



Oracleホワイト・ペーパー  
2010年5月

# ORACLE<sup>®</sup> VM SERVER FOR SPARC<sup>®</sup> を使用してネットワーク可用性を高める ためのベスト・プラクティス

はじめに .....	1
Oracle VM Server for SPARCの概要 .....	1
このホワイト・ペーパーについて .....	2
ネットワーク可用性の概要 .....	3
冗長性による可用性 .....	3
Oracle SolarisのIPマルチパス .....	5
Oracle VM Server for SPARCのIPマルチパス .....	6
ネットワーク可用性の構成 .....	10
Oracle VM Server for SPARCでIPMPを使用するための前提条件 .....	11
2つのI/Oドメインを使用したIPMPの構成 .....	13
2つ目のI/Oドメインの構成 .....	13
単一のI/Oドメインを使用したIPMPの構成 .....	18
I/Oドメインの構成 .....	18
ゲスト・ドメインでのIPMPの構成とテスト .....	20
まとめ .....	20
著者について .....	21
謝辞 .....	21
参考資料 .....	21

## はじめに

リソースにアクセスしてクライアントにサービスを提供するために、事実上、近代のデータセンターのすべてのサーバーでネットワーク接続が重要になっています。ネットワーク接続を強化するためのベスト・プラクティスとして、機器の問題や人為的ミスが原因でネットワーク・アダプタ、ケーブル、または上流スイッチに障害が発生した場合に、2番目のパスが引き続きトラフィックを伝送するよう、ネットワークへの複数の物理パスを構成することが推奨されます。エラーが発生した場合にリンク間でフェイルオーバーを実行するソフトウェアによって、冗長の物理接続を補完する必要があります。Oracle<sup>®</sup> Solaris オペレーティング・システムでは、IPマルチパス (IPMP) をサポートしています。そのため、使用できるすべてのリンクでアウトバウンド・トラフィックの負荷を共有すると同時に、複数のインターフェースでホストされているIPアドレスのグループを自動的にフェイルオーバーするよう、システムを構成できます。

物理ネットワークへの物理サーバーの接続性を最大限に高めるためのベスト・プラクティスは十分把握されていますが、これらのルールを仮想環境にどのように適用すればよいでしょうか。接続とサーバー自体が仮想の場合、サーバーへの複数のネットワーク接続を使用するとは、どのようなことでしょうか。このテクニカル・ホワイト・ペーパーは、物理環境のI/Oに関するベスト・プラクティスを、論理ドメインによってサポートされる仮想環境に適用するのを支援するために計画されたシリーズの第二弾となります。

## Oracle VM Server for SPARCの概要

Oracle VM Server for SPARC<sup>®</sup> (旧名Sun Logical Domain) は、CoolThreadsテクノロジーを採用したオラクル・サーバーでサポートされている仮想化テクノロジーで、UltraSPARC<sup>®</sup> T1、T2、およびT2 Plusプロセッサを搭載しています。Oracle VM Server for SPARCでは、サーバー・リソースをパーティション化して別の仮想マシンに割り当てることができます。リソースをCPUスレッド、暗号化プロセッサ、メモリ、およびPCIバスに割り当てることができます。

Oracle VM Server for SPARCのアーキテクチャは、シン・ハイパーバイザー・レイヤーで実行される4つのクラスの仮想マシンで構成されています (図1)。単一の制御ドメインが仮想化環境を管理します。1つ以上のI/Oドメインが実際のI/Oデバイスを保有し、このI/Oデバイスは、サービス・ドメインがゲスト・ドメインに対して仮想I/Oサービスを提供するために使用されます。各ドメインは、サーバー・ハードウェアの独自の専用パーティションで動作します。1つのドメインが複数の役割を果たすことができます。たとえば、I/Oドメインとサービス・ドメインは通常、物理I/Oと仮想I/Oを処理する1つのドメインに統合されます。多くの構成では、制御ドメインはI/Oとサービスのドメインと統合されます。このホワイト・ペーパーでは、I/Oとサービスが統合されたドメインを分かりやすくI/Oドメインと呼んでいます。

論理ドメインを使用すると、各I/OドメインがPCIバスおよびPCIバスに接続されたデバイスを'所有'できるよう、個々のPCIルート連結 (nexus) ノード (このホワイト・ペーパーではPCIバスと呼ばれる) をI/Oドメインに割り当てることができます。2つのPCIバスを装備したサーバーでは、ゲスト・ドメインからI/Oリソースに複数のパスを提供するよう、2つのI/Oドメインを構成できます。

I/Oドメイン、そのPCIバス、またはその周辺機器に障害が発生した場合や、ドメインを再起動する必要がある場合、ゲスト・ドメインを適切に構成していれば、ゲスト・ドメインでは2番目のI/Oドメインを使用してI/Oリソースに引き続きアクセスできます。

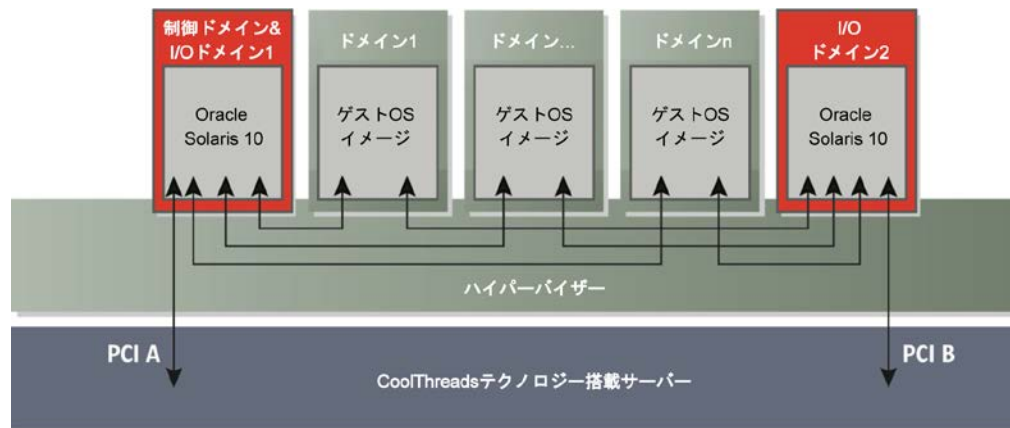


図1. Oracle VM Server for SPARCは、それぞれがサーバー・ハードウェアの独自のセキュア・パーティションを持つ複数のゲスト・ドメインをサポート。I/Oは、1つ以上のPCIバスを「所有」するI/Oドメインにより処理

Oracle Solaris OSで論理ドメインとIPMPを組み合わせて使用すると、ゲスト・オペレーティング・システムと物理ネットワーク間のネットワーク可用性を高めることができます。これらのテクノロジーの組み合わせは、1つまたは2つのPCIバスを装備した、CoolThreadsテクノロジーを採用したサーバーで使用できます。UltraSPARC T2プロセッサベースのシステムの場合は、オラクルの10ギガビット・イーサネット・ネットワーク・インタフェース・ユニット (NIU) などの同等のI/Oデバイスを装備した、CoolThreadsテクノロジーを採用したサーバーで使用できます。これにより、単一のI/Oパスに障害が発生した場合に、ゲストOSによってネットワーク接続が維持されます。このテクニカル・ホワイト・ペーパーでは、サーバー自体のリソース、つまりサーバー自体のPCIバス、組み込みのディスク・コントローラ、およびディスク・ドライブを使用して、冗長性によってデータの信頼性を確立するためのアプローチとトレードオフについて説明します。Oracle VM Server for SPARCは、仮想環境にI/Oの可用性技術と信頼性技術を実装する可能性を広げるもので、このホワイト・ペーパーはこの広範なテーマに対処するシリーズの第一弾となります。当然のことながら、このホワイト・ペーパーで説明している技術にはすべてメリットとデメリットがあります。データセンター要件に対する各ソリューションのメリットを慎重に比較した上で、どの技術を実装するのか選択する必要があります。

## このホワイト・ペーパーについて

このホワイト・ペーパーでは、物理ネットワークと論理ドメイン間で複数のネットワーク・パスを使用してネットワーク可用性を高めるためのアプローチとトレードオフについて説明します。

- "ネットワーク可用性の概要"では、ネットワークへの複数のパスを使用して可用性を高める方法、およびオラクル・サーバーで利用できるソリューションについて概要を説明します。
- "ネットワーク可用性の構成"では、これらのソリューションを構成するための共通の背景情報と手順について説明します。
- "2つのI/Oドメインを使用したIPMPの構成"では、2つのI/Oドメインを使用してIPMPを構成するための詳細な手順について説明します。

- "単一のI/Oドメインを使用したIPMPの構成"では、単一のI/Oドメインを使用してIPMPを構成するための詳細な手順について説明します。

## ネットワーク可用性の概要

ネットワーク可用性を高める、つまり物理サーバーまたは仮想サーバーがネットワーク接続を維持する能力を高めるには、シングル・ポイント障害を排除する冗長リンクを使用します。ネットワークからサーバーまたはゲストOSへのパス上には、人為的ミスやハードウェア障害が原因でネットワーク・リンクに障害が発生して接続が切断される可能性がある、多くのポイントが存在します（表1）。スイッチ・ポートに障害が発生したり、スイッチ・ポートが誤って構成されたりすることにより、ネットワーク・トラフィックのフローが妨げられる可能性があります。ケーブルについては、間違ったポートに接続したり、ケーブルのコンダクタやコネクタが故障したりする可能性があります。また、コネクタを不適切に取り付けたり、タイトな角度で曲げたりしたために、ケーブルのツイストのパターンが分断されて、ケーブルに障害が発生する可能性があります。サーバーでは、コネクタと同様に、ネットワーク・インタフェースに障害が発生する可能性があります。

サーバー内部では、PCIバス・コントローラやプロセッサへのインタフェースの障害は、サーバー自体の障害を引き起こす壊滅的なイベントに関連する可能性が高いです。ただし、仮想化を使用している場合、ソフトウェアが物理NICとゲスト・オペレーティング・システムの間に入介入するため、障害が起こり得る箇所がさらに増えます。論理ドメインを使用している場合、I/Oドメインは物理NICが接続されているPCIバスを'所有'し、ハードウェアとゲストOSの仲介役として機能します。特にI/Oドメインは、仮想NICによってゲスト・ドメインが接続されている仮想スイッチを使用して構成されます。ゲスト・ドメイン自体では、物理インタフェースと同様に管理可能な仮想ネットワーク・インタフェースが認識されます。I/Oドメインに障害が発生する可能性がある場合、またはOSの更新を適用した後でI/Oドメインを再起動する必要がある場合は、障害が起こり得る箇所のリストにI/Oドメインを含める必要があります。

表1. ネットワーク関連の障害箇所

ネットワーク・コンポーネント	障害の原因
スイッチ・ポートおよびスイッチ	ハードウェア障害、不適切な構成、ファームウェアの更新
ケーブル接続	人為的ミス、コネクタやコンダクタの故障
NIC	ハードウェア、ソフトウェア、またはコネクタの障害
PCIバス	ハードウェア障害。他のサーバー機能に深刻な影響を及ぼす可能性あり（ディスクへのアクセスなど）
I/Oドメイン	OSのアップグレードで再起動が必要、Oracle VM Server for SPARCの構成変更で再起動が必要、OSの障害

## 冗長性による可用性

ネットワーク可用性を高めるための最初の手順では、ゲスト・ドメインと物理ネットワーク間に冗長パスを確立します。複数のI/Oドメインによって物理ハードウェアへの冗長パスを提供することで、単一のI/Oドメインの可用性が一時的に妨げられた場合でもネットワーク・トラフィックのフローが停止しないようにします。

PCIバスから上流スイッチ・ポートへの複数の物理バスを確立することで、単にネットワーク・ケーブルを取り外したなどの、起こりがちな障害から保護できます。

Oracle VM Server for SPARCによって複数の物理接続をゲスト・ドメイン内の複数の仮想デバイスにどのようにマップできるかは、サーバー・アーキテクチャによって異なります。ネットワーク接続のために、装備しているPCIバスまたはNIUと同数のI/Oドメインを使用してサーバーを構成できます。複数の仮想ネットワーク・デバイスを単一のI/Oドメインによって提供することもできます。そのため、サーバーが装備しているPCIバスが1つなのか2つなのかに関係なく、常に、ある程度の冗長性を提供する2つの仮想ネットワーク・デバイスを使用してゲスト・ドメインを構成できます。

推奨される構成は、2つの異なるPCIバス、または1つのPCIバスと1つのNIUを使用し、2つの異なるI/Oドメインを介し、ゲストOSの2つの異なる仮想ネットワーク・デバイスに対して、2つの物理ネットワーク接続をサーバーに接続したものです。NIUを使用するのが現実的でない1つのPCIバスを装備したサーバーでも（以下を参照）、単一のI/Oドメインを介して2つのネットワーク・インタフェースを接続できますが、この方法では、その単一のI/Oドメインがシングル・ポイント障害となる可能性を排除することはできません。

### サーバーI/Oアーキテクチャ概要

表2は、このホワイト・ペーパーの作成時点での、CoolThreadsテクノロジーを採用したすべてのオラクル・サーバーのI/Oアーキテクチャの概要を示しています。一連のネットワークとI/Oドメインの推奨構成を以下に示しています。

表2. ORACLE VM SERVER FOR SPARCを使用したI/Oドメイン構成に影響するサーバー構成

サーバー	PCIバスの数	NIU	各PCIバスに接続されているインタフェースの数			
			PCIバス	ディスク・コントローラ	組み込みイーサネット・ポート	拡張スロット
オラクルのSun SPARC Enterprise T5240サーバー	2	0	A	1	0	3
			B	0	4	3
オラクルのSun SPARC Enterprise T5140サーバー	2	0	A	1	0	1
			B	0	4	2
オラクルのSun SPARC Enterprise T5220サーバー	1	1	A	1	4	6
			NIU	0	0	10Gbpsイーサネット ×2
オラクルのSun SPARC Enterprise T5120サーバー	1	1	A	1	4	3
			NIU	0	0	10Gbpsイーサネット ×2
オラクルのSun SPARC Enterprise T2000サーバー	2	0	A	1	2	1
			B	0	2	4
オラクルのSun SPARC Enterprise T1000サーバー	2	0	A	0	0	1
			B	1	4	0

- オラクルのサーバーのうち、Sun SPARC Enterprise T5120サーバーおよびT5220サーバーの2つのサーバーは、1つのPCIバスと1つのNIUを装備しています。1つまたは2つのPCI Expressカード・スロットを1つまたは2つの10ギガビット・イーサネット・インタフェースに置き換えるよう、NIUを構成できます。
- NIUと10ギガビット・イーサネットを使用する場合、2つのI/Oドメインを使用して、一方のI/OドメインをPCIバス上の1つ以上のギガビット・イーサネットNICに接続して、もう一方のI/OドメインをNIU上の1つ以上の10ギガビットNICに接続できます。
- 組み込みのインタフェースまたはPCI Expressベースのインタフェースを使用してギガビット・イーサネットを使用する場合、単一のI/Oドメインを介してこれらのインタフェースを任意に組み合わせ使用できます。
- 2つのPCIバスを装備したサーバーでは、組み込みのネットワーク・インタフェースとPCI Express拡張スロットの追加のNICを任意に組み合わせ使用して、2つのPCIバスのそれぞれに少なくとも1つのバスを確立します。ゲスト・ドメインに2つの仮想ネットワーク・デバイスを提供するには、それぞれが各PCIバスに接続された2つのI/Oドメインを構成します。拡張スロットを追加のNICに使用できない場合、2つ以上の組み込みのインタフェースに接続された単一のI/Oドメインを使用します。
- Sun SPARC Enterprise® T2000サーバーは、2つのPCIバスと、それぞれが各バスに接続された2つの組み込みのネットワーク・インタフェースを装備しています。このサーバーは、オラクルの現在の製品ラインで、拡張スロットを使用せずに2つのI/Oドメインと2つの物理的に独立したネットワークをサポートできる唯一のサーバーです。
- Sun SPARC Enterprise T1000サーバーは、2つのPCIバスを装備していますが、拡張スロットは1つです。このスロットが未使用の場合は、1つ目のネットワーク接続にPCI Express NICを使用して、2つ目のネットワーク接続に組み込みのインタフェースを使用します。

## Oracle SolarisのIPマルチパス

前のセクションでは、ゲスト・ドメインへの冗長パスを最適に確立するように物理リンクとI/Oドメインを構成する方法について説明しました。このセクションでは、これらのリンクを使用して可用性を高める方法について説明します。

IPマルチパスの詳細については、  
『System Administration Guide: IP Services』  
(Part Number 816-4554) (docs.sun.com) を  
参照してください。

Oracle Solaris OSのIPマルチパスでは、2つ以上の物理インタフェースをIPMPグループに構成します。IPMPグループを構成している場合、IPMPデーモン (in.mpathd) が各リンクの状態を監視し、そのグループのプライマリIPアドレスをサポートしているインタフェースに障害が発生した場合は、そのプライマリIPアドレスをフェイルオーバーします。したがって、IPMPグループに構成されているすべてのインタフェースが、1つ以上のIPアドレスを外部で継続的に利用できるようにするために役立っています。

IPMPはさまざまな速度のリンクに対応できるため、組み込みのギガビット・イーサネット・インタフェースやPCI Expressベースのギガビット・イーサネット・インタフェースとともにNIUで10ギガビット・イーサネットを使用するよう、IPMPグループを構成できます。

IPMPでは、いずれかでネットワーク接続が切断された場合にフェイルオーバーされる各インタフェースで、'パブリック'アドレスを使用します。代表的な使用方法として、インバウンド・トラフィックがいずれかのパブリックIPアドレスをサポートしているインタフェースを介して到達する、アクティブ-アクティブ構成があります。

IPMPグループのスループットを高めるために、アウトバウンド・トラフィックは利用可能な複数のインタフェースに分散されます。この方法は、小さな受信リクエストによってより大きな応答が生じるWebサーバーなどのサービスに適しています。受信トラフィックをIPMPグループの各パブリック・アドレスにロード・バランスするという方法もあります。通常、2つのインタフェースをIPMPに構成しますが、構成するインタフェースの数が多ければほど可用性とスループットが高くなります。

IPMPデーモンでは、IPMPグループのインタフェースの状態を2つの方法で監視します。リンクの状態を監視する方法と、定期的なICMPプローブを各インタフェースからデフォルトのルーター・アドレスに送信する方法です。論理ドメイン内から構成している場合、仮想ネットワーク・デバイスは常に仮想スイッチに接続されるため、リンクベースの障害検出は機能しません。代わりに、IPMPデーモンはプローブベースの検出を使用して障害を検知します。

## Oracle VM Server for SPARCのIPマルチパス

IPMPを論理ドメインで構成するには、3つの方法があります。推奨されるアプローチは、1つまたは2つのI/Oドメインを介してサポートされている仮想デバイスを使用して、ゲストOSでIPMPを構成する方法です。

### 2つのI/Oドメインを使用したIPMP

2つのI/Oドメインは、2つのPCIバスまたは1つのPCIバスと1つのNIUを装備したサーバーで使用できます。2つのI/Oドメインを使用するには、組込みのインタフェースおよび拡張スロットで内蔵NIU用のNICまたはXAUIカードを使用するか、拡張スロットで2つのNICを使用します。2つのI/Oドメインを使用すると、ネットワークI/Oに影響を与えずに片方のI/Oドメインを再起動できるため、2つのPCIバスを装備したサーバーではこのソリューションを推奨しています。

図2にこの構成を示します。2つの各I/Oドメインが1つのPCIバスを所有し、物理デバイスに仮想スイッチを接続させています。各仮想スイッチが、ゲスト・ドメインへの仮想ネットワーク・インタフェースを提供しています。



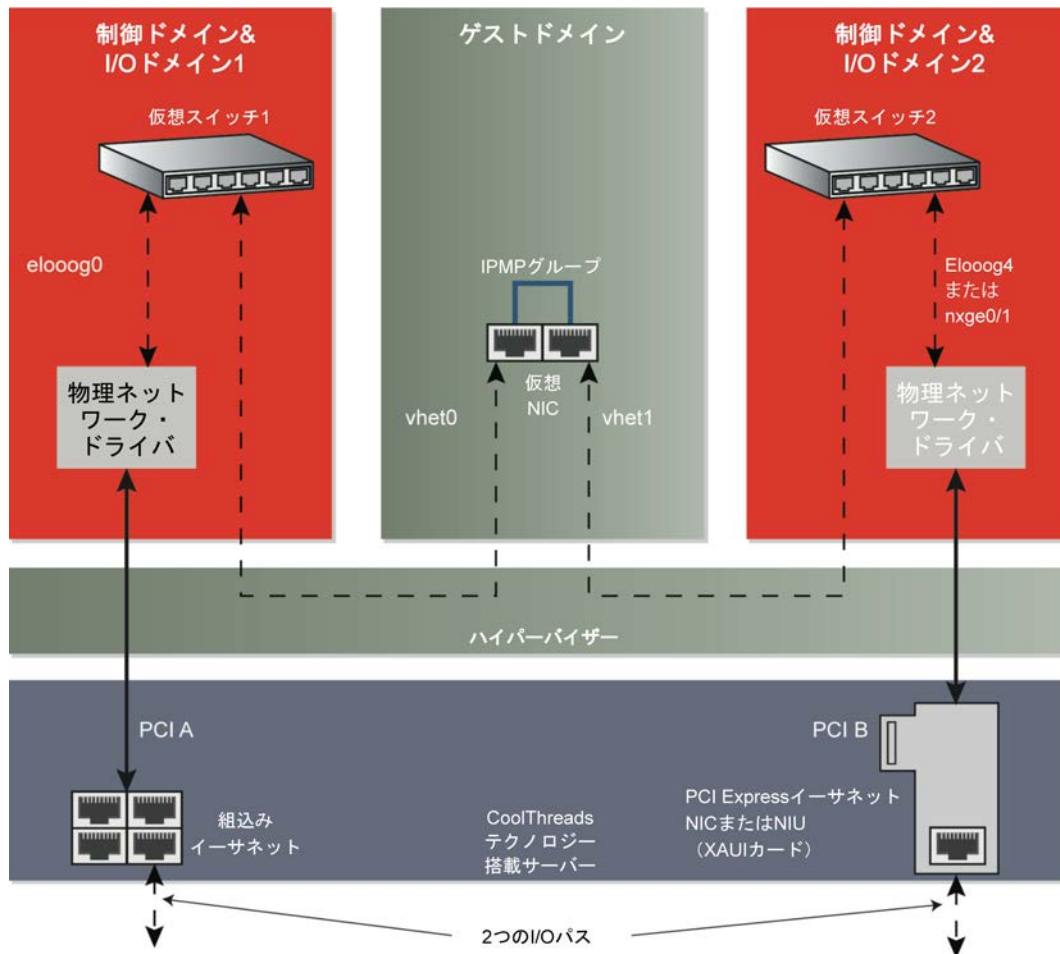


図2. 2つのI/Oドメインでサポートされている仮想ネットワーク・デバイスを使用して、ゲスト・ドメイン内にIPMPを構成可能

IPMPグループはOracle Solaris OSインスタンス内から構成しており、プローブベースの障害検出を使用しています。ゲストOSでは、2つの仮想ネットワーク・デバイスは前述の例のように認識されます。ゲストOSでは、この構成と単一I/Oドメインの構成は区別されません。

このソリューションでは、サーバーとハイパーバイザー以外はシングル・ポイント障害がないため、2つのPCIバスを装備したサーバーで高可用性を実現する場合に推奨されるアプローチとなっています。

### 単一のI/Oドメインを使用したIPMP

単一のI/Oドメインを使用して、1つまたは2つのPCIバスを装備したサーバーで2つの物理インタフェースに接続できます。このソリューションは、1つのPCIバスを装備したサーバーで推奨されます。また、2つのI/Oドメインを使用するソリューションではより高い可用性が提供されるため、可用性よりもスループットを高めるためにIPMPを実装している場合にもこのソリューションを使用できます。

図3にこの構成を示します。各物理デバイスは、I/Oドメイン内の独自の仮想スイッチに接続されています。各仮想スイッチは、ゲスト・ドメイン内の仮想ネットワーク・デバイスに接続されています。IPMPグループはゲストのOracle Solaris OSインスタンス内から構成されています。

障害が発生した場合は、プローブベースの障害検出を使用して仮想ネットワーク・デバイス間でフェイルオーバーが開始されます。

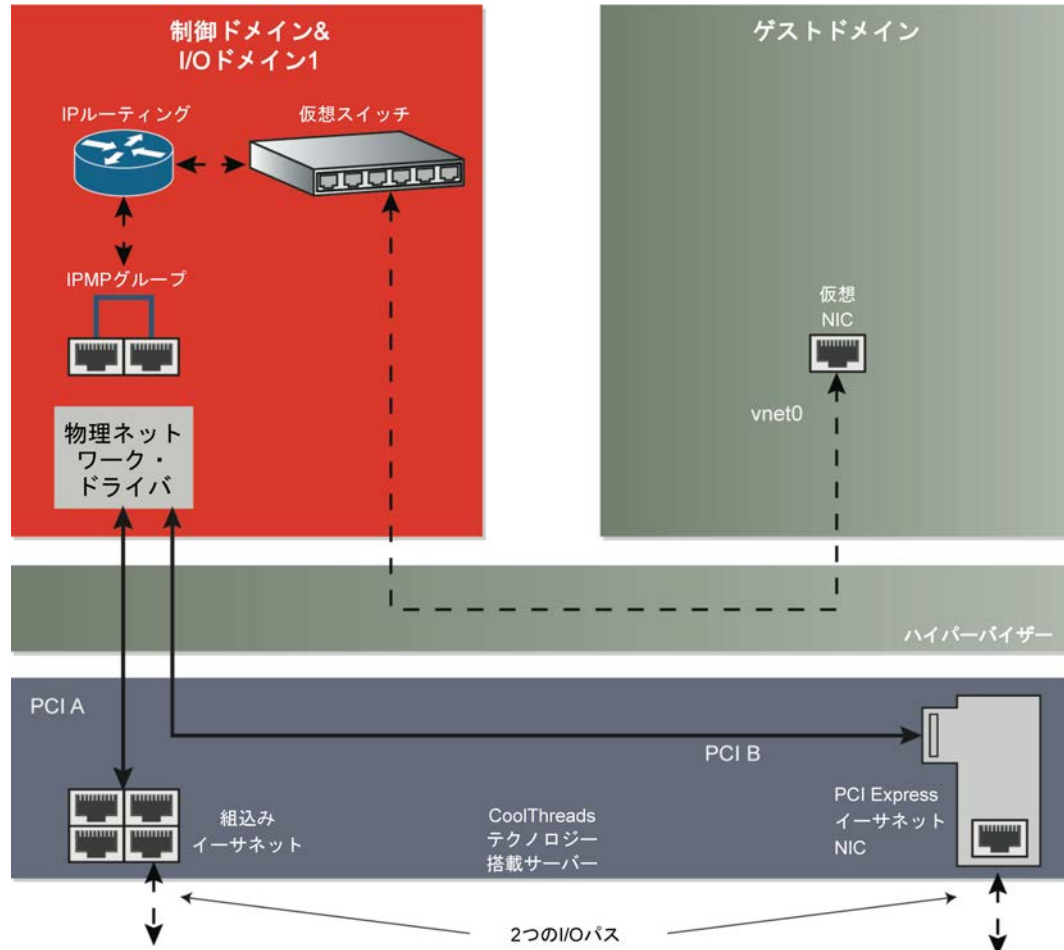


図3. 単一のI/Oドメインでサポートされている仮想ネットワーク・デバイスを使用して、ゲスト・ドメイン内にIPMPを構成可能

このソリューションで、単純なシングル・ポイント障害、つまりNIC、ケーブル、および上流スイッチ・ポートを排除できます。I/Oドメインはシングル・ポイント障害であり、I/Oドメインを再起動するとネットワーク接続が一時的に切断されます。

### I/Oドメイン内のIPMP

IPMPを完全にI/Oドメイン内に実装することで、ゲスト・ドメインでIPMPの実装が認識されないようにできます。このソリューションにより、ゲスト・ドメインの構成を簡素化できます。これは、1つまたは2つのPCIバスを装備したサーバーで使用できます。このソリューションには、単一のI/Oドメインがゲストのシングル・ポイント障害となること、およびI/Oドメインが仮想スイッチとIPMPグループ間でパケットをルーティングするため、各パケットの転送にTCP/IPスタックのオーバーヘッドが追加されるという、2つのデメリットがあります。そのため、この構成は推奨されません。

この構成は、2つのPCIバスを装備したサーバー向けのものです。ゲスト・ドメイン内の単一の仮想ネットワーク・デバイスは、I/Oドメイン内の仮想スイッチに接続されます。I/Oドメインはルーターとして構成され、パケットは仮想スイッチからIPMPグループにルーティングされます。図4に、1つの組み込みイーサネット・インタフェースおよび別のPCI Express NIC上の1つのインタフェースを使用したIPMP構成を示します。

このソリューションは、NIC、ネットワーク・ケーブル、および上流スイッチ・ポートというシングル・ポイント障害を排除できますが、I/Oドメインはシングル・ポイント障害として残ります。

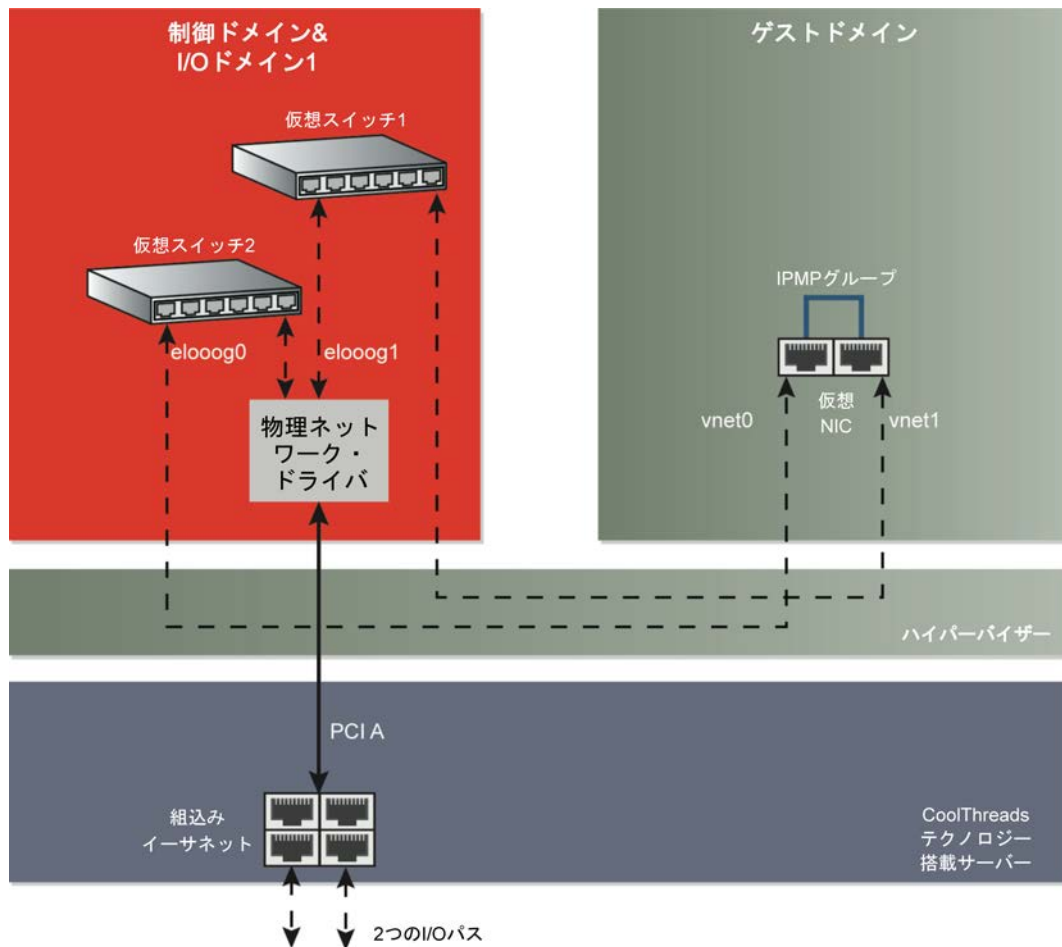


図4. IPMPをI/Oドメイン内に構成可能。ただし、このアプローチでは、仮想ネットワークにルーティングのオーバーヘッドが追加され、単一のI/Oドメインがシングル・ポイント障害として残る

#### ソリューションのまとめ

表3に示すように、CoolThreadsテクノロジーを採用したオラクル・サーバーで3つのIPMP構成をサポートできます。この表で、単一のI/Oドメインを使用するソリューションは、単一のI/Oドメインがシングル・ポイント障害として残りますが、すべてのサーバーで実行できることを確認できます。また、2つのI/Oドメインを使用するソリューションでは、2つのPCIバス、または1つのPCIバスと1つのNIUが必要であることも確認できます。

同じサーバーで複数のソリューションを使用できます。たとえば、単一のI/Oドメインを介してIPMPで特定のゲスト・ドメインをサポートしながら、2つのI/Oドメインを介してIPMPで別のゲスト・ドメインをサポートできます。

表3. サーバー構成および各構成での3つのIPMPソリューションのサポート

サーバー	適用可能なソリューション			
	PCIバスの数	単一のI/Oドメイン	2つのI/Oドメイン	I/Oドメイン内
オラクルのSun SPARC Enterprise T5240サーバー	2	4	4	4
オラクルのSun SPARC Enterprise T5140サーバー	2	4	4	4
オラクルのSun SPARC Enterprise T5220サーバー	1	4	4 (NIUを使用)	4
オラクルのSun SPARC Enterprise T5120サーバー	1	4	4 (NIUを使用)	4
オラクルのSun SPARC Enterprise T2000サーバー	2	4	4	4
オラクルのSun SPARC Enterprise T1000サーバー	2	4	4	4

## ネットワーク可用性の構成

このセクションでは、2つのI/Oドメインまたは単一のI/Oドメインを使用してIPMPを構成するための共通の背景情報について説明します。

このシリーズ第一弾のテクニカル・ホワイト・ペーパー『Oracle VM Server for SPARCを使用してデータの信頼性を高めるためのベスト・プラクティス』では、論理ドメインを理解し、Oracle Solaris OSをインストールし、サーバー・ファームウェアにパッチを適用して、Oracle VM Server for SPARC Manager (SUNWlcm) パッケージをインストールし、1つ目の制御とI/Oのドメインを作成するための基盤となる情報とリソースを提供しています。以降のセクションでは、読者がこの情報を十分に理解しており、作業するI/Oドメインとゲスト・ドメインがあることを前提としています。

## Oracle VM Server for SPARCでIPMPを使用するための前提条件

以降の2つのセクションで説明している2つのソリューションを構成するには、いくつか前提条件があります。

### デフォルト・ルートは外部

1台のサーバーで論理ドメインを相互接続するために排他的に使用している仮想スイッチで、IPMPを構成することはできますが、ほとんどの場合、そのサーバーの外部のネットワークに接続されているネットワーク・インタフェースを使用して、IPMPを構成することになります。

説明している2つのソリューションの1つであるプローブベースのIPMPでは、定期的なICMPパケットはドメインのデフォルトのルーター・アドレスに送信されます。Oracle Solaris OSの大部分のインストールでは、サーバーの外部のデフォルトのルーターが定義されます。これを確認するには、論理ドメイン内から`cat/etc/defaultrouter`を実行して、アドレスがサーバーの外部であることを確認します。

### vnetデバイスごとに別のMACアドレス

IPMPに使用する各仮想ネットワーク・デバイスに、一意のMACアドレスを指定する必要があります。デフォルトでは、MACアドレスが作成されると、Oracle VM Server for SPARC Managerによってvnetデバイスごとに一意のMACアドレスが割り当てられます。これは適切な動作です。

変数`local_mac_address`を`false`に設定すると、OpenBootソフトウェアによって仮想ネットワークのMACアドレス設定をオーバーライドできます。この場合、各vnetデバイスでは、システムのMACアドレスとIPMPは機能しないと見なされます。デフォルト設定は`local_mac_address?=true`です。`printenv`コマンドを使用して、ゲスト・ドメインの`ok`プロンプトでこれを確認できます。

```
{0} ok printenv
```

### DNSとHostsファイルにおけるアドレスの構成

定義する各IPMPグループに少なくとも4つのIPアドレスを割り当てる必要があります。2つのアドレスがIPMPグループに関連付けられます。このアドレスを使用して、論理ドメインによって提供されるサービスにアクセスできます。別の2つのアドレスは、プローブベースのフェイルオーバーで使用されるテスト・アドレスです。このアドレスを割り当てる必要がありますが、このアドレスは、リンクに障害が発生した場合は使用できなくなるため、サービスにアクセスするためには使用しないでください。

表4に、このホワイト・ペーパーの後半で使用しているアドレスのネーミング規則を示します。これらのアドレスは、ゲストの`/etc/hosts`ファイルにネーム・サービス（DNSまたはNIS）で定義する必要があります。

表4. 各IPMPグループに4つのIPアドレスを定義する必要がある

アドレス	インタフェース	目的
guest1	vnet0	サービスが提供されるアドレス。このアドレスをクライアントで使用する必要がある
guest1-0	vnet0	テスト・アドレス。クライアントでは使用できない
guest1-sec	vnet1	2つ目のインタフェースで構成されるアドレス
guest1-1	vnet1	2つ目のインタフェースのテスト・アドレス。クライアントでは使用できない

上記のネーミング規則に従って、ゲスト・ドメインの一般的な/etc/hostsファイルには次のようなエントリが含まれます。

```
#
# Internet host table
#
127.0.0.1    localhost
::1        localhost
192.168.0.31  guest1 loghost
192.168.0.32  guest1-sec
192.168.0.33  guest1-0
192.168.0.34  guest1-1
```

#### mpathdタイムアウトの調整

デフォルトの場合、IPMPデーモンではフェイルオーバーに10秒間のタイムアウトが使用されるため、フェイルオーバーが実行されるまで最大10秒間、問題が続く可能性があります。このデフォルト値を短くすることもできます。ただし、このデフォルト値を1秒未満に短くした場合、ネットワーク・ロードや一時的な状態のために、かなり多くの誤ったフェイルオーバーが発生する可能性があります。タイムアウト値は、/etc/default/mpathdを使用してミリ秒で指定します。

```
#
# Time taken by mpathd to detect a NIC failure in ms. The minimum time
# that can be specified is 100 ms.
#
FAILURE_DETECTION_TIME=10000
```

この値を変更する場合は、mpathdデーモンを再開します。

```
fraser# pkill -HUP in.mpathd
```

## 2つのI/Oドメインを使用したIPMPの構成

ゲスト・ドメインにIPMPを構成している場合、ゲスト・ドメインで認識されるのは、2つの仮想ネットワーク・デバイスがIPMPと連携して動作しているということだけです。その2つのデバイスが1つのI/Oドメインを介して提供されるのか2つのI/Oドメインを介して提供されるのかは、全く認識されません。そのため、ゲスト・ドメインでの2つのI/Oドメインを使用したIPMPの構成は、1つのI/Oドメインを使用したIPMPの構成とまったく同じになります。実際に、異なる仮想スイッチに接続されているvnetデバイスを切断して再接続することで、1つのI/Oドメインと2つのI/Oドメインを切り替えることもできます。

この2つのソリューションの違いは、ゲスト・ドメインに仮想ネットワーク・デバイスを提供するよう、2つ目のI/Oドメインを設定する点だけです。概念的には、2つ目のI/Oドメインはゲスト・ドメインにI/Oサービスを提供し、そのI/O構成はOracle VM Server for SPARC Managerと管理コマンドを使用してプライマリ・ドメインで管理されます。

### 2つ目のI/Oドメインの構成

2つ目のI/Oドメインを設定するには、新しい論理ドメインを作成し、その論理ドメインがPCI Express拡張スロットに構成されている1つ以上のNICにアクセスできるように、サーバーの未使用のPCIバスを割り当てる必要があります。その際、非常に重要な手順として、2つ目のI/Oドメインに内蔵ディスク・コントローラをサポートしていないPCIバスを割り当てる必要があります。内蔵ディスク・コントローラをサポートしているPCIバスを割り当てると、プライマリI/Oドメインやその他の論理ドメインがそこから起動するため、サーバーを内蔵ディスクから起動できなくなります。

### 新しい論理ドメインの作成

I/Oドメインは、I/Oリソースを所有して他のゲスト・ドメインにI/Oサービスを提供する論理ドメインです。既存の論理ドメインをI/Oドメインとして使用できます。新しい論理ドメインを作成することもできます。この例では、Oracle Solaris JumpStartを使用してOracle Solaris OSをインストールする新しいゲスト・ドメインを作成します。別のゲスト・ドメインのブート・ディスクのコピーを作成して、新しいI/Oドメインで使用することもできます。

この例では、I/Oドメインにsecondaryという名前を指定しています。また、ネットワーク・インストール・プロセスに対して使用できる仮想ネットワーク・インタフェース（プライマリI/Oドメインによってサポート）を使用して、このI/Oドメインを構成しています。

```
fraser# mkfile 10G /domains/secondary.img
fraser# ldm create secondary
fraser# ldm set-vcpu 4 secondary
fraser# ldm set-mem 4g secondary
fraser# ldm add-mau 1 secondary
fraser# ldm add-vnet vnet0 primary-vsw0 secondary
fraser# ldm add-vdsdev /domains/secondary.img vol8@primary-vds0
fraser# ldm add-vdisk vdisk0 vol8@primary-vds0 secondary
fraser# ldm bind secondary
```

サーバーの電源を入れ直しても保持されるよう構成をサービス・プロセッサに保存します。この例では、構成をfourguestsの下に保存しています。

```
fraser# ldm remove-spconfig fourguests
fraser# ldm add-spconfig fourguests
```

これで、シリーズ第一弾のホワイト・ペーパー『Oracle VM Server for SPARCを使用してデータの信頼性を高めるためのベスト・プラクティス』の説明に従って、ゲスト・ドメインのオペレーティング・システムをネットワーク経由でインストールできるようになりました。

Oracle Solaris 10 OSのみを、ゲスト・ドメインにインストールする必要があります。Oracle VM Server for SPARC Managerを2つ目のI/Oドメインにインストールしないでください。

### I/Oリソースの割当て

新しいゲスト・ドメインにI/Oリソースを割り当てて、ゲスト・ドメインをI/Oドメインにします。これを行うため、プライマリI/OドメインからいずれかのPCIバスを削除し、そのPCIバスを2つ目のI/Oドメインに割り当てます。

注 - サーバーが起動しなくなるような問題を発生させないため、どのバスがどのデバイスをサポートしているのかを把握している場合でも、必ずこの手順を実行してください。サーバーの製造日によりますが、一部のサーバーでは、内蔵ディスク・コントローラは異なるPCIバスに割り当てられています。

### PCIバス構成の指定

プライマリI/Oドメインのバインディングを表示して、サーバーに2つのPCIバスがあることと、それらのPCIバスの名前を確認します。

```
fraser# ldm list-bindings primary
...
IO
  DEVICE          PSEUDONYM      OPTIONS
  pci@780         bus_a
  pci@7c0         bus_b
```

サーバーがSun SPARC Enterprise T5120サーバーまたはT5220サーバーである場合、出力結果からどちらがPCIバスでどちらがNIUなのかが分かります。2つ目のI/OドメインにNIUを割り当てるため、サーバーを内蔵ディスクから起動できなくなることを心配する必要はありません。実行する手順では、2つ目のI/Oドメインに割り当てるPCIバスとして、NIU (niu@80など) の名前を使用しています。

### ディスク・コントローラの確認

プライマリI/Oドメインのルート・ディスクを提供するディスク・コントローラを構成しているバスを確認します。まず、ブート・ディスクに該当するディスク・デバイスを確認します。この場合、/dev/dsk/c1t0d0s0です。

```
fraser# df /
/                (/dev/dsk/c1t0d0s0 ):47729512 blocks   6297665 files
```

次に、このディスク・デバイスが接続されているPCIバスを確認します。

```
fraser# ls -l /dev/dsk/c1t0d0s0
lrwxrwxrwx  1 root    root          65 Mar 11 15:33 /dev/dsk/c1t0d0s0 ->
../../../../devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@0,0:a
```



文字列**pci@7c0**は、ディスク・コントローラがbus\_b上にあることを示しています。この結果は、この手順の実行が重要であることを示すための例として役立ちます。このコマンドをテストしたサーバーは、ディスク・コントローラがPCI-Xスロットをbus\_bで使用している、初期に製造されたSun SPARC Enterprise T2000サーバーです。同じサーバーの以降のモデルでは、ディスク・コントローラはbus\_aで構成されています。この手順で、新しいI/OドメインおよびそのI/Oドメインに接続されているネットワーク・デバイスにbus\_aを割り当てることができることが分かります。

### ネットワーク・デバイスの確認

ネットワーク・デバイスが構成されているバスを確認します。この例で使用しているSun SPARC Enterprise T2000サーバーでは、bus\_a上に2つの組込みのインタフェースがあり、bus\_b上にも2つの組込みのインタフェースがあります。他のすべてのサーバーでは、内蔵NICがないPCI ExpressバスでNICを提供するように、サーバーのいずれかの拡張スロットにPCI Express NICをインストールする必要があります。どのPCI ExpressスロットがサーバーのどのPCIバスに接続されているのかを示す、情報の中央ソースはありません。多くの場合、サーバーのホワイト・ペーパーに記載されているアーキテクチャ・ブロック図を確認するとこれを推測できますが、この手順でデバイスの場所を確認する以外に方法はありません。

この例では、インタフェースe1000g0およびe1000g1がbus\_aに接続されており、これらのインタフェースを2つ目のI/Oドメインで使用できることを示しています。

```
fraser# ls -l /dev/e1000g0
lrwxrwxrwx  1 root    root          48 Mar 11 15:32 /dev/e1000g0 ->
../devices/pci@780/pci@0/pci@1/network@0:e1000g0
fraser# ls -l /dev/e1000g1
lrwxrwxrwx  1 root    root          50 Mar 11 15:32 /dev/e1000g1 ->
../devices/pci@780/pci@0/pci@1/network@0,1:e1000g1
```

### PCIバスの割当て

ここでは、プライマリI/Oドメインのbus\_bを割り当てることはできません。bus\_aを割り当てる必要があります。ldm remove-ioバリエーションを使用して、プライマリI/Oドメインからbus\_aを削除します。サーバーに正しいPCIバスを選択していることを確認します。

```
fraser# ldm remove-io pci@780 primary
```

プライマリI/Oドメインを再起動して、セカンダリI/Oドメインにbus\_aを割り当てます。ドメインが実行されている場合、まずそのドメインを停止して、新しいデバイスを追加できるようバインドを解除する必要があります。その後、このドメインを再バインドし、新しい機能とともに再起動できます。

```
fraser# ldm add-io pci@780 secondary
```

新しいデバイスを検出できるように、-rオプションを使用してセカンダリI/Oドメインを起動します。

```
{0} ok boot -r
```

サービス・プロセッサに新しい論理ドメイン構成を保存します。新しい名前splitを使用します。

```
ldm add-spconfig split
```

### 新しい仮想スイッチの作成

セカンダリI/Oドメインが1つのPCIバスと2つのネットワーク・デバイスを所有しており、ネットワーク・デバイスをゲスト・ドメインに接続させる仮想スイッチを設定できるようになりました。次のコマンドを実行して、セカンダリI/Oドメインに仮想スイッチsecondary-vsw1を作成します。これらのコマンドは、プライマリI/Oドメインから実行します。

```
fraser# ldm add-vsw net-dev=e1000g0 secondary-vsw1 secondary
```

セカンダリI/Oドメインを再起動して、変更を適用します。I/Oドメイン内のe1000g0デバイスのネットワーク・インタフェースの詳細を確認する必要はありません。スイッチ上でポートとして機能しているだけです。

### ゲスト・ドメインへの仮想ネットワーク・デバイスの接続

ゲスト・ドメインにvnetデバイスを追加します。次のコマンドを実行して、ドメインguest1にvnet1を接続します。ゲストを再起動すると、ゲストで新しいデバイスが認識されて、ゲスト・ドメイン内からIPMPを構成できるようになります。

```
fraser# ldm stop guest1
LDom guest1 stopped fraser# ldm unbind guest1
fraser# ldm add-vnet vnet1 secondary-vsw1 guest1
Initiating delayed reconfigure operation on LDom guest1. All configuration
changes for other LDom are disabled until the LDom reboots, at which time
the new configuration for LDom guest1 will also take effect.
fraser# ldm bind guest1
fraser# ldm start guest1
LDom guest1 started
```

### ゲスト・ドメインでのIPMPの構成

2つのvnetデバイスを用いて構成したゲスト・ドメインで、ドメインを起動してこの2つのデバイスを使用するIPMPグループを構成できます。

IPMPを構成するには、仮想ネットワーク・インタフェースを構成するifconfigコマンドに追加のパラメータを指定する必要があります。適切なパラメータを指定してifconfigを入力するか、/etc/hostname.deviceファイルを変更してゲスト・ドメインを再起動できます。再起動しても構成が保持されることをテストする必要があります。そのため、ホスト名ファイルを変更して再起動し、構成が適切であり一貫性があることを確認することをお勧めします。

"DNSとHostsファイルにおけるアドレスの構成"で説明しているネーミング規則を思い出してください。2つのパブリック・アドレスはguest1とguest1-secで、テスト・アドレスはguest1-0とguest1-1となつ

ています。この例ではIPMPグループにblueという名前を任意に指定しています。/etc/hostsおよびネーム・サービスでこれらのエントリを指定したら、次のコマンドを入力します。

```
guest1# echo >/etc/hostname.vnet0 "guest1 netmask + broadcast + group blue up
addif guest1-0 netmask + broadcast + -failover deprecated up"
guest1# echo >/etc/hostname.vnet1 "guest1-sec netmask + broadcast + group blue up
addif guest1-1 netmask + broadcast + -failover deprecated up"
```

再起動せずにIPMPグループを設定する場合は、次に示すように、/etc/hostnameファイルのコンテンツをifconfigの引数として使用できます。

```
guest1# ifconfig `cat /etc/hostname.vnet0`
guest1# ifconfig `cat /etc/hostname.vnet1`
```

### ゲスト・ドメインでのIPMPのテスト

guestドメインを再起動し、ifconfigコマンドを使用してIPMPグループが動作していることを確認します。この例では、2つのパブリック・アドレスと2つのテスト・アドレスが動作していることを示しています。

```
guest1# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index
1
    inet 127.0.0.1 netmask ff000000
vnet0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.168.0.31 netmask ffffffff broadcast 192.168.0.255
    groupname blue
    ether 0:14:4f:fb:58:90
vnet0:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 2
    inet 192.168.0.33 netmask ffffffff broadcast 192.168.0.255
vnet1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.0.32 netmask ffffffff broadcast 192.168.0.255 groupname blue
    ether 0:14:4f:f9:e:3b
vnet1:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 3
    inet 192.168.0.34 netmask ffffffff broadcast 192.168.0.255
```

別のシステムからコマンドping -s guest1を使用して、IPMPグループによって提供されている接続を監視します。次に、ネットワーク・ケーブルを引き抜いたりスイッチ・ポートを無効にしたりして、いくつか障害をシミュレートします。

vnet0をサポートしているデバイスに取り付けられたネットワーク・ケーブルを引き抜くと、次のコンソール・メッセージが表示されます。

```
Jun 26 16:50:26 guest1 in.mpathd[158]: NIC failure detected on vnet0 of group
blue
Jun 26 16:50:26 guest1 in.mpathd[158]: Successfully failed over from NIC vnet0 to
NIC vnet1
```

ゲストからインタフェース状態を確認すると、デバイスvnet0とvnet0:1にFAILEDのフラグが付いており、2つのパブリックIPアドレスがvnet1:2に関連付けられていることが分かります。

```

guest1# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index
1
    inet 127.0.0.1 netmask ff000000vnet0:flags=19000842<BROADCAST,RUNNING,
MULTICAST,IPv4,NOFAILOVER,FAILED> mtu 0 index 2
    inet 0.0.0.0 netmask 0
    groupname blue
    ether 0:14:4f:fb:58:90
vnet0:1: flags=19040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,
NOFAILOVER,FAILED> mtu 1500 index 2
    inet 192.168.0.33 netmask ffffffff broadcast 192.168.0.255
vnet1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.0.32 netmask ffffffff broadcast 192.168.0.255
    groupname blue
    ether 0:14:4f:f9:e:3b
vnet1:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 3
    inet 192.168.0.34 netmask ffffffff broadcast 192.168.0.255
vnet1:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.0.31 netmask ffffffff broadcast 192.168.0.255

```

外部接続をリストアすると、IPMPにより修復されたインタフェースのフェイルバックが行われたことを示すコンソール・メッセージが表示されます。

```

Jun 26 16:59:03 guest1 in.mpathd[158]: NIC repair detected on vnet0 of group blue
Jun 26 16:59:03 guest1 in.mpathd[158]: Successfully failed back to NIC vnet0

```

同様に、vnet1インタフェースに接続された物理リンクの障害と修復をシミュレートできます。

## 単一のI/Oドメインを使用したIPMPの構成

"ネットワーク可用性の構成"セクションで説明している前提条件を確認したら、単一のI/Oドメインを使用してIPMPを設定するため、2つ目の仮想スイッチを作成し、それらの仮想スイッチをネットワーク・デバイスに割り当てて、ゲスト・ドメインにIPMPを設定します。

### I/Oドメインの構成

最初の手順として、2つのネットワーク・デバイスを使用してゲスト・ドメインを構成します。この手順では、すでに、プライマリI/Oドメインのネットワーク・デバイスの1つを構成し、仮想スイッチを作成して、ゲスト・ドメインに仮想ネットワーク・デバイスを接続していることを前提としています。この例では、すでに、仮想スイッチprimary-vsw0にデバイスe1000g2を接続し、guest1という名前のゲスト・ドメインに接続されたスイッチでvnetデバイスを接続しています。

## 2つ目のネットワーク・デバイスの構成

I/Oドメインで2つ目のネットワーク・インタフェースを構成していない場合は、これを構成します。/etc/hostsにホスト名エントリを追加し、/etc/hostname.interfaceファイルに新しいホスト名を追加します。ネットワーク・デバイスe1000g3に、この例ではホスト名switch2を使用しています。

次の手順で作成する仮想スイッチでデバイスに直接アクセスできるため、この手順は必ずしも必要ではありません。2つ目のネットワーク・デバイスを構成すると、これを使用してゲスト・ドメインにサービスを提供する前に、このデバイスが動作しているかどうかテストできます。

```
fraser# echo switch2 >/etc/hostname.e1000g3
```

I/Oドメインを再起動して変更を適用します。ifconfigコマンドを手動で実行してインタフェースを構成できますが、変更を行ったら、ドメインを再起動して構成変更をテストします。エラーがあった場合は、今後ドメインを再起動する際ではなく、すぐにエラーを確認します。

## 新しい仮想スイッチの作成

I/Oドメインでprimary-vsw1という名前の新しい仮想スイッチを作成します。ここでは、primaryという名前のI/Oドメインでネットワーク・デバイスe1000g3を使用していると想定しています。

```
fraser# ldm add-vsw net-dev=e1000g3 primary-vsw1 primary
```

I/Oドメインが2つの仮想スイッチを使用して構成されていることを確認できます。

```
fraser# ldm list-services primary
...
VSW
  NAME          MAC          NET-DEV  DEVICE  MODE
  primary-vsw0  00:14:4f:fb:ae:95  e1000g2  switch@0  prog,promisc
  primary-vsw1  00:14:4f:f8:6a:40  e1000g3  switch@1  prog,promisc
```

電源を入れ直しても保持されるようサービス・プロセッサに構成を保存し、保留にした変更が適用されるようプライマリI/Oドメインを再起動します。この例では、既存のアクティブな構成fourguestsを新しい構成に置き換えています。

```
fraser# ldm remove-sponconfig fourguests
fraser# ldm add-sponconfig fourguests
fraser# reboot
```

## ゲスト・ドメインへの仮想ネットワーク・デバイスの接続

再起動した後で、仮想スイッチを論理ドメインに接続します。この例では、guest1という名前のドメインを使用しています。

```

fraser# ldm stop guest1
LDom guest1 stopped
fraser# ldm unbind guest1
fraser# ldm add-vnet vnet1 primary-vsw1 guest1
Initiating delayed reconfigure operation on LDom guest1. All configuration
changes for other LDom are disabled until the LDom reboots, at which time
the new configuration for LDom guest1 will also take effect.
fraser# ldm bind guest1
fraser# ldm start guest1
LDom guest1 started

```

ゲスト・ドメインのコンソール・ポートに接続すると、okプロンプトで、2つのvnetデバイスがゲストで使用できるようになっていることが分かります。

```

{0} ok show-nets
a) /virtual-devices@100/channel-devices@200/network@1
b) /virtual-devices@100/channel-devices@200/network@0
q) NO SELECTION
Enter Selection, q to quit: q

```

## ゲスト・ドメインでのIPMPの構成とテスト

ゲスト・ドメインでIPMPを構成し、"ゲスト・ドメインでのIPMPの構成"セクションの手順を実行してテストできます。

## まとめ

CoolThreadsテクノロジーを採用したオラクル・サーバーでOracle VM Server for SPARCを使用すると、仮想環境でネットワーク可用性メカニズムを物理サーバー上と同様に実装する必要がある組織では、さまざまな可能性が広がります。ネットワーク可用性を高めるには、ネットワーク・リソースへの別々の物理パスを確立し、1つまたは2つのI/Oドメインを介してそれらの物理パスをゲスト・ドメインで利用できるようにします。

- 2つのPCIバス、または1つのPCIバスと1つのNIUを装備したサーバーでは、内蔵ネットワーク・インタフェースと拡張スロットで使用しているNICを組み合わせ、2つの冗長パスを確立できます。これらの構成では、I/Oドメインに障害が発生したり、I/Oドメインを再起動したりしても、ネットワーク可用性が中断されません。
- NIUを使用して10ギガビット・イーサネットを実現できない1つのPCIバスを装備したサーバーでは、2つの物理ネットワーク・パスを確立し、単一のI/Oドメインを介してルーティングできます。この構成はI/Oドメインの障害や再起動の影響を受けやすいですが、一般的なネットワーク障害からの保護を提供します。スイッチを不適切に構成したり、ケーブルを誤って取り外したり、ケーブルを間違っ

たスイッチ・ポートやサーバーに差し込んだりするなどの人為的ミスから保護できます。NIC、ケーブル、およびスイッチ・ポート自体の障害からも保護できます。

ネットワークへの2つ以上のパスを確立したら、ゲスト・ドメインによって認識される仮想ネットワーク・デバイスを、2つのリンクのいずれかに障害が発生した場合でも引き続きネットワーク・アクセスを提供するIPMPグループに構成できます。仮想環境にIPMPを設定しているため、ゲスト・ドメインのIPMP構成は、ネットワーク・アクセスが1つのI/Oドメインで提供されるのか2つのI/Oドメインで提供されるのかには関係しません。この簡潔なソリューションの魅力は、同じ物理サーバーで両方の構成を使用でき、ゲスト・ドメインによって認識されない状態で、1つのI/Oドメインを使用したり、2つのI/Oドメインを使用したりとゲスト・ドメインを切り替えることができる点です。

### 著者について

Peter A. Wilsonは16年間を超える業界経験があります。うち、12年間はSunとOracleで勤務しており、ハードウェア、ソフトウェア、システム、および製品マーケティングでさまざまな役割を果たしてきました。2000年に英国から米国に異動し、SunのNetraとフォルト・トレラント・サーバーのお客様向けテストを担当しました。現在、テクニカル・マーケティングのエンジニアとして、CoolThreadsテクノロジーを採用したオラクル・サーバーのすべての側面を担当しています。Peterは、英国のニューカッスル・アポン・タイン大学で、マイクロエレクトロニクスとソフトウェア・エンジニアリングのM.Eng（工学修士）の学位を取得しています。

### 謝辞

このホワイト・ペーパーの著者は、このホワイト・ペーパーで説明している問題と構成に取り組み、それらに基づいてこのホワイト・ペーパーを作成したことに対して、フリーのテクニカル・ライターでありエンジニアであるSteve Gaedeに感謝の意を表明しています。また、Oracle VM Server for SPARCのニュアンスを理解できるよう支援したことに対して、Alexandre ChartreおよびNarayan Venkatにも感謝の意を表明しています。

### 参考資料

関連するOracleのホワイト・ペーパーについては、このホワイト・ペーパーで紹介しています。



Oracle is committed to developing practices and products that help protect the environment

Oracle VM Server for SPARCを使用してネットワーク

可用性を高めるためのベスト・プラクティス

2010年5月

著者 : Peter A. Wilson

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

海外からのお問い合わせ窓口 :

電話 : +1.650.506.7000

ファクシミリ : +1.650.506.7200

oracle.com

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。UNIXはX/Open Company, Ltd.によってライセンス提供された登録商標です。

0310

**Hardware and Software, Engineered to Work Together**