IBM System p Advanced POWER Virtualization ベスト・プラクティス

Advanced POWER Virtualization 機能を効率的に使用するために作成された推奨プラクティス集

既存の IBM System p 資料に記載されている知識を基に構成された追加資料

経験豊富なシステム管理者やアーキテクトのための価値ある参考文献

Bill Adra
Tae-Hun Lee
Chris Milsted
Lars G. Hopner Petersen
Stanley Wood

ibm.com/redbooks

Redpaper

International Technical Support Organization

IBM System p Advanced POWER Virtualization ベスト・プラクティス

2006 年 10 月

注 : 本書および本書で紹介する製品をご使用になる前に、『特記事項』（ix ページ）に記載されている情報をお読みください。

本書は、以下の製品に適用されます。

- IBM 仮想 I/O サーバーのバージョン 1、リリース 3（製品番号 5765-G34）。
- IBM AIX 5L for POWER のバージョン 5、リリース 3、テクノロジー・レベル 5（製品番号 5765-G03）
- POWER5 システム・ファームウェアのバージョン 240、リリース 219
- ハードウェア管理コンソールのバージョン 5、リリース 2、モディフィケーション 1、フィックス MH00688 および MH00695

IBM 発行のマニュアルに関する情報のページ

http://www.ibm.com/jp/manuals/main/mail.html

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。(URL は、変更になる場合があります)
お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原　典：REDP-4194-00
IBM System p
Advanced POWER Virtualization
Best Practices

発　行：日本アイ・ビー・エム株式会社

担　当：ナショナル・ランゲージ・サポート
テクニカル部門による翻訳監修：中野 淳、黒川 佳昭、延 寛隆

第 1 刷 2007.6

目次

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。
〒 106-8711 東京都港区六本木 3-2-12 IBM World Trade Asia Corporation Intellectual Property Law Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾 :

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

商標

以下は、IBM Corporation の商標です。

AIX 5L™	ibm.com®	Redbooks（ロゴ）™
AIX®	IBM®	Redbooks™
BladeCenter®	POWER Hypervisor™	System i™
DS4000™	POWER5+™	System p5™
DS6000™	POWER5™	System p™
DS8000™	POWER™	System Storage™
eServer™	pSeries®	Tivoli®
HACMP™	PTX®	TotalStorage®

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

前書き

この IBM® Redpaper には、IBM System p5 サーバーで Advanced POWER™ Virtualization 機能を使用するときに活用できる各種機能の計画、インストール、保守、操作に関するベスト・プラクティスが記載されています。

Advanced POWER Virtualization 機能は、IBM POWER5™ プロセッサーと POWER5+™ プロセッサーを基盤とするシステムで仮想 I/O 環境をサポートしたり管理したりするために使用するハードウェアとソフトウェアの組み合わせです。

本書では、特に以下の内容を重点的に取り上げます。

- 一般的なベスト・プラクティス
- 管理とバックアップとリカバリー
- パフォーマンスと計画
- ストレージとネットワーキング

この Redpaper は、最初から最後まで読み通すこともできますが、本来の意図は、関連のあるトピックにアクセスするためのノートブックとして活用していただくことです。本書は、*「Advanced POWER Virtualization on IBM System p5」*（SG88-4018）の内容をさらに拡充し、大小さまざまなインストール済み環境を運用するクライアント・サイトやアウトソーシング・サイトで勤務する厳選チームによって用意された追加のサンプルやシナリオを記載しています。本書に載せているのは、実際の経験を基盤とした厳選されたベスト・プラクティスです。

Advanced POWER Virtualization 機能、論理パーティショニング、および IBM AIX® 5L™ に関する実践レベルの理解と、ネットワークや VLAN タグに関する基本的な理解が必要です。

この Redpaper の執筆チーム

この Redpaper は、世界各地のスペシャリストからなるチームが International Technical Support Organization の Austin Center で共同作業し執筆されました。

Bill Adra は、オーストラリアのシドニーにある IBM Systems and Technology Group で勤務する IT スペシャリストです。IBM System p™ と IBM TotalStorage® のソリューションに関して 7 年余りの経験があります。専門は、System p Advanced POWER Virtualization の機能です。オーストラリアとニュージーランドにある IBM のビジネス・パートナーとクライアントに販売前の技術サポートと販売後の実装支援を提供しています。pSeries® ソリューション販売の認定スペシャリストであり、System p AIX 5L の IBM 認定システム管理者でもあります。

Tae-Hun Lee は、System p の分野で 6 年、Manufacturing Execute System（MES）の分野で 3 年の経験があります。IBM では、これまで 6 年間勤務してきました。IBM General Parallel File System、System p サーバー上の Oracle、System p サーバー上の仮想化などが専門分野です。

Chris Milsted は、英国出身の IT スペシャリストです。IBM では、これまで 5 年間勤務してきました。そのうちの 4 年間は、AIX 5L と High-Availability Cluster Multi-Processing（HACMP™）を手がけ、現在は Advanced POWER Virtualization を手がけています。University of Nottingham で化学の学位を取得しました。System p テクノロジー、Linux®、AIX 5L、HACMP などが専門分野です。

Lars G. Hopner Petersen は、デンマーク出身の IT アーキテクトです。AIX と AIX 5L の分野で 15 年の経験があります。Engineering College of Copenhagen でエレクトロニクスとコンピューター・エンジニアリングの学士号を取得しました。仮想化、クラスター化、UNIX® などが専門分野です。

Stanley Wood は、IBM CIO のオフィスで勤務する上級 IT アーキテクトであり、分散システムや分散アプリケーションの設計と展開に関して 11 年の経験があります。IBM インフラストラクチャー内部の仮想化とストレージの戦略を担当するエンタープライズ・アーキテクトです。Texas A&M University 卒であり、コンピューター・サイエンスの理学士の学位を取得しています。サーバーとストレージの仮想化、ネットワーク・ファイル・システム（NFS）、IBM General Parallel File System、UNIX システム・パフォーマンスなどが専門分野です。

この資料の作成プロジェクトの管理者は、以下のとおりです。

Scott Vetter、PMP
IBM Austin

このプロジェクトに貢献した以下の方々に謝意を表します。

John Banchy、Ron Barker、Indulis Bernsteins、Lupe Brown、Dave Buffaline、Erin C. Burke、Bradford Cobb、Ranadip Das、Azur Gucer、Anil K、Steven Knudson、Bob Kovacs、Kiet H. Lam、David Larson、Timothy Marchini、Bob Minns、Jonathan Van Niewaal、Jorge R. Nogueras、Jim Pafumi、Jim Partridge、Viraf Patel、Amartey S. Pearson、Thomas Prokop、Pacelli Pujals、Vani D. Ramagiri、Rakesh Sharma、Limei Shaw、Jaya Srikrishnan、Tony Steel、John Tesch、Kate Tinklenberg、Vasu Vallabhaneni、Venkat Venkatsubra
IBM U.S.、IBM India

Andrew Goodall、Philip Dotchon、Andy Slim、Nigel Griffiths、Dai Williams
IBM U.K.

1

概要

「Advanced POWER Virtualization on IBM System p5」（SG88-4018）の資料では、IBM System p5™ プラットフォームでの仮想化の概要が説明されています。本書では、その基盤の上に構築する形でベスト・プラクティスを取り上げます。Advanced POWER Virtualization 機能は、System p プラットフォームだけでなく、IBM BladeCenter® システムや IBM System i™ システムでも活用できるので、本書のトピックの中には、それらのプラットフォームにも適用されるものが多く含まれています。

本書の資料を活用する前に、概要が説明されている資料の内容をよく理解し、仮想化に関する実践経験をいくらか積んでおくことをお勧めします。

Advanced POWER Virtualization は、実動環境でテストされています。

IBM の Advanced POWER Virtualization 機能は、エンタープライズ規模の実動ワークロードの仮想化に関するあらゆる要件に対応できるように設計されています。仮想 I/O の開発とテストのプロセスは、物理 I/O の場合と同じほど厳密です。この機能は、実動環境、開発環境、テスト環境を含め、通常は別個のサーバーや専用パーティションを使用するようなさまざまな環境に配置することができます。

本書では、Advanced POWER Virtualization 機能のアーキテクチャー、構成、文書に関する推奨事項を示します。複数の構成オプションが存在する分野で本書が提示する推奨事項は、あくまでも執筆者たちの環境と経験に基づく推奨事項です。

本書は、最初から最後まで読み通すこともできますが、興味のある個々のトピックを選択して、そのトピックに直接ジャンプできるように構成されています。主に 5 つの見出しに編成されています。

- この章、つまり第 1 章、『概要』（1 ページ）では、概要を示すトピックとベスト・プラクティスを取り上げます。
- 第 2 章、『管理、バックアップ、および リストア』（17 ページ）では、仮想 I/O サーバーの管理、バックアップ、およびリカバリーについて説明します。
- 第 3 章、『ネットワーキング』（53 ページ）では、仮想環境内のネットワークのアーキテクチャーと構成について説明します。
- 第 4 章、『ストレージ』（95 ページ）では、ストレージのアーキテクチャーと構成について説明します。
- 第 5 章、『パフォーマンスと計画』（129 ページ）では、パフォーマンス、CPU 管理、およびモニターについて説明します。

1.1 仮想 I/O サーバーの機能強化と更新

このセクションでは、仮想 I/O サーバー（VIOS）と使用可能なサービス更新に関する詳細情報を確認する方法を示します。

VIOS の詳しい使用方法については、以下の Web ページで追加情報を確認してください。

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphb1/iphb1kickoff.htm

1.1.1 仮想化に関する機能強化の最新情報

サービスと新機能について最新の情報を入手してください。

仮想化の新しい機能の開発プロセスは、現在も進行しています。したがって、新しい機能や他の機能の詳細については、以下の Web サイトを参照するのが最善です。

http://techsupport.services.ibm.com/server/vios/documentation/home.html

このセクションでは、本書の執筆時点で検討に値する機能を簡単に取り上げます。

VIOS Version 1.2 には、システム管理と可用性のための新しい機能がいくつか用意されています。

- Integrated Virtualization Manager（IVM）を使用すれば、ハードウェア管理コンソール（HMC）がなくても、Web ブラウザーによって仮想 I/O サーバーと仮想 I/O クライアントを管理することが可能になります。IVM は、HMC の代わりに IBM System p5 プラットフォーム（特にローエンド・システム）で使用できます。IVM は、IBM System p5 のモデル 570、575、590、595 では使用できません。
- 仮想光メディアは、仮想 I/O サーバーと仮想 I/O クライアントの間の仮想 SCSI によってサポートできます。仮想光メディアのサポートは、最初に VIOS Version 1.2 で導入されました。CD-ROM、DVD-RAM、DVD-ROM によって、仮想光ディスク装置をサポートできます。
- 旧バージョンの場合、二重仮想 I/O サーバーによるネットワーク高可用性は、AIX 5L の Network Interface Backup（NIB）機能によってのみ構成できました。このバージョンでは、共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）によって、仮想 I/O サーバー間のネットワーク・フェイルオーバーを構成できるようになっています。
- ストレージ・プールも新しい機能です。ボリューム・グループとよく似ていますが、ストレージ・プールを使用すれば、初心者のユーザーでもデバイス管理を容易に行えます。

VIOS Version 1.3 には、以下のような追加機能が組み込まれています。

- モニター機能の改善（例えば、パフォーマンス・ツール **topas** と **viostat** が追加され、パフォーマンス PTX® エージェントが使用可能になりました）。（PTX は、別途購入可能なライセンス・プログラムです。）
- 仮想 SCSI と仮想イーサネットのパフォーマンスが改善され、コマンドが強化され、追加のストレージ・ソリューションが使用可能になりました。
- Integrated Virtualization Manager（IVM）によって、管理対象パーティションのメモリーとプロセッサーの動的論理パーティショニングのサポートという最先端機能がこのリリースに追加されました。さらに、仮想 I/O サーバーのネットワーク構成のためにブラウザー・ベースのインターフェースをサポートするなど、使いやすさを重視した数多くの機能強化が組み込まれています。
- IBM System Planning Tool（旧称 LVT）の機能が強化されています。

  仮想化の計画に関する要件に対応するために、System Planning Tool（SPT）を以下のサイトから無償でダウンロードできるようになっています。

  http://www.ibm.com/servers/eserver/support/tools/systemplanningtool/

System p と System i のパーティションの設計に System Planning Tool を使用できます。その結果として生成される System Plan 形式のファイルを ハードウェア管理コンソール（HMC）Version 5.2 にインポートすれば、新しいシステム・プラン機能を使用して、System Planning Tool によって設計したパーティションの構成を自動化できます。HMC のシステム・プラン機能を使用すれば、**mksysplan** コマンドによってシステム・プランを生成することも可能になります。

1.1.2 フィックスパックによる仮想 I/O サーバーの更新

仮想 I/O サーバー（VIOS）の既存のインストール済み環境に最新のフィックスパックを適用すれば、最新の VIOS レベルに更新できます。

フィックスパックには、VIOS の既存のインストール済み環境のマイグレーション・パスが用意されています。最新のフィックスパックを適用することによって、VIOS を最新レベルにアップグレードできます。すべての VIOS フィックスパックは累積パックであり、以前のフィックスパックの修正がすべて組み込まれています。VIOS のダウンロード・ページでは、各フィックスパックに組み込まれているすべての修正のリストが管理されています。

仮想 I/O サーバーを新しいバージョンに更新するためのサービスが用意されています。

フィックスパックは通常、VIOS のすべてのインストール済み環境に対応した汎用の更新です。HMC 管理の仮想 I/O サーバーにも IVM 管理の仮想 I/O サーバーにもフィックスパックを適用できます。フィックスパックを適用する前に、VIOS に適用されているすべての暫定修正を手動で除去する必要があります。VIOS に暫定修正を適用した仮想 I/O クライアントでは、フィックスパックを適用する前に、VIOS の Web サイトで説明されている手順を実行して暫定修正を除去してください。

図 1-1 は、フィックスパックによって VIOS の Version 1.1 から Version 1.2.1.x 以降にマイグレーションする方法をまとめたものです。

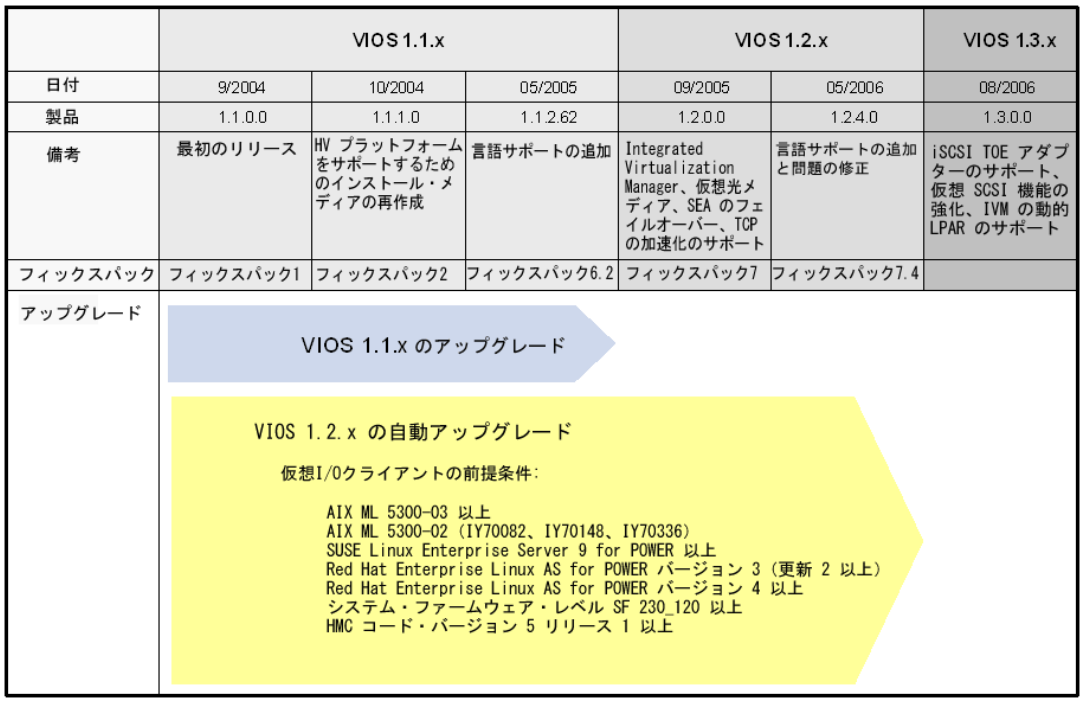


図 1-1　フィックスパックによる VIOS のアップグレードとマイグレーション

VIOS の最新リリースとインストール手順を確認するには、以下のサイトを参照してください。

http://www14.software.ibm.com/webapp/set2/sas/f/vios/home.html

ヒント：すべての VIOS フィックスパックは累積パックであり、以前のフィックスパックの修正がすべて組み込まれています。最新の VIOS フィックスパックを適用すれば、既存の VIOS が最新のサポート・レベルにアップグレードされます。

1.2 仮想 I/O サーバーの操作

実動環境に移す前に、テスト環境でワークロードを確認します。

システムのサイズ見積もりでは、インフラストラクチャーの設計上の限界の範囲内で環境に関する計画を策定するのがベスト・プラクティスといえます。その点では、引用されている理論上の最大値を実動システムの設計の基盤として使用すると、キャパシティーと期待するパフォーマンスの間でトレードオフの関係が生じる可能性があります。どんな場合でも、実動システムで実装する前に、テスト構成で最も重要なワークロードを確認するのがベスト・プラクティスです。

1.2.1 仮想イーサネット

各仮想 I/O サーバーごとに、最大で 256 個の仮想イーサネット・アダプターを定義できます。1 つの仮想イーサネット・アダプターに接続できる仮想 I/O クライアントの数は、使用可能メモリーによって制限される設計上の値になります。クライアント接続の最大数に達する前に、構成をテストすることをお勧めします。仮想イーサネットのアーキテクチャー上の制限を以下のリストにまとめます。

- 各仮想イーサネット・アダプターは、最大で 21 個の仮想ローカル・エリア・ネットワーク（VLAN）、つまり 20 個の VID と 1 個の PVID に関連付けることができます。
- 1 つのシステムで最大 4096 個の VLAN をサポートできます（この数は、IEEE 802.1Q 標準で定義されています）。
- 各共用イーサネット・アダプター（SEA）には、最大で 16 個の仮想イーサネット・アダプターを定義でき、それぞれに最大で 21 個の VLAN（20 個の VID と 1 個の PVID）を関連付けることができます。SEA の場合は、これらの VLAN で 1 つの物理ネットワーク・アダプターを共用できます。

1.2.2 仮想 SCSI

各仮想 I/O サーバーごとに、最大で 256 個の仮想 SCSI アダプターを定義できます。

デバイスの数に応じてキュー・エントリー数を分割します。

AIX 5L Version 5.3（PTF IY81715 などの最新のサービスを適用したシステム）では、クライアントの SCSI デバイスごとに、1 個から 256 個のキュー・エントリー数を設定できます。この値によって、ディスク・コントローラーが 1 度に仮想 SCSI クライアント・ドライバーのキューに入れる要求の数が決まります。この値は、以下のコマンドによって変更できます。

```
chdev -l hdiskN -a'queue_depth= value'
```

hdisk*N* は物理ボリュームの名前、*value* は 1 から 256 の数値です。queue_depth 属性の現行値は、以下のコマンドによって表示できます。

```
lsattr -El hdiskN
```

hdisk*N* は、物理ボリュームの名前です。

キュー・エントリー数の詳しい説明については、4.8、『SCSI 用キューのエントリー数』（126 ページ）を参照してください。

1.3 仮想 I/O サーバーの可用性の概要

仮想 I/O サーバーは、Advanced POWER Virtualization ハードウェア機能の一部であり、この機能を使用することによって、仮想 SCSI や仮想ネットワーキングなど、論理パーティション同士の間で物理リソースを共用することが可能になります。

二重仮想 I/O サーバー構成によって、ソフトウェアの保守作業を同時に実行できるようになります。

HACMP や Veritas Cluster Server などのクラスター化ソフトウェアは、システム・ノードのフェイルオーバーをサポートするように設計されています。仮想化環境で高可用性パスやデバイスの高可用性構成を実装するために、仮想 I/O サーバーそのもので HACMP を実装する必要はありません。複数の仮想 I/O サーバーと冗長デバイスを用意すれば、ソフトウェアの保守とハードウェアの交換に関する計画を策定しやすくなります。複数の仮想 I/O サーバーの管理には、HMC を使用する必要があります。

HACMP の実装は、仮想 I/O クライアントで仮想 I/O リソースを使用することによってサポートできます。HACMP のインストールは、クライアント・オペレーティング・システムのコンポーネントです。

仮想 I/O サーバーの回復力を高めるための方法

仮想 I/O サーバーの可用性をさらに高めるために、以下の項目を組み合わせて使用します。

- 冗長物理ハードウェア
- Network Interface Backup のイーサネット構成
- 共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）構成
- ストレージの論理ボリューム・マネージャー（LVM）のミラーリングと RAID 構成
- ストレージ・エリア・ネットワーク（SAN）のマルチパス I/O（MPIO）
- RAID ストレージ（RAID は、ストレージ・サブシステムまたは RAID アダプターによって用意する）
- 効率的で詳細なインストール前の計画

保守性を高めるための別の方法は、仮想 I/O サーバーのために、組み込みの統合ネットワーク・アダプターではなくホット・プラグ可能ネットワーク・アダプターを使用することです。アップグレードや保守の計画の実行時には、PCI スロット・アダプターを交換するほうが容易です。

以下のセクションでは、仮想 I/O 環境の実装時の考慮事項に関連して使用できるさまざまなオプションを取り上げます。

1.3.1 単一仮想 I/O サーバー構成

仮想 I/O サーバーは、耐久力と回復力の高いセキュアな環境であり、要件が非常に過酷な実動ワークロードにも対応できるように設計されています。

仮想 I/O サーバーでは、ごく少数の信頼性の高い（サーバーのパッケージに組み込まれているテスト済みの）デバイス・ドライバーだけを実行します。クライアント・パーティションでは、ローカルの物理デバイス・ドライバーを使用する代わりに、仮想リソース・デバイス・ドライバーを使用して、仮想 I/O サーバーと通信します。物理出入力は仮想 I/O サーバーが実行します。仮想（仮想 I/O サーバー）デバイス・ドライバーと物理リソース・デバイス・ドライバーを除けば、仮想 I/O サーバーで実行されるプロセスはごくわずかしかないので、サーバー・サイドでソフトウェアの問題が発生するリスクは非常に小さいと言えます。

複数のオンライン・ソフトウェア更新を同時に実行する必要がある場合や、可能な構成の数を増やす場合は、定期保守ポリシーの一環として 2 つの仮想 I/O サーバーを使用することをお勧めします。そのようにすれば、リンク集約を使用する場合に、フェイルオーバー・ネッ

トワーク接続を別のスイッチに切り替えることも可能になります。これらの機能が不要であれば、クライアント構成には単一仮想 I/O サーバーが最適と言えます。

二重仮想 I/O サーバーの実装方法については、1.3.2、『二重仮想 I/O サーバー構成』（11 ページ）を参照してください。

この後のトピックでは、単一の仮想 I/O サーバーを使用して仮想化環境を実装する場合の推奨手順を取り上げます。

共用イーサネット・アダプター（Shared Ethernet Adapter）の可用性

リンク集約によって、ネットワークの可用性とスループットを改善できます。

仮想 I/O サーバーで構成する最初のリソースは、共用イーサネット・アダプター（Shared Ethernet Adapter）です。共用イーサネット・アダプター（Shared Ethernet Adapter）は、少なくとも 1 個の物理ポートを必要としますが、物理ネットワークに対する集約リンクとして複数の物理イーサネット・ポートを使用することも可能です。複数のネットワーク・アダプターを使用することによって、単一ネットワーク接続に追加の帯域幅を用意したり、冗長性を高めたりすることが可能になります。

集約リンクを実装するときには、すべての基本ポートを同じスイッチに接続し、集約のために使用する各スイッチ・ポートを 802.3ad リンク集約または Cisco EtherChannel に対応するように構成する必要があります。共用イーサネット・アダプター（Shared Ethernet Adapter）を作成する前に、リンク集約のスイッチ設定を作成しなければなりません。

リンク集約の主な利点は、すべてのアダプターのネットワーク帯域幅が 1 つのリンクのように取り扱われることです。1 つのアダプターが使用不能になっても、次に使用可能なアダプターにパケットが自動的に送信されるので、既存の接続が中断されることはありません。ただし、リンク集約は、完全な高可用性ネットワーキング・ソリューションではありません。すべての集約リンクを同じスイッチに接続しなければならないからです。この要件は、バックアップ・アダプターの使用によって対応できます。つまり、同じ VLAN で別のイーサネット・スイッチに接続しているリンク集約にもう 1 つのリンクを追加できます。そのリンクは、バックアップ専用にします。図 1-2 は、1 つの仮想 I/O サーバーと 1 つの集約リンクを使用して可用性を高めるための、推奨共用イーサネット・アダプター（Shared Ethernet Adapter）構成を示したものです。

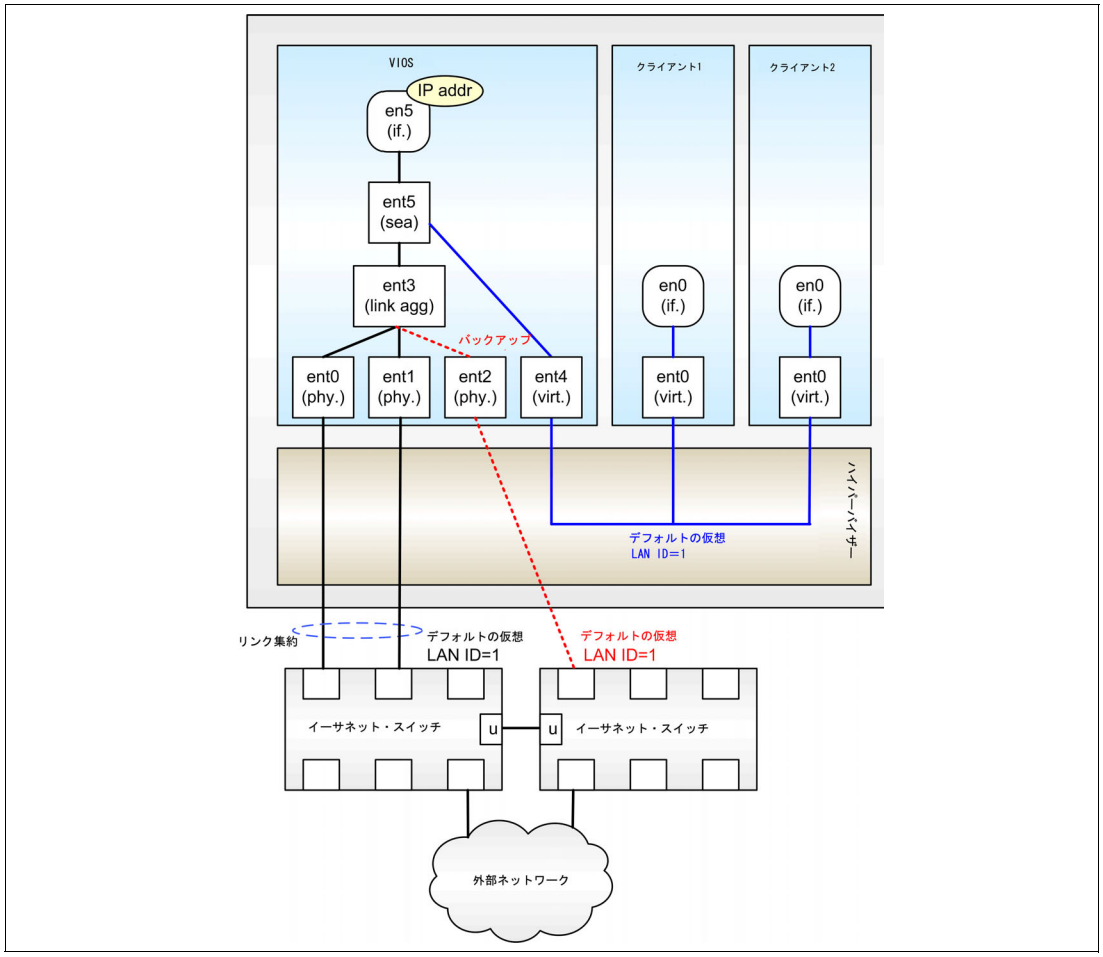


図 1-2　単一仮想 I/O サーバーの推奨ネットワーク構成

仮想 I/O サーバーのブート・ディスクの冗長性

冗長ブート・ディスクによって、ディスク交換時にも可用性を実現できます。

仮想 I/O サーバー・ソフトウェアは、論理パーティションに存在します。したがって、仮想 I/O サーバーには、独自のリソース（プロセッサー、メモリー、専用ディスクなど）が必要です。仮想 I/O サーバー・ソフトウェアは、仮想 I/O サーバーのルート・ボリューム・グループ（rootvg）である専用ディスクにインストールされます。ローカル接続の SCSI ディスクから仮想 I/O サーバーをブートする場合は、ルート・ボリューム・グループ（例えば hdisk0）が 2 番目の SCSI ディスク（hdisk1）とそのディスクにミラーリングされているデータを含むように拡張されていることが必要です。この操作を実行すると、仮想 I/O サーバーがリブートされます。したがって、この操作は、インストール時に実行するか、その仮想 I/O サーバーを経由してリソースを使用しているすべての仮想 I/O クライアントが定義済みの仮想リソースを必要としないタイミングで実行することをお勧めします。使用するコマンドは、以下のとおりです。

```
$ extendvg –f rootvg hdisk1
$ mirrorios hdisk1
This command causes a reboot. Continue [y|n]?
```

ミラーリングが完了したら、仮想 I/O サーバーの rootvg 以外の物理ストレージをクライアント・パーティションと起動済みのクライアントにエクスポートできます。

ローカル接続の SCSI ディスクからブートする方法に加えて、仮想 I/O サーバーを SAN ストレージに接続することも可能です。割り当てられている SAN ディスクに仮想 I/O サーバーのルート・ボリューム・グループをインストールできます。ストレージ・サブシステムが

ブート時のマルチパス機能をサポートしていて、追加のパスを用意できる場合は、SAN ストレージから仮想 I/O サーバーをブートすることをお勧めします。SAN からのブートに関する考慮事項については、ストレージのベンダーに問い合わせてください。

ユーザー・データとシステム・データを混在させないのが、一般的な習慣です。したがって、仮想 I/O サーバーのブート・ディスクと同じディスクにクライアントのブート・ディスクを配置しないようにすることをお勧めします。

仮想 I/O クライアントのための SAN ストレージの構成

仮想 I/O サーバーでは、SAN を基盤としたストレージ・デバイスを仮想 I/O クライアントのために使用できます。図 1-3 は、単一仮想 I/O サーバーを経由して SAN を基盤としたバッキング・デバイスを使用する仮想 I/O クライアントの推奨構成を示したものです。

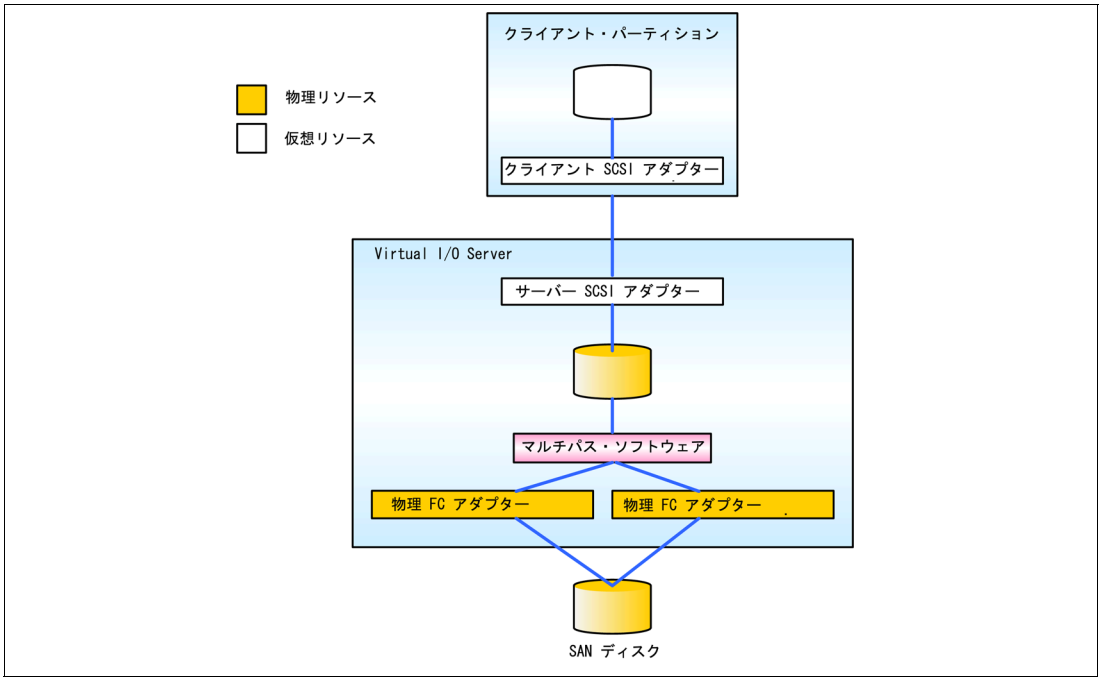


図 1-3 SAN ストレージに接続した単一仮想 I/O サーバー

MPIO（マルチパス機能）によって、重要なデータに対する複数のパスを確保できます。

図 1-3 は、SAN ストレージに接続した 2 つのファイバー・チャネル・アダプターを含んだ単一仮想 I/O サーバー構成です。ストレージ・ベンダーのマルチパス・ソフトウェアを仮想 I/O サーバーにインストールするには、**oem_setup_env** コマンドを使用します。マルチパス・ソフトウェアによって、1 つのアダプターが使用不能になった場合のロード・バランシングとフェイルオーバーの機能を用意できます。クライアント・パーティションに SAN LUN を提供する場合は、仮想 I/O サーバーでストレージ・プールやボリューム・グループを構成する代わりに、LUN 全体をパススルーすることをお勧めします。論理ボリュームやストレージ・プールは、二重仮想 I/O サーバー構成ではサポートされていないからです。仮想 I/O クライアントに必要なディスク・サイズも、SAN ストレージ・サブシステムを使用して構成する必要があります。

二重仮想 I/O サーバーと SAN Volume Controller（SVC）を使用することによって、二重 VIOS lvvscsi ディスク（例えば、分割されたストレージ・サブシステムから提供される大きな LUN）の効果を実現できます。

仮想 I/O クライアントのためのディスクのミラーリング

ディスクのミラーリングによって、クライアント・データを保護できます。

仮想 I/O サーバーでは、SCSI を基盤としたストレージ・デバイスを仮想 I/O クライアントのために使用できます。図 1-4 は、単一仮想 I/O サーバーを経由して SCSI を基盤としたバッキング・デバイスを使用するクライアント・パーティションの推奨構成を示したものです。

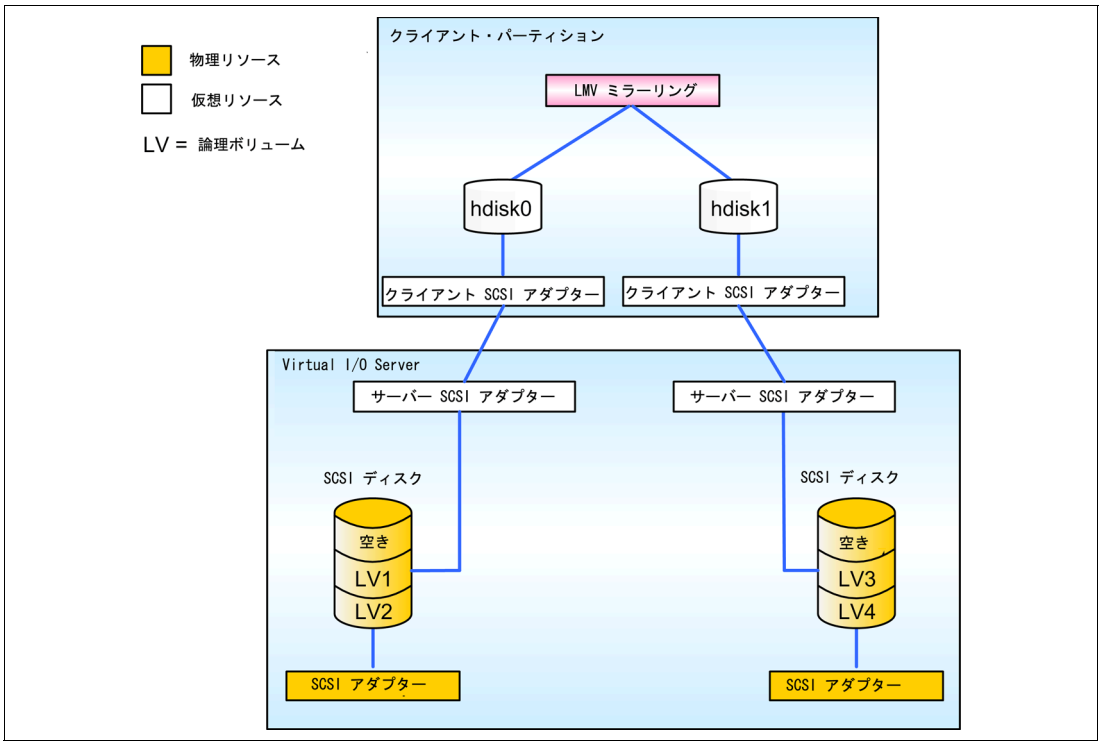


図 1-4 クライアント・パーティションで LVM のミラーリングを使用した単一仮想 I/O サーバー

図 1-4 は、2 つの SCSI アダプターを含んだ単一仮想 I/O サーバー構成であり、各 SCSI アダプターが SCSI ディスク・ドライブのセットに接続されています。SCSI ディスクを使用する仮想 I/O クライアントにディスクを提供する場合は、2 つの物理ディスクまたは 2 つの論理ボリューム（それぞれの論理ボリュームが別々のディスクに由来し、それぞれのディスクが別々のコントローラーに存在する構成）を仮想 I/O クライアント・パーティションに渡すことをお勧めします。仮想 I/O クライアント・パーティションは、クライアント・オペレーティング・システムの LVM のミラーリングによって仮想 I/O クライアントでミラーリングされている 2 つの仮想 SCSI ディスクを検出します。

重要 : 仮想ディスクとして使用する論理ボリュームのサイズは、最大で 1 TB になります。また、仮想 I/O サーバー上で仮想ディスクとして使用する論理ボリュームについては、ミラーリングもストライプ化もできませんし、不良ブロック再配置を有効にすることもできません。

仮想ディスクとしてクライアントにエクスポートする論理ボリュームは、複数の物理ボリュームにまたがっているべきではありません。さらにスペースが必要な場合は、第 2 の論理ボリュームを別個の仮想ディスクとして別のディスクにエクスポートしてください。

1.3.2 二重仮想 I/O サーバー構成

複数の仮想 I/O クライアント・パーティションが仮想 I/O サーバー経由でリソースを使用している状態では、追加のシステム・サービスや構成オプションを用意するために二重仮想 I/O サーバーを実装し、デバイスへのパスやデバイスそのものを二重化することができます。

二重仮想 I/O サーバー構成によって、保守性を高めることができます。

基本的に、二重仮想 I/O サーバー構成をお勧めする主な理由は、以下のとおりです。

- 将来的にハードウェアを拡張したり、新しい機能を追加したりするため
- 人的な操作による予想外の障害に対応するため
- 物理デバイスの故障や自然災害による予想外の障害に対応するため
- 仮想 I/O サーバーの定期保守のための停止に対応するため
- ネットワークとストレージのワークロードを切り離して、仮想 I/O クライアント・パーティションのパフォーマンスを向上させるため
- MPIO などの複数のマルチパス・コードやデバイスの冗長性を実現するため

この後のセクションでは、二重仮想 I/O サーバー構成を実装するときの一般的な推奨事項やベスト・プラクティスを取り上げます。

ネットワークの可用性の強化

二重仮想 I/O サーバー構成で仮想 I/O クライアント・パーティションのネットワークの冗長性を実現するために使用できる一般的な方法は、以下の 2 つです。

- Network Interface Backup（NIB）
- 共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）

この後のセクションでは、その両方の方法を取り上げます。セクション 3.8.2、『二重仮想 I/O サーバー構成の拡張可用性オプション』（80 ページ）では、Network Interface Backup（NIB）と共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）の両方のオプションを比較します。

Network Interface Backup

図 1-5（12 ページ）は、二重仮想 I/O サーバーを使用した高可用性ネットワーク構成です。この構成では、デフォルトの仮想 LAN ID で作成された仮想イーサネット・アダプターを使用しており、それぞれの仮想イーサネット・アダプターは、タグの外されたポートだけを使用する物理イーサネット・スイッチに接続しています。この例の場合、VIOS 1 には、仮想 LAN ID 2 を使用する仮想イーサネット・アダプターを経由してクライアント・パーティション 1 に対する外部接続を提供する共用イーサネット・アダプター（Shared Ethernet Adapter）があります。VIOS 2 にも、仮想 LAN ID 3 を使用する仮想イーサネット・アダプターを経由してそのクライアント・パーティションに対する外部接続を提供する共用イーサネット・アダプター（Shared Ethernet Adapter）があります。クライアント・パーティション 2 のセットアップもほとんど同じですが、仮想 LAN ID 2 を使用する仮想イーサネット・アダプターが基本になっており、仮想 LAN ID 3 がバックアップになっている点だけが異なっています。つまり、クライアント・パーティション 2 では、VIOS 1 によって基本接続を、VIOS 2 によってバックアップ接続をそれぞれ確保できるようになっています。

クライアント・パーティション 1 では、Network Interface Backup によって仮想イーサネット・アダプターが構成されており、仮想 LAN ID 3 ネットワークが基本、仮想 LAN ID 2 ネットワークがバックアップになっています。ハートビートには、デフォルトのゲートウェイの IP アドレスを使用します。つまり、クライアント・パーティション 1 では、VIOS 2 によって基本接続を、VIOS 1 によってバックアップ接続をそれぞれ確保できるようになっています。

アダプターのメインの仮想 I/O サーバーが使用不能になった場合は、ゲートウェイへのパスが切断されるので、Network Interface Backup メカニズムによってその状態が検出されます。Network Interface Backup のセットアップは、バックアップ仮想 I/O サーバーによって接続しているバックアップ・アダプターにフェイルオーバーします。

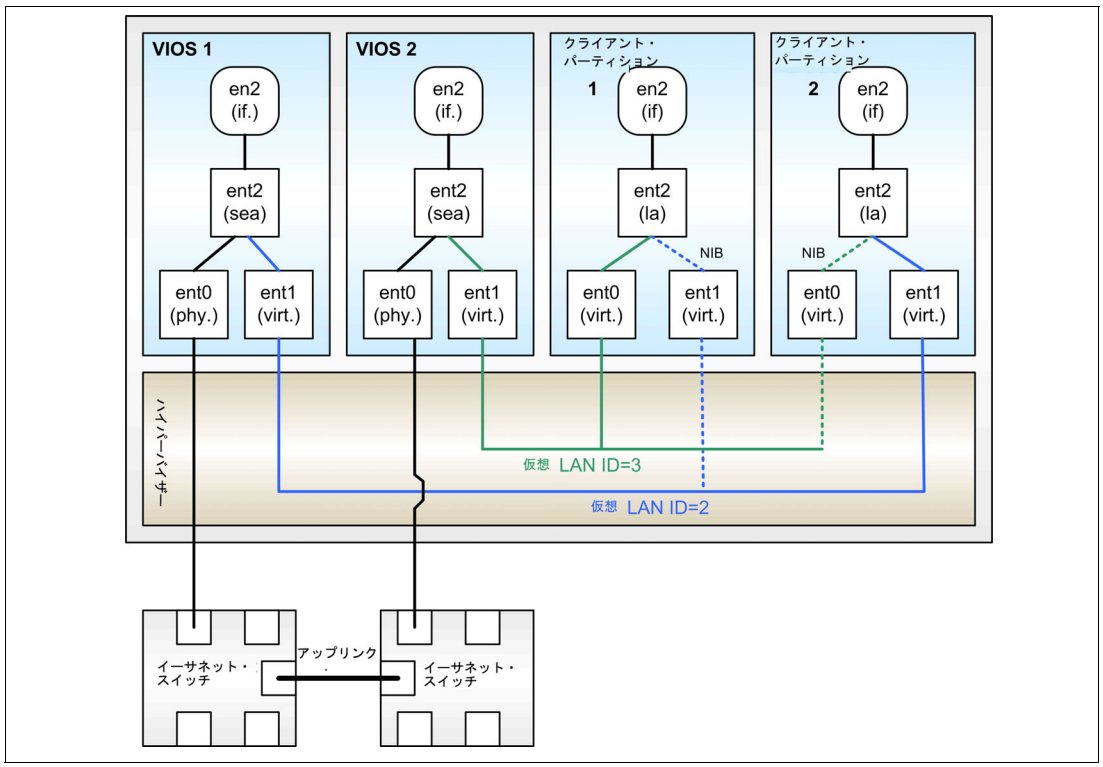


図 1-5　二重仮想 I/O サーバーを使用した Network Interface Backup

共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）

共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）を仮想 I/O サーバーに実装するには、外部ネットワークにアクセスするためのブリッジング（第 2 層）方式を採用します。SEA のフェイルオーバーは、Network Interface Backup とは異なり、IEEE 802.1Q の VLAN タグをサポートしています。

SEA のフェイルオーバーでは、2 つの仮想 I/O サーバーが共用イーサネット・アダプター（Shared Ethernet Adapter）のブリッジング機能を実行できるので、1 つの仮想 I/O サーバーが使用不能になったり、共用イーサネット・アダプター（Shared Ethernet Adapter）が物理イーサネット・アダプターによって外部ネットワークにアクセスできなくなったりした場合に、自動的にフェイルオーバーするようになっています。手動フェイルオーバーを起動することも可能です。

図 1-6（13 ページ）にあるとおり、両方の仮想 I/O サーバーが同じ仮想イーサネット・ネットワークと物理イーサネット・ネットワークと VLAN に接続し、両方の共用イーサネット・アダプター（Shared Ethernet Adapter）の両方の仮想イーサネット・アダプターで外部ネットワーク・アクセス・フラグが有効になっています。2 つの仮想 I/O サーバーの間の別個の VLAN として追加の仮想イーサネット接続をセットアップし、その VLAN をコントロール・チャネルとして共用イーサネット・アダプター（SEA）に接続しなければなりません。その VLAN は、ブリッジング機能のフェイルオーバーを制御する 2 つの仮想 I/O サーバーの間でキープアライブ・メッセージやハートビート・メッセージをやり取りするためのチャネルとしての機能を果たします。そのコントロール・チャネルのイーサネット・アダプターにネッ

トワーク・インターフェースを接続する必要はありません。そのコントロール・チャネル・アダプターは、他の目的のために使用されていない専用の VLAN に専用のアダプターとして配置するのが望ましいと言えます。

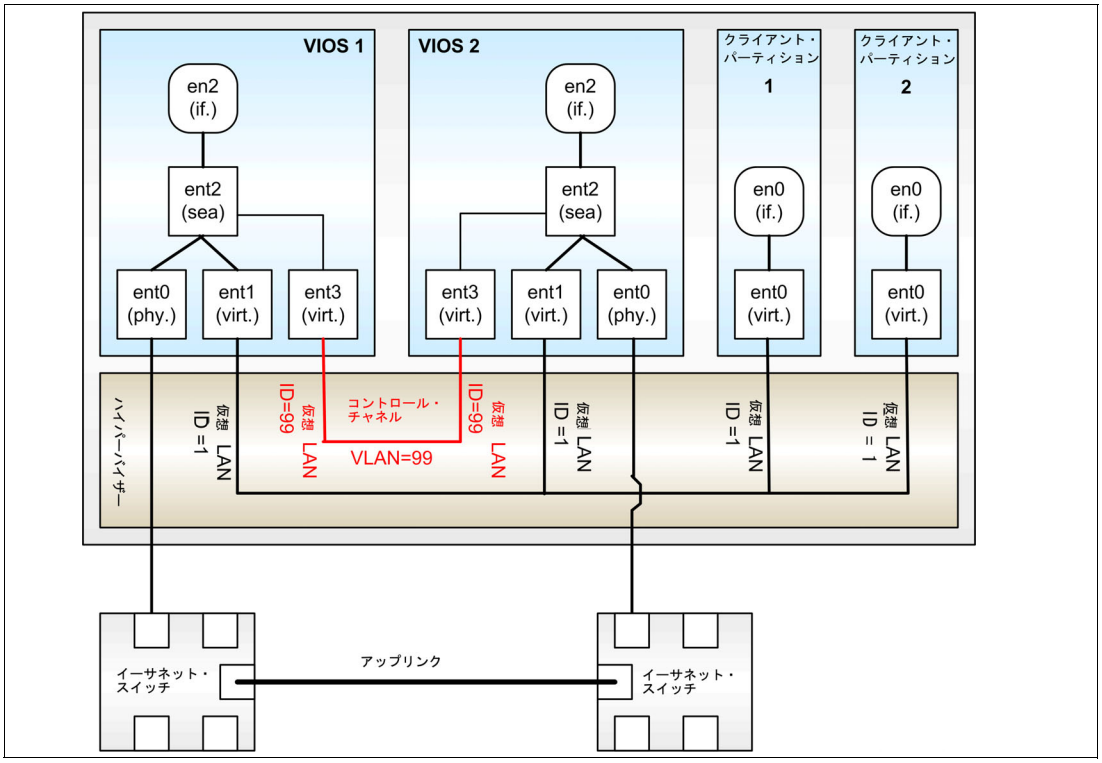


図 1-6　二重仮想 I/O サーバーを使用した SEA のフェイルオーバー

各仮想 I/O サーバーの共用イーサネット・アダプター（Shared Ethernet Adapter）は、別々の優先順位の値で構成する必要があります。優先順位の値によって、2 つの共用イーサネット・アダプター（Shared Ethernet Adapter）のうちのどちらが基本（アクティブ）になり、どちらがバックアップ（スタンバイ）になるかが決まります。優先順位が高いのは、優先順位の値の小さいほうです（例えば、優先順位 1 は最高の優先順位です）。

共用イーサネット・アダプター（Shared Ethernet Adapter）に IP アドレスを割り振り、そこから周期的に ping を実行して、ネットワーク接続が有効かどうかを確認することも可能です。この構成は、Network Interface Backup（NIB）で ping を実行するように構成した IP アドレスの場合とよく似ています。ただし、NIB を使用する場合は、SEA で 1 回だけ ping を実行する場合とは異なり、接続を確認するための ping をすべてのクライアントで実施するよう構成する必要があります。

詳しくは、3.9、『SEA のフェイルオーバーをサポートするための共用イーサネット・アダプター（Shared Ethernet Adapter）の作成』（82 ページ）を参照してください。

SEA のフェイルオーバーでは、ネットワークがトラフィックを転送するときに、最大で 15 個から 30 個のパケットがドロップされる可能性があります。

二重仮想 I/O サーバー構成のためのブート・ディスクの冗長性

『仮想 I/O サーバーのブート・ディスクの冗長性』（8 ページ）で説明されている手順で、仮想 I/O サーバーのルート・ボリューム・グループのミラーリングを実装します。

マルチパス機能

二重仮想 I/O サーバー構成で SAN 接続ストレージを使用して仮想 I/O クライアント・パーティションから物理ディスクにアクセスする場合は、仮想 I/O クライアント・パーティションからマルチパス機能を使用して、ディスクに対する冗長パスを用意できます。仮想 I/O クライアント・パーティションからマルチパス機能を使用することによって、可用性が非常に高い自動化可用性ソリューションを実現できます。図 1-7（14 ページ）は、その構成を詳しく示したものです。

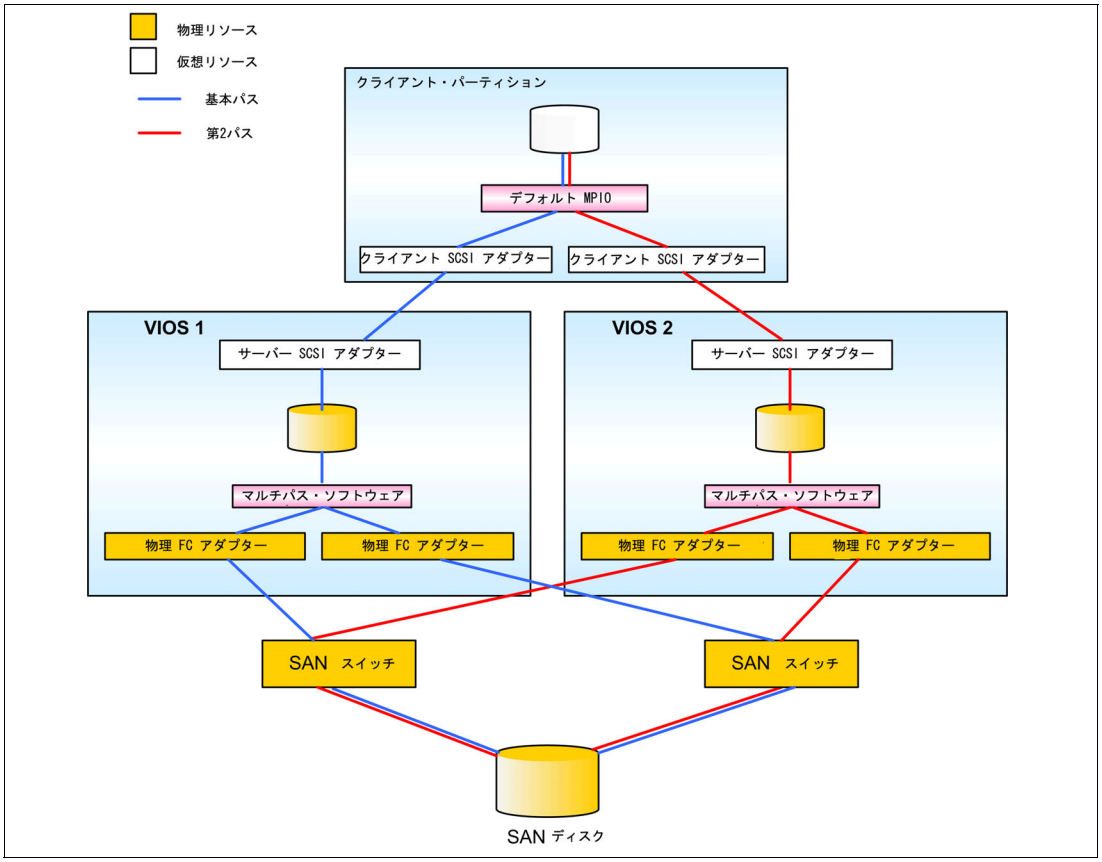


図 1-7 MPIO を使用して SAN ストレージに接続した二重仮想 I/O サーバー

仮想 I/O クライアント・パーティションと二重仮想 I/O サーバーの間で仮想 SCSI を使用した MPIO サポートは、フェイルオーバー・モードに対応しています。仮想 I/O クライアント・パーティション LUN は、通常は VIOS 1 への基本パスを使用し、フェイルオーバー時には VIOS 2 を使用するために第 2 のパスに切り替えます。両方のパスが有効になっていても、1 度に 1 つのパスだけが使用されます。二重仮想 I/O サーバーを使用した MPIO の構成方法の詳細については、4.7、『MPIO の計画と配置』（121 ページ）を参照してください。

この構成を使用すれば、定期保守のために 1 つの仮想 I/O サーバーをシャットダウンしても、すべてのアクティブなクライアントは、自動的にバックアップ仮想 I/O サーバーによってディスクにアクセスするようになります。仮想 I/O サーバーがオンライン状態に戻ったときにも、仮想 I/O クライアントの側では何の操作も必要ありません。

LVM のミラーリングによって、ディスク障害以上の障害にも対応できます。

LVM のミラーリング構成

LVM のミラーリングによって、従来の RAID 構成の場合よりも、アダプター、ストレージ・サーバー、仮想 I/O サーバーの冗長性をさらに追加できます。この構成によって、単なるディスク障害を超えた重大な障害に対する保護機能を確保できます。

仮想 I/O クライアント・パーティションで論理ボリュームのミラーリングを使用すれば、各仮想 I/O サーバーで、別のディスクに物理的に接続している仮想 SCSI デバイスを用意してから、クライアント・パーティションで AIX 5L の LVM のミラーリングを使用することによって、さらに高水準の可用性を実現できます。仮想 I/O サーバーから仮想 I/O クライアント・パーティションの仮想 SCSI デバイスとして論理ボリュームを使用する場合にも、クライアントのボリューム・グループのミラーリングが必要になります。その場合は、それぞれの仮想 SCSI デバイスを別々の SCSI ディスクに関連付け、それぞれの SCSI ディスクを 2 つの仮想 I/O サーバーのうちのいずれかによって制御するようにします。

図 1-8 は、クライアント・パーティションで LVM のミラーリングを使用した拡張構成を示したものです。二重仮想 I/O サーバーがクライアント・パーティションのディスクのホストになっています。この例の場合、クライアント・パーティションは、LVM のミラーリングを使用して 2 つの SCSI ディスクにアクセスします。

この構成の場合、1 つの仮想 I/O サーバーを保守のためにシャットダウンすると、ミラーが一時的に失われます。仮想 I/O サーバーのリストア時には、仮想 I/O クライアントでミラーの再同期を実行する必要があります。そのためには、クライアント側で **varyonvg** コマンドを実行します。

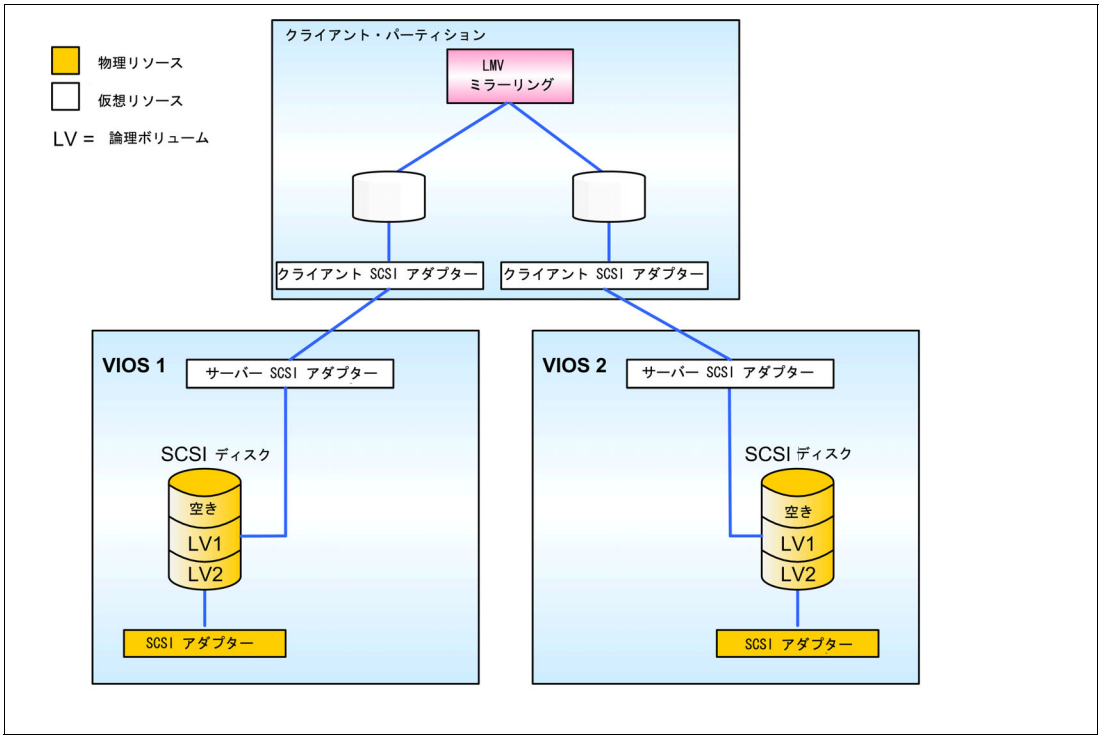


図 1-8 LVM のミラーリングを使用して SCSI ストレージに接続した二重仮想 I/O サーバー

1.3.3 定期保守

定期保守の時間を確保するのは、ベスト・プラクティスです。

仮想化によって、各クライアントは、別々のワークロードを同じシステムに統合できるようになります。その結果、コスト削減や効率アップが可能になります。ただし、同じシステムで多種多様なワークロードやクライアント・セットを実行している場合は、定期保守に関する考慮事項が新たに生まれることにもなります。

システム運用への影響を最小限に抑えるには、年間を通じて定期保守の時間帯をできるだけ多く設定する必要があります。多くの組織では、年次保守の時間帯を 2 回設定するのが適しているようです。特に、会計年度末にしばしば発生する要求量のピーク時を避けてスケジュールを設定するのは望ましいことです。

本書では、システムのさまざまな要素の保守を同時に実行するために仮想 I/O を構成する方法（2.5、『仮想 I/O サーバーの保守』（44 ページ）を参照）や、1 つのシステムから別のシステムにパーティションをすぐに変更する方法（4.6.4、『LPAR の移動』（119 ページ）を参照）を取り上げています。それでも、ダウン時間のスケジュール設定が必要なシステム保守作業（ファームウェアの更新など）があるのも事実です。

大規模な共用システムの管理者は、定期保守の時間帯のスケジュールを設定する必要があります。ただし、スケジュールは定期的に設定しておきますが、保守作業を実行するのは、必要な場合だけに限定します。保守作業を実行する必要がなければ、その時間帯もシステムを使用し続けることが可能になります。保守作業が必要であれば、スケジュールとして設定されている時間帯にのみその作業を実行します。そうすれば、すべてのユーザーが保守作業のスケジュールに基づいて、各自のスケジュールや操作をあらかじめ計画できるようになります。

保守作業のスケジュールを設定するときには、時間管理が必要です。スケジュールの回数が少なすぎると、その時間帯以外に保守作業を実行しなければならなくなる可能性が高くなります。一方、スケジュールの回数が多すぎると、クライアントの側が、実際にはその時間帯に保守作業が実行されないだろうと考えて、各自の作業を続けてしまう可能性があります。どちらの場合も、やがてはスケジュールの意味が薄れてしまいます。

2

管理、バックアップ、およびリストア

総合的なバックアップ計画がベスト・プラクティスです。

この章では、仮想 I/O 環境の管理に関する一般的なトピックのベスト・プラクティスを取り上げ、特に仮想 I/O サーバーのバックアップとリストアに重点を置いて説明します。さらに、ジョブのスケジュール設定、サーバーの始動とシャットダウンの順序付け、仮想 I/O サーバーの保守作業についても見ていきます。

仮想 I/O クライアントは、仮想 I/O サーバーのサービスに依存しているので、仮想 I/O 環境全体をバックアップし、システムのリストアと始動の手順に仮想 I/O サーバーを含めることは重要です。

2.1 仮想 I/O サーバーのバックアップとリストア

仮想 I/O サーバーは、IT 環境内の他のあらゆるサーバーと同様に、企業のデータ・リカバリー・プログラムの一環としてバックアップを作成することが必要です。このセクションでは、そのための計画について説明し、IBM AIX 5L または Linux のオペレーティング・システム環境の既存のバックアップ計画や新規のバックアップ計画の中に仮想 I/O サーバーのバックアップを組み込む方法について解説します。

このセクションでは、マシンのタイプや型式番号にかかわりなく、仮想 I/O サーバーを別のサーバーにリストアするために使用できる総合的なソリューションを取り上げます。仮想 I/O サーバーのバックアップだけを実行する場合は、このセクションの一部だけが必要になります。

2.1.1 仮想 I/O サーバーのバックアップを実行するタイミング

仮想 I/O サーバーの完全な災害時回復計画には、仮想デバイスと物理バッキング・デバイスを回復するために、以下のコンポーネントのバックアップを含める必要があります。仮想 I/O サーバーをバックアップするタイミングは、必要に応じて AIX 5L または Linux オペレーティング・システム・ベースの論理パーティションを再作成するサーバー・バックアップ計画の前になります。

これらのいずれかの項目が変更された場合は、バックアップが必要です。

仮想環境は以下のコンポーネントで構成されており、そのいずれかが変更された場合には、新しいバックアップが必要になります。

- 外部デバイスの構成（SAN やストレージ・サブシステムの構成など）。
- メモリー、CPU、仮想デバイス、物理デバイス。
- 仮想 I/O サーバー・オペレーティング・システム。
- 仮想環境と物理環境を結合するユーザー定義の仮想デバイス。これは仮想デバイスのマッピングまたはメタデータともいえます。

ここでは、オペレーティング・システム、インストール済みのアプリケーション、クライアントのアプリケーション・データなどは取り上げていません。仮想 I/O サーバーは、仮想 I/O オペレーティング・システムのデバイスとそのデバイスのリンクだけを管理しているからです。AIX 5L または Linux のオペレーティング・システム・ベースの仮想 I/O サーバーのクライアントについては、既存のサーバー・バックアップ計画の中で別個にバックアップ計画を作成する必要があります。

例えば、仮想ディスクと仮想ネットワークで構成されている AIX 5L サーバーの場合は、システムをバックアップするために、**mksysb**、**savevg**、またはそれに相当するツールを使用することも可能です。そのバックアップ計画では、仮想インフラストラクチャーを利用できます。例えば、物理共用イーサネット・アダプター（Shared Ethernet Adapter）を使用した仮想ネットワーク・インターフェースによって、IBM Tivoli Storage Manager サーバーへバックアップすることなどが考えられます。

仮想 I/O サーバーのバックアップだけを実行する場合は、仮想 I/O サーバー・オペレーティング・システムを取り上げた 3 番目のポイントが最も重要になります。

2.1.2 仮想 I/O サーバーのバックアップ計画

このセクションでは、バックアップ操作をいつ、どのように実行するかを説明します。

災害時回復の要件を確認します。

外部デバイスの構成

自然災害や人的災害によってサイト全体が破壊されてしまう可能性を考えれば、そのような状況に対応するための計画をエンドツーエンドのバックアップ計画に組み込む必要がありま

す。その計画は、災害時回復（Disaster Recovery）計画に組み込むのが普通ですが、いずれにしてもバックアップ計画全体の中で検討することが必要です。そのためのバックアップ計画は、ストレージ、ネットワーク装置、SAN デバイスなどをはじめとする数多くのハードウェアの特性によって左右されます。例えば、ネットワークの仮想ローカル・エリア・ネットワーク（VLAN）やストレージ・サブシステムからの論理装置番号（LUN）の情報などを記録しておくことが必要です。

この種の情報は本書の守備範囲を越えていますが、ここでこのようなことを取り上げているのは、物理サーバー環境や仮想サーバー環境の総合的な DR ソリューションがこの種の情報に依存することを強調するためです。その情報を収集して記録する方法は、プライマリー・サイトのインフラストラクチャー・システムのベンダーやモデルだけでなく、DR サイトの構成要素によっても左右されます。

ハードウェア管理コンソール で定義されているリソース

HMC 構成をバックアップします。

仮想 I/O サーバーの論理パーティションについては、CPU やメモリーの容量、使用する物理アダプターなどの情報が HMC で定義されています。さらに、仮想デバイスの構成（仮想イーサネット・アダプターとその仮想 LAN ID など）の情報も取り込む必要があります。この種のデータのバックアップとリストアは、本書の守備範囲を越えています。詳しくは、IBM Systems ハードウェア Information Center の「区画プロファイル・データのバックアップ」のトピックを参照してください。

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphai/backupprofdata.htm

特に災害時回復計画では、一部の HMC プロファイルを新しいハードウェアで最初から作成し直さなければならない場合もあります。その場合は、必要なイーサネット・カードの数などの構成情報を詳細に文書化しておくことが重要になります。システム・プランとビューアーを使用すれば、そのような情報を記録しやすくなりますが、いずれにしても、その情報が正しいかどうか、必要な情報がすべて記録されているかどうかを確認する必要があります。

仮想 I/O サーバー・オペレーティング・システム

VIOS オペレーティング・システムのデータをバックアップします。

仮想 I/O サーバー・オペレーティング・システムは、基本コード、フィックスパック、ディスク・サブシステムをサポートするカスタム・デバイス・ドライバー、ユーザー定義のカスタマイズで構成されています。ユーザー定義のカスタマイズの中には、Message of the Day（/etc/motd）の変更といった簡単なものもあれば、セキュリティー設定の変更などもあります。

初期のセットアップの後では、フィックスパックを適用するとき以外にこれらの設定が変更されることはまずありません。したがって、仮想 I/O サーバーのバックアップ計画では、フィックスパックの適用後または構成変更の後が重要になります。ユーザー定義の仮想デバイスについては次のセクションで取り上げますが、仮想 I/O サーバーのバックアップでその種のデータが一部取り込まれることは、注目に値します。この点を押さえておけば、仮想 I/O オペレーティング・システムとユーザー定義のデバイスを単一ステップでカバーするために、仮想 I/O オペレーティング・システムのバックアップ・スケジュールの頻度を高く設定できます。

Virtual I/O Server Version 1.3 のリリースから、ジョブのスケジュールを **crontab** コマンドで設定できるようになりました（2.3、『仮想 I/O サーバーでのジョブのスケジュール設定』（41 ページ））。そのコマンドを使用すれば、以下のバックアップ手順を定期的な間隔で実行するためのスケジュールを設定できます。

backupios コマンドは、テープ・デバイス、光ディスク・デバイス、ファイル・システム（ローカルのファイル・システムまたはリモート・マウントのネットワーク・ファイル・システム（NFS）のいずれか）に対して仮想 I/O サーバーのバックアップを実行します。

注 : 以下のような注意点があります。

- 仮想デバイスのマッピング（つまり、カスタマイズしたメタデータ）は、デフォルトでバックアップされます。特別な操作は必要ありません。
- クライアント・データはバックアップされません。

注 : デフォルトで **savevgstruct** コマンドがすべてのオンライン・ボリューム・グループに対して呼び出されることを示すために、ストレージ・プール storage01 とボリューム・グループ volgrp01 が定義されているシステムで以下のバックアップを作成しました。そのシステムで **lssp** コマンドを実行した場合の出力は、以下のとおりです。

```
$ lssp
Pool            Size(mb)   Free(mb)   Alloc Size(mb)  BDs
rootvg          139776     107136     128             0
storage01       69888      69760      64              1
volgrp01        69888      69760      64              1
```

CD または DVD への仮想 I/O サーバー・オペレーティング・システムのバックアップ

仮想 I/O サーバーを DVD-RAM にバックアップするためのコマンドは、例 2-1 のようなコマンドになります（DVD ドライブが /dev/cd0 にあるかどうかによって異なります）。

例 2-1 DVD-RAM への仮想 I/O サーバーのバックアップ

```
$ backupios -cd /dev/cd0 -udf -accept

Creating information file for volume group volgrp01.

Creating information file for volume group storage01.
Backup in progress.  This command can take a considerable amount of time
to complete, please be patient...

Initializing mkcd log: /var/adm/ras/mkcd.log...
Verifying command parameters...
Creating image.data file...
Creating temporary file system: /mkcd/mksysb_image...
Creating mksysb image...

Creating list of files to back up.
Backing up 44933 files.........
44933 of 44933 files (100%)
0512-038 mksysb: Backup Completed Successfully.
Populating the CD or DVD file system...
Copying backup to the CD or DVD file system...
........................................
........................................
........................................
........................................
..............
Building chrp boot image...
```

複数の CD にバックアップを作成する場合（例えば、バックアップが大きすぎて 1 枚の CD に収まらない場合）は、例 2-2 のようなコマンドを使用します。

例 2-2 CD への仮想 I/O サーバーのバックアップ

```
$ backupios -cd /dev/cd0 -accept
```

```
Creating information file for volume group volgrp01.

Creating information file for volume group storage01.
Backup in progress.  This command can take a considerable amount of time
to complete, please be patient...

Initializing mkcd log: /var/adm/ras/mkcd.log...
Verifying command parameters...
Creating image.data file...
Creating temporary file system: /mkcd/mksysb_image...
Creating mksysb image...

Creating list of files to back up.
Backing up 44941 files.............
44941 of 44941 files (100%)
0512-038 mksysb: Backup Completed Successfully.
Creating temporary file system: /mkcd/cd_fs...
Populating the CD or DVD file system...
Copying backup to the CD or DVD file system...
.
Building chrp boot image...
Creating temporary file system: /mkcd/cd_images...
Creating Rock Ridge format image: /mkcd/cd_images/cd_image_315528
Running mkisofs ...
..
mkrr_fs was successful.

Making the CD or DVD image bootable...
Writing the CD or DVD image to device: /dev/cd0...
Running cdrecord ...
Cdrecord 1.9 (rs6000-ibm-aix) Copyright (C) 1995-2000 Jorg Schilling
scsidev: '0,0'
scsibus: 0 target: 0 lun: 0
Using libscg version 'schily-0.1'
Device type    : Removable CD-ROM
Version        : 2
Response Format: 2
Capabilities   : WBUS16 SYNC
Vendor_info    : 'IBM     '
Identifikation : 'RMB000020501     '
Revision       : 'H106'
Device seems to be: Generic mmc2 DVD.
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags   : SWABAUDIO
Starting to write CD/DVD at speed 4 in write mode for single session.
Last chance to quit, starting real write in 31 seconds.
.........................................
.........................................
.....................Track 01: Total bytes read/written: 673855488/673855488 (329031
sectors).
.......
burn_cd was successful.

The backup will require an additional CD or DVD.
Remove the current writable CD or DVD (volume 1) from the
CD or DVD device and place a new writable CD or DVD (volume 2),
in device /dev/cd0.
Press the <enter> key when ready...

Copying the remainder of the backup to the CD or DVD file system...
```

```
Creating Rock Ridge format image: /mkcd/cd_images/cd_image_315528
Running mkisofs ...
.
mkrr_fs was successful.

Writing the CD or DVD image to device: /dev/cd0...
Running cdrecord ...
Cdrecord 1.9 (rs6000-ibm-aix) Copyright (C) 1995-2000 Jorg Schilling
scsidev: '0,0'
scsibus: 0 target: 0 lun: 0
Using libscg version 'schily-0.1'
Device type    : Removable CD-ROM
Version        : 2
Response Format: 2
Capabilities   : WBUS16 SYNC
Vendor_info    : 'IBM     '
Identifikation : 'RMB00020501     '
Revision       : 'H106'
Device seems to be: Generic mmc2 DVD.
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags   : SWABAUDIO
Starting to write CD/DVD at speed 4 in write mode for single session.
Last chance to quit, starting real write in 1 seconds.
.........................................
.....................................Track 01: Total bytes read/written:
496412672/496412672 (242389 sectors).
.......
burn_cd was successful.


Removing temporary file system: /mkcd/cd_images...
Removing temporary file system: /mkcd/cd_fs...
Removing temporary file system: /mkcd/mksysb_image...
```

ヒント：どんなメディアがサポートされているのかについては、ドライブのベンダーに確認してください。今回のテストでは、DVD マルチ・レコーダー・デバイスの DVD+R 形式のメディアに記録しました。そのために、上記のコマンドに **-cdformat** フラグを追加し、CD の代わりに DVD+R に焼き付けるようにしました。そのコマンドは以下のようになります。

```
$ backupios -cd /dev/cd0 -accept -cdformat
```

CD のプロセスでは、バックアップが大きすぎて複数のディスクにまたがる場合に、追加の CD（この場合は 1 枚の追加の CD）をドライブに挿入するように促すプロンプトが表示されます。順序を正しく識別できるように、メディアの外部にラベルを貼り付けておいてください。

CD のバックアップ方法でも DVD のバックアップ方法でも、仮想 I/O サーバーのリストアに使用できるブート可能メディアが生成されます（『仮想 I/O サーバー・オペレーティング・システムのリストア』（28 ページ）を参照）。

テープへの仮想 I/O サーバー・オペレーティング・システムのバックアップ

テープの場合は、例 2-3（23 ページ）のようにして **backupios** コマンドを使用します。

例 2-3　テープへの仮想 I/O サーバーのバックアップ

```
$ backupios -tape /dev/rmt0

Creating information file for volume group volgrp01.

Creating information file for volume group storage01.
Backup in progress.  This command can take a considerable amount of time
to complete, please be patient...


Creating information file (/image.data) for rootvg.

Creating tape boot image.............

Creating list of files to back up.
Backing up 44950 files.........................
44950 of 44950 files (100%)
0512-038 mksysb: Backup Completed Successfully.
```

ファイルへの仮想 I/O サーバー・オペレーティング・システムのバックアップ

ファイルにバックアップする場合も、**backupios** コマンドを使用します。テープや光メディアの場合に使用した前のコマンドではいずれも、結果として、仮想 I/O サーバーを直接回復するために使用できるブート可能メディアが生成されましたが、ファイルにバックアップする場合は、その点が大きく異なります。

ファイルへのバックアップでは、以下のいずれかが生成されます。

- リストアに必要なすべての情報を含んだ tar ファイル
- **mksysb** イメージ

どちらの方法も、リストアのためのインストール・サーバーが必要です。

リストア・サーバーは、以下のいずれかになります。

- Network Installation Manager on Linux 機能と **installios** コマンドを使用する HMC
- AIX 5L のネットワーク・インストール管理（NIM）サーバーと標準の **mksysb** システムのインストール

この両方の方法については、2.1.3、『仮想 I/O サーバーのリストア』（28 ページ）で取り上げます。

重要 : インストールに NIM サーバーを使用する場合は、仮想 I/O サーバーのインストールをサポートしている AIX 5L のレベルでなければなりません。そのため、NIM サーバーは、常に最新のテクノロジー・レベルとサービス・パックで稼働している必要があります。

backupios コマンドを使用して、仮想 I/O サーバーのローカル・ファイルにバックアップを作成することもできますが、より一般的なシナリオは、リモートの NFS ベースのストレージに対してバックアップを実行することです。この場合は、NIM サーバーを宛先として使用するのが理想的です。そのサーバーは、これらのバックアップのリストアにも使用できるからです。以下の例では、NIM サーバーのホスト名が SERVER5、仮想 I/O サーバーが LPAR01 になっています。

最初の手順は、NFS ベースのストレージのエクスポートを NIM サーバーにセットアップすることです。ここでは、/export/ios_backup という名前のファイル・システムをエクスポートします。この場合、/etc/exports は以下のようになります。

```
#more /etc/exports
/export/ios_backup -sec=sys:krb5p:krb5i:krb5:dh,rw=lpar01.itsc.austin.ibm.com,ro
ot=lpar01.itsc.austin.ibm.com
```

重要 : バックアップを正常に実行するには、NFS サーバーで、仮想 I/O サーバーの論理パーティションへエクスポートするファイル・システムに、ルート・アクセス NFS 属性を設定しておく必要があります。

さらに、IP アドレスとホスト名の両方について、NIM サーバーから仮想 I/O サーバーへの名前解決と逆引きの名前解決が機能していることを確認しなければなりません。仮想 I/O サーバーで名前解決を編集するには、**hostmap** コマンドを使用して /etc/hosts ファイルを操作するか、**cfgnamesrv** コマンドを使用して DNS パラメーターを変更します。

仮想 I/O サーバーのバックアップは大容量になることがあるため、NIM サーバーに関するシステムの **ulimits** が、大きなファイルの作成を可能にする設定になっていることを確認してください。

NFS のエクスポートと名前解決をセットアップしたら、ファイル・システムを仮想 I/O サーバーにマウントする必要があります。適切なユーザー ID を使用して、例 2-4 のように **mount** コマンドを実行できます。

例 2-4　リモートの NFS ベースのストレージのマウントとバックアップの実行

```
$ mount server5:/export/ios_backup /mnt
$ mount
  node       mounted        mounted over    vfs       date        options
-------- ---------------  ---------------  ------ ------------ ---------------
         /dev/hd4         /                jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /dev/hd2         /usr             jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /dev/hd9var      /var             jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /dev/hd3         /tmp             jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /dev/hd1         /home            jfs2   Jun 27 10:48 rw,log=/dev/hd8
         /proc            /proc            procfs Jun 27 10:48 rw
         /dev/hd10opt     /opt             jfs2   Jun 27 10:48 rw,log=/dev/hd8
server5.itsc.austin.ibm.com /export/ios_backup /mnt             nfs3   Jun 27 10:57
$ backupios -file /mnt

Creating information file for volume group storage01.

Creating information file for volume group volgrp01.
Backup in progress.  This command can take a considerable amount of time
to complete, please be patient...
$
```

このコマンドによって、フル・バックアップの tar ファイル・パッケージが作成されます。このパッケージには、**installios** コマンドによって HMC から仮想 I/O サーバーをインストールするために必要なすべてのリソース（**mksysb**、bosinst.data、ネットワーク・ブート・イメージ、SPOT）が含まれています。

この例では、コマンドの引数としてディレクトリーを指定しています。ファイル名は、nim_resources.tar になります。リストアの方法については、2.1.3、『仮想 I/O サーバーのリストア』（28 ページ）で取り上げます。

選択するリストア方式によっては、仮想 I/O サーバーの **mksysb** バックアップを以下の例のようにして作成することも可能です。この場合は、コマンドの引数として、絶対パスによるファイル名を指定します。本書の執筆時点で、NIM サーバーは、**mksysb** のリストア方式だけをサポートしています。

注 : NIM サーバーから nim_resources.tar ファイルに対して **installios** コマンドを実行する機能は、APAR IY85192 で有効になります。このサービスを使用できるかどうかについては、最寄りの IBM サポート担当員に確認してください。

仮想 I/O サーバーの **mksysb** バックアップは、フル・バックアップで作成される tar ファイルから抽出できるので、NIM サーバーを使ってリストアする場合はどちらの方式も有効です。

例 2-5 仮想 I/O サーバーのバックアップの作成

```
$ backupios -file /mnt/VIOS_BACKUP_27Jun2006_1205.mksysb -mksysb

/mnt/VIOS_BACKUP_27Jun2006_1205.mksysb  doesn't exist.

Creating /mnt/VIOS_BACKUP_27Jun2006_1205.mksysb

Creating information file for volume group storage01.

Creating information file for volume group volgrp01.
Backup in progress.  This command can take a considerable amount of time
to complete, please be patient...


Creating information file (/image.data) for rootvg.

Creating list of files to back up...
Backing up 45016 files.........................
45016 of 45016 files (100%)
0512-038 savevg: Backup Completed Successfully.
$
```

どちらの方法でも、HMC または NIM サーバーによる仮想 I/O サーバーの回復に使用できる仮想 I/O サーバー・オペレーティング・システムのバックアップが作成されます。

ユーザー定義の仮想デバイスのバックアップ

バックアップの一部として VIOS 構成を記録します。

前のセクションで作成した仮想 I/O サーバー・オペレーティング・システムのバックアップは、あくまでもオペレーティング・システムのバックアップであり、サーバーの再作成にはそれよりも多くの情報が必要です。

- 同じサーバーにリストアする場合は、データ構造（ストレージ・プールまたはボリューム・グループと論理ボリューム）などの一部の情報が rootvg 以外のディスクに残っている可能性があります。
- 新しいハードウェアにリストアする場合は、ディスク構造が存在しないので、これらのデバイスを自動的に回復することはできません。
- 物理デバイスが同じ場所に存在し、論理ボリュームなどの構造が無傷で残っていれば、仮想ターゲット SCSI や共用イーサネット・アダプター（Shared Ethernet Adapter）などの仮想デバイスは、リストア時に回復されます。

ディスク構造が存在せず、ネットワーク・カードが別のロケーション・コードとなる DR の状況では、以下の内容をバックアップする必要があります。

- ユーザー定義のディスク構造（ストレージ・プールまたはボリューム・グループと論理ボリュームなど）
- 仮想デバイスを物理デバイスに結合するリンク

これらのデバイスは、仮想 I/O サーバーの構築時または展開時に作成されるのが一般的ですが、新しいクライアントが追加されたり変更が生じたりすると、更新されます。したがって、週次スケジュールを設定するか、構成変更時に手動バックアップ手順を実行するのは、望ましいといえます。

ユーザー定義のディスク構造をバックアップするには、**savevgstruct** コマンドを使用します。このコマンドを実行すると、指定したボリューム・グループ（およびストレージ・プール）の構造のバックアップが /home/ios/vgbackups ディレクトリーに作成されます。

例えば、storage01 というボリューム・グループの構造をバックアップするには、以下のコマンドを実行します。

```
$ savevgstruct storage01

Creating information file for volume group storage01.

Creating list of files to back up.
Backing up 6 files
6 of 6 files (100%)
0512-038 savevg: Backup Completed Successfully.
```

注 : バックアップを正常に実行するには、ボリューム・グループまたはストレージ・プールをアクティブにする必要があります。**backupios** コマンドでは、アクティブ・ボリューム・グループまたはアクティブ・ストレージ・プールだけが自動的にバックアップされます。バックアップを開始する前に、必要に応じて、**lsvg** コマンドまたは **lssp** コマンドでボリューム・グループまたはストレージ・プールをリストし、**activatevg** コマンドでアクティブにしてください。

backupios コマンドを実行すると、バックアップが始まる前に、仮想 I/O サーバー上の rootvg 以外のすべてのアクティブなボリューム・グループまたはストレージ・プールに対して **savevgstruct** コマンドが自動的に呼び出されます。このコマンドは、バックアップが始まる前に呼び出されるので、システムのバックアップにボリューム・グループの構造が組み込まれることになります。つまり、**backupios** コマンドを使用すれば、ディスク構造もバックアップできるので、このコマンドの実行頻度を高くすることも可能です。

最後にバックアップされるのは、リンクの情報です。この情報は、例 2-6 のような **lsmap** コマンドの出力から収集できます。

例 2-6 lsmap コマンドの出力例

```
$ lsmap -net -all
SVEA   Physloc
------ --------------------------------------------
ent1   U9113.550.105E9DE-V2-C2-T1

SEA                  ent2
Backing device       ent0
Physloc              U787B.001.DNW108F-P1-C1-T1

$ lsmap -all
SVSA            Physloc                                      Client Partition ID
--------------- -------------------------------------------- ------------------
```

```
vhost0          U9113.550.105E9DE-V2-C10                    0x00000000

VTD                   target01
LUN                   0x8200000000000000
Backing device        lv02
Physloc

VTD                   vtscsi0
LUN                   0x8100000000000000
Backing device        lv00
Physloc
```

この例でディスクに関係があるのは、HMC 上でスロット 10（ロケーション・コードの C10 値）の vhost0 デバイスが lv02 デバイスにリンクされて target01 という名前になっていること、そして lv00 デバイスにリンクされて vtscsi0 という名前になっていることです。ネットワークについては、仮想イーサネット・アダプター ent1 が ent0 イーサネット・アダプターにリンクされ、ent2 共用イーサネット・アダプター（Shared Ethernet Adapter）になっています。

考慮事項 : 前の出力では、SEA コントロール・チャネル（SEA フェイルオーバー用）、ping 対象の IP アドレス、SEA デバイスのスレッド化を有効にするかどうかなどの情報を収集していません。これらの設定や変更内容（例えば MTU の設定）についても、別途文書化しておく必要があります。その方法については、このセクションの後半で取り上げます。

注 : 仮想 SCSI デバイスと仮想イーサネット・デバイスの参照として、vhost 番号や ent 番号ではなくスロット番号を使用することも非常に重要です。

vhost デバイスと ent デバイスは、ブート時に検出された時点で、または **cfgdev** コマンドの実行時に、仮想 I/O サーバーによって割り当てられます。それ以降のブート時または **cfgdev** コマンドの実行時にさらにデバイスを追加すると、それらのデバイスには順番に番号が割り振られます。

vhost0 の例に関して押さえておく必要がある重要な点は、論理ボリューム lv00 と lv02 にマップされるのが、vhost0 ではなく、スロット 10（ロケーション・コードの C10 値）の仮想 SCSI サーバーであるということです。vhost 番号と ent 番号は、初期のディスカバリー時に仮想 I/O サーバーによって順番に割り当てられるので、ユーザー定義のリンク・デバイスを再作成するときには、それらの番号は慎重に取り扱う必要があります。

これらのコマンドを使用して、追加情報を記録します。

lsmap コマンドに加え、以下の追加情報を収集することをお勧めします。その情報があれば、必要に応じて、インストール・メディアから仮想 I/O サーバーを再作成できます。

- ネットワーク設定

  netstat -state、**netstat -routinfo**、**netstat -routtable**、**lsdev -dev entX -attr**、**cfgnamesrv -ls**、**hostmap -ls**、**optimizenet -list**、**entstat -all entX**
- すべての物理ボリューム SCSI デバイスと論理ボリューム SCSI デバイス

  lspv、**lsvg**、**lsvg -lv VolumeGroup**
- すべての物理アダプターと論理アダプター

  lsdev -type adapter
- コード・レベルとユーザーとセキュリティー

  ioslevel、**viosecure -firewall -view**、**viosecure -view -nonint**、**motd**、**loginmsg**、**lsuser**

この情報は、前のディスク情報と同じ時間フレームで収集することをお勧めします。これらのコマンドをスクリプト化し、**tee** コマンドを使用して /home/padmin ディレクトリー内のファイルに情報を書き込むという解決策も考えられます。このジョブのスケジュール設定には、**crontab** コマンドを使用できます。

/home/padmin ディレクトリーは、**backupios** コマンドの実行時にバックアップされるので、バックアップの前に構成情報を収集しておくための適切な場所といえます。

詳しくは、2.3、『仮想 I/O サーバーでのジョブのスケジュール設定』（41 ページ）を参照してください。

2.1.3 仮想 I/O サーバーのリストア

データをリストアすることによって、バックアップ・ポリシーをテストします。

これまで、さまざまなバックアップ方法を取り上げ、バックアップの頻度にも触れてきました。ここからは、サーバーを最初から作成し直す方法を説明します。この作業の前提になるのは、仮想ディスクと仮想ネットワークが稼働する AIX 5L オペレーティング・システム・ベースのクライアント・パーティションにサービスを提供している仮想 I/O サーバーです。まだインストールしていない新しい 仮想 I/O サーバーから、リストアを実行しながら、それぞれのバックアップ計画をどこで使用するのかを取り上げていきます。

この総合的なエンドツーエンド・ソリューションは、今回のような極端な災害時回復シナリオに特化したものです。仮想 I/O サーバーをバックアップして同じサーバーにリストアする必要がある場合は、オペレーティング・システムのリストアの部分が特に関係してきます。

HMC 構成のリストア

自然災害や人的災害によってデータ・センター全体が破壊されたり利用不能になったりする極端な状況では、システムを災害時回復サイトにリストアしなければならない場合もあります。その場合は、設定を回復するための別の HMC とサーバーの場所が必要になります。さらに、災害時回復サーバーを設置して、HMC プロファイルでシステムの回復をすぐに開始できる状態にしておくことも必要です。

この点に関する詳細は本書の守備範囲を越えていますが、以下のセクションの内容とあわせて、こうした作業が DR 回復の第一歩になります。

その他の IT インフラストラクチャー・デバイスのリストア

ほかにも、ネットワーク・ルーター、スイッチ、ストレージ・エリア・ネットワーク、DNS サーバーなどをはじめとする数多くの IT インフラストラクチャー・デバイスがあります。これらのデバイスも、総合的な IT 災害時回復ソリューションに組み込まなければなりません。すでに見たとおり、回復作業の正常な実行のためには、仮想 I/O サーバーだけでなく、IT インフラストラクチャー全体がこれらの一般的なサービスに依存することになります。

仮想 I/O サーバー・オペレーティング・システムのリストア

IT インフラストラクチャーと、HMC によって管理されているサーバーが稼働しており、仮想 I/O サーバーのシステム・プロファイルで、オリジナル・システムに存在するのと同じ数の物理イーサネット・アダプターとディスクが定義されていれば、仮想 I/O サーバー・オペレーティング・システムをリストアすることができます。バックアップとリストアに関する標準ポリシーの一環として、仮想 I/O サーバー・オペレーティング・システムのリストアを行うには、ここがエントリー・ポイントになります。

バックアップからのインストールには、AIX 5L Version 5.3 TL5 のサービス APAR IY85192 をお勧めします。

使用したバックアップ方式に応じて、適切なリストア方式を選択しなければなりません。その点をこれから見ていきます。

CD または DVD のバックアップからの仮想 I/O サーバーのリストア

この章でバックアップ手順について取り上げたように、スタンドアロン・バックアップのリストアに使用できるブート可能メディアが生成されます。

バックアップ・セットの最初のディスクを光ディスク・ドライブに挿入し、SMS モードでマシンをブートし、そのパーティションで CD または DVD のドライブが使用可能になっていることを確認します。SMS のメニューを使用して、光ディスク・ドライブからのインストールを選択し、通常のインストール手順を実行します。

注 : CD または DVD のバックアップが複数のディスクにまたがる場合は、ディスク・セット内の次のディスクを挿入することを求める以下のようなメッセージがインストール時に表示されます。

```
Please remove volume 1, insert volume 2, and press the ENTER key.
```

テープのバックアップからの仮想 I/O サーバーのリストア

テープの場合の手順も、CD または DVD の場合とよく似ています。テープもブート可能メディアなので、テープ・ドライブにバックアップ・メディアを挿入し、SMS モードでブートします。テープ・ドライブからのインストールを選択し、先ほどと同じ手順を実行します。

HMC からの仮想 I/O サーバーのリストア

ファイルへのフル・バックアップ（**mksysb** バックアップではなく、nim_resources.tar ファイル）を作成した場合は、HMC を使用して、**installios** コマンドによるインストールを実行できます。

この tar ファイルは、DVD でも NFS 共有でも使用できます。このシナリオで採用するのは、NFS 方式です。nim_resources.tar ファイルが含まれているディレクトリーが NFS サーバーからエクスポートされているという前提で、適切なユーザー ID によって HMC にログオンし、例 2-7 のようにして **installios** コマンドを実行します（この例では、入力項目を太字で示しています）。

注 : NFS ロケーション server5:/export/ios_backup/ の末尾のスラッシュは、そのとおりにコマンドに組み込む必要があります。

以下の例のように、クライアント・ネットワークを構成するかどうかの設定は、no にする必要があります。このバックアップのインストールに使用する物理アダプターは、SEA によってすでに使用されている可能性があり、もしそうであれば、IP 構成は失敗するからです。必要に応じて、インストール後にコンソール・セッションを使用してログインし、IP を構成してください。

例 2-7 HMC からの installios プロセス

```
hscroot@server1:~> installios
The following objects of type "managed system" were found.  Please select one:
1.  p570-ITSO
Enter a number: 1
The following objects of type "virtual I/O server partition" were found.  Please select
one:
1.  VIOS_DR_MACHINE
2.  VIO_Server1
Enter a number (1-2): 1
The following objects of type "profile" were found.  Please select one:
1.  normal
Enter a number: 1
Enter the source of the installation images [/dev/cdrom]: server5:/export/ios_backup/
```

```
Enter the client's intended IP address: 9.3.5.123
Enter the client's intended subnet mask: 255.255.255.0
Enter the client's gateway: 9.3.5.41
Enter the client's speed [100]:auto
Enter the client's duplex [full]:auto
Would you like to configure the client's network after the
        installation [yes]/no? no
Retrieving information for available network adapters
This will take several minutes...
The following objects of type "Ethernet adapters" were found.  Please select one:
1.  ent U9117.570.107CD9E-V2-C2-T1 523300002002
2.  ent U7311.D20.10832BA-P1-C01-T1 00096b6e8458
3.  ent U7311.D20.10832BA-P1-C01-T2 00096b6e8459
Enter a number (1-3): 2
Here are the values you entered:
managed system = p570-ITSO
virtual I/O server partition = VIOS_DR_MACHINE
profile = normal
source = server5:/export/ios_backup/
IP address = 9.3.5.123
subnet mask = 255.255.255.0
gateway = 9.3.5.41
speed = 100
duplex = full
configure network = no
ethernet adapters = 00:09:6b:6e:84:58
Press enter to proceed or type Ctrl-C to cancel...
nimol_config MESSAGE: No NIMOL server hostname specified, using server1.itsc.austin.ibm.com
as the default.
Starting RPC portmap daemon
done
Starting kernel based NFS server
done
nimol_config MESSAGE: Added "REMOTE_ACCESS_METHOD /usr/bin/rsh" to the file
"/etc/nimol.conf"
nimol_config MESSAGE: Removed "disable = yes" from the file "/etc/xinetd.d/tftp"
nimol_config MESSAGE: Added "disable = no" to the file "/etc/xinetd.d/tftp"
Shutting down xinetd:
done
Starting INET services. (xinetd)
done
nimol_config MESSAGE: Removed "SYSLOGD_PARAMS=" from the file "/etc/sysconfig/syslog"
nimol_config MESSAGE: Added "SYSLOGD_PARAMS=-r " to the file "/etc/sysconfig/syslog"
nimol_config MESSAGE: Removed "local2,local3.*  -/var/log/localmessages" from the file
"/etc/syslog.conf"
nimol_config MESSAGE: Added "local3.*  -/var/log/localmessages" to the file
"/etc/syslog.conf"
nimol_config MESSAGE: Added "local2.* /var/log/nimol.log" to the file "/etc/syslog.conf"
Shutting down syslog services
done
Starting syslog services
done
nimol_config MESSAGE: Executed /usr/sbin/nimol_bootreplyd -l -d -f /etc/nimoltab -s
server1.itsc.austin.ibm.com.
nimol_config MESSAGE: Successfully configured NIMOL.
nimol_config MESSAGE: target directory: /info/default5
nimol_config MESSAGE: Executed /usr/sbin/iptables -I INPUT 1 -s server5 -j ACCEPT.
nimol_config MESSAGE: source directory: /mnt/nimol
nimol_config MESSAGE: Checking /mnt/nimol/nim_resources.tar for existing resources.
nimol_config MESSAGE: Executed /usr/sbin/iptables -D INPUT -s server5 -j ACCEPT.
```

```
nimol_config MESSAGE: Added "/info/default5 *(rw,insecure,no_root_squash)" to the file
"/etc/exports"
nimol_config MESSAGE: Successfully created "default5".
nimol_install MESSAGE: The hostname "lpar11.itsc.austin.ibm.com" will be used.
nimol_install MESSAGE: Added "CLIENT lpar11.itsc.austin.ibm.com" to the file
"/etc/nimol.conf"
nimol_install MESSAGE: Added
"lpar11.itsc.austin.ibm.com:ip=9.3.5.123:ht=ethernet:gw=9.3.5.41:sm=255.255.255.0:bf=lpar11
.itsc.austin.ibm.com:sa=9.3.5.194:ha=00096b6e8458" to the file "/etc/nimoltab"
nimol_install MESSAGE: Executed kill -HUP 5149.
nimol_install MESSAGE: Created /tftpboot/lpar11.itsc.austin.ibm.com.
nimol_install MESSAGE: Executed /sbin/arp -s lpar11.itsc.austin.ibm.com 00:09:6b:6e:84:58
-t ether.
nimol_install MESSAGE: Executed /usr/sbin/iptables -I INPUT 1 -s lpar11.itsc.austin.ibm.com
-j ACCEPT.
nimol_install MESSAGE: Created /info/default5/scripts/lpar11.itsc.austin.ibm.com.script.
nimol_install MESSAGE: Created /tftpboot/lpar11.itsc.austin.ibm.com.info.
nimol_install MESSAGE: Successfully setup lpar11.itsc.austin.ibm.com for a NIMOL install
# Connecting to VIOS_DR_MACHINE
# Connected
# Checking for power off.
# Power off complete.
# Power on VIOS_DR_MACHINE to Open Firmware.
# Power on complete.
# Client IP address is 9.3.5.123.
# Server IP address is 9.3.5.194.
# Gateway IP address is 9.3.5.41.
# /pci@800000020000011/pci@2/ethernet@1 ping successful.
# Network booting install adapter.
# bootp sent over network.
# Network boot proceeding, lpar_netboot is exiting.
# Finished.
```

この時点で、ユーザー入力が必要になった場合のために、リストア先のサーバーの端末コンソールを開きます。

ヒント : このコマンドによるリストアに長い時間がかかる場合は、ネットワークの速度や全二重、半二重の構成のエラーが原因になっている可能性が高いといえます。

NIM は、多くの実稼働環境で役立つツールです。

NIM サーバーによる仮想 I/O サーバーのリストア

installios コマンドは NIM サーバーでも使用できますが、現時点では、仮想 I/O サーバーの基本メディアからのインストールだけがサポートされています。NIM サーバーを使用する方法では、**mksysb** イメージをインストールします。前に取り上げた **backupios** コマンドで **-mksysb** フラグを使用して生成した **mksysb** イメージを使用することも、nim_resources.tar ファイルから抽出した **mksysb** イメージを使用することも可能です。

注 : NIM サーバーから nim_resources.tar ファイルに対して **installios** コマンドを実行する機能は、APAR IY85192 で有効になります。このサービスを使用できるかどうかについては、最寄りの IBM サポート担当員に確認してください。

どちらの方法で **mksysb** イメージを NIM サーバーに配置したとしても、以下のようにして、**mksysb** を NIM リソースとして登録することが必要です。

```
# nim -o define -t mksysb -aserver=master
-alocation=/export/ios_backup/VIOS_BACKUP_27Jun2006_1205.mksysb VIOS_mksysb
# lsnim VIOS_mksysb
VIOS_mksysb     resources       mksysb
```

mksysb をリソースとして登録したら、**mksysb** から SPOT を生成します。

```
# nim -o define -t spot -a server=master -a location=/export/ios_backup/SPOT -a
source=VIOS_mksysb VIOS_SPOT

Creating SPOT in "/export/ios_backup/SPOT" on machine "master" from "VIOS_mksysb" ...
Restoring files from BOS image.  This may take several minutes ...
# lsnim VIOS_SPOT
VIOS_SPOT     resources       spot
```

SPOT と **mksysb** イメージを NIM に定義したら、バックアップから仮想 I/O サーバーをインストールします。インストール先のマシンが NIM に定義されていない場合は、そのマシンを登録してから、**smitty nim_bosinst** ファスト・パスを入力します。定義した **mksysb** イメージと SPOT を選択し、各オプションを 図 2-1（32 ページ）に合わせて設定します。

```
                                                Install the Base Operating System on Standalone Clients

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Installation Target                                 lpar01
* Installation TYPE                                   mksysb
* SPOT                                                VIOS_SPOT
  LPP_SOURCE                                         []
  MKSYSB                                              VIOS_mksysb

  BOSINST_DATA to use during installation            []
  IMAGE_DATA to use during installation              []
  RESOLV_CONF to use for network configuration       []
  Customization SCRIPT to run after installation     []
  Customization FB Script to run at first reboot     []
    ACCEPT new license agreements?                   [yes]
  Remain NIM client after install?                   [no]
  PRESERVE NIM definitions for resources on          [yes]
    this target?

  FORCE PUSH the installation?                       [no]

  Initiate reboot and installation now?              [no]
    -OR-
  Set bootlist for installation at the               [no]
     next reboot?

  Additional BUNDLES to install                      []
    -OR-
  Additional FILESETS to install                     []
    (bundles will be ignored)

  installp Flags
    COMMIT software updates?                         [yes]
    SAVE replaced files?                             [no]
    AUTOMATICALLY install requisite software?        [yes]
    EXTEND filesystems if space needed?              [yes]
    OVERWRITE same or newer versions?                [no]
    VERIFY install and check file sizes?             [no]
    ACCEPT new license agreements?                   [no]
      (AIX V5 and higher machines and resources)
    Preview new LICENSE agreements?                  [no]

  Group controls (only valid for group targets):
    Number of concurrent operations                  []
    Time limit (hours)                               []

  Schedule a Job                                     [no]
  YEAR                                               []
  MONTH                                              []
  DAY (1-31)                                         []
  HOUR (0-23)                                        []
  MINUTES (0-59)                                     []
```

図 2-1　NIM による仮想 I/O サーバーのリストア

重要 : 「Remain NIM client after install」フィールドが no に設定されていることに注目してください。これが no に設定されていない場合は、NIM インストールの最後の手順として、今回の仮想 I/O サーバーのインストールに使用した物理アダプターに IP アドレスを構成しようとします。その IP アドレスは、NIM サーバーへの登録に使用されているものです。そのアダプターが既存の共用イーサネット・アダプター（Shared Ethernet Adapter）によって既に使用されている場合は、以下のようなエラー・メッセージが表示される原因となります。そうなった場合は、必要に応じて仮想 I/O サーバーをリブートしてから、端末セッションを使用して仮想 I/O サーバーにログオンし、IP アドレスの情報と SEA を除去してください。その後、SEA を再作成し、IP アドレスの情報を仮想 I/O サーバーに戻します。

```
inet0 changed
if_en: ns_alloc(en0) failed with errno = 19
if_en: ns_alloc(en0) failed with errno = 19
Method error (/usr/lib/methods/chgif):
        0514-068 Cause not known.
0821-510 ifconfig: error calling entry point for /usr/lib/drivers/if_en: The specified
device does not exist.
0821-103 : The command /usr/sbin/ifconfig en0 inet  9.3.5.111 arp netmask 255.255.255.0
mtu 1500 up failed.
0821-007 cfgif: ifconfig command failed.
        The status of"en0" Interface in the current running system is uncertain.
0821-103 : The command /usr/lib/methods/cfgif -len0 failed.
0821-510 ifconfig: error calling entry point for /usr/lib/drivers/if_en: The specified
device does not exist.
0821-103 : The command /usr/sbin/ifconfig en0 inet 9.3.5.111 arp netmask 255.255.255.0
mtu 1500 up failed.
0821-229 chgif: ifconfig command failed.
The status of"en0" Interface in the current running system is uncertain.

mktcpip: Problem with command: chdev
, return code = 1
if_en: ns_alloc(en0) failed with errno = 19
if_en: ns_alloc(en0) failed with errno = 19
if_en: ns_alloc(en0) failed with errno = 19
if_en: ns_alloc(en0) failed with errno = 19
```

NIM サーバーでバックアップ・イメージを復元するための準備ができたので、今度は仮想 I/O サーバーでリモート IPL のセットアップを完了する必要があります。その手順については、IBM eServer™ pSeries および AIX インフォメーション・センターの「AIX 情報」→「インストールおよびマイグレーション」→「区分化」のカテゴリーを参照してください。

http://publib16.boulder.ibm.com/pseries/index.htm

ヒント : NIM を使用した場合に、インストールの問題が発生する主な原因の 1 つは、NIM サーバーからの NFS エクスポートです。NIM サーバーで /etc/exports ファイルが正しいことを確認してください。

仮想 I/O サーバーのインストールは完了しますが、既存のサーバーへのリストアと新しい災害時回復サーバーへのリストアには大きな違いがあります。NIM インストールの場合は、ターゲットでリソースのための NIM 定義を保存するオプションを使用できます。そのオプションを使用すると、NIM は、オリジナル・バックアップに定義されていたすべての仮想デバイスをリストアしようとします。ただし、そのためには、ロケーション・コードが変わらないように、パーティション・プロファイル（仮想および物理）に同じデバイスが定義されていることが必要です。

したがって、論理パーティション・プロファイルが変更されていなければ、仮想ターゲットSCSI デバイスと共用イーサネット・アダプター（Shared Ethernet Adapter）はすべて回復され、再作成の必要はないことになります。同じマシンにリストアする場合は、依存条件として、インポート対象の rootvg 以外のボリューム・グループが存在し、それらのボリューム・グループに含まれている論理ボリューム構造が無傷で残っていることが必要です。その点を確認するために、診断用の CD から仮想 I/O サーバーをブートし、仮想 I/O サーバー・オペレーティング・システムのディスクをフォーマットして検査することによって、すべてのデータを（デモンストレーションのために）破棄してみました。ボリューム・グループやストレージ・プールが含まれている残りのディスクについては、何の操作も実行していません。

その状態で NIM サーバーを使用して、バックアップ・イメージを元の仮想 I/O サーバー・オペレーティング・システムのディスクにリストアしました。インストール後に仮想デバイスを調べてみると、例 2-8 のように、vtscsi アダプターと共用イーサネット・アダプター（Shared Ethernet Adapter）がすべて回復されていました。

例 2-8 同じ論理パーティションへの仮想 I/O サーバー（リカバー・デバイスを設定）のリストア

```
$ lsdev -virtual
name            status                                      description
ent1            Available  Virtual I/O Ethernet Adapter (l-lan)
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
target01        Available  Virtual Target Device - Logical Volume
vtscsi0         Available  Virtual Target Device - Logical Volume
ent2            Available  Shared Ethernet Adapter
$ lsmap -all
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost0          U9113.550.105E9DE-V2-C10                    0x00000000

VTD                   target01
LUN                   0x8200000000000000
Backing device        lv02
Physloc

VTD                   vtscsi0
LUN                   0x8100000000000000
Backing device        lv00
Physloc

$ lsmap -net -all
SVEA   Physloc
------ --------------------------------------------
ent1   U9113.550.105E9DE-V2-C2-T1

SEA                   ent2
Backing device        ent0
Physloc               U787B.001.DNW108F-P1-C1-T1
```

先ほどの HMC 回復手順によって、同じような仮想デバイスが定義されている別の論理パーティションにリストアする場合は、リンク・デバイスが欠落することになります。

注 : これらのデバイスは、マシンによって常に変わってきます。マシンのシリアル番号が仮想デバイスのロケーション・コードに組み込まれているからです。例えば、以下のようになります。

```
$ lsdev -dev ent1 -vpd
  ent1     U9113.550.105E9DE-V2-C2-T1  Virtual I/O Ethernet Adapter (l-lan)
.
. Lines omitted for clarity
.
```

そのようになるのは、リンクのためのバッキング・デバイスが存在しないからです。つまり、物理ロケーション・コードが変更されているので、マッピングが失敗してしまいます。例 2-9 は、元々 p5-550 で稼働していた仮想 I/O サーバーを、同じ仮想デバイスが同じスロットに定義されている p5-570 に同じ手順でリストアした場合の結果です。

例 2-9 別のサーバーにリストアする場合に回復されるデバイス

```
$ lsdev -virtual
name            status                                          description
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
$ lsmap -all -net
SVEA   Physloc
------ --------------------------------------------
ent2   U9117.570.107CD9E-V2-C2-T1

SEA                 NO SHARED ETHERNET ADAPTER FOUND
$ lsmap -all
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost0          U9117.570.107CD9E-V2-C10                    0x00000000

VTD                   NO VIRTUAL TARGET DEVICE FOUND
$
```

この場合は、ユーザー定義の仮想デバイスとバッキング・ディスク構造を回復する必要があります。

ユーザー定義の仮想デバイスとディスク構造の回復

オリジナルの仮想 I/O サーバーには、rootvg に由来する追加のディスクがさらに 2 つあります。これらのディスクが SAN ディスクであったり、仮想 I/O クライアントに直接マップされているディスクであったりした場合は、仮想デバイス・リンクをそのままリストアできます（ここでリンクするのは hdisk デバイスです）。ところが、それらのディスクに論理ボリュームまたはストレージ・プールの構造がある場合は、その構造を最初にリストアしなければなりません。そのためには、ボリューム・グループのデータ・ファイルを使用する必要があります。

ボリューム・グループまたはストレージ・プールのデータ・ファイルは、前のバックアップ処理で取り込まれているはずです。**backupios** コマンドによってフル・バックアップを実行した場合は、/home/ios/vgbackups ディレクトリーにそれらのファイルが置かれています。以下のコマンドを実行すると、使用可能なバックアップがすべてリストされます。

```
$ restorevgstruct -ls
total 16
-rw-r--r--   1 root     staff          1486 Jul 10 17:56 storage01.data
-rw-r--r--   1 root     staff          1539 Jul 10 17:56 volgrp01.data
```

```
$
```

以下の例（例 2-10）では、リストア対象として、新しいブランクのディスクがいくつかあり、同じボリューム・グループ storage01 と volgrp01 があります。

例 2-10　リストアするディスクとボリューム・グループ

```
$ lspv
NAME            PVID                                VG              STATUS
hdisk0          00c7cd9e1f89130b                    rootvg          active
hdisk1          00c7cd9e1adcd58a                    rootvg          active
hdisk2          0021768aaae4ae06                    None
hdisk3          0021768accc91a48                    None
hdisk4          none                                None
hdisk5          0021768aa445b99d                    None
```

restorevgstruct コマンドを実行すると、それらのボリューム・グループ構造が空のディスクにリストアされます。例 2-11 は、そのコマンドを呼び出す方法を示したものです。

例 2-11　ボリューム・グループ構造のリストア

```
$ $ restorevgstruct -vg storage01 hdisk2
hdisk2
storage01
lv00

Will create the Volume Group:   storage01
Target Disks:   Allocation Policy:
        Shrink Filesystems:     no
        Preserve Physical Partitions for each Logical Volume:   no
```

すべての論理ボリューム構造をリストアした時点で残っている手順は、物理バッキング・デバイスを仮想にリンクする仮想デバイスのリストアだけになります。それらの仮想デバイスをリストアするには、『ユーザー定義の仮想デバイスのバックアップ』（25 ページ）のバックアップ手順で記録した **lsmap** の出力を使用するか、作成時の文書を使用します。すでに見たとおり、それらのリンクをリストアするときには、スロット番号とバッキング・デバイスを使用することが重要です。

共用イーサネット・アダプター（Shared Ethernet Adapter）のリストアでは、正しい仮想イーサネット・アダプターを正しい物理アダプターに結合するリンクが必要になります。通常、物理アダプターは、組織のネットワーク・インフラストラクチャー内の VLAN に配置されます。したがって、正しい仮想 VLAN を正しい物理 VLAN にリンクすることが重要です。この作業では、ネットワーク・サポート・チームの支援やスイッチの構成データが役立ちます。

DR リストアでは、仮想リンク・デバイス（vtscsi と SEA）の再作成のための手作業がさらに必要になるので、十分なユーザー文書への依存度が高くなります。仮想 I/O サーバーのマルチパス設定を保存する必要がなければ、インストール・メディアから仮想 I/O サーバーを新規にインストールしてから、作成時の文書に基づいてリストアを実行するという解決策もあり得ます。

mkvdev コマンドを実行してマッピングを再作成したら、仮想 I/O サーバーが仮想ディスクと仮想ネットワークのホストになるので、次はこの環境を使用して、AIX 5L または Linux のクライアントを再作成できます。

AIX 5L または Linux のオペレーティング・システムのリストア

仮想 I/O サーバーが操作可能な状態に戻り、すべてのデバイスを再作成できたら、AIX 5L または Linux のクライアントのリストアを開始する準備が整ったことになります。そのため

の手順は、それぞれの組織ですでに定義されているはずであり、基本的には、専用のディスク・リソースやネットワーク・リソースを使用するサーバーのリストア手順と特に変わりはありません。それでも、使用するソリューションによって手順は左右されるので、それぞれの組織で策定する必要があります。

AIX 5L クライアントの場合は、その情報をインフォメーション・センターで確認できます。

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.baseadmn/doc/baseadmndita/backmeth.htm

2.2 新しい LPAR の定義

新しい構成を作成する前に計画を策定するのがベスト・プラクティスです。

新しい LPAR を定義する場合は、仮想デバイスを使用するかどうかにかかわりなく、計画が非常に重要になります。計画の策定を始めるにあたり、企業内のすべての管理対象システムに適用される命名規則とスロット番号の番号付け規則を整備しておくことをお勧めします。役立つ番号付け規則については、3.3.2、『仮想デバイスのスロット番号』（62 ページ）と 4.3.2、『仮想デバイスのスロット番号』（103 ページ）を参照してください。さらに IBM では、管理対象システムの計画と設計に役立つ System Planning Tool などのツールを以下のサイトに用意しています。

http://www.ibm.com/servers/eserver/support/tools/systemplanningtool/

2.2.1 新しい仮想 I/O サーバーの作成

新しい仮想 I/O サーバーを作成するための手順については、*「Advanced POWER Virtualization on IBM System p5」*（SG88-4018）を参照してください。仮想 I/O サーバーの LPAR の作成にあたっては、最初のウィンドウでパーティション環境を選択する必要があります。仮想 I/O サーバーが使用不可の状態であれば、仮想化もアクティブになりません。仮想化をアクティブにするには、まず以下の Web ページにアクセスして、活動化コードを入手する方法を確認してください。このコードはフィーチャー・コードであり、マシン構成に組み込む必要があります。その活動化コードについては、以下のサイトを参照してください。

http://www.ibm.com/systems/p/cod/

HMC の「属性（Server Properties）」タブでマシンのタイプとシリアル番号を確認する必要があります。その Web サイトでマシンのタイプとシリアル番号を入力してから、VET コード（仮想化をアクティブにするためのコード）を見つけてください。

パーティション・プロファイルを作成するときには、SMS モードでクライアントをブートするために、すべてのプロファイルで通常バージョンと SMS バージョンを用意する必要はありません。「活動化（Activation）」ウィンドウで**「詳細（Advanced）」**を選択し、ブート・モードを**「SMS」**に変更することによって、ブート・モードを変更できます（図 2-2 を参照）。

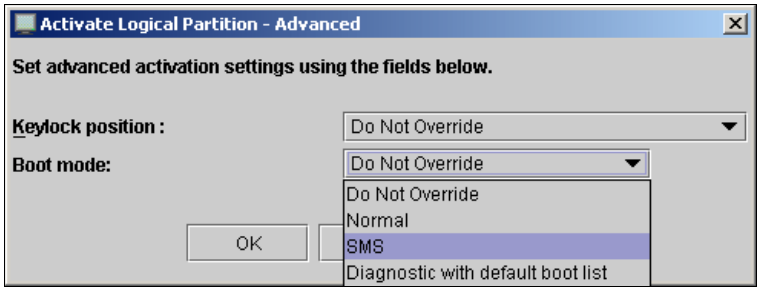


図 2-2　論理パーティションのアクティブ化

Virtual I/O Server Version 1.2 以降では、仮想 I/O サーバーを経由して光ディスク・デバイスを仮想 I/O クライアントに対して仮想化することが可能です。例 2-12（39 ページ）では、仮想デバイスが SCSI CD-ROM デバイスとして表示されています。このメディアは、一度に 1 つの仮想 I/O クライアントにしか割り当てることができません。別の仮想 I/O クライアントでそのメディア・デバイスが必要になった場合は、現在の仮想 I/O クライアントでその割り当てを解除してから、新しい仮想 I/O クライアントに割り当てる必要があります。光ディスク・デバイスを共用する場合は、NFS などの製品を使用できます。管理対象システムで光メディアを共用するために NFS を使用する場合は、仮想イーサネットを使用してその光メディアにアクセスできます。ただし、NFS 共用光ディスク・デバイスを使用して光メディアから仮想 I/O クライアントをブートすることはできません。

例 2-12　仮想 I/O クライアントの SMS メニュー

```
PowerPC Firmware
Version SF240_219
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------------
Select Device
Device  Current  Device
Number  Position  Name
1.  -      Virtual Ethernet
                ( loc=U9113.550.105E9DE-V3-C3-T1 )
2.  -      SCSI CD-ROM
                ( loc=U9113.550.105E9DE-V3-C21-T1-W8200000000000000-L0 )









-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type menu item number and press Enter or select Navigation key:
```

パーティション・プロファイルの作成時には、CPU とメモリーを現実的に見積もります。

仮想 I/O サーバーの LPAR のメモリー値を選択するときには、最大値を慎重に選択してください。128 GB などの大きな値を選択すると、大量のメモリーを滞留させる結果になってしまいます。ハイパーバイザー・ファームウェアは、最大値として入力された値の 64 分の 1 をハイパーバイザー・ファームウェアのシステム・メモリーとして確保します。したがって、128 GB を選択した場合は、ハイパーバイザーのシステム・メモリーとして 2 GB のメモリーを確保することになりますが、ほとんどの場合、それが必要とされることはありません。使用されるハイパーバイザーのメモリーを確認するには、IBM System Planning Tool（SPT）が利用できます。

CPU の数を選択する場合は、CPU の必要数を現実的に見積もってください。例えば、現在のワークロードには 2 つの CPU で対応できますが、将来的に 4 つの CPU に拡張する可能性があるとしましょう。その場合に、CPU の数を 12 に設定することはありません。実動システムのワークロードによっては小さな値を設定することもできますが、ネットワーク・トラフィックが高いレベルになりそうであれば、最初は少なくとも 1 つの CPU 全体を仮想 I/O サーバーに割り振るようにします。一般的に、ネットワーク・トラフィックが高いレベルになれば、CPU の使用率は上がります。一方、ディスク・トラフィックの場合は、それと同じほどの CPU を必要としないのが普通です。低速のデバイスであれば、I/O がキューに入れられるからです。CPU の見積もりの詳細については、5.4.2、『仮想 I/O サーバーの CPU の計画』（137 ページ）を参照してください。

共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）を実装した二重仮想 I/O サーバーのシナリオを実行する場合は、プライマリー仮想 I/O サーバーの「幹線優先順位（Trunk priority）」パネルで値 **1** を選択し、スタンバイ仮想 I/O サーバーで値 **2** を使用します。さらに、2 つの仮想 I/O サーバーの間に、共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）で使用する制御パスを作成します。その仮想アダプターでは、「Access external network」を選択しないでください。リンク集約を使用する場合は、ネットワーク・スイッチが IEEE 802.3ad に対応していることを確認する必要があります。仮想イーサネットの定義の詳細については、3.6、『仮想ネットワークへの VLAN の拡張』（68 ページ）を参照してください。

2.2.2 動的 LPAR の割り当て

動的 LPAR による割り当てとパーティション・プロファイルの更新には、計画が必要です。

動的 LPAR を使用して、I/O アダプター、メモリー、CPU、仮想アダプターの除去、変更、追加を行っても、そのような操作は、パーティションに定義されているパーティション・プロファイルに自動的には反映されません。したがって、再構成を実行した場合は、プロファイルを更新することをお勧めします。望ましいのは、最初に変更内容をパーティション・プロファイルに記述してから、その変更を動的に適用するという手順です。そうすれば、活動停止やシャットダウンが発生しても、変更内容が失われることはありません。一方、パーティション・プロファイルでパーティションに加えた変更は、**shutdown -Fr** などのリブートを実行しても LPAR には反映されません。変更がアクティブになるのは、「非活動（Not Activated）」の状態からのみです。

アクティブなパーティションの割り当てを保管する別の方法は、パーティションのコンテキスト・メニューから「Save」オプションを選択することです（図 2-3 を参照）。新しいプロファイルを保管するときには、20jun06_newnormal などのわかりやすい名前を選択します（図 2-4 を参照）。新しいパーティション・プロファイルをテストしたら、古いパーティション・プロファイルの名前を 21jun06_oldnormal などに変更し、新しいパーティション・プロファイル 20jun06_newnormal を normal に名前変更するかコピーします。新しいパーティション・プロファイルの変更点は、忘れずに文書化しておいてください。さらに、使用しなくなった古いパーティション・プロファイルを最新の保管内容からクリーンアップしたり、名前を変更したりすることもできます。

複数のワークロードがあって、複数のパーティション・プロファイルを必要とする場合は、パーティション・プロファイルのクリーンアップや名前変更を行いません。複数のパーティション・プロファイルをそのままアクティブにしてください。名前も、日付に基づく名前ではなく、その状況に当てはまるわかりやすい名前（DB2_high など）にします。

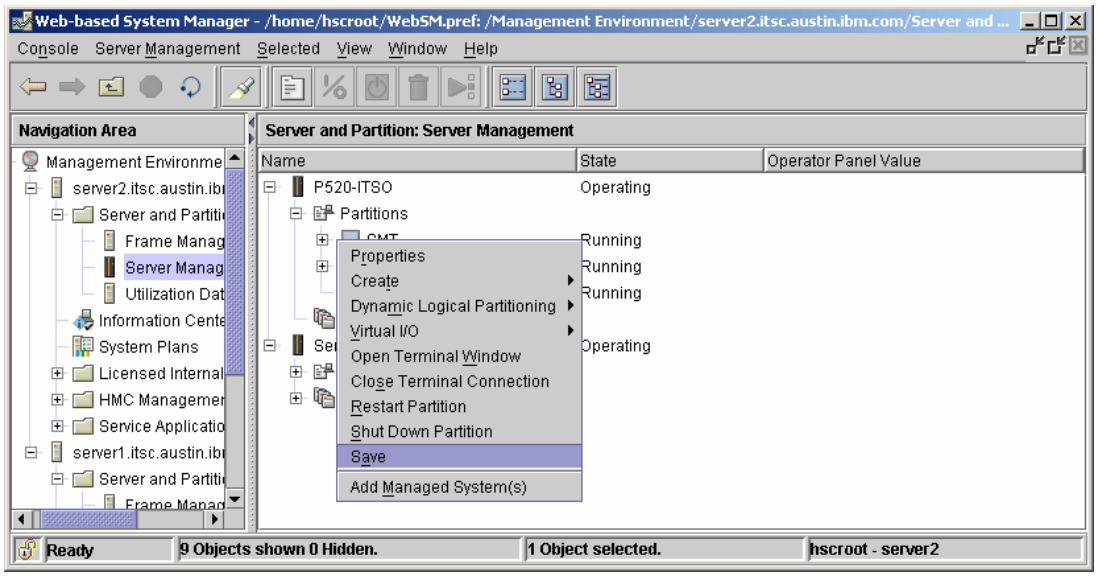


図 2-3　パーティションのコンテキスト・メニュー

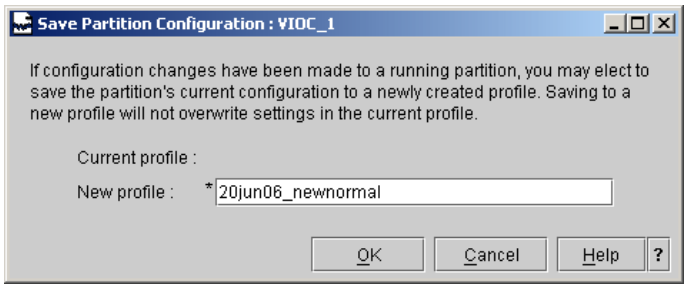


図 2-4　実行中のパーティション構成の保管

2.3 仮想 I/O サーバーでのジョブのスケジュール設定

cron を使用して重要な操作を自動化します。

Virtual I/O Server Version 1.3 には、cron ジョブの設定、編集、リスト、除去を実行するための **crontab** コマンドが用意されています。cron ジョブとは、cron デーモンによって定期的なスケジュール間隔で実行されるコマンドのことであり、例えば、システム・タスク、夜間のセキュリティー検査、分析レポート、バックアップなどが考えられます。

仮想 I/O サーバーでは、**crontab** コマンドに **-e** フラグを指定することによって、cron ジョブを設定できます。**crontab** コマンドで呼び出される編集セッションによって、padmin ユーザーの crontab ファイルを変更して、各 cron ジョブの項目をそのファイルに作成できます。

注 : ジョブのスケジュールを設定するときには、padmin ユーザーの crontab ファイルを使用します。他のユーザーの crontab ファイルを作成したり編集したりすることはできません。

項目を作成してそのファイルを終了すると、**crontab** コマンドがそのファイルを /var/spool/cron/crontabs ディレクトリーにコピーして、padmin ファイルに置き換えます。

crontab コマンドの構文は、以下のとおりです。

```
crontab [ -e padmin | -l padmin | -r padmin | -v padmin ]
```

-e padmin	padmin の crontab ファイルのコピーを編集します。編集が完了すると、そのファイルは、padmin の crontab ファイルとして crontab ディレクトリーにコピーされます。
-l padmin	padmin の crontab ファイルをリストします。
-r padmin	padmin の crontab ファイルを crontab ディレクトリーから除去します。
-v padmin	padmin の cron ジョブの状況をリストします。

2.4 パーティションの開始とシャットダウンの順序の自動化

システムの初期化時には、まず VIOS サービスを使用可能な状態にします。

仮想 I/O サーバーによって、AIX 5L または Linux のオペレーティング・システム・ベースのクライアントに仮想デバイスを提供している環境では、サーバーのシャットダウン時または開始時にその 2 種類のサーバーを区別する必要があります。AIX 5L または Linux のオペレーティング・システム・ベースのサーバーで仮想デバイスを使用している場合は、仮想 I/O サーバーを最初に開始するようにしてください。

仮想 I/O サーバーの始動プロセスを簡略化するには、システム・プロファイルに仮想 I/O サーバーを組み込んで、そのシステム・プロファイルを開始するようにします。そのシステム・プロファイルには、使用する論理パーティションとそれに関連したパーティション・プ

ロファイルを組み込みます。システム・プロファイルの詳細については、IBM Systems ハードウェア Information Center を参照してください。

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphbl/iphblmanagesprofile.htm

この総合的なソリューションを文書化しておくために、IBM System p5 サーバー全体を電源オフの状態から開始する方法についても取り上げます。そのための手順全体を自動化するには、HMC の SSH リモート・コマンド実行機能を使用します。

SSH を実行する中央の制御コンソールから HMC にリモート・コマンドを送信することによって、システムの電源オン、仮想 I/O サーバーのシステム・プロファイルの開始、AIX 5L または Linux のオペレーティング・システム・ベースのすべてのクライアントのシステム・プロファイルの開始のために必要なすべての操作を実行できます。その手順を自動化するには、管理コンソールと HMC の間で SSH 鍵の交換を実行しなければなりません。

そのためのプロセスについては、IBM Systems ハードウェア Information Center を参照してください。

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphai/settingupsecurescriptexecutionsbetweensshclientsandthehmc.htm

ヒント: SSH 鍵の交換を実行した場合は、コマンドをリモートから呼び出すことも可能です。例えば、**date** コマンドをリモートから実行して、HMC から日付の情報を出力するには、以下のコマンドを実行します。

```
# ssh hscroot@<hostname or IP address of HMC> date
Fri Jun 30 10:54:28 CDT 2006
```

SSH 鍵を交換していれば、パスワード・プロンプトが表示されなくなるので、スクリプトによってリモートからコマンドを実行できます。

この後のセクションでは、開始プロセスを自動化するための HMC コマンドの例を取り上げます。それらのコマンドは、本書の執筆時点では有効ですが、それぞれの環境でコマンドや構文が変更されていないかどうかを確認してください。

シャットダウン手順は、以下の手順の逆になります。

2.4.1 パーティション・スタンバイ・モードへの System p5 サーバーのリモート電源オン

System p5 サーバーのリモート電源オンを実行するには、HMC で **chsysstate** コマンドを使用します。システムの電源をオンにし、パーティション・スタンバイ・モードにするには、以下のコマンドを実行します（managed system name はサーバーの名前です）。

```
chsysstate -m <managed system name> -o onstandby -r sys
```

ヒント: サーバーの始動状況をモニターするには、**lsrefcode** コマンドを使用して、LED 状況コードを確認します。そのコマンドの例を以下に示します。

```
lsrefcode -r sys -m <managed system name> -F refcode
```

2.4.2 仮想 I/O サーバーのリモート開始

仮想 I/O サーバーのプロファイルを開始するには、図 2-5（43 ページ）にあるように、2 つのシステム・プロファイル（つまり、Virtual_IO_Servers と Virtual_IO_Client）をセットアップします。

仮想 I/O サーバーを開始するために、以下のようにして最初のシステム・プロファイルを呼び出します。

```
chsysstate -m <managed system name> -o on -r sysprof -n Virtual_IO_Servers
```

ヒント: 開始するパーティションの状態をモニターするには、以下のようにして **lsrefcode** コマンドを使用します。

```
lsrefcode -r lpar -m <managed system name> -F lpar_name,refcode
```

2.4.3 仮想デバイスのある論理パーティションのリモート開始

仮想 I/O サーバーを開始したら、2 番目のシステム・プロファイルを呼び出して、AIX 5L と Linux のオペレーティング・システム・ベースの論理パーティションをサーバー上で開始します。

```
chsysstate -m <managed system name> -o on -r sysprof -n Virtual_IO_Client
```

コマンドの実行後に HMC を調べると、図 2-5（43 ページ）のように、仮想 I/O サーバーとクライアントがすべて始動中になっているか、実行中になっているはずです。

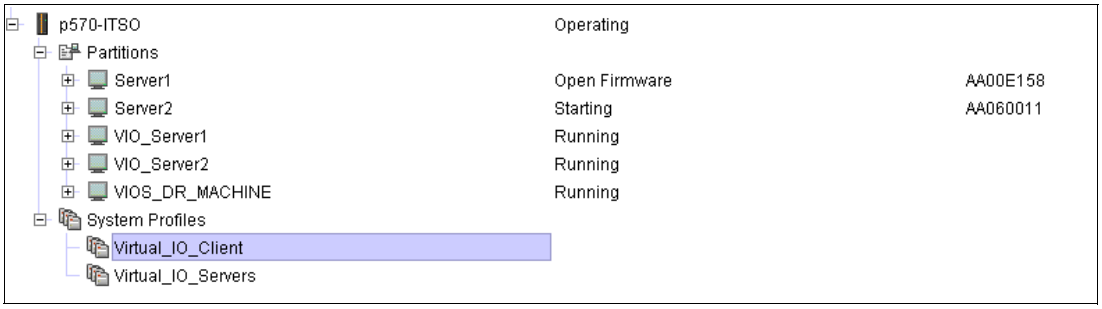


図 2-5　コマンド行から開始した仮想 I/O サーバーと AIX 5L または Linux のパーティション

ヒント: システム・プロファイルを使用しない場合の高度な解決策は、**lssyscfg** コマンドを使用して、lpar_env フィールドを確認することです。以下の例のようにすれば、管理対象サーバーに存在する仮想 I/O サーバーと AIX 5L または Linux の各オペレーティング・システム・ベースの論理パーティションのリストを作成できます。

```
hscroot@server1:~> lssyscfg -r lpar -m p570-ITSO  -F name,lpar_env
Server1,aixlinux
VIOS_DR_MACHINE,vioserver
VIO_Server1,vioserver
VIO_Server2,vioserver
Server2,aixlinux
```

2.4.4 自動化に関するまとめ

ここでは、始動手順の例を取り上げました。シャットダウン手順は始動手順の逆であり、AIX 5L または Linux のオペレーティング・システム・ベースのクライアントをまずシャットダウンしてから仮想 I/O サーバーをシャットダウンし、最後に System p5 サーバー・ハードウェアをシャットダウンするという流れになります。

2.5 仮想 I/O サーバーの保守

このセクションでは、仮想 I/O サーバーのアップグレードに関する 2 つのシナリオを取り上げます。まず、定期的な保守作業の実行中もクライアントが仮想 I/O リソースに継続的に接続できる環境を構築する場合は、二重仮想 I/O サーバー環境をお勧めします。一方、それほど重要ではない仮想リソースを使用するクライアントや、保守作業の時間帯に仮想 I/O サーバーをリブートできるような環境については、単一仮想 I/O サーバーのシナリオを取り上げます。

2.5.1 単一仮想 I/O サーバーのシナリオ

VIOS の保守作業では、クライアントについて検討することが必要です。

単一仮想 I/O サーバー環境でリブートを必要とするルーチン保守サービスを適用する場合は、その仮想 I/O サーバーのリソースを使用する仮想 I/O クライアントへの影響について検討する必要があります。仮想 I/O サーバーの更新やアップグレードによって、クリティカル・リソースに影響が及ばないことを確認してください。

ヒント : アップグレードまたは更新の前に、現在のバックアップが存在しなければ仮想 I/O サーバーと仮想 I/O クライアントのバックアップを作成し、仮想イーサネット・デバイスと仮想 SCSI デバイスを文書化しておきます。そのようにすれば、リカバリー・シナリオに費やす時間を短縮できます。

アップグレードや更新の実行中に複雑な問題が発生しないようにするために、仮想 I/O サーバーのアップグレードや更新を実行する前に環境をチェックすることをお勧めします。仮想 I/O クライアントと仮想 I/O サーバーに対して使用できる便利なコマンドの例を以下にまとめます。

lsvg rootvg	仮想 I/O サーバーと仮想 I/O クライアントで stale PP や stale PV がないかどうかを確認します。
lsvg -pv rootvg	仮想 I/O サーバーで、missing のディスクがないかを確認します。
netstat -cdlistats	仮想 I/O サーバーのすべての使用中のインターフェースでリンク状況が「Up」になっていることを確認します。
errpt	仮想 I/O クライアントで、CPU、メモリー、ディスク、イーサネットのエラーがないかどうかを確認し、エラーがあれば先に進む前に解決します。
lsvg -p rootvg	仮想 I/O クライアントで、missing のディスクがないかを確認します。
netstat -v	仮想 I/O クライアントのすべての使用中のインターフェースでリンク状況が「Up」になっていることを確認します。

アップグレードを開始する前に、現在のバックアップが存在しなければ、仮想 I/O サーバーと仮想 I/O クライアントのバックアップを作成します。仮想 I/O サーバーのバックアップには、**backupios** コマンドを使用します（2.1、『仮想 I/O サーバーのバックアップとリストア』（18 ページ）を参照）。仮想 I/O クライアントのバックアップには、**mksysb**、**savevg** の各コマンド、または IBM Tivoli® StorageManager などのバックアップ製品を使用します。

仮想 I/O サーバーを更新するには、以下の手順を実行します（この場合は、接続済みの光ディスク・ドライブを使用します）。

1. 仮想 I/O サーバーに接続している仮想 I/O クライアントをシャットダウンするか、使用中のすべての仮想リソースを使用不可にします。

2. **updateios** コマンドによって更新を適用します。更新を開始するために、y を押します。この場合は、光ディスク・ドライブからインストールしますが、**bffcreate** コマンドによって CD を NIM サーバーにコピーしてから、NFS を使用してそのファイル・システムを仮想 I/O サーバーにマウントすることもできます。

```
$ updateios -dev /dev/cd0 -install -accept

********************************************************************************
installp PREVIEW:  installation will not actually occur.
********************************************************************************

+-------------------------------------------------------------------------------+
                    Pre-installation Verification...
+-------------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

WARNINGS
--------
  Problems described in this section are not likely to be the source of any
  immediate or serious failures, but further actions may be necessary or
  desired.

  Already Installed
  -----------------
.
. (Lines omitted for clarity)
.
RESOURCES
---------
  Estimated system resource requirements for filesets being installed:
                (All sizes are in 512-byte blocks)
      Filesystem                    Needed Space             Free Space
      /usr                                7248                1098264
      -----                             --------                 ------
      TOTAL:                              7248                1098264

  NOTE:  "Needed Space" values are calculated from data available prior
  to installation.  These are the estimated resources required for the
  entire operation.  Further resource checks will be made during
  installation to verify that these initial estimates are sufficient.

********************************************************************************
End of installp PREVIEW.  No apply operation has actually occurred.
********************************************************************************

Continue the installation [y|n]?
```

3. 更新が終了したら、スタンバイ仮想 I/O サーバーをリブートします。

```
$ shutdown -restart

SHUTDOWN PROGRAM
Mon Nov 30 21:57:23 CST 1970

Wait for 'Rebooting...' before stopping.
Error reporting has stopped.
.
. (Lines omitted for clarity)
.
```

4. **ioslevel** コマンドによって新しいレベルをチェックします。
5. 仮想 I/O サーバーで、ディスク、イーサネット・アダプターなどをチェックします。
6. 仮想 I/O クライアントを開くか、仮想 I/O リソースを再接続して、仮想 I/O クライアントをチェックします。

仮想 I/O サーバー環境を検証して、更新内容を文書化しておきます。

2.5.2 二重仮想 I/O サーバーのシナリオ

二重仮想 I/O サーバー環境で仮想 I/O サーバーに更新を適用する場合は、仮想 I/O サービスのダウン時間を発生させずにその作業を実行できます。Virtual I/O Server を Version 1.1 から 1.2 以降にアップグレードする場合は、それと同時にクライアントで新しい機能に移行することも可能であり、例えば、Network Interface Backup から共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）に移行することなどが可能ですが、その際には、仮想ネットワークのセットアップの変更に伴って仮想 I/O クライアントにどのような影響が及ぶかを計画しておくことが必要です。もちろん、Network Interface Backup から共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）への移行は必須ではなく、新しい機能の使用例にすぎません。

ヒント：アップグレードまたは更新の前に、現在のバックアップが存在しなければ仮想 I/O サーバーと仮想 I/O クライアントのバックアップを作成し、仮想イーサネット・デバイスと仮想 SCSI デバイスを文書化しておきます。そのようにすれば、リカバリー・シナリオに費やす時間を短縮できます。

アップグレードを実行する前に、問題が存在しないかどうかを確認します。

二重仮想 I/O サーバー環境で Virtual I/O Server を更新する場合も、Version 1.2 から 1.3 への移行や、既存の Virtual I/O Server への更新の適用については、手順に違いはありません。いずれかの仮想 I/O サーバーの更新を開始する前に、仮想 I/O サーバーと仮想 I/O クライアントの仮想イーサネットやディスク・デバイスをチェックするのは、ベスト・プラクティスです。また、物理アダプターをチェックして、接続を確認してください。例 2-13、例 2-14、例 2-15（47 ページ）では、すべての仮想アダプターが稼働中になっています。

例 2-13　仮想 I/O クライアントで実行した netstat -v コマンド

```
netstat -v
.
. (Lines omitted for clarity)
.
Virtual I/O Ethernet Adapter (l-lan) Specific Statistics:
---------------------------------------------------------
RQ Length: 4481
No Copy Buffers: 0
Filter MCast Mode: False
Filters: 255
  Enabled: 1  Queued: 0  Overflow: 0
```

LAN State: Operational

```
Hypervisor Send Failures: 0
  Receiver Failures: 0
  Send Errors: 0

Hypervisor Receive Failures: 0

ILLAN Attributes: 0000000000003002 [0000000000002000]
.
. (Lines omitted for clarity)
.
#
```

例 2-14　仮想 I/O サーバー 1 で実行した netstat -cdlistats コマンド

```
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
Virtual I/O Ethernet Adapter (l-lan) Specific Statistics:
---------------------------------------------------------
RQ Length: 4481
No Copy Buffers: 0
Trunk Adapter: True
  Priority: 1  Active: True
Filter MCast Mode: False
Filters: 255
  Enabled: 1  Queued: 0  Overflow: 0
LAN State: Operational
.
. (Lines omitted for clarity)
.
$
```

例 2-15　仮想 I/O サーバー 2 で実行した netstat -cdlistats コマンド

```
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
Virtual I/O Ethernet Adapter (l-lan) Specific Statistics:
---------------------------------------------------------
RQ Length: 4481
No Copy Buffers: 0
Trunk Adapter: True
  Priority: 2  Active: False
Filter MCast Mode: False
Filters: 255
  Enabled: 1  Queued: 0  Overflow: 0
LAN State: Operational
.
. (Lines omitted for clarity)
.
$
```

さらに調べていくと、物理アダプター（例 2-16 と例 2-17）がダウンしており、接続が存在していません。これがバックアップ回線などであれば、システム管理者も気付いていない可能性があります。物理アダプターをチェックしてください。

例 2-16　仮想 I/O サーバー 1 で実行した netstat -cdlistats コマンド

```
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902) Specific Statistics:
---------------------------------------------------------------------
Link Status : Down
Media Speed Selected: Auto negotiation
.
. (Lines omitted for clarity)
.
$
```

例 2-17　仮想 I/O サーバー 2 で実行した netstat -cdlistats コマンド

```
$ netstat -cdlistat
.
. (Lines omitted for clarity)
.
10/100/1000 Base-TX PCI-X Adapter (14106902) Specific Statistics:
---------------------------------------------------------------
Link Status : Down
Media Speed Selected: Auto negotiation
.
. (Lines omitted for clarity)
.
$
```

ディスク状況のチェック方法は、仮想 I/O サーバーによるディスクの共用方法によって異なります。

図 2-6 のような MPIO セットアップの場合は、ディスク・パス状況を確認するために、最初の仮想 I/O サーバーの更新時の前と後に以下のコマンドを実行します。

lspath　仮想 I/O クライアントで、ディスクに対するすべてのパスをチェックします。すべてのパスが有効状態になっている必要があります。

lsattr -El hdisk0　仮想 I/O クライアントで、hdisk0 の MPIO ハートビートをチェックし、属性 hcheck_mode が nonactive に、属性 hcheck_interval が 60 にそれぞれ設定されているかどうかを確認します。IBM SAN ストレージを使用している場合は、reserve_policy が no_reserve になっていることを確認します。他のストレージ・ベンダー製品の場合には、reserve_policy を別の値に設定しなければならない場合があります。このコマンドは、仮想 I/O サーバーから共用されているすべてのディスクに対して実行しなければなりません。

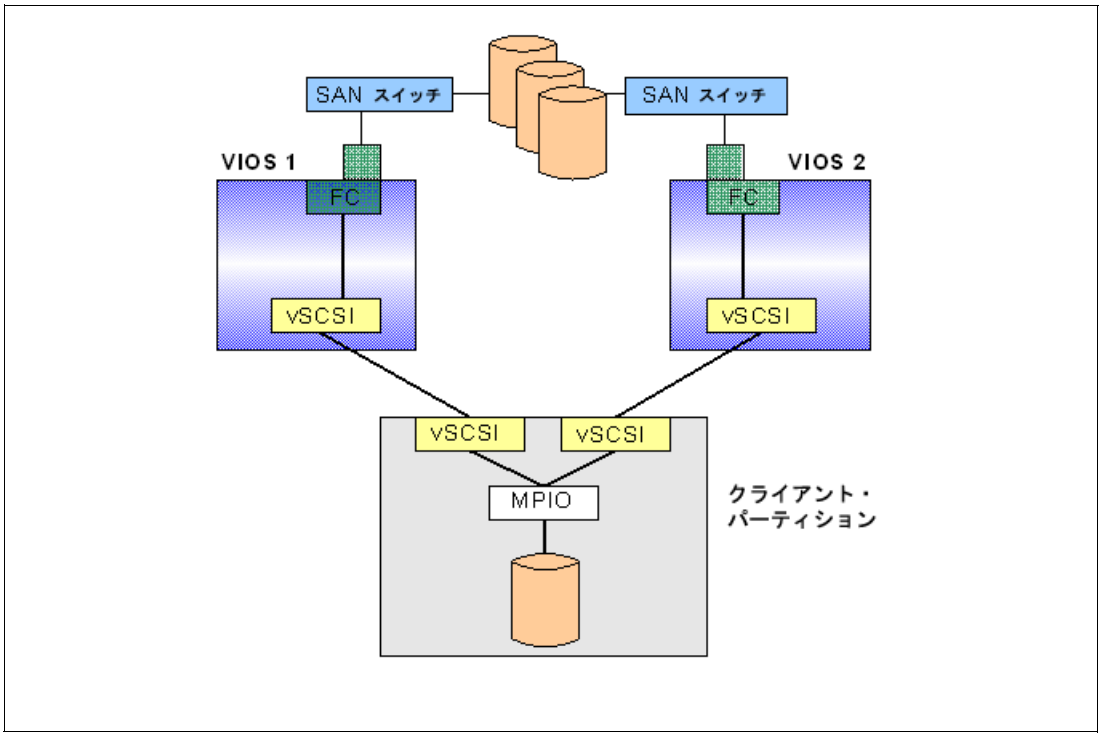


図 2-6 クライアントの MPIO

図 2-7（49 ページ）のような LVM ディスク環境の場合は、以下のコマンドを実行して、仮想 I/O サーバーから共用されているディスクの LVM 状況をチェックします。

lsvg rootvg 仮想 I/O クライアントで、stale PP がないかどうかを確認します。クォーラムはオフになっている必要があります。

lsvg -p rootvg 仮想 I/O クライアントで、hdisk が missing でないことを確認します。

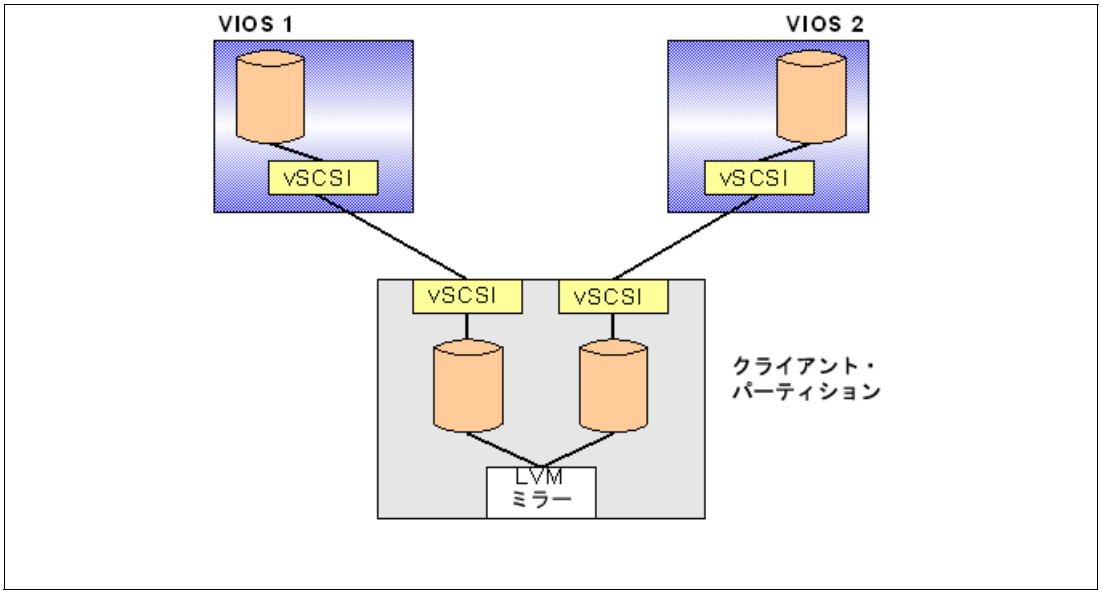


図 2-7 クライアントの LVM ミラーリング

環境をチェックして問題点を解決した時点で、現在のバックアップが存在しなければ、仮想 I/O サーバーと仮想 I/O クライアントのバックアップを作成します。

仮想 I/O サーバーの更新またはアップグレードを実行するには、以下のようにします。

1. スタンバイ仮想 I/O サーバーを見つけて、**netstat** コマンドを入力します。出力の末尾で、共用イーサネット・アダプター（Shared Ethernet Adapter）の優先順位を見つけて、アクティブになっているかどうかを確認します。この場合は、スタンバイ・アダプターがアクティブではないので、そのサーバーのアップグレードを開始できます。

```
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
Trunk Adapter: True
  Priority: 2  Active: False
Filter MCast Mode: False
Filters: 255
  Enabled: 1  Queued: 0  Overflow: 0
LAN State: Operational
.
. (Lines omitted for clarity)
.
$
```

2. **updateios** コマンドによって更新を適用し、更新を開始するために y を押します。この場合は、光ディスク・ドライブからインストールしますが、**bffcreate** コマンドによって更新を NIM サーバーにコピーしてから、そのファイル・システムを仮想 I/O サーバーにマウントすることもできます。

```
$ updateios -dev /dev/cd0 -install -accept

********************************************************************************
installp PREVIEW:  installation will not actually occur.
********************************************************************************

+-------------------------------------------------------------------------------+
                    Pre-installation Verification...
+-------------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

WARNINGS
--------
  Problems described in this section are not likely to be the source of any
  immediate or serious failures, but further actions may be necessary or
  desired.

  Already Installed
  -----------------
.
. (Lines omitted for clarity)
.
RESOURCES
---------
  Estimated system resource requirements for filesets being installed:
                (All sizes are in 512-byte blocks)
      Filesystem                    Needed Space            Free Space
      /usr                               7248               1098264
      -----                           --------               ------
      TOTAL:                             7248               1098264

  NOTE:  "Needed Space" values are calculated from data available prior
  to installation.  These are the estimated resources required for the
```

```
  entire operation.  Further resource checks will be made during
  installation to verify that these initial estimates are sufficient.

********************************************************************************
End of installp PREVIEW.  No apply operation has actually occurred.
********************************************************************************

Continue the installation [y|n]?
```

3. 更新が完了したら、スタンバイ仮想 I/O サーバーをリブートします。

```
$ shutdown -restart

SHUTDOWN PROGRAM
Mon Nov 30 21:57:23 CST 1970

Wait for 'Rebooting...' before stopping.
Error reporting has stopped.
.
. (Lines omitted for clarity)
```

4. **ioslevel** コマンドによって新しいレベルをチェックします。
5. スタンバイ仮想 I/O サーバーと仮想 I/O クライアントが仮想 I/O サーバー環境に接続していることを確認します。図 2-6（49 ページ）のような MPIO 環境の場合は、仮想 I/O クライアントで **lspath** コマンドを実行して、すべてのパスが使用可能になっていることを確認します。図 2-7（49 ページ）のような LVM 環境の場合は、**varyonvg** コマンドを実行する必要があります。ボリューム・グループの同期が開始されるはずです。開始されない場合は、仮想 I/O サーバー環境からの仮想ディスクを使用するボリューム・グループに対して **syncvg -v** コマンドを実行し、各ボリューム・グループを同期します。

```
# lsvg -p rootvg
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk0            active            511         488         102..94..88..102..102
hdisk1            missing           511         488         102..94..88..102..102
# varyonvg rootvg
# lsvg -p rootvg
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk0            active            511         488         102..94..88..102..102
hdisk1            active            511         488         102..94..88..102..102
# lsvg rootvg
VOLUME GROUP:       rootvg                   VG IDENTIFIER:  00c478de00004c00000
00006b8b6c15e
VG STATE:           active                   PP SIZE:        64 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      1022 (65408 megabytes)
MAX LVs:            256                      FREE PPs:       976 (62464 megabytes)
LVs:                9                        USED PPs:       46 (2944 megabytes)
OPEN LVs:           8                        QUORUM:         1
TOTAL PVs:          2                        VG DESCRIPTORS: 3
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         2                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
#
```

6. 仮想 I/O サーバーで **netstat -cdlistats** コマンドを使用して、仮想 I/O サーバーに接続しているイーサネット・サービスが共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）のシナリオを使用していることを確認し、仮想 I/O ク

ライアントで **netstat -v** コマンドを使用して、Network Interface Backup を使用していることを確認します。Network Interface Backup アダプターのリンク状況もチェックしてください。

7. 共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）を使用している場合は、**chdev** コマンドを使用して、仮想 I/O サーバーに対するスタンバイ接続と基本接続を変更し、**netstat -cdlistats** コマンドを使用して、以下の例のように状態が変更されていることを確認します。

```
$ chdev -dev ent4 -attr ha_mode=standby
ent4 changed
$ netstat -cdlistats
.
. (Lines omitted for clarity)
.
Trunk Adapter: True
  Priority: 1  Active: False
Filter MCast Mode: False
.
. (Lines omitted for clarity)
.
$
```

8. **updateios** コマンドを使用して、現時点でスタンバイ仮想 I/O サーバーになっている仮想 I/O サーバーに更新を適用します。
9. **shutdown -restart** コマンドによって仮想 I/O サーバーをリブートします。
10. **ioslevel** コマンドによって新しいレベルをチェックします。
11. スタンバイ仮想 I/O サーバーと仮想 I/O クライアントが仮想 I/O サーバー環境に接続していることを確認します。図 2-6（49 ページ）のような MPIO 環境の場合は、仮想 I/O クライアントで **lspath** コマンドを実行して、すべてのパスが使用可能になっていることを確認します。図 2-7（49 ページ）のような LVM 環境の場合は、**varyonvg** コマンドを実行します。ボリューム・グループの同期が開始されるはずです。開始されない場合は、仮想 I/O サーバー環境からの仮想ディスクを使用する各ボリューム・グループに対して **syncvg -v** コマンドを実行します。
12. **netstat -cdlistats** コマンドを使用して、イーサネットが仮想 I/O サーバーに接続していることを確認し、**netstat -v** コマンドを使用して、リンク状況を確認します。
13. 以下の例のように **chdev** コマンドを使用して、この仮想 I/O サーバーの役割をプライマリーに戻します。

```
$ chdev -dev ent4 -attr ha_mode=auto
ent4 changed
$
```

これで、更新が完了しました。

3

ネットワーキング

仮想環境でのネットワーク接続は、非常にフレキシブルです。この章では、仮想ネットワーク構成のベスト・プラクティスを取り上げます。具体的には、共用イーサネット・アダプター（Shared Ethernet Adapter）構成、高可用性シナリオ、VLAN タグ、セキュリティー・ゾーン、パフォーマンスを最適化するためのパケット・サイズの調整などについて説明します。

3.1 仮想 I/O サーバーの IP アドレス

仮想 I/O サーバーのネットワーク接続を確認します。

仮想 I/O サーバーには、HMC からアクセス可能です。セキュアな専用 HMC を使用してサービス・プロセッサー・ネットワークに接続し、コンソール・セッションを開始することができます。その場合、仮想 I/O サーバーで管理用の専用ネットワーク・アドレスを構成するかどうかはオプションになります。ただし、仮想 I/O サーバーがどのネットワークにも表示されない場合は、仮想 I/O サーバーに接続する方法がないので、仮想 I/O サーバーの動的リソース割り振りが有効になりません。その構成作業では、HMC の mkvterm パネルと vtmenu パネルが有用です。

3.1.1 仮想 I/O サーバーの複数の IP アドレス

通常、仮想 I/O サーバーで必要なのは、管理 VLAN で構成する 1 つの IP アドレス（HMC からアクセスできる IP アドレス）だけです。HMC と仮想 I/O サーバーが通信できなければ、動的リソース割り振りは有効にならないので、その接続は重要です。

高可用性のために、複数の IP アドレスが必要になる場合もあります。そのような状況は、仮想 I/O サーバーで共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）と Network Interface Backup（NIB）をセットアップする作業に関連しています。

共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）環境の IP アドレス

仮想 I/O サーバーで共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）を使用していても、ネットワーク障害の種類によっては、共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）が起動しない場合があります。キープアライブ・メッセージは、コントロール・チャネルだけで送信されるからです。つまり、キープアライブ・メッセージは、他の SEA ネットワークや外部ネットワークでは送信されません。それでも、SEA のフェイルオーバー機能を構成して、特定の IP アドレスとの疎通を周期的に確認することができます。SEA は、その IP アドレスに周期的に ping を実行することによって、特定のネットワーク障害を検出できます。その場合、共用イーサネット・アダプター（Shared Ethernet Adapter）がその周期的な自己診断機能を実行するには、IP アドレスが関連付けられているネットワーク・インターフェースが必要になります。

これらの IP アドレスは固有でなければならず、各共用イーサネット・アダプター（Shared Ethernet Adapter）ごとに別々の IP アドレスを使用しなければなりません。特定の IP アドレスに対して ping を実行したときに、ICMP-Echo-Requests を送信して ICMP-Echo-Replies を受信するための返信先アドレスを用意するには、共用イーサネット・アダプター（Shared Ethernet Adapter）に IP アドレスが必要になります。図 3-1（55 ページ）は、そのオプション機能を構成する方法を示した例です。共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）機能全体を構成する方法の詳細については、3.9、『SEA のフェイルオーバーをサポートするための共用イーサネット・アダプター（Shared Ethernet Adapter）の作成』（82 ページ）を参照してください。

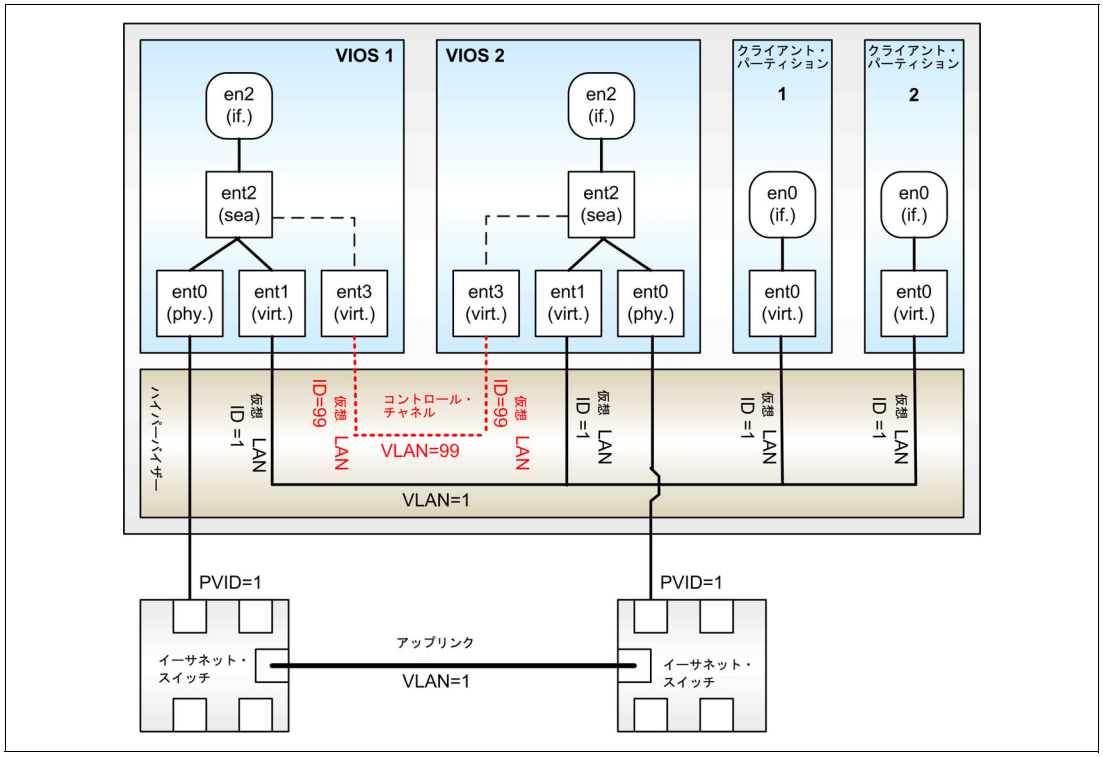


図 3-1 SEA のフェイルオーバーの例

図 3-1 に構成例を示します。この後の説明文では、そのオプション機能を構成する方法の例を取り上げます。

1. 各仮想 I/O サーバーで、**mkvdev -sea** コマンドを使用して、共用イーサネット・アダプター（Shared Ethernet Adapter）を構成します。netaddr 属性を使用するのは、共用イーサネット・アダプター（Shared Ethernet Adapter）で特定の IP アドレスとの通信が可能かどうかを周期的に確認するためです。

   ```
   $ mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 1 -attr ha_mode=auto
   ctl_chan=ent3 netaddr=9.3.5.41
   ent2 Available
   en2
   et2
   ```

2. **lsdev** コマンドを使用して、netaddr 属性に IP アドレスが構成されているかどうかを確認します。

   ```
   $ lsdev -dev ent2 -attr
   attribute     value    description                                          user_settable

   ctl_chan      ent3     Control Channel adapter for SEA failover                    True
   ha_mode       auto     High Availability Mode                                      True
   large_send    0        Enable HardwareTransmit TCP Resegmentation                  True
   netaddr       9.3.5.41 Address to ping                                             True
   pvid          1        PVID to use for the SEA device                              True
   pvid_adapter  ent1     Default virtual adapter to use for non-VLAN-tagged packets  True
   real_adapter  ent0     Physical adapter associated with the SEA                    True
   thread        1        Thread mode enabled (1) or disabled (0)                     True
   virt_adapters ent1     List of virtual adapters associated with the SEA (comma separated) True
   ```

3. 次に、**mktcpip** コマンドを使用して、共用イーサネット・アダプター（Shared Ethernet Adapter）（en2）で IP アドレスを構成します。

Network Interface Backup を使用する場合の IP アドレス

仮想環境でネットワークの冗長性を確保するために、二重仮想 I/O サーバーを使用して、クライアント・パーティションで Network Interface Backup（NIB）機能を構成できます。この機能を仮想 I/O サーバー自体で実装して、ネットワーク・スイッチが使用不能になった場合でも、ネットワークの冗長性を強化することも可能です。図 3-2 は、単一仮想 I/O サーバーを使用した構成例です。

注 : 2 つの仮想 I/O サーバーを別々のスイッチに接続している場合は、NIB が不要になることもあります。スイッチの障害に対する保護機能は、仮想 I/O クライアントの SEA のフェイルオーバーのメカニズムによって用意することも可能だからです。一方、二重仮想 I/O サーバーを実行している場合でも、そのいずれかを保守や交換のために操作状態から外す場合は、残りの仮想 I/O サーバーを使用して NIB による冗長性を確保できます。NIB と SEA は、組み合わせて使用することが可能であり、相互に排他的な関係になっているわけではありません。

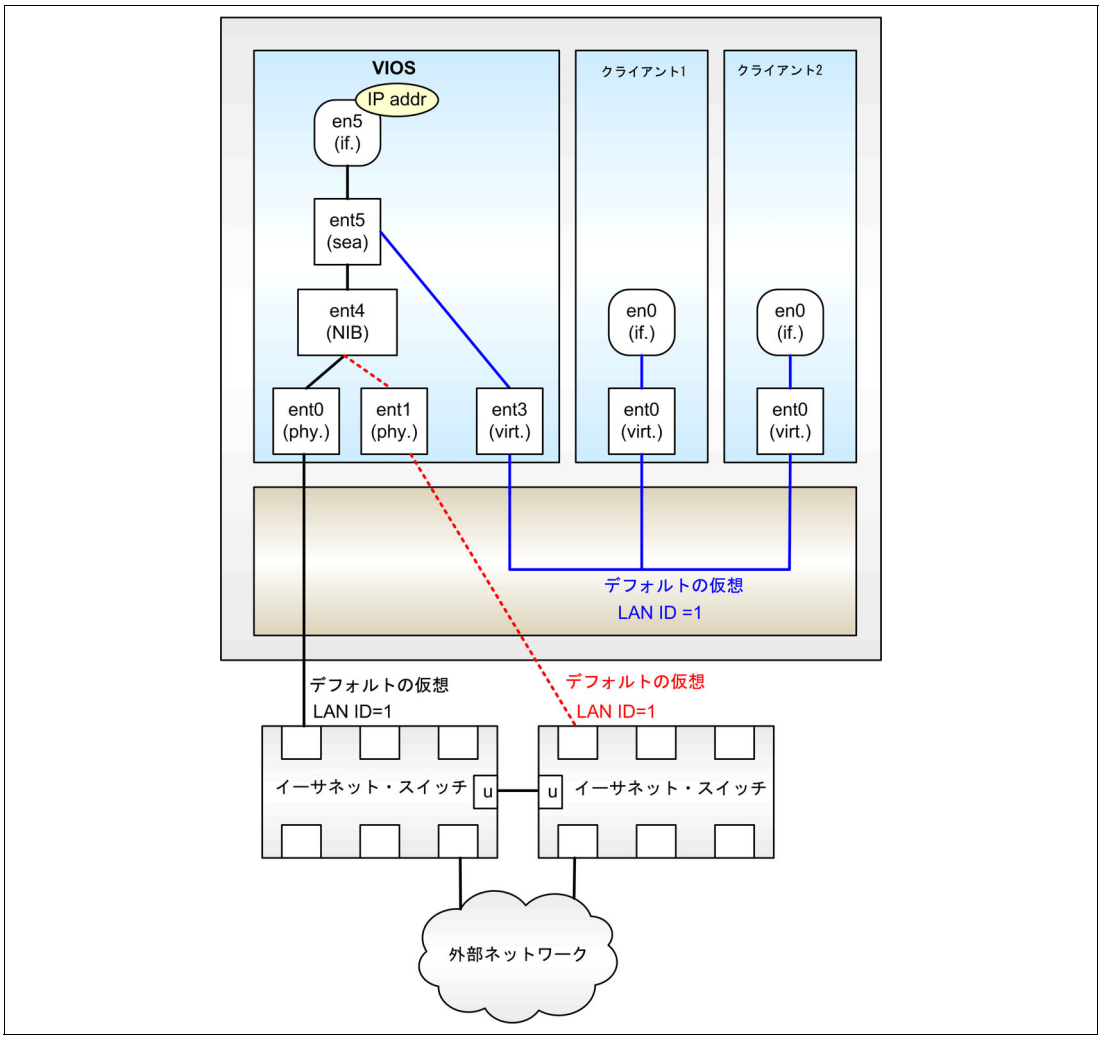


図 3-2　単一仮想 I/O サーバーでの NIB 構成

図 3-2（56 ページ）に構成例を示します。この後の説明文では、単一仮想 I/O サーバーを使用した構成例を取り上げます。

1. **mkvdev** コマンドを使用して、仮想 I/O サーバーで Network Interface Backup デバイスを構成します。この例では、ent0 がプライマリー・アダプター、ent1 がバックアップ・アダプターです。さらに、ping 対象のインターネット・アドレスとして netaddr 属性を使用しています。通常は、netaddr 属性として、デフォルト・ゲートウェイを使用します。SEA 経由で有効な経路指定を使用して、そのアドレスとの通信が可能であることを確認してください。

```
$ mkvdev -lnagg ent0 -attr backup_adapter=ent1 netaddr=9.3.5.198
ent4 Available
en4
et4
```

注 : クライアント・パーティションでいずれかのネットワーク・インターフェース（en0 または en1）がすでに NIM インストール（またはクローン）の一部として構成されている場合は、そのインターフェースを取り外すだけでなく削除することが必要です。そうすれば、Network Interface Backup インターフェースの構成前にそのインターフェースが構成されているという状態を回避できます。

2. **lsdev** コマンドを使用して、Network Interface Backup デバイスの構成を確認します。

```
$ lsdev -dev ent4 -attr
attribute       value          description                                user_settable

adapter_names   ent0           EtherChannel Adapters                      True
alt_addr        0x000000000000 Alternate EtherChannel Address             True
auto_recovery   yes            Enable automatic recovery after failover   True
backup_adapter  ent1           Adapter used when whole channel fails      True
hash_mode       default        Determines how outgoing adapter is chosen  True
mode            standard       EtherChannel mode of operation             True
netaddr         9.3.5.198      Address to ping                            True
num_retries     3              Times to retry ping before failing         True
retry_time      1              Wait time (in seconds) between pings       True
use_alt_addr    no             Enable Alternate EtherChannel Address      True
use_jumbo_frame no             Enable Gigabit Ethernet Jumbo Frames       True
```

3. Network Interface Backup デバイスを物理アダプターとして使用して、共用イーサネット・アダプター（Shared Ethernet Adapter）を作成します。

```
$ mkvdev -sea ent4 -vadapter ent3 -default ent3 -defaultid 1
ent5 Available
en5
et5
```

4. **lsdev** コマンドを使用して、SEA デバイスの作成を確認します。

```
$ lsdev -type adapter
name            status                                      description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  Virtual I/O Ethernet Adapter (l-lan)
ent3            Available  Virtual I/O Ethernet Adapter (l-lan)
ent4            Available  EtherChannel / IEEE 802.3ad Link Aggregation
ent5            Available  Shared Ethernet Adapter
ide0            Available  ATA/IDE Controller Device
sisioa0         Available  PCI-X Dual Channel U320 SCSI RAID Adapter
vhost0          Available  Virtual SCSI Server Adapter
vhost1          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
```

5. **mktcpip** コマンドを使用して、新しく作成した SEA で IP アドレスを構成します。

```
$ mktcpip -hostname lpar02 -inetaddr 9.3.5.112 -interface en5 -netmask 255.255.255.0
-gateway 9.3.5.41 -nsrvaddr 9.3.4.2 -nsrvdomain itsc.austin.ibm.com
```

6. **netstat** コマンドを使用して、共用イーサネット・アダプター（Shared Ethernet Adapter）の IP アドレスを確認します。

```
$ netstat -num -state
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en5   1500  link#5      0.2.55.2f.a7.e        3936     0     1181     0     0
en5   1500  9.3.5       9.3.5.112             3936     0     1181     0     0
lo0   16896 link#1                           14370     0    14567     0     0
lo0   16896 127         127.0.0.1           14370     0    14567     0     0
lo0   16896 ::1                           14370     0    14567     0     0
```

3.1.2 仮想 I/O サーバーで IP アドレスを構成するべき場所

仮想 I/O サーバーのどこに IP アドレスを構成するべきかについては、共用イーサネット・アダプター（Shared Ethernet Adapter）自体に IP アドレスを構成する方式をお勧めします。そのようにすれば、仮想 I/O サーバーへのネットワーク接続が内部の仮想ネットワーク構成に依存しなくなるばかりか、共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）と Network Interface Backup を実装した仮想 I/O サーバー環境を構築する作業も簡略化されます。そのような環境では、リモート・アドレスに対する ping 機能が必要になるからです。

このような事情を踏まえて、仮想 I/O サーバーの共用イーサネット・アダプター（Shared Ethernet Adapter）で IP アドレスを構成する標準方式をお勧めします。

注 : インストール済み環境によっては、共用イーサネット・アダプター（Shared Ethernet Adapter）以外で IP アドレスを構成しなければならない状況もあり得ます。仮想イーサネットのパフォーマンスの問題が発生する場合は、IBM サポート・サービスに問い合わせてください。

3.2 IP アドレスまたは VLAN の変更

このセクションでは、仮想 I/O 環境で IP アドレスと VLAN を変更する方法を説明し、仮想 I/O サーバーと仮想 I/O クライアントに対する影響について取り上げます。

3.2.1 仮想 I/O サーバー

ここからは、Virtual I/O Server に関連したセクションです。

IP アドレスの変更

共用イーサネット・アダプター（Shared Ethernet Adapter）の IP アドレスには、以下の用途があります。

- 共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）機能のためのハートビート
- 仮想 I/O サーバーで動的 LPAR を使用するための RMC 通信
- 仮想 I/O サーバーのログオン・プロセス
- NIM による仮想 I/O サーバーのインストールまたはリストア（**installios** コマンド）
- **topas** コマンドによるパフォーマンス・モニター
- NIM サーバーからの仮想 I/O サーバーの更新またはアップグレード
- NIM サーバーまたは他のネットワーク・サーバーへの仮想 I/O サーバーのバックアップ

共用イーサネット・アダプター（Shared Ethernet Adapter）に割り当てる IP アドレスは、仮想 I/O クライアントには何の影響も及ぼしません。したがって、共用イーサネット・アダプター（Shared Ethernet Adapter）デバイスの IP アドレスを変更しても、共用イーサネット・アダプター（Shared Ethernet Adapter）デバイスを使用する仮想 I/O クライアントは影響を受けません。

en4 の IP アドレスを 9.3.5.112 から 9.3.5.113 に、ホスト名を lpar02 から lpar03 に変更しなければならない場合は、以下のコマンドを使用します。

```
mktcpip -hostname lpar03 -inetaddr 9.3.5.113 -interface en4
```

アダプターも同時に変更する場合（例えば、en4 から en8 に変更する場合）は、まず en4 で **rmtcpip** コマンドを実行して TCP/IP 定義を削除してから、en8 で **mktcpip** コマンドを実行する必要があります。

VLAN の変更

動的 LPAR を使用すれば、仮想 I/O サーバーで実行されているサービスを中断することなく、既存のタグ付き仮想イーサネット・アダプターで VLAN の追加や削除を実行できます。その場合は、パーティションのコンテキスト・メニューの「Save」オプションまたは「Properties」オプションを使用して、動的な変更をパーティション・プロファイルに反映する必要があります。

新しいタグ付きの VLAN ID のブリッジングを開始するには、まず一時的な PVID といずれかのタグ付き VID を使用して、新しい仮想イーサネット・アダプターを作成します。その後、動的 LPAR を使用してその新しいアダプターを仮想 I/O サーバーに移動し、**chdev** コマンドを使用して SEA の virt_adapters リストに追加します。新しい VLAN ID のブリッジングが中断なしですぐに始まります。

新しい物理イーサネット・アダプター、新しい共用イーサネット・アダプター（Shared Ethernet Adapter）、新しい仮想イーサネット・アダプターを追加して、タグ付き仮想イーサネット・アダプターを作成することもできます。その場合は、タグなしの仮想イーサネット・アダプターからタグ付きの仮想イーサネット・アダプターに切り替えることも可能で

す。ただしそのためには、仮想 I/O クライアントをタグなしのアダプターからタグ付きのアダプターに切り替えるときに、非仮想環境で IP アドレスを変更する場合に必要になるような小規模な保守の時間帯を計画しておく必要があります。

この要件は、タグ付きの仮想イーサネット・アダプターからタグなしの仮想イーサネット・アダプターに切り替えるときにも当てはまります。タグなしの仮想イーサネット・アダプターとタグ付きの仮想イーサネット・アダプターの間の切り替えについては、計画を策定し、変更内容を文書化しておくことをお勧めします。

3.2.2 仮想 I/O クライアント

ここからは、仮想 I/O クライアントに関連したセクションです。

IP アドレスの変更

仮想 I/O クライアントの仮想イーサネット・アダプターの IP アドレスを変更するには、SMIT または **mktcpip** コマンドを使用します。この例では、IP アドレスを 9.3.5.113 から 9.3.5.112 に、ホスト名を lpar03 から lpar02 にそれぞれ変更します。仮想イーサネット・アダプターを変更する方法は、物理アダプターを変更する方法と同じであり、以下のコマンドを使用します。

```
mktcpip -h lpar02 -a 9.3.5.112 -i en0
```

VLAN の変更

仮想 I/O サーバーで VLAN 情報を変更する場合は、動的 LPAR を使用することによって、仮想 I/O クライアントのために実行されているネットワーク・サービスを中断しないで、既存のタグ付きの仮想イーサネット・アダプターで VLAN の追加や削除を実行できます。仮想 I/O クライアントで IP アドレスを追加する場合は、別名 IP アドレスとしてその作業を完了することにより、その仮想 I/O クライアントに対するネットワーク・サービスを継続できます。動的 LPAR の変更作業では、パーティションのコンテキスト・メニューの「Save」オプションまたは「Properties」オプションを使用して、パーティション・プロファイルを変更することを忘れないようにする必要があります。

仮想 I/O クライアントの場合は、ネットワーク・サービスを中断しないでタグなしの仮想イーサネット・アダプターからタグ付きの仮想イーサネット・アダプターに切り替えることができません。ただし、新しい仮想イーサネット・アダプターを追加して、そのアダプターをタグ付きの仮想イーサネット・アダプターにすることはできます。そのようにすれば、タグなしの仮想イーサネット・アダプターからタグ付きの仮想イーサネット・アダプターに切り替えることが可能になりますが、仮想 I/O クライアントをタグなしのアダプターからタグ付きのアダプターに切り替えるときに、小規模な保守の時間帯を計画しておく必要があります。このような中断は、非仮想環境で IP アドレスを変更する場合にも必要になります。

この要件は、タグ付きの仮想イーサネット・アダプターからタグなしの仮想イーサネット・アダプターに切り替えるときにも当てはまります。タグなしの仮想イーサネット・アダプターとタグ付きの仮想イーサネット・アダプターの間の切り替えについては、計画を策定し、変更内容を文書化しておくことをお勧めします。

3.3 ネットワーク・デバイスのマッピングの管理

仮想環境を管理するための重要な要素の 1 つは、仮想オブジェクトと物理オブジェクトの対応関係を追跡管理することです。ネットワークの領域では、物理ネットワーク・アダプターと仮想ネットワーク・アダプターとの関係、さらには複数のホストやスイッチと VLAN との関係がかかわってきます。このマッピングは、パフォーマンスを管理するためにも、ハードウェアの保守によってどのシステムが影響を受けるのかを把握するためにも重要です。

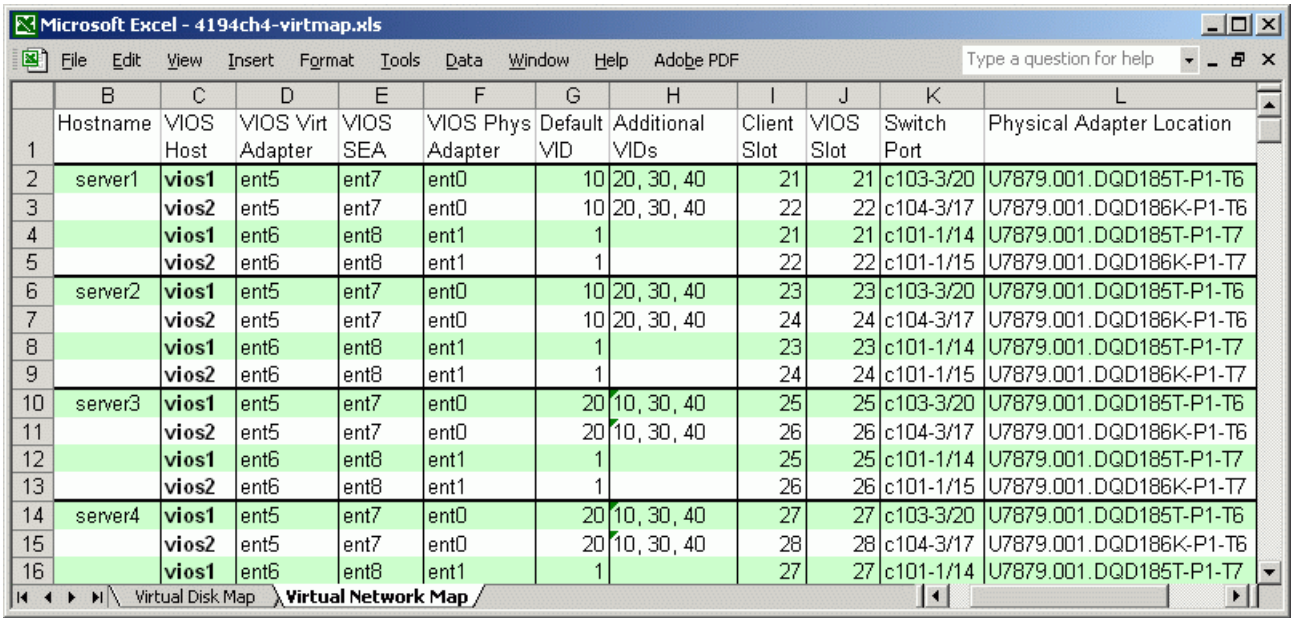
ストレージ・デバイスのマッピングの詳細については、4.3、『LUN と VSCSI と hdisk の対応関係の管理』（101 ページ）を参照してください。

ネットワーク接続の冗長性を必要とする環境については、このセクションでは、Network Interface Backup による方式よりも、SEA フェイルオーバーによる方式を優先して重点的に取り上げます。

802.1Q のタグ付きの VLAN を使用するかどうかによっては、以下の情報の追跡管理が必要になる場合があります。

- 仮想 I/O サーバー
  - サーバーのホスト名
  - 物理アダプターのデバイス名
  - スイッチ・ポート
  - SEA のデバイス名
  - 仮想アダプターのデバイス名
  - 仮想アダプターのスロット番号
  - ポートの VLAN ID（タグ付きとタグなしの両方の使用法）
  - 追加の VLAN ID
- 仮想 I/O クライアント
  - クライアントのホスト名
  - 仮想アダプターのデバイス名
  - 仮想アダプターのスロット番号
  - ポートの VLAN ID（タグ付きとタグなしの両方の使用法）
  - 追加の VLAN ID

追跡管理するフィールドの数からして、この種の情報の管理には、図 3-3（62 ページ）のようなスプレッドシートまたはデータベースのプログラムを使用することをお勧めします。システムのインストール時にデータを記録し、その後は構成変更のたびにデータを追跡管理するようにしてください。



Microsoft Excel - 4194ch4-virtmap.xls

File Edit View Insert Format Tools Data Window Help Adobe PDF

Type a question for help

	B	C	D	E	F	G	H	I	J	K	L
1	Hostname	VIOS Host	VIOS Virt Adapter	VIOS SEA	VIOS Phys Adapter	Default VID	Additional VIDs	Client Slot	VIOS Slot	Switch Port	Physical Adapter Location
2	server1	**vios1**	ent5	ent7	ent0	10	20, 30, 40	21	21	c103-3/20	U7879.001.DQD185T-P1-T6
3		**vios2**	ent5	ent7	ent0	10	20, 30, 40	22	22	c104-3/17	U7879.001.DQD186K-P1-T6
4		**vios1**	ent6	ent8	ent1	1		21	21	c101-1/14	U7879.001.DQD185T-P1-T7
5		**vios2**	ent6	ent8	ent1	1		22	22	c101-1/15	U7879.001.DQD186K-P1-T7
6	server2	**vios1**	ent5	ent7	ent0	10	20, 30, 40	23	23	c103-3/20	U7879.001.DQD185T-P1-T6
7		**vios2**	ent5	ent7	ent0	10	20, 30, 40	24	24	c104-3/17	U7879.001.DQD186K-P1-T6
8		**vios1**	ent6	ent8	ent1	1		23	23	c101-1/14	U7879.001.DQD185T-P1-T7
9		**vios2**	ent6	ent8	ent1	1		24	24	c101-1/15	U7879.001.DQD186K-P1-T7
10	server3	**vios1**	ent5	ent7	ent0	20	10, 30, 40	25	25	c103-3/20	U7879.001.DQD185T-P1-T6
11		**vios2**	ent5	ent7	ent0	20	10, 30, 40	26	26	c104-3/17	U7879.001.DQD186K-P1-T6
12		**vios1**	ent6	ent8	ent1	1		25	25	c101-1/14	U7879.001.DQD185T-P1-T7
13		**vios2**	ent6	ent8	ent1	1		26	26	c101-1/15	U7879.001.DQD186K-P1-T7
14	server4	**vios1**	ent5	ent7	ent0	20	10, 30, 40	27	27	c103-3/20	U7879.001.DQD185T-P1-T6
15		**vios2**	ent5	ent7	ent0	20	10, 30, 40	28	28	c104-3/17	U7879.001.DQD186K-P1-T6
16		**vios1**	ent6	ent8	ent1	1		27	27	c101-1/14	U7879.001.DQD185T-P1-T7

Virtual Disk Map | **Virtual Network Map**

図 3-3　ネットワークの追跡管理のためのスプレッドシート

3.3.1 仮想ネットワーク・アダプターと VLAN

仮想ネットワーク・アダプターは、メモリーの速度で稼働します。追加の物理アダプターが必要でも、追加の仮想アダプターは必要ない場合が少なくありません。それでも、仮想環境内の転送のために別個のアダプターを用意して大きな MTU サイズを使用することにはメリットがあります。そのようすれば、仮想環境内の転送のパフォーマンスが向上し、CPU 使用率が低下する可能性があるからです。詳しくは、3.11、『ジャンボ・フレームとパス MTU ディスカバリー』（86 ページ）を参照してください。

POWER Hypervisor™ は、システム内のトラフィックを分離するために使用できるタグつきの VLAN に対応しています。その同じ目的のために、別個のアダプターを使用することもできます。どちらの方式を選択するか、あるいは両方を組み合わせるかは、それぞれのデータ・センターのネットワークに関する一般的な習慣に基づいて決定してください。

3.3.2 仮想デバイスのスロット番号

仮想ストレージ・デバイスと仮想ネットワーク・デバイスは、スロット番号を共用します。複雑なシステムでは、ネットワーク・デバイスよりもストレージ・デバイスのほうがはるかに多くなる傾向があります。それぞれの仮想 SCSI デバイスは、1 つのサーバーまたはクライアントとしか通信できないからです。ネットワーク・デバイスを 1 つのグループとしてまとめるために、すべての LPAR のネットワーク・デバイス用として 20 までのスロット番号を予約しておくことをお勧めします。多数のアダプターを使用する複雑なネットワーク環境では、ネットワーク用にさらに多くのスロットが必要になる場合もあります。ストレージ・スロットの番号付けの詳細については、4.3.2、『仮想デバイスのスロット番号』（103 ページ）を参照してください。

LPAR を作成する時に、LPAR の仮想アダプター・スロットの最大数をデフォルト値の 10 よりも大きくする必要があります。それぞれの環境に適した値は、各システムで使用する LPAR の数とアダプターの数によって左右されます。仮想アダプター・スロットを使用しないままにしておくと、少量のメモリーを消費することになるので、要件の見積もりと実際の割り振りの間でバランスを取らなければなりません。それぞれのシステム構成で必要なメモリー所要量を計画するときには、以下の URL に用意されている System Planning Tool を使用してください。

http://www.ibm.com/servers/eserver/iseries/lpar/systemdesign.html

3.3.3 構成のトレース

記録の保持に万全を期したシステムでも、物理ハードウェアに対する仮想ネットワーク接続を手動でトレースすることが必要になる場合もあります。

複数の仮想ネットワーク・アダプターを使用する仮想 I/O クライアント・パーティションでは、**lscfg** コマンドによって確認できるアダプターの物理的な位置から各アダプターのスロット番号を判別できます。仮想アダプターの場合は、以下の例のように、そのフィールドに C で始まるカード・スロット番号が表示されます。

```
# lscfg -l ent\*
  ent2             U9111.520.10DDEDC-V5-C5-T1  Virtual I/O Ethernet Adapter (l-lan)
  ent1             U9111.520.10DDEDC-V5-C8-T1  Virtual I/O Ethernet Adapter (l-lan)
  ent0             U9111.520.10DDEDC-V5-C2-T1  Virtual I/O Ethernet Adapter (l-lan)
```

物理的な位置のフィールドでスロット番号を確認できたら、HMC によってトレースバックして、各アダプターで使用可能になっている接続と VLAN タグを判別できます（図 3-4 を参照）。この場合は、ent0 が VLAN 1 のスロット 2、ent1 が VLAN 40 のスロット 8、ent2 が VLAN 10 のスロット 5 にそれぞれあります。

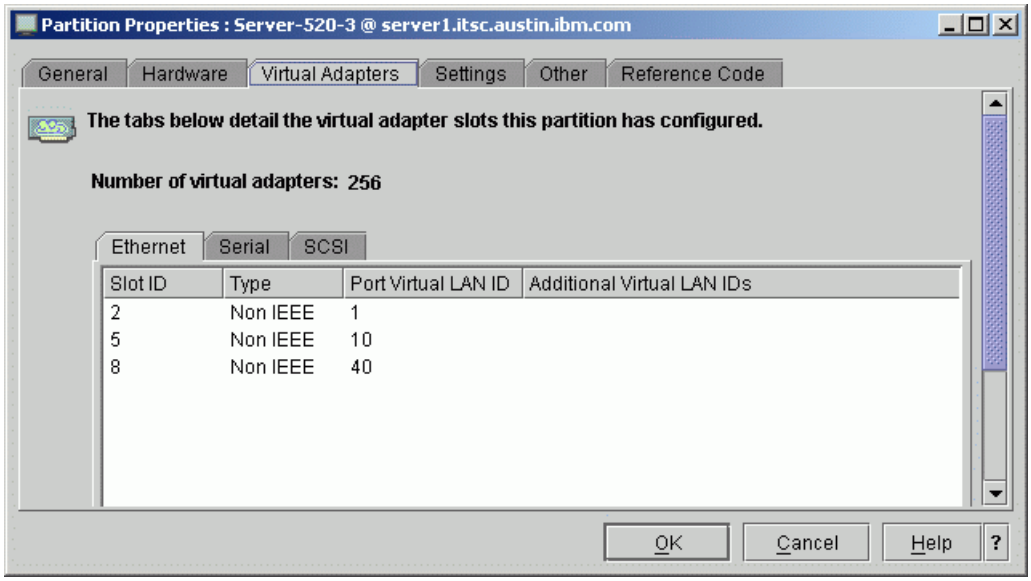


図 3-4　仮想イーサネット・アダプターのスロットの割り当て

3.4 複数のネットワーク・セキュリティー・ゾーンの管理

VLAN タグは、1 つのリンクのネットワーク・トラフィックを分離するための手段になります。

VLAN タグ付きの複数のネットワークを 1 つの仮想化環境に拡張することも可能です。ハイパーバイザーは、IEEE 802.1Q の VLAN タグに対応したイーサネット・スイッチをエミュレートするからです。仮想アダプターの作成時に IEEE 802.1Q 互換アダプターのチェック・ボックスを選択すると、すべてのデータ・フレームがタグ付きになり、同じ VLAN に設定された仮想イーサネット・アダプターだけに送信されるようになります。

図 3-5（64 ページ）の例では、VLAN 10 と VLAN 20 を使用しているので、パーティション 1 と 3 は、ハイパーバイザー・ファームウェアを使用して相互に通信できますが、パーティション 2 は 1 や 3 と通信できません。さらに、ハイパーバイザーは、クライアント・パーティション内の操作が、そのクライアント・パーティションに割り当てられていない共用リソースの制御を取得したり、その共用リソースを使用したりすることができないような設計になっています。

仮想 I/O サーバー上でクライアント・パーティションへの割り当てを変更可能なので、ユーザー ID とパスワードはイーサネット・スイッチの管理者パスワードなどと同様にセキュアな状態で保存する必要があります。一般に、仮想 I/O サーバーでは IP アドレスを設定するのが望ましいといえます。

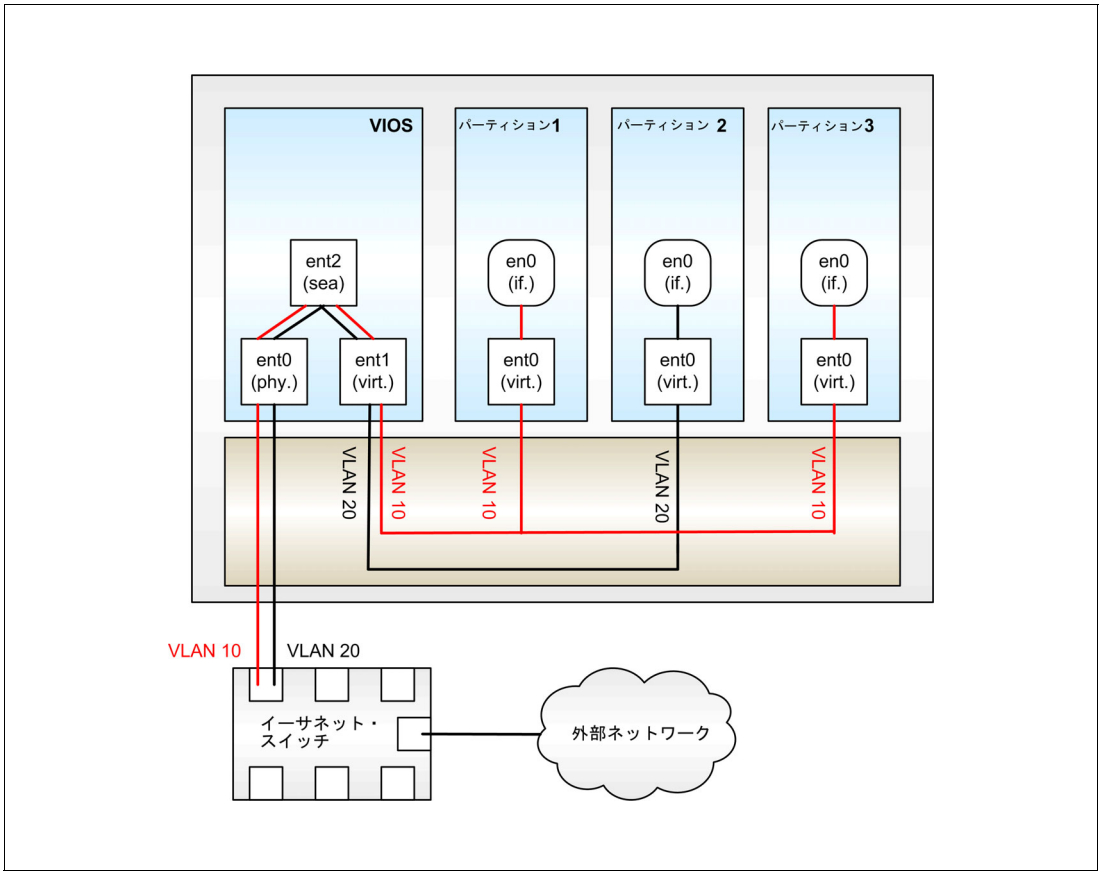


図 3-5　VLAN タグ環境

この実装では、エンタープライズのセキュリティー・ポリシーに以下のような内容を組み込む必要があります。

- エンタープライズのセキュリティー・ポリシーで、IEEE 802.1Q の VLAN タグを認めます。

Advanced POWER Virtualization ハイパーバイザー・ファームウェアには、IEEE 802.1Q の VLAN タグが実装されています。仮想 I/O サーバーでは、各共用イーサネット・アダプター（Shared Ethernet Adapter）ごとに最大で 21 個の VLAN をサポートできますが、その機能を使用するには、物理ネットワーク・ポートがそれと同じ数の VLAN をサポートしている必要があります。したがって、エンタープライズの仮想 VLAN ポリシーは、物理 VLAN ポリシーによって決まることになります。詳しくは、3.6、『仮想ネットワークへの VLAN の拡張』（68 ページ）を参照してください。

- セキュリティー・ポリシーで、複数の VLAN が 1 つのネットワーク・スイッチを共用することを認めます。

  エンタープライズ・ポリシーで、複数の VLAN が 1 つのネットワーク・スイッチを共用することが認められていれば（非物理セキュリティー）、図 3-5 のようなシナリオを実装できます。一方、1 つのネットワーク・スイッチで 1 つの VLAN だけをサポートするのがセキュリティー要件になっている場合は、各 VLAN に別個の管理対象システムが必要になります。管理対象システム内に別個の仮想 I/O サーバーを構成すれば、ハイパーバイザー・ファームウェアは、複数の VLAN に対応した 1 つのスイッチのように機能するので、この場合は、仮想 I/O サーバーの外部のセキュリティー・ポリシーによって認められない環境になってしまいます。

注 : 仮想環境のセキュリティーは、HMC または Integrated Virtualization Manager（IVM）と仮想 I/O サーバーに依存しています。管理対象システム内の既存のネットワークと VLAN の割り当てが無許可で変更されたり、新しいネットワークの割り当てが無許可で確立されたりする事態を回避するには、HMC、IVM、VIOS へのアクセスを厳密にモニターする必要があります。

社内のセキュリティー・ポリシーにセキュリティー・ゾーンが含まれていることを確認します。

複数のネットワーク・セキュリティー・ゾーンを実行する環境で動的 LPAR を使用する場合は、図 3-6 のような環境を実装する必要があります。

それぞれのセキュリティー・ゾーンを別々の専用 VLAN（VLAN 80 と 90）に配置し、HMC を基本 VLAN に配置するか、VLAN のアクセス制御リストを使用することが必要です。現行バージョンの HMC は VLAN に対応していないからです。

そのようにすれば、VLAN 80 と 90 は HMC と通信できますが、VLAN 80 と 90 が相互に通信することはできなくなります。

HMC は、HMC にアクセスできるユーザーや IP アドレスの数を限定した専用ネットワークに配置することをお勧めします。

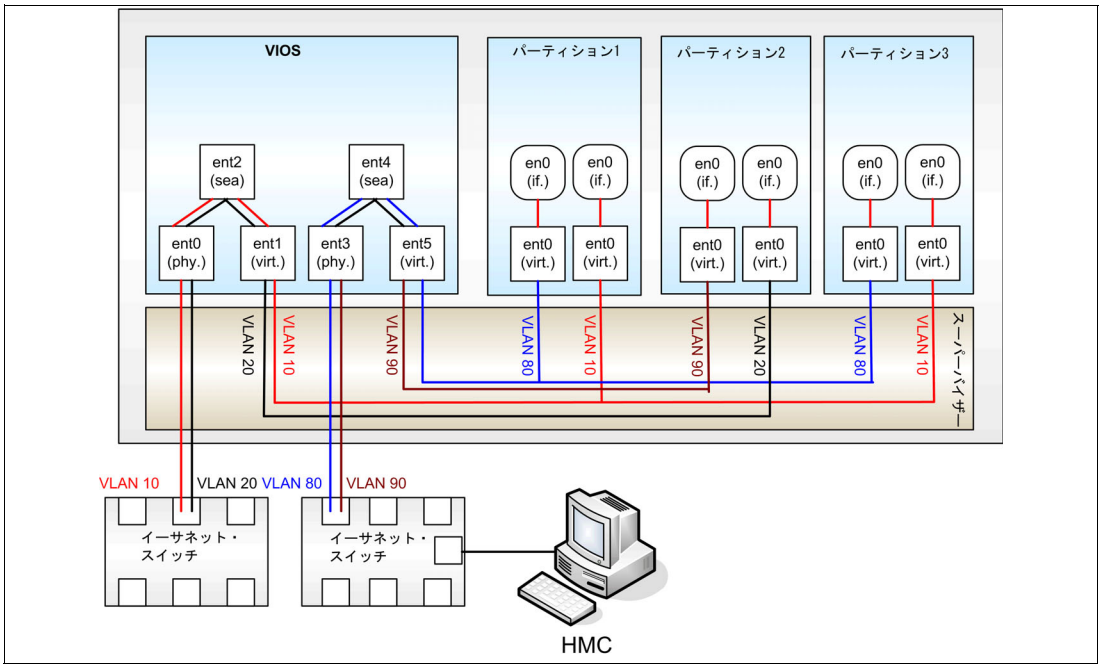


図 3-6　VLAN タグ環境の HMC

3.5 ネットワーク・ポート・セキュリティーを有効にした環境と仮想 I/O サーバー

現在のさまざまなネットワーク装置に用意されているオプションの 1 つは、1 つのメディア・アクセス制御（MAC）アドレスまたは 1 セットの MAC アドレスにポートをロックする機能です。SEA の機能を使用すれば、仮想ネットワーク・アダプターによって多数の論理パーティションにネットワーク接続を提供できます。ポート・セキュリティーと仮想 I/O サーバーを組み合わせて使用する場合は、どの MAC アドレスを物理ネットワーク・ポートにロックするのかを決定しなければなりません。

仮想 I/O サーバーの SEA に割り当てられている IP アドレスがあれば、その IP アドレスこそ、ネットワーク・トラフィックが最初に入ってくる場所であり、SEA の MAC アドレスになります。したがって、そのアドレスをリストに追加します。例 3-1（67 ページ）は、MAC アドレスを確認する方法を示したものです。

例 3-1　仮想 I/O サーバーの SEA の MAC アドレスの検出

```
$ lsmap -net -all
SVEA   Physloc
------ --------------------------------------------
ent1   U9113.550.105E9DE-V2-C2-T1

SEA                  ent2
Backing device       ent0
Physloc              U787B.001.DNW108F-P1-C1-T1

$ entstat ent2 |grep Hardware
Hardware Address: 00:02:55:d3:dd:00
```

SEA は、第 2 層のブリッジとして機能するので、SEA に IP アドレスがなければ、SEA の MAC アドレスをロックする必要はありません。

SEA によってブリッジが提供されるパケットについては、すべての MAC アドレスが保存されます（ネットワーク・パケットで表示されるのは、AIX 5L または Linux の論理パーティションにある仮想イーサネット・アダプターの MAC アドレスになります）。SEA によってトラフィックのブリッジが提供される仮想イーサネット・アダプターを使用する AIX 5L または Linux の各論理パーティションについては、その仮想イーサネット・アダプターの MAC アドレスも追加する必要があります。

仮想ネットワークの冗長性を確保するために、仮想 I/O サーバーで物理ネットワーク・アダプターまたは Network Interface Backup のリンク集約を使用する場合は、正しい MAC アドレスを使用する必要があります。ほとんどの場合は、最初に指定されているアダプターの MAC アドレスが使用されますが、このような状況では、ユーザー定義の MAC アドレスを使用することもできます（代替 MAC アドレスは、通常の物理アダプターまたは仮想アダプターでも有効です）。代替 MAC アドレスは、アダプター・カードまたは仮想イーサネット・アダプターのデフォルトの MAC アドレスとは異なっていてもかまいません。

3.6 仮想ネットワークへの VLAN の拡張

VLAN テクノロジーを使用すれば、従来のネットワーク・テクノロジーの場合よりも柔軟なネットワーク展開が可能になります。環境の物理的な制約を乗り越えて、必要なスイッチ、ポート、アダプター、ケーブル接続、アップリンクの数を抑えるのに役立つからです。ただし、物理的な展開が簡略化されるとはいっても、それには代償が伴わないわけではありません。つまり、VLAN を使用すれば、スイッチやホストの構成が複雑になってしまいます。それでも、全体的に見れば、複雑さの程度が高くなるわけではありません。つまり、物理から仮想へのシフトということです。このセクションは、VLAN の概念を理解していることが前提になります。これから、VLAN タグを使用して VLAN の構成とセットアップを行うための推奨方式を見ていきましょう。

3.6.1 VLAN をセットアップするためのシナリオ

仮想 I/O サーバー環境では、仮想 I/O サーバーが内部の仮想 LAN と外部の物理 LAN の間のリンクを提供します。この場合は、サーバー内の複数の VLAN をサーバー外の複数の VLAN に安全な方法で接続しなければならないので、複雑さのレベルが高くなってしまいます。つまり、仮想 I/O サーバーをすべての VLAN に接続することが必要ですが、それと同時に VLAN 同士の間でパケットが移動しないようにすることも必要になります。

このシナリオ（図 3-7（69 ページ））では、以下のような要件を満たす必要があります。

- すべてのクライアント・パーティションが同じ VLAN に存在する他のクライアント・パーティションと通信できることが必要です。
- すべてのクライアント・パーティションが仮想 I/O サーバーにある 1 つの仮想イーサネット・アダプターと通信できることが必要です。そのためには、仮想 I/O サーバーの仮想イーサネット・アダプターで IEEE 802.1Q を使用して、複数の VLAN ID が仮想イーサネット・アダプターで受け入れられるようにする必要があります。
- クライアント・パーティションから送られてくるパケットの VLAN タグが除去されないようにする必要があります。VLAN タグが除去されると、仮想 I/O サーバーが正しい外部 VLAN にパケットを転送できなくなります。
- 共用イーサネット・アダプター（SEA）で複数の VLAN からのパケットをサポートできるようにする必要があります。

図 3-7 は、4 つのパーティションと 1 つの仮想 I/O サーバーを使用した環境です。以下の VLAN ID を使用します。

- 10
- 20
- 30

さらに、デフォルトの VLAN ID として 199 も使用します。このデフォルトの VLAN ID は、固有の ID でなければならず、ネットワーク内の他のクライアントや物理イーサネット・スイッチのポートで使用されていない ID でなければなりません。詳しくは、3.6.4、『VLAN タグが仮想 I/O サーバーで除去されないようにする設定』（72 ページ）を参照してください。

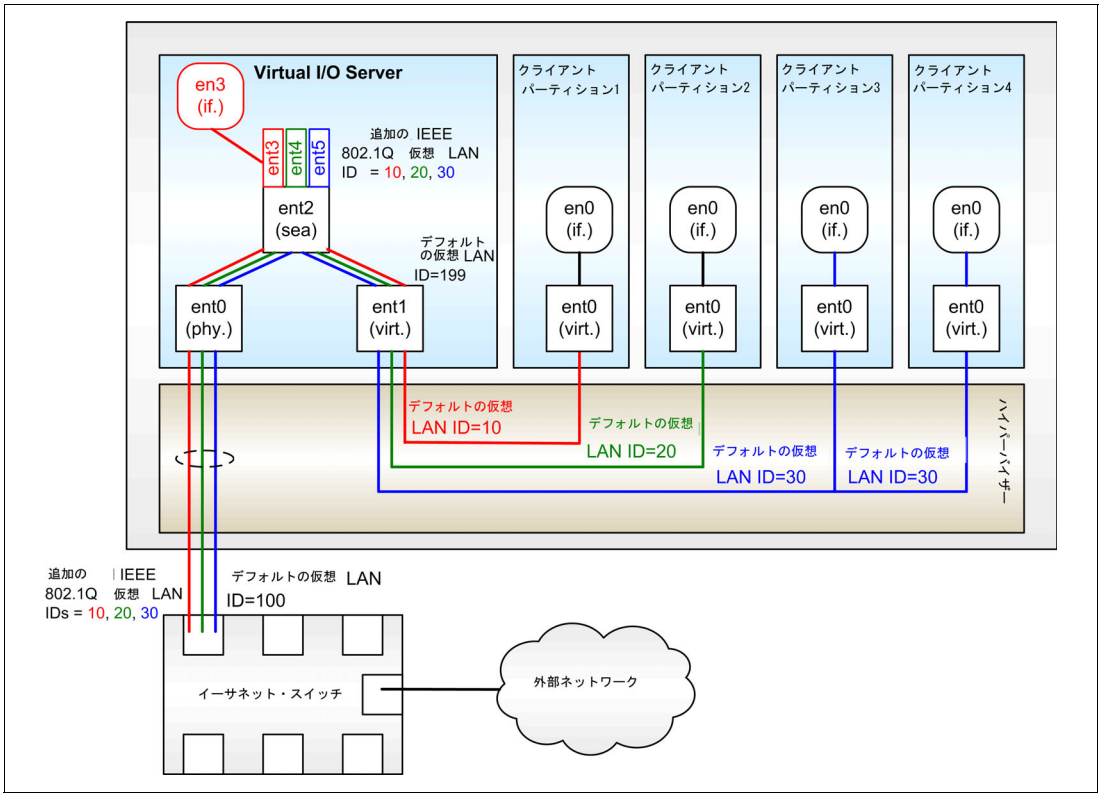


図 3-7 VLAN 構成のシナリオ

注 : すべての物理ネットワーク・スイッチが VLAN タグに対応しているわけではありません。仮想ネットワークの範囲を超えて物理ネットワークにまで VLAN タグを拡張する場合は、その物理ネットワークも VLAN タグに対応している必要があります。仮想ネットワークの VLAN 設定に合わせて物理ネットワークを構成する作業を忘れてしまうのは、よくある間違いです。

3.6.2 内部 VLAN のセットアップ

図 3-7（69 ページ）からもわかるとおり、内部 VLAN の構成とセットアップは、単純明快なプロセスです。ここでは、図 3-7（69 ページ）のようなシナリオを使用して、各クライアント・パーティションで以下の手順を実行し、内部 VLAN を構成します。

1. HMC にログオンします。
2. パーティションの作成を開始します。
3. 仮想イーサネット・アダプターのウィンドウで、図 3-7（69 ページ）のようにして、デフォルトの VLAN ID を設定した仮想イーサネット・アダプターを各クライアント・パーティションに割り当てます。図 3-8 は、クライアント・パーティション 1 の仮想イーサネット・アダプターを作成する例を示したものです。
4. 「Access external network」フラグと「IEEE 802.1Q compatible adapter」フラグが両方ともクリアされていることを確認します。
5. パーティションの作成を終了します。

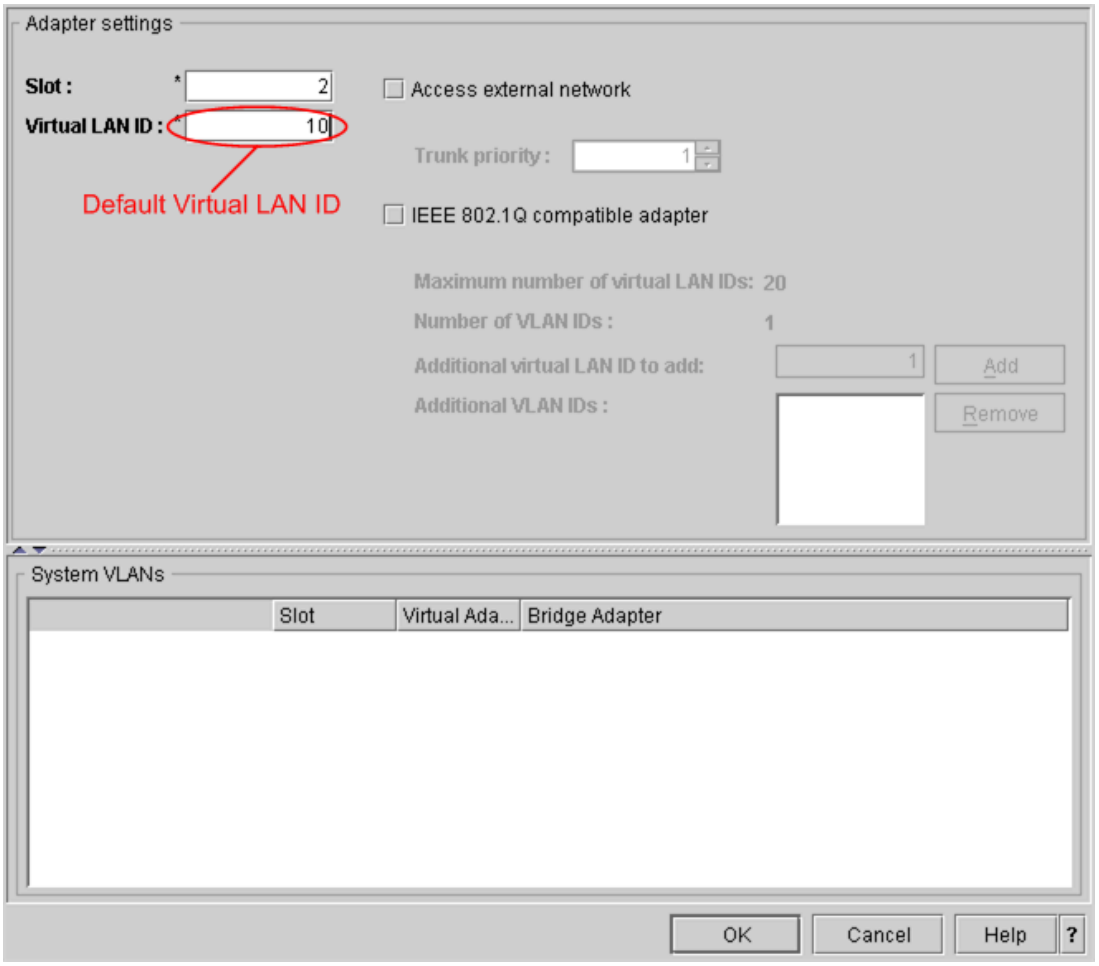


図 3-8　HMC によるクライアント・パーティションの仮想イーサネット構成

この構成が完了すると、クライアント・パーティションから送られてくるパケットに仮想イーサネット・アダプターの VLAN ID が追加されるようになります。パケットは、この ID に基づいて正しいパーティションに経路指定されますが、送信前に ID は除去されます。例えば、図 3-7（69 ページ）からわかるとおり、クライアント・パーティション 3 とクライアント・パーティション 4 は、VLAN ID が同じなので、直接通信できます。一方、クライアント・パーティション 1 とクライアント・パーティション 2 は、VLAN ID がそれぞれ異なるので、クライアント・パーティション 3 やクライアント・パーティション 4 と通信することも、相互に通信することもできませんが、マシンの外部の VLAN にある他のマシンとは通信できる状態になっています。

3.6.3 仮想 I/O サーバーの仮想イーサネット・アダプターの構成

仮想 I/O サーバーは、複数の内部 VLAN と外部の物理イーサネット・アダプターや LAN を結びつける唯一の中継ポイントになります。仮想ネットワークと物理ネットワークの間のブリッジを設定するには、共用イーサネット・アダプター（SEA）デバイスを使用して、1 つ以上の仮想アダプターを物理アダプターにリンクします。ここでは、図 3-7（69 ページ）のようなシナリオを使用して、複数の内部 VLAN に接続する 1 つの仮想イーサネット・アダプターを実装します。

この構成を正しくセットアップするには、図 3-9 のようにして、仮想 I/O サーバーで作成する仮想イーサネット・アダプターで、接続先の内部 VLAN をすべて指定する必要があります。以下の手順を実行します。

1. HMC にログオンします。
2. 仮想イーサネット・アダプターのウィンドウで、固有のデフォルト VLAN ID を設定した仮想イーサネット・アダプターを仮想 I/O サーバーに割り当てます。その ID は、クライアント・パーティションや物理ネットワークで使用されていない ID でなければなりません。このシナリオでは、VLAN ID として 199 を使用します。
3. 「**IEEE 802.1Q compatible adapter**」フラグを選択します。
4. クライアント・パーティションに関連付ける VLAN ID を追加します。このシナリオの場合、追加の VLAN ID は、10、20、30 です。
5. 「**Access external network**」フラグが選択されていることを確認します。SEA のフェイルオーバー構成を使用する場合を除き、「Trunk priority」はデフォルトのままにしておきます。

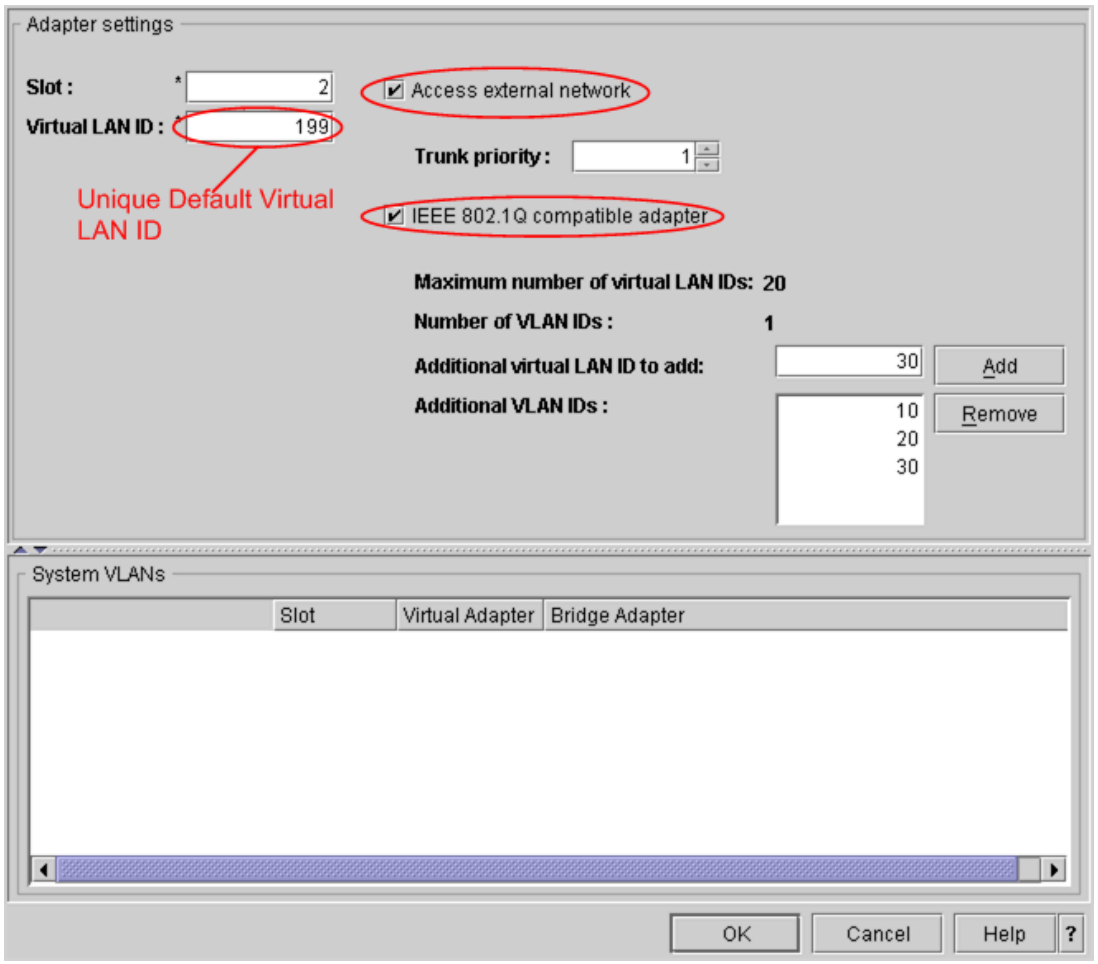


図 3-9 HMC による仮想 I/O サーバーの仮想イーサネット構成

この場合は、「Access external network」フラグを選択します。この仮想 I/O サーバーは、共用イーサネット・アダプター（Shared Ethernet Adapter）機能を使用して、外部ネットワークとの間でパケットをやり取りするからです。「IEEE 802.1Q compatible adapter」フラグも選択して、共用イーサネット・アダプター（Shared Ethernet Adapter）が追加の VLAN ID でパケットを送信できるようにします。これらの追加の VLAN ID は、クライアント・パーティションで構成する VLAN ID と同じなので、ハイパーバイザーによって追加された VLAN タグの付いたクライアント・パーティションから送られてくるパケットは、共用イーサネット・アダプター（Shared Ethernet Adapter）をパススルーできるようになります。

3.6.4 VLAN タグが仮想 I/O サーバーで除去されないようにする設定

クライアント・パーティションから送られてくるパケットに追加される VLAN ID（デフォルトの VLAN ID 番号）が、仮想 I/O サーバーに入ってくる時点で除去されないようにすることは重要です。デフォルトの構成を使用する場合は、そのような動作になります。

図 3-9（71 ページ）のように、仮想 I/O サーバーの仮想イーサネット・アダプターのデフォルトの VLAN ID を未使用の VLAN ID（この場合は 199）に設定する必要があるのは、そのためです。

その VLAN ID（199）のパケットが入ってきても、VLAN ID タグが除去されてしまえば（つまり、タグが外されてしまえば）、正しい外部 VLAN に転送できなくなります。そのパケットが共用イーサネット・アダプター（Shared Ethernet Adapter）経由で外部物理ネットワークに送信される場合は、タグが外された状態でイーサネット・スイッチに到着することになります。その結果は、物理イーサネット・スイッチの設定によって異なります。そのパケットは、破棄されるか、デフォルトの VLAN に送信される可能性がありますが、ネットワーク管理者が VLAN ID 199 を明示的にセットアップしている場合（例えば、イーサネット・スイッチ・ポートのデフォルトの VLAN ID として 199 を設定している場合など）を除き、VLAN ID 199 に送られると考えるのは現実離れしています。

3.6.5 複数の VLAN にアクセスするための共用イーサネット・アダプター（Shared Ethernet Adapter）の構成

このシナリオでは、仮想イーサネット・アダプターと物理イーサネット・アダプターを使用する仮想 I/O サーバーで **mkvdev -sea** コマンドを実行して、共用イーサネット・アダプター（Shared Ethernet Adapter）を作成します。共用イーサネット・アダプター（Shared Ethernet Adapter）を構成して、複数の VLAN にアクセスするには、以下の手順を実行します。

1. 仮想 I/O サーバーで **mkvdev** コマンドを使用して、共用イーサネット・アダプター（Shared Ethernet Adapter）を作成します。このシナリオでは、仮想イーサネット・アダプター ent1 と物理イーサネット・アダプター ent0 を使用して、新しい共用イーサネット・アダプター（Shared Ethernet Adapter）ent2 を作成します。

   ```
   $ mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 199
   ent2 Available
   en2
   et2
   ```

   共用イーサネット・アダプター（Shared Ethernet Adapter）の作成時には、-defaultent1 オプションによって、デフォルトの内部仮想 VLAN（つまり、パケットのタグが外された場合にパケットの送信先にする VLAN）を指定します。このシナリオでは、仮想イーサネット・アダプターが 1 つしかありませんが、このコマンドに複数の仮想イーサネット・アダプターを指定するような複雑なシナリオでは、このオプションを使用することになります。さらに、-defaultid 199 オプションでは、タグの外されたパケットに使用する VLAN ID を指定します（つまり、これが SEA のデフォルトの VLAN ID になります）。ここでは、タグの外されたパケットを想定していませんし、そのような未使用の番号を指定しても、VLAN 199 のパケットを受け入れるクライアントが存在しないので、そのようなパケットは送信されなくなります。**lsdev** コマンドを実行すると、新しく作成された SEA が表示されます。

   ```
   $ lsdev -type adapter
   name            status                                      description
   ent0            Available  10/100/1000 Base-TX PCI-X Adapter (14106902)
   ent1            Available  Virtual I/O Ethernet Adapter (l-lan)
   ent2            Available  Shared Ethernet Adapter
   ide0            Available  ATA/IDE Controller Device
   sisioa0         Available  PCI-X Dual Channel U320 SCSI RAID Adapter
   vhost0          Available  Virtual SCSI Server Adapter
   ```

```
vsa0            Available  LPAR Virtual Serial Adapter
```

2. **mkvdev -vlan** コマンドを使用して、新しく作成した共用イーサネット・アダプター（Shared Ethernet Adapter）（ent2）を構成し、追加の VLAN ID（10、20、30）のパケットを許可するようにします。

```
$ mkvdev -vlan ent2 -tagid 10
ent3 Available
en3
et3
$ mkvdev -vlan ent2 -tagid 20
ent4 Available
en4
et4
$ mkvdev -vlan ent2 -tagid 30
ent5 Available
en5
et5
```

   ここで、前のそれぞれのコマンドに対応した新しいイーサネット・アダプターが作成されます。

3. **mktcpip** コマンドを使用して、新しい VLAN のいずれかで IP アドレスを構成し、管理者がネットワーク経由で仮想 I/O サーバーにアクセスできるようにします。このシナリオでは、図 3-7（69 ページ）のように、VLAN 10 にある en3 を使用して、仮想 I/O サーバーのネットワーク接続のための IP アドレスを構成します。

注 : 通常、仮想 I/O サーバーで IP アドレスが必要になるのは、管理作業のために限られます。したがって、管理 VLAN で IP アドレスを構成する目的も、管理者がネットワーク経由で仮想 I/O サーバーにアクセスすることだけになります。

3.7 仮想 I/O サーバーのリンク集約

リンク集約とは、いくつかのイーサネット・アダプターをまとめて 1 つの疑似イーサネット・アダプターを形成するためのネットワーク・ポート集合テクノロジーです。基本的には、1 つのネットワーク・アダプターの帯域幅の制限を乗り越えたり、多数のクライアント・パーティションで 1 つのネットワーク・アダプターを共用する場合のボトルネックを回避したりするために、このテクノロジーを使用します。さまざまな種類のリンク集約テクノロジーの詳細については、*「Advanced POWER Virtualization on IBM System p5」*（SG88-4018）を参照してください。このセクションでは、共用イーサネット・アダプター（Shared Ethernet Adapter）の一部としてリンク集約デバイスを使用する方法を取り上げます。

3.7.1 リンク集約の作成

図 3-10（74 ページ）は、2 つの物理イーサネット・アダプターと 1 つのバックアップ・アダプターによるリンク集約を使用した単一仮想 I/O サーバー構成です。

注 : バックアップ・アダプター・オプションを使用するのは、単一仮想 I/O サーバー構成の場合に限ることをお勧めします。二重仮想 I/O サーバー構成の場合は、バックアップ・アダプターなしでリンク集約を実装してください。SEA のフェイルオーバー機能によって、必要なフェイルオーバー機能を用意できるからです。

SEA のフェイルオーバーでは、ネットワークがトラフィックを転送するときに、最大で 15 個から 30 個のパケットが失われる可能性があります。

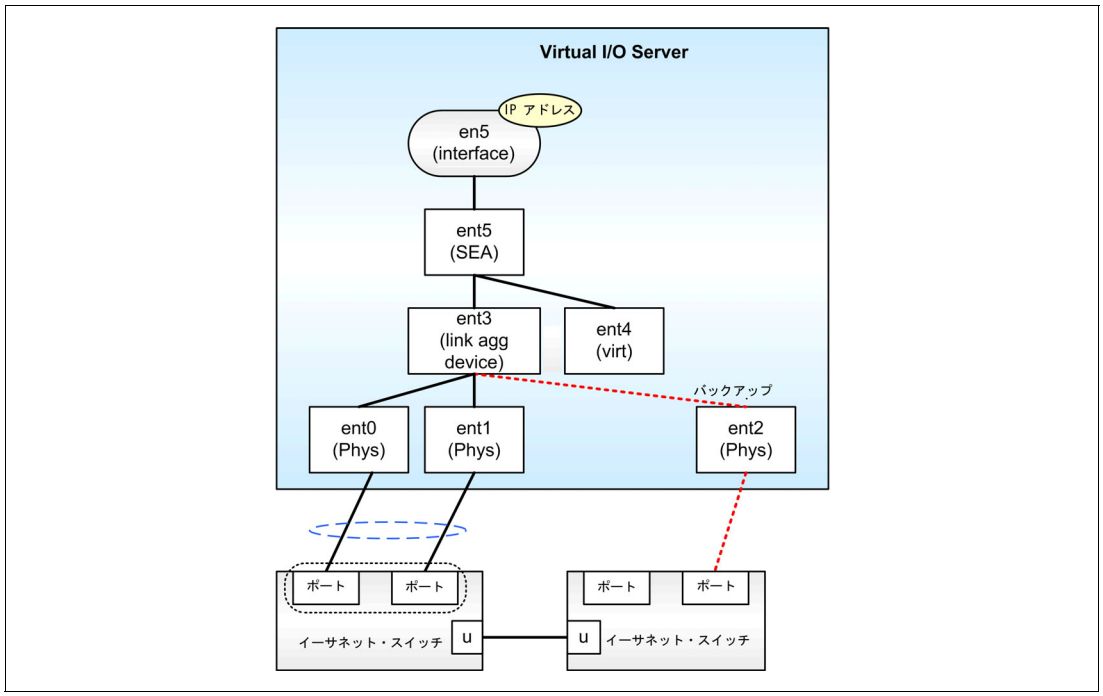


図 3-10　単一仮想 I/O サーバー構成を使用したリンク集約

図 3-10 は、2 つの物理イーサネット・アダプターと 1 つのバックアップ・アダプターによるリンク集約を使用した単一仮想 I/O サーバー構成であり、この後の説明文では、その構成のために必要な手順の概略を示します。

1. ここでは、図 3-10 の例を使用して、以下の物理イーサネット・アダプターによってリンク集約を構成します。

– ent0
– ent1
– ent2

```
$ lsdev -type adapter
name            status                                          description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ide0            Available  ATA/IDE Controller Device
sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
```

注 : リンク集約を形成するすべてのネットワーク・アダプター（バックアップ・アダプター以外）は、同じネットワーク・スイッチに接続する必要があります。

2. リンク集約内の各アダプターについて、メディアの速度やジャンボ・フレームなどの必須属性を適切に設定します。リンク集約では、すべてのアダプターに同じ属性値を設定する必要があります。

```
$ chdev -dev ent0 -attr media_speed=Auto_Negotiation jumbo_frames=yes
ent0 changed
```

3. 属性が有効になっていることを確認します。

```
$ lsdev -dev ent0 -attr
attribute      value            description
user_settable

alt_addr       0x000000000000   Alternate ethernet address                     True
busintr        88               Bus interrupt level                           False
busmem         0xc0100000       Bus memory address                            False
chksum_offload yes              Enable hardware transmit and receive checksum True
compat_mode    no               Gigabit Backward compatability                True
copy_bytes     2048             Copy packet if this many or less bytes        True
delay_open     no               Enable delay of open until link state is known True
failback       yes              Enable auto failback to primary               True
failback_delay 15               Failback to primary delay timer               True
failover       disable          Enable failover mode                          True
flow_ctrl      yes              Enable Transmit and Receive Flow Control      True
intr_priority  3                Interrupt priority                            False
intr_rate      10000            Max rate of interrupts generated by adapter   True
jumbo_frames   yes              Transmit jumbo frames                         True
large_send     yes              Enable hardware TX TCP resegmentation         True
media_speed    Auto_Negotiation Media speed                                   True
rom_mem        0xc0040000       ROM memory address                            False
rx_hog         1000             Max rcv buffers processed per rcv interrupt   True
rxbuf_pool_sz  2048             Rcv buffer pool, make 2X rxdesc_que_sz        True
rxdesc_que_sz  1024             Rcv descriptor queue size                     True
slih_hog       10               Max Interrupt events processed per interrupt  True
tx_que_sz      8192             Software transmit queue size                  True
txdesc_que_sz  512              TX descriptor queue size                      True
use_alt_addr   no               Enable alternate ethernet address             True
```

注 : Network Backup Interface オプションを使用する場合は、図 3-10（74 ページ）で図示されているように、プライマリー・ネットワーク・スイッチが使用不能になった場合でもネットワーク・スイッチの冗長性を確保するために、バックアップ・アダプターを別個のスイッチに接続することをお勧めします。

4. 仮想 I/O サーバーで、**mkvdev -lnagg** コマンドを実行して、リンク集約デバイスを作成します。この場合は、単一仮想 I/O サーバー構成なので、ネットワーク・バックアップ・オプションを構成し、バックアップ・ネットワーク・デバイスとして ent2 を使用します。

```
$ mkvdev -lnagg ent0 ent1 -attr backup_adapter=ent2
ent3 Available
en3
et3
```

注 : クライアント・パーティションでいずれかのネットワーク・インターフェース（en0 または en1）がすでに NIM インストールの一部として構成されている場合は、そのインターフェースを取り外すだけでなく削除することが必要です。そうすれば、Network Interface Backup インターフェースの構成前にそのインターフェースが構成されているという状態を回避できます。

-attr フラグを使用して、必要なリンク集約のタイプを指定します。用意されている属性は、standard、round_robin、8023ad です。正しい設定は、物理ネットワーク・スイッチの構成によって決まります。基本的には、standard オプションを選択しますが、ネットワーク管理者に問い合わせて、ネットワーク・スイッチで構成されているリンク集約のタイプを確認するようにしてください。さらに、**netaddr** フラグを使用して、バックアップ・アダプターの ping 対象アドレスを指定することもできます。

注 : 二重仮想 I/O サーバー構成の場合は、バックアップ・アダプターなしでリンク集約を実装してください。SEA のフェイルオーバー機能によって、必要なフェイルオーバー機能を用意できるからです。

5. **lsdev** コマンドを使用して、イーサチャンネル・デバイスをリストします。

```
$ lsdev -type adapter
name            status                                      description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent3            Available  EtherChannel / IEEE 802.3ad Link Aggregation
ide0            Available  ATA/IDE Controller Device
sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
$ lsdev -dev ent3 -attr
attribute       value          description                                  user_settable

adapter_names   ent0,ent1      EtherChannel Adapters                        True
alt_addr        0x000000000000 Alternate EtherChannel Address               True
auto_recovery   yes            Enable automatic recovery after failover     True
backup_adapter  ent2           Adapter used when whole channel fails        True
hash_mode       default        Determines how outgoing adapter is chosen    True
mode            standard       EtherChannel mode of operation               True
netaddr         0              Address to ping                              True
noloss_failover yes            Enable lossless failover after ping failure True
```

```
num_retries     3              Times to retry ping before failing          True
retry_time      1              Wait time (in seconds) between pings        True
use_alt_addr    no             Enable Alternate EtherChannel Address       True
use_jumbo_frame yes            Enable Gigabit Ethernet Jumbo Frames        True
```

6. HMC を使用して、新しい仮想イーサネット・アダプターを仮想 I/O サーバーに追加します。「**Access external network**」フラグが選択されていることを確認してください。動的 LPAR を使用するための情報については、2.2.2、『動的 LPAR の割り当て』（40 ページ）を参照してください。

```
$ lsdev -type adapter
name            status                                          description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent3            Available  EtherChannel / IEEE 802.3ad Link Aggregation
ent4            Available  Virtual I/O Ethernet Adapter (l-lan)
ide0            Available  ATA/IDE Controller Device
sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
```

7. 新しく作成した ent3 リンク集約デバイスと ent4 仮想イーサネット・アダプターを使用して、共用イーサネット・アダプター（Shared Ethernet Adapter）デバイスを構成します。

```
$ mkvdev -sea ent3 -vadapter ent4 -default ent4 -defaultid 2
ent5 Available
en5
et5
$ lsdev -type adapter
name            status                                          description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent3            Available  EtherChannel / IEEE 802.3ad Link Aggregation
ent4            Available  Virtual I/O Ethernet Adapter (l-lan)
ent5            Available  Shared Ethernet Adapter
ide0            Available  ATA/IDE Controller Device
sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
```

3.7.2 既存のリンク集約への物理イーサネット・アダプターの追加

仮想 I/O サーバーの共用イーサネット・アダプター（Shared Ethernet Adapter）を使用したクライアント・パーティションのネットワーク帯域要件を高めるには、既存のリンク集約デバイスに追加の物理アダプターを動的に構成します。ここでは、図 3-10（74 ページ）の例を使用して、既存のリンク集約に新しいアダプターを追加するために、以下の手順を実行します。

1. HMC を使用して、新しい物理イーサネット・アダプターを仮想 I/O サーバーに追加します。動的 LPAR を使用するための情報については、2.2.2、『動的 LPAR の割り当て』（40 ページ）を参照してください。

2. **lsdev** コマンドを実行して、仮想 I/O サーバーで新しいアダプターが構成されていることを確認します。

```
$ lsdev -type adapter
name            status                                          description
ent0            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
```

```
ent1            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent3            Available  EtherChannel / IEEE 802.3ad Link Aggregation
ent4            Available  Virtual I/O Ethernet Adapter (l-lan)
ent5            Available  Shared Ethernet Adapter
ent6            Available  2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ide0            Available  ATA/IDE Controller Device
sisscsia0       Available  PCI-X Ultra320 SCSI Adapter
sisscsia1       Available  PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0          Available  Virtual SCSI Server Adapter
vsa0            Available  LPAR Virtual Serial Adapter
```

3. 以下のように **chdev** コマンドと **lsdev** コマンドを使用して、新しいアダプターの属性が既存のリンク集約に含まれている物理イーサネット・アダプターの属性と同じかどうかを確認し、必要に応じて変更を加えます（例えば、media_speed、jumbo_frames）。

```
$ lsdev -dev ent6 -attr
attribute      value            description
user_settable

alt_addr       0x000000000000   Alternate ethernet address                    True
busintr        88               Bus interrupt level                           False
busmem         0xc0100000       Bus memory address                            False
chksum_offload yes              Enable hardware transmit and receive checksum  True
compat_mode    no               Gigabit Backward compatability                True
copy_bytes     2048             Copy packet if this many or less bytes        True
delay_open     no               Enable delay of open until link state is known True
failback       yes              Enable auto failback to primary               True
failback_delay 15               Failback to primary delay timer               True
failover       disable          Enable failover mode                          True
flow_ctrl      yes              Enable Transmit and Receive Flow Control      True
intr_priority  3                Interrupt priority                            False
intr_rate      10000            Max rate of interrupts generated by adapter   True
jumbo_frames   yes               Transmit jumbo frames                         True
large_send     yes              Enable hardware TX TCP resegmentation         True
media_speed    Auto_Negotiation Media speed                                   True
rom_mem        0xc0040000       ROM memory address                            False
rx_hog         1000             Max rcv buffers processed per rcv interrupt   True
rxbuf_pool_sz  2048             Rcv buffer pool, make 2X rxdesc_que_sz        True
rxdesc_que_sz  1024             Rcv descriptor queue size                     True
slih_hog       10               Max Interrupt events processed per interrupt  True
tx_que_sz      8192             Software transmit queue size                  True
txdesc_que_sz  512              TX descriptor queue size                      True
use_alt_addr   no               Enable alternate ethernet address             True
```

4. **oem_setup_env** シェルにログインし、**ethchan_config** コマンドを使用して、新しいアダプターを既存のリンク集約に構成します。その後、**oem_setup_env** シェルを終了します。仮想 I/O サーバーの将来の更新では、**oem_setup_env** コマンドの使用が廃止され、新しいコマンド・オプションまたはコマンドに置き換えられる可能性があります。

```
$ oem_setup_env
# ethchan_config -a -p ent5 ent3 ent6
# exit
```

5. **lsdev** コマンドを使用して、新しいアダプターがリンク集約に組み込まれていることを確認します。

```
$ lsdev -dev ent3 -attr
attribute       value          description                              user_
settable

adapter_names   ent0,ent1,ent6 EtherChannel Adapters                    True
alt_addr        0x000000000000 Alternate EtherChannel Address           True
```

```
auto_recovery   yes             Enable automatic recovery after failover    True
backup_adapter  ent2            Adapter used when whole channel fails       True
hash_mode       default         Determines how outgoing adapter is chosen   True
mode            standard        EtherChannel mode of operation              True
netaddr         0               Address to ping                             True
noloss_failover yes             Enable lossless failover after ping failure True
num_retries     3               Times to retry ping before failing          True
retry_time      1               Wait time (in seconds) between pings        True
use_alt_addr    no              Enable Alternate EtherChannel Address       True
use_jumbo_frame no              Enable Gigabit Ethernet Jumbo Frames        True
```

バックアップ・アダプターを追加する場合は、**ethchan_config** コマンドで **-b** フラグを使用します。

```
#ethchan_config -a -b -p <SEA device> <EtherChannel device> <adapter to add as backup>
```

アダプターを除去する場合は、**ethchan_config** コマンドで **-a** フラグの代わりに **-d** フラグを使用します。

```
#ethchan_config -a -p <SEA device> <EtherChannel device> <adapter to add as primary>
```

重要 : リンク集約に新しいアダプターを追加する場合は、物理ネットワーク・スイッチのポートが、リンク集約のその新しいアダプターに対応するように構成されていることを確認しなければなりません。

3.8 ネットワークの可用性のためのオプション

このセクションでは、仮想環境のさまざまな種類のネットワーク可用性構成で使用できる各種のオプションを取り上げ、その利点について説明します。

3.8.1 単一仮想 I/O サーバー構成の高可用性オプション

単一仮想 I/O サーバーを使用する構成では、仮想 I/O サーバーで Network Interface Backup 機能を使用することによって、ネットワークの可用性を強化できます。このセットアップでは、共用イーサネット・アダプター（Shared Ethernet Adapter）を使用するすべてのクライアント・パーティションで物理イーサネット・アダプターの冗長性を確保します。この種の構成によって、物理アダプターまたは物理ネットワーク・スイッチのいずれかが使用不能になった場合でも、クライアント・パーティションを保護できます。仮想 I/O サーバーで Network Interface Backup を構成する方法については、3.7、『仮想 I/O サーバーのリンク集約』（74 ページ）を参照してください。

3.8.2 二重仮想 I/O サーバー構成の拡張可用性オプション

二重仮想 I/O サーバー構成では、ネットワークの可用性を強化するために 2 つの一般的なオプションを使用できます。

- 1 つは、クライアント・パーティションに Network Backup Interface（NIB）機能を実装するオプションです。
- もう 1 つは、仮想 I/O サーバー V1.2 で導入された共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）機能を使用するオプションであり、クライアント・パーティションでは追加の構成が必要ありません。

このセクションでは、それぞれの機能を使用するべき状況について説明します。

Network Interface Backup の利点

NIB を使用すれば、リソースの使用率を改善できます。その理由は、以下のとおりです。

- 共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）では、2 つの共用イーサネット・アダプター（Shared Ethernet Adapter）のうちのいずれか 1 つだけが常にアクティブに使用され、もう 1 つはスタンバイになります。したがって、スタンバイ共用イーサネット・アダプター（Shared Ethernet Adapter）の物理イーサネット・アダプターの帯域幅は使用されません。
- 一方、NIB の場合は、両方の共用イーサネット・アダプター（Shared Ethernet Adapter）に各クライアントを分散させ、半分は最初の仮想 I/O サーバーの共用イーサネット・アダプター（Shared Ethernet Adapter）をプライマリー・アダプターとして使用し、もう半分は第 2 の仮想 I/O サーバーの共用イーサネット・アダプター（Shared Ethernet Adapter）をプライマリー・アダプターとして使用するような構成が可能です。この場合は、両方の仮想 I/O サーバーの共用イーサネット・アダプター（Shared Ethernet Adapter）における物理イーサネット・アダプターの帯域幅を別々のクライアント・パーティションで使用できます。ただし、そのためには、追加の VLAN のペアと追加のハードウェアが必要になります。

Network Interface Backup を使用するべき状況

共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）オプションよりも Network Interface Backup のほうが適しているのは、以下のような場合です。

- 既存の Network Interface Backup 構成が存在する環境で、Virtual I/O Server Version 1.1 からのアップグレードを実行するときに、イーサネットのセットアップを変更したくない場合。

- AIX 5L オペレーティング・システムを基盤とするパーティションだけを実行している場合。
- VLAN タグを使用していない場合。
- 両方の仮想 I/O サーバーを使用して、クライアント・パーティションのロード・バランシングを実施したい場合。

共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）の利点

共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）には、Network Interface Backup と比較した場合に以下のような利点があります。

- 共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）は、仮想 I/O サーバーに実装されているので、仮想ネットワークの管理作業がシンプルになります。
- クライアント・パーティションでは、1 つの仮想イーサネット・アダプターと VLAN だけが必要であり、フェイルオーバーのロジックを実装する必要はないので、クライアントの構成作業が容易になります。
- Network Interface Backup の方式を採用した場合は、NIB 機能に加えて、すべてのクライアントについて、別の VLAN で第 2 の仮想イーサネット・アダプターを構成し、リンク集約アダプターを作成する必要があるので、クライアント・パーティションの構成が複雑になります。
- 共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）には、IEEE 802.1Q の VLAN タグに関する追加サポートがあります。
- SEA のフェイルオーバーの場合は、クライアント・パーティションで必要な仮想イーサネット・デバイスが 1 つだけなので、NIM インストールがシンプルになります。NIM インストールの後にイーサネット構成を変更する必要はありません。
- ping を設定する必要があるのは、1 箇所だけです。一方、NIB の場合は、各クライアントで ping を設定しなければなりません。

SEA のフェイルオーバーを使用するべき状況

Network Interface Backup オプションよりも共用イーサネット・アダプター・フェイルオーバー（Shared Ethernet Adapter Failover）のほうをお勧めするのは、以下のような場合です。

- VLAN タグを使用している場合。
- Linux オペレーティング・システムを実行している場合。
- プライマリー仮想 I/O サーバーとスタンバイ仮想 I/O サーバーの間で共用イーサネット・アダプター（Shared Ethernet Adapter）ごとのロード・バランシングを実施する必要がない場合。

ほとんどの場合は、SEA フェイルオーバーの利点のほうが NIB の利点を上回るので、外部ネットワークに対するブリッジ・アクセスの高可用性を実現するためのデフォルトの方式は、SEA フェイルオーバーにするのが望ましいといえます。

3.9 SEA のフェイルオーバーをサポートするための共用イーサネット・アダプター（Shared Ethernet Adapter）の作成

多くのクライアントでは、二重仮想 I/O サーバーによって柔軟性を最大限強化できます。つまり、仮想 I/O サーバーの保守などの必要な機能を各仮想 I/O サーバーで実行することが可能になるので、クライアントの中断が最小限に抑えられます。さらに、ネットワークの可用性を確保するために（3.8、『ネットワークの可用性のためのオプション』（80 ページ））、SEA のフェイルオーバー方式を使用できます。この方式は、既存の SEA デバイスを拡張したものであり、単一仮想 I/O サーバー・セットアップで SEA を構成する場合に比べて、少なくとも 2 つの新しいパラメーターを組み込む必要があります。

ctl_chan もう一方の仮想 I/O サーバーの SEA との間で状況情報をやり取りするために使用する仮想イーサネット・アダプター。仮想 I/O サーバーは、それらの SEA のうちのどちらが仮想ネットワークから物理ネットワークへのブリッジングを提供するのかを決定できます。

ha_mode SEA の機能モード。auto 設定が最も一般的ですが、standby などの他の設定を使用して、SEA フェイルオーバーを強制設定することも可能です。

二重仮想 I/O サーバー・セットアップでこれらの共用イーサネット・アダプター（Shared Ethernet Adapter）を作成するときには、コントロール・チャネルと高可用性モードを作成時に指定することが非常に重要です。これらの設定がないと、共用イーサネット・アダプター（Shared Ethernet Adapter）同士がそれぞれの相手を認識できないので、仮想ネットワークから物理ネットワークへのネットワーク・トラフィックのブリッジングをどちらからも開始できます。その結果、ネットワークでループが発生し、それがネットワークの中断につながることもあります。図 3-11（83 ページ）は、SEA のフェイルオーバーをセットアップする方法を示した図です。

重要 : ctl_chan 属性を共用イーサネット・アダプター（Shared Ethernet Adapter）として定義していない場合は、どちらの SEA がブリッジング機能を提供するのかに関するネゴシエーションができません。この時点では、両方の SEA がブリッジング機能を提供することになるので、スパンニング・ツリー・ループが形成される可能性があります。その状態を回避するには、SEA の作成時にそれらのパラメーターを必ず指定しなければなりません。

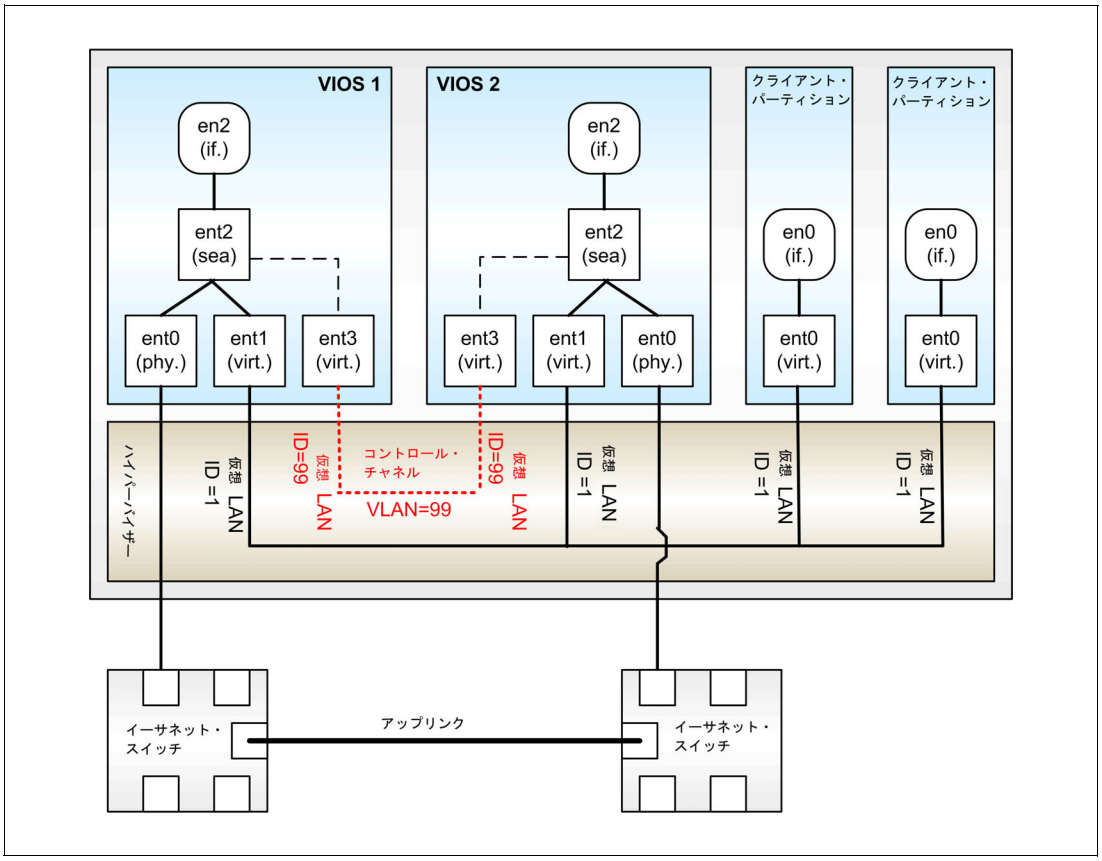


図 3-11 SEA のフェイルオーバーのセットアップ図

SEA のフェイルオーバーのペアでこれらの SEA を作成する場合は、**ctl_chan** 属性と **ha_mode** 属性を指定して、SEA オプションを以下の例のように記述することが重要です。

```
$mkvdev  -sea TargetDevice -vadapter VirtualEthernetAdapter ...
               -default DefaultVirtualEthernetAdapter
               -defaultid SEADefaultPVID [-attr Attributes=Value ...]
               [-migrate]
$mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid=10 -attr ha_mode=auto
ctl_chan=ent2
```

単一仮想 I/O サーバー構成で 1 つの SEA を作成する場合は、以下のようなコマンドを使用します。

```
$mkvdev  -sea TargetDevice -vadapter VirtualEthernetAdapter ...
               -default DefaultVirtualEthernetAdapter
               -defaultid SEADefaultPVID [-attr Attributes=Value ...]
               [-migrate]
$mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid=10
```

3.10 仮想 I/O サーバーの SEA のスレッド化

仮想 I/O サーバーを使用すれば、AIX 5L と Linux のオペレーティング・システムを基盤としたクライアントのディスク・トラフィックとネットワーク・トラフィックの両方を仮想化できます。その 2 種類のトラフィックの主な違いは、永続性です。仮想 I/O サーバーがネットワーク・データを移動する場合は、その移動をすぐに実行しなければなりません。ネットワーク・データには永続ストレージがないからです。そのため、仮想 I/O サーバーに用意されているネットワーク・サービス（共用イーサネット・アダプター（Shared Ethernet Adapter）など）は、最高の優先順位で実行する必要があります。一方、仮想 SCSI デバイスのディスク・データの実行優先順位は、ネットワーク・データよりも低くなります。そのデータはディスクに格納されているので、タイムアウトによってデータが失われてしまう危険性も低いからです。そのためのデバイスも、通常は処理速度がそれほど高くありません。

thread 属性によって、SEA のスレッドのスケジュール設定を変更できます。

Virtual I/O Server Version 1.3 よりも前の共用イーサネット（Shared Ethernet）プロセスは、ハイパフォーマンスのために最適化された割り込みレベルで実行されます。つまり、この方式では、ネットワーク・トラフィックのレベルが非常に高くなった場合に、仮想 SCSI プロセスよりも共用イーサネット（Shared Ethernet）プロセスのほうが高い優先順位で実行されるということです。その両方のための十分な CPU リソースが仮想 I/O サーバーに用意されていなければ、仮想 SCSI のパフォーマンスは低下し、サービスの質が落ちる結果になりかねません。

Virtual I/O Server Version 1.3 では、カーネルのスレッドを使用して共用イーサネット（Shared Ethernet）機能を実装できるようになりました。つまり、仮想ディスクと仮想ネットワークの間で、処理能力をより均等に配分することが可能になっています。

このスレッド化機能は、thread 属性を変更することによって、共用イーサネット・アダプター（Shared Ethernet Adapter）ごとにオン / オフを設定できます。しかも、SEA の作動中でも、サービスを中断しないで設定を変更できます。スレッド化機能を使用する場合は、値として 1 を設定し、元の割り込み方式を使用する場合は、0 を設定します。

```
$ lsdev -dev ent2 -attr thread
value

0
$ chdev -dev ent2 -attr thread=1
ent2 changed
$ lsdev -dev ent2 -attr thread
value

1
```

大きな負荷をかけてパフォーマンスの差異をテストした結果、スレッド化機能を使用しない場合のほうが、同じネットワーク・スループットに対して必要な CPU の量が約 8% 少なくなります。ただし、ネットワーク・トラフィックはバーストしやすい性質があるので、スレッド化機能を有効にすることをお勧めします（現在はその設定がデフォルトです）。バーストしやすいというのは、ユーザーのログオンや Web ページのロードなどによってネットワーク・トラフィックが大きく変動するという意味です。このような変動は、ディスク・アクセスの増加と同時に発生することもあります。例えば、ユーザーがシステムにログオンすると、ネットワーク・アクティビティーのレベルも高くなります。ログオン・プロセスでは、ディスクに格納されている一種のパスワード・データベースにアクセスしたり、ユーザー・プロファイルを読み込んだりすることが多いからです。

スレッド化機能を無効にすることを検討するべき状況としては、1 つの仮想 I/O サーバーをネットワーク専用、もう 1 つの仮想 I/O サーバーをディスク専用として構成している場合があります。このような構成は、CPU に制約のある 1 つのサーバーで非常に大きなディスク・ロードとネットワーク・ロードが混在している場合にのみお勧めします。

5.4、『仮想 I/O サーバーのサイズの決定』（136 ページ）でも見たとおり、CPU の消費量は、基本的にディスクよりもネットワークのほうが大きくなります。さらに、定期保守のために一方の仮想 I/O サーバーを構成から除去する場合は、ディスクの仮想 I/O サーバー・セットアップによって、SEA のフェイルオーバーを使用したネットワーク・バックアップを用意するのが通常の方式です。その場合は、同じ仮想 I/O サーバーでディスクとネットワークの両方を操作することになるので、スレッド化機能を使用するのが望ましいといえます。

3.11 ジャンボ・フレームとパス MTU ディスカバリー

このセクションでは、最大転送単位（MTU）について取り上げ、ジャンボ・フレームの使用方法を説明します。さらに、パス MTU ディスカバリーに関する AIX 5L Version 5.3 の変更点についても説明し、パス MTU ディスカバリーによって仮想イーサネットを調整する作業に関する推奨事項を示します。

ここで取り上げるのは、以下のトピックです。

- 最大転送単位
- パス MTU ディスカバリー
- ジャンボ・フレームの使用方法と仮想イーサネットに関する推奨事項

3.11.1 最大転送単位

イーサネットや IEEE 802.x のローカル・エリア・ネットワークなどでは、フレームのサイズに関する制限があります。例えば、イーサネットの場合は、フレームの最大長が 1526 バイトなので、最大で 1500 バイトの長さのデータ・フィールドをサポートできます。IEEE 802.3 では、データ・フィールドの長さが伝送速度によって左右されます。表 3-1 は、標準的な最大転送単位（MTU）をまとめたものです。

表 3-1 標準的な最大転送単位（MTU）

ネットワーク	MTU（バイト）
公式の最大 MTU	65535
イーサネット（10 または 100 MBps）	1500
イーサネット（ギガビット）	9000
FDDI	4352
X.25	576
公式の最小 MTU	68

ネットワーク経由でデータを送信するときに、そのデータがネットワークの MTU よりも大きい場合は、そのデータが小さな部分に分割され、フラグメント化することがあります。各フラグメントのサイズは、MTU 以下になるはずです。

MTU サイズは、ソース・システムとターゲット・システムの間のネットワークのパフォーマンスに影響を及ぼすことがあります。MTU サイズとして大きな値を使用すると、オペレーティング・システムは、大きなサイズのパケットを送信できるので、同じネットワーク・スループットで考えた場合に、パケット数を減らすことができます。そのようにしてパケットのサイズを大きくして、パケットの数を減らせば、オペレーティング・システムで必要な処理の量が減ります。どのパケットでも、必要な処理の量は同じだからです。ただし、小さなメッセージを送信するだけのワークロードであれば、MTU サイズを大きくしても効果はありません。

MTU サイズによって、最大セグメント・サイズ（MSS）も影響を受けます。MSS は、TCP 層が宛先の IP に送信できるデータの最大サイズです。接続が確立された時点で、各システムは MSS の値を通知できます。一方のシステムが他方のシステムから MSS を受け取らない場合は、デフォルトの MSS 値を使用することになります。

AIX 5L Version 5.2 以前では、デフォルトの MSS 値は 512 バイトでしたが、AIX 5L Version 5.3 では、デフォルト値として 1460 バイトがサポートされています。AIX 5L Version 5.2 に APAR IY57637 を適用すれば、デフォルトの MSS 値は 1460 バイトに変更されます。

no -a コマンドを実行すると、デフォルトの MSS 値が tcp_mssdflt として表示されます。例えば、例 3-2（87 ページ）のような情報が表示されます。

例 3-2 AIX 5L V5.3 のデフォルトの MSS 値

```
# no -a | grep tcp
       tcp_bad_port_limit = 0
                  tcp_ecn = 0
       tcp_ephemeral_high = 65535
        tcp_ephemeral_low = 32768
             tcp_finwait2 = 1200
          tcp_icmpsecure = 0
        tcp_init_window = 0
    tcp_inpcb_hashtab_siz = 24499
             tcp_keepcnt = 8
            tcp_keepidle = 14400
            tcp_keepinit = 150
           tcp_keepintvl = 150
     tcp_limited_transmit = 1
             tcp_low_rto = 0
            tcp_maxburst = 0
             tcp_mssdflt = 1460
         tcp_nagle_limit = 65535
       tcp_nagleoverride = 0
              tcp_ndebug = 100
            tcp_newreno = 1
          tcp_nodelayack = 0
       tcp_pmtu_discover = 1
           tcp_recvspace = 16384
           tcp_sendspace = 16384
           tcp_tcpsecure = 0
            tcp_timewait = 1
                 tcp_ttl = 60
          tcprexmtthresh = 3
```

接続が最初に確立された時点でソース・ネットワークが MSS を受け取らなければ、システムはデフォルトの MSS 値を使用します。ほとんどのネットワーク環境はイーサネットなので、少なくとも 1500 バイトの MTU をサポートできます。

例えば、AIX 5L Version 5.2 以前で、MSS 値を受け取っていない状況で FTP アプリケーションを実行すると、そのアプリケーションは、最初の接続ではデフォルトの MSS 値である 512 バイトを使用することになり、パフォーマンスが低下しかねません。ただし、MSS については、接続ごとにネゴシエーションが行われるので、次の接続では別の MSS を使用する可能性もあります。

3.11.2 パス MTU ディスカバリー

すべてのネットワーク・リンクでは、3.11.1、『最大転送単位』（86 ページ）で取り上げた MTU によって最大パケット・サイズが決まっています。実際にデータグラムが 1 つのシステムから別のシステムに転送されるときには、それぞれ MTU 値の異なる多数のリンクを経由することがあります。ソース・システムと宛先システムの MTU 値が異なる場合は、フラグメント化が発生したり、最小の MTU のリンクが選択されたときにパケットがドロップされたりする可能性があります。パス内のすべてのリンクの最小 MTU のことをパス MTU といい、ソースから宛先までのパス全体の最小 MTU を判別するプロセスのことをパス MTU ディスカバリー（PMTUD）といいます。

AIX 5L Version 5.2 以前では、インターネット制御 MTU ディスカバリー（ICMP）エコー要求と ICMP エコー応答のパケットを使用して、IPv4 を基盤としたパス MTU を検出します。

そのための基本的な手順はシンプルです。まず、1 つのシステムがパス MTU ディスカバリーによって送信操作を最適化しようとして、最大サイズのパケットを送信します。それらのパケットが 2 つのシステム間のいずれか 1 つのリンクに収まらない場合は、そのリンクから、そのリンクでサポートできる最大サイズを示した通知が送り返されます。その通知は、IP データグラムのソースに対して、「fragmentation needed and DF set」というコードの付いた ICMP の「Destination Unreachable」メッセージを戻します（タイプ 3、タイプ 4）。

ソースは、その ICMP メッセージを受け取ると、送信 MSS の値を下げて、その値で再び送信を試みます。リンクのすべてのステップの最大可能値が検出されるまで、この操作を繰り返します。

パス MTU ディスカバリーの手順を実行すると、以下のような結果になることがあります。

- パケットがフラグメント化なしですべてのリンクを通過して宛先システムに到着するという結果
- ソース・システムが、宛先システムまでのパスのいずれかのホップから、MSS が大きすぎるのでそのリンクではサポートできないという趣旨の ICMP メッセージを受け取るという結果

この ICMP エコー要求と応答の手順には、いくつかの考慮事項があります。例えば、システム管理者の中には、サービス妨害（DoS）アタックのリスクがあると考えて、パス MTU ディスカバリーを使用しない人もいます。

また、すでにパス MTU ディスカバリーを使用している場合でも、ルーターやファイアウォールによって、ソース・システムに戻される ICMP メッセージがブロックされることがあります。その場合、ソース・システムは、ネットワーク環境からメッセージを受け取れないので、デフォルトの MSS 値を設定しますが、その値はすべてのリンクでサポートされているとは限りません。

AIX 5L Version 5.2 以前では、検出された MTU 値は、クローン作成メカニズムによってルーティング・テーブルに格納されるので、マルチパス・ルーティングには使用できません。この場合は、2 つのマルチパス・ネットワーク経路が交互に使用されるのではなく、クローン経路が常に使用されるからです。したがって、**netstat -rn** コマンドを使用すれば、検出された MTU 値を確認できます。

AIX 5L Version 5.3 では、パス MTU ディスカバリーの手順が変更されました。ICMP エコー応答と要求のパケットは、使用しません。AIX 5L Version 5.3 では、ICMP エコー応答と要求のパケットの代わりに、TCP パケットと UDP データグラムを使用します。さらに、検出された MTU もルーティング・テーブルには格納されません。したがって、パス MTU ディスカバリーとマルチパス・ルーティングを連係させることが可能になっています。

まず、1 つのシステムがパス MTU ディスカバリーによって送信操作を最適化しようとすると、パス MTU（PMTU）テーブルに pmtu 項目が作成されます。そのテーブルを表示するには、例 3-3 のようにして、**pmtu display** コマンドを使用します。pmtu 項目が累積されないようにするために、使用されない pmtu 項目は、pmtu_expire の時間が経過した時点で、期限切れになって削除されます。

例 3-3　パス MTU の表示

```
# pmtu display

  dst           gw            If    pmtu     refcnt   redisc_t    exp

 --------------------------------------------------------------------------

9.3.5.120     127.0.0.1       lo0   16896        6         24       0

9.3.4.155     9.3.5.41        en0    1500        1          0       0
```

```
9.3.5.195      9.3.5.120       en0    1500        0         20        4

127.0.0.1      127.0.0.1       lo0    16896       2          1        0
```

パス MTU テーブルの項目の有効期限は、**no** コマンドの pmtu_expire オプションによって制御できます。pmtu_expire のデフォルトの設定値は、10 分です。

IPv6 では、PMTU を検出するために ICMPv6 パケットを送信することはありません。接続の最初のパケットが常にこのプロセスを開始します。さらに、IPv6 のルーターは、パケットのフラグメント化を実行できないので、発信 MTU が小さすぎてパケットを転送できない場合は、常に ICMPv6 の「Packet too big」メッセージを戻す必要があります。したがって、IPv6 の場合は、PMTU ディスカバリーとマルチパス・ルーティングを連係させるために変更を加える必要はありません。

3.11.3 ジャンボ・フレームの使用

AIX 5L オペレーティング・システムの物理イーサネット・アダプターのジャンボ・フレーム・サポートは、シンプルな設計になっており、物理アダプターの 1 つの属性によって制御します。仮想イーサネット・アダプターは、すべての有効な MTU サイズを自動的にサポートします。

仮想イーサネット接続でジャンボ・パケットのためのブリッジを用意できます。

仮想イーサネット・アダプターには、ジャンボ・フレームのための属性がありません。仮想イーサネット・アダプターの上にインターフェースを構成する場合は、その仮想イーサネット・インターフェースに MTU 値が存在することになります。Virtual I/O Server Version 1.3 では、共用イーサネット・アダプター（SEA）インターフェースからジャンボ・フレームを送信する操作はサポートされていませんが、ジャンボ・パケットのブリッジングはサポートされています。本書の執筆時点では、SEA インターフェース自体との間でやり取りされるパケットは、MTU として 1500 を使用します。

ただし、SEA の主な目的は、仮想 I/O クライアントと外部ネットワークとの間をつなぐネットワーク通信のブリッジを用意することです。仮想 I/O クライアントの仮想アダプターで MTU が 9000 に設定され、SEA に関連付けられている仮想 I/O サーバーの物理イーサネット・アダプターでジャンボ・フレームが有効になっている場合は、仮想 I/O クライアントから外部ネットワークへのトラフィックで、MTU 値 9000 を使用できます。SEA は、ジャンボ・フレームを使用してネットワーク・トラフィックを開始することはできませんが、そのトラフィックのブリッジとして機能することはできます。

仮想 I/O クライアントと外部ネットワーク間のジャンボ・フレーム通信を構成するには、以下の手順を実行します。

1. 仮想イーサネット・アダプターには、以下に示すとおり、物理イーサネット・アダプターのようなジャンボ・フレームの属性や大規模送信（large_send）の属性がありません。

```
# lsattr -El ent0
alt_addr       0x000000000000 Alternate Ethernet Address        True
chksum_offload yes            Checksum Offload Enable           True
copy_buffs     32             Transmit Copy Buffers             True
copy_bytes     65536          Transmit Copy Buffer Size         True
max_buf_huge   64             Maximum Huge Buffers              True
max_buf_large  64             Maximum Large Buffers             True
max_buf_medium 256            Maximum Medium Buffers            True
max_buf_small  2048           Maximum Small Buffers             True
max_buf_tiny   2048           Maximum Tiny Buffers              True
min_buf_huge   24             Minimum Huge Buffers              True
min_buf_large  24             Minimum Large Buffers             True
min_buf_medium 128            Minimum Medium Buffers            True
```

```
min_buf_small  512            Minimum Small Buffers             True
min_buf_tiny   512            Minimum Tiny Buffers              True
trace_debug    no             Trace Debug Enable                True
use_alt_addr   no             Enable Alternate Ethernet Address True
```

2. 仮想 I/O サーバー（VIOS）構成。

   VIOS で 1 つの属性を変更する必要があります。つまり、物理アダプターの MTU 値を変更します。物理インターフェースの場合は、以下のコマンドによってジャンボ・フレーム属性を設定することによって、MTU 9000 を構成できます。

   ```
   $ chdev -dev ent0 -attr jumbo_frames=yes
   ent0 changed
   ```

   この物理アダプターが SEA によって使用されている場合は、SEA を除去してから再作成します。物理アダプターのジャンボ・フレーム属性をチェックするには、**lsdev -dev ent0 -attr** コマンドを使用できます。

3. 仮想 I/O クライアント。

   仮想デバイスでは、MTU 値を変更するために、仮想イーサネット・アダプターに以下の手順を適用できます。

   ```
   lpar10:/]chdev -l en0 -a mtu=9000
   en0 changed
   lpar10:/]lsattr -El en0
   alias4                      IPv4 Alias including Subnet Mask           True
   alias6                      IPv6 Alias including Prefix Length         True
   arp           on            Address Resolution Protocol (ARP)          True
   authority                   Authorized Users                           True
   broadcast                   Broadcast Address                          True
   mtu           9000          Maximum IP Packet Size for This Device     True
   netaddr       9.3.5.120     Internet Address                           True
   netaddr6                    IPv6 Internet Address                      True
   netmask       255.255.255.0 Subnet Mask                                True
   prefixlen                   Prefix Length for IPv6 Internet Address    True
   remmtu        576           Maximum IP Packet Size for REMOTE Networks True
   rfc1323                     Enable/Disable TCP RFC 1323 Window Scaling True
   security      none          Security Level                             True
   state         up            Current Interface Status                   True
   tcp_mssdflt                 Set TCP Maximum Segment Size               True
   tcp_nodelay                 Enable/Disable TCP_NODELAY Option          True
   tcp_recvspace               Set Socket Buffer Space for Receiving      True
   tcp_sendspace               Set Socket Buffer Space for Sending        True
   ```

ヒント：スイッチで、SEA に関連した実アダプターに接続しているポートをチェックしてください。ジャンボ・フレームが有効になっている必要があります。

3.11.4 パス MTU ディスカバリーによる仮想イーサネットの調整に関する推奨事項

このセクションの目的は、インターフェース固有のネットワーク・オプション（ISNO）の変更に伴って、VLAN のスループットがどうなるのかをテストすることです。ここでは、同時マルチスレッディングを有効にしたキャップ・モードで、それぞれの仮想 I/O クライアントに 1 つの仮想プロセッサーと 0.5 個分の共用プロセッサーを割り当てます。

図 3-12（91 ページ）は、仮想イーサネット・アダプターによって POWER Hypervisor を通過する仮想 I/O クライアント間の接続を示したものです。両方のパーティションで、tcp_sendspace と tcp_recvspace をそれぞれ異なる値に設定して、単信モードと二重モードで TCP_STREAM ベンチマーク・プログラムを実行します。

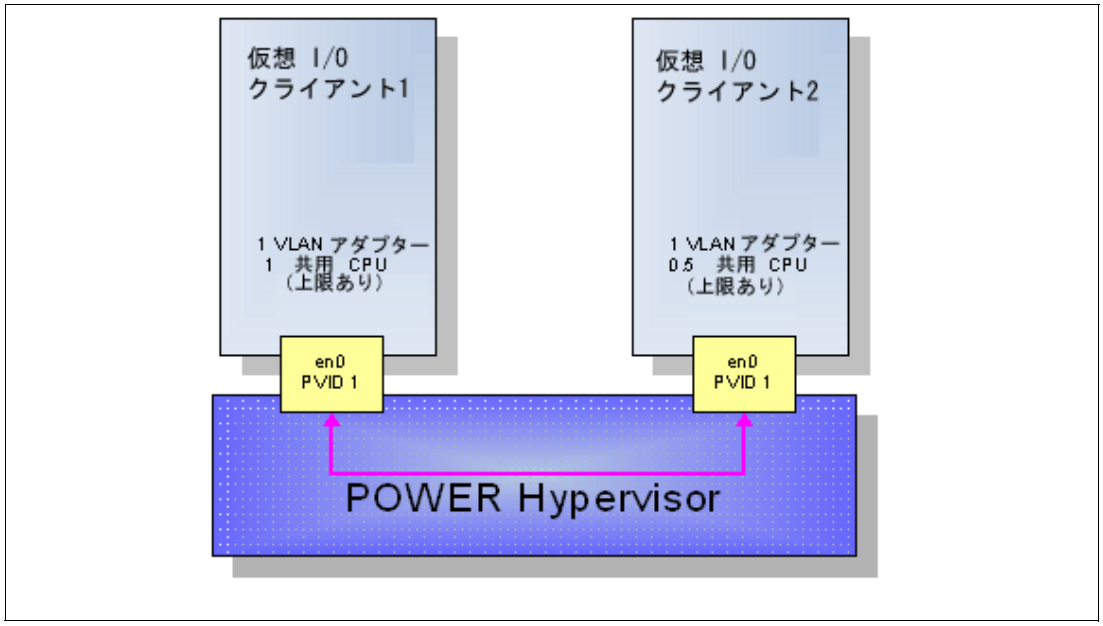


図 3-12 VLAN のパフォーマンス構成

VLAN のパフォーマンス

図 3-13、図 3-14（92 ページ）、図 3-15（92 ページ）は、MTU サイズをそれぞれ 1500、9000、65394 とした場合に、tcp_sendspace と tcp_recvspace の値の違いによってスループットがどのように変化するかを示したものです。

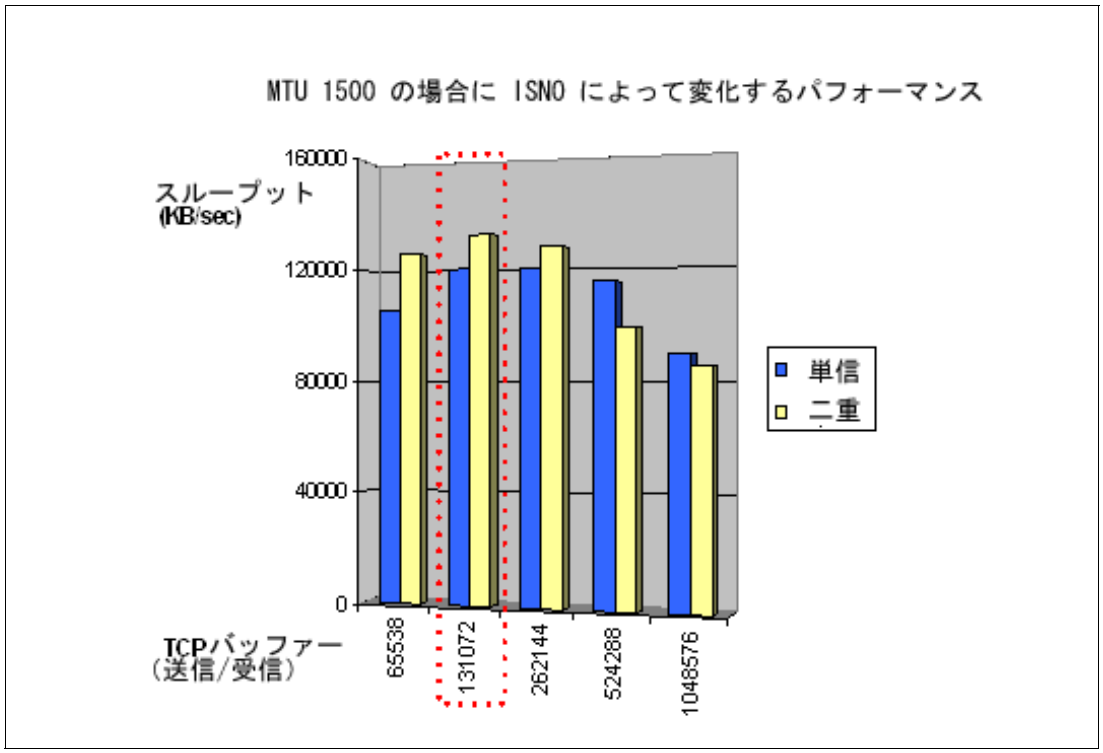


図 3-13 MTU 1500 の場合に ISNO によって変化するパフォーマンス

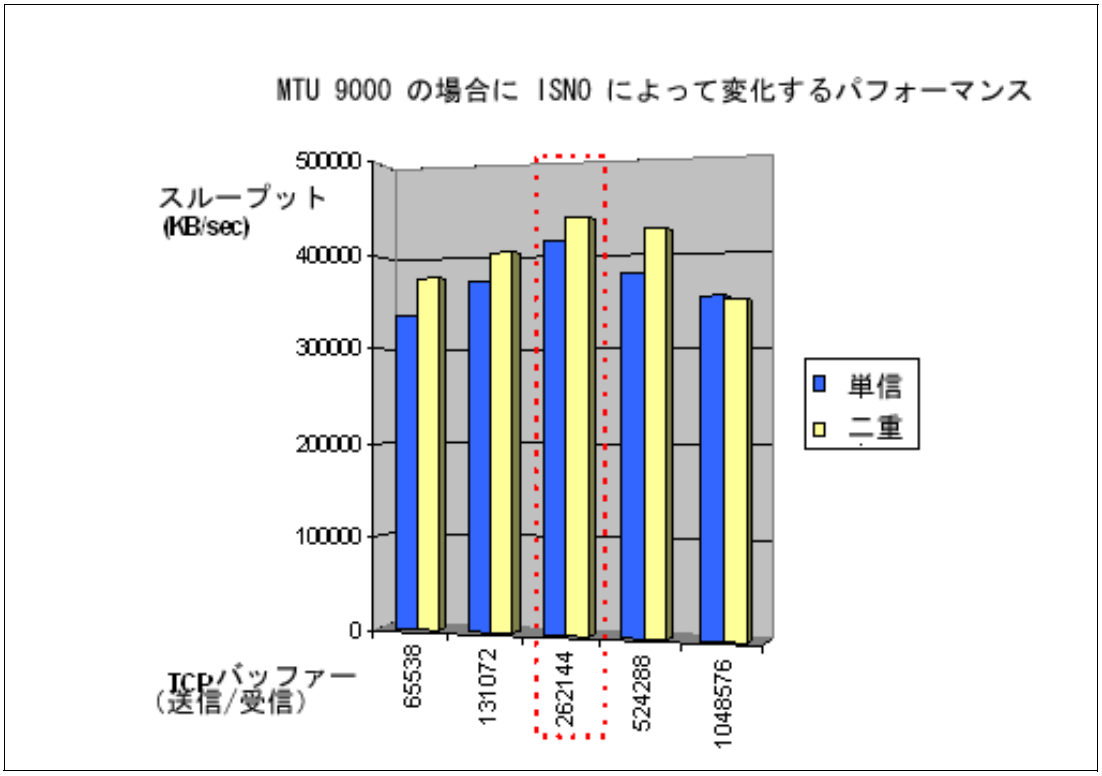


図 3-14 MTU 9000 の場合に ISNO によって変化するパフォーマンス

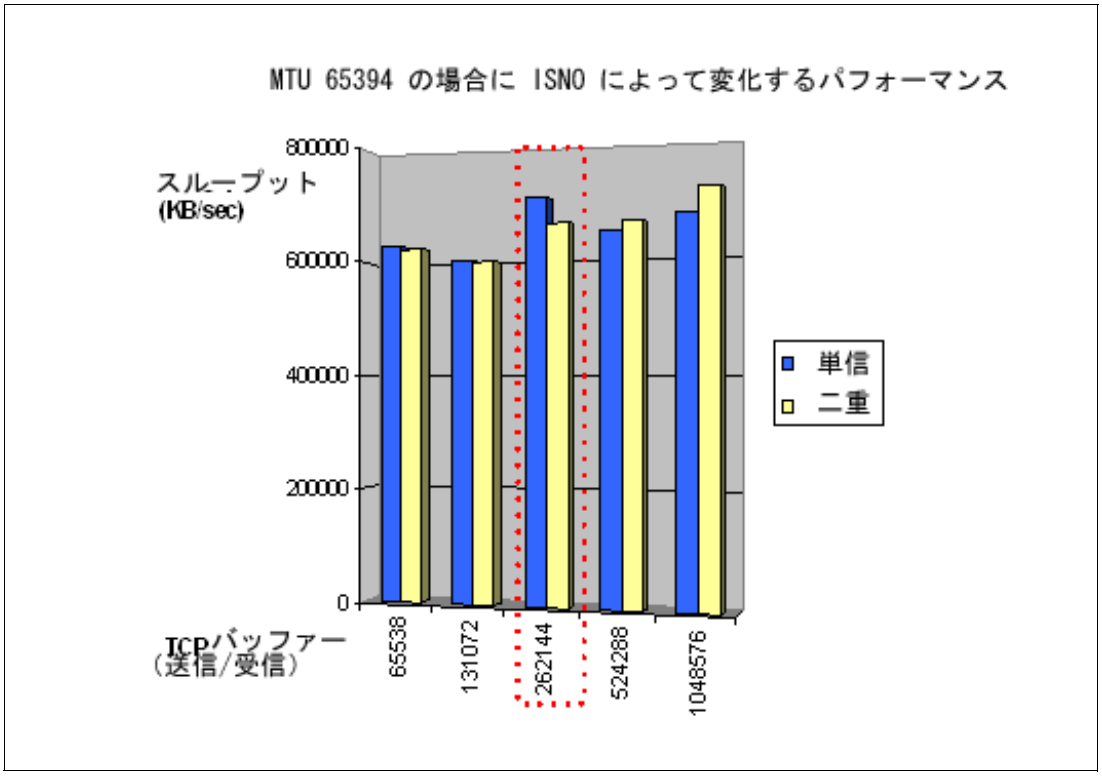


図 3-15 MTU 65394 の場合に ISNO によって変化するパフォーマンス

これらの例からもわかるとおり、MTU と ISNO の設定を調整することによって、ネットワークのパフォーマンスを改善できます。仮想ネットワークでパフォーマンスを最適化するための考慮事項を以下にまとめます。

- 仮想イーサネットのパフォーマンスは、CPU ライセンス容量と TCP/IP パラメーター（MTU サイズ、バッファー・サイズ、rfc1323 設定など）によって左右されます。
- 大きなデータ・パケットがある場合は、MTU サイズとして大きな値を選択することによって、パフォーマンスを改善できます。そうすれば、1 つのパケットで送信できるデータ量が増えるので、データ送信のためのパケット数を減らすことができるからです。
- tcp_pmtu_discover 属性と chksum_offload 属性は、デフォルト値に設定しておきます。
- 同時マルチスレッディングはオフにしません（ただし、アプリケーションのためにその機能をオフにする必要がある場合は例外です）。

3.11.5 TCP チェックサム・オフロード

TCP チェックサム・オフロード・オプションを使用すれば、ネットワーク・アダプターが送信 / 受信時に TCP チェックサムを確認することが可能になるので、ホストの CPU がチェックサムを計算しないですみます。この機能を使用すると、物理アダプターでパケット内のデータの破損を検出できるので、仮想イーサネット・アダプターがチェックサム処理を実行する必要はありません。仮想イーサネット・アダプターのチェックサム・オフロード・オプションは、物理イーサネット・アダプターの場合と同じく、デフォルトで有効になります。仮想イーサネット・アダプター同士の間でパフォーマンスを最適化するには、ソース・システムと宛先システムの両方でチェックサム・オフロード・オプションを有効にします。ソース・システムと宛先システムのいずれかでチェックサム・オフロード・オプションが無効になっていると、ハイパーバイザーがその状態を検出し、必要な場合にチェックサムを確認します。

3.11.6 大規模送信（large_send）オプション

IBM System p のギガビット以上のイーサネット・アダプターは、TCP セグメンテーション・オフロードをサポートしています（この機能は、大規模送信ともいいます）。この機能は、TCP の大規模送信機能を仮想イーサネット・アダプターと共用イーサネット・アダプター（SEA）に拡張したものです。大規模送信環境では、TCP が宛先のアダプターで大規模送信がサポートされていることを検出すると、そのアダプターに対してデータの大きな*チャンク*を送信します。そのアダプターは、発信 MTU のサイズに合わせて、その大きな TCP パケットを複数の小さな TCP パケットに分割することによって、システムの CPU の負荷を削減し、ネットワークのスループットを改善します。

TCP の大規模送信機能は、LPAR から VIOS の実アダプターにまで拡張されています。LPAR の TCP スタックは、VIOS で大規模送信がサポートされているかどうかを確認します。VIOS で TCP の大規模送信がサポートされていれば、LPAR は大きな TCP パケットを VIOS に直接送信します。

ただし、LPAR-LPAR 環境で仮想イーサネット・アダプターを使用している場合は、大きな TCP パケットを複数の小さなパケットに分割する必要がありません。基盤になっているハイパーバイザーが、LPAR 間で大きなデータ・チャンクを送信する処理を担当するからです。

この機能を使用すれば、LPAR-LPAR 通信で大きな MTU を設定できるので、結果として、CPU の大幅な節約とネットワーク・スループットの改善が可能になります。

Virtual I/O Server Version 1.3 では、SEA をブリッジ・デバイスとして使用する場合に、SEA で large_send オプションを使用できます。ただし、Virtual I/O Server Version 1.3 でも、SEA インターフェースから発信されるパケット（仮想 I/O サーバー自体から送られてくるパケット）に関して large_send オプションを使用することはできません。

SEA で大規模送信機能を使用するには、LPAR のすべての仮想アダプターで large_send 属性が値 1 になっていることを確認する必要があります。

Virtual I/O Server Version 1.3 を使用している場合は、例 3-4 のようにして、SEA の large_send オプションを確認できます。この例では、オフになっています。

例 3-4 SEA の large_send オプション

```
$ lsdev -dev ent6 -attr
attribute     value    description
user_settable

ctl_chan              Control Channel adapter for SEA failover
True
ha_mode       disabled High Availability Mode
True
large_send     0        Enable HardwareTransmit TCP Resegmentation
True
netaddr       0        Address to ping
True
pvid          1        PVID to use for the SEA device
True
pvid_adapter  ent5     Default virtual adapter to use for non-VLAN-tagged packets
True
real_adapter  ent0     Physical adapter associated with the SEA
True
thread        1        Thread mode enabled (1) or disabled (0)
True
virt_adapters ent5     List of virtual adapters associated with the SEA (comma separated)
True
```

ネットワークのパフォーマンス調整の詳細については、IBM eServer pSeries および AIX インフォメーション・センターを参照してください。

http://publib16.boulder.ibm.com/pseries/index.htm

追加のパフォーマンス・テストの結果やさらに詳しい推奨事項については、*Advanced POWER Virtualization on IBM* Eserver *p5 Servers: Architecture and Performance Considerations*（SG24-5768）を参照してください。

4

ストレージ

仮想 I/O サーバーは、物理ストレージを仮想 I/O クライアントにマップします。この章では、仮想環境でストレージを管理し、物理ストレージを追跡管理し、物理ストレージを仮想 I/O クライアントに割り当てるためのベスト・プラクティスを取り上げます。

ここで取り上げるのは、ディスクの交換や、エクスポートした論理ボリュームのサイズの拡張などの保守作業のシナリオです。

この章では、移行作業のシナリオについても取り上げます。例えば、仮想ストレージが含まれているパーティションをサーバー間で移動したり、可能な場合に既存のストレージ（物理ストレージや専用ストレージなど）を仮想環境に移動したりする作業があります。

4.1 VIOS の物理ストレージの管理とエクスポート

SAN を使用して、ストレージ要件を管理します。

仮想 I/O サーバー（VIOS）は、ディスク・ストレージを仮想 SCSI ディスクとして仮想 I/O クライアント（VIOC）に提供します。VIOS では、これらの仮想ディスクを物理ストレージにマップする必要があります。このマッピングを実行するには 3 種類の方法があり、それぞれに利点があります。

- 物理ボリューム
- 論理ボリューム
- ストレージ・プール

これらのオプションのうちのどれを選択するかについては、一般的な規則があります。まず、SAN 経由でアクセスするディスク・デバイスは、SAN でストレージ割り振りを管理するために、物理ボリュームとしてエクスポートします。一方、内部のディスク・デバイスや SCSI 接続のディスク・デバイスは、サーバー内でストレージを割り振るために、論理ボリュームまたはストレージ・プールでエクスポートします。この後のセクションでは、それぞれのオプションを詳しく取り上げます。

4.1.1 物理ボリューム

仮想 I/O サーバーは、物理ボリュームをそのまま仮想 I/O クライアントにエクスポートできます。このストレージ・エクスポート方式は、論理ボリュームの方式に比べていくつかの利点があります。

- 物理ディスク・デバイスは、マルチパスの冗長性を確保するために、複数の仮想 I/O サーバーから同時にエクスポートすることが可能です。
- 物理ボリュームをエクスポートするためのコード・パスは短くてすむので、パフォーマンスの向上につながります。
- 物理ディスク・デバイスは、比較的簡単に仮想 I/O サーバー間で移動できます。
- 場合によっては、データを破損しないで物理サーバーの既存の LUN を仮想環境に移行することが可能です。
- 物理ボリュームをエクスポートする場合は、1 つの考慮事項があります。つまり、仮想 I/O サーバーでは、デバイスのサイズを管理することや、1 つのデバイスを複数のクライアントの間で区分化することができません。ただしこれは、基本的に内部のディスクや SCSI 接続のディスクに関する問題です。

SAN 接続のディスクを分割することは基本的に必要ありません。ストレージ割り振りは、ストレージ・サーバーで管理できるからです。SAN 環境では、ストレージ・サーバーの側で各 LPAR の LUN のプロビジョニングと割り振りを行い、仮想 I/O サーバーから物理ボリュームとして LUN をエクスポートします。

SAN ディスクを使用できる場合、仮想 I/O クライアントに関連したストレージは、rootvg やページング・スペースも含めて、すべて SAN に格納します。そのようにすれば、管理作業がシンプルになります。各パーティションが内部の論理ボリュームと外部の LUN の両方に依存することはなくなるからです。また、仮想 I/O サーバー間で LPAR を移動する作業も容易になります。詳しくは、4.6、『サーバー間の AIX 5L LPAR の移動』（114 ページ）を参照してください。

4.1.2 論理ボリューム

仮想 I/O サーバーは、論理ボリュームを仮想 I/O クライアントにエクスポートできます。この方式は、物理ボリュームの方式に比べていくつかの利点があります。

- 論理ボリュームでは、クライアント間で物理ディスク・デバイスを分割することが可能です。
- 論理ボリューム・インターフェースは、AIX 5L のユーザーにとってなじみやすい設計になっています。

注 : エクスポートした論理ボリュームのホストとして仮想 I/O サーバーの rootvg を使用する構成は、現在お勧めしていません。rootvg 内の論理ボリュームについては、論理ボリュームとターゲット・デバイスの間の対応関係が、ある種のソフトウェア・アップグレードやシステム・リストアによって変更される可能性があり、その場合は手操作による介入が必要になります。

冗長性を確保する必要がある場合は、仮想 I/O クライアントで LVM のミラーリングを使用することを推奨します。

内部のディスクまたは SCSI 接続のディスクを使用する場合は、論理ボリューム・マネージャーによって、仮想 I/O クライアント間でディスク・デバイスを分割することが可能になります。小規模なサーバーの場合は、複数の LPAR が内部のディスクまたは RAID アレイを共用できます。

論理ボリュームの場合は、複数の仮想 I/O サーバーから同時にアクセスすることができないので、仮想 I/O クライアントでマルチパス I/O（MPIO）構成にすることはできません。

4.1.3 ストレージ・プール

Integrated Virtualization Manager（IVM）で管理する場合は、仮想 I/O サーバーによって、ストレージ・プール・バッキング・デバイスを仮想 I/O クライアントにエクスポートできます。この方式は、論理ボリュームの方式とよく似ており、物理ボリュームの方式に比べていくつかの利点があります。

- ストレージ・プール・バッキング・デバイスでは、クライアント間で物理ディスク・デバイスを分割することが可能です。
- ストレージ・プール・インターフェースは、IVM からも使いやすい設計になっています。

重要 : IVM のデフォルトのストレージ・プールは、仮想 I/O サーバーのルート・ボリューム・グループです。rootvg 内の論理ボリュームについては、論理ボリュームとターゲット・デバイスの間の対応関係が、ある種のソフトウェア・アップグレードやシステム・リストアによって変更される可能性があり、その場合は手操作による介入が必要になるので、ルート・ボリューム・グループ内のバッキング・デバイスを割り当てることがないように注意しなければなりません。

IVM で管理する単一サーバー環境のシステムは、SAN に接続しない場合が多く、通常は内部のディスク・ストレージや SCSI 接続のディスク・ストレージを使用します。この場合は、IVM インターフェースを使用して、物理ストレージ・デバイスでストレージ・プールを作成できるので、1 つの物理ディスク・デバイスを複数の仮想 I/O クライアントの間で分割することが可能になります。

論理ボリュームの場合と同じく、ストレージ・プール・バッキング・デバイスも、複数の仮想 I/O サーバーから同時にアクセスすることができないので、仮想 I/O クライアントで MPIO と連係させることも不可能です。

冗長性を確保する必要がある場合は、仮想 I/O クライアントで LVM のミラーリングを使用することを推奨します。

4.1.4 論理ボリュームをエクスポートするためのベスト・プラクティス

IVM 管理環境と HMC 管理環境とでは、ストレージ管理のための同じようなインターフェースの名前がそれぞれ異なります。IVM のストレージ・プール・インターフェースと HMC の論理ボリューム・マネージャー・インターフェースは基本的に同じものであり、文書の中では、それぞれの用語を同義語として使用する場合もあります。この後のセクションでは、ボリューム・グループとストレージ・プールの両方を指してボリューム・グループという語を使用し、論理ボリュームとストレージ・プール・バッキング・デバイスの両方を指して論理ボリュームという語を使用します。

論理ボリュームを使用すれば、仮想 I/O サーバーによって、複数の仮想 I/O クライアントの間で物理ボリュームを分割することが可能になります。多くの場合、物理ボリュームとしては、内部のディスク、または内部のディスクで構成した RAID アレイを使用します。

1 つのボリューム・グループの中に、仮想 I/O クライアントで使用する論理ボリュームと、仮想 I/O サーバー・オペレーティング・システムで使用する論理ボリュームの両方を組み込むべきではありません。仮想 I/O サーバーの systems ファイルは rootvg 内に保持し、仮想 I/O クライアントの論理ボリュームのホストとしては他のボリューム・グループを使用します。

1 つのボリューム・グループまたは論理ボリュームに 2 つの仮想 I/O サーバーから同時にアクセスすることはできません。仮想 I/O クライアントでは、論理ボリューム上の VSCSI デバイスを MPIO 構成しないでください。論理ボリューム構成で冗長性を確保する必要がある場合は、仮想 I/O クライアントで LVM のミラーリングを使用し、別々の仮想 I/O サーバーで別々の論理ボリュームを構成する形でミラーを構築します。

複数の物理ボリュームにまたがる論理ボリュームもサポートされていますが、パフォーマンスを最適化するためには、論理ボリューム全体を 1 つの物理ボリュームに配置するのが望ましいといえます。そのためには、ボリューム・グループを 1 つの物理ボリュームで構成します。

重要 : エクスポートしたストレージ・プール・バッキング・デバイスまたは論理ボリュームを 1 つの hdisk で保持すれば、パフォーマンスが最適化されます。

論理ボリュームをクライアントにエクスポートするときには、個々の論理ボリュームと仮想 I/O クライアントの間の対応関係を VIOS で管理します。論理ボリューム・マネージャーによって追加の抽象レベルが提供されるので、物理ディスク・デバイスと仮想 I/O クライアントの間の関係を追跡管理することは重要です。詳しくは、4.3、『LUN と VSCSI と hdisk の対応関係の管理』（101 ページ）を参照してください。

4.2 仮想ストレージ・デバイスのサイズの拡張

仮想 I/O サーバーでエクスポートするストレージ・デバイスは、論理ボリュームの場合もあれば、物理ボリュームの場合もあります。エクスポートする論理ボリュームのサイズは、仮想 I/O サーバーで拡張できます。また、SAN を基盤とした多くの種類の物理ボリュームについても、拡張が可能です。ご使用のストレージ・サブシステムで既存の LUN のサイズの拡張がサポートされているかどうかについては、各ストレージのベンダーに問い合わせてください。

ここでは、エクスポートする論理ボリュームを拡張して、その変更を仮想 I/O クライアントで認識させるための手順の概略を説明します。仮想 I/O サーバーでは、論理ボリュームのサイズを変更するための追加のコマンドがありますが、仮想 I/O クライアントでは、内部のデバイスの変更についても外部のデバイスの変更についても同じコマンドを使用します。

注 : クラシック並行モードまたは拡張並行モードでボリューム・グループがアクティブ化されている AIX 5L の rootvg にある物理ボリュームは、サイズ変更ができません。rootvg を拡張するには、物理ボリュームを追加します。

SAN でボリュームを拡張した場合も、仮想 I/O サーバーの論理ボリューム・マネージャーでボリュームを拡張した場合も、管理者はその拡張操作の後に、仮想 I/O クライアントで **chvg -g** コマンドを実行する必要があります（AIX 5L の場合）。**chvg** コマンドでは、ボリューム・グループ内のすべてのディスクについて、サイズが拡張されているかどうかを確認できます。LVM のボリューム・グループでは、ディスクのサイズ変更を確認するために、**varyoffvg** コマンドと **varyonvg** コマンドを実行しなければならない場合もあります。

仮想 I/O サーバーで論理ボリュームを拡張し、仮想 I/O クライアントでその変更を認識するには、以下の手順を実行します。

1. 例 4-1（99 ページ）は、論理ボリュームの状況と、仮想 I/O サーバーで論理ボリュームのサイズを拡張する方法を示しています。論理ボリューム（LV）のサイズを変更するには、以下のようにして **extendlv** コマンドを使用します。

   ```
   extendlv  LogicalVolume Size [PhysicalVolume ...]
   ```

例 4-1　VIOS での LV の状況の確認と LV のサイズの拡張

```
$ lslv db_lv
LOGICAL VOLUME:     db_lv                  VOLUME GROUP:   db_sp
LV IDENTIFIER:      00cddeec00004c000000000c4a8b3d81.1 PERMISSION:     read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
TYPE:               jfs                    WRITE VERIFY:   off
MAX LPs:            32512                  PP SIZE:        32 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                320                    PPs:            320
STALE PPs:          0                      BB POLICY:      non-relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       middle                 UPPER BOUND:    1024
MOUNT POINT:        N/A                    LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
DEVICESUBTYPE : DS_LVZ

$ extendlv db_lv 5G

$ lslv db_lv
LOGICAL VOLUME:     db_lv                  VOLUME GROUP:   db_sp
LV IDENTIFIER:      00cddeec00004c000000000c4a8b3d81.1 PERMISSION:     read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
```

```
TYPE:               jfs                     WRITE VERIFY:   off
MAX LPs:            32512                   PP SIZE:        32 megabyte(s)
COPIES:             1                       SCHED POLICY:   parallel
LPs:                480                     PPs:            480
STALE PPs:          0                       BB POLICY:      non-relocatable
INTER-POLICY:       minimum                 RELOCATABLE:    yes
INTRA-POLICY:       middle                  UPPER BOUND:    1024
MOUNT POINT:        N/A                     LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
DEVICESUBTYPE :
DS_LVZ
```

2. 例 4-2 は、**chvg** コマンドを次のように使用して、仮想 I/O クライアントで変更を認識する方法を示しています。

```
chvg -g Volumegroup_name
```

例 4-2 VIOC での LV の変更の認識

```
# lspv hdisk1
PHYSICAL VOLUME:    hdisk1                  VOLUME GROUP:     oravg
PV IDENTIFIER:      00cddeec6828645d VG IDENTIFIER     00cddeec00004c000000000c4a8d4346
PV STATE:           active
STALE PARTITIONS:   0                       ALLOCATABLE:      yes
PP SIZE:            16 megabyte(s)          LOGICAL VOLUMES:  0
TOTAL PPs:          639 (10224 megabytes)   VG DESCRIPTORS:   2
FREE PPs:           639 (10224 megabytes)   HOT SPARE:        no
USED PPs:           0 (0 megabytes)         MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  128..128..127..128..128
USED DISTRIBUTION:  00..00..00..00..00

# chvg -g oravg

# lspv hdisk1
PHYSICAL VOLUME:    hdisk1                  VOLUME GROUP:     oravg
PV IDENTIFIER:      00cddeec6828645d VG IDENTIFIER     00cddeec00004c000000000c4a8d4346
PV STATE:           active
STALE PARTITIONS:   0                       ALLOCATABLE:      yes
PP SIZE:            16 megabyte(s)          LOGICAL VOLUMES:  0
TOTAL PPs:          959 (15344 megabytes)   VG DESCRIPTORS:   2
FREE PPs:           959 (15344 megabytes)   HOT SPARE:        no
USED PPs:           0 (0 megabytes)         MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  192..192..191..192..192
USED DISTRIBUTION:
00..00..00..00..00
```

4.3 LUN と VSCSI と hdisk の対応関係の管理

仮想環境を管理するための重要な要素の 1 つは、仮想オブジェクトと物理オブジェクトの対応関係を追跡管理することです。特に、個々の LPAR で数百個の仮想ディスクを使用するストレージ環境では、この管理が大きな課題になります。このマッピングは、パフォーマンスを管理するためにも、ハードウェアの保守によってどのシステムが影響を受けるのかを把握するためにも重要です。

仮想ディスクと物理ディスクの対応関係を定義するには、2 つの方法があります（図 4-1 (101 ページ)）。

- 物理ボリューム
- 論理ボリューム

論理ボリュームでは、ボリューム・グループまたはストレージ・プールとの対応関係を定義できます。それぞれの環境でどの方式が適切かについては、4.1、『VIOS の物理ストレージの管理とエクスポート』（96 ページ）を参照してください。

ネットワーク・デバイスと仮想 LAN の対応関係については、3.3、『ネットワーク・デバイスのマッピングの管理』（61 ページ）を参照してください。

HMC コマンド **lshwres** を使用すれば、仮想 I/O クライアントと vhost の対応関係やその他の情報が正しく定義されていない場合に、それらを確認することが可能になります。

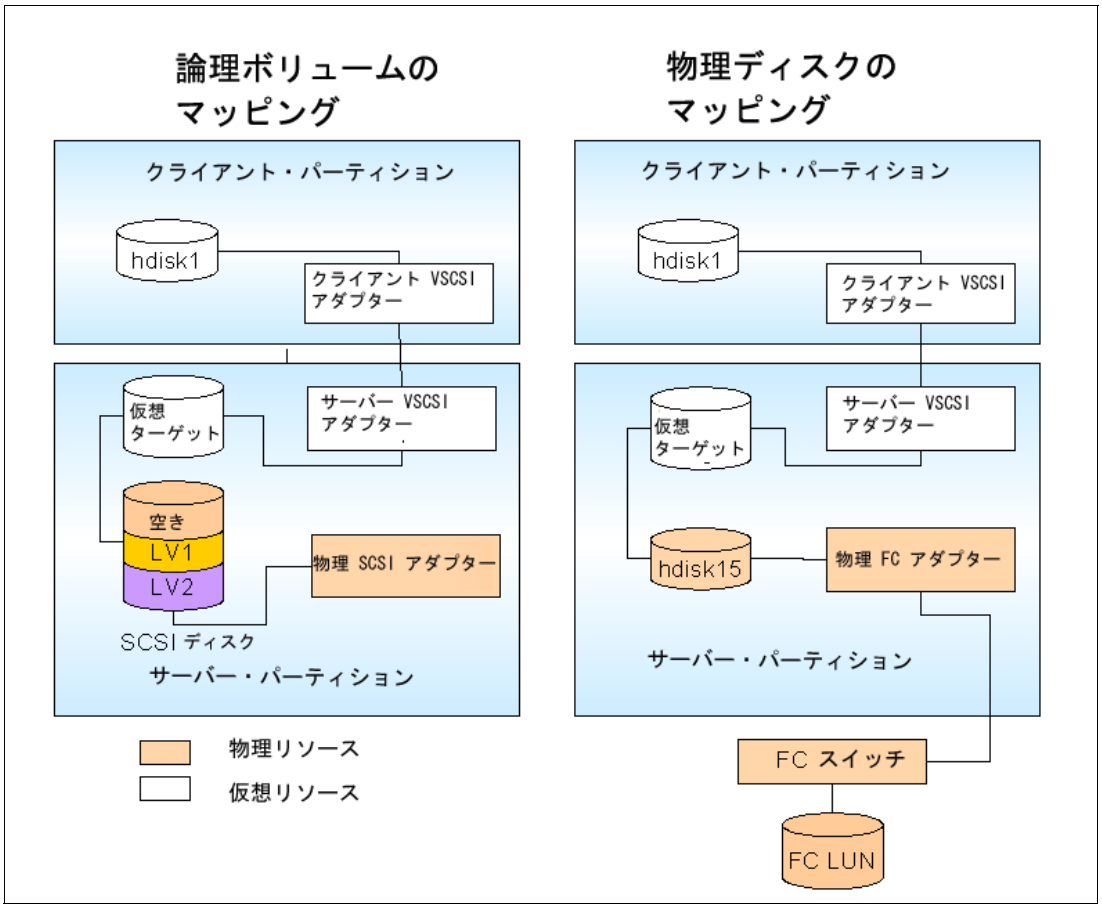


図 4-1　論理ドライブと物理ドライブの対応関係

選択する方式によっては、以下の情報の追跡管理が必要になる場合があります。

- 仮想 I/O サーバー
  - サーバーのホスト名
  - 物理ディスクの場所
  - 物理アダプターのデバイス名
  - 物理 hdisk のデバイス名
  - ボリューム・グループまたはストレージ・プールの名前[1]
  - 論理ボリュームまたはストレージ・プール・バッキング・デバイスの名前[1]
  - VSCSI アダプターのスロット
  - VSCSI アダプターのデバイス名
  - 仮想ターゲット・デバイス
- 仮想 I/O クライアント
  - クライアントのホスト名
  - VSCSI アダプターのスロット
  - VSCSI アダプターのデバイス名
  - 仮想 hdisk のデバイス名

追跡管理するフィールドの数からして、この種の情報の管理には、図 4-2（102 ページ）のようなスプレッドシートまたはデータベースのプログラムを使用することをお勧めします。システムのインストール時にデータを記録し、その後は構成変更のたびにデータを追跡管理するようにしてください。

Microsoft Excel - diskmap.xls

File Edit View Insert Format Tools Data Window Help Adobe PDF

Type a question for help

	B	C	D	E	F	G	H	I	J	K
1	Hostname	VIOS Host	VIOS VG	VIOS Device	VIOS LV Name	VSCSI Name	VIOS vhost	VIOS Slot	Client Slot	Physical Disk Location
2	server1	**vios1**	mapvg1	hdisk3	server1_rootvg1	srv1_rvg1_dev	vhost0	21	21	U787B.001.DNW0974-P1-
3		**vios2**	mapvg2	hdisk3	server1_rootvg2	srv1_rvg2_dev	vhost0	22	22	U787B.001.DNW0974-P1-
4		**vios1**	mapvg1	hdisk4	server1_datavg1	srv1_dvg1_dev	vhost0	21	21	U787B.001.DNW0974-P1-
5		**vios2**	mapvg2	hdisk4	server1_datavg2	srv1_dvg2_dev	vhost0	22	22	U787B.001.DNW0974-P1-
6	server2	**vios1**	mapvg1	hdisk3	server2_rootvg1	srv2_rvg1_dev	vhost1	23	23	U787B.001.DNW0974-P1-
7		**vios2**	mapvg2	hdisk3	server2_rootvg2	srv2_rvg2_dev	vhost1	24	24	U787B.001.DNW0974-P1-
8		**vios1**	mapvg1	hdisk5	server2_datavg1	srv2_dvg1_dev	vhost1	23	23	U787B.001.DNW0974-P1-
9		**vios2**	mapvg2	hdisk5	server2_datavg2	srv2_dvg2_dev	vhost1	24	24	U787B.001.DNW0974-P1-
10	server3	**vios1**	mapvg1	hdisk3	server3_rootvg1	srv3_rvg1_dev	vhost2	25	25	U787B.001.DNW0974-P1-
11		**vios2**	mapvg2	hdisk3	server3_rootvg2	srv3_rvg2_dev	vhost2	26	26	U787B.001.DNW0974-P1-
12		**vios1**	mapvg1	hdisk6	server3_datavg1	srv3_dvg1_dev	vhost2	25	25	U787B.001.DNW0974-P1-
13		**vios2**	mapvg2	hdisk6	server3_datavg2	srv3_dvg2_dev	vhost2	26	26	U787B.001.DNW0974-P1-
14	server4	**vios1**	mapvg1	hdisk3	server4_rootvg1	srv4_rvg1_dev	vhost3	27	27	U787B.001.DNW0974-P1-
15		**vios2**	mapvg2	hdisk3	server4_rootvg2	srv4_rvg2_dev	vhost3	28	28	U787B.001.DNW0974-P1-
16		**vios1**	mapvg1	hdisk7	server4_datavg1	srv5_dvg1_dev	vhost3	27	27	U787B.001.DNW0974-P1-

Virtual Disk Map

図 4-2　ディスクの追跡管理のためのスプレッドシート

4.3.1 命名規則

エンタープライズの命名規則を作成します。

この種の情報の管理では、スプレッドシートなどの追跡管理ツールのほかに、適切な命名規則も重要です。追跡管理しなければならないデータの量を減らすための 1 つの方法は、仮想 I/O クライアントとサーバーの間で、デバイスの名前とスロットをできるだけ一致させるということです。

[1] 論理ボリュームまたはストレージ・プールのエクスポートのみ

このような命名規則には、対応するボリューム・グループ、論理ボリューム、仮想ターゲット・デバイスの名前を含めることもできます。仮想 I/O クライアントのホスト名を仮想ターゲット・デバイス名に統合すれば、サーバーでの追跡管理が容易になります。

LUN の命名をサポートするストレージ・サーバーでファイバー・チャネル・ディスクを使用する場合は、LUN を識別しやすくするためにその機能を使用できます。以下の例のようにして、IBM System Storage™ の DS8000™ シリーズと DS6000™ シリーズでは **lssdd** コマンド、DS4000™ シリーズでは **fget_config** コマンドをそれぞれ使用して、hdisk デバイスと LUN の名前を一致させることができます。ただし、**fget_config** コマンドはストレージ・デバイス・ドライバーの一部なので、仮想 I/O サーバーでそのコマンドを実行するには、**oem_setup_env** コマンドを使用しなければなりません。

```
$ oem_setup_env
# fget_config -Av

---dar0---

User array name = 'FAST200'
dac0 ACTIVE dac1 ACTIVE

Disk     DAC   LUN Logical Drive
utm             31
hdisk4   dac1    0 Server1_LUN1
hdisk5   dac1    1 Server1_LUN2
hdisk6   dac1    2 Server-520-2-LUN1
hdisk7   dac1    3 Server-520-2-LUN2
```

基本的には、ファイバー・チャネルのワールド・ワイド・ポート名や数値の LUN ID を使用してデバイスをトレースするよりも、LUN 名を使用するほうが簡単です。

4.3.2 仮想デバイスのスロット番号

エンタープライズのスロット番号の番号付けポリシーを作成します。

命名規則を設定したら、仮想 I/O アダプターのスロット番号の番号付け規則も設定します。

仮想ストレージ・デバイスと仮想ネットワーク・デバイスは、スロット番号を共用します。複雑なシステムでは、ネットワーク・デバイスよりもストレージ・デバイスのほうがはるかに多くなる傾向があります。それぞれの仮想 SCSI デバイスは、1 つのサーバーまたはクライアントとしか通信できないからです。ネットワーク・デバイスとストレージ・デバイスを 1 つのグループとしてまとめるために、すべての LPAR のネットワーク・デバイス用として 20 までのスロット番号を予約しておくことをお勧めします。

仮想 I/O クライアントとサーバーの間でスロット番号を統一すれば、管理作業は容易になります。ただし、サーバー間でパーティションを移動すると、それが不可能になる場合もあります。詳しくは、4.6.3、『ストレージ計画』（115 ページ）を参照してください。

単一仮想 I/O サーバー環境では、ストレージ・アダプターのスロット番号を 21 から順番に 1 ずつ大きくしていきます。2 つの仮想 I/O サーバーにクライアントを接続している場合は、アダプターのスロット番号を VIOS ごとに交互に付けていきます。つまり、第 1 の VIOS では 21 を開始点として奇数のスロット番号を使用し、第 2 の VIOS では 22 を開始点として偶数のスロット番号を使用します。二重サーバーのシナリオでは、各クライアントでスロット番号がペアになるように、例えば 21 と 22、33 と 34 といった具合に隣接する番号を割り振ってください。

1 つの LPAR だけを作成する場合は、LPAR の仮想アダプター・スロットの最大数をデフォルト値の 10 よりも大きくする必要があります（図 4-3（104 ページ））。それぞれの環境に適した値は、各システムで使用する LPAR の数とアダプターの数によって左右されます。仮想アダプター・スロットを使用しないままにしておくと、少量のメモリーを消費することになるので、割り振りのバランスを考えなければなりません。それぞれのシステム構成で必要な

メモリー所要量を計画するときには、以下の URL に用意されている System Planning Tool を使用してください。

http://www.ibm.com/servers/eserver/iseries/lpar/systemdesign.html

重要 : LPAR の仮想 I/O スロットの数について計画するときには、パーティションで使用できる仮想アダプター・スロットの最大数をパーティションのプロファイルで設定します。このプロファイルを変更するには、LPAR をシャットダウンする必要があります。スロットの最大数を設定する場合は、新しい仮想 I/O クライアントを追加するときに、LPAR や仮想 I/O サーバー・パーティションをシャットダウンしないですむように、拡張のための余地を十分に残しておくことをお勧めします。

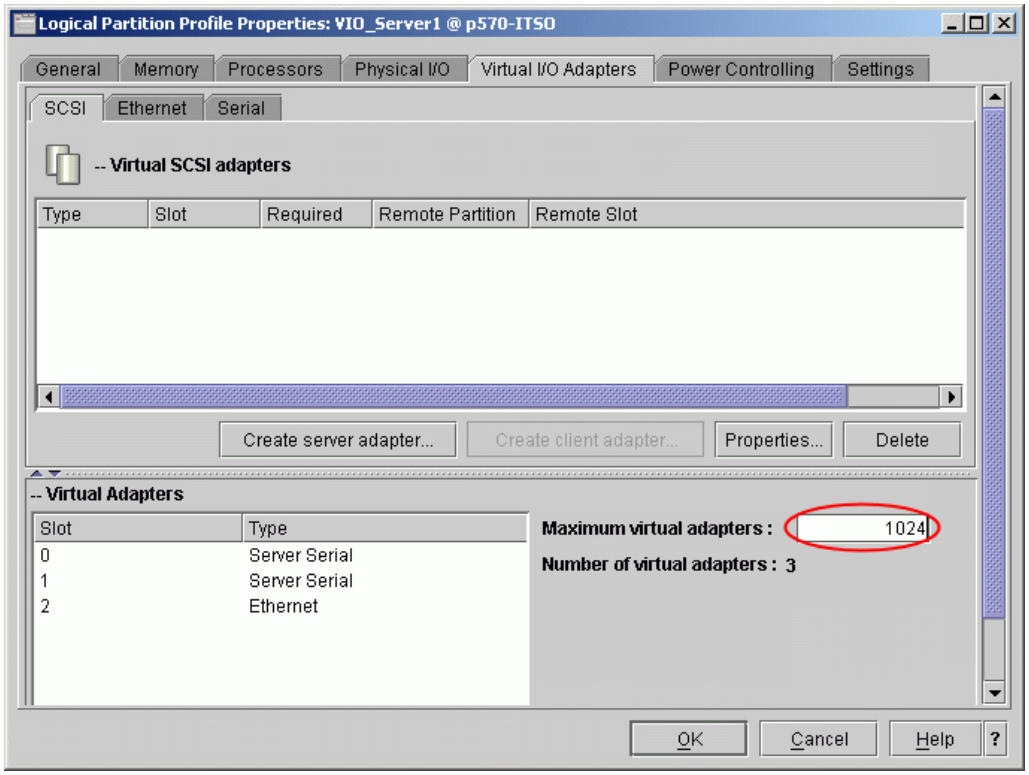


図 4-3　仮想アダプターの最大数

VSCSI 接続はメモリーの速度で作動するので、仮想 I/O サーバーとクライアントの間に複数の仮想 I/O アダプターを追加しても、基本的にパフォーマンスの向上は望めません。デフォルトのキュー・エントリー数である 3 が設定されている状況では、各アダプター・ペアで最大 85 個の仮想デバイスを操作できます。パーティションごとの仮想デバイスがその数を超えそうな場合や、一部のデバイスのキュー・エントリー数がデフォルトを上回りそうな場合は、サーバーごとに追加のアダプター・スロットを 1 つ確保します（つまり、クライアントが通信する VIOS ごとに追加のスロットを 1 つ確保します）。

4.3.3 構成のトレース

記録の保持に万全を期したシステムでも、クライアント仮想ディスクを物理ハードウェアに手動でトレースバックすることが必要になる場合もあります。以下のサイトにある IBM Systems ハードウェア Information Center には、仮想ディスクのトレースに関するガイドが用意されています。

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphb1/iphb1_vios_managing_mapping.htm

4.4 仮想 I/O サーバーでのディスクの交換

仮想 I/O サーバーでディスクを交換することが必要になった場合は、まず影響を受ける仮想 I/O クライアントとターゲット・ディスク・ドライブを識別しなければなりません。

ヒント : この手順でディスク・デバイスを交換する前に、ディスクのホット・スワップが可能かどうかを確認してください。

ホット・スワップは、ハードウェアだけの問題ではありません。アクティブなシステムとの調整が必要になります。

仮想 I/O クライアントでディスクのミラーリングを構成していない単一仮想 I/O サーバー環境の場合は、ディスクを交換するときに、データのリストアが必要になります。MPIO を使用して 2 つの仮想 I/O サーバーで同じディスクをエクスポートする場合にも、この条件は当てはまります。MPIO それ自体は、ディスク交換時の停止に対する保護機能になりません。単一仮想 I/O サーバーや二重仮想 I/O サーバーを使用する場合でも、MPIO を使用する場合でも、ディスクまたは LUN のデータを保護するには、ミラーリングまたは RAID のテクノロジーを使用する必要があります。

ディスク・ストレージを仮想 I/O クライアントにエクスポートするには、3 つの方法があります。詳細については、4.1、『VIOS の物理ストレージの管理とエクスポート』（96 ページ）を参照してください。このセクションでは、論理ボリューム環境とストレージ・プール環境でディスク・デバイスを交換するための手順の概要を説明します。

論理ボリューム環境でもストレージ・プール環境でも、物理ボリュームの状況を確認するには、以下の例のようにして **lsvg -pv** コマンドを使用します。

```
#lsvg -pv vioc_rootvg
vioc_rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk3            active            1092        580         219..00..00..142..219
```

注 : ディスクを交換する前に、仮想 I/O クライアント、論理ボリューム（LV）、バッキング・デバイス、関連する vhost と vtscsi、vtscsi デバイスのサイズを文書化しておいてください。この管理の詳細については、4.3、『LUN と VSCSI と hdisk の対応関係の管理』（101 ページ）を参照してください。

4.4.1 LVM 環境でのディスクの交換

この LVM のシナリオでは、ボリューム・グループ vioc_rootvg_1 の hdisk2 を交換します。そのボリューム・グループでは、LV の vioc_1_rootvg が vtscsi0 に関連付けられています。属性は以下のとおりです。

- サイズは 32 GB です。
- 仮想ディスクは、仮想 I/O クライアントで LVM のミラーリングが構成されています。
- 仮想 I/O クライアントの障害ディスクは、hdisk1 です。
- 仮想 I/O クライアントの仮想 SCSI アダプターは、vscsi1 です。
- 仮想 I/O クライアントのボリューム・グループは、rootvg です。

図 4-4 は、このセットアップを示したものです。

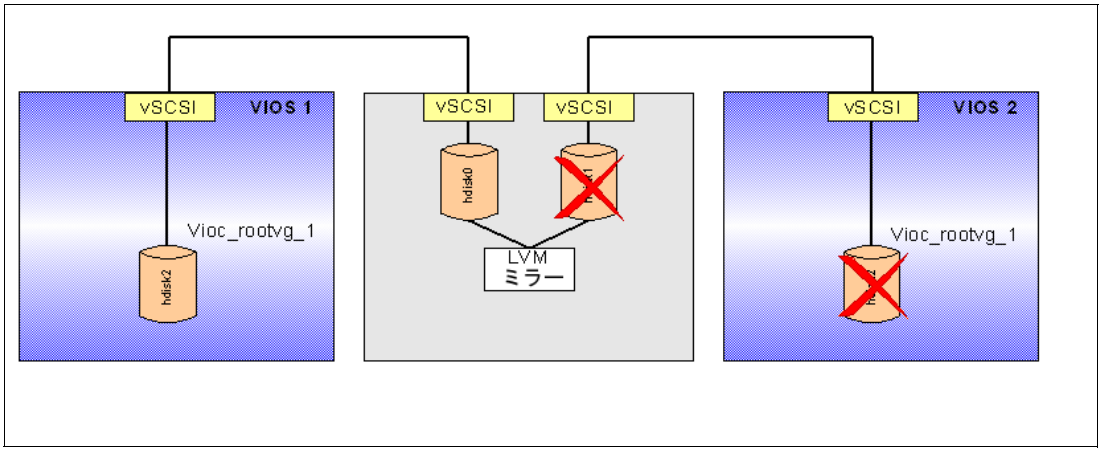


図 4-4 LVM 環境

ディスクを交換するには、以下のようにします。

1. **diagmenu** コマンドによって物理ディスク・ドライブを識別します。
2. 「**Task Selection (Diagnostics, Advanced Diagnostics, Service Aids, etc.)**」を選択します。次のリストで「**Hot Plug Task**」を選択します。
3. このリストで、「**SCSI and SCSI RAID Hot Plug Manager**」を選択し、「**Identify a Device Attached to a SCSI Hot Swap Enclosure Device**」を選択します。
4. 次のリストで hdisk を見つけて、Enter を押します。図 4-5 のような画面が表示されます。ただしこれは、内部のディスクを選択した場合の p5-570 の例です。

```
IDENTIFY DEVICE ATTACHED TO SCSI HOT SWAP ENCLOSURE DEVICE                802483

The following is a list of devices attached to SCSI Hot Swap Enclosure devices.
Selecting a slot will set the LED indicator to Identify.

Make selection, use Enter to continue.

[MORE...4]
     slot  3+-----------------------------------------------------+
     slot  4|                                                     |
     slot  5|                                                     |
     slot  6| The LED should be in the Identify state for the     |
            | selected device.                                    |
            |                                                     |
  ses1      | Use 'Enter' to put the device LED in the            |
     slot  1| Normal state and return to the previous menu.       |
     slot  2|                                                     |
     slot  3|                                                     |
     slot  4|                                                     |
     slot  5|                                                     |
     slot  6|                                                     |
[BOTTOM]    |                                                     |
            | F3=Cancel        F10=Exit         Enter             |
F1=Help     +-----------------------------------------------------+
```

図 4-5 取り外すディスクの検出

5. 仮想 I/O クライアントで、以下のようにして rootvg のミラーリング解除を実行します。

```
# unmirrorvg -c 1 rootvg hdisk1
0516-1246 rmlvcopy: If hd5 is the boot logical volume, please run 'chpv -c <diskname>'
        as root user to clear the boot record and avoid a potential boot
        off an old boot image that may reside on the disk from which this
        logical volume is moved/removed.

0301-108 mkboot: Unable to read file blocks. Return code: -1
0516-1132 unmirrorvg: Quorum requirement turned on, reboot system for this
        to take effect for rootvg.
0516-1144 unmirrorvg: rootvg successfully unmirrored, user should perform
        bosboot of system to reinitialize boot records.  Then, user must modify
        bootlist to just include:  hdisk0.
```

6. 仮想 I/O クライアントで、rootvg から hdisk1 を除去します。

```
# reducevg rootvg hdisk1
#
```

7. 仮想 I/O クライアントで、hdisk1 デバイスを除去します。

```
# rmdev -l hdisk1 -d
hdisk1 deleted
#
```

8. 仮想 I/O サーバーで、vtscsi/vhost の関連付けを削除します。

```
$ rmdev -dev vtscsi0
vtscsi0 deleted
$
```

9. 仮想 I/O サーバーで、ボリューム・グループを削減します。以下の例のようにエラーが発生した場合でも、ボリューム・グループごとに 1 つの hdisk しかないことが間違いなければ、**deactivatevg** コマンドと **exportvg** コマンドを使用できます。

注 : exportvg コマンドを使用すると、ボリューム・グループ内のすべての論理ボリュームが削除されます。ボリューム・グループに複数の hdisk が含まれている場合は、この hdisk の論理ボリュームも操作の対象になります。確認のためには、**lspv** コマンドを使用できます。この場合は、**exportvg vioc_rootvg_1** コマンドを使用するほうが無難です。

```
$ lspv
NAME            PVID                                VG              STATUS
hdisk0          00c478de00655246                    rootvg          active
hdisk1          00c478de008a399b                    rootvg          active
hdisk2          00c478de008a3ba1                    vioc_rootvg_1   active
hdisk3          00c478deb4b0d4b0                    None
$
```

```
$ reducevg -rmlv -f vioc_rootvg_1 hdisk2


Some error messages may contain invalid information
for the Virtual I/O Server environment.

0516-062 lqueryvg: Unable to read or write logical volume manager
        record. PV may be permanently corrupted. Run diagnostics
0516-882 reducevg: Unable to reduce volume group.
$ deactivatevg vioc_rootvg_1


Some error messages may contain invalid information
```

```
for the Virtual I/O Server environment.

0516-062 lqueryvg: Unable to read or write logical volume manager
        record. PV may be permanently corrupted. Run diagnostics
$ exportvg vioc_rootvg_1
$
```

10. 仮想 I/O サーバーで、hdisk デバイスを除去します。

```
$ rmdev -dev hdisk2
hdisk2 deleted
$
```

11. 物理ディスク・ドライブを交換します。
12. 仮想 I/O サーバーで、**cfgdev** コマンドによって新しい hdisk デバイスを構成し、その新しいディスクが構成されたことを確認します。

```
$ cfgdev
$ lspv
NAME            PVID                                VG              STATUS
hdisk2          none                                None
hdisk0          00c478de00655246                    rootvg          active
hdisk1          00c478de008a399b                    rootvg          active
hdisk3          00c478deb4b0d4b0                    None
$
```

13. ボリューム・グループごとに 1 つのディスクしかない場合は、仮想 I/O サーバーで **mkvg** コマンドを使用して、その新しい hdisk でボリューム・グループを拡張します。ボリューム・グループごとに複数のディスクがある場合は、**extendvg** コマンドを使用します。この場合は、1 つのディスクに 1 つのボリューム・グループが対応しています（これが推奨構成です）。ディスクに PVID がある場合は、**mkvg** コマンドの **-f** フラグを使用します。

```
$ mkvg -vg vioc_rootvg_1 hdisk2
vioc_rootvg_1
0516-1254 mkvg: Changing the PVID in the ODM.
$
```

14. 仮想 I/O サーバーで、vtscsi デバイスの論理ボリュームを作成します。

```
$ mklv -lv vioc_1_rootvg vioc_rootvg_1 32G
vioc_1_rootvg
$
```

15. 仮想 I/O サーバーで、LV が複数のディスクにまたがっていないことを確認します。

```
$ lslv -pv vioc_1_rootvg
vioc_1_rootvg:N/A
PV               COPIES        IN BAND       DISTRIBUTION
hdisk2           512:000:000   42%           000:218:218:076:000
$
```

16. 仮想 I/O サーバーで、仮想デバイスを作成します。

```
$ mkvdev -vdev vioc_1_rootvg -vadapter vhost0
vtscsi0 Available
$
```

17. 仮想 I/O クライアントで、新しい hdisk1 を再構成します。

```
# cfgmgr -l vscsi1
#
```

親デバイスが不明な場合は、以下のようにします。

```
# cfgmgr
#
```

18. 仮想 I/O クライアントで、rootvg を拡張します。

```
# extendvg rootvg hdisk1
0516-1254 extendvg: Changing the PVID in the ODM.
#
```

19. 仮想 I/O クライアントで、rootvg をミラーリングします。

```
# mirrorvg -c 2 rootvg hdisk1
0516-1124 mirrorvg: Quorum requirement turned off, reboot system for this
        to take effect for rootvg.
0516-1126 mirrorvg: rootvg successfully mirrored, user should perform
        bosboot of system to initialize boot records.  Then, user must modify
        bootlist to include:  hdisk0 hdisk1.
#
```

20. 仮想 I/O クライアントで、ブート・レコードを初期化し、ブート・リストを設定します。

```
# bosboot -a

bosboot: Boot image is 18036 512 byte blocks.
# bootlist -m normal hdisk0 hdisk1
#
```

4.4.2 ストレージ・プール環境でのディスクの交換

このストレージ・プールのシナリオでは、ストレージ・プール vioc_rootvg_1 の hdisk2 を交換します。属性は以下のとおりです。

- バッキング・デバイス vioc_1_rootvg が vhost0 に関連付けられています。
- サイズは 32 GB です。
- 仮想ディスクは、仮想 I/O クライアントでミラーリングされています。
- 仮想 I/O クライアントのボリューム・グループは、rootvg です。

ディスクを交換するには、以下の手順を実行します。

1. 物理ディスク・ドライブを識別します。1（106 ページ）の手順を参照してください。
2. 仮想 I/O クライアントで、rootvg のミラーリング解除を実行します。

```
# unmirrorvg -c 1 rootvg hdisk1
0516-1246 rmlvcopy: If hd5 is the boot logical volume, please run 'chpv -c <diskname>'
        as root user to clear the boot record and avoid a potential boot
        off an old boot image that may reside on the disk from which this
        logical volume is moved/removed.

0301-108 mkboot: Unable to read file blocks. Return code: -1
0516-1132 unmirrorvg: Quorum requirement turned on, reboot system for this
        to take effect for rootvg.
0516-1144 unmirrorvg: rootvg successfully unmirrored, user should perform
        bosboot of system to reinitialize boot records.  Then, user must modify
        bootlist to just include:  hdisk0.
```

3. 仮想 I/O クライアントで、rootvg から hdisk1 を除去します。

```
# reducevg rootvg hdisk1
#
```

4. 仮想 I/O クライアントで、hdisk1 デバイスを除去します。

```
# rmdev -l hdisk1 -d
hdisk1 deleted
#
```

5. 仮想 I/O サーバーでバッキング・デバイスを除去します。

```
$ rmbdsp -bd vioc_1_rootvg
vtscsi0 deleted
$
```

6. ディスク・プールからディスクを除去します。以下の例のようにエラー・メッセージが表示された場合でも、ストレージ・プールごとに 1 つの hdisk しかないことが間違いなければ、**deactivatevg** コマンドと **exportvg** コマンドを使用できます。

> **注 : exportvg** コマンドを使用すると、ボリューム・グループ内のすべての論理ボリュームが削除されます。ボリューム・グループに複数の hdisk が含まれている場合は、この hdisk の論理ボリュームも操作の対象になります。確認のためには、**lspv** コマンドを使用できます。この場合は、**exportvg vioc_rootvg_1** コマンドを使用するほうが無難です。

```
$ lspv
NAME            PVID                                VG              STATUS
hdisk0          00c478de00655246                    rootvg          active
hdisk1          00c478de008a399b                    rootvg          active
hdisk2          00c478de008a3ba1                    vioc_rootvg_1   active
hdisk3          00c478deb4b0d4b0                    None
$
```

```
$ chsp -rm -f -sp vioc_rootvg_1 hdisk2


Some error messages may contain invalid information
for the Virtual I/O Server environment.

0516-062 lqueryvg: Unable to read or write logical volume manager
        record. PV may be permanently corrupted. Run diagnostics
0516-882 reducevg: Unable to reduce volume group.
$ deactivatevg vioc_rootvg_1


Some error messages may contain invalid information
for the Virtual I/O Server environment.

0516-062 lqueryvg: Unable to read or write logical volume manager
        record. PV may be permanently corrupted. Run diagnostics
$ exportvg vioc_rootvg_1
$
```

7. 仮想 I/O サーバーで、hdisk デバイスを除去します。

```
$ rmdev -dev hdisk2
hdisk2 deleted
$
```

8. 物理ディスク・ドライブを交換します。
9. 仮想 I/O サーバーで、**cfgdev** コマンドによって新しい hdisk デバイスを構成し、**lspv** コマンドによってその新しいディスクが構成されたことを確認します。

```
$ cfgdev
$ lspv
NAME            PVID                                VG              STATUS
hdisk2          none                                None
hdisk0          00c478de00655246                    rootvg          active
hdisk1          00c478de008a399b                    rootvg          active
hdisk3          00c478deb4b0d4b0                    None
$
```

10. ストレージ・プールごとに複数のディスクがある場合は、仮想 I/O サーバーで **chsp** コマンドを使用して、ストレージ・プールに hdisk を追加します。1 つのストレージ・プールが 1 つの hdisk に対応している場合は、**mksp** コマンドを使用します。

```
$ mksp vioc_rootvg_1 hdisk2
vioc_rootvg_1
0516-1254 mkvg: Changing the PVID in the ODM.
$
```

11. 仮想 I/O サーバーで、バッキング・デバイスを作成し、そのバッキング・デバイスを仮想デバイスに接続します。

```
$ mkbdsp -sp vioc_rootvg_1 32G -bd vioc_1_rootvg -vadapter vhost0
vtscsi0 Available
$
```

12. 仮想 I/O サーバーで、バッキング・デバイスがストレージ・プール内のディスクにまたがっていないことを確認します。この場合は、1 つの hdisk に 1 つのストレージ・プールが対応しています。
13. 仮想 I/O クライアントで、新しい hdisk1 を再構成します。

```
# cfgmgr -l vscsi1
#
```

親デバイスが不明な場合は、以下のようにします。

```
# cfgmgr
#
```

14. 仮想 I/O クライアントで、rootvg を拡張します。

```
# extendvg rootvg hdisk1
0516-1254 extendvg: Changing the PVID in the ODM.
#
```

15. 仮想 I/O クライアントで、rootvg をミラーリングします。

```
# mirrorvg -c 2 rootvg hdisk1
0516-1124 mirrorvg: Quorum requirement turned off, reboot system for this
        to take effect for rootvg.
0516-1126 mirrorvg: rootvg successfully mirrored, user should perform
        bosboot of system to initialize boot records.  Then, user must modify
        bootlist to include:  hdisk0 hdisk1.
#
```

16. 仮想 I/O クライアントで、ブート・レコードを初期化し、ブート・リストを設定します。

```
# bosboot -a

bosboot: Boot image is 18036 512 byte blocks.
# bootlist -m normal hdisk0 hdisk1
```

4.5 複数のストレージ・セキュリティー・ゾーンの管理

注 : 仮想環境のセキュリティーは、ハードウェア管理コンソールと仮想 I/O サーバーの整合性に依存しています。HMC と仮想 I/O サーバーにアクセスできるユーザーは、管理対象システム内の既存のストレージの割り当てを変更したり、LPAR の新しいストレージの割り当てを確立したりすることが可能になるので、HMC と仮想 I/O サーバーへのアクセスは厳密にモニターする必要があります。

SAN 環境で複数のストレージ・セキュリティー・ゾーンを計画するときには、SAN 環境に関するエンタープライズのセキュリティー・ポリシーと現在の SAN 構成を確認してください。

複数のセキュリティー・ゾーンやディスク・サブシステムが SAN スイッチを共用する場合は、複数の仮想 SCSI デバイスが HBA を共用できます。ハイパーバイザー・ファームウェアは、SAN スイッチと同じように機能するからです。仮想 I/O サーバーによって LUN が 1 つのパーティションに割り当てられている場合は、他のパーティションからその LUN を使用したり参照したりすることはできません。ハイパーバイザーは、クライアント・パーティション内の操作が、そのクライアント・パーティションに割り当てられていない共用リソースの制御を取得したり、その共用リソースを使用したりすることができないような設計になっています。

SAN 環境で複数のパーティションに 1 つの LUN を割り当てると、仮想 I/O サーバーによってゾーン（区分け）が設定されます。したがって、SAN 環境では、仮想 I/O サーバーによって使用されている HBA にすべての LUN を割り当てるようにしてください。仮想 I/O サーバーは、それらのパーティションによって使用されている仮想 SCSI クライアント・アダプター（vscsi）に関連付けられている仮想 SCSI サーバー・アダプター（vhost）に LUN（hdisk）を割り当てます。図 4-6 を参照してください。

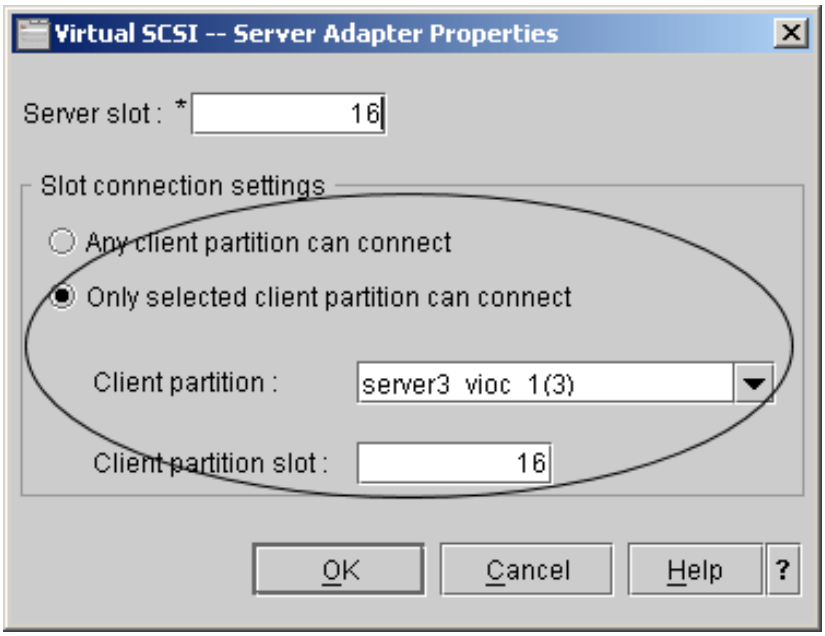


図 4-6 仮想 SCSI の作成

LUN 割り当てリストに基づいてアカウンティング監査やセキュリティー監査を実行しても、LUN の実際の所有者を確認することはできません。すべての LUN が同じ HBA に割り当てられているからです。したがって、監査では注釈が必要になることがあります。

lsmap コマンドを使用すれば、仮想 I/O サーバーから同じようなリストを生成できますが、LUN のマッピングをストレージ・リストで定義するのが業務上の要件になっている場合は、それぞれのアカウント・ゾーンやセキュリティー・ゾーンで別々の HBA ペアを使用する必要があります。その場合でも、同じ仮想 I/O サーバーを使用することは可能です。そのようにしても、セキュリティー・ポリシーには影響しないからです。

セキュリティー・ゾーンまたはディスク・サブシステムが SAN スイッチを共用しない構成が、ハードウェアの問題ではなくセキュリティー上の要件になっている場合は、HBA を共用できません。その場合は、複数の仮想 I/O サーバーを使用して 1 つの管理対象システムで LUN を仮想化することもできません。ハイパーバイザー・ファームウェアは、1 つの SAN スイッチとして機能するからです。

4.6 サーバー間の AIX 5L LPAR の移動

CEC 間で LPAR を移動するためのベスト・プラクティス

このセクションでは、本書の作成中に使用した手順の概要を説明します。この手順は、実稼働環境で使用する前に、テスト環境で確認することをお勧めします。

綿密に計画を立てれば、サーバー CEC 間で LPAR を移動することも可能です。そのためには、ソース・システムで LPAR をシャットダウンし、ターゲット・システムで LPAR をアクティブにします。パーティションを正常に移動するために重要なのは、両方のシステムの仮想デバイス環境がまったく同じであるということです。

この機能を使用すれば、サーバー間で負荷のバランスを取ったり、ダウン時間を最小限に抑えてハードウェアの保守作業を実行したりすることが可能になります。

仮想環境の管理作業を複雑にしないために、パーティションの移動については、以下のような構成をお勧めします。

- 移動する LPAR では仮想デバイスだけを使用します。
- 仮想デバイスについては、ソースとターゲットでスロット番号を統一します。
- すべてのディスク・ストレージを共用ファイバー・チャネル SAN に接続します。
- 両方のサーバーですべての FC LUN を VIOS パーティションから参照できるようにします。
- ソースの VIOS パーティションでマルチパス・ドライバーを使用する場合は、ターゲットの仮想 I/O サーバーでも同じドライバーを同じソフトウェア・レベルで使用します。
- すべての VSCSI ディスクのバッキング・デバイスとして、論理ボリュームではなく物理ボリュームを使用します。

注 : ソース・システムとターゲット・システムの両方で、同じスロット番号に移動するクライアント LPAR の仮想デバイスは、同じスロット番号を割り当てます。仮想 I/O クライアントでそれらのデバイスが同じスロットに存在しない場合は、ターゲット・システムで LPAR をブートしてネットワーク内に構成するために、手動による変更作業が必要になる可能性があります。

これらの推奨事項を守れば、LPAR の数が増えても、システムの管理作業を複雑にしないですみます。

4.6.1 CPU とメモリーの計画

ソース・システムとターゲット・システムで、CPU やメモリーのリソースを統一するのは望ましいことです。これらの可変要素を定数化できれば、移動後にアプリケーションの動作が変わっても、トラブルシューティングが容易になります。ただし、ターゲット・システムをソース・システムとは違った構成にするのが望ましい状況もあります。

例えば、定期保守を計画する場合、いくつかのサーバーを一時的にオフラインにするときに、サービスの質を落としてもかまわなかったり、オフピークの負荷に対応するだけでよかったりするのであれば、ターゲットの LPAR の CPU やメモリーを小さく設定するのは妥当な判断です。

プロセッサー速度の異なるサーバーにパーティションを移動する場合は、まったく同じ量のリソースを構成できない可能性があります。ターゲット・システムの CPU やメモリーの要件を計画するときには、使用可能なリソースや移動の理由について検討することが必要です。

4.6.2 ネットワーク計画

LPAR を移動するときに、DNS やネットワーク・ルーティングの変更が不要になるようにするには、ソースとターゲットの両方の LPAR で、仮想イーサネット・アダプターの数、各アダプターのスロット、アクセスする仮想 IP アドレスと物理 IP アドレスの範囲と VLAN（802.1Q を使用する場合）を統一する必要があります。

注 : ソース LPAR とターゲット LPAR とで仮想イーサネット・アダプターのスロットが異なっている場合は、新しいイーサネット・デバイスがターゲットで作成されます。その LPAR では、ネットワークにアクセスするために手操作による介入が必要になる場合もあります。

可用性を高める必要がある場合は、方法が 2 つあります。つまり、SEA のフェイルオーバーと Network Interface Backup です。

- Network Interface Backup は、最初にサポートされた方式であり、仮想ネットワークを使用する各 AIX LPAR で構成変更が必要になります。
- 新しい SEA フェイルオーバー方式は、Virtual I/O Server Version 1.2 から導入されたものであり、仮想 I/O サーバーで制御できる構成プロセスはシンプルです。さらに、SEA フェイルオーバー方式では、802.1Q の VLAN タグを物理ネットワークから仮想ネットワークに拡張する機能がサポートされているばかりか、Network Interface Backup に対応していないクライアント（Linux LPAR など）のためのオプションも用意されています。優先的な選択肢として SEA フェイルオーバー方式をお勧めするのは、そのためです。

仮想イーサネット・アダプターの MAC アドレスは、ソースの LPAR からターゲットの LPAR に移動するときに変わります。したがって、移動の前に LPAR と通信していた同じサブネットにある他のホストのアドレス解決プロトコル（ARP）キャッシュを更新するには、無差別の ARP が必要になります。AIX 5L オペレーティング・システムは、ターゲット・システムでアダプターのネットワーク・アドレスが構成されると、無差別の ARP を自動的に生成します。これはブロードキャスト・パケットなので、ネットワークまたはホストが極度のビジー状態になっている場合は 別のホストによってドロップされる可能性があります。その場合は、対象のホストに ping を実行して、キャッシュを更新する必要があります。

4.6.3 ストレージ計画

LPAR 移動時のストレージ・リソースの管理は、ネットワーク・リソースの管理よりも複雑になります。LPAR に属するストレージ・リソースをターゲット・システムにきちんと配置するには、綿密な計画が必要です。

仮想アダプターのスロット番号

仮想 SCSI アダプターの追跡管理には、仮想 I/O サーバーとクライアントの両方でスロット番号とパーティション ID を使用します。ソースの LPAR とターゲットの LPAR では、仮想 SCSI アダプター構成を統一して、アダプターのスロット番号を同一にする必要があります。ソースとターゲットのクライアント LPAR 内の VSCSI クライアント・スロット番号は同一でなければなりませんが、ソース・システムとターゲット・システムの仮想 I/O サーバー・パーティションでそれに対応する VSCSI サーバー・スロット番号は一致していなくてもかまいません。サンプル・パーティションの HMC 設定の例については、図 4-7（116 ページ）を参照してください。ソース・プロファイルと宛先プロファイルは、2 つの別々のシステム（CEC）にあります。

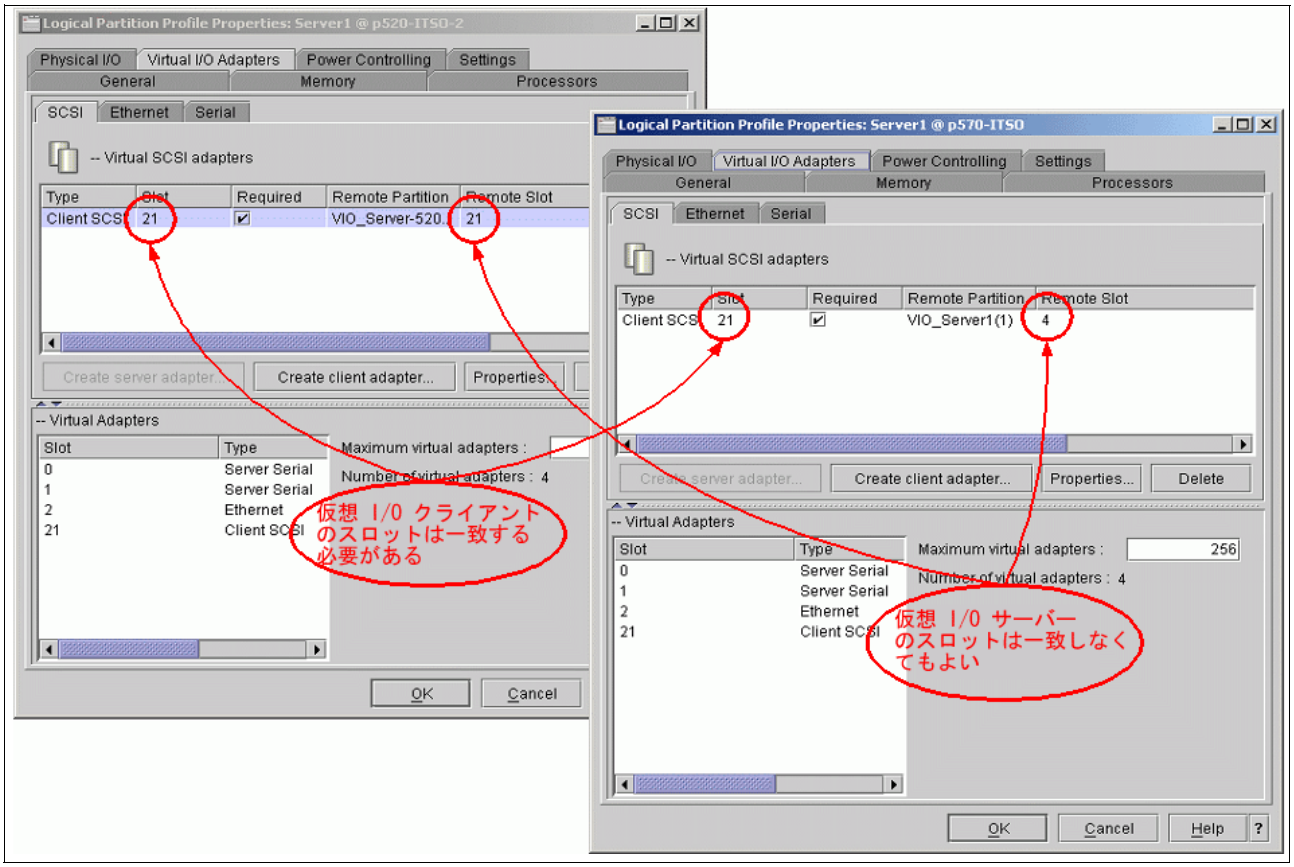


図 4-7　ソース・パーティションとターゲット・パーティションの HMC パーティション・プロファイル・プロパティーの設定

できれば、ソース・システムとターゲット・システムで仮想 I/O サーバーのスロット番号の構成を統一することをお勧めします。ただし、多数のサーバーの間で LPAR を移動する可能性がある大規模な環境でそのような構成にすると、多数のサーバーの間で固有のスロット番号を追跡管理することが必要になります。サーバーの数が増えれば増えるほど、この方式は現実的ではなくなります。また、スロット番号がシステム間で重なり合っている既存の環境の場合も、この方式には無理があります。そのような環境では、代替方式をお勧めします。

- 1 つの代替方式は、ターゲットの仮想 I/O サーバーで次に使用可能なスロット番号を割り振り、図 4-2（102 ページ）のようなスプレッドシートでその割り振りを追跡管理する、という方法です。
- もう 1 つの方式は、各サーバーにスロット番号の範囲を割り当て、クライアントのスロット番号を増分してサーバーのスロット番号を計算する、という方法です（図 4-8（117 ページ））。

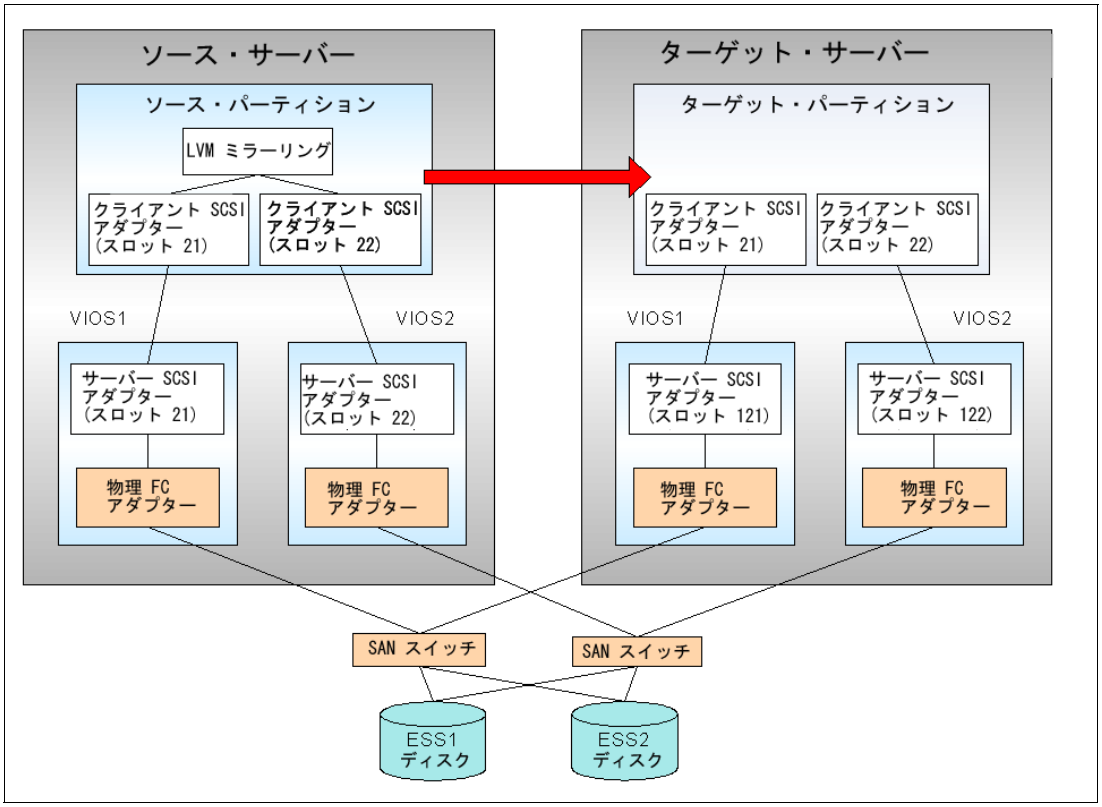


図 4-8　二重 VIOS と LVM ミラーリングを使用した場合の移動可能な LPAR のための仮想ストレージ構成

SAN に関する考慮事項

移動する LPAR に関連したすべてのストレージは、ファイバー・チャネル SAN にある LUN をホストとして使用し、仮想 I/O サーバーの物理ボリュームとしてエクスポートするようにします。移動する LPAR で使用する LUN は、ソースとターゲットの両方の仮想 I/O サーバーから参照できるようにしなければなりません。そのためには、SAN ファブリック・デバイスやストレージ・デバイスの区分け（ゾーン設定）や LUN マスキングなどの構成がかかわってくる可能性がありますが、この点は本書の守備範囲を超えています。

ソース・システムの仮想 I/O サーバーでマルチパス・ソフトウェアを使用する場合は、ターゲット・システムの仮想 I/O サーバーでも同じマルチパス・ソフトウェアを用意する必要があります。例えば、ソース・サーバーで SDDPCM（SDD または Powerpath）を使用している場合は、ターゲット・サーバーでもそれと同じレベルを使用しなければなりません。LPAR の移動時には、別々のマルチパス環境間でストレージを移行しないようにします。そのようにすると、ディスク・デバイスで固有のタグの可視性が影響を受ける可能性があるからです。

もう 1 つの重要な考慮事項は、LPAR で使用されている LUN への同時アクセスを認めるかどうかです。デフォルトでは、仮想 I/O サーバーの hdisk が仮想 SCSI のターゲット・デバイスとして構成されている場合に、仮想 I/O サーバーは SCSI 予約を取得します。つまり、LPAR に LUN をエクスポートできる VIOS は、一度に 1 つだけになります。SCSI 予約によって、LPAR が一度に複数のシステムで開始されることはなくなります。そのような状況が発生すれば、データが破損される可能性があります。

ただし、SCSI 予約があると、移動プロセスは複雑になります。ソース・サーバーで LPAR をシャットダウンしてから、ターゲット・サーバーで LPAR をアクティブにするまでの間に、ソースとターゲットの両方の仮想 I/O サーバーで構成が必要になるからです。

LUN に関連付けられている hdisk デバイスで SCSI 予約をオフにすると、仮想 I/O サーバー間で構成変更なしで LPAR を移動することが可能になります。ただしその場合は、両方のサーバーで LPAR が同時にアクティブ化されてしまった場合に、データが破損する可能性があります。

重要 : rootvg 内の hdisk では、SCSI 予約をアクティブにしておくことをお勧めします。そうすれば、2 つのサーバーでオペレーティング・システムを同時にブートして、データを破損する可能性がなくなります。

仮想 I/O サーバーで SCSI 予約の設定を照会する場合は **lsdev** コマンド、その設定を変更する場合は **chdev** コマンドをそれぞれ使用します。実際の設定は、ストレージの種類によって異なる場合があります。IBM ストレージ・サーバーの LUN では、no_reserve 設定を使用しないでください。

```
$ lsdev -dev hdisk7 -attr reserve_policy
value

no_reserve
$ chdev -dev hdisk7 -attr reserve_policy=single_path
hdisk7 changed
$ lsdev -dev hdisk7 -attr reserve_policy
value

single_path
```

他の種類のストレージの予約設定については、各ストレージのベンダーの資料を参照してください。

LPAR が同時データ・アクセスに通常参加するような環境（GPFS クラスターなど）では、同時アクセスの対象になる hdisk で SCSI 予約を非アクティブにしておきます。それらの hdisk を別個のボリューム・グループに配置し、rootvg ではすべての hdisk の予約をアクティブにすれば、パーティションの同時ブートを回避できます。

バッキング・デバイスと仮想ターゲット・デバイス

ソースと宛先のパーティションは、ソース・システムと宛先システムの仮想 I/O サーバーから同じバッキング・デバイスにアクセスできるようでなければなりません。各バッキング・デバイスには、それぞれ対応する仮想ターゲット・デバイスが必要です。仮想ターゲット・デバイスは、バッキング・ディスクまたは LUN として SCSI ターゲットを参照するのに対し、パーティションの移動先のシステムは、宛先サーバーになります。

重要 : ソース VIOS と宛先 VIOS とで、ファイバー・チャネル LUN の hdisk デバイス番号が異なる場合もあります。hdisk デバイス番号は、新しいデバイスが検出されるたびに増分されるので、接続の順序や他のデバイスの数によって、割り当てられる hdisk 番号が変わってきます。ソース・パーティションと宛先パーティションの hdisk 番号の対応関係を定義するには、デバイスの物理ロケーションで WWPN と LUN 番号を使用します。

ソース VIOS で **lsmap** コマンドを使用すれば、宛先 VIOS で作成する必要がある仮想ターゲット・デバイスとそれに対応するバッキング・デバイスをリストできます。ソース VIOS の vhost アダプター番号を確認できれば、**lsmap** コマンドの **-vadapter** フラグにアダプターを指定できます。そうでない場合は、**lsmap** コマンドの **-all** フラグを指定して、ソース・パーティションに接続している仮想アダプターを確認できます。以下に示すのは、DS4000 シリーズのデバイスに関するリストです。

```
$ lsmap -all
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
```

```
vhost0          U9117.570.107CD9E-V1-C4                      0x00000007

VTD                   lpar07_rootvg
LUN                   0x8100000000000000
Backing device        hdisk5
Physloc               U7879.001.DQD186N-P1-C3-T1-W200400A0B8110D0F-L0
```

ソース VIOS の各バッキング・デバイスの Physloc ID を使用すれば、**lsdev -vpd** コマンドの出力から宛先 VIOS で対象になる hdisk デバイスを識別できます。マルチパス I/O とマルチコントローラーのストレージ・サーバーの場合は、ソースと宛先の VIOS で使用しているパスまたはコントローラーによって、Physloc ストリングのいくつかの文字が変わることがあります。

```
$ lsdev -vpd -dev hdisk4
  hdisk4           U787A.001.DNZ00XY-P1-C5-T1-W200500A0B8110D0F-L0  3542     (20
0) Disk Array Device

  PLATFORM SPECIFIC

  Name:  disk
    Node:  disk
    Device Type:  block
```

ソース VIOS の各バッキング・デバイスに対応する宛先 VIOS の hdisk デバイスをメモしておきます。移動プロセスで LPAR が非アクティブになっている間に、宛先 VIOS で仮想ターゲット・デバイスを作成するには、この対応関係が必要になります（この点については、以下のセクションで取り上げます）。この対応関係に不備や間違いがあると、移動プロセスで LPAR が非アクティブになっている時間が長引いてしまう可能性があります。

4.6.4 LPAR の移動

LPAR を移動しようとする前に、構成が正しく、移動のための準備が整った環境になっていることを確認するのは重要です。以下の手順を実行します。

1. HMC インターフェースを使用して、ソースと宛先の構成に整合性があることを確認します。
   - 宛先の CPU とメモリーは、ソースの設定とまったく同じではないとしても、ワークロードの見積もりに適合している必要があります。
   - LPAR 内の宛先の仮想アダプターとスロット番号は、ソースと一致している必要があります。ただし、VIOS パーティション内のスロット番号は一致していなくてもかまいません。
2. LPAR で引き続き同じ IP アドレスを使用するために、宛先 VIOS でも同じネットワークを使用できることを確認します。
3. 宛先 VIOS ですべての必要な hdisk デバイスを参照できること、hdisk と各仮想ターゲット・デバイスの対応関係が定義されていることを確認します。
4. 移動の前にソース LPAR のバックアップを作成します。

初期の計画とセットアップが完了したら、LPAR の移動プロセスは比較的簡単です。LPAR を移動するには、以下の手順を実行します。

1. オペレーティング・システムの **shutdown** コマンドまたは HMC インターフェースを使用して、ソース・システムで LPAR をシャットダウンします。
2. ソース LPAR が「Not Activated」になっていることを確認してから、**rmdev** コマンドを使用して、ソース VIOS から仮想 SCSI ターゲット・デバイスを除去します。

   ```
   $ rmdev -dev lpar07_rootvg
   lpar07_rootvg deleted
   ```

3. 先ほど作成したリストに基づき、**mkvdev** コマンドを使用して、宛先 VIOS で仮想 SCSI ターゲット・デバイスを構成します。**lsmap** コマンドを使用して、すべてのターゲット・デバイスが正常に作成されたことを確認します。

```
$ mkvdev -vdev hdisk4 -vadapter vhost0 -dev lpar07_rootvg
lpar07_rootvg Available
$ lsmap -vadapter vhost0
SVSA            Physloc                                     Client Partition ID
--------------- ------------------------------------------- ------------------
vhost0          U9111.520.10DDEDC-V1-C21                    0x00000000

VTD                   lpar07_rootvg
LUN                   0x8100000000000000
Backing device        hdisk4
Physloc               U787A.001.DNZ00XY-P1-C5-T1-W200500A0B8110D0F-L0
```

4. HMC を使用して、宛先システムで LPAR をアクティブにします。端末ウィンドウが開いていない場合は、その LPAR で端末ウィンドウを開きます（図 4-9）。

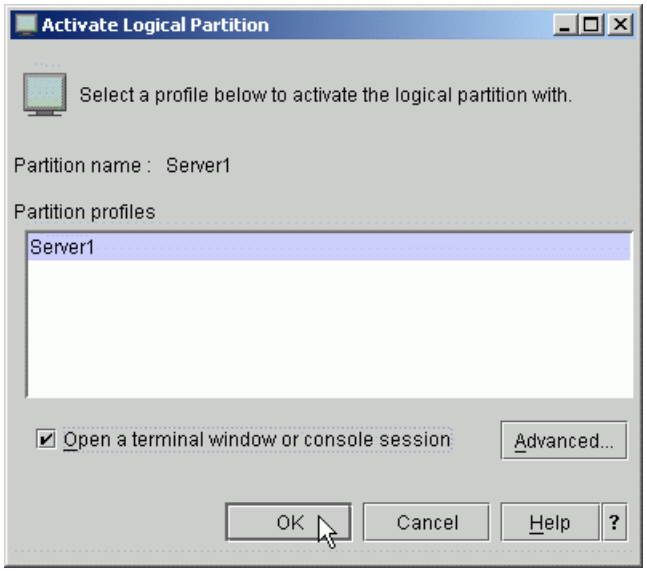


図 4-9　パーティションをアクティブにして、端末ウィンドウを開く

5. 端末ウィンドウを使用して、LPAR が宛先のオペレーティング・システムでブートすることを確認します。最初のブート時に、SMS メニューを使用して、アクティブなコンソールとブート・リストを更新しなければならない場合もあります。

4.7 MPIO の計画と配置

仮想化環境では、MPIO を配置することによって、効果的かつ効率的に柔軟性を確保し、システムの可用性を最大限に高めることができます。このセクションでは、MPIO の配置を基盤とする二重仮想 I/O サーバー・ソリューションを計画する場合の一般的な考慮事項と、構成しなければならない推奨パラメーターを取り上げます。二重仮想 I/O サーバーと MPIO を構成するための段階的なプロセスの詳細については、*「Advanced POWER Virtualization on IBM System p5」*（SG88-4018）を参照してください。

このセクションの基礎になっているのは、図 4-10 のような推奨二重仮想 I/O サーバー構成の配置です。ここでは、仮想 I/O サーバーでベンダー固有のデバイス・ドライバーを使用し、仮想 I/O クライアント・パーティションでデフォルトの AIX 5L MPIO を使用しています。

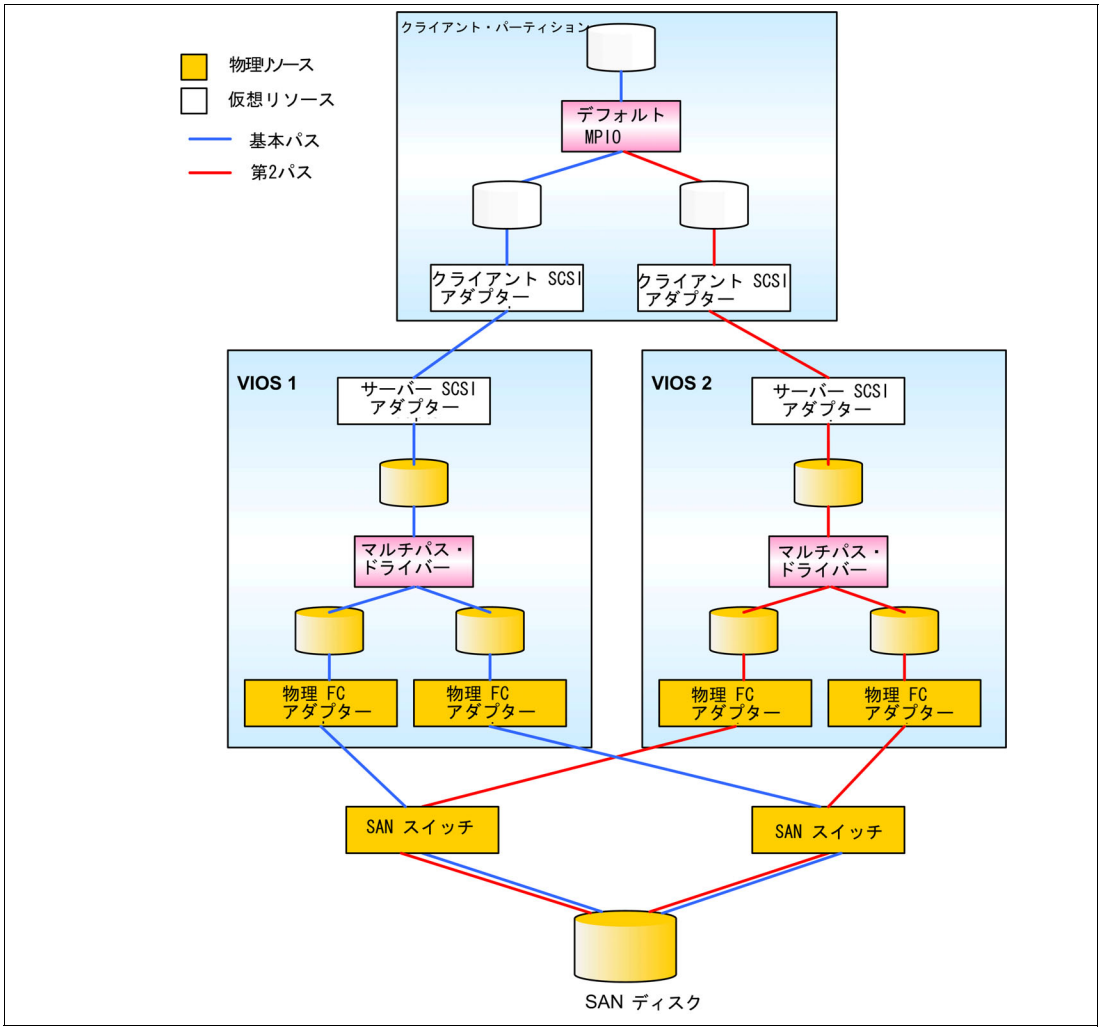


図 4-10　二重仮想 I/O サーバーを使用した推奨 MPIO の配置

4.7.1 二重 VIOS 環境（MPIO）の計画

二重仮想 I/O サーバーの MPIO に関するベスト・プラクティス

二重仮想 I/O サーバー構成で MPIO を基盤とした環境を計画する場合の考慮事項は、以下のとおりです。

- 仮想アダプターは、仮想 I/O サーバーの初期のプロファイルの中で定義することもできれば、クライアント・パーティションの作成時に動的に追加することもできます。仮想

I/O サーバーで仮想 I/O リソースを事前定義する場合は、リモート・パーティションとスロットの接続が可能になるように仮想 SCSI サーバー・アダプターを設定します。クライアント・パーティションの作成後には、特定のクライアント・パーティションと仮想スロット番号に合わせて、仮想 I/O サーバーのプロファイルで仮想 SCSI サーバー・アダプターの接続情報を更新できます。

- 仮想 SCSI サーバー・アダプターを動的に追加する場合は、次回仮想 I/O サーバーをシャットダウンして再びアクティブ化するときに、その新しいアダプターに合わせて仮想 I/O サーバーのプロファイルを更新する必要があります。
- 動的 LPAR を使用する場合は、仮想アダプターの追加と除去だけが可能になります。仮想 I/O サーバーのパーティションを定義するときには、仮想アダプターを「Desired」として選択することをお勧めします。そのようにすれば、動的 LPAR が使用可能な場合に、仮想 I/O サーバーが稼働中でも、アダプターの除去と追加が可能になります。
- 仮想 I/O サーバーのブートに必要な物理アダプターは、「Required」にします。他のすべての物理アダプターは、「Desired」でもかまいません。その設定であれば、動的に除去したり、他のパーティションに移動したりすることが可能になります。
- 仮想アダプターを作成するときには、次に使用可能なスロット番号をデフォルトで使用する代わりに、仮想スロット番号を設定することも可能です。仮想スロット番号の番号付けに関する推奨方式については、4.3.2、『仮想デバイスのスロット番号』（103 ページ）を参照してください。

4.7.2 仮想 I/O サーバーの構成

このセクションでは、二重仮想 I/O サーバー環境で MPIO を配置するときに仮想 I/O サーバーで必要になる構成パラメーターを取り上げます。

ファイバー・チャネル・アダプター・デバイスの構成

すべての HBA でファイバー・チャネル・アダプターの init_link 属性をファイバー・チャネル・アービトレーション・ループ（FCAL）のデフォルト設定から pt2pt に変更すれば、ファイバー・チャネル・スイッチ・ポートに接続できます。その属性を pt2pt に変更するには、**chdev** コマンドを使用します。この手順はオプションです。この設定を pt2pt に変更するのは、スイッチに接続する場合だからです。変更を確認するには、**lsdev** コマンドを使用します。

例 4-3 lsmap -all コマンドの出力

```
$ chdev -dev fcs0 -attr init_link=pt2pt -perm
fcs0 changed
$ lsdev -dev fcs0 -attr
attribute     value      description                                          user_settable

bus_intr_lvl  163        Bus interrupt level                                  False
bus_io_addr   0xffc00    Bus I/O address                                      False
bus_mem_addr  0xf8020000 Bus memory address                                   False
init_link     pt2pt      INIT Link flags                                      True
intr_priority 3          Interrupt priority                                   False
lg_term_dma   0x800000   Long term DMA                                        True
max_xfer_size 0x100000   Maximum Transfer Size                                True
num_cmd_elems 200        Maximum number of COMMANDS to queue to the adapter True
pref_alpa     0x1        Preferred AL_PA                                      True
sw_fc_class   2          FC Class for Fabric                                  True
```

ファイバー・チャネル SCSI デバイスの構成

fscsi デバイスには、両方の仮想 I/O サーバーで変更しなければならない特定の属性が含まれています。つまり、fc_err_recov 属性と dyntrk 属性です。どちらの属性も、以下のような **chdev** コマンドによって変更できます。

```
$ chdev -dev fscsi0 -attr fc_err_recov=fast_fail dyntrk=yes -perm
fscsi0 changed
```

fc_err_recov 属性を fast_fail に変更すると、アダプターでリンク・イベント（ストレージ・デバイスとスイッチの間のリンクの破損など）が検出された場合に、I/O がすぐに失敗するようになります。fast_fail 設定は、二重仮想 I/O サーバー構成の場合にのみお勧めします。一方、dyntrk 属性を yes に設定すると、仮想 I/O サーバーが SAN のケーブル接続の変更を許容するようになります。これらの属性の変更を有効にするには、両方の仮想 I/O サーバーをリブートする必要があります。あるいは、fscsi デバイスを構成解除してから再構成することによって、これらの設定を有効にすることもできます。

SDDPCM とサード・パーティーのマルチパス・ソフトウェア

IBM SAN ストレージ（SVC、DS8000、DS6000、ESS など）に接続している仮想 I/O サーバーについては、Subsystem Device Driver Path Control Module（SDDPCM）マルチパス・ソフトウェアを仮想 I/O サーバーにインストールすることによって、仮想 I/O サーバーと IBM SAN ストレージの間に確立する複数のファイバー・チャネル接続のロード・バランシングと冗長性を実現できます。SDDPCM ソフトウェアは、以下の IBM ストレージ Web サイトからダウンロードできます。

http://www.ibm.com/support/docview.wss?uid=ssg1S4000201

仮想 I/O サーバーは、IBM SAN ストレージだけでなく、さまざまなサード・パーティーの SAN ストレージ・ソリューションにも対応しており、サード・パーティーのベンダー固有のマルチパス・ソフトウェアをインストールできるようになっています。仮想 I/O サーバーに対応した IBM とサード・パーティーのすべてのストレージ・ソリューションのサポート・マトリックスについては、以下の Web サイトを参照してください。

http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/VIOS_datasheet_063006.html

重要 : サード・パーティーのストレージ・ベンダーは、AIX 5L と仮想 I/O サーバーに自社のストレージ・デバイスをインストールするための推奨事項を独自に用意しています。ベンダー固有のインストール・ガイドを参照して、構成方法を確認してください。

AIX 5L の hdisk デバイスの構成

各ディスクで no_reserve 設定が必要です。

二重仮想 I/O サーバーによってクライアント・パーティションに物理ドライブを正しく表示できるようにするには、各ディスクの reserve_policy 属性を no_reserve に設定する必要があります。例えば、hdisk1 であれば、**chdev** コマンドを使用して、hdisk の予約ポリシーとアルゴリズムの両方を変更できます。:

```
$ chdev -dev hdisk1 -attr reserve_policy=no_reserve
hdisk1 changed
```

さらに、基本 AIX 5L MPIO サポートを使用する場合に、仮想 I/O サーバーの複数の HBA でロード・バランシングを実現するには、各物理ドライブのアルゴリズムを round_robin に設定します。

```
$ chdev -dev hdisk1 -attr algorithm=round_robin
hdisk1 changed
```

hdisk デバイスの変更を確認するには、**lsdev** コマンドを使用します。

```
$ lsdev -dev hdisk1 -attr
   attribute       value                                description
   user_settable

   PCM             PCM/friend/scsiscsd                  Path Control Module           False
   algorithm       round_robin                          Algorithm                     True
   dist_err_pcnt   0                                    Distributed Error Percentage  True
   dist_tw_width   50                                   Distributed Error Sample Time True
   hcheck_interval 0                                    Health Check Interval         True
   hcheck_mode     nonactive                            Health Check Mode             True
   max_transfer    0x40000                              Maximum TRANSFER Size         True
   pvid            0021768a0151feb40000000000000000     Physical volume identifier    False
   queue_depth     3                                    Queue DEPTH                   False
   reserve_policy  no_reserve                           Reserve Policy                True
   size_in_mb      18200                                Size in Megabytes             False
```

仮想 SCSI サーバーのサポート

AIX 5L クライアント・パーティションにある仮想 SCSI デバイスのマルチパス・サポートのためには、仮想 I/O サーバーからクライアント・パーティションに対して、SAN LUN を物理ドライブ（hdiskx）として提供する必要があります。二重仮想 I/O サーバーを使用する場合は、いったん大規模な SAN LUN を用意してから、仮想 I/O サーバーのレベルでその LUN を小さな論理ボリュームに分割することはできません。この構成のストレージ管理は SAN で実行されるので、仮想 I/O サーバーの SAN LUN とクライアント・パーティションの仮想 SCSI ドライブの間には 1 対 1 の対応関係があります。詳しくは、4.1、『VIOS の物理ストレージの管理とエクスポート』（96 ページ）を参照してください。

重要 : 仮想 I/O サーバーごとにドライブの数が違っていたり、ドライブが別々の時点でゾーン設定されていたりする場合は、仮想 I/O サーバーによってデバイス名（hdiskx）が異なる可能性があります。二重仮想 I/O サーバーを使用する場合は、同じクライアント・パーティションにドライブを提供するときに、LUN ID が一致していることを常に確認してください。両方の仮想 I/O サーバーで同じデバイス名を使用するのは、管理作業の視点からすると非常に便利です。

仮想 I/O サーバーの SAN LUN でボリューム・グループを作成するべきではありません。物理ディスクと仮想 SCSI サーバー・アダプター（例えば vhost0）の間の対応関係を定義するには、**mkvdev** コマンドを使用して、ターゲット・デバイスとして物理ディスク（例えば hdisk1）を指定します。

```
$ mkvdev -vdev hdisk1 -vadapter vhost0 -dev vtscsi0
vtscsi0 Available
```

4.7.3 仮想 I/O クライアントの構成

両方の仮想 I/O サーバーをインストールして、仮想 SCSI リソースのマッピングを設定したら、クライアント・パーティションにオペレーティング・システムをインストールします。MPIO を実装するには、さらに追加の構成手順が必要です。

MPIO のパス管理とロード・バランシング

クライアント・パーティションと二重仮想 I/O サーバーの間の仮想 SCSI に関する MPIO サポートは、フェイルオーバー・モードにのみ対応しています。どの仮想 SCSI ディスクについても、クライアント・パーティションは、通常は 1 つの仮想 I/O サーバーへのプライマリー・パスを使用し、フェイルオーバー時にはもう 1 つの仮想 I/O サーバーを使用するためのセカンダリー・パスに切り替えます。両方のパスが有効になっていても、一度に 1 つのパスだけが使用されます。

二重仮想 I/O サーバーで複数のクライアント・パーティションのロード・バランシングを実現するには、クライアント・パーティションの各仮想 SCSI ディスクで、プライマリー・パスを（つまり、特定の仮想 I/O サーバーを）選択して優先順位を設定します。優先順位は、仮想 SCSI ディスクごとに設定するので、各ディスクのプライマリー・パスを柔軟に決定できます。このようなきめ細かな設定によって、システム管理者は、クライアント・パーティションのすべてのディスクがプライマリー・パスとして 1 つの仮想 I/O サーバーを使用するのか、各ディスクが交互に仮想 I/O サーバーを使用するのかを指定できます。推奨方式は、2 つの仮想 I/O サーバーの間でクライアント・パーティションを分割することです。

重要 : 優先パスは VSCSI LUN ごとに設定できるので、二重 VIOS 構成のロード・バランシングをきめ細かく管理できます。

個々のパスの優先順位を調整できます。

例えば、クライアント・パーティションで仮想 I/O クライアント SCSI アダプターとして vscsi0 と vscsi1 を使用する場合は、クライアント・パーティションで **chpath** コマンドを使用して、vscsi0 パスの優先順位を下げることによって vscsi1 をプライマリー・パスとして設定できます。

```
# chpath -l hdisk1 -a priority=2 -p vscsi0
path Changed
```

それぞれのパスの優先順位を確認するには、以下のような **lspath** コマンドを使用します。

```
# lspath -E -l hdisk1 -p vscsi0
priority 2 Priority True

# lspath -E -l hdisk1 -p vscsi1
priority 1 Priority True
```

二重仮想 I/O サーバー構成で MPIO を使用する場合に、パス障害の自動検出機能を有効にするには、クライアント・パーティションの各仮想 SCSI ディスクの正常性チェック間隔属性を変更する必要があります。hcheck_interval 属性のデフォルト値は、0（無効）です。例えば、各仮想 SCSI ディスクの health_interval 属性を 20 秒に変更するには、クライアント・パーティションで以下のような **chdev** コマンドを使用します。この属性の変更を有効にするには、クライアント・パーティションをリブートしなければなりません。

```
# chdev -l hdisk1 -a hcheck_interval=20 -P
hdisk1 changed
```

ディスクの属性をリストするには、**lsattr** コマンドを使用します。

```
# lsattr -El hdisk1
PCM             PCM/friend/vscsi                 Path Control Module        False
algorithm       fail_over                        Algorithm                  True
hcheck_cmd      test_unit_rdy                    Health Check Command       True
hcheck_interval 20                               Health Check Interval      True
hcheck_mode     nonactive                        Health Check Mode          True
max_transfer    0x40000                          Maximum TRANSFER Size      True
pvid            00c5e9de252ef77f0000000000000000 Physical volume identifier False
queue_depth     100                              Queue DEPTH                True
reserve_policy  no_reserve                       Reserve Policy             True
```

4.8 SCSI 用キューのエントリー数

前のセクションにもあった queue_depth の値をデフォルト値の 3 から増やすと、いくつかの構成ではディスクのスループットが改善される可能性があります。ただし、この点については、他の要因も考慮に入れる必要があります。例えば、1 つの要因として、クライアント・パーティションのディスク・インスタンスが仮想ターゲット・デバイスとして使用する仮想 I/O サーバー上のすべての物理ストレージ・デバイスの queue_depth 属性の値があります。さらに、ディスク・インスタンスの親デバイスになっている仮想 SCSI クライアント・アダプターの最大転送サイズも 1 つの要因になります。

仮想 SCSI クライアント・アダプターの最大転送サイズは、仮想 I/O サーバーが設定します。つまり、サーバー上の使用可能なリソースと、サーバー上の物理ストレージ・デバイスの最大転送サイズに基づいて、その値を決定します。さらに、ボリューム・グループのミラーリング構成または MPIO 構成にかかわっている他のデバイスのキュー・エントリー数と最大転送サイズも、その他の要因として考慮に入れる必要があります。いくつかのデバイスでキュー・エントリー数を増やすと、親アダプターが同じ他のデバイスで使用できるリソースが減り、スループットが落ちる可能性もあります。

最も単純明快な構成は、仮想ターゲット・デバイスとして 1 つの物理 LUN を使用する構成です。仮想 SCSI クライアント・デバイスのキュー・エントリー数を効率的に使用するには、キュー・エントリー数の値をその物理 LUN のキュー・エントリー数よりも大きくするべきではありません。大きな値を設定しても、リソースを無駄にするだけであり、パフォーマンスが改善されることはありません。仮想ターゲット・デバイスが論理ボリュームの場合は、その論理ボリュームに含まれているすべてのディスクのキュー・エントリー数を考慮に入れる必要があります。論理ボリュームをミラーリングする場合は、ミラーで使用する物理デバイスの最小のキュー・エントリー数よりも vSCSI クライアントのキュー・エントリー数を大きくするべきではありません。ミラーリングすると、LVM はミラーのすべてのデバイスにデータを書き込み、すべての書き込みが完了するまで書き込み完了を報告しません。したがって、スループットは、キュー・エントリー数が最小のデバイスに合わせて効率的に縮小されることになります。この動作は、仮想 I/O サーバーとクライアントのミラーリングに当てはまります。

ボリューム・グループに含まれているすべての仮想ディスクでキュー・エントリー数を統一します。

仮想ディスクのキュー・エントリー数は、物理ディスクのキュー・エントリー数と同じ値にすることをお勧めします。複数の仮想ディスクにまたがるボリューム・グループがクライアントにある場合は、そのボリューム・グループに含まれているすべての仮想ディスクでキュー・エントリー数を統一します。そのボリューム・グループで論理ボリュームをミラーリングする場合は、最後のディスクにデータが書き込まれるまで書き込みが完了しないので、この構成が特に重要になります。

クライアントの MPIO 構成で、基本パスのキュー・エントリー数を第 2 パスのキュー・エントリー数よりもかなり大きくすると、フェイルオーバー時にパフォーマンスが突然低下する可能性があります。

仮想 SCSI クライアント・ドライバーは、各仮想 I/O クライアント・アダプター・インスタンスに 512 個のコマンド・エレメントを割り振ります。そのうち 2 つのコマンド・エレメントは、エラー・リカバリー時にアダプターが使用するものとして予約されています。また、エラー・リカバリー時に使用できる各オープン・デバイスのために 3 つのコマンド・エレメントが予約されています。残りは、I/O 要求用として共通プールに入っています。新しいデバイスがオープンすると、共通プールにあるコマンド・エレメントは除去されます。それぞれの I/O 要求では、仮想 I/O サーバーでその I/O 要求がアクティブになるときのために 1 つのコマンド・エレメントが必要です。

1 つの仮想デバイスのキュー・エントリー数を増やすと、そのアダプターで一度にオープンできるデバイスの数が減ります。また、他のデバイスが仮想 I/O サーバーでアクティブにできる I/O 要求の数も減ります。

一例として、図 4-11 のような場合を取り上げてみましょう。このシナリオでは、1 つの物理ディスクを 1 つの仮想ディスクに対応付けます。

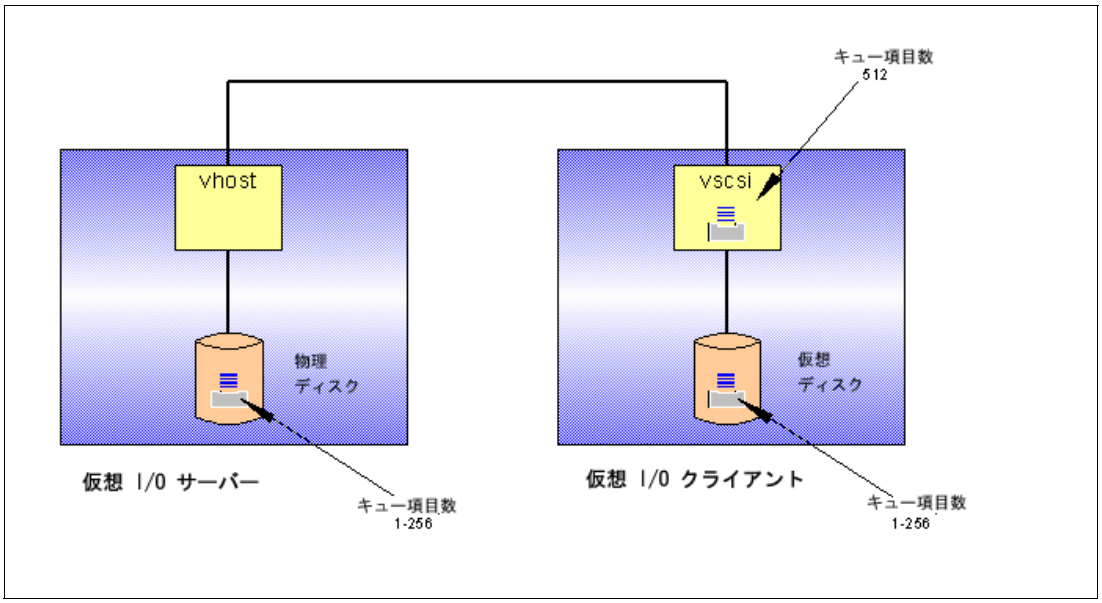


図 4-11　1 つの物理ディスクと 1 つの仮想ディスクの対応関係

仮想 I/O サーバーで、仮想 I/O クライアントに対して仮想化した物理ディスクを指定して、**lsdev -dev hdisk*N* -attr** コマンドを実行します。キュー・エントリー数を記録しておいてください（例 4-4 では太字になっています）。

例 4-4　仮想 I/O サーバーでの lsdev コマンドの使用

```
$ lsdev -dev hdisk2 -attr
attribute       value                                description
user_settable

PCM             PCM/friend/scsiscsd                  Path Control Module           False
algorithm       fail_over                            Algorithm                     True
dist_err_pcnt   0                                    Distributed Error Percentage  True
dist_tw_width   50                                   Distributed Error Sample Time True
hcheck_interval 0                                    Health Check Interval         True
hcheck_mode     nonactive                            Health Check Mode             True
max_transfer    0x40000                              Maximum TRANSFER Size         True
pvid            00cddeec68220f190000000000000000     Physical volume identifier    False
queue_depth      3                                   Queue DEPTH                   False
reserve_policy  single_path                          Reserve Policy                True
size_in_mb      36400                                Size in Megabytes             False
$
```

仮想 I/O クライアントで、**chdev -l hdiskN -a queue_depth=*x*** コマンドを実行します（***x*** は、先ほど仮想 I/O サーバーで記録した数です）。この方式を使用すれば、物理ディスクと仮想ディスクのキュー・エントリー数のバランスが取れます。すでに見たとおり、キュー・エントリー数のサイズによって、仮想 I/O クライアント SCSI アダプターで使用できるデバイスの数は制限されます。割り当て済みのリソースに対して、使用可能なストレージがパーティション化されるからです。

キュー・エントリー数のデフォルト値 3 を調整しなければならない場合もあります。

図 4-12（128 ページ）は、別の例を示したものです。つまり、仮想 I/O サーバーの 1 つの論理ボリュームを仮想 I/O クライアントの 1 つの仮想ディスクに対応付けています。ここで注意しなければならないのは、仮想 I/O サーバーのすべての論理ボリュームが同じディスク・

キューを共用するということです。したがって、パフォーマンスをきめ細かく調整することによって、ディスク・キューがあふれる事態を回避しなければなりません。この例では、キュー・エントリー数が 3 というデフォルト値になっている物理ディスクが 1 つあります。そのディスクでは 3 つの論理ボリュームが仮想化されており、仮想 I/O クライアントのすべての仮想ディスクのキュー・エントリー数がデフォルト値の 3 になっています。この場合は、3 というキュー・エントリー数で 9 個の I/O が保留になる可能性があります。この状況での 1 つの解決策は、仮想ディスクのキュー・エントリー数の合計に合わせて物理ディスクのキュー・エントリー数を増やすことです。

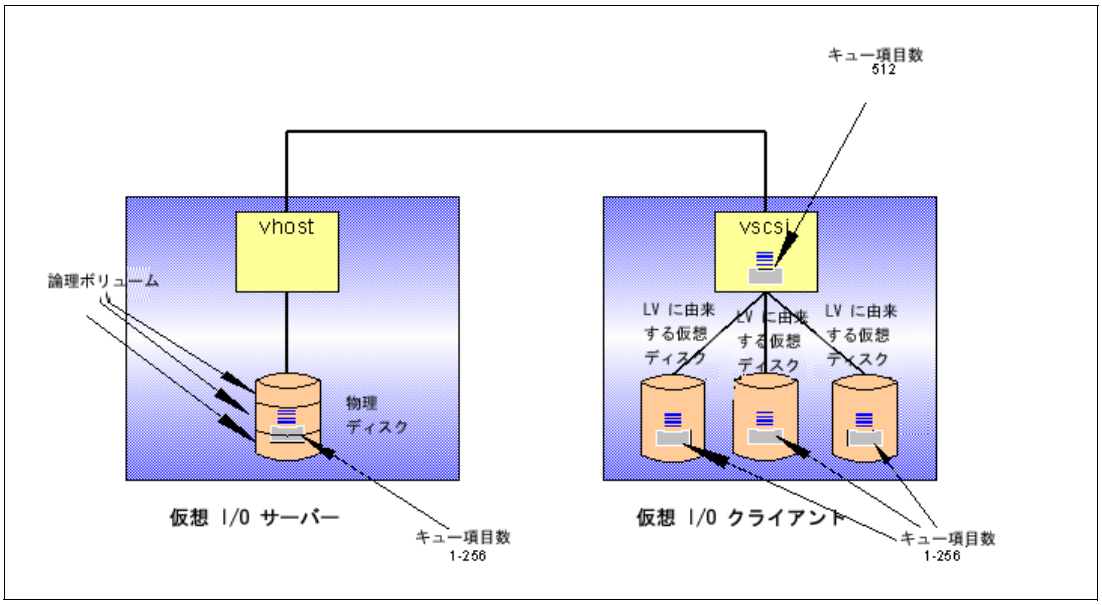


図 4-12 1 つの LV と 1 つの仮想ディスクの対応関係

コマンドのキューイングに関するベスト・プラクティス

最適化のために VSCSI クライアントのキューイングを増やすのが効果的なのは、以下のような場合です。

- ストレージがファイバー・チャネル接続の場合。

  SCSI のキュー・エントリー数がデフォルト設定の 3 になっていて、すでに制限要因になっています。

- VSCSI デバイスが LUN で接続されている場合。
  - 仮想 I/O サーバーでは、複数のディスク /LUN の間で LV のストライピングができません。
  - LV が複数ディスクの LUN に存在し、その LUN で複数クライアントの要求に関する大きな競合がなければ、LV を効率的に使用できます。
- LUN に複数の物理ディスクが含まれている場合。

  LUN に含まれているディスクが多ければ多いほど、同時に処理できる I/O の数も多くなる可能性があります。
- 未解決の I/O 要求を多数出してしまうほど多くのプロセスまたは非同期 I/O がワークロードにある場合。

5

パフォーマンスと計画

パーティショニングを使用した仮想 I/O 環境では、CPU、メモリー、ネットワーク、ストレージなどのシステム・リソースにアクセスするためのさまざまな方法があります。仮想 I/O クライアントのパフォーマンスは、構成方法や種々のパラメーターによって改善されることもあれば、悪影響を受けることもあります。この章では、仮想環境のパフォーマンスの計画と構成に関するベスト・プラクティスを重点的に取り上げます。

5.1 仮想プロセッサーの構成

マイクロ・パーティション環境では、システム管理者が HMC を使用して特定のパーティションの仮想プロセッサーの構成を管理できます。実際に、1 つのパーティションに割り当てる仮想プロセッサーの妥当な数を決定するには、ある程度の計画とテストが必要です。上限付き（capped モード）のパーティションでも上限なし（uncapped モード）のパーティションでも、仮想プロセッサーの数によってパフォーマンスが影響を受ける可能性があるからです。

プロセッサー・フォールディング機能によって、アイドル状態の仮想プロセッサーを効率的に管理できます。

ハイパーバイザーでは、オンラインの仮想プロセッサーの管理に関連した処理が実行されるので、それらの属性の値を選択する前に、仮想プロセッサーのキャパシティー要件を確認する必要があります。ただし、AIX 5L Version 5.3 ML3 には、アイドル状態の仮想プロセッサーを管理するためのプロセッサー・フォールディングという新しい機能が組み込まれています。カーネル・スケジューラーが拡張され、パーティションの物理的な使用状況に基づいてパーティションの瞬間的な負荷を測定することによって、仮想プロセッサーの使用を動的に増減できるようになりました。

カーネル・スケジューラーは、パーティションの物理的な使用状況に対応するために活性化しなければならない仮想プロセッサーの数を毎秒評価します。その数で仮想プロセッサーの使用率が高くなると、必要な仮想プロセッサーの基底数を増分して、ワークロードの拡張に対応できるようにします。追加の仮想プロセッサーを要求するには、**schedo** コマンドを使用します。その値に基づいて、1 つの仮想プロセッサーを有効にする必要があるのか、無効にする必要があるのかを決定します。スケジューラーは、使用する仮想プロセッサーの数を毎秒 1 つずつ調整するからです。つまり、その計算数が、現在アクティブになっている仮想プロセッサーの数より大きければ、1 つの仮想プロセッサーが活性化されます。その逆に、その数が、現在アクティブになっている仮想プロセッサーの数より小さければ、1 つの仮想プロセッサーが非活性化されます。

動的 LPAR の場合と同じく、非活性化した仮想プロセッサーは、パーティションから動的に除去されるわけではありません。その仮想プロセッサーは、アンバインドされた処理を受け取って実行することができなくなるだけで、バインドされたジョブを実行することは依然として可能です。ユーザーまたはアプリケーションから見て可視状態のオンライン論理プロセッサーとオンライン仮想プロセッサーの数は、変わりません。アクティブな仮想プロセッサーとアクティブでない仮想プロセッサーは、システム内部の問題なので、システムで稼働するミドルウェアやアプリケーションには何の影響もありません。

仮想プロセッサーのフォールディングに関する仮想プロセッサー管理機能の有効 / 無効を切り替えるには、vpm_xvcpus 調整可能オプションを使用します。vpm_xvcpus 調整可能オプションのデフォルト値は 0（フォールディング有効）です。つまり、仮想プロセッサーは管理された状態になります。vpm_xvcpus 調整可能オプションを変更するには、**schedo** コマンドを使用します。

以下に示すのは、仮想プロセッサー管理機能を無効にする例です。

```
# schedo -o vpm_xvcpus=-1
Setting vpm_xvcpus to -1
```

仮想プロセッサー管理機能が有効かどうかを確認するには、以下のコマンドを使用します。

```
# schedo -a | grep vpm_xvcpus
            vpm_xvcpus = -1
```

使用する仮想プロセッサーの数を 1 つ増やすには、以下のコマンドを使用します。

```
# schedo -o vpm_xvcpus=1
Setting vpm_xvcpus to 1
```

それぞれの仮想プロセッサーは、最大で 1 つの物理プロセッサーを消費できます。必要な仮想プロセッサーの数を判別するには、物理 CPU の使用率の合計と vpm_xvcpus 調整可能オプションの値を使用して以下の式を計算します。

必要な仮想プロセッサーの数 =（物理 CPU の利用率）+（有効にする追加の仮想プロセッサーの数）

必要な仮想プロセッサーの数が現在の有効な仮想プロセッサーの数より少なければ、1 つの仮想プロセッサーが無効になります。必要な仮想プロセッサーの数が現在の有効な仮想プロセッサーの数より多ければ、1 つの無効な仮想プロセッサーが有効になります。ただし、無効な仮想プロセッサーに接続しているスレッドは、依然として実行可能です。

先ほどの式で計算した値は、必ず整数に切り上げるようにします。使用する仮想プロセッサーの数の計算例を以下に示します。

最後のインターバルで、パーティション A は 2.5 個分のプロセッサーを使用していました。vpm_xvcpus 調整可能オプションは、1 に設定されています。先ほどの式を使用して計算すると、以下のようになります。

- 物理 CPU の使用率 = 2.5
- 有効にする追加の仮想プロセッサーの数（vpm_xvcpus）= 1
- 必要な仮想プロセッサーの数 = 2.5 + 1 = 3.5

この計算で得た値を次の整数に切り上げると、4 になります。つまり、このシステムで必要な仮想プロセッサーの数は、4 個です。パーティション A に 8 個の仮想プロセッサーがあれば、4 個の仮想プロセッサーが無効になり、4 個の仮想プロセッサーが有効な状態で残ります。同時マルチスレッディング機能が有効になっていれば、それぞれの仮想プロセッサーが 2 個の論理プロセッサーに対応することになります。したがって、8 個の論理プロセッサーが無効になり、8 個の論理プロセッサーが有効な状態で残ります。

以下の例では、フォールディング機能を有効にしていない限定的なワークロードを取り上げます。このワークロードは、パーティションに割り振られている各仮想プロセッサーの最低量だけを消費しています。4 個の仮想プロセッサーがあるシステムで **mpstat -s** コマンドを実行すると、各仮想プロセッサーと、それぞれに関連した 2 個の論理プロセッサーの使用率を示した出力が表示されます。

```
# mpstat -s 1 1

System configuration: lcpu=8 ent=0.5

     Proc0            Proc2            Proc4            Proc6
     19.15%           18.94%           18.87%           19.09%
 cpu0    cpu1    cpu2    cpu3    cpu4    cpu5    cpu6    cpu7
 11.09%   0.07%  10.97%   7.98%  10.93%   7.93%  11.08%   8.00%
```

フォールディング機能を有効にすると、システムは、先ほどの式に基づいて、必要な仮想プロセッサーの数を計算します。さらに、その計算値に基づいて、仮想プロセッサーの数を減らします。つまり、パフォーマンスを落とさずにこの限定的なワークロードを実行するために必要な数に減らすということです。

4 個の仮想プロセッサーがあるシステムで **mpstat -s** コマンドを実行すると、各仮想プロセッサーと、それぞれに関連した 2 個の論理プロセッサーの使用率を示した出力が表示されます。

```
# mpstat -s 1 1

System configuration: lcpu=8 ent=0.5

     Proc0            Proc2            Proc4            Proc6
     54.63%            0.01%            0.00%            0.08%
 cpu0    cpu1    cpu2    cpu3    cpu4    cpu5    cpu6    cpu7
 38.89%  15.75%   0.00%   0.00%   0.00%   0.00%   0.03%   0.05%
```

このデータからわかるとおり、補助的なプロセッサーの使用率と管理の必要性を低減する一方で、1 つの仮想プロセッサーに処理が集中することによって親和性が高まっています。つまり、ワークロードの効率が高くなりました。また、ワークロードが大きくなっても、必要な場合にすべての仮想プロセッサーを使用する機能がフォールディング機能によって干渉されるわけではありません。

ピーク時のロードに合わせて仮想プロセッサーを構成します。

上限なしのパーティションでは、この機能を有効にすると特に大きな効果が得られます。そのパーティションで多数の仮想プロセッサーを構成しても、パフォーマンスの問題が大きくなることはないからです。一般的なルールとして、仮想プロセッサーの数としては、即時のスパイク、つまり短期的に負荷が急増するときの要件に十分に対応できる数を構成することをお勧めします。

5.2 CPU の共有と重みづけ

POWER Hypervisor は、サーバーの各パーティションに対する CPU リソースの割り振りを管理します。各パーティションには、それぞれのシステム・プロファイルに基づいて CPU リソースが割り振られます。専用プロセッサーを使用していないすべてのパーティションは、サーバー内の同じプロセッサー・プールを共用します。サーバーの各パーティションは、上限付き（capped）モードでも上限なし（uncapped）モードでも構成できます。上限付きのパーティションでは、CPU リソースの最大量が事前設定されています。

パーティション・プロファイルで正しい CPU 設定を使用します。

CPU 使用量の上限が設定されていないパーティションは、すべての CPU 割り振り量とプール内のすべての未使用の CPU サイクルを消費できます。使用率の急上昇があり得るアプリケーションでは、上限なし（uncapped）モードに大きな利点があります。

上限なしのパーティションが 1 つだけ存在している環境では、そのパーティションが割り振り量を超えても、すべての未使用のサイクルを使用できます。上限なしのパーティションには、HMC で上限なしの加重値を割り振ります（図 5-1（133 ページ）を参照）。複数の上限なしパーティションが割り振り量を超えた場合は、その加重値に基づいて、未使用の CPU サイクルがハイパーバイザーによって割り振られます。

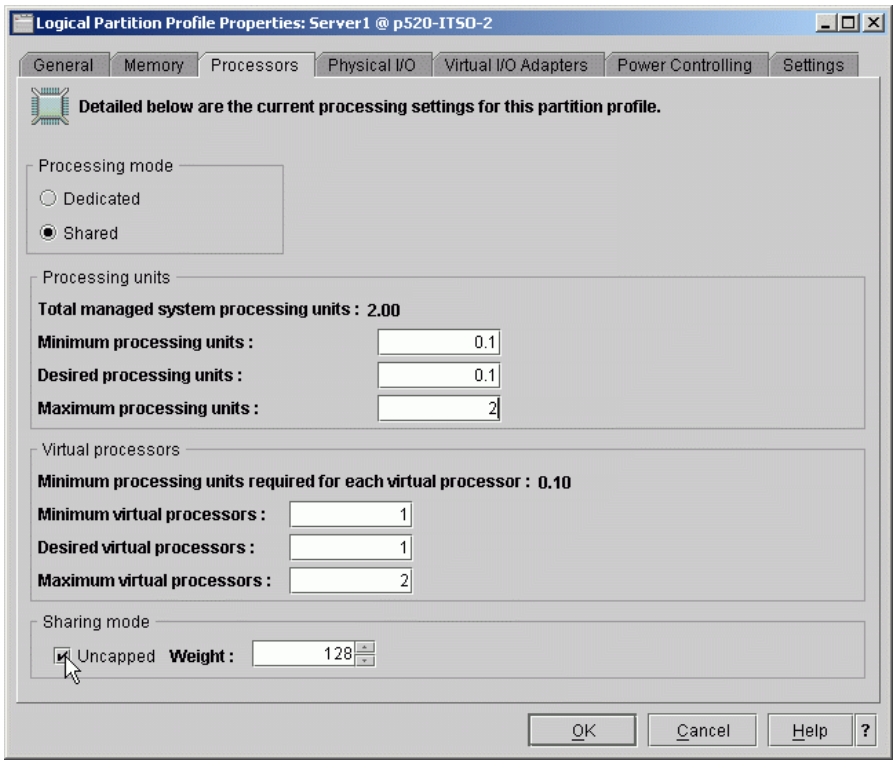


図 5-1　ロジカル・パーティションの上限なしの加重値

ハイパーバイザーは、未使用の CPU サイクルを巡って競合する各パーティションの加重値に基づいて、使用可能なサイクルの比率を割り当てます。加重値が 128 の 2 つのパーティションがサイクルを巡って競合している場合は、使用可能なサイクルの 128/(128+128)=1/2 が各パーティションに割り振られます。

加重値が変わると、比率も変わります。例えば、加重値 64 のパーティションと加重値 128 のパーティションが競合している場合は、使用可能なサイクルがそれぞれの比率で割り当てられます。つまり、加重値 64 のパーティションは 64/(64+128)=1/3 のサイクル、加重値 128 のパーティションは 128/(64+128)=2/3 のサイクルをそれぞれ受け取ります。

既存のシステムに新しいパーティションを追加する場合や、既存のパーティションの加重値を調整する場合は、システム内の他のパーティションに対する影響を考慮することが重要になります。例えば、1 つのアプリケーションに属する 1 つのパーティションがプロセッサーの割り振り量をいつも超過している状況で、別のパーティションの加重値を変更したり、別の LPAR を追加したりすると、そのアプリケーションで使用できるリソースの量が減り、パフォーマンスが悪影響を受ける可能性があります。

最初は加重値を 128 に設定し、それから小刻みに調整します。

上限なしのパーティションを最初に配置するときには、すべての加重値を 128 に設定することをお勧めします。経験を積みながら加重値と上限なし（uncapped）モードの影響を理解できるようになるまで、時間の経過と共に加重値を小刻みに調整しながら、他のパーティションに対する副次作用を回避するようにしてください。

システム・パフォーマンスをモニターしながら、CPU 割り振り量をいつも超過しているパーティションを記録するようにします。そのようなパーティションの中で、パフォーマンスが重要な意味を持つアプリケーションのホストになっているパーティションがあれば、そのパーティションの割り振り量または加重値を増やして、他のパーティションからの影響を回避することが必要になる場合もあります。

5.3 上限なしパーティションとライセンス交付

上限なしパーティションの場合は、共用処理プールに使用可能なリソースがあれば、そのパーティションの処理キャパシティーがキャパシティーの割り振り量を超過することが可能になります。つまり、上限なしパーティションは、サーバー内のアイドル状態のプロセッサー・リソースを使用できるので、全体として物理プロセッサー・リソースの使用効率が高くなります。ただし、上限なしパーティションでは、構成する仮想プロセッサーの総数によって、実際に消費できる物理プロセッサー・リソースの総量が限定されます。例えば、サーバーに 8 個の物理プロセッサーがあるとしましょう。そのサーバーの上限なしパーティションには、キャパシティーの割り振り量として 2 個の処理単位（2 個の物理プロセッサーに相当）が構成されており、4 個の仮想プロセッサーが構成されているとします。このパーティションでは、最大で 4 個の物理プロセッサーを使用できるにすぎません。1 個の仮想プロセッサーは、最大で 1 個の物理プロセッサーに相当するリソースしか消費できないからです。そのパーティションでさらに多くの物理プロセッサー・リソースを使用するには、仮想プロセッサーをさらに追加する動的 LPAR 操作が必要になります。

上限なしパーティションを使用する場合は、ソフトウェア・ライセンスについても検討します。

ソフトウェア・ライセンスの観点からすれば、上限なしパーティションで実行するアプリケーションのライセンスを交付するための基礎になる価格設定構造は、ベンダーによって異なります。アプリケーションは、パーティションのキャパシティーの割り振り量を超えてプロセッサー・リソースを消費する可能性があるので、プロセッサー単位で料金を設定するソフトウェア・ベンダーの中には、追加のプロセッサー・ライセンスの購入を求めるところも少なくありません。上限なしパーティションを実装するかどうかを決定するときには、ライセンス条項の詳細をソフトウェア・ベンダーに問い合わせてください。

5.4 仮想 I/O サーバーのサイズの決定

サーバーのサイズを決定するための唯一確実な方法は、実際のワークロードでサーバーを実行し、モニターし、それから値を調整することです。

この後のセクションでは、調整のための開始点になる基本的な情報を取り上げます。

IBM Systems ハードウェア Information Center には、CPU とメモリーの計画に役立ついくつかの詳細な計算方法が記載されています。

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphb1/iphb1_vios_planning.htm

以下の指針は、あくまでも開始点にすぎず、実動システムで実際に実行してモニターした後に追加の調整が必要になる可能性もありますが、使用する CPU やメモリーの量をある程度正確に計画するために役立ちます。

サーバーのネットワーク使用量とディスク使用量は、基本的に一定ではありません。操作の実行時には、ネットワーク・トラフィックやディスク・アクティビティーが急増します。したがって、仮想 I/O サーバーは、共用 CPU マイクロ・パーティション・モデルを使用するための理想的な環境であり、そのような環境として仮想 I/O サーバーをセットアップすることをお勧めします。

サーバーのサイズの決定に役立つ IBM Systems Workload Estimator ツールの使用方法については、以下のサイトを参照してください。

http://www-912.ibm.com/supporthome.nsf/document/16533356

5.4.1 仮想 I/O サーバーのメモリーの計画

仮想 I/O サーバーも、他のワークロードの場合と同じように、操作のためにシステム・メモリーを必要とします。システム・メモリーの一部は、ネットワーク通信をサポートするために使用されます。そのネットワーク・メモリーは、ネットワーク・トラフィックをバッファーに入れるために使用されるので、綿密な計画を立てるには、メッセージのサイズ（MTU またはジャンボ・フレーム）や使用する共用イーサネット・アダプター（Shared Ethernet Adapter）の数などの要因について確認する必要があります。

仮想 I/O サーバーのメモリー使用については、シンプルな指針があります。

最も簡単なルールとして、シンプルな仮想 I/O サーバーの場合、つまり、いくつかのネットワーク間のブリッジングを提供するだけで、ジャンボ・フレームや大きな MTU 値を使用しない仮想 I/O サーバーの場合は、512 MB のメモリーが必要になります。一方、20 から 30 のクライアントがあり、多数の LUN がある場合や、多めのの安全率を見込んだシステム構成では、期待するパフォーマンスを実現するために、さらに多くのメモリーが必要になる可能性があります。

さらに多くのネットワークのブリッジングを開始したり、ジャンボ・フレームを使用したりする可能性があれば、仮想 I/O サーバーのために 1 GB の RAM を使用します。予備のメモリーが十分にあれば、仮想 I/O サーバーのために 1 GB を使用して柔軟性を最大限に高めるのが理想的ですが、基本的にはそれが上限です。10 Gbps のイーサネット・アダプターを多く実装するのであれば、それだけのメモリー所要量に対応するためにさらに変更が必要になる可能性があります。

5.4.2 仮想 I/O サーバーの CPU の計画

CPU の計画は、以下の 2 つの分野に大別されます。

- 仮想ネットワークをサポートするために必要な CPU
- 仮想ディスクをサポートするために必要な CPU

仮想 I/O サーバーのネットワーク用の CPU 所要量

サーバーの CPU サイズの見積もりを得るための最善の方法は、まず 1 つの値で実行を開始し、実動ワークロードをモニターし、それから値を調整することです。このセクションでは、その開始値を適切に見積もるための簡単な方法を取り上げます。

最初に考えるのは、仮想 I/O サーバーを通過するネットワーク・トラフィックがどれほどの量になるか、という問題です。ここで重要なのは、ネットワーク・アダプターの数やビジー状態の程度ではありません。いずれにしても、ネットワーク・アダプター・カードをサポートするために、最低限の CPU が必要になります。それよりも重要なのは、各ネットワーク・パケットごとに、CPU がネットワーク・チェックサムなどを計算しなければならないということです。

使用する開始点を表 5-1 にまとめます。

表 5-1 1 GB のネットワーク・トラフィックに対して VIOS で必要になる CPU 量の概算

MTU（バイト） CPU 速度（GHz）	1500	9000 またはジャンボ・フレーム
1.5	1.1	0.55
1.65	1.0	0.5
1.9	0.87	0.44
2.2	0.75	0.38

例えば、仮想 I/O サーバーに 1 ギガビットのネットワーク・アダプターが 4 つあり、通常の操作時には全体として約 100 Mbps のトラフィックがあるとします。ただし、夜間のバックアップの時間帯には、2 時間にわたって 1 ギガビットのネットワーク・トラフィックをフルに使用することが分かっています。

パーティションのネットワーク・トラフィックの影響を調べます。

これらの値をネットワーク・サポートのための CPU 所要量に変換できます。すでに見たとおり、そのためには共用 CPU モデルを使用するのが最善です。サーバーに 1.65 GHz の CPU があり、MTU 値 1500 を使用しているとすれば、通常の操作時に必要な CPU はわずか 0.1 個分という計算になります。また、ピーク時の負荷については、1.0 個分の CPU が必要です。さらに、ユーザー・ベースが夜間のバックアップ時にシステムを使用しないのであれば、空きプールに未使用の CPU が大量に存在するはずであり、その CPU をこの場合の CPU 所要量として利用できます。そのためのセットアップは、図 5-2（138 ページ）のようになります。

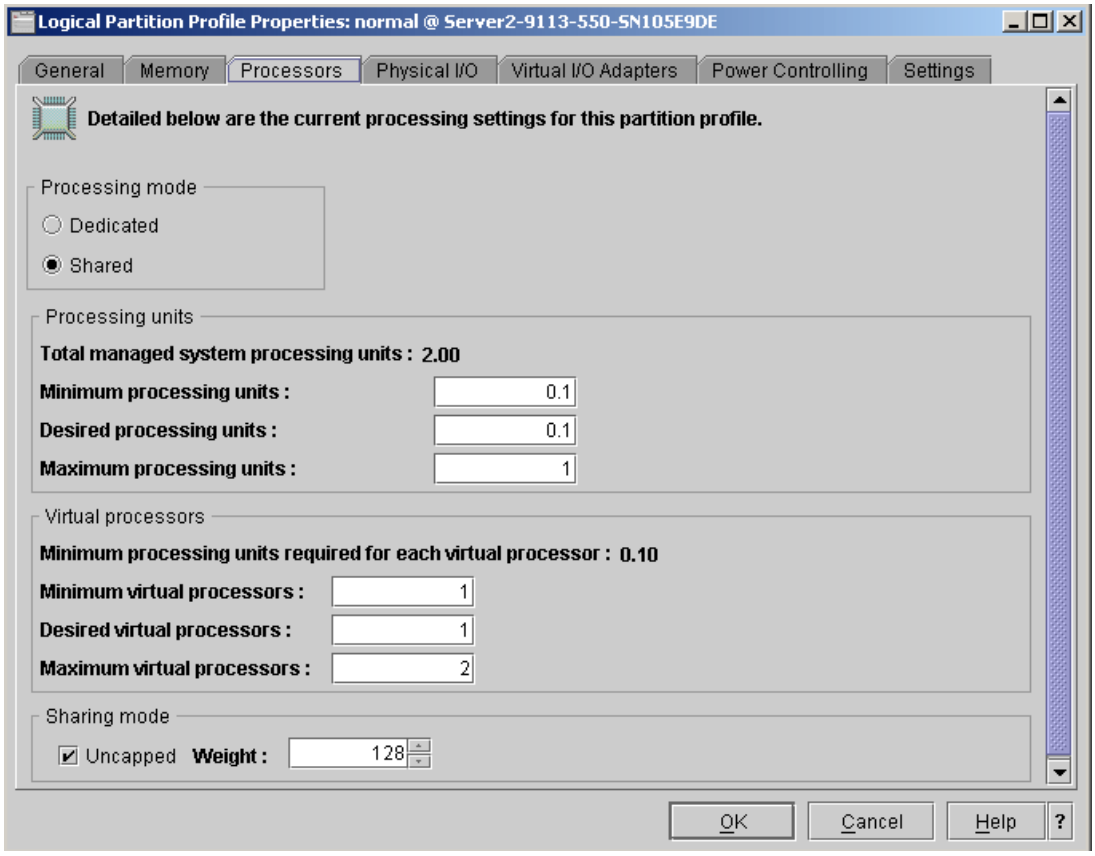


図 5-2　キャパシティー計画 : ネットワークの例

この例では、日常的なネットワーク使用量のために 0.1 個分の CPU を確保していますが、その一方で、上限なしの CPU リソースを使用することによって、仮想 I/O サーバーが必要に応じて（CPU プールから予備の CPU サイクルを使用することで）1.0 個分の CPU を使用できるように設定しています。

ネットワーク・トラフィックをサポートするには CPU が必要ですが、ネットワーク・カードを追加して同じだけのネットワーク・トラフィックを提供しても、追加の CPU が必要になるわけではありません。表 5-1（137 ページ）を使用して、通常の操作をサポートするために必要なネットワーク帯域幅の値を見積もってください。処理単位として、その量（プラス、次のセクションで取り上げるディスクの値）をロジカル・パーティションに確保します。後は、サーバーのプロファイルで上限なしの CPU を使用することによって、仮想 I/O サーバーが空きプールから予備の処理能力を活用して、急増した負荷を処理できるようにしておきます。

仮想 I/O サーバーは、企業のネットワーキング要件のほとんどに対処できる設計になっています。

仮想 I/O サーバーのディスク用の CPU 所要量

ディスクの CPU 所要量を正確に見積もるのは容易ではありません。そのためには、ブロック・サイズや 1 秒あたりの I/O 回数など、I/O に関する詳細な情報を確認することが必要になります。その情報を得た上で、IBM Systems ハードウェア Information Center を参照し、計画のための詳細な計算方法を確認してください。

仮想化環境でディスク I/O が CPU の使用率に与える影響は大きくありません。

大まかな指針として、使用するディスクで何を提供するのかをまず確認してから、それらのディスクのビジー状態がどの程度になるのかを見積もるほうが、おそらく簡単です。例えば、基本的な内部セットとして 4 つの SCSI ディスクがあるだけのシンプルな仮想 I/O サー

バーを取り上げてみましょう。それらのディスクは、クライアントにすべての I/O を提供するためのものであり、10 K rpm のディスクになります。ブロック・サイズとしては、8 KB という標準的なワークロードを使用します。それらのディスクの 1 秒あたりの通常の I/O 回数を最大で約 150 回として計算すれば、CPU は約 0.02 個分というごくわずかな量になります。

1.65 GHz の CPU に関して、1 秒あたりの I/O 回数とさまざまな I/O ブロック・サイズに応じた CPU 所要量を作図すると、図 5-3 のようなグラフになります。

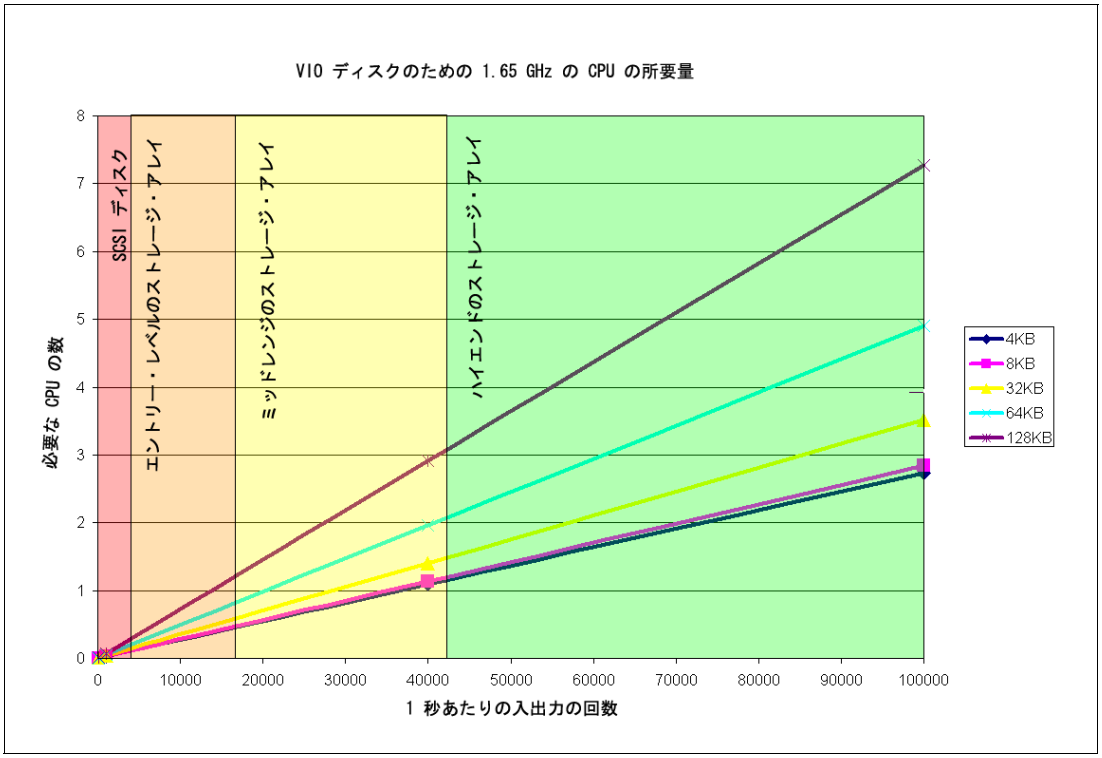


図 5-3　仮想 I/O サーバーの CPU に応じて I/O を提供するために必要なストレージ・アレイのサイズの見積もり

このグラフで示した I/O の回数とストレージ・サブシステムのサイズの見積もりは、ストレージ・サブシステムが 100% 仮想 I/O クライアントだけによって利用されていること（つまり、すべての I/O が仮想ディスクであること）を前提としており、ディスク・サブシステムをグラフに書き込んでいるのは、その程度の負荷を生成するときに必要になるパフォーマンスを示すためにすぎません。ここで示したいのは、ディスク・サブシステムが大きくなればなるほど、必要な CPU 能力も大きくなるということではありません。むしろ、ここから実際に理解できるのは、仮想 I/O サーバーでは、ディスク使用量という観点からして、大規模な CPU を必要とする最も強力なストレージ・サブシステムを用意する必要があるということです。例えば、フルスピードで実行するミッドレンジのディスク・サブシステムは、250 余りのドライブを構成した場合に、グラフの黄色と緑色の境界（つまり、ミッドレンジとハイエンドの境界）付近の領域に位置します。ほとんどの場合、そのストレージは、仮想 I/O サーバーの 100% 専用ではありません。ただし、専用になれば、フルスピードで実行する場合でも、仮想 I/O には、ほとんどのブロック・サイズで 1 個分の CPU しか必要としません。

大半のシステムの I/O 要求は、左下のセクションに該当するので、ディスクに対応するための開始点として 0.1 個分から 0.2 個分の CPU という数値を見積もるのは、ほとんどの場合に適しているといえます。

構成を実動システムに移す前に、必ずテストすることをお勧めします。

5.4.3 二重仮想 I/O サーバーのサイズ見積もり

二重仮想 I/O サーバーのシステムを計画している場合は、それぞれの仮想 I/O サーバーでクライアントのワークロードを半分ずつサポートするようにサイズを設定できます。1 つの仮想 I/O サーバーを保守のために構成から除去する場合に備えて、上限なしの CPU を使用して、予備の CPU を十分に確保しておく必要があります。

そのための詳しい方法については、4.7、『MPIO の計画と配置』（121 ページ）と 3.8.2、『二重仮想 I/O サーバー構成の拡張可用性オプション』（80 ページ）を参照してください。

5.4.4 仮想 I/O サーバーのサイズ見積もりに関するまとめ

ほとんどのワークロードでは、I/O の急増を予期しなければなりません。例えば、クライアントがアプリケーションにアクセスすると、データがクライアントに送信されるときにディスクとネットワークのアクティビティーが急増します。あるいは、Web サイトに変更を送信した場合も、バックエンド・データベースを更新するためのディスク・アクティビティーが急増し、ある程度のネットワーク使用量が発生します。

このような散発的なアクティビティーについては、仮想 I/O サーバーでマイクロ・パーティションと上限なしの CPU の機能を最大限に活用するのが、最も妥当な方法です。仮想 I/O サーバーでは、開始点として 1 GB を使用し、そこから必要に応じて規模を増減するのが、基本な指針になります。CPU については、それよりもさらに検討が必要ですが、正確を期すためには、実際のロードでシステムを実行し、それから値を調整するしか方法はありません。その調整を始めるための程よい開始点として、すでに取り上げた見積もりは役立つはずです。

二重仮想 I/O サーバーの場合は、よく計画して 2 つの小さな仮想 I/O サーバーを構成し、仮想 I/O クライアントを半分ずつサポートするようにサイズを設定できます。仮想 I/O の弾力性を確保するための追加のキャパシティーは、サーバーの上限なしの機能によって用意できます。

5.5 仮想 I/O サーバーのパフォーマンス測定

パフォーマンスの測定は、基本的に以下の 2 つに分類できます。

- 短期の測定（テスト、サイズ設定、トラブルシューティングのシナリオで使用します）。多くの場合、長期の測定によって初期化されます。
- 長期の測定（キャパシティー管理の入力データとして使用します）。パフォーマンスの傾向やワークロードの変化についてのデータが含まれています。

5.5.1 短期のパフォーマンス測定

短期モニターの開始点としてこれらのツールを使用します。

仮想 I/O サーバーでは、短期のパフォーマンス測定のために以下のツールを使用できます。

viostat CPU の統計とシステム全体、アダプター、tty デバイス、ディスク、CD-ROM の I/O の統計を報告します。

netstat ネットワークの統計を報告します。

topas CPU、ネットワーク、プロセス、メモリーなど、ローカル・システムの統計を選択して報告します。

仮想 I/O サーバーで短期のパフォーマンス測定を開始するときに、特定のターゲット（例えば、ネットワークのパフォーマンスの低下など）がない場合は、**topas** コマンドを使用します。このツールによって、仮想 I/O サーバーの全体的なパフォーマンスの状況を確認できます。**topas** コマンドを実行すると、**viostat** コマンドまたは **netstat** コマンドによってさらに調査できる一般的な問題点を強調表示したシステム統計が出力されます。その 2 つのコマンドからは、さらに詳しい情報が出力されます。パフォーマンスの測定を計画し、システムの時刻と状況を前提事項と一緒に文書化しておくようにしてください。パフォーマンスのボトルネックを調べることが必要になったときに、それは貴重な情報になります。

例 5-1 は、**topas** コマンドの出力です。

例 5-1 10 秒のインターバルを設定した topas コマンドの出力

```
$ topas -interval 10
Topas Monitor for host:    lpar01               EVENTS/QUEUES    FILE/TTY
Mon Jul 10 08:40:37 2006   Interval: 10         Cswitch      83  Readch     2722
                                                Syscall      95  Writech      39
Kernel    2.6   |#                          |   Reads          4  Rawin         0
User      1.6   |#                          |   Writes         0  Ttyout       39
Wait      0.0   |                           |   Forks          0  Igets         0
Idle     95.9   |########################## |   Execs          0  Namei         5
Physc =  0.01                    %Entc=   6.6   Runqueue    0.1  Dirblk        0
                                                Waitqueue   0.0
Network  KBPS   I-Pack  O-Pack   KB-In  KB-Out
en2       0.4     3.6     0.2     0.4     0.1   PAGING           MEMORY
lo0       0.0     0.2     0.2     0.0     0.0   Faults        8  Real,MB    1024
                                                Steals        0  % Comp     33.0
Disk    Busy%     KBPS     TPS KB-Read KB-Writ  PgspIn        0  % Noncomp   6.9
hdisk3    0.0      0.0     0.0     0.0     0.0  PgspOut       0  % Client    6.9
hdisk2    0.0      0.0     0.0     0.0     0.0  PageIn        0
                                                PageOut       0  PAGING SPACE
WLM-Class (Active)     CPU%    Mem%  Disk-I/O%  Sios          0  Size,MB    1536
System                    1      17          0                  % Used      0.6
Unmanaged                 1      19          0  NFS (calls/sec)  % Free     99.3
                                                ServerV2       0
Name            PID CPU% PgSp Class             ClientV2       0   Press:
xmwlm        368860  0.2  0.8 System            ServerV3       0   "h" for help
topas        360476  0.1  0.9 System            ClientV3       0   "q" to quit
```

viostat コマンドの出力は、AIX 5L バージョンの **iostat** コマンドの出力とよく似ています。例 5-2 は、**viostat** コマンドの出力例です。この出力は、AIX 5L バージョンの **iostat** コマンドの出力と同じように解釈できます。追加情報については、AIX 5L の製品資料を参照してください。

例 5-2 パラメーターなしの viostat コマンド

```
$ viostat
System configuration: lcpu=2 drives=4 ent=0.10 paths=4 vdisks=3

tty:      tin         tout    avg-cpu: % user % sys % idle % iowait physc % entc
          0.1          2.0                 0.3   0.8   98.9      0.0   0.0    1.9

Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk2           0.0       0.0       0.0          9     14428
hdisk3           0.0       0.2       0.0      18945    108860
hdisk0           0.0       0.4       0.1      21574    230561
hdisk1           0.0       0.3       0.1          6    230561
$
```

viostat コマンドでは、例 5-3（142 ページ）の出力のような特殊な出力を生成することもできます。ただし、この出力は長期の測定には使用しないでください。情報を記録するために大量のディスク・スペースが必要になるからです。この出力の内容を制限するには、例 5-3（142 ページ）のように、パラメーター・リストの中で hdisk 番号を記述します。

例 5-3 viostat による hdisk2 と hdisk3 の拡張ディスク出力

```
$ viostat -extdisk hdisk2 hdisk3
System configuration: lcpu=2 drives=4 paths=4 vdisks=3

hdisk2         xfer:  %tm_act      bps      tps      bread      bwrtn
                         0.0     22.2      0.0        0.1       22.2
                read:      rps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      6.0      0.3     12.5           0          0
               write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      9.4      1.6     18.1           0          0
               queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0      0.0      0.0        0.0          7
hdisk3         xfer:  %tm_act      bps      tps      bread      bwrtn
                         0.0    196.0      0.0       29.0      167.1
                read:      rps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      6.6      0.2     11.5           0          0
               write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      6.9      1.2     20.1           0          0
               queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.8      0.0     30.0      0.0        0.0       5724
--------------------------------------------------------------------------------
$
```

仮想 I/O サーバー上での **netstat** コマンドの出力は、AIX 5L バージョンの **netstat** コマンドの出力とよく似ています。仮想 I/O サーバーの **netstat** コマンドは、パフォーマンス・データだけでなく、ルーティング・テーブルなどのネットワーク・データを表示することもできます。パフォーマンス関連データを表示するには、例 5-4 のように、インターバルを指定してコマンドを実行しなければなりません。**netstat** コマンドは、Ctrl+C で停止する必要があるので、長期の測定スクリプトで使用するのは容易ではありません。**viostat** コマンドとは異なり、指定できるのはインターバルだけであり、カウントを指定することはできないからです。

例 5-4 netstat コマンドの出力

```
$ netstat 1
    input   (en2)      output           input   (Total)    output
 packets  errs  packets  errs colls  packets  errs  packets  errs colls
 1038255     0    45442     0     0  1075627     0    83225     0     0
       9     0        1     0     0        9     0        1     0     0
       2     0        1     0     0        2     0        1     0     0
       3     0        1     0     0        3     0        1     0     0
       3     0        1     0     0        3     0        1     0     0
       8     0        1     0     0        8     0        1     0     0
^C$
```

5.5.2 長期のパフォーマンス測定

長期のパフォーマンス・モニターでは、ワークロード・マネージャーを使用できます。

長期のパフォーマンス測定で使用する主なツールは、以下のとおりです。

wkldmgr ワークロード・マネージャー制御コマンド。ワークロード・マネージャーを開始したり停止したりします。

wkldagent 記録エージェント。システム・パフォーマンスとワークロード・マネージャーの測定データを /home/ios/perf/wlm にあるバイナリー・ファイルに記録します。

wkldout **wkldagent** の出力をテキスト形式に変換して端末に書き出します。

仮想 I/O サーバーでは、**viostat** や **netstat** などのコマンドを長期のパフォーマンス・ツールとして使用することもできます。スクリプトの中に組み込んだり、**crontab** コマンドによって開始したりすれば、カスタマイズ・レポートを生成できます。

長期のパフォーマンスを測定するために仮想 I/O サーバーに用意されているツールは、ワークロード・マネージャー（WLM）のサブセットです。ワークロード・マネージャーの各コンポーネントをパッシブ・モードで実行すると、CPU、メモリー、ディスク I/O の使用率がモニターされますが、調整は行われません。仮想 I/O サーバーのリソース割り振りメカニズムも影響を受けません。

ワークロード・マネージャーを基盤とするコマンドを使用してパフォーマンスのモニターを開始するには、例 5-5 のように、まず **wkldmgr** コマンドによってワークロード・マネージャーを開始して状況を確認します。このコマンドは、パフォーマンスのモニターや情報の記録を開始するわけではありません。

例 5-5 wkldmgr の開始と状況の確認

```
$ wkldmgr -start
$ wkldmgr -status
WorkLoad Manager is running
$
```

システム・パフォーマンス・データの記録を開始するには、**wkldagent** コマンドを使用します。例 5-5 は、記録セッションの開始を示しています。ワークロード・マネージャー・エージェントの状況を確認するには、**wkldagent** コマンドを使用するか、/home/ios/perf/wlm/xmwlm.log1 ファイルに対して **cat** コマンドを実行します。

例 5-6 wkldagent によるエージェントの開始と状況の確認

```
$ wkldagent -start
$ wkldagent -status
WorkLoad Manager Agent is running
$ cat /home/ios/perf/wlm/xmwlm.log1

=====  xmtrend scheduled at Jul 10 10:50:42 =====
Initially logging to "/home/ios/perf/wlm/xmwlm.log1"
Initializing WLM and checking the state
WLM is turned on, activating the recording...
$
```

出力データは、/home/ios/perf/wlm にあるバイナリー・ファイルに収集されます。ファイル名は、xmwlm.*yymmdd* という形式（例えば xmwlm.060710）になります。このファイルを実際に読める形式にするには、**wkldout** コマンドを使用してテキスト形式に変換しなければなりません。このコマンドは、**wkldagent** コマンドによる記録中にも実行できます。例 5-7 はその一例です。入力ファイルと出力ファイルの絶対パスを必ず指定してください。

例 5-7 wkldout コマンドの出力

```
$ ls -l /home/ios/perf/wlm
total 160
-rw-r--r--   1 root     staff         30872 Jul 10 11:09 xmwlm.060710
-rw-r--r--   1 root     staff           190 Jul 10 10:50 xmwlm.log1
$ wkldout -filename /home/ios/perf/wlm/xmwlm.060710
Time="2006/07/10 11:06:03", WLM/System/DISK/hardmax=100.00
Time="2006/07/10 11:06:03", WLM/System/DISK/softmax=100.00
Time="2006/07/10 11:06:03", WLM/System/DISK/min=0.00
Time="2006/07/10 11:06:03", WLM/System/DISK/shares=1.00
Time="2006/07/10 11:06:03", WLM/System/DISK/desired=100.00
Time="2006/07/10 11:06:03", WLM/System/DISK/consumed=0.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/hardmax=100.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/softmax=100.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/min=0.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/shares=1.00
Time="2006/07/10 11:06:03", WLM/Shared/DISK/desired=100.00
.
. （途中省略）
.
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/hardmax=100.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/softmax=100.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/min=0.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/shares=1.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/desired=100.00
Time="2006/07/10 11:09:06", WLM/Unclassified/CPU/consumed=0.00
Time="2006/07/10 11:09:06", WLM/System/tier=0.00
Time="2006/07/10 11:09:06", WLM/Shared/tier=0.00
Time="2006/07/10 11:09:06", WLM/Default/tier=0.00
Time="2006/07/10 11:09:06", WLM/Unmanaged/tier=0.00
Time="2006/07/10 11:09:06", WLM/Unclassified/tier=0.00
Time="2006/07/10 11:09:06", WLM/wlmstate=1.00
Time="2006/07/10 11:09:06", PagSp/%totalused=0.70
Time="2006/07/10 11:09:06", Proc/swpque=0.00
Time="2006/07/10 11:09:06", Proc/runque=0.93
Time="2006/07/10 11:09:06", CPU/glwait=0.00
Time="2006/07/10 11:09:06", CPU/gluser=1.56
Time="2006/07/10 11:09:06", CPU/glkern=2.28
$ ls -l /home/ios/perf/wlm
total 168
```

```
-rw-r--r--   1 root     staff         32924 Jul 10 11:12 xmwlm.060710
-rw-r--r--   1 root     staff         41015 Jul 10 11:11 xmwlm.060710_01
-rw-r--r--   1 root     staff           190 Jul 10 10:50 xmwlm.log1
$
```

例 5-7（144 ページ）のように **ls** コマンドを実行した場合は、**wkldout** コマンドのテキスト出力として xmwlm.yymmdd_01 ファイルが生成されます。**more** コマンドを使用して出力をページ単位で表示したり、**ftp** コマンドを使用して別のサーバーに移動したりすることも可能です。**wkldagent** コマンドで作成されるバイナリー・ファイルは保管され、各ファイルに 1 日分のデータが書き込まれます。それらのファイルを保管する場合は、/home/ios/perf/wlm ディレクトリーからコピーしてください。

5.6 仮想 I/O クライアントの同時マルチスレッディング機能

マルチスレッディング機能にはさまざまな形態があります。POWER5 アーキテクチャーに実装されている形態のマルチスレッディング機能のことを同時マルチスレッディング機能といいます。この機能は、1 つのプロセッサーで 2 つの別々のスレッドを同時に実行することを可能にしたハードウェア設計の機能拡張です。

5.6.1 同時マルチスレッディング機能が適している環境

仮想化環境では同時マルチスレッディング機能を使用できます。

同時マルチスレッディング機能は、一般的なワークロードで全体的なスループットを重視する場合に適しています。ほとんどのトランザクションの実行要件が似通っていれば、そのワークロードは同時マルチスレッディング機能に適しているといえます。例えば、Web サーバー、オンライン・トランザクション・アプリケーション・サーバー、データベース・サーバーなどは、いずれも以下のような理由で適しています。

- ほとんどのスレッドの実行要件が似通っています。
- ランダム・データ・アクセスでは、データをキャッシュにロードしたり、ディスクからロードしたりするために、スレッドをスリープ状態にする必要があります。
- 全体的なスループットが最重要のパフォーマンス要因です。

1 命令あたりのサイクル数（CPI）のカウントが非常に高いトランザクション・タイプは、プロセッサーやメモリーのリソースの使用効率が低くなる傾向があり、同時マルチスレッディング機能の利点を最大限に活用できる場合が多いといえます。そのような高い CPI の基本的な原因は、巨大な作業セットでキャッシュ・ミス率が高くなることです。ほとんどの商用トランザクションにはその特性がありますが、2 つの同時スレッドが命令やデータを共用するのか、それぞれが完全に独立しているのかによっても違いが生じます。命令やデータを共用するトランザクション（例えば、主にオペレーション・システムで実行されるトランザクションや、1 つのアプリケーションで実行されるトランザクションなど）のほうが、同時マルチスレッディング機能の利点を十分に活用できる傾向があります。

CPI とキャッシュ・ミス率が低いワークロードの場合は、効果が薄れます。試しに、ハイパフォーマンス・コンピューティングで同時マルチスレッディング機能を有効にして、パフォーマンスをモニターしてみてください。短いループによるデータ集中型のトランザクション・タイプであれば、キャッシュやメモリーの競合が多くなり、パフォーマンスが低下する可能性があります。

5.6.2 同時マルチスレッディング機能が適していない環境

同時マルチスレッディング機能は、特殊性の高いワークロードにはあまり適していないといえます。個々のソフトウェア・スレッドの大半がプロセッサーやメモリーのリソースを効率よく使用しているトランザクションでは、同時マルチスレッディング機能の効果がほとんどありません。例えば、実行時間の長い浮動小数点集中型のトランザクションの場合は、同時マルチスレッディング機能を無効にしたほうがパフォーマンスが高くなる可能性があります。

ほとんどの商用トランザクション・ワークロードは、同時マルチスレッディング機能を使用することによって、パフォーマンスとスループットを改善できます。

5.6.3 仮想 I/O クライアントで同時マルチスレッディング機能を使用するかどうかの判断

同時マルチスレッディング機能によってパフォーマンスが向上するかどうかは、アプリケーションのタイプやスレッドの数によって左右されます。それでも、ほとんどの場合は、同時マルチスレッディング機能をオンにすることによって、パフォーマンスが大幅に改善されます。

簡単なテストを実施して、実動ワークロードを測定することをお勧めします。

IBM のテストに基づき、同時マルチスレッディング環境で実行するアプリケーションのパフォーマンスに関する一般的なルールをまとめれば、以下のようになります。

- 標準的な商用環境で実行するトランザクションの場合は、同時マルチスレッディング機能を有効にすることによって、パフォーマンスが大幅に向上します。
- トランザクションの数が増える場合は、同時マルチスレッディング機能によってパフォーマンスが改善されます。

各種のトランザクション・タイプで実験してみると、同時マルチスレッディング機能による効果の程度はさまざまであり、改善幅は 19% から 35% の範囲でした。平均すると、ほとんどのワークロードは、同時マルチスレッディング・モードで実行するほうがパフォーマンスが高くなります。

重要 : System p5 サーバーには、同時マルチスレッディング・モードのほうが適しているトランザクションについて、AIX 5L V5.3 が同時マルチスレッディング・モードと単一スレッド・モードの切り替えを動的に実行するための機能が用意されています。

関連資料

このセクションでは、この Redpaper で説明されているトピックの詳細を理解するために特に適していると思われる資料を取り上げます。

IBM Redbooks

これらの資料の注文については、『IBM Redbooks の入手方法』（150 ページ）を参照してください。ただし、ここで取り上げる資料の中には、ソフトコピー版しか用意されていないものもあります。

- 「*Advanced POWER Virtualization on IBM System p5*」（SG88-4018）
- 「*Advanced POWER Virtualization on IBM @*server *p5 Servers: Architecture and Performance Considerations*」（SG24-5768）
- 「*NIM: From A to Z in AIX 4.3*」（SG24-5524）
- 「*Virtual I/O Server Integrated Virtualization Manager*」（REDP-4061）
- 「*Hardware Management Console（HMC）Case Configuration Study for LPAR Management*」（REDP-3999）

オンライン・リソース

関連資料の情報源として、以下の Web サイトがあります。

- Advanced POWER Virtualization 機能と仮想 I/O サーバーに関する詳細な資料

- 最新の 「*Multipath Subsystem Device DriverUser's Guide*」
  http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S7000303
- IBM System Planning Tool
  http://www.ibm.com/servers/eserver/support/tools/systemplanningtool/
- 仮想 I/O サーバー・ホーム・ページ
  http://techsupport.services.ibm.com/server/vios/home.html
- 仮想 I/O サーバー・ホーム・ページ（代替）
  http://www14.software.ibm.com/webapp/set2/sas/f/vios/home.html
- Capacity on Demand
  http://www.ibm.com/systems/p/cod/
- SDDPCM ソフトウェア・ダウンロード・ページ
  http://www.ibm.com/support/docview.wss?uid=ssg1S4000201
- SDD ソフトウェア・ダウンロード・ページ
  http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&dc=D430&uid=ssg1S4000065&loc=en_US&cs=utf-8&lang=en

- 仮想 I/O サーバーに対応しているハードウェア

  http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/VIOS_datasheet_063006.html
- 仮想 I/O サーバーの資料

  http://techsupport.services.ibm.com/server/vios/documentation/home.html
- IBM Systems Workload Estimator

  http://www-912.ibm.com/supporthome.nsf/document/16533356
- IBM Systems ハードウェア Information Center

  http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp
- IBM eServer pSeries および AIX インフォメーション・センター

  http://publib16.boulder.ibm.com/pseries/index.htm
- Virtualization wiki

  http://www-941.ibm.com/collaboration/wiki/display/virtualization/Home
- IBM System p の IBM Advanced POWER Virtualization に関する Web ページ

  http://www.ibm.com/systems/p/apv/

IBM Redbooks の入手方法

Redbooks、Redpapers、ヒント、ドラフト版の資料、追加の資料の検索、表示、ダウンロード、またはハードコピー版の Redbooks や CD-ROM の注文については、以下の Web サイトを参照してください。

ibm.com/redbooks

IBM による支援サービス

IBM サポートとダウンロード

ibm.com/support

IBM グローバル・サービス

ibm.com/services

IBM System p Advanced POWER Virtualization ベスト・プラクティス

Advanced POWER Virtualization 機能を効率的に使用するために作成された推奨プラクティス集

既存の IBM System p 資料に記載されている知識を基に構成された追加資料

経験豊富なシステム管理者やアーキテクトのための価値ある参考文献

この IBM Redpaper には、IBM System p5 サーバーで Advanced POWER Virtualization 機能を使用するときに活用できる各種機能の計画、インストール、保守、操作に関するベスト・プラクティスが記載されています。

Advanced POWER Virtualization 機能は、IBM POWER5 プロセッサーと POWER5+ プロセッサーを基盤とするシステムで仮想 I/O 環境をサポートしたり管理したりするために使用するハードウェアとソフトウェアの組み合わせです。主なテクノロジーは、以下のとおりです。

- 仮想イーサネット
- 共用イーサネット・アダプター（Shared Ethernet Adapter）
- 仮想 SCSI サーバー
- マイクロ・パーティショニング

この Redpaper は、最初から最後まで読み通すこともできますが、本来の意図は、関連のあるトピックにアクセスするためのノートブックとして活用していただくことです。本書は、*「Advanced POWER Virtualization on IBM System p5」*（SG88-4018）の内容をさらに拡充し、大小さまざまなインストール済み環境を運用するクライアント・サイトやアウトソーシング・サイトで勤務する厳選チームによって用意された追加のサンプルやシナリオを記載しています。本書に載せているのは、実際の経験を基盤とした厳選されたベスト・プラクティスです。

Advanced POWER Virtualization 機能、論理パーティショニング、および IBM AIX 5L に関する実践レベルの理解と、ネットワークや VLAN タギングに関する基本的な理解が必要です。

SG88-4019-00

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

実際の経験に基づく技術情報の蓄積

IBM Redbooks は、IBM International Technical Support Organization によって作成されています。IBM、顧客企業、パートナー企業で勤務する世界各地の専門家によって、現実的なシナリオに基づくタイムリーな技術情報が作成されました。それぞれの環境に IT ソリューションを効率的に実装するために役立つ具体的な推奨事項も記載されています。

詳細情報：
ibm.com/redbooks