループの負荷として CPU 使用率およびネットワークにおける送受信バイト量の 2 種類を測定し，その状況に応じて 3 種類のサーバ構成（Light condition，Moderate condition，Heavy condition）に変更する．

Light condition ではロードバランサの制御，バックアップサーバの起動および実サーバのサスペンド処理により省電力化を実現する．本章で使用する PSS は以下に示す変更を行っている．サービスグループから測定する負荷はメモリの使用率，CPU 使用率およびネットワークにおける送受信率の 3 種類としている．また，クライアントからアクセスするサーバの変更は，ロードバランサの制御ではなく仮想 IP アドレスの設定変更により行う．さらに，省電力状態のサーバ構成は，仮想 IP アドレスの設定，Live migration によるサーバ移動および実サーバのサスペンド処理により実現する．

4.2.1 提案システムの構成概要

図 4.1 に提案システムの構成概要を示す．本提案システムは，高可用性および負荷分散を行う上で一般的に使用されるロードバランサを導入したサーバシステムを基本とする．

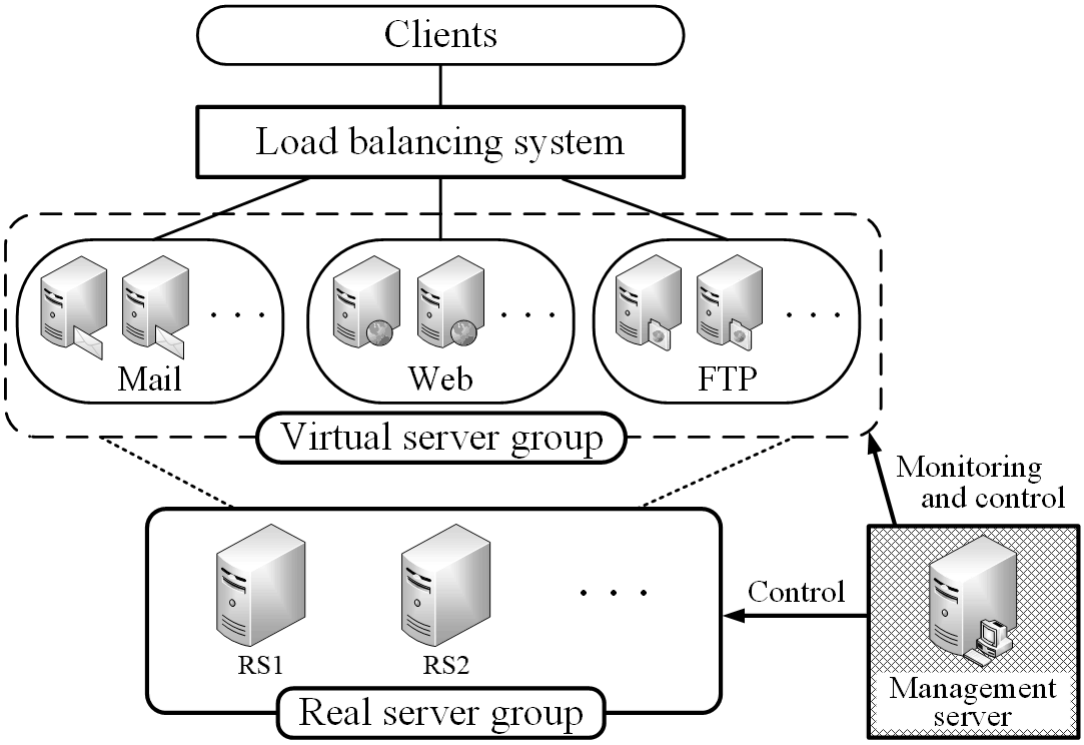


図 4.1　提案システムの構成概要

通常，ロードバランサは 1 台ではなく複数台設置し，冗長化構成したロードバランシングシステムとして採用される．Real Server Group（RSG）は仮想サーバを起動する実サーバ群を示し，Virtual Server Group（VSG）はクライアントにサービスを提供する仮想サーバ群を示す．ここではクライアントに Mail，Web や FTP のサービスを提供可能としている．1 種類のサービスに対して仮想サーバは冗長化構成され，そのサービス提供状態をここではサービスグループとする．クライアントからのアクセスはロードバランシングシステムによってサービスグループ毎の各仮想サーバにラウンドロビンアクセスされる．管理サーバは VSG に対して様々な調査および制御を行い，RSG に対しては制御を行う．本システムではこの管理サーバに MSBS と PSS をインストールする．

4.2.2 MSBS の動作概要

図 4.2 に本章における MSBS の動作概要を示す．MS は管理サーバ，M1，M2，W1 と W2 は管理対象の仮想サーバ，Mb と Wb は故障サーバの機能を復旧するためのバックアップサーバ, RS1 と RS2 は仮想サーバを起動する実サーバ, VIP は仮想 IP アドレスを示す．また，管理対象サーバ W2 を管理する DBSS を DBSS(W2)と示す．図に示すように MSBS は DBSS を複合動作させることで実現する．DBSS は，管理対象サーバのネットワークおよびサービス提供状態を調査する．異常を検出した場合にはバックアップサーバを管理サーバ上に起動後，管理対象サーバに設定していた仮想 IP アドレスをバックアップサーバに設定してそのサーバ機能を復旧する．通常，冗長化された管理対象サーバはロードバランシングシステムで管理可能となるが，管理対象サーバ故障時に冗長性が維持できなくなる可能性がある．そこで，冗長性を維持するために MSBS を併用する．

(a)では，DBSS(W2)が W2 の故障を検出し，バックアップサーバ Wb を管理サーバ上に起動後，仮想 IP アドレス VIP5 を設定して W2 の機能を復旧している．その後，管理サーバの負荷軽減のため Live migration により Wb を RS2 上に移動する．また，このシステムでは，DBSS (W2)が Wb に対して管理を行う DBSS (Wb)を自動起動する．その後，DBSS (W2)は終了する．

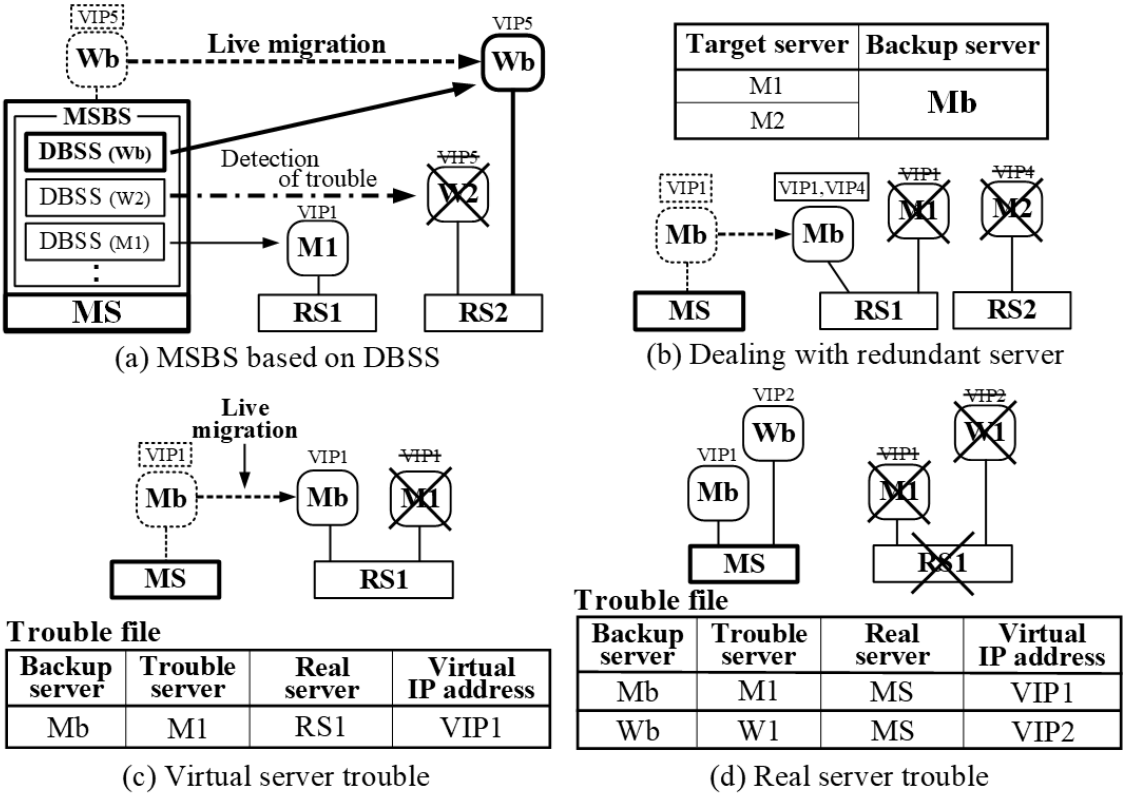


図 4.2　MSBS の特徴

基本的に DBSS は 1 台の管理対象サーバを管理するために，そのバックアップサーバを 1 台確保する．ここで，仮想サーバシステムの管理を考慮するとバックアップサーバの台数が非常に多くなることが懸念される．そこで，(b)に示すように，冗長化されている仮想サ

ーバの管理においては同じバックアップサーバを設定可能にすることで，その問題を解決している．(b)では仮想サーバ M1 と M2 のバックアップサーバは Mb と設定している．ここで，M1 のトラブルを検出した場合，Mb を起動し，VIP1 を設定して M1 の機能を復旧する．その後，M2 のトラブルを検出した場合はバックアップサーバである Mb が既に起動していることから，VIP4 を設定することで M2 の機能を復旧する．(c)に示すように，仮想サーバが故障した場合には，管理サーバ上に起動したバックアップサーバを故障した仮想サーバを起動していた実サーバに Live migration での移動を行う．また，仮想サーバを起動している実サーバが故障した場合には，(d)に示すように，各実サーバのリソースを考慮し，管理サーバ上に起動したバックアップサーバに対して Live migration での移動は行わない．各 DBSS がトラブルの状況を確認可能にするため，(c)と(d)に示す Trouble ファイルが設定されており，サーバ機能を復旧するために起動したバックアップサーバ名，故障した仮想サーバ名，バックアップサーバを起動している実サーバ名とそのバックアップサーバに設定している仮想 IP アドレスが記録されている．

DBSS は管理データを引数として実行する．管理データには管理に必要となる情報が記録されており，その詳細を図 4.3 に示す．データの 1 行目から 5 行目までは管理対象サーバの情報, 6 行目と 7 行目は管理対象サーバ故障時にその機能を復旧するために使用するバックアップサーバの情報，8 行目には管理対象サーバを起動している実サーバの IP アドレスが記録されている．具体的には 1 行目に管理対象サーバに設定する仮想 IP アドレス，2 行目にホスト名，3 行目に IP アドレス，4 行目にサービスグループ名，5 行目にはクライアントに提供しているサービス用のデーモン名が記録され, 6 行目と 7 行目にバックアップサーバのホスト名と IP アドレスが記録されている．ここで，5 行目のサービスデーモン名は 1 台のサーバが複数のサービスを提供する可能性があるため, スペース区切りで複数記述することができる．

管理対象サーバを M1 とすると，その管理データ名は M1.dat となる．そのデータを引数として管理プログラムを実行すると，M1 の監視および故障時の対処が可能となる．故障時の対処としては，管理データの 6 行目と 7 行目に記録されているバックアップサーバを起動し，1 行目に記録されている仮想 IP アドレスを設定することで管理対象サーバの機能を復旧する. 本システムで使用する MSBS はバックアップサーバに対しても管理を行うため，そのデータも必要となる．バックアップサーバのデータである Mb.dat を例とすると，仮想 IP アドレス，実サーバの IP アドレスは状況に応じて設定されるため記録されていない．また，ここでは Mb が故障した場合のバックアップサーバは設定していないため，6 行目と 7 行目には何も記録されていない.

Management data	M1.dat	Mb.dat
1st line: Virtual IP address (VIP)	172.21.14.101	—
2nd line: Host name	M1	Mb
3rd line: IP address (IPV)	172.21.14.1	172.21.14.51
4th line: Service group name	Mail	Mail
5th line: Service daemon name	dovecot postfix	dovecot postfix
6th line: Backup server host name	Mb	—
7th line: Backup server IP address (IPB)	172.21.14.51	—
8th line: Real server IP address	172.21.14.201	—

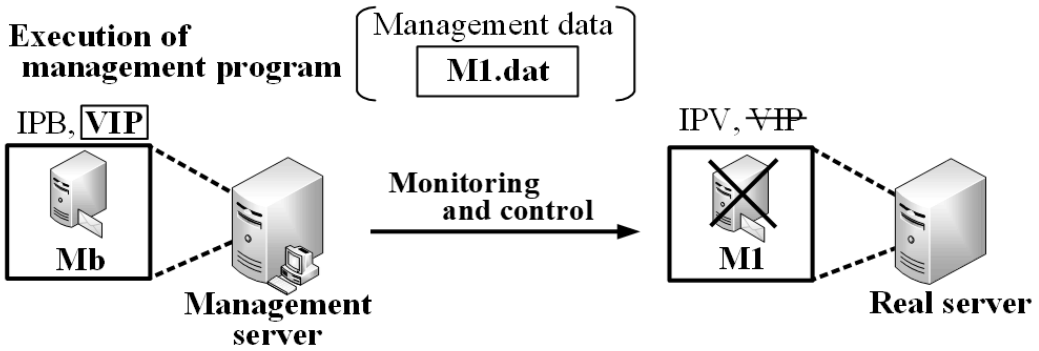


図 4.3　管理データの詳細

図では管理サーバ上で M1.dat を引数として DBSS が実行され，その内容をもとに M1 の管理を行う．もし，M1 がトラブルを発生した場合には，管理サーバが Mb を起動後，仮想 IP アドレス VIP を Mb に設定することで M1 の機能を復旧する. この管理データを引数として実行するサーバ管理方法により，管理プログラムを管理対象サーバに応じて変更する必要がなくなる.

4.2.3 PSS の動作フローおよびその詳細

図 4.4 に本章における PSS の概要およびその動作フローを示す．PSS において制御の対象となる仮想サーバは M1, M2, W1, W2, F1 と F2 で, それらを起動する実サーバは RS1, RS2 と RS3 とする．また，LB はロードバランサを示し，VIP は仮想 IP アドレスを示す．M1 と M2 はサービスグループ Mail，W1 と W2 はサービスグループ Web，F1 と F2 はサービスグループ FTP に所属し，各サービスグループにおける仮想サーバは冗長化構成されている．PSS は各サービスグループに所属する仮想サーバの負荷（メモリの使用率，CPU 使用率およびネットワークにおける送受信率）を測定し, その負荷から Moderate load class または Light load class を判定する．その Load class に応じて Moderate condition（通常状態）または Light condition（省電力状態）にサーバ構成を変更する.

図の例ではすべてのサービスグループが同じ Load class となっており，Moderate load class と判断された場合，サーバシステム起動時と同様の構成となり，クライアントからサーバへのアクセスはロードバランサによって各 VIP に対してラウンドロビンアクセスされる．Light load class と判断された場合，ここでは RS1 が Base server となっているため，M2 と W2 の仮想 IP アドレスは M1 と W1 にそれぞれ設定される．また，F2 の仮想 IP アドレスが設定された F1 は RS2 から RS1 に Live migration により移動される．その後，RS2 と RS3 はサスペンド状態となる．クライアントからのアクセスはロードバランサによ

って各 VIP に対してラウンドロビンアクセスされるが，実際には，各サービスグループにおいて 1 台の仮想サーバのみにアクセスされる．図に示す実サーバの稼働状態から，この Light condition における電力消費は Moderate condition の約 3 分の 1 となる．

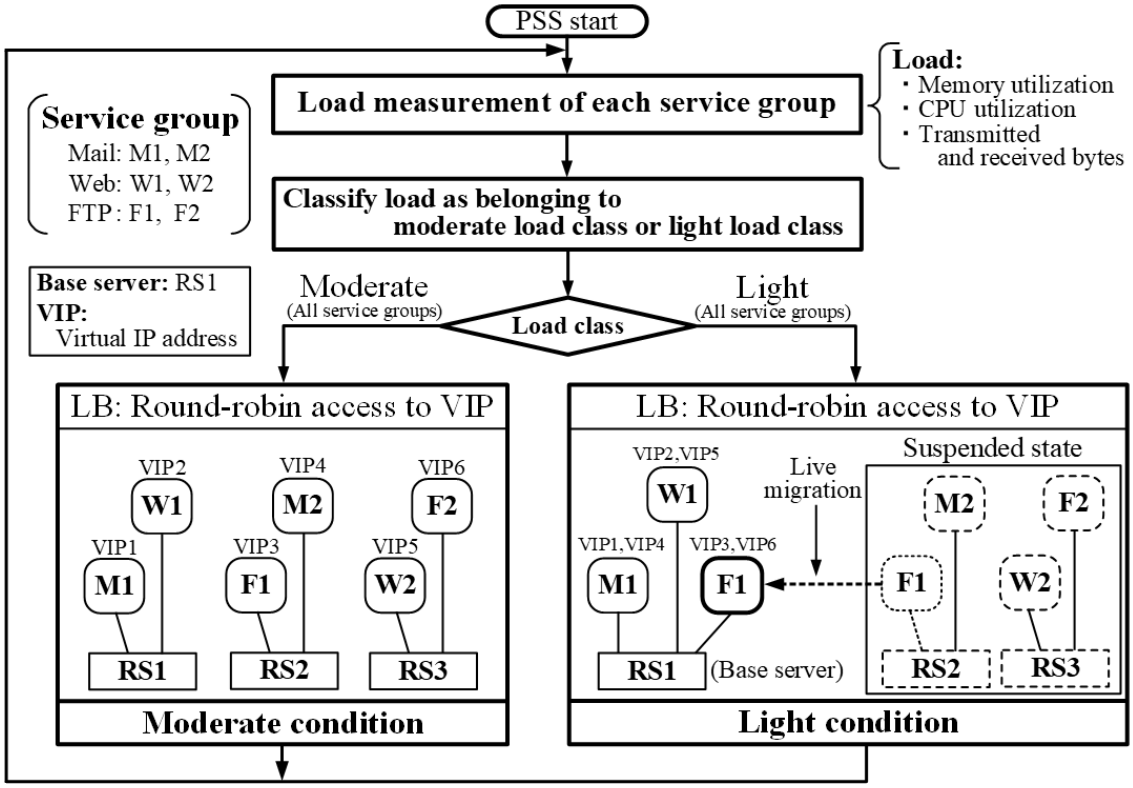


図 4.4　PSS の構成

測定した負荷の算出方法，Load class の分類およびサーバシステムの構成変更の判断について図 4.5 に示す．負荷の算出方法に関してはサービスグループ Web を例とする．メモリ負荷では，UNIX の free コマンドを使用して測定を行う．*M*use はメモリ使用量とし，*M*max は最大メモリ量とする．これらは各対象サーバにおいて測定される．サービスグループにおけるメモリ負荷は式(4.1)で算出できる．

$$Lm\ [\%] = \frac{M_{use\_w1} + M_{use\_w2}}{M_{max\_w1} + M_{max\_w2}} \times 100 \quad (4.1)$$

CPU 負荷では，UNIX の sar コマンドを使用して測定を行う．Cuse は CPU 使用率とし Ccor は CPU のコア数とする．これらは各対象サーバにおいて測定される．サービスグループにおける CPU 負荷は式(4.2)で算出できる．

$$Lc\ [\%] = \frac{C_{use\_w1} \times C_{cor\_w1} + C_{use\_w2} \times C_{cor\_w2}}{C_{cor\_w1} + C_{cor\_w2}} \quad (4.2)$$

ネットワーク負荷では，sar コマンドを使用して測定を行う．Ntrb は各対象サーバの転送および受信バイト量とし，Nrat は送受信の定格転送バイト量とする．ここで，1,000Mbps のルータを例とすると，定格転送量は 1 秒当たり 128×10^3KB の転送が可能なため，送受信の合計から Nrat は 256×10^3KB となる．サービスグループにおけるネットワーク負荷は式(4.3)で算出できる．

$$Ln\ [\%] = \frac{N_{trb\_w1} + N_{trb\_w2}}{N_{rat\_w1} + N_{rat\_w2}} \times 100 \quad (4.3)$$

各サービスグループの負荷から Load class への分類を行う．2 種類の Load class に分類するため，図に示す 2 種類のしきい値（Moderate load，Light load）を設定している．(a)から(d)の矢印は負荷の変動を示す．黒丸が初期の状態を示し，矢印の先が最終の状態を示す．領域 ML は初期の状態を維持する領域となっており，これはしきい値近辺での変動による影響を低くするために設定している．(a)と(d)の状態は Moderate load class，(b)と(c)の状態は Light load class となる．サービスグループの負荷は 3 種類の Load class において 1 つでも Moderate load class の場合には Moderate load class, すべてが Light load class の場合のみ Light load class と判断する．

図 4.4 で示したように PSS は分類される Load class に応じてサーバシステムの構成を変更する．サーバシステムにおける負荷状態は様々である．そのため，負荷変動に応じた構成変更に関しては，サーバシステムの使用環境を考慮して決定することが望ましい．また，負荷の変動による頻繁なサーバシステムの構成変更を防ぐために，現時点の状態とは異な

る Load class の判断が連続で複数回発生した場合にサーバシステムの構成変更を行う．図では Moderate condition において Light load class との判断が連続複数回行われており，時刻 t_{i+7} においてサーバ構成が Light condition に変更されている．

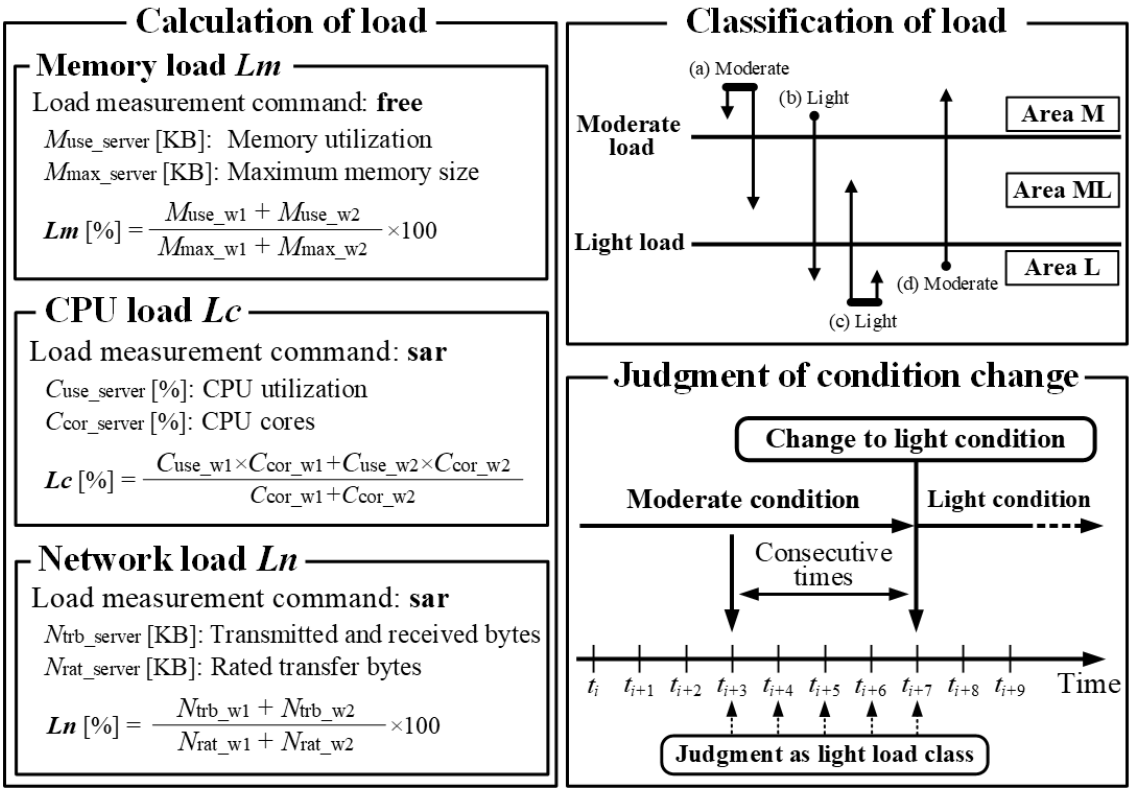


図 4.5　負荷に応じたサーバシステムにおける構成変更の判断

表 4.1 に PSS の構成ファイルの詳細を示す．Operation ファイルは起動している仮想サーバの詳細情報を記録したファイルで，仮想サーバが所属するサービスグループ名，仮想サーバを起動する実サーバの IP アドレス，仮想サーバに設定する仮想 IP アドレス，ホスト名，IP アドレスとクライアントに提供するサービスのデーモン名が記録されている．Suspend ファイルはサーバシステムが構成変更を行う場合に使用するファイルで，サスペンド対象となった仮想サーバの情報を Operation ファイルから Suspend ファイルに移動す

る．その仮想サーバがサスペンド対象から解除された場合には，その仮想サーバの情報を Suspend ファイルから Operation ファイルに戻す.

表 4.1　PSS の構成ファイル

Operation file

Service group	Real: IP address	Virtual IP address	Host name	Virtual: IP address	Service daemon
Mail	172.21.14.201	172.21.14.101	M1	172.21.14.1	dovecot postfix
Web	172.21.14.201	172.21.14.102	W1	172.21.14.2	httpd
FTP	172.21.14.202	172.21.14.103	F1	172.21.14.3	vsftpd
Mail	172.21.14.202	172.21.14.104	M2	172.21.14.4	dovecot postfix
⋮					

Suspend file, Live migration file

Service group	Real: IP address	Virtual IP address	Host name	Virtual: IP address	Service daemon

RSG file

Real: IP address	MAC address
172.21.14.201	00:00:00:00:00:01
172.21.14.202	00:00:00:00:00:02
⋮	

State file

Service group	Condition	Base server Real: IP address
Mail	Moderate	172.21.14.201
Web	Light	172.21.14.201
⋮		

また，Light condition への移行時に Live migration が行われた場合，その対象となる仮想サーバの情報は Live migration ファイルにコピーされ, Operation ファイル内において，その仮想サーバを起動する実サーバの IP アドレスが変更される．Moderate condition に戻る場合には，Live migration ファイルの内容が Operation ファイルに戻される．RSG ファイルは仮想サーバを起動する実サーバに関する情報が記録されたファイルで，その IP アドレスと MAC アドレスが記録されている．ここで，MAC アドレスはサスペンド状態から復

帰させるために使用される．State ファイルは各サービスグループの状態を示すファイルで，サービスグループ名，そのグループの状態（Moderate，Light）や省電力時にサービスグループ毎の仮想サーバをどの実サーバに集約するのかを示す Base server の IP アドレスが記録されている．

4.3. 複合動作の問題点とその解決方法および SIF の特徴

通常，2 種類の管理プログラムを同時に動作させる場合，各制御における誤動作等を防ぐため，それぞれの管理プログラムは複雑となる．ここでは PSS における管理プログラムは無編集とし，複合動作を実現するために MSBS を構成する DBSS における管理プログラムのみ編集を行う．

4.3.1 複合動作における問題点とその解決方法

図 4.6 に MSBS と PSS を複合動作した場合の問題点およびその解決方法を示す．M1，M2，W1，W2，F1 と F2 は管理対象となる仮想サーバ，Fb はバックアップサーバ，RS1 と RS2 は仮想サーバを起動する実サーバ，MS は管理サーバを示し，VIP は仮想 IP アドレスを示す．

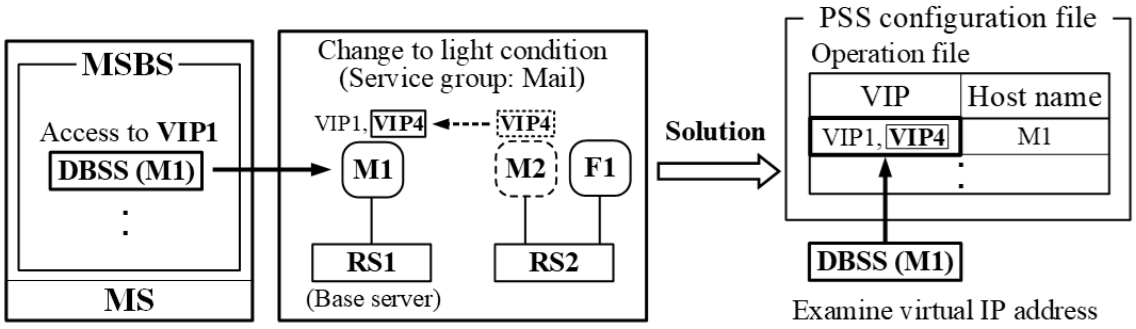
(a)に PSS によって Light condition に変更された場合における仮想 IP アドレスの管理に関する問題点およびその解決方法を示す．ここでは DBSS が M1 を監視しており，VIP1 へのアクセスを確認している．PSS によってサービスグループ Mail が Light condition に変更された場合，M2 に設定されていた VIP4 が M1 に設定される．DBSS はこの設定変更を確認することはできない. この対策として, DBSS が PSS の構成ファイルである Operation ファイルに記録されている M1 のデータを確認することが必要となる．

(b)に PSS によって Light condition に変更された場合における DBSS のトラブル誤検出

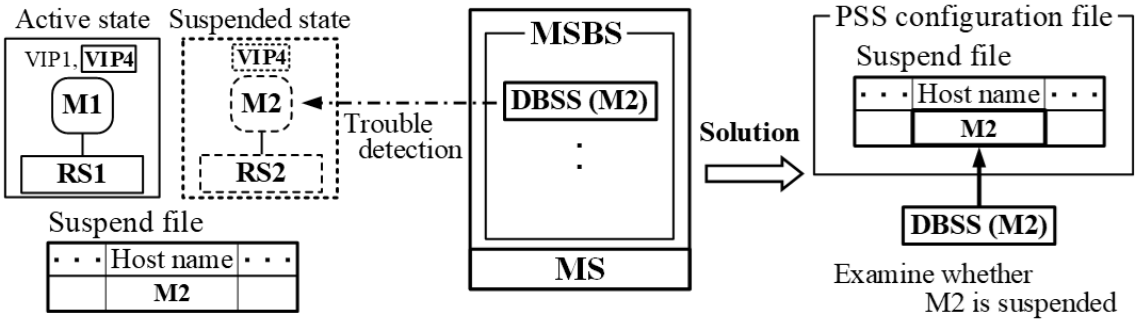
に関する問題およびその解決方法を示す．ここでは DBSS が M2 を監視している状態で，PSS が Light condition に制御するため M2 に設定していた VIP4 を M1 に設定後，M2 を起動している RS2 をサスペンドしている．このため，DBSS は M2 をトラブル状態と判断し，復旧処理を開始する．この対策としては DBSS が PSS の構成ファイルである Suspend ファイルに記録されている M2 のデータを確認することが必要となる．

(c)にクライアントにサービスを提供する仮想サーバが故障した場合に PSS がその故障した仮想サーバが所属するサービスグループの負荷測定が困難となる問題点およびその解決方法を示す．PSS は Operation ファイルに記録されている仮想サーバの負荷測定を行い，それをサービスグループ毎にまとめる．その結果をもとに適切なサーバ構成に制御する．ここで，サービスグループ Mail に所属する M1 が故障した場合，PSS はそのグループの負荷測定ができなくなる．この対策として，DBSS が M1 の故障を検出後，バックアップサーバ Mb を起動し，M1 のサーバ機能を復旧する．その後，PSS の構成ファイルである Operation ファイルの M1 のデータ行を削除し, バックアップサーバ Mb のデータ行を追加することが必要となる．これにより PSS がサービスグループ Mail の負荷測定が可能となる. また, M1 が復旧した場合には, Operation ファイルにおける Mb のデータ行を削除し，M1 のデータ行を追加する．

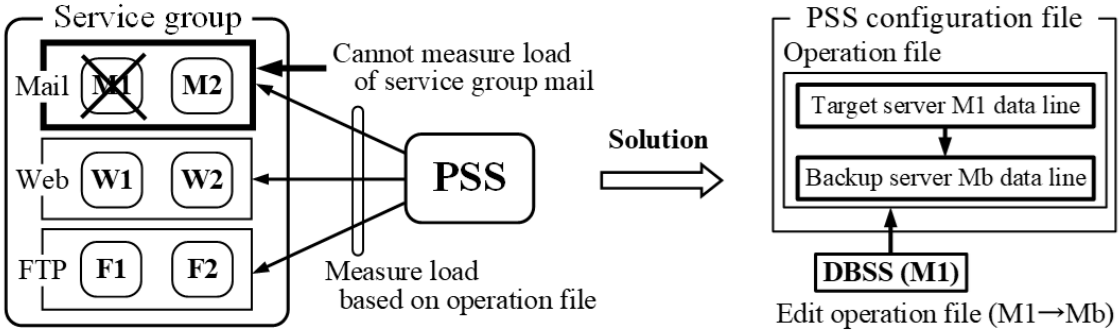
(d)に他の実サーバから Live migration により移動してきたサーバに関する問題点およびその解決方法を示す．PSS によってサービスグループ FTP が Light condition となった場合，RS2 上に起動している F1 は Live migration により RS1 上に移動され，PSS の構成ファイルである Live migration ファイルに F1 の情報が記録される．また，Operation ファイルにおいて F1 のデータ行における実サーバの IP アドレスが RS2 から RS1 に変更される．この状態で F1 が故障した場合，バックアップサーバ Fb が管理サーバ上に起動され，Live migration により RS1 上に移動される．



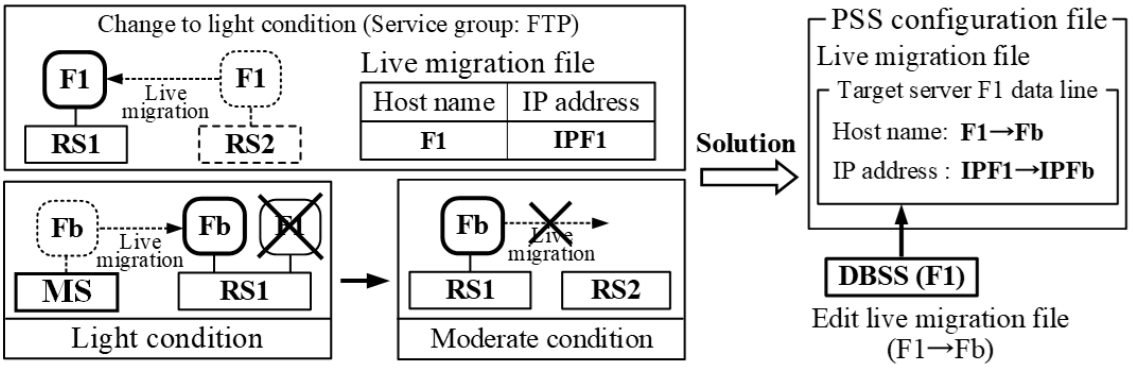
(a) Problem of virtual IP address management



(b) Malfunction of trouble detection



(c) Cannot measure load due to target server trouble



(d) Trouble of server moved by live migration

図 4.6　MSBS と PSS における複合動作時の問題点およびその解決方法

ここで，Light condition から Moderate condition に戻す場合，Fb の情報が Live migration ファイルに記録されていないため，RS2 に Live migration により戻すことができない．この対策として，バックアップサーバを起動してサーバ機能を復旧した場合には，Live migration ファイルを確認し，故障サーバのデータが記録されていた場合には，そのデータ行のホスト名およびその IP アドレスのみをバックアップサーバのデータに変更することが必要となる．

4.3.2 SIF の機能とその特徴

DBSS と PSS の複合動作を実現するため，DBSS において PSS を制御する処理の追加を行う．図 4.7 に DBSS の動作フローチャートを示し，PSS の制御処理を追加するタイミングについて示す．DBSS は 1 台の管理対象サーバを管理するシステムで，その管理プログラムは非常に単純な構造となっている．DBSS における復旧処理の動作フローを以下に示す．管理対象サーバの故障を検出した場合，その再確認後，バックグラウンドジョブで管理対象サーバの復旧処理を行うとともにバックアップサーバを起動し，管理対象サーバの機能を復旧する．バックグラウンドジョブでは 2 種類の復旧処理を行う．サービストラブルの場合，管理対象サーバの再起動を行う．ネットワークトラブルの場合，管理対象サーバが再起動状態であるかの確認を行い，再起動状態でないと判断した場合には管理対象サーバの強制起動を試みる．その後，バックグラウンドジョブによる管理対象サーバにおける復旧処理の結果に応じて，管理対象サーバの監視状態に戻るか，管理プログラムを停止する．

ここで，DBSS に追加する処理のタイミングを点線の矢印（A1～A5）で示す．DBSS において管理対象サーバを監視する直前に点線 A1 の処理を行う．ここでは，管理対象サーバがサスペンド状態であるかの確認と管理対象サーバに設定されている仮想 IP アドレスの調

査を行う. DBSS が管理対象サーバのトラブルを検出後, 点線 A2 の処理を行う. ここでは, 管理対象サーバがサスペンド状態であるか確認を行う．DBSS がその故障サーバのトラブルを再確認後，点線 A3 の処理を行う．ここでは，管理対象サーバが Live migration によって実サーバの移動を行っている可能性があるため，その管理対象サーバを起動している実サーバの IP アドレスの調査を行う．この調査は，バックグラウンドジョブで行う管理対象サーバの復旧処理において，そのサーバの再起動処理を行う場合に必要となる．DBSS がサーバ機能の復旧処理を行う場合，点線 A4 の処理を行う．ここでは，バックアップサーバを起動して管理対象サーバの機能を復旧するため, Operation ファイル内の管理対象サーバのデータ行を削除し，バックアップサーバのデータ行を追加する．これにより，PSS は故障サーバが所属するサービスグループの負荷測定が可能となる.

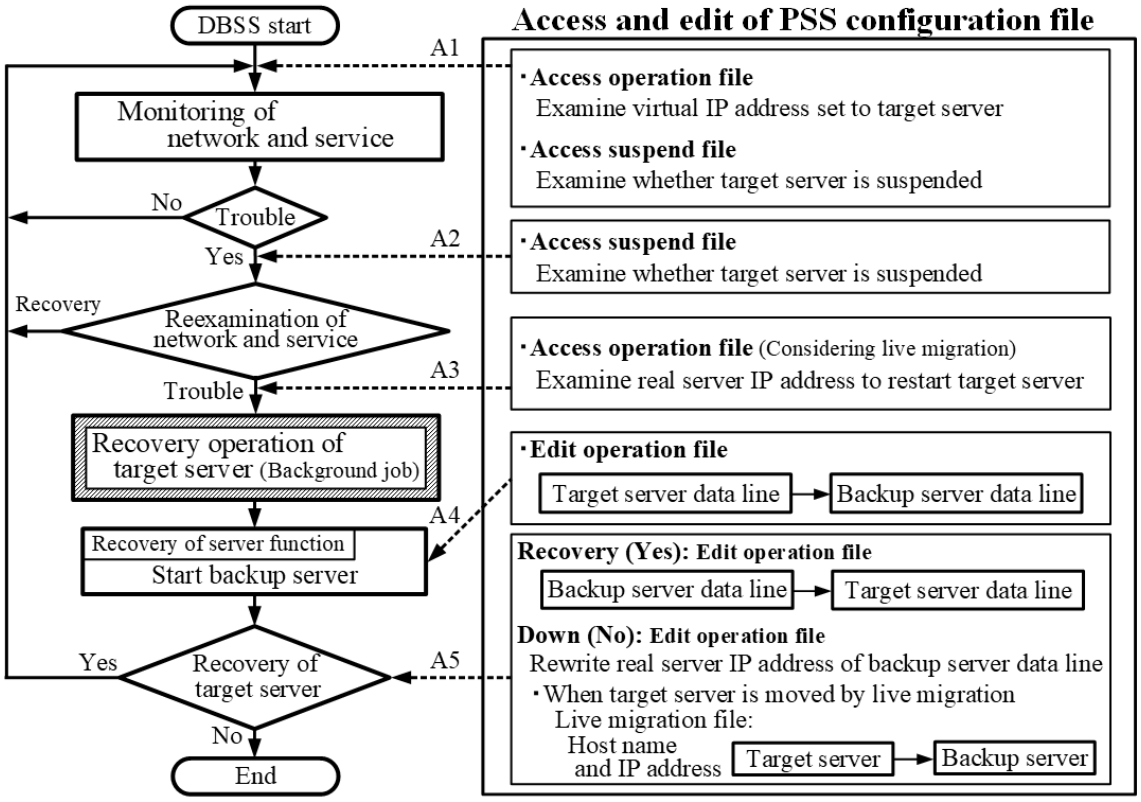


図 4.7　PSS との複合動作における DBSS に対する処理の追加

バックグラウンドジョブによる管理対象サーバの復旧処理の結果に応じて，点線 A5 の処理を行う．ここでは，管理対象サーバが復旧した場合，その結果を PSS に伝えるため，Operation ファイル内のバックアップサーバのデータ行を削除し，管理対象サーバのデータ行を戻す．復旧不可能と判断された場合，管理サーバ上に起動したバックアップサーバを Live migration により移動するため，Operation ファイルに記録されているバックアップサーバのデータ行における実サーバの IP アドレスを移動先の実サーバに変更する．ここで，管理対象サーバが Live migration により移動されたサーバの場合，Live migration ファイルに記録されている管理対象サーバのデータ行におけるホスト名とその IP アドレスをバックアップサーバのデータに変更する．

DBSS は 1 台の管理対象サーバのみを管理するプログラムであり，非常に単純なプログラム構成となっている．しかし，図 4.7 に示した PSS 制御用の処理を追加した場合，DBSS におけるプログラム構成が複雑になる．そこで，PSS との複合動作において，DBSS における管理プログラムのシンプルな構成を維持するために SIF 方式を提案する．図 4.8 に SIF の構成および動作概要を示す．

DBSS 起動時にそれに対応した SIF を自動起動し，SIF は DBSS がプロセス間通信によって送信するシグナルに応じた PSS へのアクセスおよび制御を行う．また，必要に応じて結果を DBSS に返答する．ここで，SIF は DBSS のシグナルを割り込み処理で受け取るため，高速な対処を実現している．

M1 を管理する DBSS を対象とし，SIF の一部動作を示す．DBSS が Signal1 およびデータとしてホスト名 M1 を SIF に送信した場合，SIF は PSS の Operation ファイルと Suspend ファイルにアクセスし，M1 における仮想 IP アドレスの情報および M1 がサスペンド状態か確認し，その結果を DBSS に返答する．また，DBSS が Signal2 およびデータとしてホ

スト名 M1 と Mb を SIF に送信した場合，SIF は Operation ファイル内の，故障した M1 のデータ行を削除後，Mb のデータ行を追加する．ここでは DBSS に対する返答が不必要なため，DBSS は Signal2 を送信後，直ちに次の処理を実行することができる．SIF 方式により DBSS における管理プログラムにはシグナル送受信部分の追加のみで PSS との複合動作が可能となる．そのため，DBSS のプログラム構成はシンプルな状態を維持することができる．

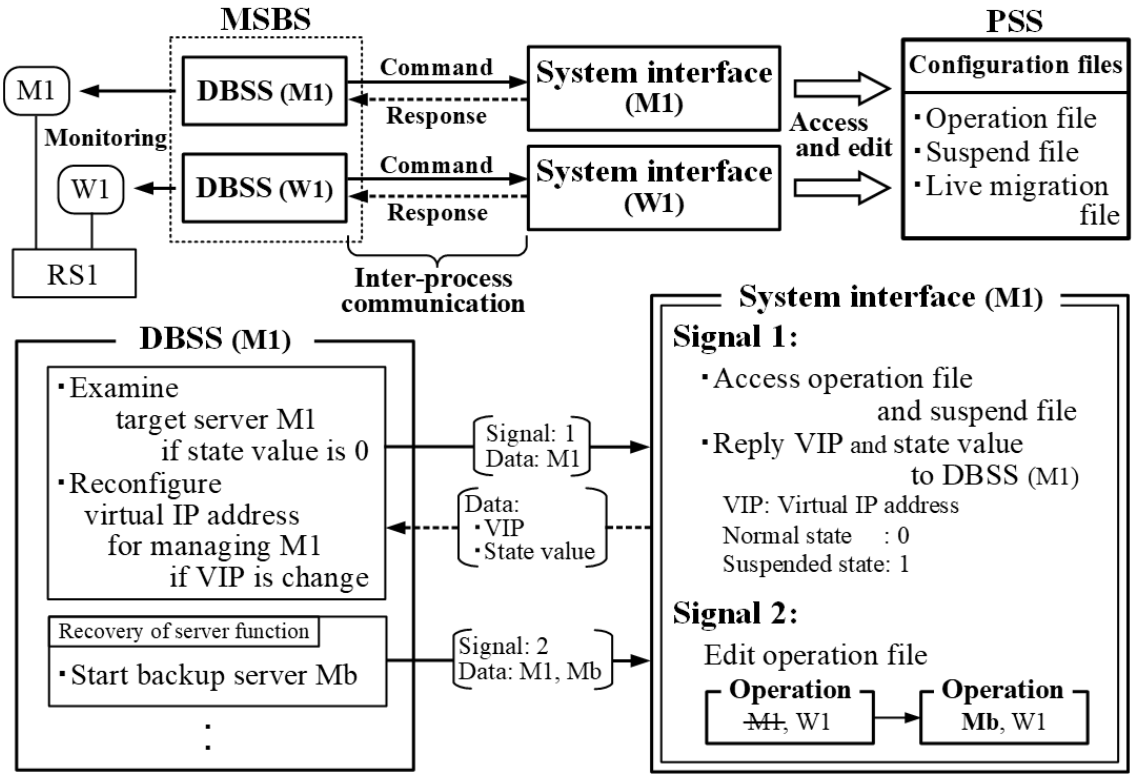


図 4.8　SIF の動作概要

4.4　省電力かつ高可用性サーバシステムの稼働実験

4.4.1　実験システムの構成および仕様

図 4.9 に実験サーバシステムの構成を示す．実験システムは管理サーバが 1 台，仮想サーバを起動する実サーバ（RS1，RS2，RS3）が 3 台，ロードバランサが 1 台，クライアントが 1 台，1000BASE-T のスイッチング HUB が 1 台と 100BASE-TX のスイッチング HUB が 1 台から構成される．管理サーバには MSBS と PSS がインストールされ，サーバシステムの管理と制御を行う．また，管理サーバはファイルサーバの機能を有している．一般的にサーバ間のネットワーク転送速度に対してクライアントからの転送速度は 10 分の 1 程度となるため，100BASE-TX のスイッチング HUB を使用している．

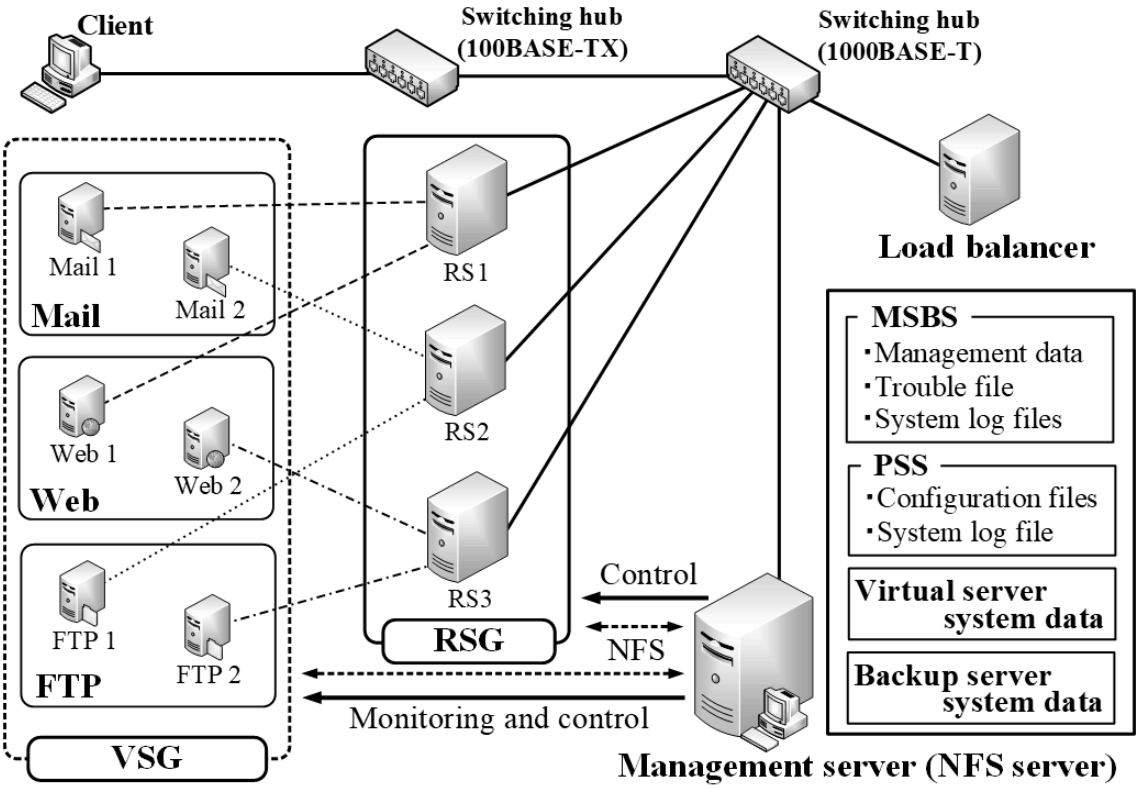


図 4.9　実験システムの構成

VSG はクライアントにサービスを提供する仮想サーバ群で，RSG は VSG を起動する実サーバ群を示す．仮想サーバはクライアントに対して Mail，Web と FTP のサービスを提供可能としており，サービスグループ Mail, Web と FTP を展開し，Mail サーバ 2 台（Mail1, Mail2），Web サーバ 2 台（Web1，Web2）と FTP サーバ 2 台（FTP1，FTP2）はそれぞれ冗長化構成されている．本実験システムでは Mail サーバは SMTP, Web サーバは HTTP, FTP サーバは FTP のサービスをクライアントに提供する．すべての仮想サーバおよびバックアップサーバを起動するためのシステムデータは，Live migration を可能とするため管理サーバに保存している．冗長化構成となっている仮想サーバはクライアントに同様のサービスを提供するため，また，各実サーバは管理サーバ上に保存されているシステムデータを使用して仮想サーバやバックアップサーバを起動するために管理サーバと NFS 接続している．

表 4.2 に実験システムにおける管理サーバ，実サーバ，仮想サーバ，バックアップサーバおよびロードバランサの仕様を示す．すべてのサーバにはサーバ用の OS として採用されることが多い CentOS の 64bit 版をインストールしている．管理サーバは，MSBS および PSS における管理プログラムを動作させるとともに NFS の機能を有することから，CPU として Intel Core i7-930 を搭載し，物理メモリは 8,192MB とする．また，仮想化ソフトウエアとして Linux で標準サポートしている KVM を採用している．各実サーバは同様の仕様で，CPU は Intel Core i7-960 を搭載し，物理メモリは 8,192MB とし，仮想化ソフトウエアとして KVM を採用する．各実サーバはメモリを有効活用するため CUI 環境で稼働させている．ロードバランサはソフトウエア処理方式で，UltraMonkey-L7 をインストールしている．仮想サーバやバックアップサーバには，CPU 数を 1，メモリを 2,048MB，サービス提供プログラムとして Mail サーバには postfix，Web サーバには httpd，FTP サーバには vsftpd を採用している．

表 4.2　実験システムの仕様

Specifications	Management server (MSBS, PSS)	Real server (RS1, RS2, RS3)
CPU	Intel Core i7-930 2.80 GHz (TB: 3.06 GHz)	Intel Core i7-960 3.20 GHz (TB: 3.46 GHz)
Memory	8,192 MB	8,192 MB
Hard disk	SATA 2 (7,200 rpm)	
System software	KVM 1.5.3 nfsd	KVM 1.5.3
OS	CentOS 7.4 (64-bit)	

Specifications	Virtual server / Backup server
CPU	1
Memory	2,048 MB
System software	postfix, httpd, vsftpd
OS	CentOS 7.4 (64-bit)

Specifications	Load balancer
CPU	Intel Core i3-530 2.92 GHz
Memory	4,096 MB
System software	UltraMonkey-L7
OS	CentOS 6.9 (64-bit)

4.4.2　実験システムの特性

実験システムの特性を表 4.3 に示す．実サーバ，仮想サーバとバックアップサーバの各特性は UNIX の time コマンドを使用し，測定を 10 回行った結果の平均値とする．実サーバの起動時間は 46.21 秒で，サスペンド状態からの起動時間は 8.09 秒となった．“Virtual server (NFS)”は仮想サーバのシステムデータを保存している管理サーバとの NFS 経由によって実サーバから起動した仮想サーバの特性を示し，初回起動時間は 58.37 秒で 2 回目の起動時間は 40.56 秒となった．この時間差は仮想サーバがソフトウエアであるためコンピュータシステムのディスクキャッシュが影響したと考えられる．“Backup server (Local)”は管理サーバ上に起動するバックアップサーバの特性を示し，初回起動時間は 30.38 秒で 2

回目の起動時間は 13.54 秒となった．また，サスペンド状態のバックアップサーバを起動するために要する時間は 1.14 秒となった．この結果より，バックアップサーバの起動時間を短縮するため，バックアップサーバは常にサスペンド状態のまま待機させる．

PSS において，すべてのサービスグループが Moderate condition の場合，3 台の実サーバが起動するので，RSG の消費電力は 263W となり，すべてのサービスグループが Light condition の場合，1 台の実サーバが起動し，2 台の実サーバがサスペンド状態となるので，RSG の消費電力は 93W となり，35%の電力での運用が可能となる．また，すべてのサービスグループが Moderate condition から Light condition に移行した場合，その移行時間は 6.54 秒となり，すべてのサービスグループが Light condition から Moderate condition に移行した場合，その移行時間は 14.66 秒となった．

表 4.3　実験システムの特性

Performances	**Real server**	**Virtual server (NFS)**		**Backup server (Local)**	
		First	Second	First	Second
Start time (s)	46.21	58.37	40.56	30.38	13.54
Restart time (s)	46.91	67.20		15.94	
Shutdown time (s)	3.90	3.96		3.49	
Time to suspend active server (s)	2.27			4.54	
Time to activate suspended server (s)	8.09			1.14	

PSS		
Condition	**Active real server (Number of active virtual servers)**	**Power consumption (W) (RSG)**
Moderate (All service groups)	RS1, 2, 3 (6)	263
Light (All service groups)	RS1 (3)	93
Condition	**Operation for condition change**	**Required time (s)**
Moderate → Light (All service groups)	Set three virtual IP addresses → execute live migration → suspend two active real servers	6.54
Light → Moderate (All service groups)	Activate two suspended real servers → set three virtual IP addresses → execute live migration	14.66

4.4.3　稼働状況を記録する Log ファイル

DBSS と PSS は稼働状態の詳細情報をそれぞれの System log に時刻とともに記録を行う．また，DBSS におけるバックグラウンドジョブの稼働状況も Log に記録する．バックグラウンドジョブによって記録される Log はフォアグラウンドジョブからの Log と区別できるように先頭に文字列“_ _”を挿入している．

PSS によってサーバ構成を Moderate condition または Light condition へ移行した場合の動作状況を図 4.10 に示す．また，図には Log ファイルの内容および重要な部分を示すためにコメントを追加する．DBSS は管理対象サーバを Mail2, その IP アドレスを IPV とし，Mail2 の監視状態とする．

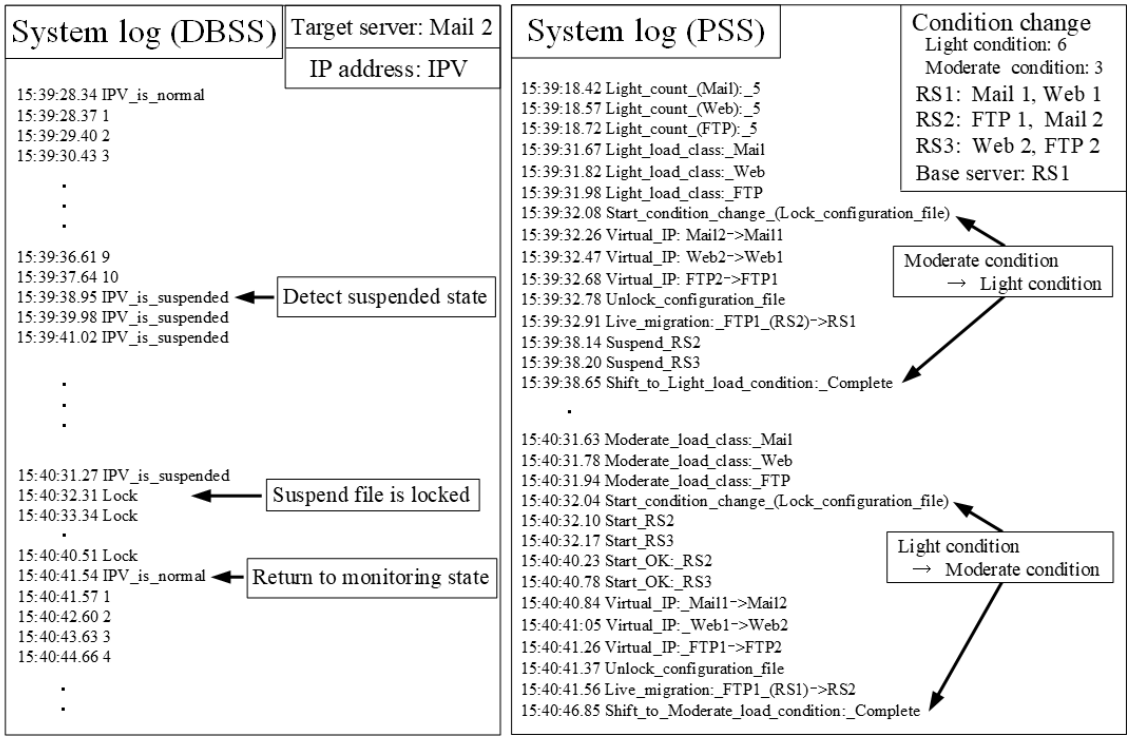


図 4.10　Moderate condition または Light condition への移行時の状態

DBSS によって，15 時 39 分 28.34 秒に Mail2 が正常状態であることを確認している．PSS では Moderate condition において Mail，Web と FTP のサービスグループにおける負荷がすべて Light load class と 6 回連続で判断され，15 時 39 分 32.08 秒にサーバシステムの構成変更を開始している．Mail2，Web2 と FTP2 に設定されている仮想 IP アドレスを Mail1，Web1 と FTP1 に再設定し，Base server が RS1 と設定されているため，FTP1 は RS2 上から RS1 上に Live migration により移動される．その後，不要となる RS2 と RS3 をサスペンド状態に変更している．DBSS では Mail2 がサスペンド状態になったことを 15 時 39 分 38.95 秒に検出している. また, すべてのサービスグループが 3 回連続で Moderate load class と判断され，15 時 40 分 32.04 秒にサーバシステムの構成変更が開始している．RS2 と RS3 が起動され, FTP1 は RS1 上から RS2 上に Live migration により移動される．その後，Mail1，Web1 と FTP1 に再設定された仮想 IP アドレスを Mail2，Web2 と FTP2 に再設定する. DBSS では Mail2 が正常状態になったことを 15 時 40 分 41.54 秒に検出し，監視状態に戻る．

管理対象サーバにトラブルが発生した場合の動作状況を図 4.11 に示す. また, 図には Log ファイルの内容および重要な部分を示すためにコメントを追加する．ここでは管理対象サーバをシャットダウンすることによりネットワークトラブルを再現する場合を示す．管理対象サーバ FTP1 の IP アドレスを IPV，バックアップサーバの IP アドレスを IPB とし，仮想 IP アドレスを VIP とする．監視間隔時間内の 16 時 23 分 42 秒に管理対象サーバをシャットダウンしており，DBSS では 16 時 23 分 50.34 秒に管理対象サーバのネットワーク異常を検出している．管理対象サーバのネットワークを再度確認し，16 時 23 分 52.86 秒に異常を再確認している．その後，サーバ機能を復旧するためにバックアップサーバの起動処理を 16 時 23 分 52.98 秒に行い，16 時 23 分 54.15 秒に起動が完了している．仮想 IP ア

ドレスの設定を行い，16 時 23 分 54.35 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻とネットワーク異常を検出した時刻との差がサーバ機能の復旧時間を示す．

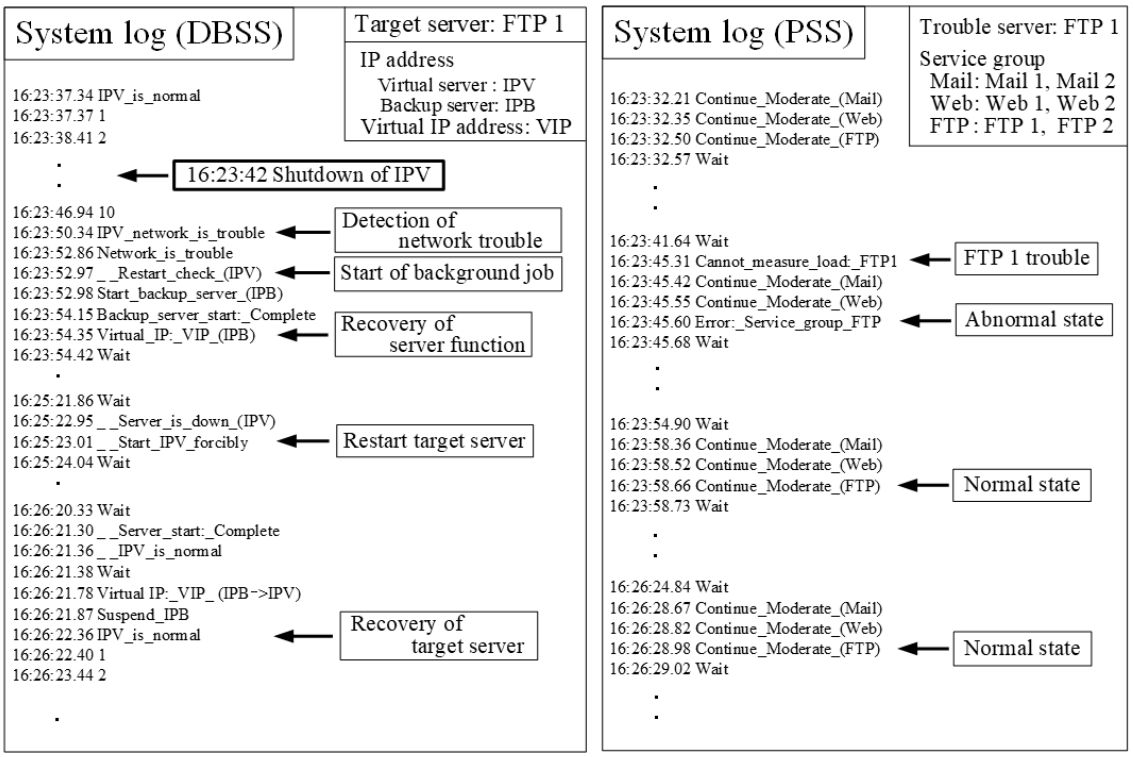


図 4.11　トラブル発生時の状態

バックグラウンドジョブの稼働状況では，16 時 23 分 52.97 秒に管理対象サーバが再起動状態であるかの確認を開始し，システムで設定した 90 秒後の 16 時 25 分 22.95 秒に管理対象サーバが停止状態であることを確認している．その後，管理対象サーバを強制的に起動させ，16 時 26 分 21.30 秒に起動の完了を確認し，16 時 26 分 21.36 秒に管理対象サーバが正常状態であることを確認後，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて管理対象サーバが正常に動作したと判断され，バックアップサーバの仮想 IP

アドレスを解除する. その後, 管理対象サーバに仮想 IP アドレスを設定し, 16 時 26 分 21.78 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻とネットワーク異常を検出した時刻との差が管理対象サーバの復旧時間を示す．PSS の System log では 16 時 23 分 45.31 秒に FTP1 の負荷測定ができないため, サービスグループ FTP の負荷調査ができない．DBSS により FTP1 のサーバ機能が復旧されたため，16 時 23 分 58.66 秒には正常状態となっている.

4.4.4 動作実験とその結果

Light condition において仮想サーバを集約する Base server を RS2 とし，FTP1 のネットワークやサービス提供プログラムおよび RS2 のネットワークにトラブルを再現する実験を行い，提案システムにおける復旧時間およびクライアントでのアクセス状態を測定する．復旧時間は DBSS によって記録された Log から実測し, クライアントでのアクセス状態は, クライアントから 1 秒間隔でアクセスを行い，その状況を記録した Log から実測する．各実測時間は 10 回行った結果の平均値とする．Moderate condition と Light condition において，RS2 または FTP1 にトラブルを再現した場合の復旧時間およびクライアントでのアクセス状態を表 4.4 に示す．“Measured time”には，サーバ機能の復旧時間，管理対象サーバが正常状態に戻るまでの時間およびクライアントでのアクセス切断時間を示す．サーバ機能の復旧時間はトラブル発生後，バックアップサーバによってサーバ機能が復旧されるまでの時間を示し，管理対象サーバが正常状態に戻るまでの時間はバックグラウンドジョブによって管理対象サーバが正常状態に戻るまでの時間を示す．実験システムにおいて監視間隔時間は 10 秒とし，トラブルは監視間隔時間が 5 秒経過時に再現する.

サービスのトラブルに関しては，サービス提供プログラムの再起動で復旧する場合と，それでは復旧せず管理対象サーバを再起動することにより復旧する場合の 2 種類により調

査を行う．サービス提供プログラムの停止において，Moderate condition および Light condition における管理対象サーバが正常状態に戻るまでの時間はともに 0.84 秒となった．また，Light condition におけるクライアントでのアクセス切断時間は 6.08 秒となった．サービス提供プログラムのトラブルにおいて，Moderate condition におけるサーバ機能の復旧時間は 5.21 秒となり，Light condition では 5.78 秒となった．また，Moderate condition における管理対象サーバが正常状態に戻るまでの時間は 73.67 秒となり，Light condition では 73.60 秒となった．Light condition におけるクライアントでのアクセス切断時間は 10.83 秒となり，これは監視間隔時間内の 5 秒とサーバ機能の復旧時間との合計値となる．そのため，サーバ機能の復旧時間より約 5 秒長くなっている．

仮想サーバのネットワークトラブルに関しては, FTP1 の意図的な再起動とシャットダウンの 2 種類により調査を行う．FTP1 の意図的な再起動において，Moderate condition におけるサーバ機能の復旧時間は 3.91 秒となり，Light condition では 4.28 秒となった．Moderate condition における管理対象サーバが正常状態に戻るまでの時間は 61.82 秒となり，Light condition では 61.74 秒となった．また，Light condition におけるクライアントでのアクセス切断時間は 11.29 秒となった．FTP1 のシャットダウンにおいて，Moderate condition におけるサーバ機能の復旧時間は 3.89 秒となり，Light condition では 4.16 秒となった．Moderate condition における管理対象サーバが正常状態に戻るまでの時間は 154.19 秒となり，Light condition では 154.34 秒となった．また，Light condition におけるクライアントでのアクセス切断時間は 11.17 秒となった. 仮想サーバのネットワークトラブルに関して，クライアントでのアクセス切断時間は監視間隔時間内の 5 秒，仮想サーバのネットワーク確認時間とサーバ機能の復旧時間との合計値となる．そのため，サーバ機能の復旧時間より約 7 秒長くなっている．

表 4.4　様々なトラブルに対する提案システムの復旧時間

Monitoring interval time: 10 s					
	Trouble list	Condition	Measured time (s)		
			Recovery of server function	Recovery of target server (Background job)	Access disconnection time (Client)
Service program	Service program stop (Recovery by service program restart)	Moderate		0.84	0.00
		Light		0.84	6.08
	Service program trouble (Recovery by server restart)	Moderate	5.21	73.67	0.00
		Light	5.78	73.60	10.83
Virtual server network	Intentional restart of FTP 1	Moderate	3.91	61.82	0.00
		Light	4.28	61.74	11.29
	Shutdown of FTP 1	Moderate	3.89	154.19	0.00
		Light	4.16	154.34	11.17
Real server network	Intentional restart of RS2	Moderate	4.11	102.34	0.00
		Light	4.14	102.04	11.15
	Shutdown of RS2	Moderate	3.83		0.00
		Light	4.04		11.05

実サーバのネットワークトラブルに関しては，RS2 の意図的な再起動とシャットダウンの 2 種類により調査を行う．RS2 の意図的な再起動において，Moderate condition におけるサーバ機能の復旧時間は 4.11 秒となり, Light condition では 4.14 秒となった. Moderate condition における管理対象サーバが正常状態に戻るまでの時間は 102.34 秒となり，Light condition では 102.04 秒となった．また，Light condition におけるクライアントでのアクセス切断時間は 11.15 秒となった．RS2 のシャットダウンにおいて，Moderate condition におけるサーバ機能の復旧時間は 3.83 秒となり，Light condition では 4.04 秒となった．ここで，DBSS は仮想サーバを管理対象とするため，実サーバに対する制御は行わない．そのため，ここでは管理対象サーバが正常状態に戻ることはない．Light condition におけるクライアントでのアクセス切断時間は 11.05 秒となった. 実サーバのネットワークトラブ

ルに関して，クライアントでのアクセス切断時間は監視間隔時間内の 5 秒，仮想サーバのネットワーク確認時間とサーバ機能の復旧時間との合計値となる．そのため，サーバ機能の復旧時間より約 7 秒長くなっている．

管理対象サーバの復旧時間は最大約 154 秒であるが，サーバ機能の復旧時間が最大約 6 秒であるため問題にはならない．Moderate condition においては，クライアントへサービスを提供する仮想サーバが冗長化構成されているため，クライアントにおけるサービス停止時間はすべて 0 秒となっている．また，省電力を向上させるため，Light condition では，それぞれのサービスグループに所属する仮想サーバは冗長化構成されていない．そのため，仮想サーバまたは実サーバに障害が発生した場合，サービス停止時間は最大で 12 秒未満となった．しかし，Light condition ではクライアントからのアクセスが少ない状態であることから，これは重大な問題にはならない．提案システムは Moderate condition または Light condition の状態で障害が発生した場合においてもサーバ機能を自動的に復旧するため，省電力と高可用性の両立を目的とした実用的なサーバシステムに適用可能である．

ここで，図 1.2 で示した稼働率による評価を表 4.5 に示す．冗長化を行っていないサーバでトラブルが発生した場合，その故障サーバ自体の復旧が求められる．そこで，管理対象サーバ自体を復旧する方式を従来方式として比較を行う．従来方式としての t_i には表 4.4 に示す管理対象サーバの復旧時間を，改善システムとしての t_{bi} にはサーバ機能の復旧時間を使用する．T_i には管理対象サーバの復旧時間における最大値を使用し，その値での従来方式の稼働率（Conventional sysytem）を 0.5 とする．また，提案方式による稼働率の改善（Improvement）を式(2.10)で算出する．

管理対象サーバが仮想サーバの場合を以下に示す．サービス提供プログラムの再起動で復旧しない場合のトラブルでは，Moderate condition において従来方式の稼働率は 0.68，提案方式の稼働率（Proposed system）は 0.98 となり 44.12%の改善，Light condition に

おいて従来方式の稼働率は 0.68，提案方式の稼働率は 0.97 となり 42.65%の改善がみられる．また，再起動した場合のトラブルでは，Moderate condition において従来方式の稼働率は 0.71，提案方式の稼働率は 0.98 となり 38.03%の改善，Light condition において従来方式の稼働率は 0.71, 提案方式の稼働率は 0.98 となり 38.03%の改善がみられる．さらに，シャットダウンした場合のトラブルでは，Moderate condition において従来方式の稼働率は 0.50，提案方式の稼働率は 0.99 となり 98.00%の改善，Light condition において従来方式の稼働率は 0.50，提案方式の稼働率は 0.99 となり 98.00%の改善がみられる．

表 4.5　提案方式による稼働率の改善

Target server
VS: Virtual server
RS: Real server

Condition
MC: Moderate
LC: Light

St2: Service program trouble (Recovery by server restart)
Nt1: Intentional restart of target server
Nt2: Shutdown of target server

				Conventional system	Proposed system	Improvement (%)
Availability rate	MC	VS	St2	0.68	0.98	44.12
			Nt1	0.71	0.98	38.03
			Nt2	0.50	0.99	98.00
		RS	Nt1	0.60	0.98	63.33
		Average		0.62	0.98	58.06
	LC	VS	St2	0.68	0.97	42.65
			Nt1	0.71	0.98	38.03
			Nt2	0.50	0.99	98.00
		RS	Nt1	0.60	0.98	63.33
		Average		0.62	0.98	58.06

実サーバが再起動した場合のトラブルでは，Moderate condition において従来方式の稼働率は 0.60，提案方式の稼働率は 0.98 となり 63.33%の改善，Light condition において従

来方式の稼働率は 0.60，提案方式の稼働率は 0.98 となり 63.33%の改善がみられる．提案方式は従来方式に比べ稼働率の面において非常に高い優位性が認められる．

付録の F2 に示すようにサーバの MTBF を 100,000 時間として計算を行うと，実サーバ数が 20 台の場合，1 年間で約 2 台が故障する．ここで，実験システムの管理サーバは，最大で同時に 4 台の仮想サーバを起動できるリソースとなっている．各実サーバでは 2 台の仮想サーバを起動していることから実サーバにおいて同時に 2 台の故障までは対応が可能となる．そのため，実サーバは 20 台で管理対象となる仮想サーバは 40 台までに設計することが望ましい．初期費用や運用コストに関する従来方式との比較は，3 章の表 3.7 で示した内容と管理対象サーバの最大値（$2n$）以外は同様となるため，提案方式は従来方式に比べコスト面においても優位性が認められる．

4.5 まとめ

独立して動作可能な MSBS と PSS を複合動作させることにより，省電力かつ高可用性サーバシステムを構築した．MSBS は 2 章で示したシステム，PSS は 3 章で示したシステムをそれぞれ本章のシステムに適用するために変更した．具体的な変更としては，MSBS では管理対象サーバを仮想サーバとし，故障サーバの機能を復旧するために起動するバックアップサーバに関しても管理を行った．また，仮想サーバシステムでは複数台の管理対象サーバにおける同時故障が予想されるため，管理サーバの負荷を軽減するために，管理サーバ上に起動したバックアップサーバは管理対象サーバを起動していた実サーバに Live migration による移動を行った．PSS では，サービスグループから測定する負荷はメモリの使用率，CPU 使用率およびネットワークにおける送受信率の 3 種類とした．また，クライアントからアクセスするサーバの変更は，仮想 IP アドレスの設定により行った．さらに，省電力状態のサーバ構成は，仮想 IP アドレスの設定，Live migration によるサーバ移動お

よび実サーバのサスペンド処理により実現した．すべてのサービスグループが Moderate condition の場合，サービスを提供するための仮想サーバを起動している実サーバの消費電力は合計 263W，Light condition では 93W となり，35%の電力での運用が可能となった．MSBS と PSS を複合動作する上での問題点を示し, その解決策を検討した. ここでは, DBSS が PSS の構成ファイルを編集可能にすることで複合動作を実現した．通常，2 種類の管理プログラムを複合動作させる場合，両方のプログラムにおいて誤動作を防止するためにそれらのプログラムは複雑となる. DBSS のプログラム構成が複雑になることを防ぐために，2 つのシステムを仲介する SIF を提案し，導入した．SIF に PSS の構成ファイルへのアクセスと編集機能を追加することにより，DBSS のプログラムがシンプルな構成を維持することができた．また，SIF は DBSS からのシグナルを割り込み処理で受け取ることで，高速な対処を実現した. SIF を採用した実験用の省電力かつ高可用性サーバシステムを構築し, その仕様を示した．PSS によって構成される Moderate condition と Light condition のサーバ構成において，管理対象サーバにサービス提供プログラムおよびネットワークに関するトラブルを再現した場合における管理対象サーバの機能およびサーバ自体の復旧に要する時間を測定した．本提案システムは，各サーバ構成においてそれらのトラブルが発生した場合においても，管理対象サーバの機能およびサーバ自体の復旧が可能であることを示した．仮想サーバや実サーバの故障に関して，6 秒未満でサーバ機能を復旧可能であることを示した．また，従来方式との比較では，提案方式は稼働率において非常に高い優位性があることを示し，コスト面においても高い優位性が認められる．

第 5 章

Peer-to-Peer 方式を採用したサーバ管理システムの開発と動作検証

5.1 まえがき

高度情報化社会の発展に伴い，高性能通信端末の普及や FTTH，LTE といったアクセスネットワークの高速かつ広帯域化が急速に進められている．また，ユーザへのサービス向上のため様々なインターネットサービスが提供されており，現在では我々の生活にとってなくてはならないサービスとなっている．そのため，そのサービスをより安定的に提供するためには，そのサービスを支えるサーバシステムの役割はより重要[87]となり，その安定稼働を支えるサーバシステムの管理およびその最適化設計や構築も更に重要となる[48].

一般的なサーバ管理システムは特定の管理サーバが複数の管理対象サーバを監視する[88]．このような管理方法を本論文ではサーバ・クライアント方式と呼ぶ．この方式では，管理対象サーバの台数に比例し，管理における負荷の増加や管理サーバの故障時には管理対象サーバの管理ができなくなるといった問題が生じる．その対策として，管理サーバに対して冗長化を行うなどの耐障害性や信頼性の強化が必要となるが，制御の複雑化や運用コストの増加が懸念される．

障害が発生した場合でもサーバに対して通信を可能とするネットワークシステムについて様々な研究がなされており，Fault-tolerant ネットワークとそのルーティングプロトコルの設計[89]，データセンタにおいてサーバを中心と考えた高可用性ネットワークの開発[90]，仮想マシンを各種サービスタイプのクラスタにグループ化する分散仮想ネットワークアーキテクチャの提案[91]，ネットワーク故障に対するロバスト性を実現するため，サーバ配置

とリンク保護を組み合わせたネットワーク設計法[92]などが報告されている．

また，データセンタに焦点をあてた障害対策において多くの重要な研究が進められており，障害発生予測と自動化またはオペレータ動作を可能とする障害位置予測との統合法の提案[81]，データセンタネットワークの配置とクラウドサービス保護の両方を考慮した統合化設計[93]，仮想データセンタにおけるサーバやネットワークリソースの保障に関して，ハードウエア故障およびサービスの異常などを考慮[80]，データセンタ装置の故障率を考慮した高可用性仮想インフラストラクチャ管理フレームワークの提案[78]，仮想マシン故障時にホットスペアノードを瞬時に起動するためのネットワークコーディング[43]などが報告されている．

データセンタにおいてサービスを提供するシステムの基本単位はサーバシステムであり，そのシステムにおける高可用性に関する様々なアイデアが報告されている．サーバシステムの冗長化において一般的に使用されているフェイルオーバクラスタシステムやロードバランサクラスタシステムはサーバ故障に対して非常に効果的な対処を可能とするが，冗長化対象のサーバ群は同一のサービスを提供する構成が基本となるためコスト面に問題がある．そのため，サーバ・クライアント方式の管理システムにおいて，実サーバのサーバ機能を仮想サーバで復旧可能であることを示し，複数台の実サーバを 1 台の実サーバでバックアップ可能な複合サーババックアップシステムの開発[40]や独立して動作可能な省電力サーバシステムと高可用性サーバシステムを交互に動作させる方式を採用した省電力かつ高可用性サーバシステムの開発[86]などが報告されている．

また，サーバ・クライアント方式とは異なり特定の管理サーバを必要としない管理システムについて報告されており，複数のサーバが使用する分散データベースシステムにおいて高可用性を実現するために，プロセス障害の即時対処を考慮し，すべてのプロセスを 1 方向に循環する形でコネクションを形成する TCP リングを採用するとともに，サーバ監視

に関してはターゲットリストをもとにしたランダムな死活監視を採用[94]している．また，高可用分散クラスタにおいて監視リングを構成し，各サーバでの監視による負荷の隔たりを考慮して多重故障発生時の対処として検出した故障情報はいったん特定のサーバに集約し，その後，そのサーバがクラスタ全体に通知する方式[95]や P2P の特徴を導入したシステムとしては，P2P システムと計算グリッドを組み合わせた P2P グリッドシステムを対象とし，そのシステムでは，タスクを実行するサーバは定期的にチェックポイントを記録しており，そのサーバ故障時には予備のサーバがそのチェックポイントの記録を参考にその役割を引き継ぐチェックポイント・アンド・リスタートメカニズムを採用[96]している．さらに，サービスを提供する仮想サーバ群とそのデータを提供するストレージサーバ群から構成したシステムで，各サーバにエージェントを動作させ，各エージェントが協調して動作する P2P システムが運用されている．そのシステムでは，複数のサーバが管理対象サーバを監視し，障害の判断に関しては特定のサーバに情報を集約して，その情報から判断する．仮想サーバを起動しているホストサーバが故障した場合，稼働していた仮想サーバを予備のホストに移動することによりサービスの継続を行う方式[97]などが報告されている．

しかしながら，サーバ・クライアント方式の管理システムには前述した問題があり，特定の管理サーバを必要としない管理システムについては，管理対象のサーバをランダムに選択することで生じるネットワーク帯域の不均一な使用，故障時対策のために稼働しているサーバと同機能の予備サーバを用意する構成のため，システムの規模に応じたコストの増加や故障時の対処として特定のサーバに情報を集約して障害の判断を行うことから，障害対処時においてその判断を行うサーバの故障時などが懸念される．また，管理グループを採用した P2P 方式サーバ管理システムも報告[98]されているが，基本的に 4 台のサーバ毎に管理グループを構成し，管理グループ毎に独立して管理を行うことから，そのグループに所属するすべてのサーバが同時に故障した場合，管理ができなくなる問題点を有して

いる．

そこで，本論文ではネットワークシステムにおいてサーバの耐障害性が高い P2P 方式を採用し，管理グループ毎に分割した管理を不要としたサーバ管理システムを提案する．本提案システムは P2P 方式の特徴を採用することから，クライアントにサービスを提供するサーバが管理サーバかつ管理対象サーバとなる．提案システムは異なるサービスを提供するサーバ群に適用可能であり，管理対象サーバの台数に依存しない．管理対象サーバの監視からその故障時の対処を各管理サーバが独立して行う．また，故障したサーバの機能を復旧するために予備の実サーバではなく仮想サーバを使用する．仮想サーバはクライアントにサービスを提供している管理対象サーバ上に起動するため，管理対象サーバ群の負荷状態を考慮した仮想サーバの起動方式を採用している．本提案システムを構築するために，P2P 方式の問題点となる管理プログラムの分散による管理の複雑化を防ぐ方式や管理の優先順位および複数サーバの同時故障における対処を可能とする動的管理拡張方式を提案し，導入する．さらに，実験システムを構築し，サービス提供プログラムやネットワークのトラブルを再現する実験を行い，サーバ故障時からサーバの機能およびサーバ自体の復旧に要する時間の測定およびその予測も行う．

5.2 P2P 方式サーバ管理システムの概要

5.2.1 サーバ管理方式

図 5.1 にサーバ管理システムにおけるサーバ・クライアント方式および P2P 方式の 2 種類を示す．サーバ・クライアント方式は(a)に示すように 1 台の管理サーバが複数の管理対象サーバにおけるネットワークとサービス提供状態を監視する従来方式である．管理対象サーバの異常を検出した場合には，そのサーバを制御し，その機能の復旧作業を行う．この方式ではサーバ管理者は 1 台の管理サーバにおいて，管理プログラムの更新および修正，

そのプログラムが記録する Log ファイルから管理対象サーバ全体の稼働状況を調査することができる．しかし，1 台の管理サーバで複数の管理対象サーバを監視および制御することから，管理対象サーバの台数に比例して管理サーバの管理における負荷が増加する．また，管理サーバが故障すると，記録していた Log ファイルの消失や管理対象サーバを管理できなくなることが予想される．その対策として，管理サーバを冗長化する必要があり，故障時における制御の複雑化も想定される．

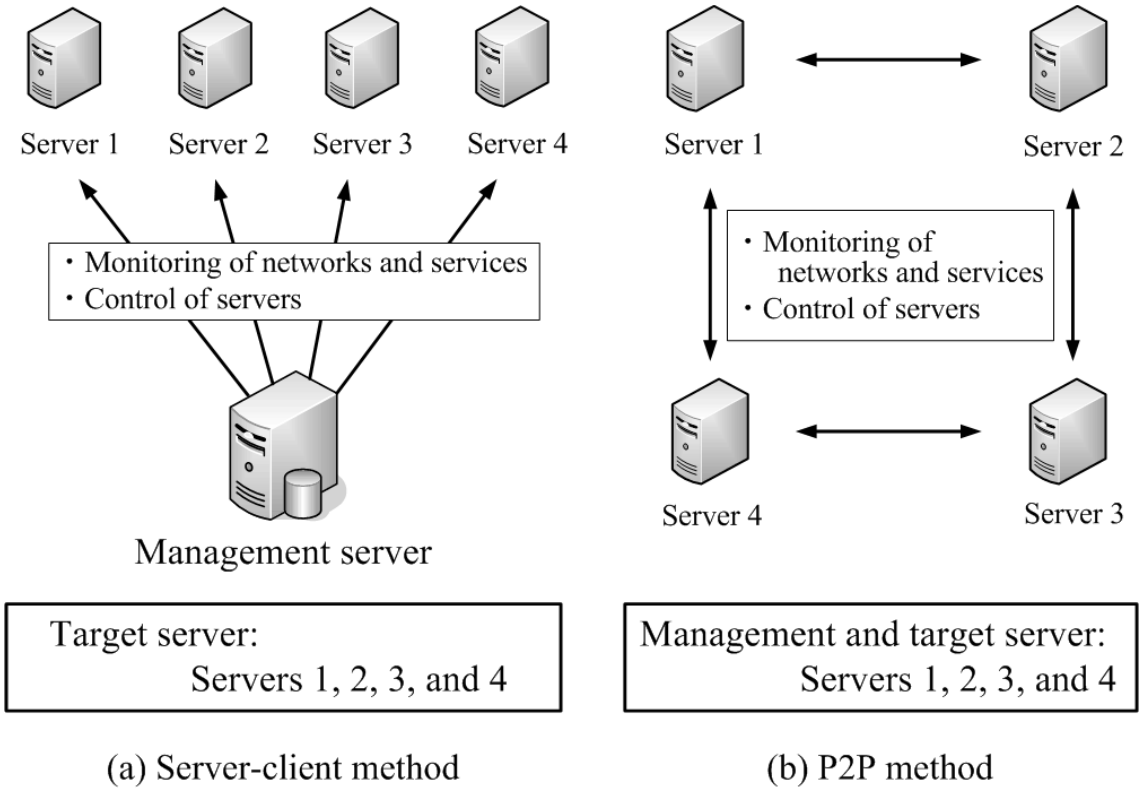


図 5.1　サーバ・クライアント方式および P2P 方式によって構成されたサーバ管理システム

提案している P2P 方式は(b)に示すように特定の管理サーバは持たず，それぞれの管理対象サーバが管理サーバの機能を有しており，1 台の管理サーバが 2 台の管理対象サーバを監

視している．管理対象サーバに異常を検出した場合には，担当する管理サーバがそのサーバを制御してその機能の復旧作業を行う．P2P 方式はサーバ・クライアント方式に比べ管理サーバの耐障害性が向上する．また，管理すべきサーバの台数を考慮する必要がない．しかし，P2P 方式は管理プログラムをすべての管理サーバにインストールする必要があるため，管理プログラムの更新や修正に要するサーバ管理者の負担が増加する．また，そのプログラムが記録する Log ファイルから管理対象サーバにおける稼働状況の調査を行うために，複数の管理サーバにアクセスしなければならない問題点を含む．

表 5.1 と 5.2 に，図 5.1 に示した従来方式であるサーバ・クライアント方式と提案している P2P 方式の比較結果を示す．従来方式としては 2 章で示した MSBS とする．提案方式を“P2P”とし，従来方式を“MSBS”と示す．提案方式と従来方式は故障したサーバの機能を復旧するために仮想サーバを使用する．

表 5.1 に管理対象サーバの台数に応じた管理サーバ 1 台における管理すべき台数を示す．MSBS は管理対象サーバの台数に比例して管理台数も増加するが，P2P では管理対象サーバが 3 台以上の場合において，それぞれの管理サーバは 2 台のみの管理となる．このため従来方式は管理サーバにおける管理のための負荷および使用するネットワーク帯域の増加が懸念される．

表 5.1　管理サーバ 1 台における管理対象サーバの台数

	The number of target servers					
	2	**3**	**4**	**・・・**	***n*-1**	***n***
MSBS	2	3	4	・・・	*n*-1	*n*
P2P	1	2	2	・・・	2	2

表5.2に故障したサーバの台数に応じた管理サーバ1台における仮想サーバの最大起動台数を示す．ここでは，管理対象サーバの総数を n 台とする．MSBS は故障サーバの台数に比例して増加するが，P2P では管理サーバが複数台存在することから，管理対象サーバの半数までの故障に対して管理サーバ 1 台における仮想サーバの最大起動台数は 1 台となる．また，P2P は半数以上のサーバが故障した場合でもその台数は低く，n - 1 台の場合のみ同値となる．

表 5.2　管理サーバ 1 台における仮想サーバの最大起動台数

	The number of troubled servers								
	1	**2**	**3**	・・・	$\frac{n}{2}$	$\frac{n}{2}+1$	$\frac{n}{2}+2$	・・・	n-1
MSBS	1	2	3	・・・	$\frac{n}{2}$	$\frac{n}{2}+1$	$\frac{n}{2}+2$	・・・	n-1
P2P	1	1	1	・・・	1	2	3	・・・	n-1

5.2.2　P2P 方式サーバ管理の基本的な管理構成

図 5.2 に一般的なサーバシステムを例として，P2P 方式の基本的な管理構成について示す．このサーバシステムは，File サーバが 2 台，DNS サーバが 2 台，サーバグループとして 6 台から構成されている．File サーバと DNS サーバは冗長化構成とする．ここで，サーバグループに所属する 6 台のサーバはクライアントにサービスを提供する管理対象サーバであり，本提案方式を採用することから管理サーバとしても動作する．Server1～Server6 は File サーバに NFS 接続し，それぞれのサーバが使用可能な共有エリアを構築する．それぞれのサーバはサービス A～E をクライアントに提供しており，Server3 と 4 のみ同サービスを提供している．本提案システムはこのようにクライアントへ異なるサービスを提供するサーバで構成することができる．

管理サーバは管理の優先順位を持つ管理プログラムを実行し，図に示すように Server1 は管理の優先順位 1（P1）として Server2 を，優先順位 2（P2）として Server6 を管理する．また，Server2 は P1 として Server1 に，P2 として Server3 によって管理される．P2P 方式では 1 台の管理対象サーバを 2 台で管理することからこの優先順位を採用し，管理対象サーバの故障時において対処する管理サーバの優先権を明確化している．

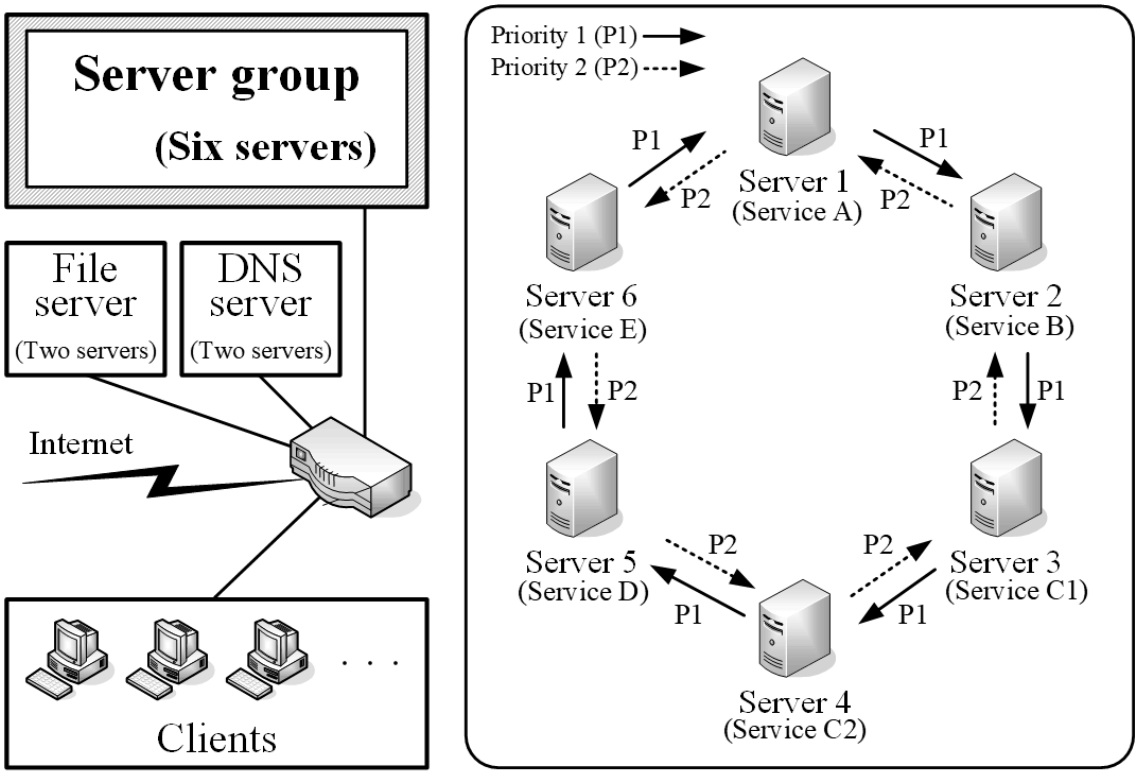


図 5.2　P2P 方式における基本的な管理構成

5.2.3　Executer システム

P2P 方式ではそれぞれの管理サーバに管理プログラムをインストールする必要があるため，プログラムの管理が困難となる問題を含んでいる．本提案方式において，その問題を解決するために Executer プログラムを開発し, 実装する. 図 5.3 にその構成と機能を示す.

各管理サーバは Executer プログラムがインストールされ，管理プログラムや管理データおよび Log ファイルの集中管理を実現するため，File サーバと NFS 接続する．管理サーバは起動時において自動的に Executer プログラムを実行する．Executer プログラムはローカルエリアにある Flag ファイルの有無を調査し，Flag ファイルが存在した場合，共有エリアに保存されている管理プログラム，サブプログラムと管理データをローカルディスクにコピーし，それらのプログラムを実行する．管理プログラムは管理の優先順位に対応し，2 台の管理対象サーバに対して実行される．管理対象サーバを管理する上で同期をとるためのプログラムと管理サーバの負荷状態を調査するためのプログラムはサブプログラムとして実行される．

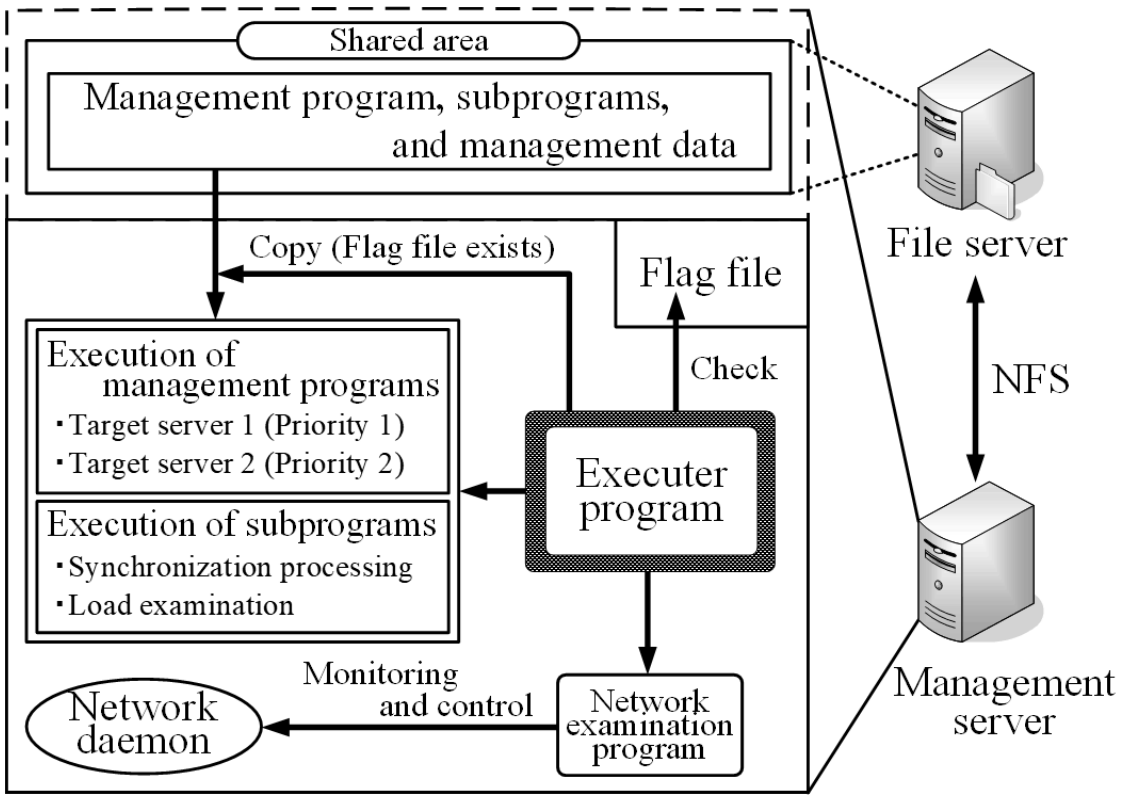


図 5.3　Executer システムの構成とその機能

Flag ファイルを削除した場合，Executer プログラムは実行中の管理プログラムおよびサブプログラムをすべて停止し，管理サーバとしての動作を停止する．これは，管理プログラムの更新や修正に対処するための機能であり，これにより，P2P 方式の問題点である管理プログラムの分散による管理の複雑化を防ぐことができる．また，Executer プログラムはネットワーク調査プログラムをバックグラウンドジョブとして実行し，そのプログラムが定期的にローカルサーバのネットワークサービス提供プログラムを監視する．これは，ローカルサーバのネットワークが異常となった場合，他サーバからの復旧作業が困難となるため，この監視機能を追加している．もし，異常を検出した場合には，ネットワークサービス提供プログラムを再起動して機能の復旧を試みる．

5.2.4 管理データ

管理プログラムは管理データを引数として実行する．管理データには管理に必要となる情報が記録されており，その詳細を図 5.4 に示す．管理データの 1 行目は管理対象サーバに設定する仮想 IP アドレス，2 行目から 6 行目までは管理対象サーバの情報，7 行目と 8 行目は管理対象サーバ故障時にその機能を復旧するための仮想サーバの情報が記録される．管理対象サーバが故障した場合，そのサーバ機能を復旧するために，管理データの 1 行目に記録されている仮想 IP アドレスが仮想サーバに設定される．ここで，5 行目のサービスデーモン名は 1 台のサーバがクライアントに対して複数のサービスを提供する可能性があるため，“postfix dovecot”のようにスペース区切りで複数記述することができる．6 行目のサーバタイプは CPU を主に使用するサーバを“CPU”，ネットワークでのデータ転送を主として行うサーバを“NETWORK”とし，その両方を行うサーバを“CPU_NET”とする．ここでの管理とは，管理対象サーバのネットワークおよびサービス提供プログラムの稼働状況を監視することや異常を検出した場合にその対処を実施することおよび管理対象

サーバの機能を復旧するために，そのサーバ機能を持つ仮想サーバを起動し，そのサーバに仮想 IP アドレスを設定することを示す.

Management data	Mail.dat	Web.dat
1st line: **Virtual IP address (VIP)**	172.21.14.201	172.21.14.202
2nd line: **Host name**	Mail	Web
3rd line: **IP address (IPR)**	172.21.14.1	172.21.14.2
4th line: **Mac address**	00:00:00:00:00:01	00:00:00:00:00:02
5th line: **Service daemon**	postfix dovecot	httpd
6th line: **Server type**	NETWORK	CPU_NET
7th line: **Virtual server host name**	Mailb	Webb
8th line: **Virtual server IP address (IPV)**	172.21.14.101	172.21.14.102

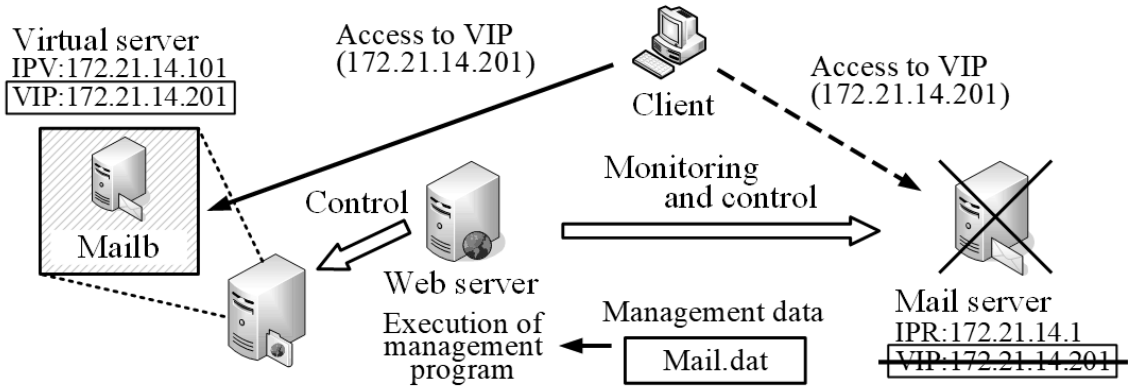


図 5.4　管理データの詳細

図 5.4 で，Mail サーバと Web サーバはクライアントにサービスを提供している．管理対象サーバを Mail サーバとすると，その管理データ名は Mail.dat となる．図では Web サーバ上で Mail.dat を引数として管理プログラムが実行され，その内容をもとに Mail サーバの管理を行う．クライアントは Mail サーバに設定されている仮想 IP アドレス“172.21.14.201”に対してアクセスを行う．もし，Mail サーバにトラブルが発生した場合には，Web サーバがすべての管理サーバの中で最適と判断した実サーバから仮想サーバ Mailb を起動後，管理データの 1 行目に記録されている仮想 IP アドレス“172.21.14.201”

を設定する．これによりクライアントからのアクセスはその仮想サーバに対して行われる．この管理データを引数として実行するサーバ管理方法により，管理プログラムを管理対象サーバの種類に応じて変更する必要がなくなる．

5.2.5 負荷状態を考慮した仮想サーバ起動方式

故障した管理対象サーバの機能を復旧するために，管理サーバ群の負荷状態を考慮した仮想サーバの起動方法に関して図 5.5 に示す．各管理サーバは図 5.3 で示したサブプログラムにより管理サーバの負荷測定が定期的に行われる．サーバ負荷に関しては，メモリ，CPU とネットワークの使用量とする．(a)に予測負荷を算出するために必要となる各係数や算出式を示す．i は測定回数を示し，m_i はサーバの利用可能メモリサイズ，c_i は CPU の Idle 率と n_i はネットワークの送受信の合計転送バイト量を示す．これら測定値をもとにそれぞれの平均負荷値である $A\mathrm{m}$，$A\mathrm{c}$ と $A\mathrm{n}$ を算出する．これらの値をもとに，最も単純な方法で負荷の予測を行う．ここでは，各管理サーバにおいて 2 段階で取得した平均負荷値から，その差分をもとに予測負荷値を算出する．メモリの予測負荷 $L\mathrm{m}_n$ は式(5.1)で示される．

$$L\mathrm{m}_n \ [\mathrm{KB}] = 2 \times A\mathrm{m}_n - A\mathrm{m}_{n-1} \tag{5.1}$$

CPU の予測負荷 $L\mathrm{c}_n$ は式(5.2)となる．

$$L\mathrm{c}_n \ [\%] = 2 \times A\mathrm{c}_n - A\mathrm{c}_{n-1} \tag{5.2}$$

ネットワークの予測負荷 $L\mathrm{n}_n$ は式(5.3)で示される．

$$L\mathrm{n}_n \ [\mathrm{KB}] = 2 \times A\mathrm{n}_n - A\mathrm{n}_{n-1} \tag{5.3}$$

式(5.1)，(5.2)と(5.3)から得られる負荷は変動によりマイナス値や上限を超えてしまう可能性があるため，最小値を 0 とし，最大値は $L\mathrm{m}$ で管理サーバの物理メモリ量，$L\mathrm{c}$ でコア数×100 とし，$L\mathrm{n}$ で定格転送速度×2 としている．各管理サーバはそれぞれリソースが異なるため，比較を行うには正規化が必要となり，そのためには以下の係数が必要となる．$M_{\max}$

はすべての管理サーバにおける利用可能な最大メモリサイズで $M_{\min}$ は利用可能な最小メモリサイズ，C_{cor} は各管理サーバにおける CPU のコア数，$C_{\max}$ は各管理サーバにおける $Lc \times C_{\mathrm{cor}}$ の最大値で $C_{\min}$ は最小値，N_{rat} はネットワークの定格送受信速度における 1 秒間の合計転送バイト量とし，$N_{\max}$ は各管理サーバにおける N_{rat} - Ln の最大値で $N_{\min}$ は最小値とする．ここで，1,000Mbps のルータを例とすると定格転送速度は 1 秒間で 128×10^3KB 転送可能となるため N_{rat} は送受信の合計値より 256×10^3KB となる．以下にメモリ，CPU とネットワークの負荷値を正規化するための計算方法を示す．正規化するにあたり，各最大値と最小値の値がそれぞれ異なる場合と同じ場合とでその計算式が異なる．各管理サーバにおけるメモリの正規化された負荷値 Nm_n は式(5.4)と(5.5)で示される．

$$Nm_n\ [\%] = \frac{M_{max} - Lm_n}{M_{max} - M_{min}} \times 100 \qquad (M_{max} \neq M_{min}) \tag{5.4}$$

$$Nm_n\ [\%] = 100 \qquad (M_{max} = M_{min}) \tag{5.5}$$

CPU の正規化された負荷値 Nc_n は式(5.6)と(5.7)となる．

$$Nc_n\ [\%] = \frac{C_{max} - Lc_n \times C_{\mathrm{cor}}}{C_{max} - C_{min}} \times 100 \qquad (C_{max} \neq C_{min}) \tag{5.6}$$

$$Nc_n\ [\%] = 100 \qquad (C_{max} = C_{min}) \tag{5.7}$$

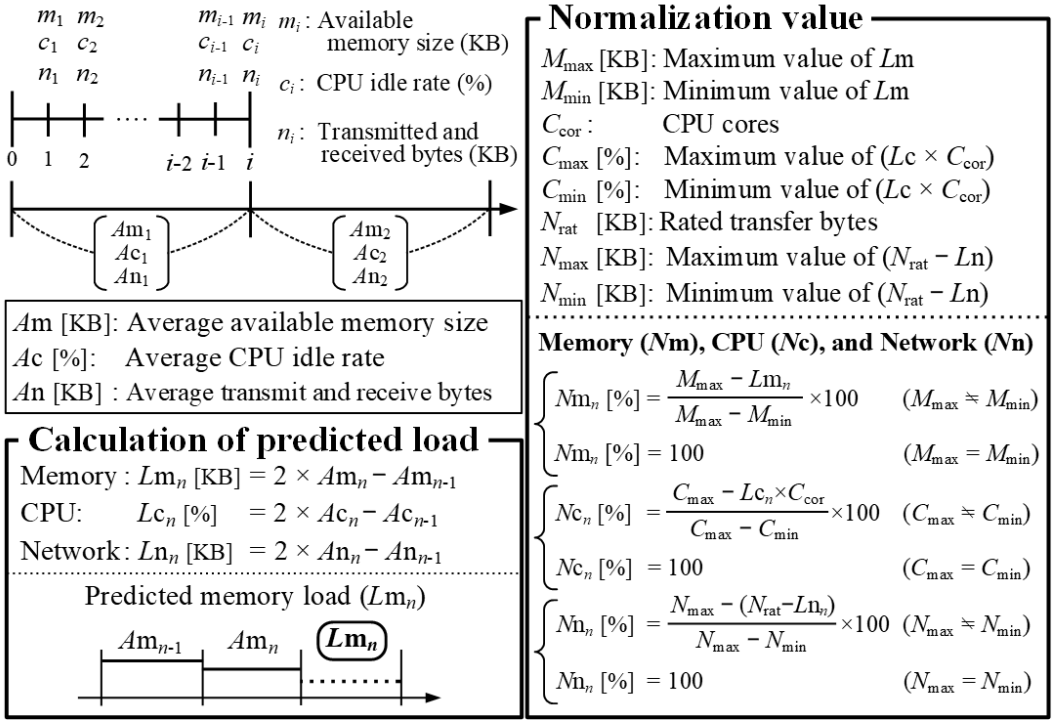
ネットワークの正規化された負荷値 Nn_n は式(5.8)と(5.9)で示される．

$$Nn_n\ [\%] = \frac{N_{max} - (N_{\mathrm{rat}} - Ln_n)}{N_{max} - N_{min}} \times 100 \quad (N_{max} \neq N_{min}) \tag{5.8}$$

$$Nn_n\ [\%] = 100 \qquad (N_{max} = N_{min}) \tag{5.9}$$

正規化された負荷はそれぞれ最小値が 0，最大値が 100 となる．管理サーバ毎にその負荷値と起動している仮想サーバの台数 VM を Real server condition ファイルに記録する．

本システムでは故障した管理対象サーバの機能を復旧するために仮想サーバを使用する．そのため，仮想サーバの起動対象となる管理サーバの選択が必要となる．Real server condition ファイルをもとに故障サーバのサーバタイプに応じた管理サーバ毎の非選択値を算出する．非選択値とはその値が大きい程，仮想サーバの起動に適切ではないことを示す．



(a) 予測負荷の算出およびその正規化

Real server condition file

Management server	*VM*	Memory (*N*m [%])	CPU (*N*c [%])	Network (*N*n [%])
Web	0	40	30	50
Mail	1	30	20	40

VM: Number of virtual servers running on management server

Server type

CPU : Login server
NETWORK: FTP, mail, and proxy servers
CPU_NET : Web server

Non-select value

CPU:
$V_{CP\_server} = VM \times 100 + Nm \times 0.5 + Nc \times 0.4 + Nn \times 0.1$
NETWORK:
$V_{NE\_server} = VM \times 100 + Nm \times 0.5 + Nc \times 0.1 + Nn \times 0.4$
CPU_NET:
$V_{CN\_server} = VM \times 100 + Nm \times 0.5 + Nc \times 0.25 + Nn \times 0.25$

FTP server failure

Server type: NETWORK

Calculation of non-select value

$V_{NE\_web} = 0 \times 100 + 40 \times 0.5 + 30 \times 0.1 + 50 \times 0.4 = 43$
$V_{NE\_mail} = 1 \times 100 + 30 \times 0.5 + 20 \times 0.1 + 40 \times 0.4 = 133$
:

Selection of server with the lowest non-select value

$V_{NE\_web} < \cdots < V_{NE\_mail} < \cdots$

⇩

Start up virtual server on **Web** server

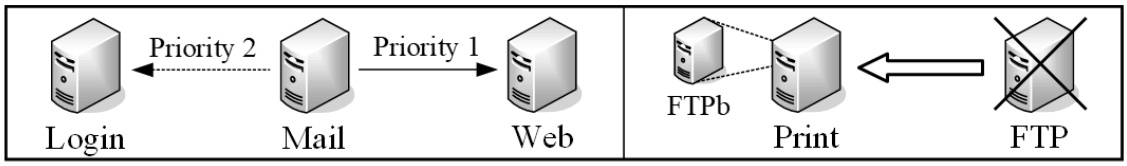
(b) 故障サーバのタイプを考慮した仮想サーバ起動用管理サーバの選択

図 5.5　負荷状態を考慮した仮想サーバ起動方式

非選択値の算出方法を図 5.5(b)に示す．サーバタイプとしては CPU を主として使用する CPU タイプ，データを主として提供する NETWORK タイプとその両方を提供する CPU_NET タイプの 3 種類とする．一般的なサーバを例とし，図にその分類を示す．Login サーバはリモートログインを許可し，プログラムの実行等，CPU 演算処理などが主なサービスとなるため CPU タイプ，FTP サーバ等はデータの提供等が主なサービスとなるため NETWORK タイプ，Web サーバは HTML データや画像データの提供と CGI 利用のため CPU_NET タイプとしている．サーバタイプに応じて Nm，Nc と Nn に関してそれぞれに重み付けを行う．メモリ負荷 Nm はどのサーバタイプにおいても重要となるため 0.5 倍とする．CPU タイプでは Nc を 0.4 倍，Nn を 0.1 倍，NETWORK タイプでは Nc を 0.1 倍，Nn を 0.4 倍とし，CPU_NET タイプでは Nc と Nn をそれぞれ 0.25 倍とする．これらから得られる合計値は最高で 100 としている．また，既に仮想サーバを起動している管理サーバは仮想サーバを起動していない管理サーバより非選択値を高くするため，VM には 100 倍の重み付けをする．図に NETWORK タイプである FTP サーバが故障した場合を示す．Real server condition ファイルの値から非選択値である V_{NE_web} や V_{NE_mail} などを算出し，その値が一番低い Web サーバが選択されている．

5.2.6 管理ファイル

本提案システムは管理動作を設定するため，図 5.6 に示す管理ファイルである Group，Operation，Separation，Trouble と Real server condition ファイルを使用する．これらのファイルは File サーバ上の共有エリアに保存されており，各管理サーバは NFS 接続により，これらを使用する．



Group file

Number	Target server	Management data
1	Mail	Mail.dat
2	Web	Web.dat
3	Proxy	Proxy.dat
4	Login	Login.dat

Operation file

Priority	Management server	Target server	Management data
1	Mail	Web	Web.dat
1	Web	Proxy	Proxy.dat
1	Proxy	Login	Login.dat
1	Login	Mail	Mail.dat
2	Mail	Login	Login.dat
2	Web	Mail	Mail.dat
2	Proxy	Web	Web.dat
2	Login	Proxy	Proxy.dat

Separation file

Number	Target server	Management data
8	FTP	FTP.dat

Trouble file

Virtual server (Backup server)	Trouble server	Management server
FTPb	FTP	Print

Real server condition file

Management server	VM	Memory (%)	CPU (%)	Network (%)

図 5.6　管理ファイルの詳細

Group ファイルには番号，管理対象サーバ名，そのサーバを管理するための管理データファイル名が記録されている．Operation ファイルは，この Group ファイルをもとに作成され，管理の優先順位，管理サーバ，管理対象サーバとその管理データファイル名が記録される．この図では Operation ファイルの太線で囲まれた部分の情報により，Mail サーバが P1 として Web サーバを，P2 として Login サーバを管理している．また，Trouble ファイルは管理対象サーバの故障時に作成され，サーバ機能を復旧する仮想サーバ名，故障した管理対象サーバ名とその仮想サーバを起動する管理サーバ名が記録される．Real server condition ファイルは，図 5.5(b)で示した内容が記録される．この図では FTP サーバの故障に対して Print サーバがその機能を復旧するために仮想サーバ FTPb を起動しており，この

状況が Trouble ファイルの太線で囲まれた部分に記録されている．ここで，仮想サーバを起動する管理サーバの選択に関しては，Real server condition ファイルをもとに図 5.5(b)で示した非選択値を算出し, その値が一番低いサーバである Print サーバが選択されている．

各ファイルは連携しており，サーバが故障したと判断した場合，Trouble ファイルに該当データを記録し, そのサーバのデータを Group ファイルから Separation ファイルに移動する．その後，Group ファイルから新たな Operation ファイルを作成する．サーバが復旧した場合には，該当するデータを Trouble ファイルから削除後，Separation ファイルから該当データを Group ファイルに戻して，Operation ファイルを再構築する．このファイルを基本として管理構成を動的に変更する．

5.3 P2P 方式サーバ管理システムの動作および復旧手順

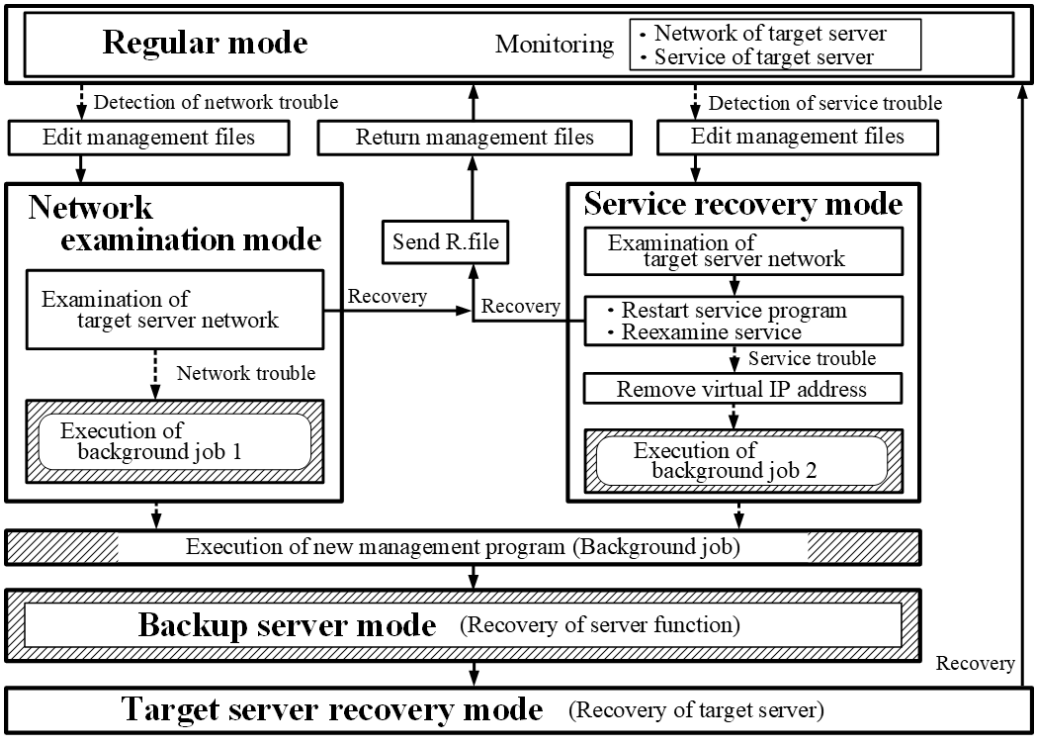
各管理サーバは管理をする上で共通エリアが必要となるため File サーバとの NFS 接続を使用する．このため File サーバの負荷を考慮し，通常の監視状態時には管理サーバ間におけるプロセス間通信を採用することにより，システム起動時と管理対象サーバのサーバ機能復旧処理以外は File サーバへのアクセスを最小限としている．

5.3.1 管理プログラムの処理フロー

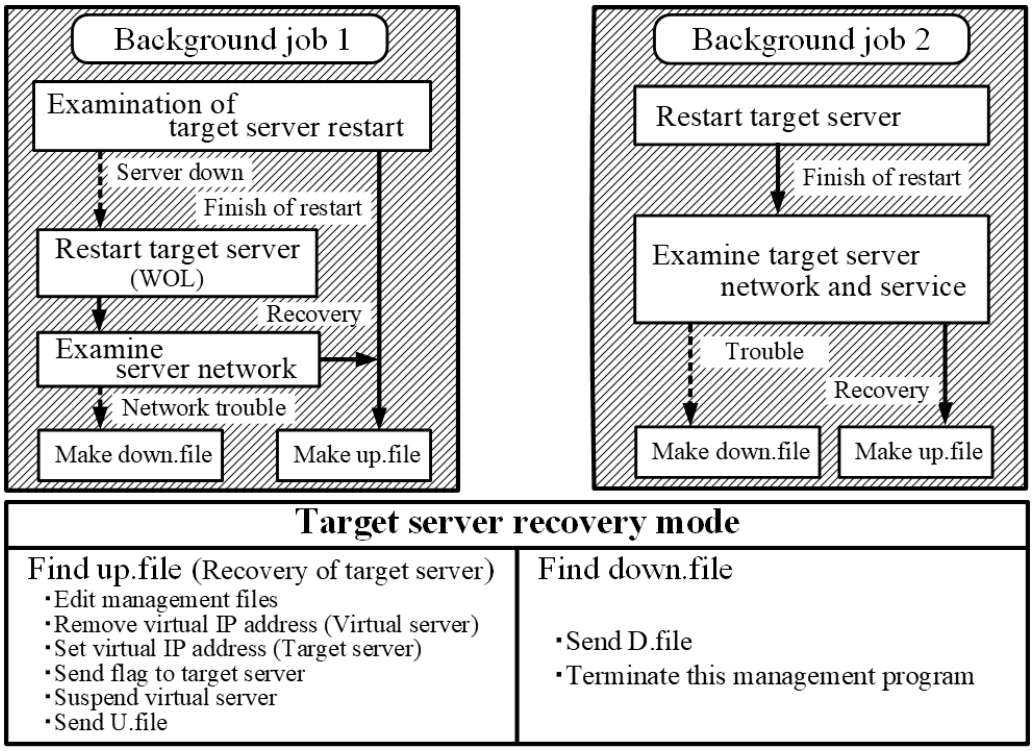
図 5.7 に管理サーバ上で動作する管理プログラムの動作フローを示す．(a)に示すように管理プログラムは 2 章で示したモード分割方式を採用し，Regular mode，Network examination mode，Service recovery mode，Backup server mode と Target server recovery mode の 5 つから構成される．ここでは，基本的に P1 で処理を行う管理サーバの動作フローを示す．管理プログラムは，Regular mode で管理対象サーバのネットワークとサービス提供状態の監視を定期的に行う．本システムではネットワークの監視に UNIX の

ping コマンドを使用している．また，サービス提供状態の監視には 2 章の図 2.4 で示した Expect プログラム言語を使用している．この言語はタイムアウト機能や管理対象サーバのサービスポートにアクセスし，その返答に応じた処理を行うことができるため，クライアントへのサービス提供状態を正確に監視することが可能となる．管理対象サーバの異常を検出した場合，図 5.6 で示した Group ファイルからそのサーバのデータを削除し，関連する管理ファイルの編集を行い，必要情報とともにその状況に応じたモードへと移行する．ネットワークに異常を検出した場合には，管理プログラムの制御状態を Network examination mode へ移行する．このモードではネットワークの調査を再度行い，この時点でネットワークが正常であれば，P2 で処理を行う管理サーバにネットワークが正常であることを伝達するための R.file を送信する．また，正常と判断されたサーバのデータを戻すために管理ファイルの編集を行い, Regular mode に戻る．ネットワークが異常の場合には，管理対象サーバの状態確認および復旧処理を行う Background job1 を実行し，管理ファイルをもとに新たな管理対象サーバに対する管理プログラムをバックグラウンドジョブとして実行する．Backup server mode ではサーバ機能を復旧するための仮想サーバを起動し，仮想 IP アドレスを設定してクライアントからのアクセスを仮想サーバに変更する．仮想サーバを起動する管理サーバの選択は図 5.5(b)で示した非選択値をもとに決められる．ここで，バックグラウンドジョブとして新たに実行する管理プログラムは P1 として実行され，この管理プログラムは新たな管理対象サーバに対し，この管理サーバを P2 として管理するプログラムを実行させる．この方式をここでは動的管理拡張方式と呼ぶ．また，Regular mode において，サービスに異常を検出した場合は，管理ファイルの編集を行い，その制御状態を Service recovery mode に移行し，管理対象サーバのネットワークを調査後，異常と判断したサービス提供プログラムの再起動を行う．その後，再度その状況調査を行う．この時点でサービス機能が復旧した場合には P2 で処理を行う管理サーバにサービスが正常に戻っ

たことを伝達するための R.file を送信し，その後，管理ファイルの編集を行い，Regular mode に戻る．復旧しない場合には管理対象サーバの仮想 IP アドレスの設定を解除後，管理対象サーバの復旧処理およびその状態の確認を行う Background job2 を実行する．管理ファイルをもとに新たな管理プログラムをバックグラウンドジョブとして実行後，制御状態を Backup server mode に移行する. Backup server mode での処理終了後, Target server recovery mode に移行して Background job1 または 2 の結果を待ち，その状況に応じた処理を行う.



(a) モード分割方式による様々なトラブル対処における動作フロー



(b) バックグラウンドジョブの役目

図 5.7　管理プログラムの動作フロー

(b)にバックグラウンドジョブの役目と Target server recovery mode の詳細な処理を示す．Background job1 と 2 は多くの時間を要する管理対象サーバの状態調査および復旧処理を行うため，バックグラウンドジョブとして実行される．Background job1 では管理対象サーバが再起動状態であるか調査を行う. 再起動が終了した場合には up.file を作成する. 再起動状態ではなく停止状態であれば, 管理対象サーバに対して図 5.4 で示した管理データに記録されている MAC アドレスを使用し, WOL を実行して起動を試みる. 起動の確認後, そのサーバのネットワークを調査し，復旧した場合には up.file を，復旧しない場合には down.file を作成して終了する. Background job2 ではサーバを再起動させ, 起動の確認後,

そのサーバのネットワークおよびサービス機能の調査を行う．サービス機能が復旧した場合には up.file を，復旧しない場合には down.file を作成して終了する．

Target server recovery mode では一定間隔時間毎にバックグラウンドジョブの結果である up.file または down.file の存在を確認する．up.file を検出した場合，管理対象サーバが復旧したと判断し，図 5.6 に示した管理ファイルを編集後，サーバ機能を復旧するために起動した仮想サーバに対して設定していた仮想 IP アドレスを解除し，管理対象サーバにその仮想 IP アドレスを設定する. さらに, 管理対象サーバで管理プログラムを起動させるため，Flag ファイルを送信する．その後，仮想サーバをサスペンド状態に変更し，P2 で処理を行う管理サーバに対して管理対象サーバが復旧したことを伝達するための U.file を送信後，Regular mode に制御状態を戻す．down.file を検出した場合，管理対象サーバが復旧不可能と判断し，P2 で処理を行う管理サーバに対して管理対象サーバが復旧不可能であることを伝達するための D.file を送信してこの管理プログラムを停止する．

5.3.2 管理の優先順位毎における動作フローチャート

図 5.8 に管理の優先順位毎における動作フローチャートを示す. ここでは 4 台のサーバのみに着目して説明を行う．S1 と S3 を管理サーバ，S2 を管理対象サーバとする．また，S2b は仮想サーバで S2 のバックアップサーバとする．(P1)と(P2)はそれぞれの優先順位において復旧処理を行っている状態を示す．S1 は P1，S3 は P2 として管理プログラムを実行しており，S1，S3 ともに，S2 のネットワークとサービス提供状態を定期的に監視する．本提案方式ではすべての管理サーバにおける監視タイミングの同期をとるために同期処理用プログラムを実行しており，図 5.6 に示した Group ファイルの先頭に記録された管理サーバから同期ファイルである S.file を一定間隔毎に一斉送信している．このため，P1 で管理している S1 と P2 で管理している S3 は同期をとりながら S2 の監視を行う．

S2 の異常を検出した場合，S1 は S2 の復旧処理を行うバックグラウンドジョブの実行および動的管理拡張方式により，S1 が P1 として S3 を，S3 が P2 として S1 を管理する新たな管理プログラムを実行する．その後，図 5.5(b)で示した非選択値が一番低いと判断された S4 上に S2b を起動するための制御を行い，仮想 IP アドレスを設定して S2 の機能を復旧する．その後，バックグラウンドジョブによって作成されるファイルを検出するために待機状態となる．S3 は S2 の異常を検出後，P1 で管理している S1 のネットワークとサービス提供状態を調査しながら S1 からの報告を待つ状態となる．もし，S1 が異常状態と判断された場合には S3 が P1 として S2 への処理を継続する．

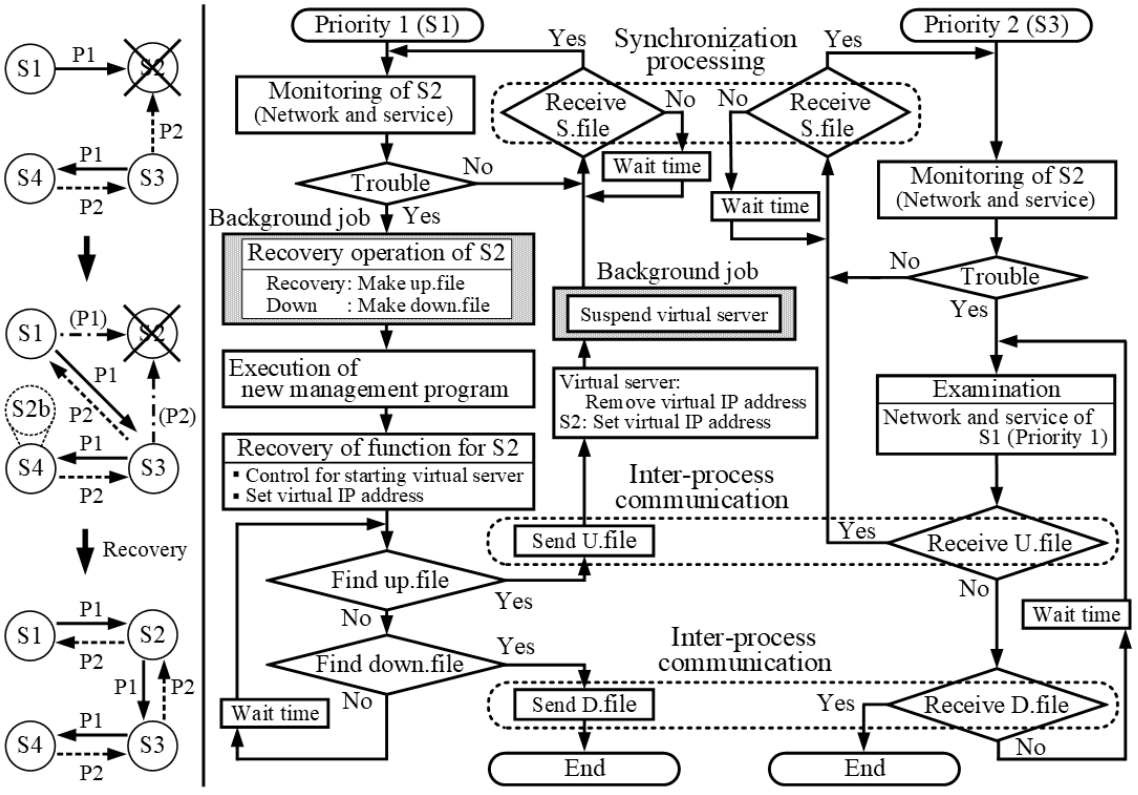


図 5.8　管理の優先順位毎の動作フローチャート

バックグラウンドジョブにより S2 が復旧したと判断されると up.file が作成される．S1 が up.file を検出すると，プロセス間通信によって S3 に対して U.file を送信し，S3 がそのファイルを受信することによって，S1，S3 ともに同期をとり S2 の監視状態に戻る．S1 は監視状態に戻る前に，S2b と S2 に対して仮想 IP アドレスの処理を行い，その後，S2b をサスペンド状態に変更する．バックグラウンドジョブにより S2 が復旧困難と判断されると down.file が作成される．S1 が down.file を検出すると，プロセス間通信によって S3 に対して D.file を送信し，S3 がそのファイルを受信することによって，S1，S3 ともにこの管理プログラムを停止する．

5.3.3 管理対象サーバの監視と負荷調査の動作タイミング

図 5.9 にサーバ管理（P2P）と負荷調査（Load）のタイムチャートを示す．サーバ管理と負荷調査は一定の間隔毎に，すべての管理サーバが同期を取りながらそれぞれの処理を行う．同期の取り方としては，図 5.8 で示した S.file の到着により行う．サーバ管理では，各管理サーバで同期をとりながら管理対象サーバの死活監視を行う．監視としてはネットワークとサービス提供状態の調査を行う．障害が検出されなかった場合は待機状態（監視間隔時間）となり，次の S.file の到着を待つ．また，管理対象サーバの障害を検出した場合は，そのサーバの復旧処理を行う．負荷調査では，同期後に一定間隔で負荷測定を 2 回行い，それぞれの平均値を求める．測定対象の負荷は図 5.5(a)で示したようにメモリ，CPU とネットワークの使用量とする．測定した負荷から図 5.5(a)で示した予測負荷を算出し，その後，仕様の異なるサーバ間での優劣を除くため，それらの負荷を正規化して Real server condition ファイルを作成する．サーバ管理と負荷調査は図に示すタイミングで動作し，復旧処理を行う場合には負荷調査で作成した最新の Real server condition ファイルを使用し，適切な管理サーバから仮想サーバを起動する．

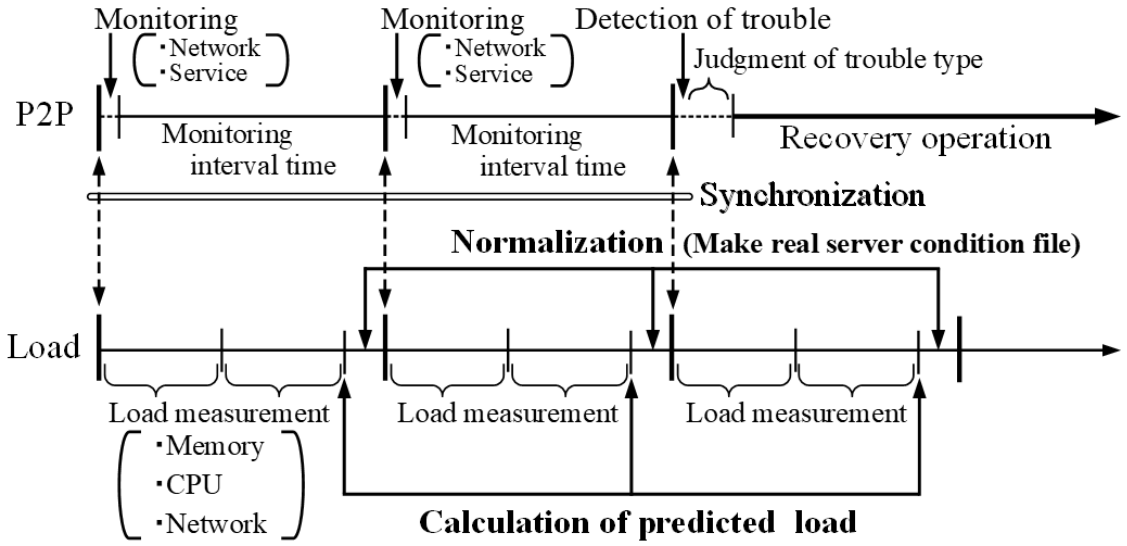


図 5.9　管理対象サーバの監視と負荷調査の動作タイミング

5.3.4　サーバ機能復旧方法および復旧後の対処

図 5.10 に管理対象サーバ故障時のサーバ機能復旧方法を示す．S1～S6 は管理サーバかつ管理対象サーバを示す．S1b，S2b，S3b と S4b はサーバ機能を復旧するための仮想サーバを示し，それらのシステムデータは File サーバに保存されている．各管理サーバは NFS 接続を使用し，サーバ機能復旧処理においてそれを使用する．

(a)に S2 が故障した場合の本方式におけるサーバ機能復旧方法を示す．P1 で管理している S1 が S2 の故障を検出した場合，動的管理拡張方式により S1 は P1 として S3 を，S3 は P2 として S1 を管理するプログラムを起動する．その後，S1 が図 5.5(b)で示した非選択値が一番低いと判断された S6 上に S2b を起動させ，仮想 IP アドレスの設定を行う．そのため，クライアントからのアクセスは S2b に切替わる．S1 によって S2 の復旧が困難と判断された場合は，(P1)と(P2)で動作している管理プログラムは終了される．ここで，S6 上で S1 を P1 として管理している管理プログラムは S2 の故障を検出していないため，S3 上

で新たに S1 を P2 として管理するプログラムが S6 上の管理プログラムにその変更を伝える．本システムでは前提として S2 の修理時間中のみ S2b がその役割を実行するため，S2b は管理対象外としている．

(b)に P1 と P2 で管理している管理サーバがサーバ機能の復旧処理を行う場合を示す. S2 の故障を S1 が検出し，S3 を P1 で管理している S2 が故障のため S3 の故障を P2 である S4 が検出している．S1 が図 5.5(b)で示した非選択値が一番低いと判断された S6 上に S2b を起動させ，この時点で非選択値が一番低い S4 が S3b を起動して，故障した S2 と S3 の機能を復旧する．また，故障検出と同時に S1 は P1 として S4 を，S4 は P2 として S1 の管理を開始する. S1 によって S2 が, S4 によって S3 が復旧困難と判断された場合, (P1)と(P2)で動作している管理プログラムは終了される．

(c)に S2，S3 と S4 が同時に故障した場合を示す．P1 である S1 が S2 を，S4 を P1 で管理している S3 が故障のため P2 である S5 が復旧処理を行う．S1 と S5 は故障検出と同時に S1 は P1 として，S5 は P2 として S3 の管理を行い，S3 の故障を検出し，復旧処理を行う．ここで，S1 は非選択値が一番低いと判断された S6 上に S2b を起動させ，次にこの時点で非選択値が一番低い S5 が S4b を起動している．さらに，この時点で仮想サーバを起動していない S1 上に S3b を起動するように制御が行われる．S1 が S3 の故障検出と同時に S1 は P1 として S5 を，S5 は P2 として S1 の管理を開始する．

本方式では，管理対象サーバは管理サーバの機能を持つことから，管理対象サーバがサーバ機能を復旧するために仮想サーバを起動している場合が考えられる．(d)に S2b を起動している S1 が故障した場合を示す. P1 である S6 はこの時点で非選択値が一番低いと判断されたため S1b を起動し，それと同時にこの時点で非選択値が一番低い S3 から S2b を起動するよう制御を行う．ここで，S2b のシステムデータは File サーバ上に保存されているため，S3 からの起動が可能となる．

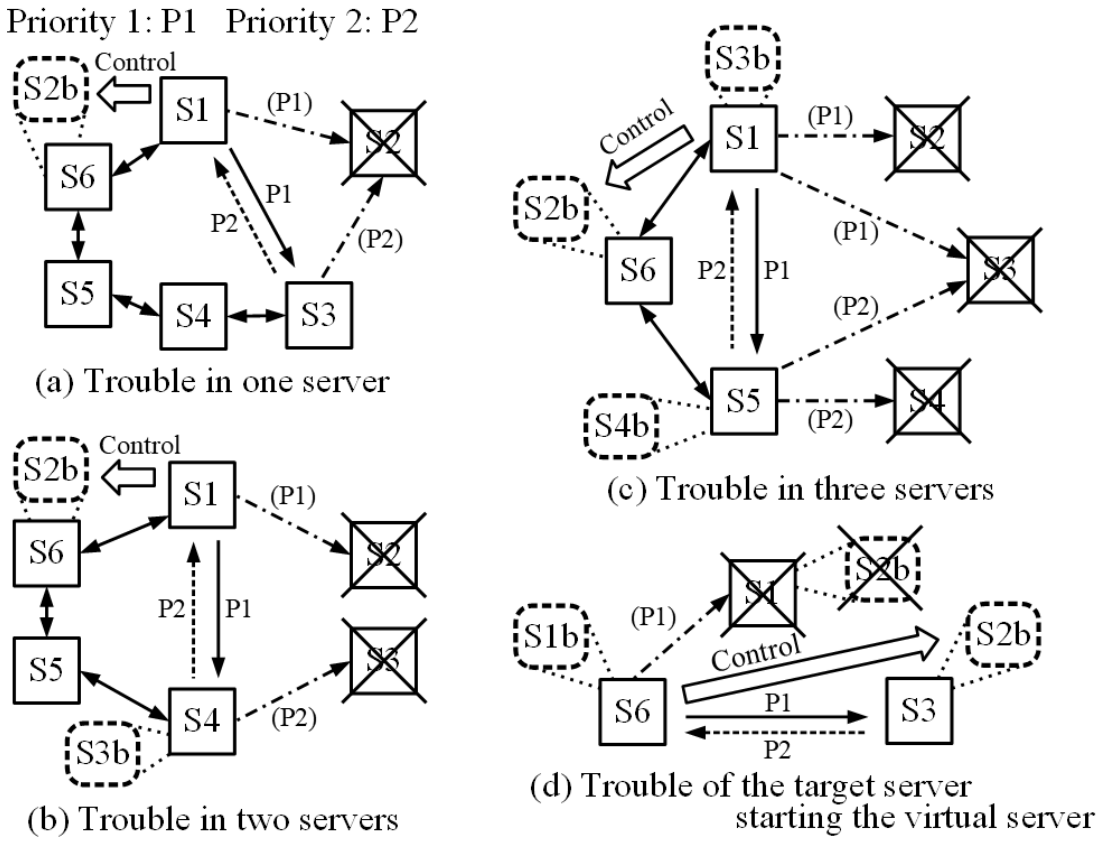


図 5.10　管理対象サーバ機能の復旧方法

実稼働しているサーバシステムにおいて，システム内の半分以上のサーバが同時故障をする確率は非常に少ないと考えられるが，本システムはそのような場合においても対処が可能である．例えば，管理対象サーバ 6 台中 5 台が同時に故障した場合，稼働しているサーバが故障したサーバの機能を復旧するために 5 台の仮想サーバを起動し，仮想 IP アドレスを設定することにより可能となる．ただし，稼働しているサーバがそれに対処可能となる十分なリソースを有していることが条件となる．

図 5.11 に，図 5.10(b)と(d)に示した状態から管理対象サーバが復旧した場合の処理手順を示す．四角で囲まれた P1 と P2 は，復旧した管理対象サーバの Executer システムに対

して，(P1)で処理を行っている管理サーバが Flag ファイルを送信することで起動した管理プログラムを示す．(a)の S2 と S4 を例とすると，四角で囲まれた P1 のプログラムを実行する管理サーバ S2 は管理対象サーバ S4 上において，S2 に対して P2 で管理するプログラムを実行させる．また，その管理対象サーバ上において，S2 以外に対して P2 の管理プログラムを実行していた場合には停止させる．

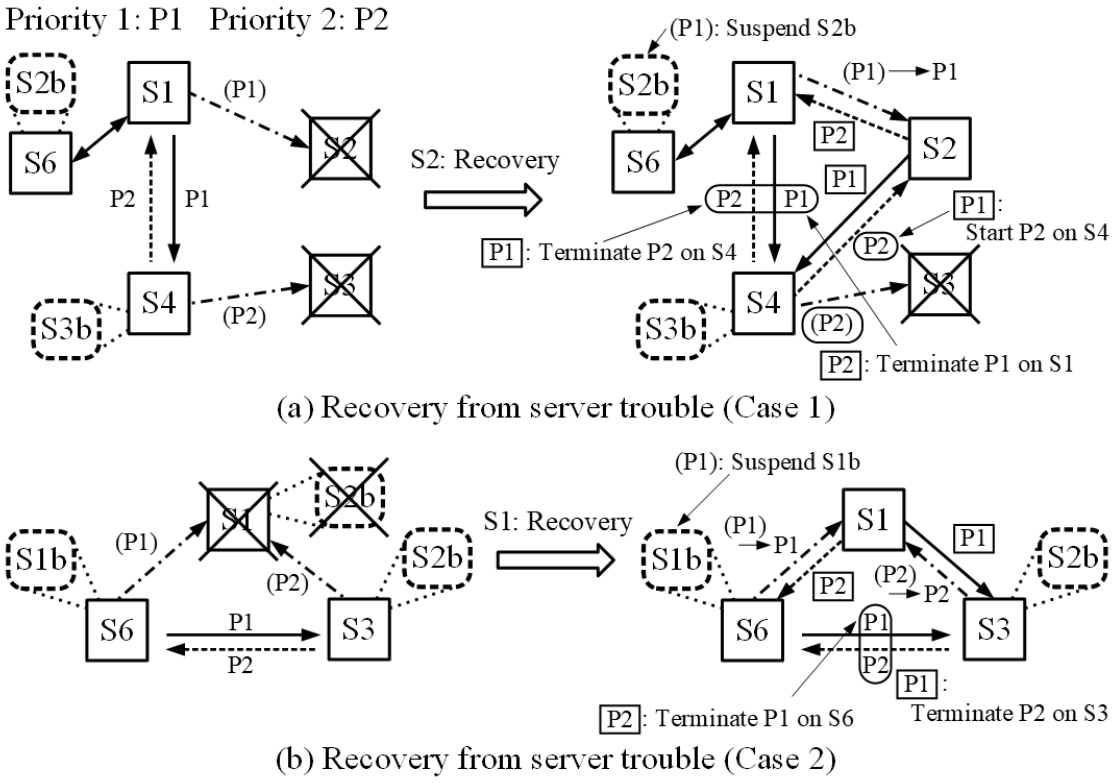


図 5.11　管理対象サーバ復旧後の対処

(a)に S2 と S3 が同時故障し，その後，S2 のみが復旧した場合を示す．(P1)は S2 と S2b に対して仮想 IP アドレスの処理を行い，S2b をサスペンドする．S2 から起動した P1 の管理プログラムは S4 に対して S2 を P2 で管理するプログラムを起動させ，S1 を P2 で管理しているプログラムを停止する．また，S3 を P2 で管理しているプログラムも停止対象と

なるが，復旧処理中であることから停止処理は行わない．S2 から起動した P2 の管理プログラムは S1 上で S4 を P1 として管理しているプログラムを停止する．その後，復旧処理状態の(P1)は管理状態の P1 に戻る．

(b)に S2b を起動している S1 が故障し，その後 S1 が復旧した場合を示す．(P1)は S1 と S1b に対して仮想 IP アドレスの処理を行い，S1b をサスペンドする．S1 から起動した P1 の管理プログラムは S3 上で S6 を P2 として管理しているプログラムを停止する．S1 から起動した P2 の管理プログラムは S6 上で S3 を P1 として管理しているプログラムを停止する．復旧処理状態の(P1)と(P2)はそれぞれ管理状態の P1 と P2 に戻る．ここで，S2b に関しては処理時間を考慮し，S1 からの起動に戻さず S3 からの起動状態としている．

5.4 P2P 方式サーバ管理システムの動作実験およびその評価

5.4.1 実験システムの構成および仕様

P2P 方式を採用したサーバ管理実験システムの構成を図 5.12 に示す．実験システムは管理サーバかつ管理対象サーバとなる Mail，Login，FTP，Proxy，Web1 と Web2 の 6 台，File1 と File2 サーバが 1 台ずつ，クライアントが 1 台，1000BASE-T と 100BASE-TX の機能を有するルータが 1 台，1000BASE-T のスイッチング HUB が 1 台と 100BASE-TX のスイッチング HUB が 1 台から構成される．ここで，File1 サーバは P2P 方式サーバ管理システムで使用し，File2 サーバはサービス提供用に使用する．実験システムのネットワーク環境としては，一般的な構成としてサーバ間のネットワーク転送速度に対してクライアントからのネットワーク転送速度を 10 分の 1 としている．各管理サーバは File1 サーバと NFS 接続され，図 5.3 で示した Executer プログラムがインストールされている．各管理サーバは起動と同時に Executer プログラムが自動実行され，ローカルエリアにある Flag ファイルの有無を調査する．

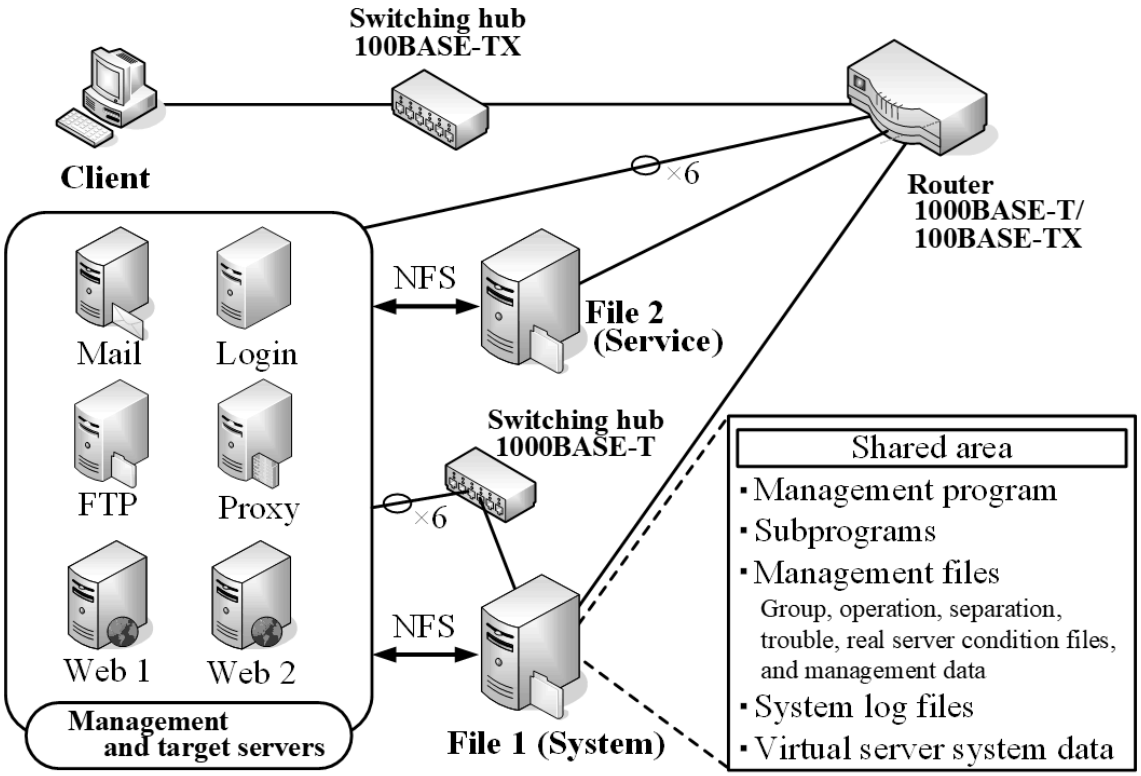


図 5.12　実験システムの構成

File1 サーバ上の共有エリアには各管理サーバが実行するための管理プログラム，サブプログラム，図 5.6 で示した Group ファイル，Operation ファイル，管理対象サーバ故障時に編集される Separation ファイルや Trouble ファイル，Real server condition ファイル，図 5.4 で示した管理データおよび各管理サーバが作成する System log ファイルやサーバ故障時にそのサーバ機能を復旧するための仮想サーバ用システムデータが保存されている．実験システムにおいて，各管理サーバ上の Flag ファイルは，File1 サーバによって作成または削除が実行される．管理サーバのローカルエリアに Flag ファイルが作成されると，Executer プログラムは Operation ファイルにもとづき管理プログラムを実行するとともに同期処理プログラムとサーバ負荷測定プログラムからなるサブプログラムを実行する．その Flag ファイルが削除されると，すべての管理プログラムとサブプログラムを停止する．

本実験システムでは負荷測定に関して UNIX の sar コマンドを使用し，サーバ監視間隔時間内に負荷測定を行い，監視時には図 5.6 で示した Real server condition ファイルを最新の状態にしている．

表 5.3 に実験システムの仕様, 基礎資料として各サーバの特性およびシステム動作時の設定時間を示す．管理サーバ 6 台はすべて同様の仕様で，CPU は Intel Core i7-3770，仮想サーバの起動を考慮して物理メモリは 8,192MB，仮想化ソフトウエアには Linux で標準的に使用される KVM を採用する．各管理サーバは CUI 環境で動作させるため，仮想サーバはディスプレイにウインドウが表示されない状態で起動する．File1 サーバは CPU に Intel Core i7-4770，物理メモリは 8,192MB とする．仮想サーバは，CPU 数を 1，メモリを 2,048MB と設定している．各サーバの OS にはサーバ用の OS として採用されることが多い CentOS の 64bit 版をインストールする．基礎資料として各サーバの特性を示すが，これらは測定を 10 回行った結果の平均値とする．時間の実測には UNIX の time コマンドを使用する．管理サーバの起動時間を Trs とし，56.67 秒，再起動時間を Trr とし，58.51 秒となった．また，管理対象サーバのネットワークが正常時におけるネットワーク調査時間は 0.01 秒となった．サービス調査時間 Tse は提供サービスの種類によって異なり，FTP サーバは 0.06 秒，Login サーバは 0.23 秒，Mail サーバは 0.36 秒，Proxy サーバは 0.59 秒となり Web サーバは 1.59 秒となった．仮想サーバにおいては，通常の起動時間が 42.84 秒でサスペンド状態からの起動時間 Tv は 4.67 秒となった．この結果より，本システムではサーバ機能復旧時における仮想サーバの起動時間を短縮するため，仮想サーバは常にサスペンド状態のまま待機させる．また，本システムでは管理対象サーバの監視間隔時間を Tim とし，10.00 秒，管理対象サーバのネットワークが異常時におけるネットワーク調査時間を Thd とし，2.00 秒，サービス調査における最大待ち時間 Tsw は Tse の最大値が 1.59 秒より 3.00 秒，管理対象サーバが再起動状態か判断する最大時間を Trc とし，Trr の約 1.5 倍

である 88.00 秒と設定している．

表 5.3　実験システムの仕様および特性や設定時間

Specifications	Management server (Target server)	File 1 (System)
CPU	Intel Core i7-3770 3.40 GHz (TB: 3.90 GHz)	Intel Core i7-4770 3.40 GHz (TB: 3.90 GHz)
Memory	8,192 MB	
Hard disk	SATA 2 (7,200 rpm)	
Virtualization software	KVM 1.5.3	
System software	postfix, sshd, httpd, vsftpd, squid	nfsd
OS	CentOS 7.3 (64-bit)	

Specifications	Virtual server
CPU	1
Memory	2,048 MB
OS	CentOS 7.3 (64-bit)

Characteristics	Management server (Target server)	
Start time (s)	T_{rs}	56.67
Restart time (s)	T_{rr}	58.51
Network examination time (s)		0.01
Service examination time (s)	T_{se}	0.06 - 1.59

Characteristics	Virtual server	
Start time (s)		42.84
Time to activate suspended server (s)	T_{v}	4.67

Setting parameters		
Monitoring interval time (s)	T_{im}	10.00
Network examination time (Disconnection) (s)	T_{nd}	2.00
Maximum waiting time for service examination (s)	T_{sw}	3.00
Maximum judgment time for restart (s)	T_{rc}	88.00

5.4.2　管理対象サーバ故障時の実験結果

図 5.12 に示した実験システムの Mail と Web1 サーバが管理を行う FTP サーバにトラブルを再現する実験を行う．監視間隔時間を 10 秒とし，管理対象サーバ FTP に対して監視間隔時間が 5 秒経過時にサービス提供プログラムやネットワークに関するトラブルを再現する．それらの対処に要する復旧時間を実測し，図 5.13 に示す．各実測時間は 10 回行った結果の平均値とする．ここでは，クライアントからのアクセスがない状態で，各管理サーバ上において UNIX の stress コマンドを使用し, 負荷として Login サーバには 512MB, その他のサーバには 1GB のメモリを使用するように設定している．サービスに関するトラブルは，FTP サーバのサービス提供プログラムの停止を行う．また，ネットワークに関す

るトラブルは，FTP サーバを再起動またはシャットダウンすることにより再現する．図に示すように，トラブルは St1，St2，Nt1 と Nt2 の 4 種類を想定し，St1 は管理対象サーバにおいてサービス提供プログラムを停止し，それの再起動で復旧可能な場合，St2 はその再起動では復旧が困難であり，管理対象サーバを再起動することで復旧可能な場合，Nt1 は管理対象サーバを意図的に再起動した場合，Nt2 は管理対象サーバがシャットダウンした場合とする．ここで，Nt1 は，サーバの稼働状態が異常な場合にサーバ管理者がリセットボタンを押すなど，そのサーバを再起動することを想定している．測定時間に関して，サーバ機能の復旧時間は仮想サーバによりサーバ機能を復旧するまでの時間，管理対象サーバの復旧時間は管理対象サーバが正常状態に復旧するまでの時間を示し，クライアントにおけるアクセス切断時間はクライアントから 1 秒間隔でサーバに対してアクセスを行い，それが切断されてから復旧するまでの時間を示す．実測値として，サーバ機能の復旧時間および管理対象サーバの復旧時間は付録の F6 で示す本システムが記録している Log ファイルからの測定方法を採用している．括弧で囲まれていない数値は実測値を示し，括弧で囲まれた数値は表 5.3 で示した測定時間や設定値から算出する予測値を示す.

サーバ機能の復旧時間において，St2 のトラブルに対する復旧時間は Ts2f，Nt1 のトラブルに対する復旧時間は Th1f，Nt2 のトラブルに対する復旧時間は Th2f とする．St2 のトラブル時におけるモードの流れは Regular mode → Service recovery mode → Backup server mode → Target server recovery mode → Regular mode となり，Ts2f はサービスの異常を検出後, Service recovery mode においてサービス調査における最大待ち時間 Tsw および Backup server mode における仮想サーバの起動時間 Tv が主な構成要素となり，予測値は式(5.10)で示される.

$$T\text{s2f [s]} = T\text{sw} + T\text{v} \tag{5.10}$$

Ts2f の実測値は 8.56 秒となり予測値は 7.67 秒となった.

Nt1 と Nt2 のトラブル時におけるモードの流れは Regular mode → Network examination mode → Backup server mode → Target server recovery mode → Regular mode となり，Th1f と Th2f はネットワーク異常を検出後，Network examination mode においてネットワーク接続調査時間 Thd および Backup server mode における仮想サーバの起動時間 Tv が主な構成要素となり，予測値は式(5.11)で示される．

$$T\text{n1f [s]} = T\text{n2f} = T\text{nd} + T\text{v} \tag{5.11}$$

Th1f の実測値は 7.47 秒，Th2f の実測値は 7.46 秒となり，ともに予測値は 6.67 秒となった．サーバ機能の復旧時間においては，測定時間と予測時間の差は 1 秒未満となった．

FTP サーバの復旧時間において，St1 のトラブルに対する復旧時間は Ts1t，St2 のトラブルに対する復旧時間は Ts2t，Nt1 のトラブルに対する復旧時間は Th1t，Nt2 のトラブルに対する復旧時間は Th2t とする．St1 のトラブル時におけるモードの流れは Regular mode → Service recovery mode → Regular mode となり，Ts1t はサービスの異常を検出後，Service recovery mode においてサービス提供プログラムを再起動し，その後，サービス再調査を行うことからサービス調査時間 Tse が主な構成要素となり，予測値は式(5.12)で示される．

$$T\text{s1t [s]} = T\text{se} \tag{5.12}$$

Ts1t の実測値は 0.31 秒となり予測値は 0.06 秒となった．St2 のトラブルにおいては FTP サーバを再起動することにより，そのサーバのサービス提供状態を復旧することから再起動時間の Trr が含まれ，予測値は式(5.13)で示される．

$$T\text{s2t [s]} = T\text{sw} + T\text{rr} + T\text{se} \tag{5.13}$$

Ts2t の実測値は 62.98 秒となり予測値は 61.57 秒となった．Nt1 のトラブルにおいては監視間隔時間が 5 秒経過時に FTP サーバを再起動させることから，ネットワークトラブルを検出するまでに Tim/2+Thd の時間を要しているため，予測値は式(5.14)で示される．

$$T\mathrm{n1t}\ [\mathrm{s}] = T\mathrm{rr} + T\mathrm{se} - (\frac{T\mathrm{im}}{2} + T\mathrm{nd}) \quad (5.14)$$

Th1t の実測値は 52.61 秒となり予測値は 51.57 秒となった. Nt2 のトラブルにおいては FTP サーバが再起動状態か判断する最大時間 Trc とそのサーバの起動時間 Trs が含まれ，予測値は式(5.15)で示される.

$$T\mathrm{n2t}\ [\mathrm{s}] = T\mathrm{nd} + T\mathrm{rc} + T\mathrm{rs} + T\mathrm{se} \quad (5.15)$$

Th2t の実測値は 146.76 秒となり予測値は 146.73 秒となった. FTP サーバの復旧時間において，測定時間と予測時間の差は約 1 秒となった．St2 では FTP サーバを再起動し，そのサービス機能を復旧することからサーバ再起動時間 Trr の影響が大きい．Nt2 では FTP サーバが再起動状態ではないことを確認後，WOL を使用してそのサーバを起動することから，その確認を行うための最大時間 Trc とサーバの起動時間 Trs の影響が大きいことがわかる.

ここで，サーバ機能の復旧時間および FTP サーバの復旧時間において，実測値と予測値には約 1 秒程度の差が見られるが，この差は予測値の構成要素に含まれていない 1 秒未満で終了する要素の合計が影響していると考えられる.

クライアントからのアクセス切断時間において, St1 のトラブルに対する切断時間は Ts1c, St2 は Ts2c, Nt1 は Th1c, Nt2 は Th2c とする. それぞれの切断時間には監視間隔時間（Tim/2 ± Tim/2）を含む．Ts1c の予測値は式(5.16)，Ts2c の予測値は式(5.17)，Th1c と Th2c の予測値は式(5.18)で示される.

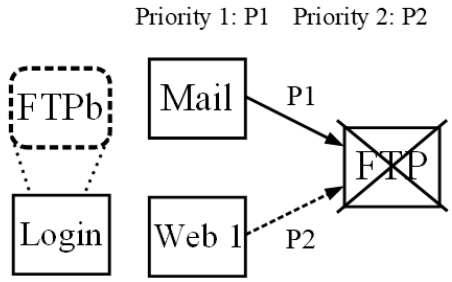
$$T\mathrm{s1c}\ [\mathrm{s}] = T\mathrm{s1t} + \frac{T\mathrm{im}}{2} \pm \frac{T\mathrm{im}}{2} \quad (5.16)$$

$$T\mathrm{s2c}\ [\mathrm{s}] = T\mathrm{s2f} + \frac{T\mathrm{im}}{2} \pm \frac{T\mathrm{im}}{2} \quad (5.17)$$

$$T\mathrm{n1c}\ [\mathrm{s}] = T\mathrm{n2c} = T\mathrm{nd} + T\mathrm{n1f} + \frac{T\mathrm{im}}{2} \pm \frac{T\mathrm{im}}{2} \quad (5.18)$$

ここで，Ts1c に関しては，監視間隔時間の範囲以外にネットワーク確認，サービス確認およびサービス再起動の処理が含まれるがそれらの合計は 1 秒未満となる．Ts1c の実測値は 5.36 秒となり予測値は 5.06±5.00 秒, Ts2c の実測値は 13.24 秒となり予測値は 12.67±5.00

秒，T_{h1c} の実測値は 13.73 秒となり予測値は 13.67±5.00 秒，T_{h2c} の実測値は 14.14 秒となり予測値は 13.67±5.00 秒となった. いずれのトラブルにおいてもアクセス切断時間の実測値は予測時間の範囲に含まれていることがわかる.



St1: Service program trouble (Recovery by service restart)
St2: Service program trouble (Recovery by server restart)
Nt1: Intentional restart of target server
Nt2: Shutdown of target server

		Target server: FTP　Value : Measured time (s) (Value) : Predicted time (s)					
Trouble list		Recovery time of server function		Recovery time for target server		Access disconnection time (Client)	
Service program	St1			T_{s1t}	0.31 (0.06)	T_{s1c}	5.36 (5.06±5.00)
	St2	T_{s2f}	8.56 (7.67)	T_{s2t}	62.98 (61.57)	T_{s2c}	13.24 (12.67±5.00)
Network	Nt1	T_{n1f}	7.47 (6.67)	T_{n1t}	52.61 (51.57)	T_{n1c}	13.73 (13.67±5.00)
	Nt2	T_{n2f}	7.46 (6.67)	T_{n2t}	146.76 (146.73)	T_{n2c}	14.14 (13.67±5.00)

図 5.13　1 台故障時における復旧時間の実測値と予測値

ここで，図 1.2 で示した稼働率による評価を表 5.4 に示す．図 1.1 で示した TPUCELS において，コストの関係上，冗長化を行っていないサーバでトラブルが発生した場合，その故障サーバ自体の復旧が求められる．そこで，管理対象サーバ自体を復旧する方式を従来方式として比較を行う．従来方式としての t_i には図 5.13 に示す管理対象サーバの復旧時間を，改善システムとしての t_{bi} にはサーバ機能の復旧時間を使用する．T_i には管理対象サーバの復旧時間における最大値を使用し，その値での従来方式の稼働率（Conventional

sysytem）を 0.5 とする．また，提案方式による稼働率の改善（Improvement）を式(2.10)で算出する．

St2 のトラブルの場合, 従来方式の稼働率は 0.70, 提案方式の稼働率（Proposed system）は 0.96 となり 37.14%の改善がみられる．また，Nt1 のトラブルの場合，従来方式の稼働率は 0.74，提案方式の稼働率は 0.96 となり 29.73%の改善がみられる．さらに，Nt2 のトラブルの場合，従来方式の稼働率は 0.50，提案方式の稼働率は 0.97 となり 94.00%の改善がみられている．提案方式は従来方式に比べ稼働率の面において非常に高い優位性が認められる．

表 5.4　提案方式による稼働率の改善

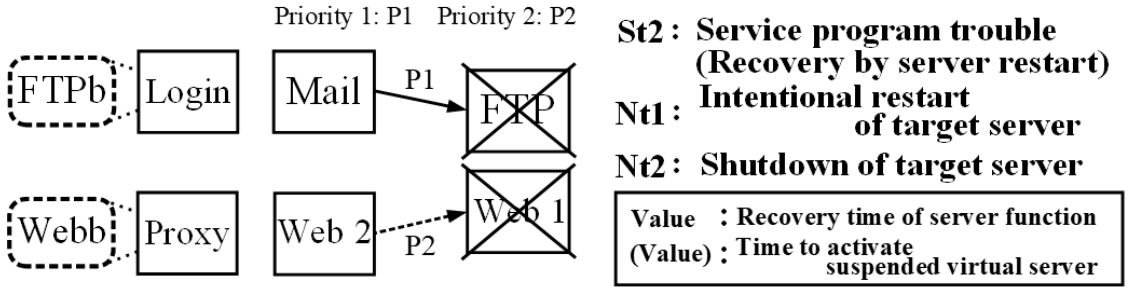
St2: Service program trouble (Recovery by server restart)
Nt1: Intentional restart of target server
Nt2: Shutdown of target server

		Conventional system	Proposed system	Improvement (%)
Availability rate	St2	0.70	0.96	37.14
	Nt1	0.74	0.96	29.73
	Nt2	0.50	0.97	94.00
	Average	0.65	0.96	47.69

図 5.14 に FTP と Web1 サーバの 2 台同時故障における復旧時間を示す．ここでは，それらのサーバに対してサービス提供プログラムおよびネットワークに関するトラブルを 2 台同時に再現し，P1 の管理サーバである Mail サーバおよび P2 の管理サーバである Web2 サーバによるサーバ機能の復旧時間およびそのために使用した仮想サーバの起動時間を示

す．仮想サーバの起動時間はサスペンド状態からの起動時間を示す．

ここでは，負荷として Login と Proxy サーバには 512MB，その他のサーバには 1GB のメモリを使用するように設定している．それぞれの管理サーバにおいて監視間隔時間を 10 秒とし，管理対象サーバに対して監視間隔時間が 5 秒経過時にサービス提供プログラムやネットワークに関するトラブルを再現する．トラブルは図 5.13 で示した St2，Nt1 と Nt2 で，ここでは 2 台の管理対象サーバに対してそれぞれのトラブルを組み合わせ，9 通りの実験を行う．括弧で囲まれていない数値はサーバ機能の復旧時間を示し，括弧で囲まれている数値は仮想サーバの起動時間を示す．



Measured time (s)		Target server: Web 1		
		St2	Nt1	Nt2
Target server: FTP	St2	11.49 (7.84) 11.55 (7.82)	10.27 (7.72) 10.85 (7.17)	10.17 (7.70) 10.61 (6.94)
	Nt1	10.79 (7.15) 10.13 (7.65)	10.84 (8.18) 10.25 (7.70)	10.72 (8.06) 10.11 (7.57)
	Nt2	11.01 (7.35) 10.09 (7.61)	10.84 (8.31) 10.31 (7.73)	10.61 (8.06) 10.25 (7.69)

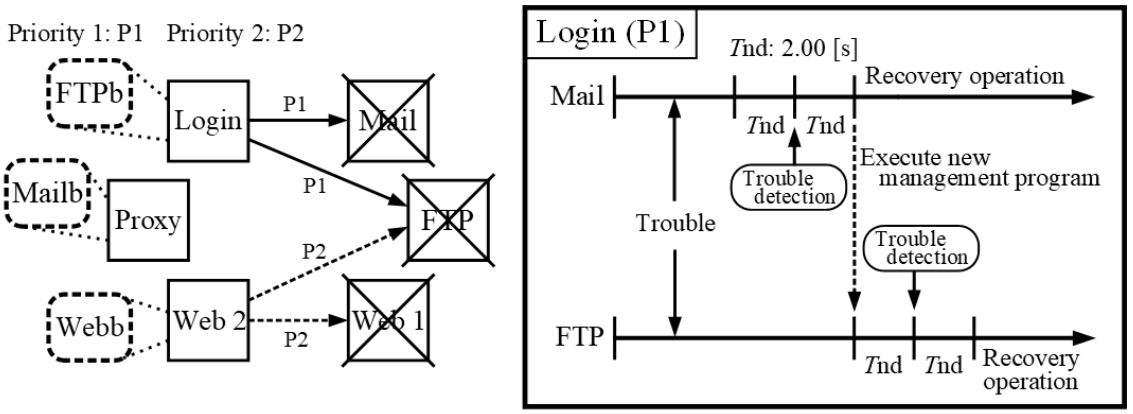
図 5.14　2 台同時故障における復旧時間

それぞれのサーバにSt2のトラブルを同時に再現した場合, サーバ機能の復旧時間はFTPサーバが 11.55 秒，Web1 サーバが 11.49 秒となり，仮想サーバ FTPb の起動時間は 7.82 秒, Webb は 7.84 秒となった. それぞれのサーバに Nt1 のトラブルを同時に再現した場合，サーバ機能の復旧時間は FTP サーバが 10.25 秒，Web1 サーバが 10.84 秒となり，仮想サーバ FTPb の起動時間は 7.70 秒，Webb は 8.18 秒となった．それぞれのサーバに Nt2 のトラブルを同時に再現した場合，サーバ機能の復旧時間は FTP サーバが 10.25 秒，Web1 サーバが 10.61 秒となり，仮想サーバ FTPb の起動時間は 7.69 秒，Webb は 8.06 秒となった．いずれの場合もサーバ機能の復旧時間において，図 5.13 の結果よりその時間は長くなっている．これは，ほぼ同時に 2 台の仮想サーバを起動するため，表 5.3 で示した Tv より起動時間が長くなったためと考えられる．

図 5.15 に Mail，Web1 と FTP サーバの 3 台同時故障における復旧時間を示す．ここでは，それらのサーバに対して図 5.13 で示した Nt2 のトラブルを 3 台同時に再現し，P1 の管理サーバである Login サーバおよび P2 の管理サーバである Web2 サーバによるサーバ機能の復旧時間，そのために使用した仮想サーバの起動時間とクライアントにおけるアクセス切断時間を示す．仮想サーバの起動時間はサスペンド状態からの起動時間を示す．ここでは，負荷として Proxy サーバには 512MB，Web2 サーバには 1GB，Login サーバには 2GB のメモリを使用するように設定している. 各サーバにおける復旧時間は故障を検出後，復旧するまでの時間とする．また，それぞれの管理サーバにおいて監視間隔時間を 10 秒とし，管理対象サーバに対して監視間隔時間が 5 秒経過時にトラブルを再現する．図に示すように 3 台同時故障の場合，本提案システムでは P1 である Login サーバは調査開始後，Thd の時間を要して Mail サーバの故障を検出し，さらに，Thd の時間を要し，その故障を再確認している．その後，FTP サーバに対して P1 の管理プログラムを実行する．そのプログラムにおいて Thd の時間経過後に FTP サーバの故障を検出する．

サーバ機能の復旧時間は，Mail サーバが 11.17 秒，Web1 サーバが 11.39 秒で FTP サーバは 9.95 秒となり，仮想サーバ Mailb の起動時間は 8.50 秒，Webb は 8.50 秒，FTPb は 7.36 秒となった. クライアントでのアクセス切断時間は, Mail サーバが 17.59 秒, Web1 サーバが 17.99 秒で FTP サーバは 20.71 秒となった.

ここで，図 5.13 の結果と比較すると各サーバ機能の復旧時間が約 4 秒長くなっており，仮想サーバの起動時間も表 5.3 で示した T_V より約 4 秒長くなっている．これは，仮想サーバの起動処理が重なったためと考えられる．仮想サーバの起動時間において FTPb が約 1 秒短いのは，約 4 秒遅れでその起動処理が開始されることから，その重なりが少なくなったためと考えられる．さらに，クライアントでのアクセス切断時間での Mail サーバと FTP サーバとの差は故障検出時刻の差によるものと考えられる.



Trouble: Nt2 (Shutdown of target server)

Target server	Recovery time of server function (s)	Time to activate suspended virtual server (s)	Access disconnection time (Client) (s)
Mail	11.17	8.50	17.59
Web 1	11.39	8.50	17.99
FTP	9.95	7.36	20.71

図 5.15　3 台同時故障における復旧時間

表 5.5 に冗長化構成を構築する上で管理対象サーバの台数に応じた予備用実サーバの台数を示す．本システムは表 5.1 に示すように，管理対象サーバの台数に依存しない管理方式を採用している．しかし，実験システムの File1 サーバの仕様（OS 上で CPU コア数を 8 と認識）では付録 F7 に示すように NFS 接続を 256 台（32×8）までに設定するのが最適となるため，管理対象サーバかつ管理サーバは 256 台までに設計することが望ましい．提案方式を“P2P”とし，従来方式としては図 1.1 で示した TPUCELS において，コストの関係上，冗長化を行っていないサーバで構成されるサーバシステムを“CS”として示す．提案方式は最低 2 台以上からの構成となるため管理対象サーバは 2 台から開始する．

表 5.5　予備用実サーバの台数比較

CS: Conventional system

	The number of target servers ($n \leqq 256$)					
	2	**3**	**4**	**・・・**	***n*-1**	***n***
CS	2	3	4	・・・	n-1	n
P2P	2	2	2	・・・	2	2

CS は管理対象サーバの台数に応じて予備用実サーバの台数が比例して増加する．P2P は予備のサーバとして仮想サーバを使用するため，管理対象サーバの台数に応じて予備用実サーバの台数は増加しない．ただし，P2P は専用の File サーバを使用することから，その File サーバが故障した場合，管理対象サーバの管理ができなくなる．そこで，その File サーバは冗長化する必要がある．そのため，P2P は管理対象サーバの台数に関係なく 2 台の

状態を維持している．ここで，予備用実サーバの台数に比例して初期費用は高くなり，関連して消費電力や保守料金などの運用コストも台数に比例して高くなる．例えば，その File サーバを 2 台で 100 万円，予備用実サーバを 1 台 30 万円とし，管理対象サーバが 100 台の場合には，P2P は初期費用だけで 2,900 万円のコストを抑えることができ，さらに，台数に比例した運用コストも抑えることができる．以上の結果から，提案方式は従来方式に比べコスト面においても優位性が認められる．

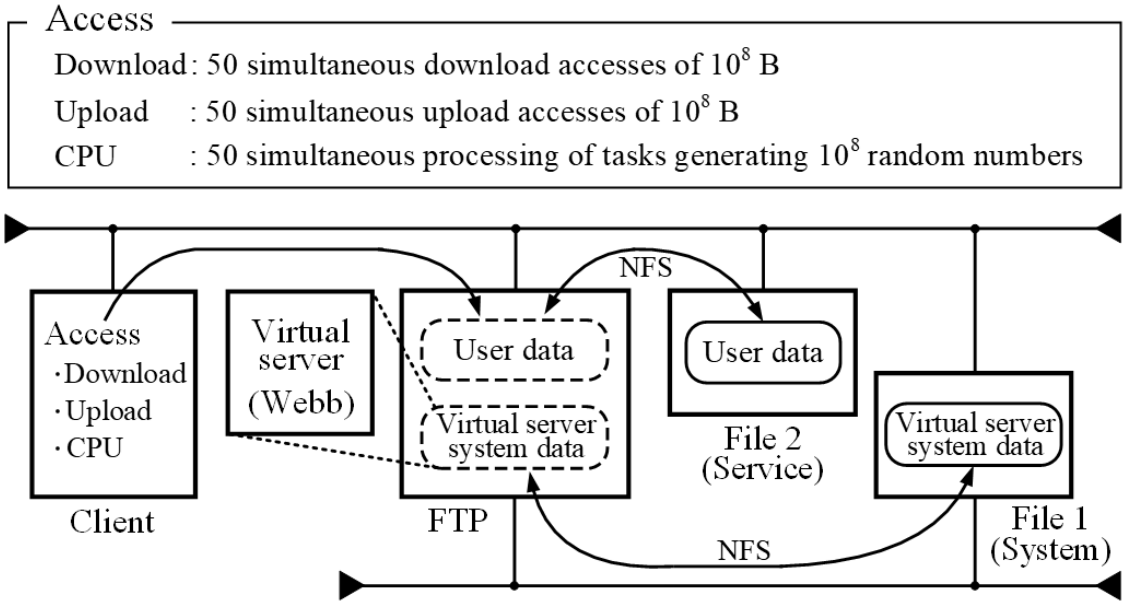
5.4.3 クライアントアクセスの影響

本システムにおいて，仮想サーバ用システムデータは File1 サーバに，ユーザデータは File2 サーバに保存されている．これは，管理サーバにクライアントからアクセスがある場合においても，管理対象サーバの機能を復旧するために起動する仮想サーバの起動時間への影響を低くするために考慮されている．しかし，本システムを実稼働するためにはその影響を調査しておく必要がある．図 5.16 に仮想サーバを起動する管理サーバに対し，クライアントからアクセスがある場合における仮想サーバの起動時間への影響を示す．

クライアントからは管理サーバである FTP サーバに対し，ダウンロードまたはアップロードアクセスおよび CPU 処理による負荷を与える．FTP サーバにおいてダウンロードおよびアップロードにおけるユーザデータへのアクセスは，図に示すように NFS 接続を経由して File2 サーバにアクセスされる．また，仮想サーバを起動する場合には，図に示すように NFS 接続を経由して File1 サーバにアクセスし，そこに保存されている仮想サーバ用システムデータを使用し，実行される．

実験においてダウンロード，アップロードや CPU による負荷を与えた状態において仮想サーバの起動時間を測定する．ダウンロードとアップロードでは 10^8B のデータ 50 個を同時にアクセスする．また，CPU 処理では 10^8 回の乱数演算を 50 個同時に実行する．クライ

アントからのアクセスがない場合の起動時間は, 表 5.3 に示すように T_V は 4.67 秒となる．ダウンロードによる負荷を与えた場合には 4.82 秒，アップロードでは 4.79 秒となり CPU 処理では 6.78 秒となった．仮想サーバの起動に関して，ダウンロードおよびアップロードに関してはほぼ影響がなく，CPU 処理による影響としては約 2 秒の増加が見られる．しかし，サーバ故障の頻度やサーバシステム構築時のコストなどを考慮すると，その影響は致命的な問題とは考えられない．



		No access	Download	Upload	CPU
Time to activate suspended virtual server (s)	T_V	4.67	4.82	4.79	6.78

図 5.16　サーバ負荷による仮想サーバの起動時間への影響

5.5 まとめ

サーバ管理システムにおいて，一般的に採用されているサーバ･クライアント方式と P2P 方式の違いを示し，管理対象サーバの台数増加における管理の負荷において，P2P 方式に優位性があることを示した．また，P2P 方式には管理プログラムの分散による管理の複雑化や管理対象サーバの稼働状況の調査において問題点があることを示し，それらの対策を施した P2P 方式サーバ管理システムを提案した．

本提案システムを構築するために，管理プログラムの分散による管理の複雑化を防ぐために Executer システムを開発し，その仕様や機能を示した．また，故障サーバの機能を復旧するために予備の実サーバではなく仮想サーバを使用する方法を示し，仮想サーバの起動対象となる管理サーバの選択に関して，管理サーバ群の負荷状態および故障サーバのサーバタイプを考慮した．さらに，サーバ管理において管理の優先順位を導入し 2 台の管理サーバが 1 台の管理対象サーバを管理する管理フォームを採用するとともに，管理対象サーバが故障した場合，新たな管理対象サーバを管理するための動的管理拡張方式を提案し，その稼働状況を示した．本方式を採用した実験システムを構築し，その構成および使用機器の仕様を示すとともに，実サーバおよび仮想サーバの起動時間を含む特性も示した．また，サーバ機能を復旧するための仮想サーバは，その起動時間を短縮するために，常時，サスペンド状態で待機することとした．実験では，サーバトラブルとしてサービス提供プログラムとネットワークのトラブルを再現し，サービストラブルとしては，サービス提供プログラムの再起動により復旧する場合とその再起動では復旧せず，管理対象サーバを再起動することにより復旧する場合の 2 種類，ネットワークトラブルとしては，管理対象サーバの再起動およびシャットダウンの 2 種類を行った．実験において，それらのトラブルを管理対象サーバに対して再現し，サーバ機能の復旧時間，管理対象サーバの復旧時間とクライアントにおけるアクセス切断時間を実測した．1 台の管理対象サーバに対し，サービ

ス提供プログラムおよびネットワークに関するトラブルを発生させた場合において，いずれも 9 秒未満でサーバ機能の復旧が可能であり，同時に 2 台の管理対象サーバに対し，それらのトラブルを発生させた場合においても 12 秒未満でサーバ機能の復旧が可能であることを示した．本提案システムは，複数台の管理対象サーバに対してサービス提供プログラムおよびネットワークに関するトラブルが発生した場合においても，管理対象サーバの機能およびサーバ自体の復旧が可能であることを示した．また，従来方式との比較では，提案方式は稼働率において非常に高い優位性があり，コスト面においても高い優位性があることを示した．

第 6 章

考察

6.1　開発システムにおける想定規模

本論文において提案するシステムの想定規模を示すために，サーバシステムの規模をサーバ台数から 3 種類（小規模，中規模，大規模）に分類する．分類に関しては集積する素子の数によって IC を分類する定義を参考にする．一般的に企業内や大学内などで使用するサーバ数が 100 台以下で構成されるサーバシステムを小規模, 1,000 台以下のサーバシステムを中規模とし，Google や Microsoft など世界規模でサービスを提供する 1,000 台より多いサーバシステムを大規模とする．以下に本研究において開発した実験システムの仕様から管理対象サーバの適切な台数を算出し，それぞれの想定規模を示す.

各章における実験システムの仕様から管理対象サーバの望ましい設計台数を述べており，第 2 章のシステムでは 40 台，第 3 章のシステムでは 170 台，第 4 章のシステムでは 40 台とし，第 5 章のシステムでは 256 台を最大としている．

2, 4 章で示したシステムでは，対象として小規模サーバシステムとなる．しかし，表 6.1 に示すように 1 台の管理サーバが管理対象サーバ 40 台を管理するシステムを 1 つの管理グループとし，それを複数運用することで，中規模サーバシステムでの管理が可能となる．ただし，管理サーバの台数が増加するため，その管理方法の効率化が必要となる．

表 6.1　中規模サーバシステムへの適用

MGS: The number of management groups

Chapter 2, 4	The number of target servers					
	40	**80**	**120**	・・・	**40(*n*-1)**	**40*n***
MGS	1	2	3	・・・	*n*-1	*n*

3 章で示したシステムでは，対象として中規模サーバシステムとなる．しかし，表 6.2 に示すように 1 台の管理サーバが管理対象サーバ 170 台を管理するシステムを 1 つの管理グループとし，それを複数運用することで，対象規模を拡大することが可能となる．ただし，管理サーバの台数が増加するため，その管理方法の効率化が必要となる．

表 6.2　サーバシステム規模の拡大

MGS: The number of management groups

Chapter 3	The number of target servers					
	170	**340**	**510**	・・・	**170(*n*-1)**	**170*n***
MGS	1	2	3	・・・	*n*-1	*n*

5 章で示したシステムでは，対象として中規模サーバシステムとなる．しかし，図 6.1 に示すように，グループ化を採用し，それぞれにおいて P2P 方式サーバ管理システムを稼働すれば対象規模を拡大することが可能となる．

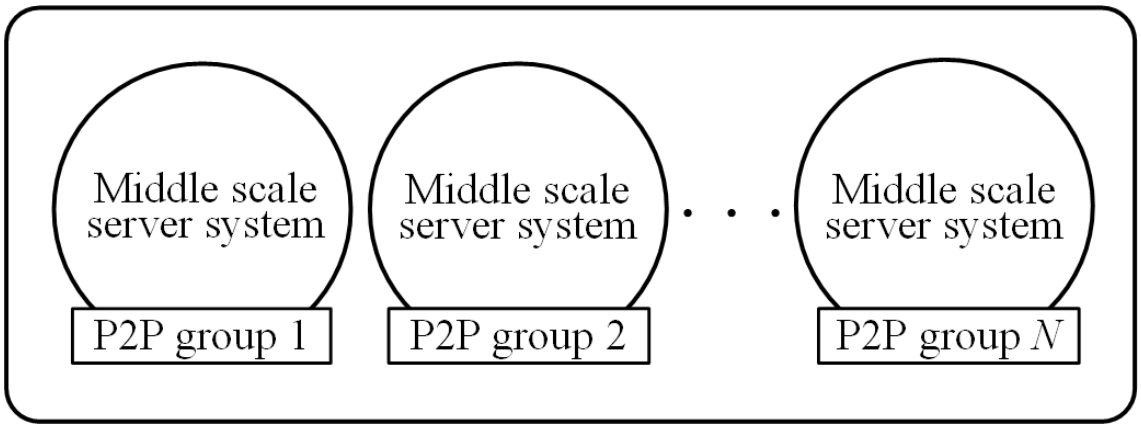


図 6.1　グループ化による管理

6.2　開発システムの総合評価

2 章から 5 章での開発システムは使用機器や OS などの仕様が異なるため，復旧時間での差では評価ができない．しかし，各章で示した稼働率は管理対象サーバの復旧時間における最大値を稼働率 0.5 としていることから，それらの結果は正規化されている．そこで，その稼働率において総合評価を行う．それぞれの開発システムの実験において，サービス提供プログラムおよびネットワークのトラブルを再現した場合の提案システムと従来システムにおける稼働率の平均値を表 6.3 にまとめる．2 章でのシステムを “MSBS”，3 章でのシステムを “Hybrid”，4 章でのシステムを “SIF” とし，5 章でのシステムを “P2P” とする．省電力かつ高可用性サーバシステムである Hybrid と SIF では Moderate condition を“MC” とし，Light condition を “LC” とする．

MSBS と P2P はいずれもほぼ同等の値となっていることが分かる．省電力かつ高可用性サーバシステムを構築する Hybrid と SIF では，提案システムの稼働率において SIF が約 0.03 高くなっている．これは，バックアップサーバの起動方式が Hybrid では表 3.4 に示すように NFS 経由であり，SIF では表 4.3 に示すように実サーバのローカルディスクからで

あるため，起動時間の差が影響したと考えられる．また，従来方式の稼働率においては Hybrid が 0.10 高くなっている．これは，仮想サーバの起動方式が Hybrid では実サーバのローカルディスクからであり，SIF では NFS 経由であるため，これも起動時間の差が影響したと考えられる．全体的には，いずれのシステムにおいても提案方式は従来方式に比べ，高い優位性を示している．

表 6.3　開発システムの平均稼働率

Chapter 2: MSBS
Chapter 3: Hybrid
Chapter 4: SIF
Chapter 5: P2P

Condition
MC: Moderate
LC: Light

			Conventional system	Proposed system	Improvement (%)
Availability rate	MSBS		0.62	0.97	56.45
	Hybrid	MC	0.72	0.94	30.56
		LC	0.72	0.95	31.94
	SIF	MC	0.62	0.98	58.06
		LC	0.62	0.98	58.06
	P2P		0.65	0.96	47.69

第 7 章

結論

本研究において高可用性サーバシステムを構築する上で必要となるサーバ管理システムを低コストの環境下で実現可能な構築方法を確立することができた．また，省電力かつ高可用性サーバシステムを構築する上で，省電力および高可用性を 1 つの制御プログラムで実現する方式に比べ，効率的な制御プログラムの実行方式も確立することができた．さらに，特定の管理サーバを必要とせず，管理対象サーバの台数に依存しないサーバ管理システムの構築方法も確立した．この研究成果により，低コスト，省電力，高可用性サーバシステムの構築技術が向上でき，より安定したインターネットサービスの提供が可能となる．

得られた成果を「複合サーババックアップシステム」,「異種システム混合処理方式」,「システムインタフェース方式」,「Peer-to-Peer 方式サーバ管理システム」の 4 つの項目に分けて列挙する．

7.1 複合サーババックアップシステム

低コストで高可用性サーバシステムの開発を推進するために複合サーババックアップシステム（MSBS）を開発した．MSBS は動的バックアップサーバシステム（DBSS）を基本としており，DBSS を様々なサーバシステムに適用可能とするためモード分割方式を提案し，導入した．モード分割方式では管理対象サーバの OS に応じた処理を行うモードのみの変更により動作が可能となる．DBSS は 1 台の管理対象サーバのみを管理することから，その管理プログラムは単純な形式であり，トラブルを発生した管理対象サーバのサーバ機能や管理対象サーバ自体の復旧が可能であることを示した．復旧処理において，時間を要

する管理対象サーバの処理にはバックグラウンドジョブを採用し，効率の良い管理方法を確立した．また，DBSS では管理対象である実サーバの故障時に，そのサーバと同様のサービスをクライアントに提供可能な仮想サーバによって管理対象サーバの機能を復旧する．そのため，予備用の実サーバが不要となり，低コストでのバックアップシステムを実現した．さらに，MSBS は 1 台の管理サーバ上で管理対象サーバ毎に対応する DBSS を複合的に実行することにより，複数台の管理対象サーバの管理が可能となるため，低コストへの効果は更に高くなる．

MSBS 実験システムを構築し，実験において，サービス提供プログラムおよびネットワークのトラブルに対し，DBSS が対処可能であることを示すとともにそれらのトラブル要因に対する復旧時間の実測値と予測値を示した．実測値と予測値はいずれの場合もほぼ同等の値となり，DBSS による管理対象サーバの機能およびそのサーバ自体の復旧時間を予測可能であることを示した．また，管理対象サーバが 2 台同時にトラブルを発生した場合，各 DBSS が独立して管理対象サーバを管理していることから，それぞれの復旧時間は 1 台の故障時に必要となる復旧時間とほぼ同等となった．従来方式との比較では，MSBS は稼働率において非常に高い優位性があり，コスト面においても高い優位性を有していることを示した．

DBSS で重要となる仮想サーバの性能調査を行うため，仮想サーバ性能調査システムを構築した．3 種類の実サーバから起動した仮想サーバに対してダウンロード，アップロードおよび CPU 処理に要する時間を実測した．MSBS を導入するサーバが管理対象のサーバと同程度の仕様であれば，サーバ機能の復旧後，応答時間に関して遅延等の問題がないことを示した．

7.2 異種システム混合処理方式

異種システム混合処理方式は複数の制御プログラムを交互に動作させることによって，それらの制御プログラムから得られる成果を実現する．そのため，それぞれのプログラムにおいて，誤動作を防ぐために共有ファイルを採用し，その役割を示した．異種システム混合処理方式の動作検証を行うため，独立して動作可能な省電力サーバシステム（PSS）と高可用性サーバシステムを開発した．

PSS ではクライアントに提供するサービス毎にサービスグループを設定し，各サービスグループでの仮想サーバは冗長化構成とした．PSS はサービスグループ毎の負荷に応じて Moderate，Light と Heavy の 3 種類の状態にサーバシステムの構成を変更可能とした．サービスグループの負荷としては一般的なサーバへのアクセスを考慮し，CPU 負荷とネットワーク負荷の 2 種類とし，それらから Load class を決定可能とした．サーバシステムの使用環境を考慮し，Load class に応じたサーバシステムの構成変更には連続回数による判断方式を採用した．

高可用性サーバシステムでは 2 章で示した MSBS の方式とは異なり，1 つの制御プログラムで複数の管理対象サーバを管理する方式を採用した．また，管理対象はクライアントにサービスを提供する仮想サーバおよびそれを起動する実サーバとした．通常状態において，それぞれのサーバトラブルに対して高可用性サーバシステムが対処可能であり，PSS によってサーバシステムを省電力状態とした場合にもトラブル対処が可能であることを示した．復旧処理において，時間を要する管理対象サーバの処理にはバックグラウンドジョブを採用し，効率の良い管理方法を確立した．

異種システム混合処理方式を採用した省電力かつ高可用性サーバシステムの実験システムを開発し，動作検証を行った．通常状態の実サーバ群が消費する電力は 290W となるが，省電力状態の電力は 112W となり，39%の電力での運用が可能となった．また，通常状態

と省電力状態におけるサーバ稼働時において，サービスを提供する仮想サーバおよびそれを起動する実サーバにトラブルを再現する実験を行い，それらの復旧時間を実測した．実サーバが故障したとしても，20 秒未満でサーバ機能を復旧可能であることを示した．従来方式との比較では，提案システムは稼働率において非常に高い優位性があり，コスト面においても高い優位性を有していることを示した．

7.3 システムインタフェース方式

省電力かつ高可用性サーバシステムの構築において，独立して動作可能である，MSBS と PSS の複合動作方式を提案した．MSBS と PSS には，ここでの環境に対応するための変更を加えた．MSBS を構成する DBSS は 1 台の管理対象サーバのみを管理するため，プログラム構成がシンプルであることが特徴である．ここでは，PSS における管理プログラムは無編集とし，DBSS が PSS の構成ファイルを編集可能にすることで複合動作を実現した．通常，2 種類の管理プログラムを複合動作させる場合，両方のプログラムにおいて誤動作を防止するためにそれらのプログラムは複雑となる．DBSS のプログラム構成が複雑になることを防ぐために，2 つのシステムを仲介するシステムインタフェース（SIF）を提案し, 導入した. SIF に PSS の構成ファイルへのアクセスと編集機能を追加することにより，DBSS のプログラムがシンプルな構成を維持することができた．SIF は DBSS からのシグナルを割り込み処理で受け取るため，高速な対処が可能となった．

提案方式を導入した省電力かつ高可用性サーバシステムの実験サーバシステムを構築し，動作検証を行った．通常状態の実サーバ群が消費する電力は 263W となるが，省電力状態の電力は 93W となり，35%の電力での運用が可能となった．稼働状況を記録する Log ファイルにおいて，DBSS と PSS が誤動作せずに，正常に複合動作していることを示した．また，通常状態または省電力状態におけるサーバ構成において，サービス提供プログラムお

よびネットワークに関するトラブルを再現する実験を行い，サーバ機能の復旧時間を実測した．本提案方式を導入した省電力かつ高可用性サーバシステムは，それらのトラブルが発生した場合でも動作を継続可能であり，仮想サーバや実サーバが故障したとしても，6 秒未満でサーバ機能を復旧可能であることを示した．従来方式との比較では，提案システムは稼働率において非常に高い優位性があり，コスト面においても高い優位性を有していることを示した．

7.4 Peer-to-Peer 方式サーバ管理システム

ネットワークシステムにおいて Peer-to-Peer（P2P）方式は，クライアント・サーバ方式に比べサーバの耐障害性に優れる．そこで P2P 方式の特徴を採用した P2P 方式サーバ管理システムを開発した．サーバ管理システムにおいて，一般的に採用されているサーバ・クライアント方式と提案した P2P 方式の違いを示した．P2P 方式の欠点と考えられる管理プログラムの分散による管理の複雑化を防ぐために Executer システムを導入し，Log ファイルなどの分散対策に関しては File サーバとの NFS 接続を使用することで対処した．P2P 方式ではクライアントにサービスを提供するサーバが管理対象サーバかつ管理サーバとなり，基本的に 1 台の管理サーバは 2 台の管理対象サーバを管理する形式を採用し，管理サーバの管理における負荷を低減した．また，サーバ管理において管理の優先順位を導入し，2 台の管理サーバが 1 台の管理対象サーバを管理する管理フォームを採用した．また，複数サーバの同時故障を考慮し，管理対象サーバの故障を検出した場合，直ちに新たな管理対象サーバの管理を実行する動的管理拡張方式を提案し，導入した．さらに，故障したサーバのタイプを考慮したサーバ機能の復旧方法も提案した．

本方式を採用した実験システムを構築し，その構成を示すとともにサービス提供プログラムやネットワークのトラブルを再現する実験を行い，管理対象サーバの機能および管理

対象サーバ自体の復旧に要する時間を実測し，その時間の予測も行った．いずれの場合も実測値と予測値ではほぼ同等の値となり，復旧時間の予測が可能であることを示した．複数台の管理対象サーバにおいてそれらのトラブルが同時に発生した場合においても，本方式は管理対象サーバの機能およびサーバ自体の復旧が可能であることを示した．従来方式との比較では，提案システムは稼働率において非常に高い優位性があり，コスト面においても高い優位性を有していることを示した．

7.5 今後の課題

今後，研究の発展する方向としては「トラブルの予測処理」,「種々の OS への自動対応」,「P2P 方式によるサーバ管理システムの共有ファイルシステム」や「特定の管理サーバを必要としない省電力サーバシステム」が挙げられる．

トラブルの予測処理について 通常，サーバ管理システムは管理対象サーバにトラブルが発生した場合，その状況に応じた対処を行う．このような死活監視では，管理対象サーバが冗長化されていない場合，サービス提供における切断時間が生じてしまう．そこで，管理対象サーバの稼働状態を記録する Log ファイルの分析やメモリ，ネットワーク，ディスクドライブおよび CPU の使用率を分析可能な性能監視を導入し，トラブル発生の予測を行い，管理対象サーバのトラブル発生前にサーバ機能の復旧準備を完了できるシステムの開発が必要である．また，性能異常を発生しているサーバ自体を復旧する方法の検討も必要となる．

種々の OS への自動対応について 本研究で提案し，採用したモード分割方式では管理対象サーバの OS に依存した部分のみのプログラム変更が必要となる．サーバ管理システムは

様々なサーバシステムに適用可能であることが望ましい．そこで，サーバ管理システムが管理対象サーバにおける OS の調査を自動的に行い, 管理のために必要なプログラムをその OS に対応可能な形式に自動変更して動作するシステムの開発が必要となる．これによって様々なサーバシステムへの導入が容易となり，サーバ管理者における負荷の軽減が期待できる．

P2P 方式によるサーバ管理システムの共有ファイルシステムについて　本研究において P2P 方式を採用したサーバ管理システムの構築方法が確立できた．しかし，各管理サーバは，特定のファイルサーバと NFS 接続を行い，各管理サーバにおける共有エリアを作成している．そのため，ファイルサーバの冗長化構成が必要となる．そこで，ファイルサーバを必要とせず，各管理サーバが共有ファイルを作成可能なファイル共有システムの構築が必要となる．

特定の管理サーバを必要としない省電力サーバシステムについて　3 章や 4 章で示した PSS のサーバ管理はサーバ・クライアント方式で行われている．そのため，その管理サーバにおける故障時の対策を考慮しなければならない．そこで，PSS に関しても 5 章で示した P2P 方式を導入する必要がある．それと同時に P2P 方式サーバ管理システムとの複合動作に関する検討も必要となる．

謝辞

本論文をまとめるに際し，御指導と数々の御配慮を賜りました東京工芸大学の宇田川佳久教授に深く感謝申し上げます．また，本論文の内容について，有益な御助言，御指導をいただきました東海大学の藤野巖教授，東京工芸大学の藤橋忠悟名誉教授，木下照弘名誉教授，曽根順治教授，片上大輔教授，辛徳准教授に厚く御礼申し上げます．

本研究は1993年に東京工芸大学　情報処理教育研究センター助手として勤務して以来取り組んできた研究をまとめたものであります．その間，東京工芸大学の藤橋忠悟名誉教授，西宮信夫教授には一貫して，本研究の遂行と本論文の完成に至るまで，御指導，御鞭撻を賜り，心より感謝申し上げます．

本研究の多くは東京工芸大学コンピュータ応用学科ネットワークシステム研究室の大学院生および卒業研究生の協力を得て行われたものであり，ネットワークシステム研究室の歴代卒業生および在学生各位に深く感謝致します．

最後に，今に至るまで筆者を暖かく見守り，激励をくださった東京工芸大学の青木彪名誉教授，コンピュータ応用学科の先生方，そして家族に心より感謝の意を表します．

参考文献

[1] P. P. C. Lee, V. Misra, D. Rubenstein, “Toward Optimal Network Fault Correction via End-to-End Inference,” Proc. IEEE INFOCOM 2007, vol. 3, pp. 1343-1351, 2007.

[2] Y. Bejerano, R. Rastogi, “Robust Monitoring of Link Delays and Faults in IP Networks,” IEEE/ACM Trans. Netw. vol. 14, no. 5, pp. 1092-1103, Oct. 2006.

[3] D. Lee, D. Chen, J. Wu, X. Yin, R. Hao, R. E. Miller, “Network Protocol System Monitoring-A Formal Approach with Passive Testing,” IEEE/ACM Trans. Netw., vol. 14, no. 2, pp. 424-437, Apr. 2006.

[4] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. N. Chuah, Y. Ganjali, C. Doit, “Characterization of failures in an Operational IP Backbone Network,” IEEE/ACMTrans. Netw., vol. 16, no. 4, pp. 749-762, Aug. 2008.

[5] C. Pinart, “A Multilayer Fault Localization Framework for IP over All-Optical Multilayer Networks,” IEEE Netw., vol. 23, no. 3, pp. 4-9, May 2009.

[6] J. H. Lee, J. H. Moon, C. H. Lee, K. M. Choi, “Seamless Maintenance and Protection Scheme for Next-Generation Access Networks,” IEEE Photonic Technol. Lett., vol. 21, no. 9/12, pp. 799-801, Jun. 2009.

[7] K. Nakayama, N. Shionmiya, H. Watanabe, “Distributed Control for Link Failure Based on Tie-Sets in Information Networks,” IEEE Int. Symp. Circuits Syst. 2010, vol. 6, pp. 3913-3916, 2010.

[8] C. P. Lai, D. Brunina, C. Ware, B. G. Bathula, BERGMAN Keren, LAI Caroline P., WARE Cedric, “Demonstration of Failure Reconfiguration via Cross-Layer Enabled

Optical Switching Fabrics," IEEE Photonic Technol. Lett., vol. 23, no. 21-24, pp. 1679-1681, Nov. 2011.

[9] H. Hamed, A. E. Atawy, E. A. Shaer, "On Dynamic Optimization of Packet Matching in High-Speed Firewalls," IEEE J Sel. Areas Commun., vol. 24, no. 10, pp. 1817-1830, Oct. 2006.

[10] T. Leighton, "Improving Performance on Internet," Commum. ACM, vol. 52, no. 2, pp. 44-51, Feb. 2009.

[11] J. D'ambrosia, "40 Gigabit Ethernet and 100 Gigabit Ethernet: The Development of A Flexible Architecture," IEEE Commun. Mag., vol. 47, no. 3, pp. S8-S14, March 2009.

[12] P. Drolet, L. Duplessis, "100G Ethernet and OTU4 Testing Challenges: Form the Lab to the Field," IEEE Commun. Mag., vol. 48, no. 7, pp. 78-82, Jul. 2010.

[13] Z. Chen, C. Lin, Y. Zhao, Q. Wang, "Accelerating Large-scale Data Distribution in Booming Internet: Effectiveness, Bottlenecks and Practices," IEEE Trans. Consum. Electron., vol. 55, no. 2, pp. 518-526, May 2009.

[14] N. Calabretta, W. Wang, T. Ditewig, O. Raz, F. G. Agis, S. Zhang, W. H. De, H. J. S. Dorren, "Scalable Optical Packet Switches for Multiple Data Formats and Data Rates Packets," IEEE Photonic Technol. Lett., vol. 22, no. 7, pp. 483-485, Apr. 2010.

[15] G. Hasegawa, and M. Murata, "Transport-layer Protocols for High-speed andLong-delay Networks," IEICE Technical Report, vol. 106, no. 524, pp. 41-46, Jan. 2007.

[16] H. Y. Kung, J. S. Hua, Y. F. Chang, C. Y. Lin, "Seamless QoS Adaptation Control for Embedded Multimedia Communications," IEEE Trans. Consum. Electron., vol. 52,

no. 1, pp. 240-248, Feb. 2006.

[17] S. K. Youm, M. Kim, C. H. Kang, “Performance Analysis of Reliable Multicast Protocols Using Transparent Proxy Servers on Wired and Wireless Networks,” IEICE Trans. Commun., vol. E89-B no. 4, pp. 1059-1069, Apr. 2006.

[18] S. Jiang, K. Davis, X. Zhang, “Coordinated Multilevel Buffer Cache Management with Consistent Access Locality Quantification,” IEEE Trans. Comput., vol. 56, no. 1, pp. 95-108, Jan. 2007.

[19] M. Bozinovski, H. P. Schwefel, R. Prasad, “Maximum Availability Server Selection Policy for Efficient and Reliable Session Control Systems,” IEEE/ACM Trans. Netw., vol. 15, no. 2, pp. 387-399, Apr. 2007.

[20] H. Lee, J. Suh, S. Jung, “An Efficient Cache Invalidation Method in Mobile Client/Server Environment,” IEICE Trans. Inf. Syst., vol. E90-D, no. 10, pp. 1672-1677, Oct. 2007.

[21] H. L. Chen, J. R. Marden, A. Wierman, “On the impact of heterogeneity and back-end scheduling in load balancing designs,” Proc. IEEE INFOCOM 2009, vol. 4, pp. 2267-2275, 2009.

[22] W. Wang, B. Li, B. Liang, “Dominant Resource Fairness in Cloud Computing Systems with Heterogeneous Servers,” Proc. IEEE INFOCOM 2014, vol. 1, pp. 583-591, 2014.

[23] W. Liu, R. Shinkuma, T. Takahashi, “A Local Resource Sharing Platform in Mobile Cloud Computing,” IEICE Trans. Commun. (Web), vol. E97-B, no. 9, pp. 1865-1874, Sep. 2014.

[24] F. Zhang, T. D. Todd, D. Zhao, V. Kezys, “Power Saving Access Points for IEEE

802.11 Wireless Network Infrastructure," IEEE Trans. Mob. Comput., vol. 5, no. 2, pp. 144-156, Feb. 2006.

[25] M. Gupta, S. Singh, "Using Low-power Modes for Energy Conservation in Ethernet LANs," Proc. IEEE INFOCOM 2007, vol. 5, pp. 2451-2455, 2007.

[26] C. Gunaratne, K. Christensen, S. Suen, B. Nordman, "Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR)," IEEE Trans. Comput., vol. 57, no. 4, pp. 448-461, Apr. 2008.

[27] Y. Park, G. U. Hwang, "An Efficient Power Saving Mechanism for Delay-Guaranteed Services in IEEE 802.16e," IEICE Trans. Commum., vol. E92-B, no. 1, pp. 277-287, Jan. 2009.

[28] C. Cicconetti, L. Lenzini, E. Mingozzi, C. Vallati, "Reducing Power Consumption with QoS Constraints in IEEE 802.16e Wireless Networks," IEEE Trans. Mob. Comput., vol. 9, no. 7, pp. 1008-1021, Jul. 2010.

[29] W. Shengquan, L. Jun, C. Jian-Jia, L. Xue, "Power Sleep: A smart Power-Saving Scheme with Sleep for Servers under Response Time Constraint," IEEE J. Emerg. Sel. Top. Circuit. Syst., vol. 1, no. 3, pp. 289-298, Sep. 2011.

[30] Y. Tarutani, Y. Ohsita, M. Murata, "A Virtual Network to Achieve Low Energy Consumption in Large-scale Datacenter," IEICE Technical Report, vol. 111, no. 475, pp. 91-96, March 2012.

[31] Y. Ohsita, M. Murata, "Network Topologies using Optical Packet Switches for Data Centers," IEICE Technical Report, vol. 111, no. 475, pp. 79-84, March 2012.

[32] A. Tsurusaki, Y. Arakawa, D. Ishi, N. Yamanaka, H. Ishikawa, and Y. Shiba, "Network Reconfigure Algorithm with Least Resource For Energy Efficient Optical

Network,” IEICE Technical Report, vol. 108, no. 360, pp. 1-5, Dec. 2008.

[33] H. Yonezu, S. Gao, S. Shimizu, D. Ishii, S. Okamoto, E. Oki, and N. Yamanaka, “High-Speed Energy Efficient Topology Calculation Method by Link Power Management,” IEICE Technical Report, vol. 110, no. 20, pp. 17-22, Apr. 2010.

[34] H. Yonezu, D. Ishii, S. Okamoto, E. Oki, and N. Yamanaka., “Energy Optimal Topology Calculation Method for Self Organized Network MiDORi,” IEICE Trans., vol. J94-B, no. 10, pp. 1323-1331, Oct. 2011.

[35] H. Inkwon and P. Massoud, "A Comparative Study of the Effectiveness of CPU Consolidation Versus Dynamic Voltage and Frequency Scaling in a Virtualized Multicore Server," IEEE Trans. Very Large Scale Integ. Syst., vol. 24 no. 6, pp.2103-2116, Jun. 2016.

[36] S. Madhukar, K. Vilas, S. Shirish, and J. Rushikesh, "Autonomic and energy-aware resource allocation for efficient management of cloud data centre," IEEE Conference Proc., vol. 2017 no. i-PACT, pp. 1-8, 2017.

[37] アジア太平洋地域および日本における災害復旧 (DR) に関する調査結果, 米 EMC 社, https://japan.emc.com/about/news/press/japan/2012/20120615-1.htm, Jun. 2017.

[38]「上流」強化で システムトラブルを防ぐ, 日経 BP 社, http://www.ipa.go.jp/files/000005222.pdf, Aug. 2017.

[39] N. Kimura, A. Yamada, H. Seshake, and T. Nishizono, “High Availability Server Platform for IP Communication Services,” IEICE Trans., vol. J88-B, no. 1, pp. 224-233, Jan. 2005.

[40] M. Kitamura, “Configuring a Low-cost, Power-saving Multiple Server Backup System: Experimental Results,” IEICE Trans. Commun., vol. E95-B, no. 1, pp.

189-197, Jan. 2012.

[41] E. Iwasa, M. Irir, M. Kaneko, T. Fukumoto, and K. Ueda, “Methods of Dynamic Scaling for High Availability Server Clusters,” IEICE Trans., vol. J97-B, no. 1, pp. 19-30, Jan. 2014.

[42] L. Zhao, and K. Sakurai, “Improving the Cloud Revenue using a Failure-aware Server Management Model,” IEICE Technical Report, vol. 111, no. 113, pp. 25-32, Jun. 2011.

[43] W. L. Yeow, C. Westphal, and U. C. Kozat, “Highly Available Virtual Machines with Network Coding,” Proc. IEEE INFOCOM 2011, vol. 1, pp. 386-390, 2011.

[44] N. Maeda, and H. Miwa, “Method for Finding Links Protected for Keeping the Shortest Distance from Clients to Servers during Failures,” IEICE Technical Report, vol. 111, no. 468, pp. 227-232, March 2012.

[45] Y. Tarutani, Y. Ohsita, and M. Murata, “Virtual Network Reconfiguration with Fault-tolerance and Low Energy Consumption for Multi-Tenant Data Centers,” IEICE Technical Report, vol. 113, no. 473, pp. 293-298, Feb. 2014.

[46] S. Urushidani, J. Matsukata, S. Abe, Y. Ji, K. Fukuda, M. Koibuchi, M. Nakamura, and S. Yamada,"Detailed Network Design of SINET3 for Advanced Network Services,” IEICE Trans., vol. J91-B, no. 10, pp. 1136-1146, Oct. 2008.

[47] Y. Gotoh, T. Yoshihisa, M. Kanazawa, and Y. Takahashi, "Design and Implementation for Division Based Broadcasting Systems for Webcast,” IEICE Trans., vol. J92-B, no. 1, pp. 353-362, Jan. 2009.

[48] M. Kitamura and C. Fujihashi, "Analysis of Response Time Characteristics of a Real Computer by Heterogeneous-Task Bursty-Arrival Models," IEICE Trans., vol. J81-B-I, no. 4, pp. 243-253, Apr. 1998.

[49] K. Utsu, N. Fukushi, H. Kawabata, K. Endo, N. Hirata, and H. Ishii, "Optimal Server Selection from the Users' Side on the Internet," IEICE Technical Report, IN2007-163(2008-3), pp.25-30, March 2008.

[50] K. Takahashi and M. Oonuki, "A Practical Large-Scale Distributed Operations Support System," IEICE Trans., vol. J92-B, no. 7, pp. 970-980, Jul. 2009.

[51] K. Takahashi and H. Tamura, "A Practical Large-Scale Distributed Traffic Data Management System," IEICE Trans., vol. J93-B, no. 7, pp. 902-915, Jul. 2010.

[52] Ministry of Economy, Trade and Industry, http://www.meti.go.jp/committee/materials/downloadfiles/g80520c03j.pdf, Aug. 2013.

[53] M. S. Islam and Logeeswaran VJ, "Nanoscale Materials and Devices for Future Communication Networks," IEEE Commun. Mag., vol. 48, no. 6, pp.112-120, Jun. 2010.

[54] R. Kubo, J. Kani, Y. Fujimoto, N. Yoshimoto, and K. Kumozaki, "Adaptive Power Saving Mechanism for 10 Gigabit Class PON Systems," IEICE Trans. Commun., vol. E93-B, no. 2, pp. 280-288, Feb. 2010.

[55] H. Tamura, Y. Yahiro, Y. Fukuda, K. Kawahara, and Y. Oie, "Performance Analysis of Energy Saving Scheme with Extra Active Period for LAN Switches," Proc. IEEE GLOBECOM 2007, pp. 198-203, 2007.

[56] M. Minami, R. Kawamura, H. Morikawa, and M. Hirabaru, "Power-Aware New Generation Network," IEICE Trans., vol. J92-B, no. 4, pp. 605-614, Apr. 2009.

[57] Y. Ogawa, G. Hasegawa, M. Murata, and S. Nishimura, "Performance Evaluation of Distributed Computing Environment Considering Power Consumption," IEICE Technical Report, IN2009-172, pp. 169-174, March 2010.

[58] D. Arai, K. Yoshihara, and A. Idoue, "Proposal on ECO-friendly Data Center Operation Scheme," IEICE Technical Report, IN2008-211, pp. 469-474, March 2009.

[59] M. Okuno, D. Ito, H. Miyamoto, H. Aoki, Y. Tsushima, and T. Yazaki, "A Study on Distributed Information and Communication Processing Architecture for Next Generation Cloud System," IEICE Technical Report, NS2009-204, pp. 241-246, March 2010.

[60] Y. Nagami, T. Okuda, Y. Kadobayashi, and S. Yamaguchi "A Method for Improving the Confidence of Performance Measurement under Virtual Machine Environment," IEICE Technical Report, NS2008-184, pp. 233-238, March 2009.

[61] C. Lalit and K. Hemangee K., "Formal Approach for DVS-Based Power Management for Multiple Server System in Presence of Server Failure and Repair," IEEE Trans. Ind. Inform., vol. 9, no. 1, pp. 502-513, Feb. 2013.

[62] J. Heeseung, K. Hwanju, J. Jae-Wan, L. Joonwon, and M. Seungryoul, "Transparent Fault Tolerance of Device Drivers for Virtual Machines," IEEE Trans. Comput., vol. 59, no. 11, pp. 1466-1479, Nov. 2010.

[63] K. Ogura and H. Miwa, "Method of Finding Protected Data Centers to Keep Loads of Servers and Links Low during Server Failures," IEICE Technical Report, NS2012-271, pp. 617-622, March 2013.

[64] B. Dario, D. Salvatore, L. Francesco, P. Antonio, and S. Marco, “Workload-Based Software Rejuvenation in Cloud Systems,” IEEE Trans. Comput., vol. 62, no. 6, pp. 1072-1085, Jun. 2013.

[65] Y. Souda, T. Mikoshi, and T. Takenaka, “High-Speed Topology Constructive Method for Network Power Saving,” IEICE Trans., vol. J96-B, no. 2, pp. 92-101, Feb. 2013.

[66] Y.-C. Chen and C.-W. Wen, “Distributed Clustering with Directional Antennas for Wireless Sensor Networks,” IEEE Sens. Jour., vol. 13, no. 6, pp. 2166-2180, Jun. 2013.

[67] S. O. D. Luiz, A. Perkusich, C. B. M. J. Cruz, B. H. M. Neves, and G. M. S. Araujo, “Optimization of Timeout-based Power Management Policies for Network Interfaces,” IEEE Trans. Consum. Electron., vol. 59, no. 1, pp. 101-106, Feb. 2013.

[68] V. Luca, V. Dung P., R. Pier G., C. Piero, C. Divanilson R., W. Shing-Wa, Y. She-Hwa, K. Leonid G., and Y. Shinji, “Energy Efficiency in Passive Optical Networks: Where, When, and How ?,” IEEE Netw., vol. 26, no. 6, pp. 61-68, Nov. 2012.

[69] M. Xu, X. Ji, J. Wu, and M. Zhang, “A Low-Power LDPC Decoder for Multimedia Wireless Sensor Networks,” IEICE Trans. Commun., vol. E96-B, no. 4, pp. 939-947, Apr. 2013.

[70] H. Ito, H. Hasegawa, and K. Sato, “Router Power Reduction through Dynamic Performance Control Based on Traffic Predictions,” IEICE Trans. Commun., vol. E95-B, no. 10, pp. 3130-3138, Oct. 2012.

[71] S. Kurumatani and M. Toyama, “A Report of Performance per Power Consumption of ARM Based Servers,” IEICE Technical Report, IN2012-154, pp. 1-6, March 2013.

[72] W. Arahata and M. Bandai, “Fundamental Study on Green Data Center with Server Virtualizations,” IEICE Technical Report, NS2012-17, pp. 7-10, May 2012.

[73] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” IEEE Commun. Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, Jan. 2015.

[74] Ministry of Internal Affairs and Communications, http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/html/nc121100.html, Aug. 2017.

[75] Z. Lin and L. Dong, "Clarifying Trust in Social Internet of Things,” IEEE Trans. Knowledge and Data Eng., vol. 30, no. 2, pp. 234-248, Feb. 2018.

[76] N. A. Davis, A. Rezgui, H. Soliman, S. Manzanares, and M. Coates, "FailureSim: A System for Predicting Hardware Failures in Cloud Data Centers Using Neural Networks,” IEEE Conference Proc., vol. 2017, no. Cloud, pp. 544-551, 2017.

[77] D. A. Popescu and A. W. Moore, "PTPmesh: Data Center Network Latency Measurements Using PTP,” IEEE Conference Proc., vol. 2017, no. MASCOTS, pp. 73-79, 2017.

[78] M.-G. Rabbani, M.-F. Zhani, and R. Boutaba, "On Achieving High Survivability in Virtualized Data Centers,” IEICE Trans. Commun., vol. E97-B, no. 1, pp. 10-18, Jan. 2014.

[79] G. Wang, L. Zhang, and W. Xu, "What Can We Learn from Four Years of Data Center Hardware Failures ?," IEEE Conference Proc., vol. 2017, no. DAN, pp. 25-36, 2017.

[80] Q. Zhang, M.-F. Zhani, M. Jabri, and R. Boutaba, "Venice: Reliable Virtual Data Center Embedding in Clouds," Proc. IEEE INFOCOM 2014, vol. 1, pp. 289-297, 2014.

[81] H. Otsuka, K. Joshi, M. Hiltunen, S. Daniels, and Y. Matsumoto, "Online Failure Prediction with Accurate Failure Localization in Cloud Infrastructures," IEICE Technical Report, vol. 113, no. 496, pp. 7-12, March 2014.

[82] M. Hirono, T. Sato, J. Matsumoto, S. Okamoto, and N. Yamanaka, "HOLST: Architecture Design of Energy-efficient Data Center Network based on Ultra High-speed Optical Switch," IEEE Conference Proc., vol. 2017, no. LANMAN, pp. 1-6, 2017.

[83] J. K. Perin, A. Shastri, and J. M. Kahn, "Design of Low-Power DSP-Free Coherent Receivers for Data Center Links," Journal of Lightwave Tec., vol. 35, no. 21, pp. 4650-4662, Sep. 2017.

[84] T. Pan, T. Zhang, J. Shi, Y.Li, L. Jin, F. Li, J. Yang, B. Zhang, and B. Liu, "Towards Zero-Time Wakeup of Line Cards in Power-Aware Routers," Proc. IEEE INFOCOM 2014, vol. 1, pp. 190-198, 2014,

[85] Y. Fukushima, T. Murase, and T. Yokohira, "Energy-Aware Optimal Server Location Decision in Server Migration Service," IEICE Technical Report, vol. 116, no. 428, pp. 53-58, Jan. 2017.

[86] M. Kitamura, "Configuration of a Power-saving High-availability Server System Incorporating a Hybrid Operation Method," JISSJ, vol. 10, no. 2, pp. 1-17, March 2015.

[87] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. C.-Pastor, and A. Monje, "On the Optimal Allocation of Virtual Resources in Cloud Computing Networks," IEEE Trans. Comput., vol. 62, no. 4/6, pp. 1060-1071, Apr. 2013.

[88] Y. Zhu, W. Wu, J. Willson, L. Ding, L. Wu, D. Li, and W. Lee, "An Approximation Algorithm for Client Assignment in Client/Server Systems," Proc. IEEE INFOCOM 2014, vol. 3, pp. 2777-2785, 2014.

[89] Y. Yu and C. Qian, "FTDC: A fault-tolerant server-centric data center network," Proc. IEEE Conf., vol. 2016, no. IWQos, pp. 1-2, 2016.

[90] Z. Li, Y. Yang, "RRect: A Novel Server-centric Data Center Network with High Availability," Proc. IEEE Conf., vol. 2016, no. ICPP, pp. 41-46, 2016.

[91] K. Ali Bashir, Y. Ohsita, M. Murata, "A Distributed Virtual Data Center Network Architecture for the Future Internet," IEICE Technical Report, vol. 114, no. 478, pp. 261-266, March 2015.

[92] D. Irie, H. Miwa, "Network Design Method by Finding Server Placement and Protected Links to Keep Connectivity to Servers Against Link Failures," Proc. IEEE Conf., vol. 2016, no. INCos, pp. 439-444, 2016.

[93] J. Xiao, B. Wu, L. Zhang, H. Wen, X. Jiang, and P.-H. Ho, "Joint Design on DCN Placement and Survivable Cloud Service Provision over All-Optical Mesh Networks," IEEE Trans. Commun., vol. 62, no. 1, pp. 235-245, Jan. 2014.

[94] T. Yagata, Oracle, “Oracle consult speaks! Super fast! Various fault detection architecture supporting Oracle Coherence high availability,” http://www.oracle.com/technetwork/jp/ondemand/middleware/application-grid/e-13-coherence-1484719-ja.pdf, Aug. 2017.

[95] T. Ono and K. Ueda, "Failure Detection and Monitoring Method for High Availability Distributed Cluster,” IEICE Technical Report, vol. 116, no. 484, pp. 137-142, March 2017.

[96] H. Wang and H. Nakazato, “Failure Detection in P2P-Grid System,” IEICE Trans. Inf. & Syst., vol. E98-D, no. 12, pp. 2123-2131, Dec. 2015.

[97] H. Yamamoto, Cybozu, “Automatic Disaster Recovery System Talk of Tsukuyomi”, http://blog.cybozu.io/entry/5799, Aug. 2017.

[98] M. Kitamura, Y. Udagawa, H. Nakagome, and Y. Shimizu, “Development of a Low-cost Server Management System Incorporating a Peer-to-Peer Method for Constructing a High-availability Server System,” Proc. Comp. Scie. and Tech. 2017, pp. 817-827, May 2017.

発表論文・口頭発表等

発表論文

1) Configuring a Low-Cost, Power-Saving Multiple Server Backup System:

Experimental Results.

M.Kitamura,

IEICE TRANSACTIONS on Communications, vol. E95-B, no. 1, pp. 189-197,

Jan. 2012.

2) Configuration of a Power-saving High-availability Server System Incorporating a

Hybrid Operation Method.

M. Kitamura,

情報システム学会誌（JISSJ: Journal of Information Systems Society of Japan），

vol. 10, no. 2, pp. 1-17, March 2015.

3) Development of a Server Management System Incorporating

a Peer-to-Peer Method for Constructing a High-availability Server System.

M. Kitamura, Y. Udagawa, H. Nakagome, and Y. Shimizu,

情報システム学会誌（JISSJ: Journal of Information Systems Society of Japan），

vol. 13, no. 2, pp. 14-40, March 2018.

4) Development of a Power-saving, High-availability Server System by Compound Operation of a Multiple-server Backup System and Power-saving Server System.
M. Kitamura, Y. Shimizu, and K. Tani,
情報システム学会誌（JISSJ: Journal of Information Systems Society of Japan）に改訂版投稿済み.

5) Simplified Digital Still Image Compression Using Histograms of Wavelet Expansion Coefficients and Subjective Fidelity Estimation between Similar Reconstructed Images.
M.Iizuka, M. Kitamura, and Y. Nakashima,
J. Light & Vis. Env., vol. 20, no. 2, pp. 48-59, Dec. 1996. （参考）

6) 実計算機応答時間特性の異種タスクバーストモデルによる分析.
北村光芳，藤橋忠悟,
電子情報通信学会論文誌 B- I vol. J81-B-I, no. 4, pp. 243-253, Apr. 1998. （参考）

7) Deterioration and Quantitative Estimation of Reconstructed Digtal Images Caused by Removal and Modification of Part of DCT & DWT Coefficients in Orthogonal Transform Techniques.
M.Iizuka, M. Kitamura, and Y. Nakashima,
Journal of Signal Processing. vol. 3, no. 1, pp. 69-76, Jan. 1999. （参考）

国際会議

1) Development of a Low-cost Server Management System Incorporating

a Peer-to-Peer Method for Constructing a High-availability Server System.

M. Kitamura, Y. Udagawa, H. Nakagome, and Y. Shimizu,

The International Conference on Computer Science and Technology (CST2017),

Proc., pp. 817-827, May 2017.

国内発表

1) サーバ・クライアントシステムにおける利用形態の実測について.
Measurement of Utilization Behavior in Server-Client System.

行谷時男，北村光芳,

1999 年　電子情報通信学会春季大会，講演論文集 B-7-31.

2) サーバ・クライアントシステムにおけるサービス形態の実測について.
Measurement of Service Behavior in Server-Client System.

北村光芳，行谷時男,

2000 年　電子情報通信学会春季大会，講演論文集 B-7-1.

3) HTTP トラヒックの実測について.
Measurement of HTTP Traffic.

北村光芳，行谷時男,

2000 年　電子情報通信学会ソサイエティ大会，講演論文集 B-7-26.

4) HTTP トラヒックにおける実測データの分析について.
Analysis of Measurement Data of HTTP Traffic.

北村光芳,

2001 年　電子情報通信学会総合大会，講演論文集 B-7-212.

5) Proxy サーバの並列化の評価について.
Estimation of the Parallel of a Proxy Server.

北村光芳,

2003 年　電子情報通信学会総合大会，講演論文集 B-7-40.

6) HTTP データの応答時間特性について.
The Response Time Characteristic of HTTP Data.

北村光芳,

2005 年　電子情報通信学会総合大会，講演論文集 B-7-30.

7) クライアントーサーバシステムの稼働状況の実測について.
Measurement of Operation Performance of the Client-Server System.

北村光芳,

2006 年　電子情報通信学会総合大会，講演論文集 B-7-130.

8) 仮想 PC を用いた冗長化サーバシステムの構築について.
Construction of Redundant Server System used Virtual PC.

北村光芳,

2008 年　電子情報通信学会総合大会，講演論文集 B-7-61.

9) 冗長化サーバシステムにおける仮想 PC の適応について.
Virtual PC Adaptation in Redundant Server System.

北村光芳，武藤数典,

2009 年　電子情報通信学会総合大会，講演論文集 B-7-115.

10) 仮想サーバレイヤシステムにおける仮想サーバの性能解析について.
Performance Analysis of Virtual Server in Virtual Server Layer System.

北村光芳，秋元秀俊，油井啓伸,

2010 年　電子情報通信学会総合大会，講演論文集 B-7-77.

11) 仮想サーバ導入における基礎調査について.
Fundamental Examination in Virtual Server Introduction.

北村光芳，加藤玄大，今井勇太，阿部瑞樹，

2010 年　電子情報通信学会総合大会，講演論文集 B-7-78.

12) 負荷状況に応じた省電力サーバシステムの開発について.
Development of Power-saving Server System Corresponding to Load Condition.

北村光芳，鎌田伸太朗，平賀春樹，藤橋忠悟，

2011 年　電子情報通信学会総合大会，講演論文集 B-7-22.

13) 仮想サーバ自動割当システムの開発について.
Development of Automatic Assignment System of Virtual Server.

北村光芳，松岡哲也，寺坂卓也，藤橋忠悟，

2011 年　電子情報通信学会総合大会，講演論文集 B-7-73.

14) 仮想サーバ複数起動における性能調査について.
Performance Examination in Multiple Activation of Virtual Server.

北村光芳，大石一輝，藤橋忠悟，

2011 年　電子情報通信学会総合大会，講演論文集 B-7-74.

15) 仮想サーバ自動割当システムの開発について.
Development of Automatic Assignment System of Virtual Server.

北村光芳，藤橋忠悟，

2012 年　電子情報通信学会総合大会，講演論文集 B-6-81.

16) 高負荷時にも対応可能な省電力サーバシステムの開発について.
Development of Power-Saving Server System Dealing with Heavy Load.

北村光芳，博田清文，藤橋忠悟，

2012 年　電子情報通信学会ソサイエティ大会，講演論文集 B-6-15.

17) 動的バックアップサーバシステムの高速化について.
High-Speed Improvement of Dynamic Backup Server System.

北村光芳，海住美里，藤橋忠悟,

2012 年　電子情報通信学会ソサイエティ大会，講演論文集 B-6-73.

18) 異種システム混合処理方式を採用した省電力かつ高可用性サーバシステムの構築について.
Configuration of Power-Saving High Availability Server System
Including Hybrid Operation Method.

北村光芳，シュウシン,

2014 年　電子情報通信学会総合大会，講演論文集 B-6-106.

19) Peer to Peer 方式を採用したサーバ管理システムの開発について.
Development of Server Management System Including Peer to Peer Method.

北村光芳，松浦晃享,

2014 年　電子情報通信学会総合大会，講演論文集 B-6-107.

20) 仮想サーバシステムにおける Peer to Peer 方式サーバ管理システムの適用について.
Application of Server Management System Incorporating Peer to Peer Method
in Virtual Server System.

シュウシン，北村光芳,

2015 年　電子情報通信学会総合大会，講演論文集 B-6-64.

21) Peer to Peer 方式を採用したサーバ管理システムの開発について.
Development of Server Management System Including Peer to Peer Method.

北村光芳，小山達也，シュウシン,

2015 年　電子情報通信学会総合大会，講演論文集 B-6-130.

22) 仮想サーバシステムにおける Peer to Peer 方式サーバ管理システムの開発について.
Development of Server Management System Incorporating Peer to Peer Method in Virtual Server System.

シュウシン，北村光芳,

2015 年　電子情報通信学会ソサイエティ大会，講演論文集 B-6-26.

23) 省電力サーバシステムにおける Peer-to-Peer 方式サーバ管理システムの適用について.
Application of Server Management System Incorporating Peer-to-Peer Method in Power-saving Server System.

北村光芳，中込仁志，シュウシン,

2016 年　電子情報通信学会総合大会，講演論文集 B-6-6.

24) Peer-to-Peer 方式サーバ管理システムにおける仮想サーバグループの管理について.
Management of Virtual Server Group in Server Management System Incorporating Peer-to-Peer Method.

シュウシン，北村光芳，中込仁志,

2016 年　電子情報通信学会総合大会，講演論文集 B-6-61.

25) サーバ管理者育成システムの開発について.
Development of Server Administrator Training System.

北村光芳，齋藤将任，中込仁志，清水陽太,

2017 年　電子情報通信学会総合大会，講演論文集 B-6-36.

26) 仮想サーバシステムにおける Peer-to-Peer 方式を採用したサーバ管理システムの開発について.
Development of Server Management System Incorporating Peer-to-Peer Method in Virtual Server System.

中込仁志，北村光芳，清水陽太,

2017 年　電子情報通信学会総合大会，講演論文集 B-6-37.

27) Peer-to-Peer 方式を採用したサーバ管理システムの開発について.
Development of Server Management System Incorporating Peer to Peer Method.

北村光芳, 清水陽太, 中込仁志,

2017 年　電子情報通信学会総合大会，講演論文集 B-6-38.

28) Peer-to-Peer 方式サーバ管理システムにおけるサーバ故障の対処.
Dealing with Server Failure in Peer-to-Peer Method Server Management System.

清水陽太，北村光芳，中込仁志,

2017 年　電子情報通信学会ソサイエティ大会，講演論文集 B-6-4.

29) 仮想サーバシステムにおける Peer-to-Peer 方式サーバ管理システムの
サーバ機能復旧方式について.
Server Function Recovery Method of Peer-to-Peer Method Server Management
System in Virtual Server System.

中込仁志，北村光芳，清水陽太,

2017 年　電子情報通信学会ソサイエティ大会，講演論文集 B-6-20.

30) 省電力サーバシステムと複合サーババックアップシステムの複合動作について.
Compound Operation of Power-saving Server System
and Multiple Server Backup System.

清水陽太，北村光芳，中込仁志，谷昂樹,

2018 年　電子情報通信学会総合大会，講演論文集 B-6-12.

31) 仮想サーバシステムにおける Peer-to-Peer 方式サーバ管理システムの開発について.
Development of Peer-to-Peer Method Server Management System
in Virtual Server System.

中込仁志，北村光芳，清水陽太，谷昂樹,

2018 年　電子情報通信学会総合大会，講演論文集 B-6-13.

32) サーバ管理者育成システムの開発について.
Development of Server Administrator Training System.

北村光芳，谷昂樹，中込仁志，清水陽太,

2018 年　電子情報通信学会総合大会，講演論文集 B-6-14.

33) 仮想サーバシステムにおける Peer-to-Peer 方式サーバ管理システムの開発.
Development of Peer-to-Peer Method Server Management System
in Virtual Server System.

北村光芳，中込仁志，清水陽太，谷昂樹，宇田川佳久,

2018 年 5 月 19 日　ソフトウェアインタプライズモデリング研究会.

付録

F1. サーバの故障

（参考：http://avance.avail-tech.co.jp/avance/avance_01.html, Aug. 2017.）

米カーネギーメロン大学が行った統計では，ハードディスクの年間故障率は 3%程度，サーバを構成する部品に対するハードディスクの故障割合は約 30%程度と試算されている．この統計をもとにすると，以下の式により，サーバ自身の年間故障率は約 10%と考えられる．

30%：3% = 100%：x

x = 10%

上記の条件で 5 台のサーバを保有する場合，1 年間の間にいずれかのサーバのハードウエア障害に遭遇する確率は以下となる．

1 -（5 台がハードウエア障害に遭遇しない確率 ）

= 1 -（90% × 90% × 90% × 90% × 90%）= 1 - 約 60% = 約 40%

F2. サーバの故障台数（参考：http://q.hatena.ne.jp/1259636067, Aug. 2017.）

例えば，Owltech Seasonic 電源 S12 ENERGY+ シリーズ SS-650HT SS-550HT を例とすると，その MTBF（平均故障間隔）は 100,000 時間（25℃）である．10 台のサーバを 24 時間 365 日連続稼働させると，1 年では延べ 87,600 時間となり，上記の MTBF では 1 年に 1 回は故障する確率があることとなる．また，100 台のサーバであれば 1 年では延べ 876,000 時間となり，年に約 9 台のサーバが故障する確率となる．

F3. サーバ故障とその平均故障台数

サーバ故障に関して，サーバ台数を N 台，サーバの故障率を λ，サーバの復旧率を μ とし，待ち行列理論によるモデルを図 F3.1 に示す.

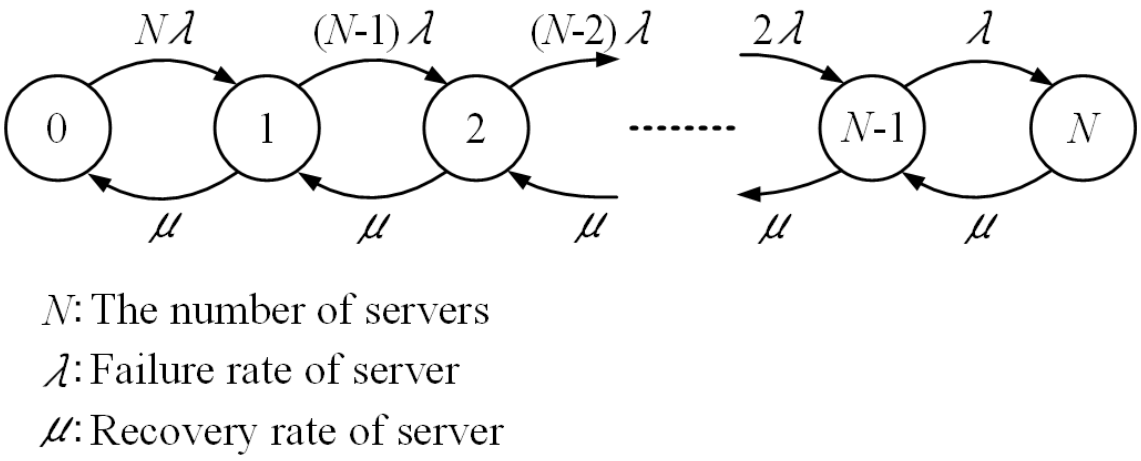


図 F3.1　サーバ故障および復旧モデル

状態確率 $p(n)$は以下の式となる.

$$p(n) = \frac{N!}{(N-n)!}\left(\frac{\lambda}{\mu}\right)^n p(0) \tag{f1}$$

$$p(0) = \frac{1}{\sum_{n=0}^{N} \frac{N!}{(N-n)!}\left(\frac{\lambda}{\mu}\right)^n} \tag{f2}$$

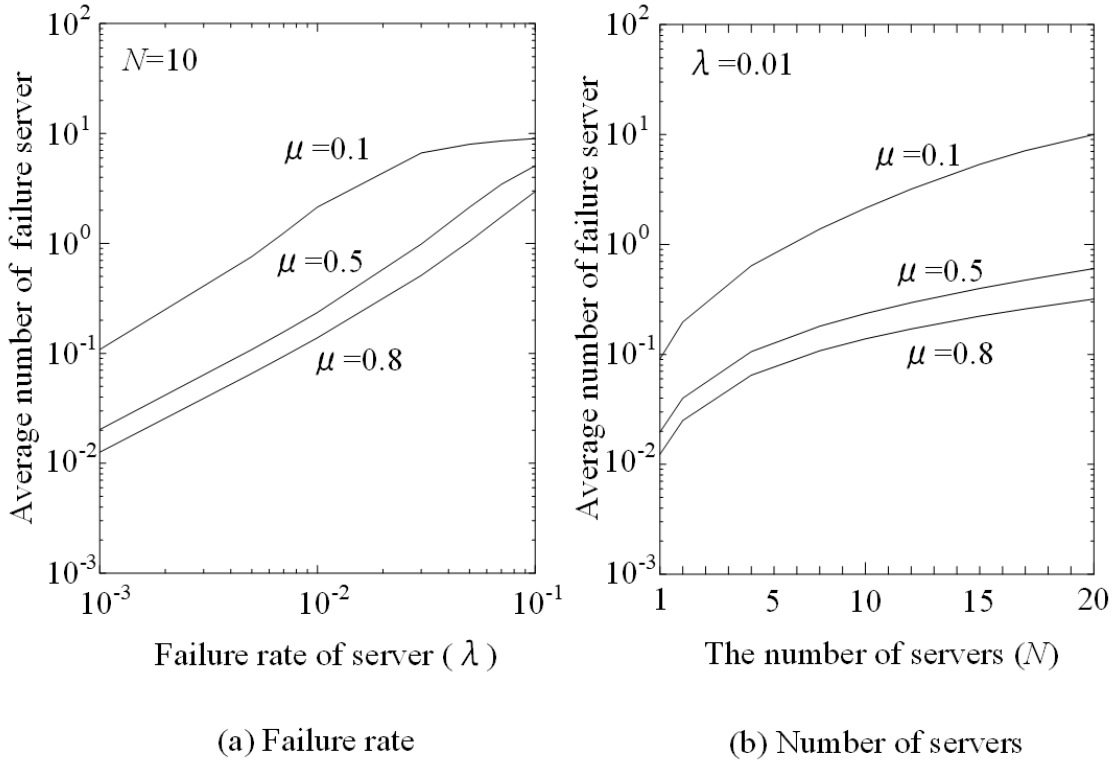
サーバの平均故障台数 $F\mathrm{u}$ は以下となり,

$$F\mathrm{u} = \sum_{n=1}^{N} np(n) \tag{f3}$$

平均復旧時間 Rt は以下となる．

$$Rt = Fu\frac{1}{\mu} \tag{f4}$$

以下に解析した結果を図 F3.2 に示す．(a)はサーバ台数を 10 台とし，サーバの故障率を可変した場合の平均故障台数を示し，(b)はサーバ台数を可変した場合の平均故障台数を示す．(a)では，いずれの復旧率においても故障率と比例して平均故障台数が増加することがわかる．(b)でもサーバ台数に応じて平均故障台数が増加している．また，(a)，(b)ともにサーバの復旧率が高い場合には，平均故障台数が低くなる．



(a) Failure rate　　(b) Number of servers

図 F3.2　平均サーバ故障台数

F4. MSBS における稼働状況を記録する Log ファイル

管理対象サーバにトラブルが発生した場合，本提案システムは検出したトラブルの種類やサーバ機能を復旧するために起動する仮想サーバの稼働状況などをその時刻とともに System log に記録する．また，図 2.3(b)で示した管理対象サーバの復旧作業を行うバックグラウンドジョブの稼働状況もその Log ファイルに記録する．バックグラウンドジョブによって記録される Log はフォアグラウンドジョブからの Log と区別できるように先頭に文字列“_ _”を挿入している．

実験において実測に使用する Log ファイルの内容，サーバ機能復旧時間と管理対象サーバの復旧時間およびクライアントにおけるサーバアクセスの切断時間に関する実測方法について示す．ここでの System log は実験用であるため，監視間隔時間において 1 秒間隔で数値を出力およびバックグラウンドジョブからの調査結果を待つ状態においても 1 秒間隔で“Wait”を出力するようにしている．クライアントにおける Access log にはクライアントが対象サービスを提供するサーバに設定されている仮想 IP アドレスに対して 1 秒間隔でアクセスを行い，そのサービス提供状態が記録される．この Log の作成には正確なサービス提供状態を調査するため図 2.4 で示した Expect プログラム言語を使用している．クライアントでの Access log には，サービスを提供できる状態であればステータス“OK”，異常状態であればステータス“XX”が時刻とともに記録される．管理対象サーバは Mail サーバで，その IP アドレスを“IPR”，Mail サーバの機能を復旧するために使用する仮想サーバの IP アドレスを“IPV”とし，その仮想 IP アドレスを“VIP”としている．

ここでは，監視間隔時間を 10 秒とし，その開始時から 5 秒経過時に管理対象サーバに対してサービス提供プログラムやネットワークのトラブルを再現する．サービスに関するトラブルは，Mail サーバのサービス提供プログラム postfix を停止，ネットワークに関するトラブルは，Mail サーバを再起動またはシャットダウンすることにより再現する．

図 F4.1 にサービス提供プログラムの再起動によりサービス機能が復旧する場合を示す．System log では，監視間隔時間内の 11 時 10 分 25 秒に管理対象サーバ上のサービス提供プログラム postfix を停止し，11 時 10 分 31 秒に管理対象サーバのネットワークを確認後，サービス異常を検出している．管理対象サーバのネットワークを再確認後，postfix を再起動し，2 秒経過後にサービス提供状態の再確認を行い，11 時 10 分 34 秒にサービス機能の復旧が確認されている．ここで，サービス機能が復旧した時刻“11:10:34”とサービスの異常を検出した時刻“11:10:31”との差が管理対象サーバの復旧時間を示し，ここでは 3 秒となる．クライアントでの Access log では，11 時 10 分 25 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，11 時 10 分 34 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，9 秒となる．

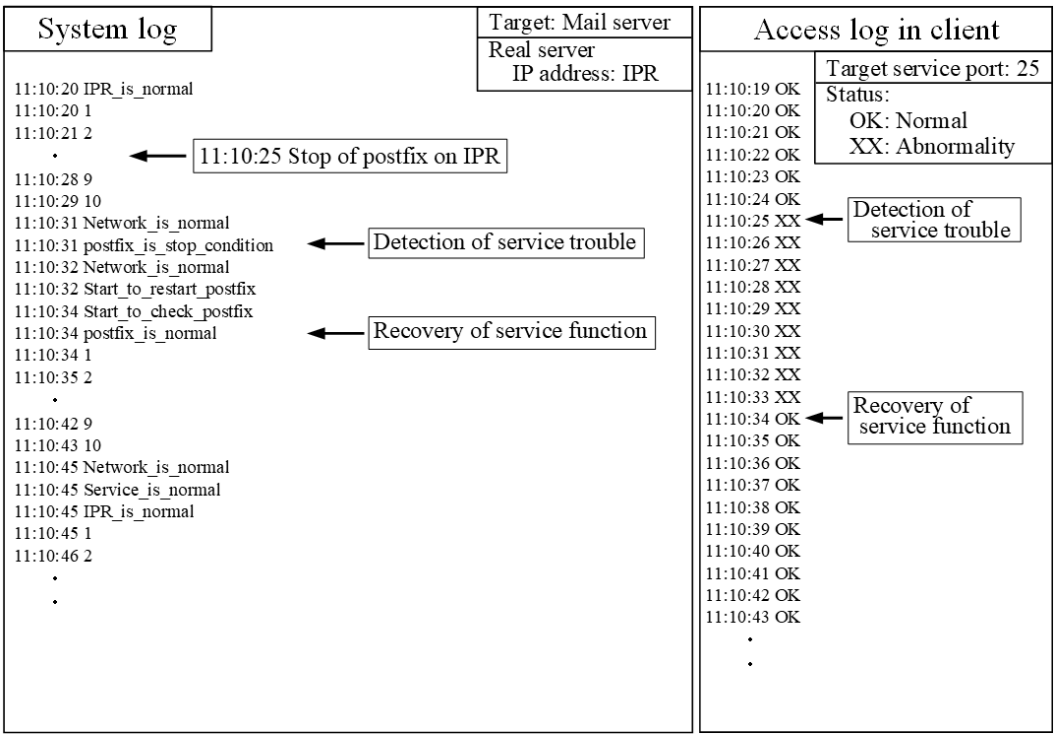


図 F4.1　サービス提供プログラムの再起動による復旧時の Log

図 F4.2 にサービスを提供しているプログラムの再起動でサービス機能が復旧せず，管理対象サーバの再起動により復旧する場合を示す．System log では，監視間隔時間内の 11 時 30 分 07 秒に管理対象サーバ上のサービス提供プログラム postfix を停止し, 11 時 30 分 13 秒に管理対象サーバのネットワークを確認後，サービス異常を検出している．管理対象サーバのネットワークを再確認後，postfix を再起動し，2 秒経過後にサービス提供状態の再確認を行っている．11 時 30 分 16 秒に復旧困難であることを確認し，管理対象サーバの仮想 IP アドレスを解除している．サーバ機能を復旧するために仮想サーバの起動処理を 11 時 30 分 16 秒に行い，11 時 30 分 18 秒に起動が完了している．その後，仮想 IP アドレスの設定を行い，11 時 30 分 18 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“11:30:18”とサービスの異常を検出した時刻“11:30:13”との差がサーバ機能の復旧時間を示し，ここでは 5 秒となる．クライアントでの Access log では，11 時 30 分 07 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し, 11 時 30 分 18 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 11 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，11 時 30 分 16 秒に管理対象サーバの再起動を開始し，11 時 31 分 04 秒に管理対象サーバの起動が完了している．その 2 秒経過後にサービス提供状態の再確認を行い，11 時 31 分 06 秒にサービス機能の復旧が確認されている．その後，図 2.3(b)で示した up.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて up.file を検出すると，管理対象サーバが正常に動作したと判断され仮想サーバの仮想 IP アドレスを解除する．その後，管理対象サーバに仮想 IP アドレスを設定し, 11 時 31 分 07 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“11:31:07”とサービスの異常を検出した時刻

“11:30:13”との差が管理対象サーバの復旧時間を示し，ここでは 54 秒となる．その管理対象サーバの復旧処理を行った 11 時 31 分 07 秒におけるクライアントでの Access log では受けているサービスに異常は見られない.

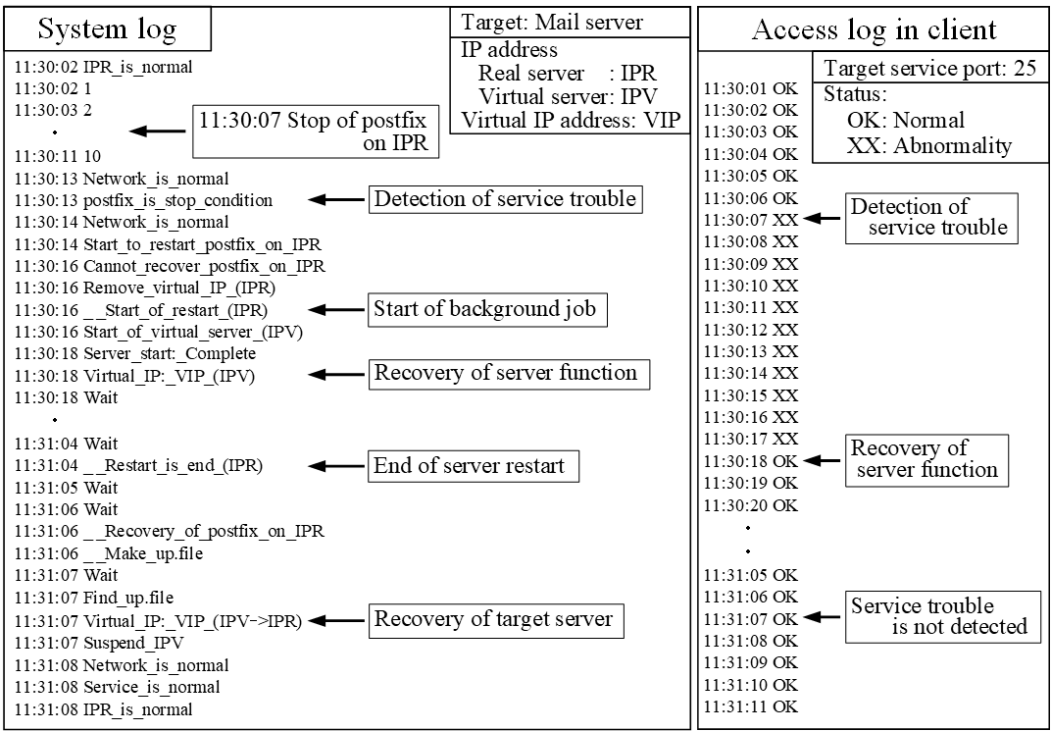


図 F4.2　管理対象サーバの再起動による復旧時の Log

図 F4.3 に管理対象サーバを再起動することによりネットワークトラブルを再現する場合を示す．System log では，監視間隔時間内の 11 時 22 分 15 秒に管理対象サーバを再起動し，11 時 22 分 22 秒に管理対象サーバのネットワーク異常を検出している．管理対象サーバのネットワークを再度確認し，11 時 22 分 24 秒に異常であることを再確認している．サーバ機能を復旧するために仮想サーバの起動処理を 11 時 22 分 24 秒に行い，11 時 22 分 26 秒に起動が完了している．その後，仮想 IP アドレスの設定を行い，11 時 22 分 26 秒に

サーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“11:22:26”とネットワーク異常を検出した時刻“11:22:22”との差がサーバ機能の復旧時間を示し，ここでは 4 秒となる．クライアントでの Access log では，11 時 22 分 15 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，11 時 22 分 26 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 11 秒となる．

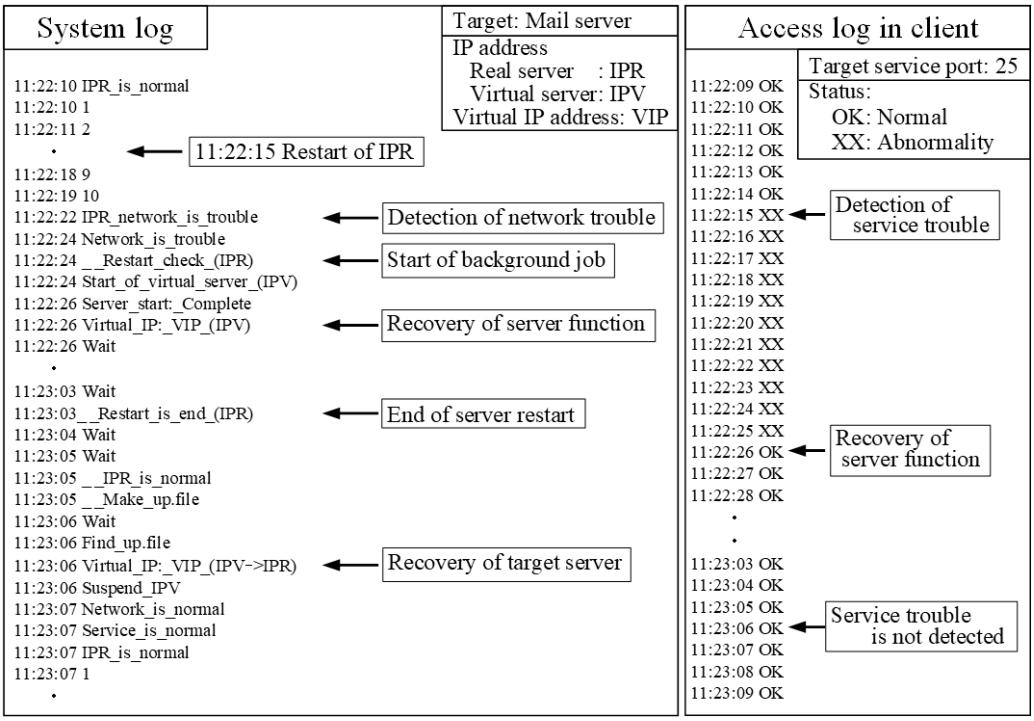


図 F4.3　管理対象サーバの再起動時の Log

また，System log においてバックグラウンドジョブの稼働状況では，11 時 22 分 24 秒に管理対象サーバが再起動状態であるかの確認を開始し，11 時 23 分 03 秒に管理対象サーバにおける再起動の完了が確認されている．その 2 秒経過後にサービス提供状態の確認を行

い，11 時 23 分 05 秒にサービス機能の復旧が確認され，図 2.3(b)で示した up.file を作成後にバックグラウンドジョブは停止する．その後，フォアグラウンドジョブにおいて up.file を検出すると，管理対象サーバが正常に動作したと判断され仮想サーバの仮想 IP アドレスを解除する．その後，管理対象サーバに仮想 IP アドレスを設定し，11 時 23 分 06 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“11:23:06”とネットワーク異常を検出した時刻“11:22:22”との差が管理対象サーバの復旧時間を示し，ここでは 44 秒となる．その管理対象サーバの復旧処理を行った 11 時 23 分 06 秒におけるクライアントでの Access log では受けているサービスに異常は見られない．

図 F4.4 に管理対象サーバをシャットダウンすることによりネットワークトラブルを再現する場合を示す．System log では，監視間隔時間内の 11 時 02 分 19 秒に管理対象サーバをシャットダウンし，11 時 02 分 26 秒に管理対象サーバのネットワーク異常を検出している．管理対象サーバのネットワークを再度確認し，11 時 02 分 28 秒に異常であることを再確認している．サーバ機能を復旧するために仮想サーバの起動処理を 11 時 02 分 28 秒に行い，11 時 02 分 30 秒に起動が完了している．その後，仮想 IP アドレスの設定を行い，11 時 02 分 30 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“11:02:30”とネットワーク異常を検出した時刻“11:02:26”との差がサーバ機能の復旧時間を示し，ここでは 4 秒となる．クライアントでの Access log では，11 時 02 分 19 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，11 時 02 分 30 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 11 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，11 時 02 分 28 秒に管理対象サーバが再起動状態であるかの確認を開始し，11 時 03 分 28 秒に管理対象サーバが停止状態であることを確認している．その後，WOL を使用して管理対象サーバを強制的

に起動させ，11 時 04 分 05 秒に起動の完了を確認している．その 2 秒経過後にサービス提供状態の確認を行い，11 時 04 分 07 秒に管理対象サーバの状態が正常であることを確認している．その後，図 2.3(b)で示した up.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて up.file を検出すると，管理対象サーバが正常に動作したと判断され仮想サーバの仮想 IP アドレスを解除する．その後，管理対象サーバに仮想 IP アドレスを設定し，11 時 04 分 08 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“11:04:08”とネットワーク異常を検出した時刻“11:02:26”との差が管理対象サーバの復旧時間を示し，ここでは 102 秒となる．その管理対象サーバの復旧処理を行った 11 時 04 分 08 秒におけるクライアントでの Access log では受けているサービスに異常は見られない．

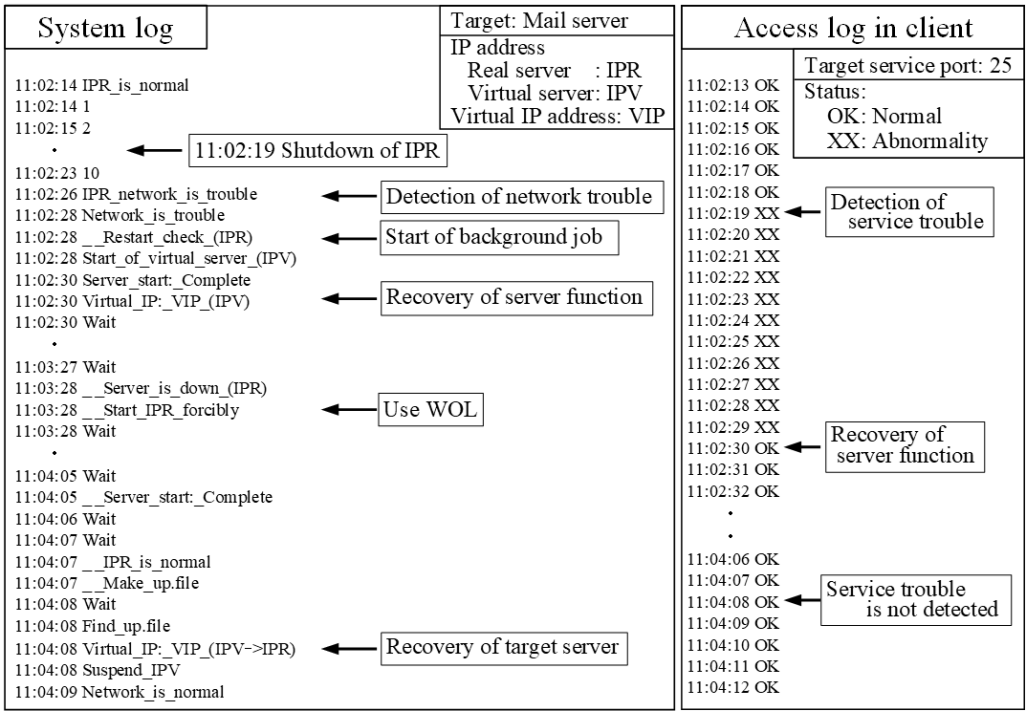


図 F4.4 管理対象サーバのシャットダウン時の Log

F5. 省電力かつ高可用性サーバシステムにおける稼働状況を記録する Log ファイル

管理対象サーバにトラブルが発生した場合，本提案システムは検出したトラブルの種類やサーバ機能を復旧するために起動するバックアップサーバの稼働状況などをその時刻とともに System log に記録する．また，図 3.7(b)で示した管理対象サーバの復旧作業を行うバックグラウンドジョブの稼働状況もその Log ファイルに記録する．バックグラウンドジョブによって記録される Log はフォアグラウンドジョブからの Log と区別できるように先頭に文字列“_ _”を挿入している．

図 F5.1 にすべてのサービスグループが Moderate condition における実験において実測に使用する Log ファイルの内容，サーバ機能復旧時間とシステムの復旧時間およびクライアントにおけるサーバアクセスの切断時間の実測方法について示す．この状態ではすべてのサービスグループを構成する仮想サーバは冗長化構成となる．ここでの System log は実験用であるため，監視間隔時間において 1 秒間隔で数値を出力するようにしている．クライアントにおける Access log にはクライアントがロードバランサを経由して対象サービスを提供するサーバに対して 1 秒間隔でアクセスを行い，そのサービス提供状態が記録される．この Log の作成には正確なサービス提供状態を調査するため，2 章の図 2.4 で示した Expect プログラム言語を使用している．クライアントでの Access log には仮想サーバがサービスを提供できる状態であればステータス“OK”が，異常状態であればステータス“XX”が時刻とともに記録される．

ここでは監視間隔時間を 10 秒とし，その開始時から 5 秒経過時に管理対象サーバに対してサービス提供プログラムやネットワークのトラブルを再現する．サービスに関するトラ

ブルは，仮想サーバである Mail サーバのサービス提供プログラム postfix を停止し，ネットワークに関するトラブルは，Mail サーバを起動している実サーバもしくは Mail サーバを再起動またはシャットダウンすることにより再現する．

(a)にサービス提供プログラムの再起動によりサービス機能が復旧する場合を示す．管理対象サーバは Mail サーバで，その IP アドレスを“IPV”とする．System log では，監視間隔時間内の 10 時 12 分 19 秒に管理対象サーバ上のサービス提供プログラム postfix を停止し，実サーバおよび管理対象サーバのネットワークを確認後，10 時 12 分 26 秒に管理対象サーバのサービス異常を検出している．実サーバおよび管理対象サーバのネットワークを再確認後，postfix を再起動し，2 秒経過後にサービス提供状態の再確認を行い，10 時 12 分 30 秒にサービス機能の復旧が確認されている．ここで，サービス機能が復旧した時刻“10:12:30”とサービスの異常を検出した時刻“10:12:26”との差が管理対象サーバの復旧時間を示し，ここでは 4 秒となる．クライアントでの Access log では，冗長化された Mail サーバに対してロードバランサ経由でアクセスしているため，サービス提供プログラム postfix を停止した 10 時 12 分 19 秒からサーバ機能の復旧時刻である 10 時 12 分 30 秒までにサービスの異常は見られない．

(b)にサービス提供プログラムの再起動でサービス機能が復旧せず，管理対象サーバの再起動により復旧する場合を示す. 管理対象サーバは Mail サーバで, その IP アドレスを“IPV”とし，Mail サーバの機能を復旧するために使用するバックアップサーバの IP アドレスは“IPB”としている．System log では，監視間隔時間内の 10 時 17 分 07 秒に管理対象サーバ上のサービス提供プログラム postfix を停止し，実サーバおよび管理対象サーバのネットワークを確認後，10 時 17 分 14 秒に管理対象サーバのサービス異常を検出している．実サーバおよび管理対象サーバのネットワークを再確認後，postfix を再起動し，2 秒経過後にサービス提供状態の再確認を行い, 10 時 17 分 18 秒に復旧困難であることを確認している．

その後，サーバ機能を復旧するためにバックアップサーバの起動処理を 10 時 17 分 18 秒に行い，10 時 17 分 27 秒に起動が完了し，ロードバランサを制御して 10 時 17 分 27 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“10:17:27”とサービスの異常を検出した時刻“10:17:14”との差がサーバ機能の復旧時間を示し，ここでは 13 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，10 時 17 分 18 秒に管理対象サーバの再起動を開始し，10 時 17 分 51 秒に管理対象サーバの再起動が完了している．その 2 秒経過後にサービス提供状態の再確認を行い，10 時 17 分 53 秒にサービス機能の復旧が確認されている．その後，図 3.7(b)で示した vup.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて vup.file を検出すると，管理対象サーバが正常に動作したと判断し，ロードバランサを制御して 10 時 17 分 54 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“10:17:54”とサービス異常を検出した時刻“10:17:14”との差が管理対象サーバの復旧時間を示し，ここでは 40 秒となる．

クライアントでの Access log では，冗長化された Mail サーバに対してロードバランサ経由でアクセスしているため，サービスの異常は見られない．

(c)に管理対象サーバを再起動することによりネットワークトラブルを再現する場合を示す．管理対象サーバの IP アドレス等は(b)と同様とする．System log では，監視間隔時間内の 10 時 25 分 20 秒に管理対象サーバを再起動し，実サーバのネットワークを確認後，10 時 25 分 29 秒に管理対象サーバのネットワーク異常を検出している．実サーバのネットワークを再度確認後，10 時 25 分 33 秒に管理対象サーバのネットワークが異常であることを再確認している．その後，サーバ機能を復旧するためにバックアップサーバの起動処理を 10 時 25 分 33 秒に行い，10 時 25 分 42 秒に起動が完了し，ロードバランサを制御して 10

時 25 分 42 秒にサーバ機能が復旧されている. ここで, サーバ機能が復旧した時刻“10:25:42”とネットワーク異常を検出した時刻“10:25:29”との差がサーバ機能の復旧時間を示し，ここでは 13 秒となる.

また，System log においてバックグラウンドジョブの稼働状況では，10 時 25 分 33 秒に管理対象サーバが再起動状態であるかの確認を開始し，10 時 25 分 53 秒に管理対象サーバにおける再起動の完了が確認されている．その 2 秒経過後にサービス提供状態の確認を行い，10 時 25 分 55 秒にサービス機能の復旧が確認され，図 3.7(b)で示した vup.file を作成後にバックグラウンドジョブは停止する．その後，フォアグラウンドジョブにおいて vup.file を検出すると，管理対象サーバが正常に動作したと判断し，ロードバランサを制御して 10 時 25 分 56 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“10:25:56”とネットワーク異常を検出した時刻“10:25:29”との差が管理対象サーバの復旧時間を示し，ここでは 27 秒となる．クライアントでの Access log では，冗長化された Mail サーバに対してロードバランサ経由でアクセスしているため，サービスの異常は見られない.

(d)に管理対象サーバをシャットダウンすることによりネットワークトラブルを再現する場合を示す．管理対象サーバの IP アドレス等は(b)と同様とする．System log では，監視間隔時間内の 10 時 32 分 15 秒に管理対象サーバをシャットダウンし, 実サーバのネットワークを確認後，10 時 32 分 24 秒に管理対象サーバのネットワーク異常を検出している．実サーバのネットワークを再度確認後，10 時 32 分 28 秒に管理対象サーバのネットワークが異常であることを再確認している．その後，サーバ機能を復旧するためにバックアップサーバの起動処理を 10 時 32 分 28 秒に行い，10 時 32 分 37 秒に起動が完了し，ロードバランサを制御して 10 時 32 分 37 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“10:32:37”とネットワーク異常を検出した時刻“10:32:24”との差がサーバ機

能の復旧時間を示し，ここでは 13 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，10 時 32 分 28 秒に管理対象サーバが再起動状態であるかの確認を開始し，システムで設定（仮想サーバにおける再起動時間の約 1.5 倍）した 49 秒経過後の 10 時 33 分 17 秒に管理対象サーバが停止状態であることを確認している．その後，管理対象サーバを強制的に起動させ，10 時 33 分 41 秒に起動の完了を確認している．その 2 秒経過後にサービス提供状態の確認を行い，10 時 33 分 43 秒に管理対象サーバの状態が正常であることを確認している．その後，図 3.7(b)で示した vup.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて vup.file を検出すると，管理対象サーバが正常に動作したと判断し，ロードバランサを制御して 10 時 33 分 44 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“10:33:44”とネットワーク異常を検出した時刻“10:32:24”との差が管理対象サーバの復旧時間を示し，ここでは 80 秒となる．クライアントでの Access log では，冗長化された Mail サーバに対してロードバランサ経由でアクセスしているため，サービス異常は見られない．

(e)に実サーバを再起動することによりネットワークトラブルを再現する場合を示す．トラブルが発生する実サーバの IP アドレスを“IPR”，Mail サーバと Web サーバはその実サーバ上に起動しており，管理対象の Mail サーバの IP アドレスを“IPV1”とし，Web サーバを“IPV2”とする．Mail サーバと Web サーバの機能を復旧するために使用するバックアップサーバの IP アドレスはそれぞれ“IPB1”，“IPB2”としている．System log では，監視間隔時間内の 10 時 46 分 26 秒に実サーバを再起動し，10 時 46 分 34 秒に実サーバのネットワーク異常を検出している．実サーバのネットワークを再度確認し，10 時 46 分 37 秒に異常であることを再確認している．その後，サーバ機能を復旧するために 2 台のバックアップサーバの起動処理を 10 時 46 分 37 秒に行い，10 時 46 分 53 秒に起動が完了し，

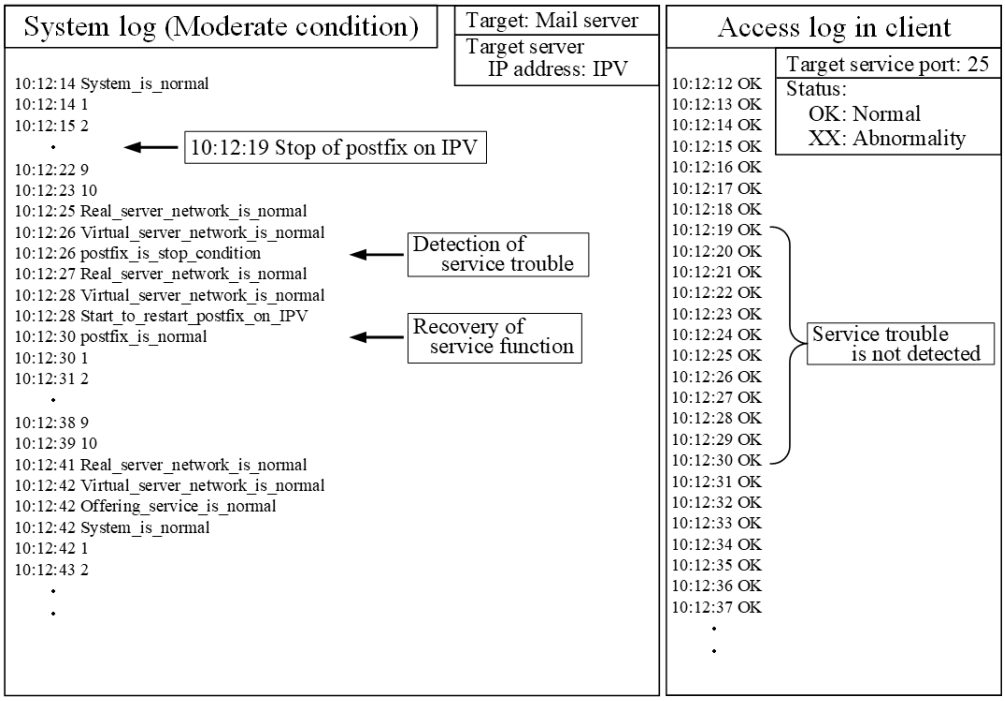
ロードバランサを制御して 10 時 46 分 53 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“10:46:53”とネットワーク異常を検出した時刻“10:46:34”との差がサーバ機能の復旧時間を示し，ここでは 19 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，10 時 46 分 37 秒に実サーバが再起動状態であるかの確認を開始し，10 時 47 分 32 秒に実サーバにおける再起動の完了が確認されている．その後，図 3.7(b)で示した rup.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて rup.file を検出すると，10 時 47 分 33 秒に仮想サーバ IPV1 と IPV2 の起動処理を開始し，10 時 48 分 22 秒にそれらの起動完了を確認している．その 2 秒経過後にロードバランサを制御して 10 時 48 分 24 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“10:48:24”とネットワーク異常を検出した時刻“10:46:34”との差が管理対象サーバの復旧時間を示し，ここでは 110 秒となる．クライアントでの Access log では，冗長化された Mail サーバに対してロードバランサ経由でアクセスしているため，サービスの異常は見られない．

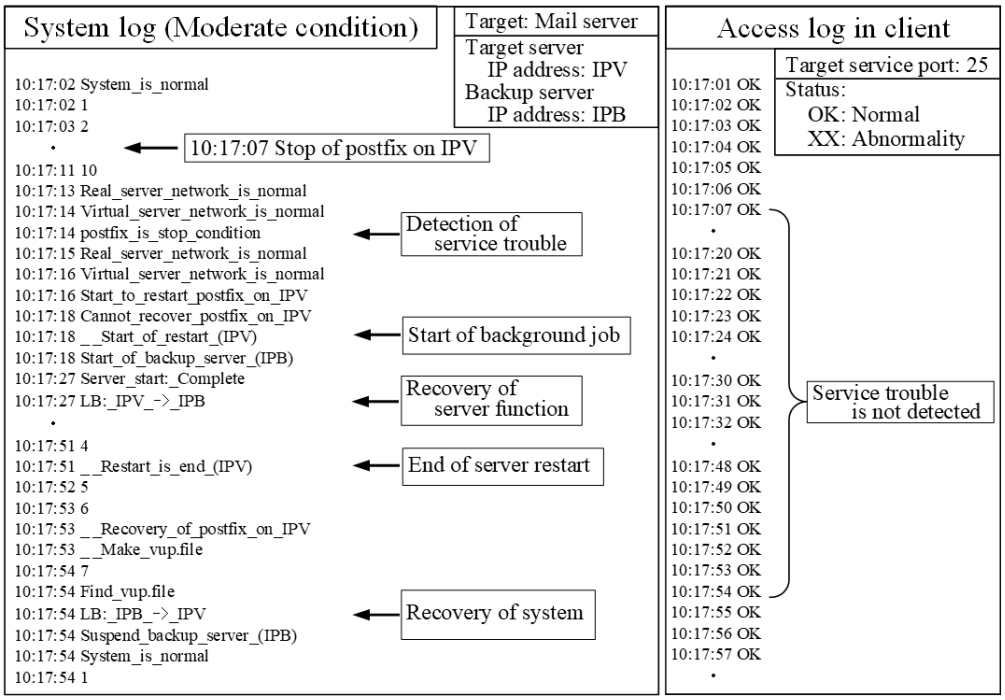
(f)に実サーバをシャットダウンすることによりネットワークトラブルを再現する場合を示す．実サーバの IP アドレス等は(e)と同様とする．System log では，監視間隔時間内の 10 時 53 分 13 秒に実サーバをシャットダウンし，10 時 53 分 21 秒に実サーバのネットワーク異常を検出している．実サーバのネットワークを再度確認し，10 時 53 分 24 秒に異常であることを再確認している．その後，サーバ機能を復旧するために 2 台のバックアップサーバの起動処理を 10 時 53 分 24 秒に行い，10 時 53 分 40 秒に起動が完了し，ロードバランサを制御して 10 時 53 分 40 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“10:53:40”とネットワーク異常を検出した時刻“10:53:21”との差がサーバ機能の復旧時間を示し，ここでは 19 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，10 時 53 分 24 秒に

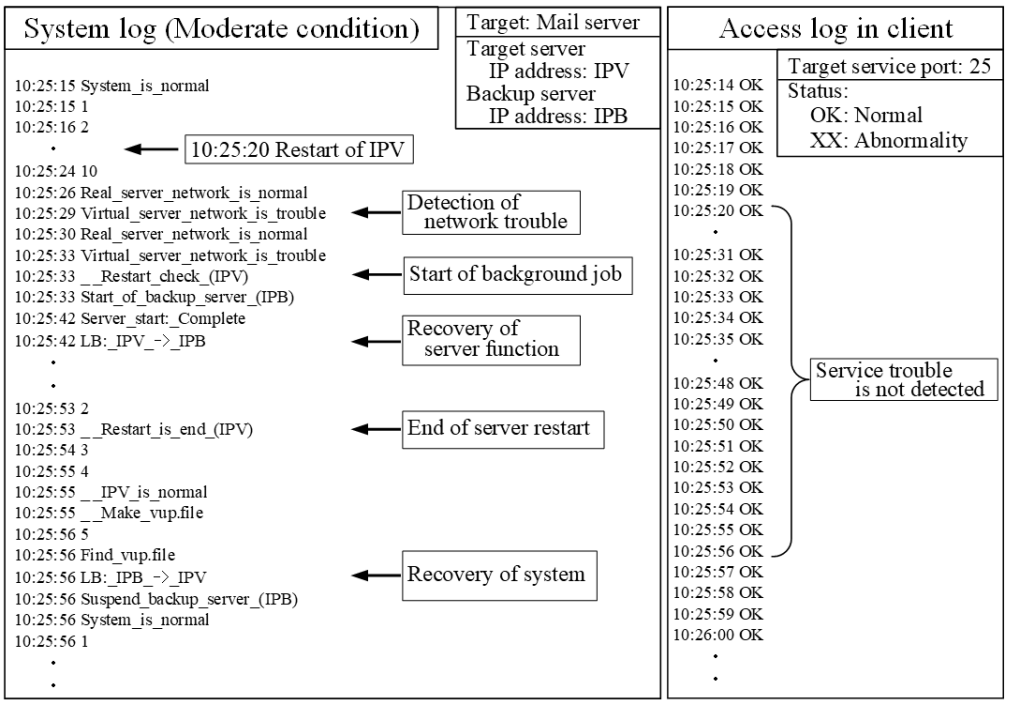
実サーバが再起動状態であるかの確認を開始し，システムで設定（実サーバにおける再起動時間の 1.5 倍）した 99 秒経過後の 10 時 55 分 03 秒に実サーバが停止状態であることを確認している．その後，WOL を使用して実サーバを強制的に起動させ，10 時 56 分 02 秒に起動の完了を確認し，図 3.7(b)で示した rup.file を作成後にバックグラウンドジョブは停止する．フォアグラウンドジョブにおいて rup.file を検出すると，10 時 56 分 03 秒に仮想サーバ IPV1 と IPV2 の起動処理を開始し，10 時 56 分 52 秒にそれらの起動完了を確認している. その 2 秒経過後にロードバランサを制御して 10 時 56 分 54 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“10:56:54”とネットワーク異常を検出した時刻“10:53:21”との差が管理対象サーバの復旧時間を示し，ここでは 213 秒となる．クライアントでの Access log では，冗長化された Mail サーバに対してロードバランサ経由でアクセスしているため，サービスの異常は見られない.



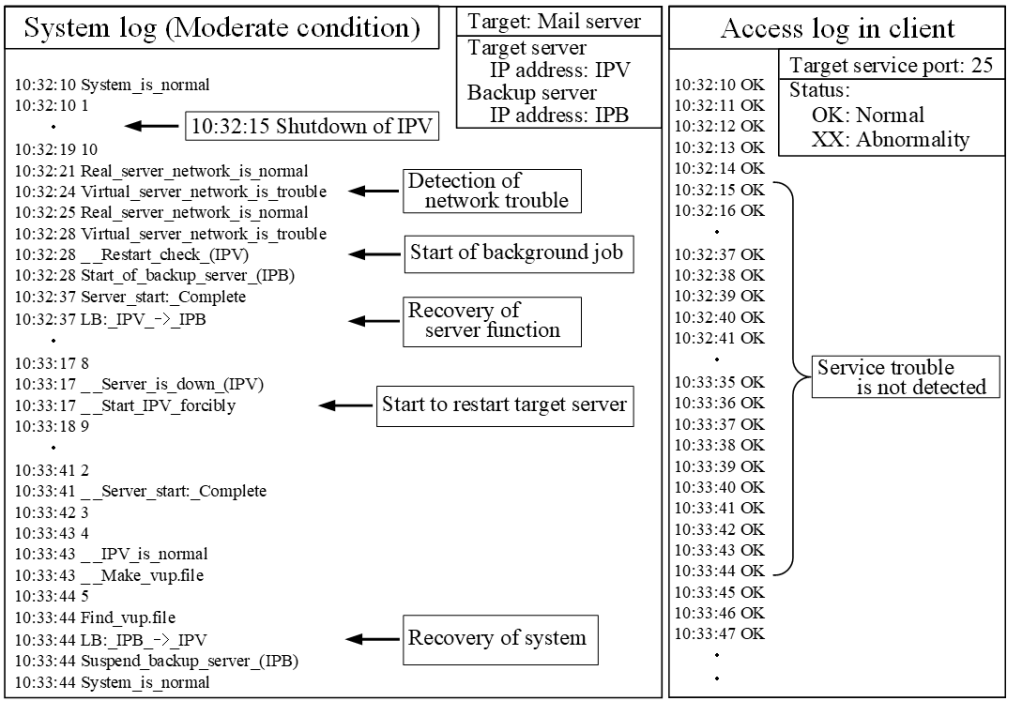
(a) サービス提供プログラムの再起動による復旧



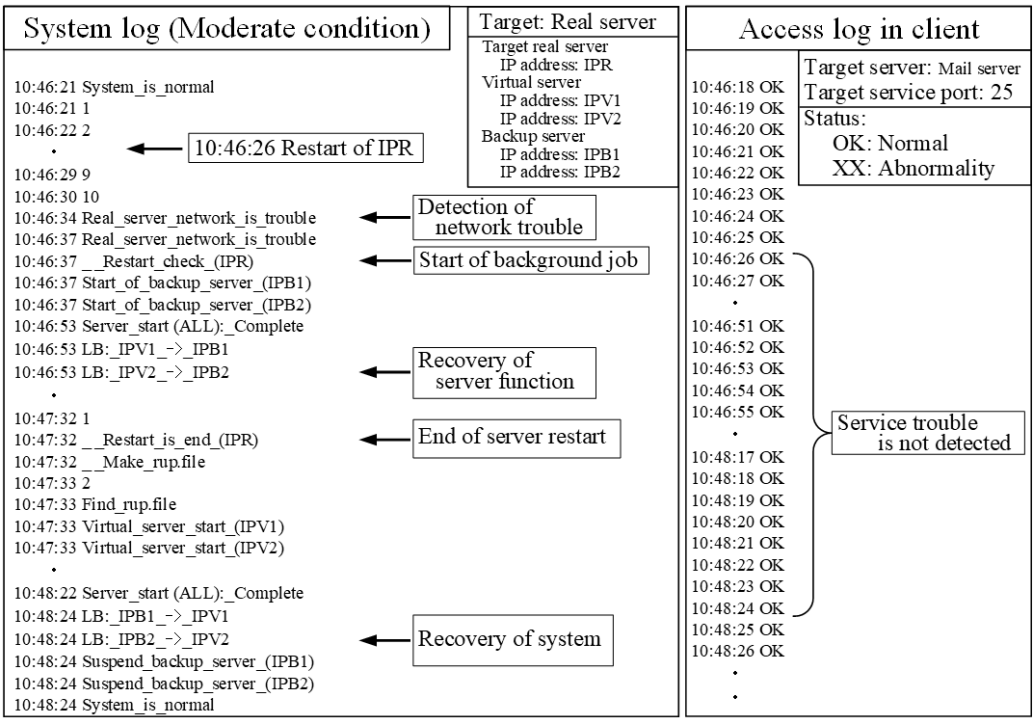
(b) 管理対象サーバの再起動による復旧



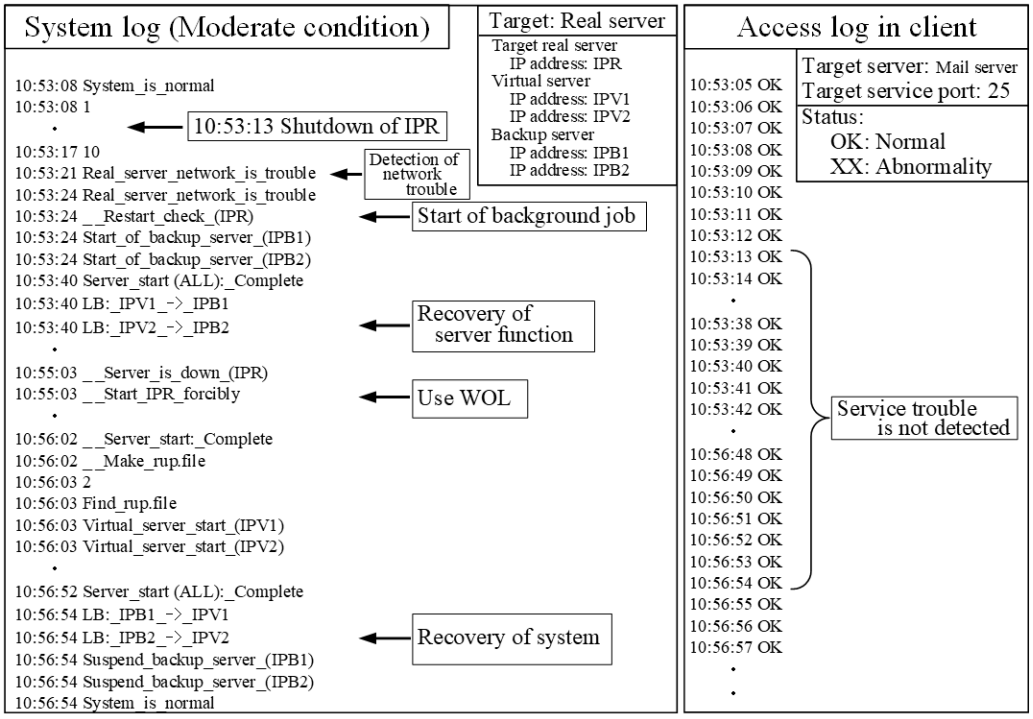
(c) 管理対象サーバの再起動



(d) 管理対象サーバのシャットダウン



(e) 実サーバの再起動



(f) 実サーバのシャットダウン

F5.1　稼働状況を記録する Log ファイル（Moderate condition）

図 F5.2 にすべてのサービスグループが Light condition における実験において実測に使用する Log ファイルの内容，サーバ機能復旧時間とシステムの復旧時間およびクライアントのサーバアクセスにおける切断時間の実測方法について示す．この状態ではクライアントにサービスを提供する仮想サーバは冗長化構成されていない．ここでの System log やクライアントにおける Access log の仕様は F5.1 と同様とする．

ここでは監視間隔時間を 10 秒とし，その開始時から 5 秒経過時に管理対象サーバに対してサービス提供プログラムやネットワークのトラブルを再現する．サービスに関するトラブルは，仮想サーバである Mail サーバのサービス提供プログラム postfix の停止を行い，ネットワークに関するトラブルは，Mail サーバを起動している実サーバもしくは Mail サ

ーバを再起動またはシャットダウンすることにより再現する．

(a)にサービス提供プログラムの再起動によりサービス機能が復旧する場合を示す．管理対象サーバは Mail サーバで，その IP アドレスを“IPV”とする．System log では，監視間隔時間内の 12 時 28 分 15 秒に管理対象サーバ上のサービス提供プログラム postfix を停止し，実サーバおよび管理対象サーバのネットワークを確認後，12 時 28 分 22 秒に管理対象サーバのサービス異常を検出している．実サーバおよび管理対象サーバのネットワークを再確認後，postfix を再起動し，2 秒経過後にサービス提供状態の再確認を行い，12 時 28 分 26 秒にサービス機能の復旧が確認されている．ここで，サービス機能が復旧した時刻“12:28:26”とサービスの異常を検出した時刻“12:28:22”との差が管理対象サーバの復旧時間を示し，ここでは 4 秒となる．

クライアントでの Access log では，Mail サーバが冗長化されていないため，12 時 28 分 15 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，12 時 28 分 25 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 10 秒となる．ここで，System log では 12 時 28 分 26 秒で対象サービスの正常を確認しているが，これはサービス再起動後の 2 秒経過後にサービス提供状態の再確認をするためである．

(b)にサービス提供プログラムの再起動でサービス機能が復旧せず，管理対象サーバの再起動により復旧する場合を示す．管理対象サーバは Mail サーバで，その IP アドレスを“IPV1”とする．その機能の復旧に関しては，サスペンド状態の実サーバを起動し，そこで起動している仮想サーバを使用する．実サーバの IP アドレスを“IPR”，仮想サーバの IP アドレスを“IPV2”とする．System log では，監視間隔時間内の 12 時 48 分 10 秒に管理対象サーバ上のサービス提供プログラム postfix を停止し，実サーバおよび管理対象サーバのネットワークを確認後，12 時 48 分 17 秒に管理対象サーバのサービス異常を検出して

いる．実サーバおよび管理対象サーバのネットワークを再確認後，postfix を再起動し，2 秒経過後にサービス提供状態の再確認を行い，12 時 48 分 21 秒に復旧困難であることを確認している．その後，サーバ機能を復旧するためにサスペンド状態の実サーバ IPR の起動処理を 12 時 48 分 21 秒に行い，12 時 48 分 30 秒に起動が完了し，ロードバランサを制御して 12 時 48 分 30 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“12:48:30”とサービスの異常を検出した時刻“12:48:17”との差がサーバ機能の復旧時間を示し，ここでは 13 秒となる．クライアントでの Access log では，Mail サーバが冗長化されていないため，12 時 48 分 10 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，12 時 48 分 31 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 21 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，12 時 48 分 21 秒に管理対象サーバの再起動を開始し，12 時 48 分 54 秒に管理対象サーバの起動が完了している．その 2 秒経過後にサービス提供状態の再確認を行い，12 時 48 分 56 秒にサービス機能の復旧が確認されている．その後，図 3.7(b)で示した vup.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて vup.file を検出すると，管理対象サーバが正常に動作したと判断し，ロードバランサを制御して 12 時 48 分 57 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“12:48:57”とサービスの異常を検出した時刻“12:48:17”との差が管理対象サーバの復旧時間を示し，ここでは 40 秒となる．その管理対象サーバの復旧処理を行った 12 時 48 分 57 秒におけるクライアントでの Access log では受けているサービスに異常は見られない．

(c)に管理対象サーバを再起動することによりネットワークトラブルを再現する場合を示す．管理対象サーバおよび実サーバの IP アドレス等は(b)と同様とする．System log では，監視間隔時間内の 12 時 56 分 28 秒に管理対象サーバを再起動し，実サーバのネットワーク

を確認後，12 時 56 分 37 秒に管理対象サーバのネットワーク異常を検出している．実サーバのネットワークを再度確認後，12 時 56 分 41 秒に管理対象サーバのネットワークが異常であることを再確認している．その後，サーバ機能を復旧するためにサスペンド状態の実サーバ IPR の起動処理を 12 時 56 分 41 秒に行い，12 時 56 分 50 秒に起動が完了し，ロードバランサを制御して 12 時 56 分 50 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“12:56:50”とネットワーク異常を検出した時刻“12:56:37”との差がサーバ機能の復旧時間を示し，ここでは 13 秒となる．クライアントでの Access log では，Mail サーバが冗長化されていないため, 12 時 56 分 28 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，12 時 56 分 51 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 23 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，12 時 56 分 41 秒に管理対象サーバが再起動状態であるかの確認を開始し，12 時 57 分 01 秒に管理対象サーバにおける再起動の完了が確認されている．その 2 秒経過後にサービス提供状態の確認を行い，12 時 57 分 03 秒にサービス機能の復旧が確認され，図 3.7(b)で示した vup.file を作成後にバックグラウンドジョブは停止する．その後，フォアグラウンドジョブにおいて vup.file を検出すると，管理対象サーバが正常に動作したと判断し，ロードバランサを制御して 12 時 57 分 04 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“12:57:04”とネットワーク異常を検出した時刻“12:56:37”との差が管理対象サーバの復旧時間を示し，ここでは 27 秒となる．その管理対象サーバの復旧処理を行った 12 時 57 分 04 秒におけるクライアントでの Access log では受けているサービスに異常は見られない．

(d)に管理対象サーバをシャットダウンすることによりネットワークトラブルを再現する場合を示す．管理対象サーバおよび実サーバの IP アドレス等は(b)と同様とする．System log では，監視間隔時間内の 13 時 12 分 21 秒に管理対象サーバをシャットダウンし，実サーバのネットワークを確認後，13 時 12 分 30 秒に管理対象サーバのネットワーク異常を検出している．実サーバのネットワークを再度確認後，13 時 12 分 34 秒に管理対象サーバのネットワークが異常であることを再確認している．その後，サーバ機能を復旧するためにサスペンド状態の実サーバ IPR の起動処理を 13 時 12 分 34 秒に行い，13 時 12 分 43 秒に起動が完了し，ロードバランサを制御して 13 時 12 分 43 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“13:12:43”とネットワーク異常を検出した時刻“13:12:30”との差がサーバ機能の復旧時間を示し，ここでは 13 秒となる．クライアントでの Access log では，Mail サーバが冗長化されていないため，13 時 12 分 21 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，13 時 12 分 44 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 23 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，13 時 12 分 34 秒に管理対象サーバが再起動状態であるかの確認を開始し，システムで設定（仮想サーバにおける再起動時間の約 1.5 倍）した 49 秒経過後の 13 時 13 分 23 秒に管理対象サーバが停止状態であることを確認している．その後，管理対象サーバを強制的に起動させ，13 時 13 分 47 秒に起動の完了を確認している．その 2 秒経過後にサービス提供状態の確認を行い，13 時 13 分 49 秒に管理対象サーバの状態が正常であることを確認している．その後，図 3.7(b)で示した vup.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて vup.file を検出すると，管理対象サーバが正常に動作したと判断し，ロードバランサを制御して 13 時 13 分 50 秒に管理対象サーバが復旧している．ここで，管理対象

サーバが復旧した時刻“13:13:50”とネットワーク異常を検出した時刻“13:12:30”との差が管理対象サーバの復旧時間を示し，ここでは 80 秒となる．その管理対象サーバの復旧処理を行った 13 時 13 分 50 秒におけるクライアントでの Access log では受けているサービスに異常は見られない．

(e)に実サーバを再起動することによりネットワークトラブルを再現する場合を示す．トラブルが発生する実サーバの IP アドレスを“IPR1”，Mail サーバ，Web サーバと FTP サーバはその実サーバ上に起動しており，管理対象である Mail サーバの IP アドレスを“IPV1”，Web サーバを“IPV2”とし，FTP サーバを“IPB”とする．その機能の復旧に関しては，サスペンド状態の実サーバを 2 台起動し，そこで起動している仮想サーバを使用する. 実サーバの IP アドレスを“IPR2”と“IPR3”, 仮想サーバの IP アドレスを“IPV3”，“IPV4”と“IPV5”とする．System log では，監視間隔時間内の 13 時 21 分 16 秒に実サーバを再起動し，13 時 21 分 24 秒に実サーバのネットワーク異常を検出している．実サーバのネットワークを再度確認し，13 時 21 分 27 秒に異常であることを再確認している．その後，サーバ機能を復旧するために 2 台のサスペンド状態の実サーバ IPR2 と IPR3 の起動処理を 13 時 21 分 27 秒に行い，13 時 21 分 36 秒に起動が完了し，ロードバランサを制御して 13 時 21 分 36 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“13:21:36”とネットワーク異常を検出した時刻“13:21:24”との差がサーバ機能の復旧時間を示し，ここでは 12 秒となる．クライアントでの Access log では，Mail サーバが冗長化されていないため，13 時 21 分 16 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，13 時 21 分 37 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 21 秒となる．

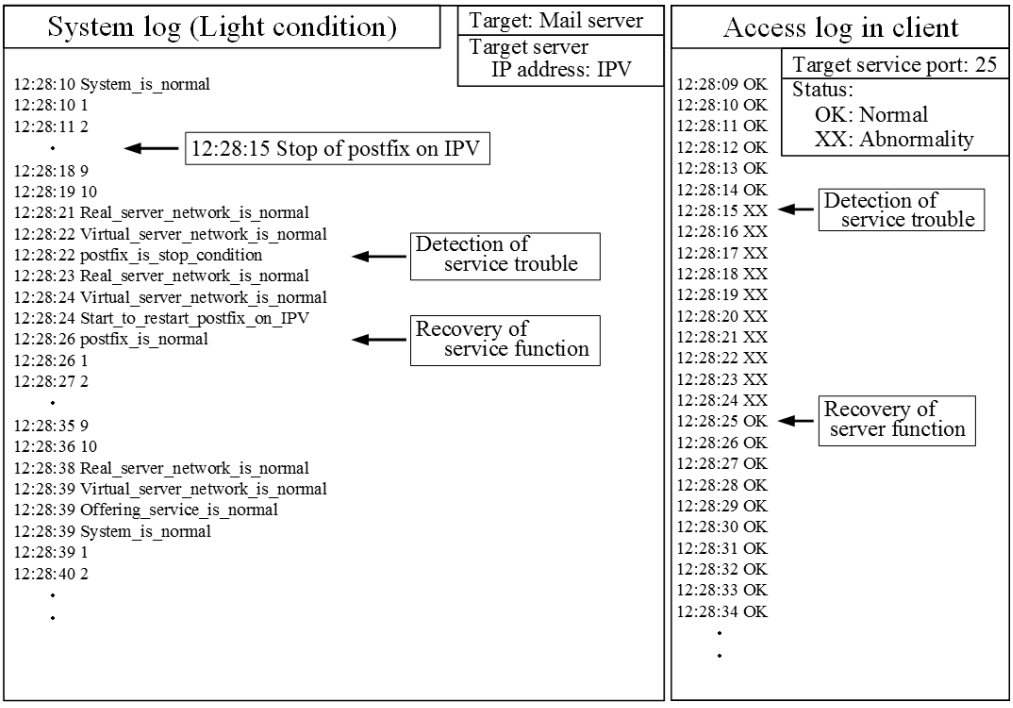
また，System log においてバックグラウンドジョブの稼働状況では，13 時 21 分 27 秒に実サーバが再起動状態であるかの確認を開始し，13 時 22 分 22 秒に実サーバにおける再起

動の完了が確認されている．その後，図 3.7(b)で示した rup.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて rup.file を検出すると，13 時 22 分 23 秒に仮想サーバ IPV1，IPV2 と IPB の起動処理を開始し，13 時 23 分 13 秒にそれらの起動完了を確認している. その 2 秒経過後にロードバランサを制御して 13 時 23 分 15 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“13:23:15”とネットワーク異常を検出した時刻“13:21:24”との差が管理対象サーバの復旧時間を示し，ここでは 111 秒となる. その管理対象サーバの復旧処理を行った 13 時 23 分 15 秒におけるクライアントでの Access log では受けているサービスに異常は見られない.

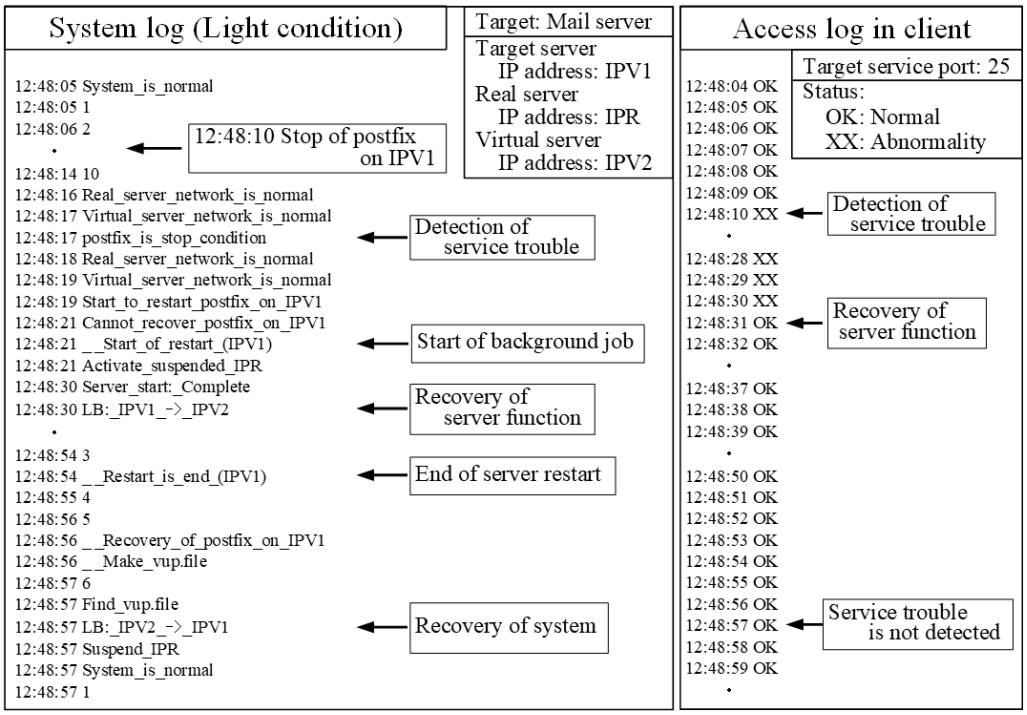
(f)に実サーバをシャットダウンすることによりネットワークトラブルを再現する場合を示す．管理対象サーバおよび実サーバの IP アドレス等は(e)と同様とする．System log では，監視間隔時間内の 13 時 34 分 21 秒に実サーバをシャットダウンし，13 時 34 分 29 秒に実サーバのネットワーク異常を検出している．実サーバのネットワークを再度確認し，13 時 34 分 32 秒に異常であることを再確認している．その後，サーバ機能を復旧するために 2 台のサスペンド状態の実サーバ IPR2 と IPR3 の起動処理を 13 時 34 分 32 秒に行い，13 時 34 分 41 秒に起動が完了し，ロードバランサを制御して 13 時 34 分 41 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“13:34:41”とネットワーク異常を検出した時刻“13:34:29”との差がサーバ機能の復旧時間を示し，ここでは 12 秒となる．クライアントでの Access log では，Mail サーバが冗長化されていないため，13 時 34 分 21 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，13 時 34 分 42 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 21 秒となる．

また，System log においてバックグラウンドジョブの稼働状況では，13 時 34 分 32 秒に実サーバが再起動状態であるかの確認を開始し，システムで設定（実サーバにおける再起

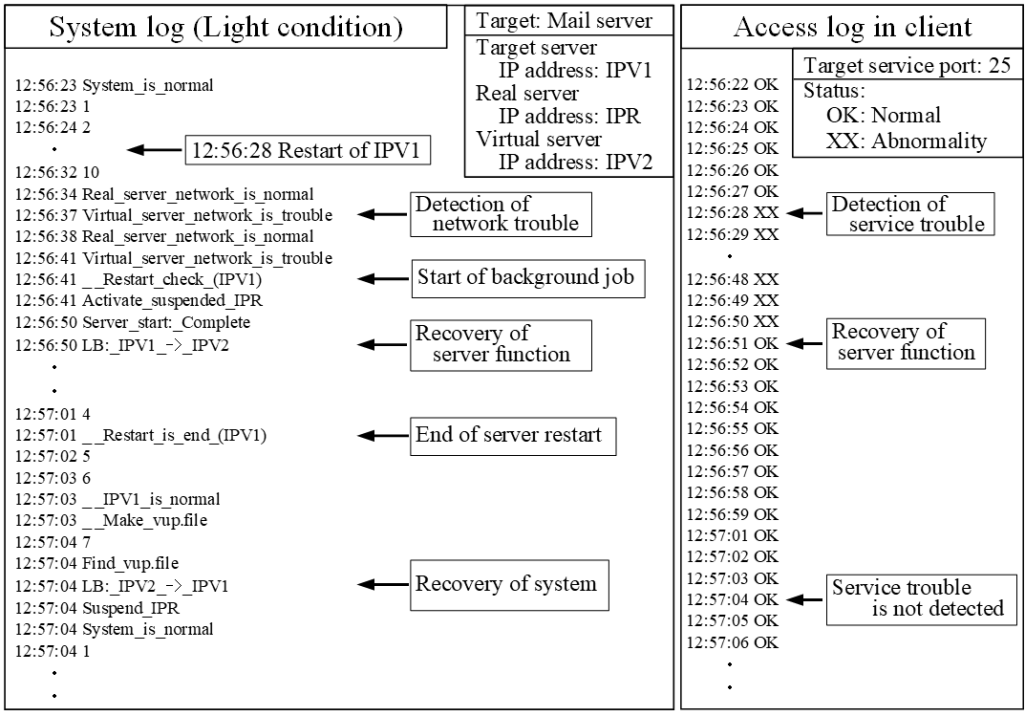
動時間の 1.5 倍）した 99 秒経過後の 13 時 36 分 11 秒に実サーバが停止状態であることを確認している．その後，WOL を使用して実サーバを強制的に起動させ，13 時 37 分 10 秒に起動の完了を確認し，図 3.7(b)で示した rup.file を作成後にバックグラウンドジョブは停止する．フォアグラウンドジョブにおいて rup.file を検出すると，13 時 37 分 11 秒に仮想サーバ IPV1，IPV2 と IPB の起動処理を開始し，13 時 38 分 01 秒にそれらの起動完了を確認している. その 2 秒経過後にロードバランサを制御して 13 時 38 分 03 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“13:38:03”とネットワーク異常を検出した時刻“13:34:29”との差が管理対象サーバの復旧時間を示し，ここでは 214 秒となる. その管理対象サーバの復旧処理を行った 13 時 38 分 03 秒におけるクライアントでの Access log では受けているサービスに異常は見られない.



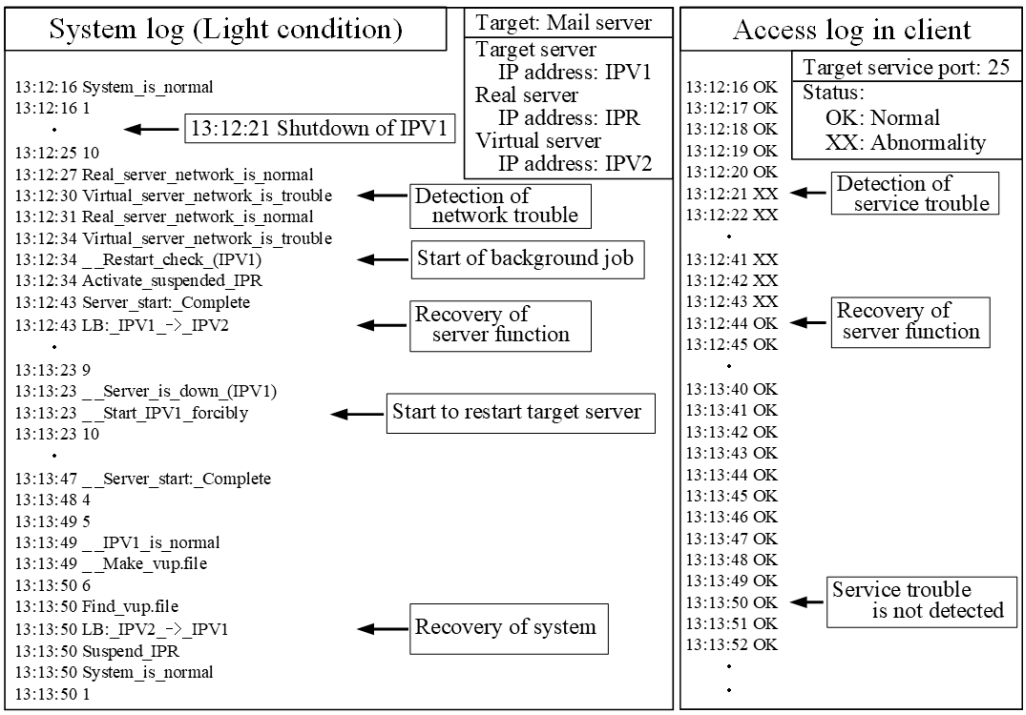
(a) サービス提供プログラムの再起動による復旧



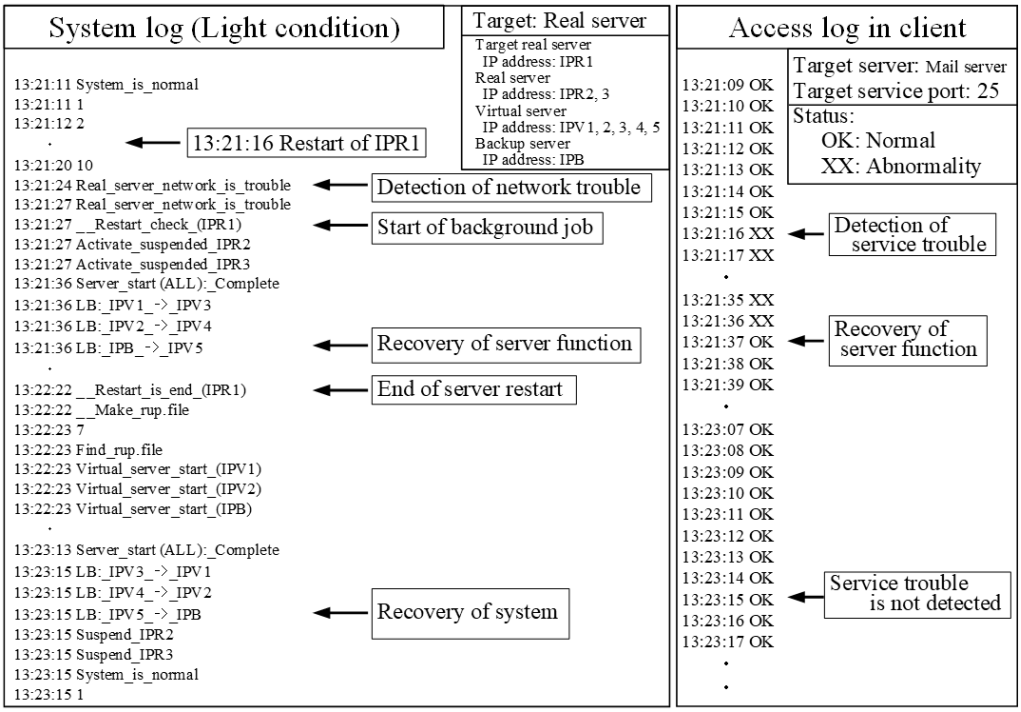
(b) 管理対象サーバの再起動による復旧



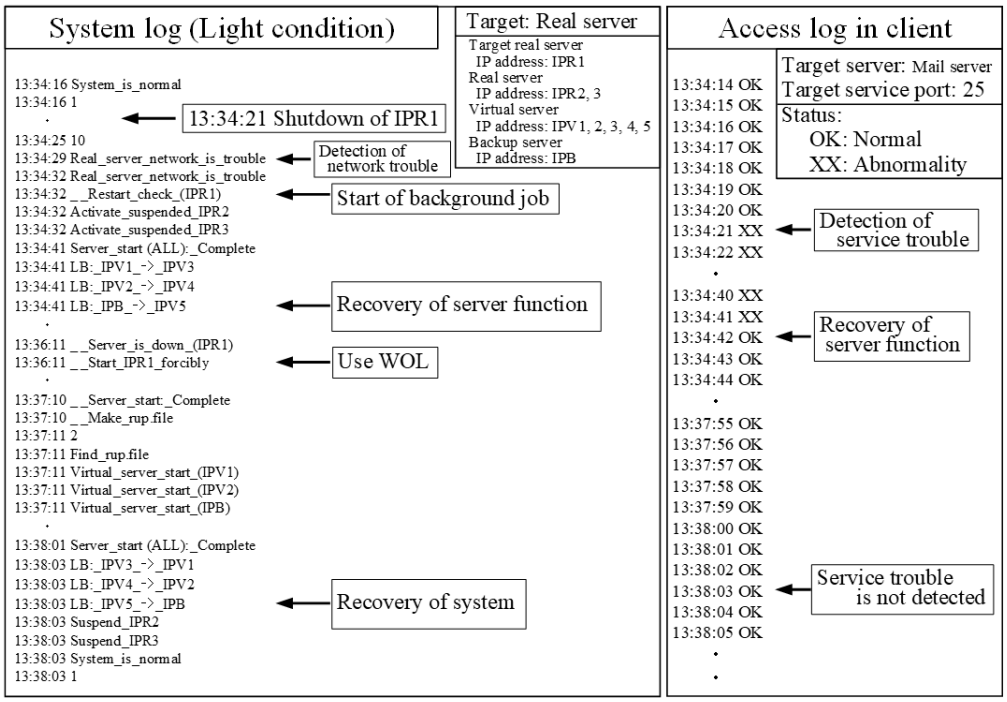
(c) 管理対象サーバの再起動



(d) 管理対象サーバのシャットダウン



(e) 実サーバの再起動



(f) 実サーバのシャットダウン

図 F5.2　稼働状況を記録する Log ファイル（Light condition）

F6. P2P 方式サーバ管理システムにおける稼働状況を記録する Log ファイル

管理対象サーバにトラブルが発生した場合，本提案システムは検出したトラブルの種類やサーバ機能を復旧するために起動する仮想サーバの稼働状況などを，その時刻とともに System log に記録する．また，図 5.7(b)で示した管理対象サーバの復旧作業を行うバックグラウンドジョブの稼働状況もその Log ファイルに記録する．バックグラウンドジョブによって記録される Log はフォアグラウンドジョブからの Log と区別できるように先頭に文字列“_ _”を挿入している．

実験において実測に使用する Log ファイルの内容，サーバ機能の復旧時間，管理対象サーバの復旧時間およびクライアントのサーバアクセスにおける切断時間の実測方法につい

て示す．各管理サーバは図 5.8 で示したように，同期をとりながら管理対象サーバの監視を行う．P1 で動作する管理サーバの Log を“System log(P1)”，P2 で動作する管理サーバの Log を“System log(P2)”とする．ここでの System log(P1)は実験用であるため，監視間隔時間において 1 秒間隔で数値を出力およびバックグラウンドジョブからの調査結果を待つ状態においても 1 秒間隔で“Wait”を出力するようにしている．System log(P2)は P1 のサーバと管理の同期をとるため，図 5.8 で示した S.file の受信状態を 1 秒間隔毎に調査して“P2_wait”を出力している．クライアントにおける Access log にはクライアントが対象サービスを提供するサーバに対し，仮想 IP アドレスを指定して 1 秒間隔でアクセスを行い，そのサービス提供状態が記録される．この Log の作成には正確なサービス提供状態を調査するため 2 章の図 2.4 で示した Expect プログラム言語を使用している．クライアントでの Access log にはサービスを提供できる状態であればステータス“OK”が，異常状態であればステータス“XX”が時刻とともに記録される．管理対象の実サーバは FTP サーバで，その IP アドレスを“IPR”，FTP サーバの機能を復旧するために使用する仮想サーバの IP アドレスは“IPV”とし，その仮想 IP アドレスを“VIP”としている．

ここでは監視間隔時間を 10 秒とし，その開始時から 5 秒経過時に管理対象サーバに対してサービス提供プログラムやネットワークのトラブルを再現する．サービスに関するトラブルは，FTP サーバのサービス提供プログラム vsftpd の停止を行い，ネットワークに関するトラブルは，FTP サーバを再起動またはシャットダウンすることにより再現する．

図 F6.1 にサービス提供プログラムの再起動によりサービス機能が復旧する場合を示す．System log(P1)では，監視間隔時間内の 13 時 05 分 07 秒に管理対象サーバ上のサービス提供プログラム vsftpd を停止し，13 時 05 分 12.30 秒に管理対象サーバのネットワークを確認後，サービス異常を検出している．管理対象サーバのネットワークを再確認後，vsftpd を再起動し，サービス提供状態の再確認を行っている．13 時 05 分 12.64 秒にサービス機能

の復旧が確認され，P2 の管理サーバに復旧を伝達するため，図 5.7(a)に示した R.file を送信して通常の監視状態に戻る．ここで，サービス機能が復旧した時刻“13:05:12.64”とサービスの異常を検出した時刻“13:05:12.34”との差が管理対象サーバの復旧時間を示し，ここでは 0.3 秒となる．System log(P2)では，P2 の管理サーバが管理対象サーバのサービス異常を 13 時 05 分 12.57 秒に検出後， P1 の結果を待つ状態となる．P2 の管理サーバは 13 時 05 分 13.63 秒に R.file を受信後，管理対象サーバの監視状態に戻る．クライアントでの Access log では，13 時 05 分 07.64 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し，13 時 05 分 12.71 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 5.07 秒となる．

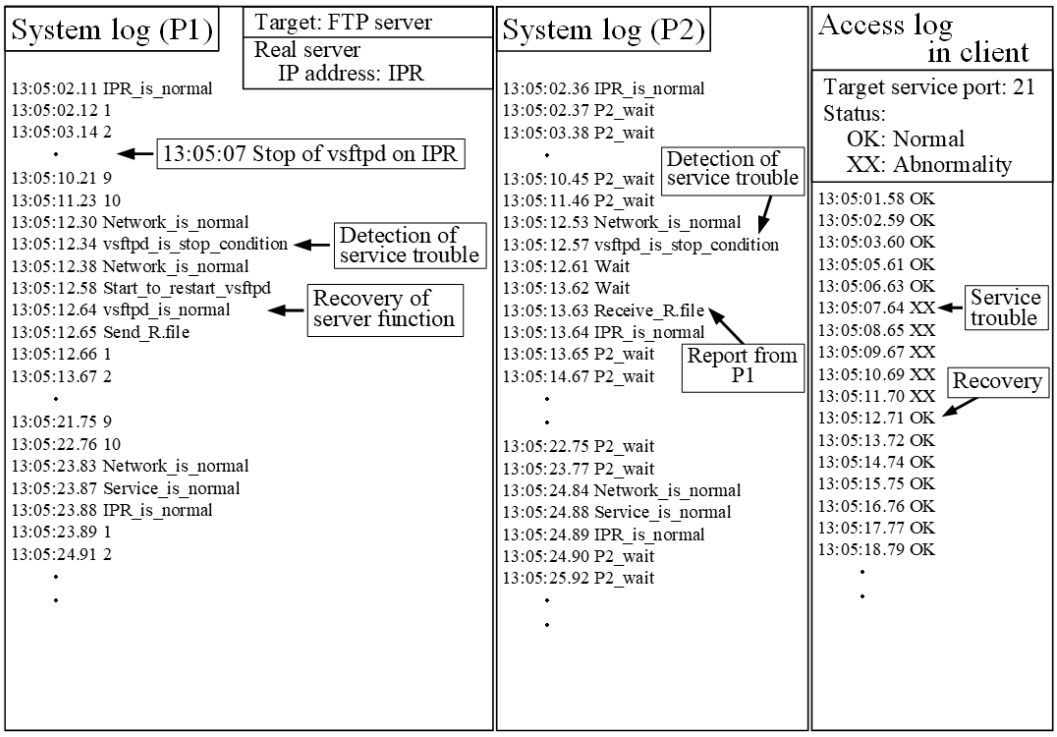


図 F6.1　サービス提供プログラムの再起動による復旧時の Log

図 F6.2 にサービス提供プログラムの再起動でサービス機能が復旧せず，管理対象サーバの再起動により復旧する場合を示す．System log(P1)では，監視間隔時間内の 13 時 50 分 36 秒に管理対象サーバ上のサービス提供プログラム vsftpd を停止し，13 時 50 分 41.49 秒に管理対象サーバのネットワークを確認後，サービス異常を検出している．管理対象サーバのネットワークを再確認後，vsftpd を再起動し，その後，サービス提供状態の再確認を行い，13 時 50 分 44.65 秒に復旧困難であることを確認している．管理対象サーバの仮想 IP アドレスを解除後，サーバ機能を復旧するために仮想サーバの起動処理を行う．13 時 50 分 49.57 秒に起動が完了し，仮想 IP アドレスの設定を行い，11 時 50 分 49.69 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“13:50:49.69”とサービスの異常を検出した時刻“13:50:41.53”との差がサーバ機能の復旧時間を示し，ここでは 8.16 秒となる．クライアントでの Access log では，13 時 50 分 36.29 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し, 13 時 50 分 50.45 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 14.16 秒となる．また，System log(P1)においてバックグラウンドジョブの稼働状況では，13 時 50 分 44.85 秒に管理対象サーバの再起動処理を開始し，13 時 51 分 43.36 秒に管理対象サーバの再起動が完了している．サービス提供状態の再確認を行い，13 時 51 分 43.42 秒にサービス機能の復旧が確認されている．その後，図 5.7(b)で示した up.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて up.file を検出すると，管理対象サーバが正常に動作したと判断し，仮想サーバの仮想 IP アドレスを解除する．その後，管理対象サーバに仮想 IP アドレスを設定し，13 時 51 分 44.66 秒に管理対象サーバが復旧している．ここで，管理対象サーバが復旧した時刻“13:51:44.66”とサービスの異常を検出した時刻“13:50:41.53”との差が管理対象サーバの復旧時間を示し，ここでは 63.13 秒となる．その管理対象サーバの復旧処理を行った 13 時 51 分 44.66 秒に

おけるクライアントでの Access log では受けているサービスに異常は見られない．System log(P2)では, P2 の管理サーバが管理対象サーバのサービス異常を 13 時 50 分 41.54 秒に検出後， P1 の結果を待つ状態となる．P2 の管理サーバは 13 時 51 分 45.28 秒に U.file を受信後，管理対象サーバの監視状態に戻る．

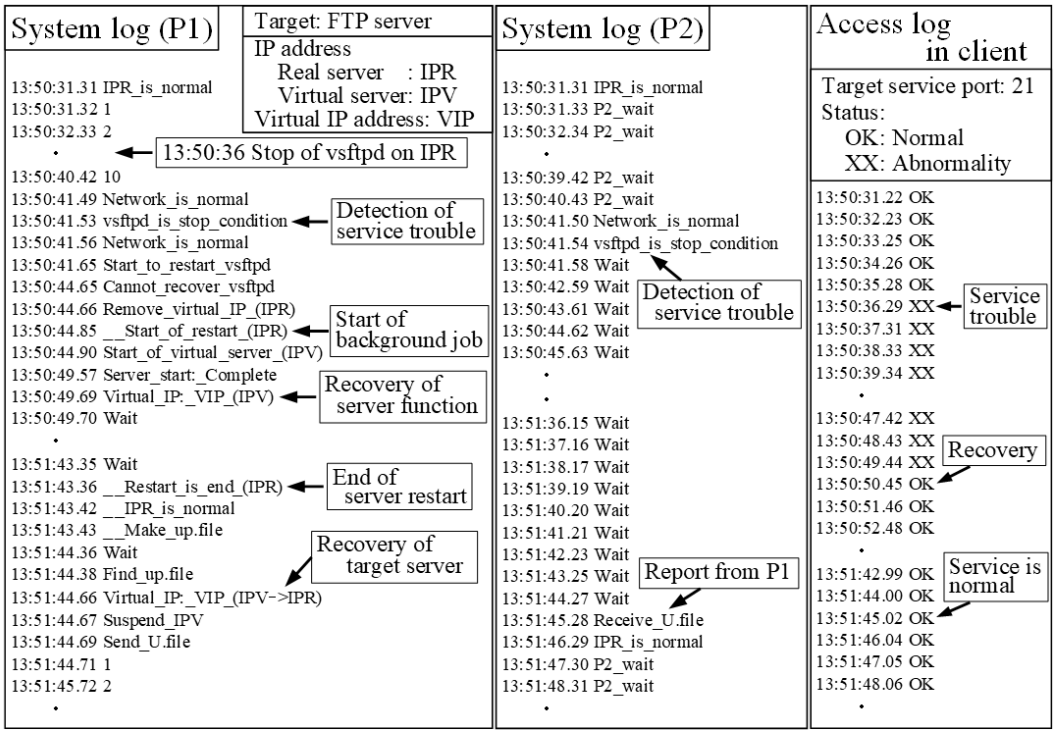


図 F6.2　管理対象サーバの再起動による復旧時の Log

図 F6.3 に管理対象サーバを再起動することによりネットワークトラブルを再現する場合を示す．System log(P1)では，監視間隔時間内の 13 時 31 分 20 秒に管理対象サーバを再起動し，13 時 31 分 27.03 秒に管理対象サーバのネットワーク異常を検出している．管理対象サーバのネットワークを再度確認し, 13 時 31 分 29.05 秒に異常であることを再確認している．その後，サーバ機能を復旧するために仮想サーバの起動処理を行い，13 時 31 分 33.97

秒に起動が完了し，仮想 IP アドレスの設定を行い，13 時 31 分 34.09 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“13:31:34.09”とネットワークの異常を検出した時刻“13:31:27.03”との差がサーバ機能の復旧時間を示し，ここでは 7.06 秒となる．クライアントでの Access log では，13 時 31 分 20.15 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し, 13 時 31 分 34.32 秒にサービスを受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 14.17 秒となる．また，System log(P1)においてバックグラウンドジョブの稼働状況では，13 時 31 分 29.06 秒に管理対象サーバが再起動状態であるかの確認を開始し，13 時 32 分 18.66 秒に管理対象サーバにおける再起動の完了が確認されている．サービス提供状態の確認を行い，13 時 32 分 18.72 秒にサービス機能の復旧が確認され，図 5.7(b)で示した up.file を作成後にバックグラウンドジョブは停止する．フォアグラウンドジョブにおいて up.file を検出すると，管理対象サーバが正常に動作したと判断され，仮想サーバの仮想 IP アドレスを解除する. その後, 管理対象サーバに仮想 IP アドレスを設定し, 13 時 32 分 19.89 秒に管理対象サーバが復旧している. ここで, 管理対象サーバが復旧した時刻“13:32:19.89”とネットワーク異常を検出した時刻“13:31:27.03”との差が管理対象サーバの復旧時間を示し，ここでは 52.86 秒となる．その管理対象サーバの復旧処理を行った 13 時 32 分 19.89 秒におけるクライアントでの Access log では受けているサービスに異常は見られない．System log(P2)では, P2 の管理サーバが管理対象サーバのサービス異常を 13 時 31 分 27.35 秒に検出後，P1 の結果を待つ状態となる. P2 の管理サーバは 13 時 32 分 20.53 秒に U.file を受信後，管理対象サーバの監視状態に戻る．

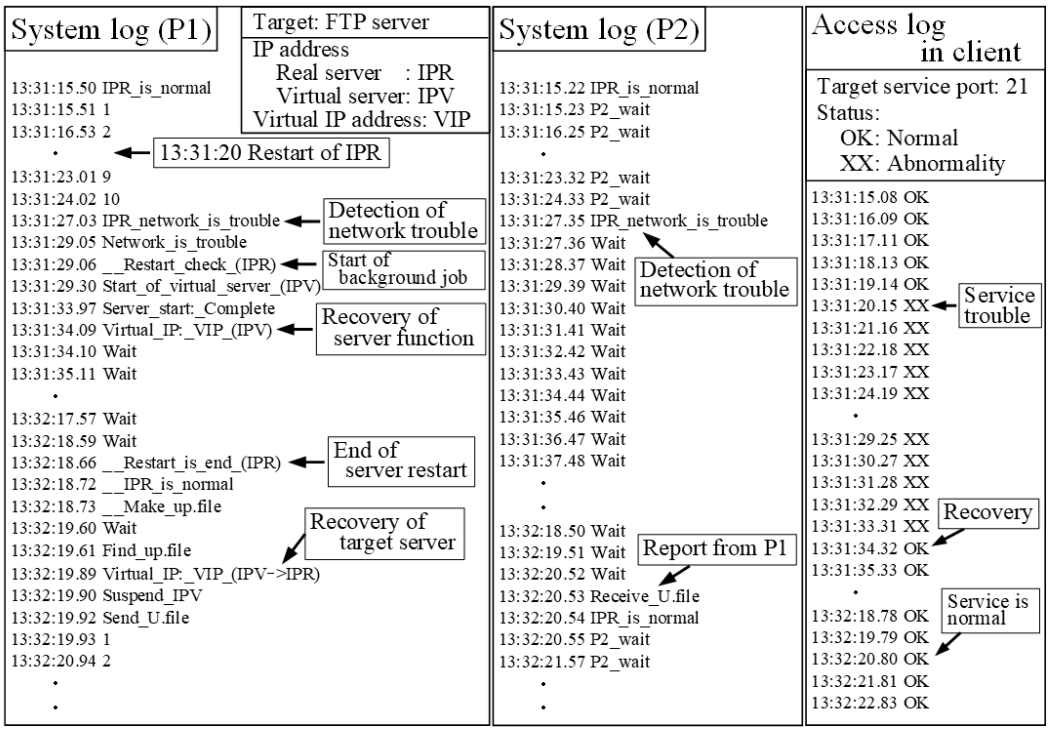


図 F6.3　管理対象サーバの再起動時の Log

図 F6.4 に管理対象サーバをシャットダウンすることによりネットワークトラブルを再現する場合を示す．System log(P1)では，監視間隔時間内の 12 時 53 分 45 秒に管理対象サーバをシャットダウンし, 12 時 53 分 52.13 秒に管理対象サーバのネットワーク異常を検出している．管理対象サーバのネットワークを再度確認し，12 時 53 分 54.18 秒に異常であることを再確認している．その後，サーバ機能を復旧するために仮想サーバの起動処理を行い，12 時 53 分 59.10 秒に起動が完了し，仮想 IP アドレスの設定を行い，12 時 53 分 59.22 秒にサーバ機能が復旧されている．ここで，サーバ機能が復旧した時刻“12:53:59.22”とネットワーク異常を検出した時刻“12:53:52.13”との差がサーバ機能の復旧時間を示し，ここでは 7.09 秒となる．クライアントでの Access log では，12 時 53 分 45.41 秒にサービスを受けるためにアクセスしていたサーバの異常を検出し, 12 時 53 分 59.56 秒にサービスを

受けることが可能となっている．この差がクライアントにおけるサーバアクセスの切断時間を示し，ここでは 14.15 秒となる．また，System log(P1)において，バックグラウンドジョブの稼働状況では, 12 時 53 分 54.19 秒に管理対象サーバが再起動状態であるかの確認を開始し，12 時 55 分 22.19 秒に管理対象サーバが停止状態であることを確認している．その後，WOL を使用して管理対象サーバを強制的に起動させ，12 時 56 分 18.87 秒に起動の完了を確認後，サービス提供状態の確認を行い，12 時 56 分 18.93 秒に管理対象サーバの状態が正常であることを確認している．図 5.7(b)で示した up.file を作成し，バックグラウンドジョブは停止する．フォアグラウンドジョブにおいて up.file を検出すると，管理対象サーバが正常に動作したと判断され仮想サーバの仮想 IP アドレスを解除する．その後，管理対象サーバに仮想 IP アドレスを設定し，12 時 56 分 19.96 秒に管理対象サーバが復旧している．

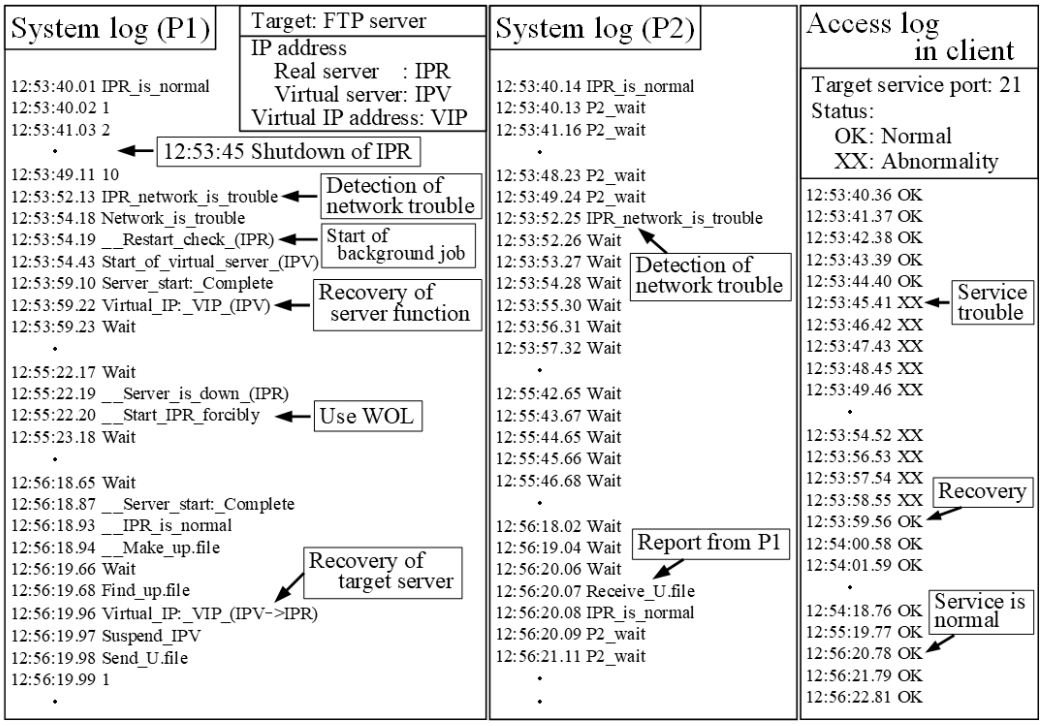


図 F6.4　管理対象サーバのシャットダウン時の Log

ここで，管理対象サーバが復旧した時刻“12:56:19.96”とネットワーク異常を検出した時刻“12:53:52.13”との差が管理対象サーバの復旧時間を示し，ここでは 147.83 秒となる．その管理対象サーバの復旧処理を行った 12 時 56 分 19.96 秒におけるクライアントでの Access log では受けているサービスに異常は見られない．System log(P2)では，P2 の管理サーバが管理対象サーバのネットワーク異常を 12 時 53 分 52.25 秒に検出後， P1 の結果を待つ状態となる．P2 の管理サーバは 12 時 56 分 20.07 秒に U.file を受信後，管理対象サーバの監視状態に戻る．

F7. NFS スレッド数の設定

（参考：https://docs.oracle.com/cd/E19620-01/806-3067/6jc88ej6o/index.html, Jul. 2018.）

NFS サーバを設定する場合には, NFS スレッドを設定する必要がある. スレッド 1 つは, NFS 要求を 1 つ処理することが可能で，スレッドプールを大きくすることにより，サーバは複数の NFS 要求を並行して処理することができる．プロセッサ数とネットワーク容量に従い，NFS サーバのスレッド数を決定することが望ましい．

NFS スレッドの数を設定するには，以下の 3 種類の方法のうち最大の値を使用する．

·CPU 1 個に対して NFS スレッド数を 16～32 に設定

·アクティブなクライアントプロセス 1 つに対して NFS スレッド数を 2 個に設定

·ネットワーク容量 10Mbit に対して NFS スレッド数を 16 個の割合で設定