ORACLE VM

Oracle VM Server for SPARCの ベスト・プラクティス

Oracleホワイト・ペーパー | 2017年1月

目次

概要

このホワイト・ペーパーでは、Oracle VM Server for SPARC（旧称Sun Logical Domains）のベスト・プラクティスについて説明します。Oracle VM Server for SPARCの仮想化テクノロジーを使用すると、1つの物理システムに複数の仮想システムを作成できます。このソリューションをOracle SPARCサーバー上に導入すると、強力かつ効率的な仮想化プラットフォームになります。オラクルのSPARC仮想化テクノロジーの概要について詳しくは、ホワイト・ペーパー『Oracle SPARC仮想化テクノロジーを使用した集約』(http://www.oracle.com/technetwork/jp/server-storage/sun-sparc-enterprise/technologies/consolidate-sparc-virtualization-2301718-ja.pdf）このドキュメントでは、SPARC物理ドメイン（PDoms）、Oracle VM Server for SPARC、Oracle Solaris Zonesをはじめとする SPARCの仮想化機能を選択する場合の評価方法について説明しています。これらは単独でも組み合わせても使用できる、補完的なテクノロジー・ソリューションです。

SPARC物理ドメイン（PDoms）、Oracle VM Server for SPARC、およびOracle Solaris Zonesは単独でも使用できますが、機能毎の独立性や冗長性を強化するために使用したり、仮想化密度の向上などのためにも使用することもできます。

このホワイト・ペーパーでは、Oracle VM Server for SPARCを使用することを前提に、リソース要件、パフォーマンス要件、および可用性要件がともに厳しい本番環境をハイエンドのSPARCサーバー上で実行する場合を中心に、その実装のベスト・プラクティスについて説明します。

このホワイト・ペーパーでは、Oracle VM Server for SPARCに関する次のトピックについて説明します。

» **概要**：VM Server for SPARCの基本的な定義、概念および導入オプション

» **要件**：ソフトウェア、ハードウェア、およびファームウェアの要件

» 最適なパフォーマンスを目的とした**ベスト・プラクティス**

» 耐障害性や可用性を目的とした**ベスト・プラクティス**

Oracle VM Server for SPARCの概要

Oracle VM Server for SPARCはOracle VM製品ファミリに含まれる製品で、Oracle SPARCサーバーに対し、非常に効率的なエンタープライズ・クラスの仮想化機能を提供します。最大4,096個のCPUスレッドを搭載できるオラクルのSPARCサーバーは、複数の物理サーバーを1つのプラットフォームに仮想化できる強力なシステムです。

Oracle VM Server for SPARCでは、システムに実装されたSPARCハイパーバイザを利用して仮想システムを作成します。各仮想システムは*論理ドメイン*または*仮想マシン*と呼ばれ、それぞれ固有のCPUリソース、メモリ・リソース、およびI/Oリソースを持つ、それぞれに独立した専用のOracle Solarisオペレーティング・システムが実行されます。

Oracle VM Server for SPARCでは、SPARC TシリーズおよびSPARC Mシリーズのサーバーが提供する大規模なスレッド・スケーラビリティを活用してパフォーマンスに優れた仮想化を実現します。SPARC Mシリーズ・サーバーは複数の物理ドメイン（"PDoms"）に分割することもでき、物理ドメインごとに独立したOracle VM Server for SPARC環境をホストできます。1つの物理サーバーまたはMシリーズの物理ドメイン上に、最大128の論理ドメインを作成できます。

Oracle VM Server for SPARCの要件

Oracle VM Server for SPARCソフトウェアは、Logical Domains Manager、SPARCサーバー・ファームウェア、およびOracle Solarisの各バージョンをベースとしています。本書の執筆時点では、3.3がLogical Domains Managerの最新バージョンで、このバージョンにはパフォーマンスや可用性、管理性の向上のための機能が搭載されています。オペレーティング・システムは、最新のサービス・レベルのOracle Solaris 11.3、またはOracle Solaris 10 1/13（推奨パッチ含む）をすべてのドメインで使用する必要があります。

ベスト・プラクティス：パフォーマンスおよび機能面での最新の改善機能を適用するために、特に制御ドメインとサービス・ドメインではSolaris 11.3以降を使用してください。Oracle VM Server for SPARC 3.3を使用する場合、Solaris 10は制御ドメインでは使用できません。ゲスト・ドメインでOracle Solaris 10を実行する場合も、サービス・ドメインではOracle Solaris 11を使用してください。Oracle Solaris OSのレベルは、異なるドメインでの"ミックス・アンド・マッチ"（混在させたり一致させたりする）が可能です。しかし、最高のパフォーマンスを発揮し、機能を最大限に使用するためには、最新バージョンのOracle Solarisを使用を推奨します。SPARC M5-32システムおよびSPARC M6-32以降のシステム上に制御ドメイン、サービス・ドメイン、およびI/Oドメインを構成する場合には、Oracle Solaris 11.2以降を実行する必要があります。また、SPARC M7、T7以降のシステムでは、制御ドメイン、サービス・ドメイン、I/Oドメインのいずれにもも Solaris 11.3以降が必要となります。

ベスト・プラクティス：ファームウェアのレベルは、プラットフォーム要件およびOracle VM Server for SPARCのバージョン要件に基づき、常に最新にしてください。たとえば、Oracle VM Server for SPARC 3.3リリース・ノートの"システム要件"の項には、サポートされるプラットフォームごとに必要なファームウェアのバージョンが記載されています。

http://docs.oracle.com/cd/E64668_01/html/E64657/requirements.html

---

Oracle Solaris 10のドメインとOracle Solaris 11のドメインを同じサーバー上で稼働させることができます。ゲスト・ドメインでOracle Solaris 10を実行する場合でも、制御ドメインおよびすべてのI/Oドメインとサービス・ドメインでは、Oracle Solaris 11.3以降を使用することを推奨します。Oracle VM Server for SPARC 3.3を使用する場合、制御ドメインではSolaris 10を使用することはできません。Solaris 10で認定されたアプリケーションは、Solaris 11ドメイン内のSolaris 10ブランド・ゾーンで実行できます。この構成では最新のサーバー向けに最適化されたSolaris 11カーネルを利用しながら、Solaris 10のように見せることができます。

Oracle VM Server for SPARCの各ドメインのロール

論理ドメインは、その構成方法、所有するリソース、提供するサービスに応じて1つ以上のロールを持ちます。

» **制御ドメイン**：サーバーまたは物理ドメインごとに1つの制御ドメインがあり、サーバーの電源を投入すると、このドメインが最初に起動します。制御ドメインはハイパーバイザに特権接続でき

るため、リソースやドメインを管理できます。Logical Domains Managerはこのドメインで稼働し、このハイパーバイザ接続を使用して他の論理ドメインを作成および管理します。制御ドメインはルート・コンプレックス・ドメインであり、通常はサービス・ドメインです。制御ドメインは`primary`と呼ばれます。

- **サービス・ドメイン**：サービス・ドメインは、他のドメインに仮想スイッチ、仮想コンソール、仮想ディスク・サーバーなどの仮想デバイス・サービスを提供します。複数のサービス・ドメインを保持することができ、どのドメインもサービス・ドメインとして構成できます。通常、サービス・ドメインはルート・ドメインとして構成されます。
- **I/Oドメイン**：I/Oドメインからは、PCI Express（PCIe）コントローラのネットワーク・カードなどの物理I/Oデバイスに直接アクセスできます。I/Oドメインで構成できるものは次のとおりです。
  - 1つ以上のPCIeルート・コンプレックス(PCIeバス)。ルート・コンプレックスを所有するドメインは、*ルート・ドメイン*とも呼ばれます。
  - PCIeスロットまたはオンボードのPCIeデバイス（ダイレクトI/O（DIO）機能を使用）。
  - シングル・ルートI/O仮想化（SR-IOV）仮想機能。

  1つのサーバー上に構成できるI/Oドメインの最大数は、サーバー固有のPCIeバスの数、PCIeデバイスの数、およびSR-IOV仮想機能の数によって決まります。

  サービス・ドメインはほとんどの場合I/Oドメインですが、I/Oドメインは必ずしもサービス・ドメインとは限りません。I/Oドメインは、他のドメインに仮想I/Oを提供するのではなく、独自のアプリケーション用のI/Oデバイスとして構成する場合があります。サービス・ドメインでもあるI/Oドメインは、物理I/Oデバイスを仮想デバイスとして他のドメインと共有します。I/Oドメインでは、物理I/Oデバイスを使用することでサービス・ドメインへの依存をなくし、独自のアプリケーションに最適なI/Oパフォーマンスを確保することができます。

  Oracle Solarisオペレーティング・システムをI/Oドメインで実行するのは、仮想化されていないシステムでOracle Solarisを実行するのと非常に良く似ています。I/Oドメインで稼働するオペレーティング・システムは、通常の（仮想化されていない）デバイス・ドライバを使用して物理I/Oデバイスにアクセスします。I/Oドメインでは、仮想I/Oデバイスを使用することもできます。
- **ルート・ドメイン**：PCIeバス（PCIeルート・コンプレックスとも呼ばれます）が割り当てられており、そのバス上にあるすべての物理I/Oデバイスに直接アクセスできるI/Oドメインのことをルート・ドメインといいます。1つのサーバー上に作成できるルート・ドメインの最大数は、サーバー・プラットフォーム上にあるPCIeバスの数によって決まります。たとえば、SPARC T4-4サーバーには最大4個、T5-8サーバーには最大16個のルート・ドメインを作成できます。制御ドメインはルート・ドメインとして構成されます。
- **ゲスト・ドメイン**：ゲスト・ドメインは、サービス・ドメインが提供する仮想ディスク、仮想コンソール、および仮想ネットワーク・デバイスを排他的に使用するドメインです。ゲスト・ドメインには物理I/Oデバイスがありません。アプリケーションは、仮想I/Oと物理I/Oのどちらを使用するかを考慮して、ゲスト・ドメインまたはI/Oドメインで実行するように構成します。

ドメイン構成のベスト・プラクティス

この項では、制御ドメインとサービス・ドメインを中心に、Oracle VM Server for SPARCを設定する際に推奨される構成のガイドラインおよびベスト・プラクティスについて説明します。

- サービス・ドメインはCPU、メモリ、およびI/Oを使用して仮想I/Oサービスを提供するため、I/Oを効果的に動作させるには、十分なリソースを割り当てる必要があります。制御ドメインをサー

ビス・ドメインとして使用する場合は、制御ドメインについても当てはまります。サービス・ドメインに十分なリソースを提供することの重要性は、いくら強調しても強調し過ぎになることはありません。

- 最初は、CPUコア2つ（‘ldm set-core 2 primary’）とメモリ16GB（‘ldm set-mem 16g primary’）の最小構成にします。
- 大規模な環境、特にI/O集中型ワークロードの場合は、CPUリソースとメモリ・リソースを追加する必要があります。アプリケーションをI/Oドメインで実行し、仮想I/Oではなく物理I/Oを使用する場合は例外とすることができます。そのような場合は、サービス・ドメインへの割当てが少なめでも十分な場合があります。
- 小規模なテスト環境では割当てを少なめにすることができますが、必ずCPUは1コア以上、メモリは4GB以上にしてください。
- 仮想ディスク・バックエンドにZFSまたはNFSが使用されている場合は、バッファ用に必要となるメモリ容量が増えるため、メモリが4GBでは十分でない場合があります。
- 10GbEのネットワーク・デバイスはCPUコア全体を消費する場合があるため、ネットワーク・デバイスを効果的に動作させるには、コアを追加します。
- ワークロードによってはI/Oの集中度が異なったり、仮想I/Oの代わりに物理I/Oを使用することがあるため、決まったサイジング方法はありません。
- サービス・ドメインでのCPU消費量とメモリ消費量を測定し、サービス・ドメインに適切なリソースが確実に割り当てられるようにします。CPU消費量を確認するには、`ldm list`を`pgstat`、`vmstat`、`mpstat`などの標準のSolarisツールと合わせて使用します。正確なコアおよびパイプラインの消費量を表示するにはpgstatを使用します。メモリ消費量が高くないかを確認するには、‘`vmstat -p`’の出力を確認します。スキャン率（“sr”）がゼロ以外になっている場合、空き容量が少ない場合、またはスワップが発生している場合は、さらに多くのメモリが必要であることを示しています。ときどきチェックするだけでは負荷が最高になるタイミングと時間を検出できないことがあるため、必要に応じて監視の頻度を調整してください。
- サービス・ドメインはと制御ドメインは、無停止で動的に再構成が可能で、要件に応じてリソース割当てを変更することができます。
- 同じCPUコアのCPUリソースを複数のドメインに割り当て場合、キャッシュのフォルス・シェアリングが発生することがあります。これを最小限に抑えるために、CPUリソースはコア単位で割り当てる必要があります。

サービス・ドメインはCPUリソースとメモリ・リソースを使用して仮想I/Oを提供するため、適宜サイジングする必要があります。CPUコア2つとメモリ16GBの最小構成から開始することをお勧めしますが、実際のリソース消費量を測定する必要があります。

- 単一障害点（Single Point of failure）をなくすために、複数のサービス・ドメインを構成します。ゲスト・ドメインは、仮想ネットワークとディスク・デバイスへのパスを冗長化して構成できます。
  - サービス・ドメインでリブートや障害が発生しても、このサービス・ドメインを使用しているゲスト・ドメインが停止されたりリセットしません。ただし、ゲスト・ドメインのI/Oを1つのサービス・ドメインで提供している場合は、サービス・ドメインが停止している間、I/O操作は一時停止し、サービス・ドメインが再稼働すると自動的に再開します。

》サービス・ドメインを冗長構成にしてマルチパス・ディスクとネットワーク・アクセスを提供することができます。この場合は、もう一方のサービス・ドメインを使用してゲストの仮想I/Oが継続されるため、サービスは中断しません。サービス・ドメインは冗長構成にするのがベスト・プラクティスです。これについては後述します。

》サービス・ドメインでは、他のサービス・ドメインが提供する仮想デバイスを使用しないでください。依存性が生じて可用性が低下します。

》サービス・ドメインでは、エクスポートする仮想デバイスに耐障害性があるI/Oバックエンドを使用する必要があります。

》仮想ディスクの冗長構成：

- `ldm add-vdsdev`コマンドを使用して仮想ディスクを`mpgroup`ペアのメンバーとして定義します。コマンドには、異なるサービス・ドメインの仮想ディスク・サービスからアクセスされる同じストレージ・デバイス・バックエンドを指定します。1つの`ldm add-vdisk`コマンドで、`mpgroup`の`vdsdev`定義の1つを指定します。これにより、アクティブ-パッシブ・ペアが形成され、サービス・ドメインに障害が発生したり、リブートしたり、ディスク・デバイスへのアクセスが失われた場合でもアクセスできるようになります。デバイスは1つの冗長仮想ディスクとしてゲストに表示されるため、ゲスト側では何も構成する必要がありません。**注**：仮想ディスク・バックエンド用にNFSを使用する場合、“ソフト”NFSマウントを使用して、バックエンドの障害がサービス・ドメインに影響を及ぼさないことを確認してください。これにより、バックエンド全体が利用可能かどうかを判断します。Oracle VM Server for SPARC 3.2以降では、`ldm set-vdisk`コマンドを使用してアクティブなバックエンドを設定できます。この設定により、サービス・ドメインおよびそのI/Oパスに対してロードバランシングを実行してパフォーマンスを改善できます。**注**：mpgroupをSCSIリザベーションが必要なゲスト・アプリケーションに使用することはできません。詳しくはこちらを参照してください。http://docs.oracle.com/cd/E64668_01/html/E64643/configuringvdiskmultipathing.html
- 各サービス・ドメインではSolaris I/Oマルチパス（MPxIO）を使用し、I/Oの経路障害から保護し、活性保守を可能にします。しかし、これはサービス・ドメインが停止した場合の対障害性を得られるわけではないため、`mpgroup`の代用にはなりません。この2つの方法を併用する必要があります。
- 仮想HBA（詳細はhttp://docs.oracle.com/cd/E64668_01/html/E64643/usingvhbas.html）を複数のサービス・ドメインで使用する場合、ゲスト・ドメインではMPxIOを使用して下さい。これより、アクティブ・アクティブパスの管理および、SCSIリザベーションが利用可能です。

》仮想ネットワークの冗長構成：

- ゲスト・ドメイン内ではIPマルチパス（IPMP）を使用します。2つ以上のサービス・ドメインが提供する仮想ネットワーク・デバイスを使用して、ゲスト・ドメインに2つ以上の仮想ネットワーク・デバイスを定義します。ゲスト・ドメイン側では、リンク・ベースまたはプローブ・ベースのIPMPを使用して仮想ネットワーク・デバイス全体に耐障害性を持たせることができます。Solaris 11が稼働しているゲストでは、オプションとして推移的プローブを使用できます。これにより、ネットワーク・デバイスまたはサービス・ドメインで障害が発生しても、IPMPグループは動作を継続します。
- サービス・ドメインではIEEE 802.3adリンク・アグリゲーションを使用します。仮想スイッチ・バックエンド・デバイスをリンク・アグリゲーションにすると、可用性を向上

させたり負荷を分散させたりすることができますが（http://docs.oracle.com/cd/E56342_01/html/E53796/gmsaa.htmlの「リンク集約の作成」の手順を使用）、サービス・ドメインが停止した場合の耐障害性は得られないため、IPMPと併用します。

» 各サービス・ドメインはルート・ドメインとして構成します。ブートしたり仮想デバイスにサービスを提供したりするために必要なディスク・リソースおよびネットワーク・リソースを代替サービス・ドメインに提供するには、代替サービス・ドメインにPCIルート・コンプレックスを割り当てる必要があります。使用すべきバスおよびデバイスはサーバーのモデルによって異なります。サービス・ドメインに割り当てるバスを特定する方法については、後述します。

サービス・ドメインが複数ある場合は、ゲスト・ドメインのI/Oを冗長構成して耐障害性を持たせることができるため、単一障害点（Single Point of Failure:SPOF）が解消されます。ディスクの冗長構成にmpgroupを使用する場合は、サービス・ドメインでMPxIOを使用します。ディスクの冗長構成に仮想HBAを使用する場合は、ゲスト・ドメイン内でMPxIOを使用します。ネットワークの冗長構成にリンク・アグリゲーションをベースとする仮想スイッチを使用する場合は、ゲスト・ドメインでIPMPを使用します。

» ドメイン操作に意図しない干渉が発生しないようにするために、制御ドメインおよび他のサービス・ドメインへのアクセスは慎重に行う必要があります。制御ドメインまたはサービス・ドメインでアプリケーションを実行することは、一般的には推奨されません。システム監視や管理ソフトウェアの場合は例外とすることができますが、ソフトウェアを追加することによるセキュリティおよび可用性への影響は十分に理解しておく必要があります。

CPUとメモリのベスト・プラクティス

この項では、CPUとメモリの構成について、どのドメインにも該当するガイドラインとベスト・プラクティスについて説明します。Oracle VM Server for SPARCでは、単純かつ効率的なモデルを使用してドメインにCPUリソースを割り当てます。CPUスレッドまたはコア全体が各ドメインに直接割り当てられ、従来のソフトウェア・ベースのハイパーバイザが行うような共有やタイムスライスは行われません。これにより、アプリケーションをネイティブに近いパフォーマンスで動作させることができます。

Oracle VM Server for SPARCでは、ハイパーバイザ・レベルでのソフトウェア・ベースのタイムスライスによって発生するオーバーヘッドがありません。また、アプリケーション要件に応じて動的かつ無停止でCPUリソースをドメインに追加したりドメインから削除したりできます。

メモリ・マッピングや割込みといったレベルのCPU状態の変更は、オペレーティング・システムの特権命令によって実行されます。仮想マシンではこれらの命令を無効にする必要があり、無効にしていない場合は実際のシステム状態が変更され、共有サーバー環境が損なわれることになります。従来からハイパーバイザではプログラム・トラップが発生したり、そうした命令の動的な置換えが行われたりしており、ハイパーバイザのブランチ（CPUのネイティブな命令が仮想マシンのコンテキストに応じてソフトウェアでエミュレートされる現象）の原因となっています。このプロセスは1秒間に何千回も発生する可能性があり、そのたびにコンテキストのスイッチングを何度も行う必要があります。このプロセスは、最近のハードウェア世代（VT-xを搭載したIntelプロセッサ）では改善されていますが、物理CPU上の仮想CPUでのオーバーヘッドは依然として存在します。Oracle VM Server for SPARCでは各ドメインが独自のCPUを持ち、他の仮想マシンに影響を与えずにそれぞれのCPUの状

態を自由に変更できるため、こうした負荷のかかるプロセスは発生しません。従って、Oracle VM Server for SPARCではこのようなオーバーヘッドが発生しません。

Oracle VM Server for SPARCでは、メモリ割当ても単純かつ効率的な方法で行われます。他の仮想化テクノロジーに見られるような、メモリの過剰予約はありません。仮想マシンにメモリを過剰予約し、"最低使用頻度"（LRU）ページをディスクにページ・アウトまたはスワップ・アウトしてメモリの要求を処理するやり方は魅力的に思えますが、実際に仮想マシンで行った場合は問題があります。仮想マシンは、保持しているメモリすべての領域にアクセスすることが多いため、参照局所性が劣っています。その結果、ワーキング・セットのサイズは仮想マシンのメモリ・サイズに近づき、メモリ・ページのすべてが常駐していない場合はパフォーマンスが低下します。つまり、実際には割り当てたメモリと同じ量の実メモリが消費されるのが一般的です。

もう1つ、問題としてあげられるのは、最新のオペレーティング・システムでは仮想メモリが使用されることです。そのため、仮想メモリ・ハイパーバイザの配下に仮想メモリOSを組み合わせると、ディスクにスワップ・アウトするページの選択が適切に行われなくなり、“二重ページング”が発生することが分かっています。実質的な影響として、メモリが過剰予約された環境にある仮想マシンは、予測不能なパフォーマンス"低下"に見舞われるリスクにさらされることになります。過剰予約するVMシステムのほとんどは、このリスクを回避するために、ワークロードが断続的な場合以外は過剰予約のレベルを意図的に低くして実行されます。他にも、“ゲストVMアプリケーションのメモリ・アドレス”を“ゲストVMの実アドレス”にマッピングするという手間がかかります。“ゲストVMの実アドレス”はハイパーバイザ内の仮想メモリであり、最終的には物理メモリ・アドレスにマッピングされます。このマッピング処理には"シャドウ・ページ表"の作成と保守が必要で、CPUサイクルとメモリが消費されます。

Oracle VM Server for SPARCでは、次のような単純なルールを使用することで、この複雑さとパフォーマンス・リスクを解消しています。仮想マシンに専用のメモリを付与します。このメモリはディスクにスワップ・インされたりスワップ・アウトされたりしません。仮想メモリを使用しないため、パフォーマンス低下のリスクがなく、他の環境で使用される"シャドウ・ページ表"のためにメモリを消費する必要もありません。ゲストVMのメモリ割当ては、アプリケーション要件の必要に応じて動的に増減することができます。

仮想マシン密度を増加させる方法としては、Solaris ゾーンが非常に効果的です。Solaris OSでは、1つのリソース・マネージャの元で十分な情報に基づいてゾーンのCPUスケジュールやメモリ管理を決定できます。Solaris 11ドメイン内のSolaris 10ブランド・ゾーンは、最新サーバー向けに最適化されたSolaris 11カーネルで運用しながら、Solaris 10で認定されたアプリケーションを実行できます。Solaris ゾーンとOracle VM Server for SPARCは補完的なテクノロジーです。組み合わせて使用すると強力な仮想化プラットフォームになります。

Oracle VM Server for SPARCのCPUとメモリの割当てモデルを使用することで管理作業は簡素化されますが、いくつかのベスト・プラクティスは依然として必要になります。

- 論理ドメインをサポートしているオラクルのSPARCサーバーには、コアあたり8個のCPUスレッドを搭載した複数のCPUコアが内蔵されています。パフォーマンス上の理由から、CPUリソースはコア単位で割り当てる必要があります。こうすると、CPUコアが複数のドメインに共有される（異なるドメインが同じコアのCPUスレッドを所有する）"スプリット・コア"と呼ばれる状態が発生しなくなります。この状態になると、ドメイン同士がコアのキャッシュを奪い合う"キャッシュのフォルス・シェアリング"が発生し、パフォーマンスが低下します。
    - スプリット・コアを回避するための対策として、Oracle VM Server for SPARCでは未使用

のコアのCPUをドメインに割り当てます。

» もっとも良い方法は、‘ldm set-core 8 mydomain‘のように、コア全体を割当てる方法です。また、コア当たりのスレッド単位で仮想CPUを割り当てる方法もあります。

» T4以降のプロセッサを搭載するSPARCサーバーでは、クリティカル・スレッド・モードを使用できます。このモードでは、1つのソフトウェア・スレッドにコアのパイプラインとキャッシュ・リソースのすべてを割り当てることで、単一スレッドのパフォーマンスを最大化します。これには、前述したようにコア全体を割り当てることと、Solarisが所有しているコアをクリティカル・スレッドに割り当てられるだけの十分なCPUコアが必要です。アプリケーションによって異なりますが、このモードでは通常の"スループット・モード"の場合よりパフォーマンスが数倍向上する可能性があります。

» 通常、このモードにすると効果があるスレッドはSolarisによって検出され、管理者が何もしなくても自動的に"適切な処理が実行"されます。ドメイン内であるかどうかは関係ありません。アプリケーションがこのように動作するように明示的に設定するには、アプリケーションのスケジューリング・クラスをFXに設定し、優先順位を60以上にします。つまり、コマンド‘priocntl -s -c FX -m 60 -p 60 -i $PID’を実行します。 Oracle Database 11g Release 2のような一部のOracleアプリケーションでは、この機能が自動的に設定され、他のプラットフォームからは得られないパフォーマンスを実現します。これについては、『How Oracle Solaris Makes Oracle Database Fast』という記事の「Optimization #2: Critical Threads」を参照してください。
http://www.oracle.com/technetwork/articles/servers-storage-admin/sol-why-os-matters-1961737.html

» このCPU モードをドメイン・レベルで設定するには、コマンドとして‘ldm set-domain threading=max-ipc <domainname>’を実行します。しかし、この設定は非推奨となっているため、Oracle VM Server for SPARC3.3ではこの機能は削除されています。

» NUMAレイテンシ – SPARC T5-8など、複数のCPUソケットを搭載したサーバーでは、CPUとメモリの間でNon-Uniform Memory Access (NUMA) レイテンシ(共有メモリへのアクセス遅延)が発生します。同じソケット上のCPUからのメモリに"ローカル"アクセスする場合は、"リモート"アクセスよりレイテンシが短くなります。このことは、キャッシュに入らないほどメモリ・フットプリントが大きいアプリケーション、またはメモリ・レイテンシに敏感なアプリケーションに影響を与える可能性があります。ただし、これはSPARC T5-1Bのようなワンボードのサーバーには該当しません。ワンボード・サーバーの場合、メモリ・アクセスはすべてローカルに行われるからです。

» Logical Domains Managerは、同じCPUソケット上に位置するCPUコアとメモリをドメインにバインドし、すべてのメモリ参照がローカルになるようにします。ドメインのサイズやコアの事前割当てのためにそれができない場合は、構成が不均衡にならないように、ドメイン・マネージャがCPUとメモリの配分がソケット全体で均等になるようにします。この最適化はドメインのバインド時に自動的に実行されます。

» CPUコアを分割するような小さなドメインを定義することは、推奨しません。複数のアプリケーションに細かい単位でCPU粒度を割り当てる必要がある場合は、より細かいリソース制御ができるようにするために、論理ドメイン内のゾーンにアプリケーションを構成します。

I/Oのベスト・プラクティス

Oracle VM Server for SPARCでは、さまざまな方法でI/Oを構成することができ、その結果として得られるパフォーマンス、利便性、柔軟性、耐障害性は構成方法によって大きく異なります。まず、仮想I/Oをベースとするゲスト・ドメインを使用するか、物理I/Oを使用するI/Oドメインを使用するかを決定します。

物理I/O

物理I/Oはパフォーマンスがネイティブであり、I/Oの動作もネイティブです。つまり、デバイスとデバイス・ドライバは、仮想化されていない環境の場合と同様に動作します。ただし、仮想I/Oが持つ柔軟性は得られず、デバイス共有機能も使用できません。その他の考慮事項は次のとおりです。

- ルート・コンプレックス・ドメインで使用できるデバイスの種類に制限はありません。
- Solarisでは、非ドメイン環境の場合と同様の構成と機能を持つ、Solaris固有のネイティブ・デバイス・ドライバを物理デバイスに使用します。
- ルート・コンプレックス・ドメインの数は、サーバー・プラットフォーム上で使用できるバスの数により制限されます。
- SR-IOVは、Ethernetデバイス、InfiniBandデバイス、およびファイバ・チャネル・デバイスに使用できます。SR-IOVおよびダイレクトI/O（DIO）を使用する場合は、個別に認証されたカードを使用する必要があります。
- SR-IOVの仮想機能とダイレクトI/Oデバイスの数は、それらの機能をサポートする認証カードの種類と数により決定されます。
- SR-IOVはダイレクトI/Oより好ましく、より細かな粒度のリソースを提供します。
- ダイレクトI/OはSPARC M7とT7システムではサポートしていません。SPARC M7とT7システムでは、SR-IOVまたは仮想I/Oを使用してください。
- Oracle VM Server for SPARC 3.2およびSolaris 11.2 SRU 8以降では、SR-IOVを使用するI/Oドメインを構成してシングル・ポイント障害を回避できます。以前のバージョンでは、SR-IOVまたはDIOを使用するドメインは、SR-IOVの仮想機能またはDIOカードに使用されるバスを所有しているドメインに依存することになるため、バスを所有しているドメインでpanicが発生したり、PCIバスがリセットされたりした場合は停止する可能性がありました。現在のバージョンでは、ドメインで異なるルート・ドメインからのSR-IOVの仮想機能を使用し、IPMPまたはMPXIOを使用して可用性を向上させることができます。耐障害性のあるI/Oドメイン（レジリエンスがあるI/Oドメイン）の構成方法については、管理ガイド(http://docs.oracle.com/cd/E64668_01/html/E64643/iodomainresiliency.html)で説明しています。
- 注：Oracle VM Server for SPARC 3.2以降では、Solaris 10はSR-IOVルート・ドメインでサポートされません。また、Solaris 10でのSR-IOV I/Oドメインのサポートは非推奨になり、今後の更新で削除される予定です。詳しくは、Oracle VM Server for SPARCのリリース・ノート(http://docs.oracle.com/cd/E56445_01/html/E56440/deprecatedfeatures.html) を参照してください。

仮想I/O

仮想I/Oは物理I/Oより柔軟性が高いですが、物理I/Oのネイティブ・パフォーマンスは得られません。ただし現在は、サービス・ドメインと仮想デバイスを適切に構成すれば、ネイティブに近いI/Oパフォーマンスを得ることができます。

- 仮想I/Oを使用できるのは、仮想HBA、仮想ディスク、仮想ネットワーク、および仮想コンソール・デバイスのみです。テープなどの他のデバイス・クラスは仮想HBAでサポートされているか、または正規の物理I/Oデバイスを使用して提供されています。
- 仮想デバイスを使用するゲスト・ドメインは、互換性のあるSPARCサーバー同士でライブ・マイグレーションすることができます。
- 仮想I/Oの場合は、同じ物理デバイスを複数の仮想デバイスや複数のドメインのバックエンドとして

共有できます。

》仮想デバイスは、ゲスト・ドメイン内およびデバイスを提供するサービス・ドメイン内の"論理デバイス・チャネル"（LDC）エンドポイントを使用します。LDCの最大数はドメインと関連付けられ、1つのドメインで所有またはサービスを提供できる仮想デバイスの数の上限が設定されることになります。98,304個のLDCのプールから、ドメイン当たりのこの上限がSPARC T4、T5、T7、M5、M6およびM7の各サーバーで1,984となります。このプールは、T4、T5およびT7ではサーバー単位で、M5、M6およびM7では物理ドメイン単位です。システムのソフトウェア要件、ファームウェア要件、およびその他の要件についてはこちらを参照してください。
http://docs.oracle.com/cd/E64668_01/html/E64643/usingldcs.html
UltraSPARC T2システムでは、ドメイン当たりのLDCの上限は512となります。UltraSPARC T2+システムとSPARC T3システムでは、ドメイン当たりのLDCの上限は768で、これ以降のマシンでも必要なファームウェアを実行していない場合は768になります。

》仮想I/Oを使用するには、サービス・ドメインのサイズを適切に設定し、仮想ディスク・サービスとネットワーク・サービスを定義する必要があります。

どのような状況にも適した1つのベスト・プラクティスというものはありません。柔軟性とデバイス共有を重視する場合は、仮想I/Oを選択した方が良いでしょう。これは一般的な構成方法で、共有サーバー上のドメインにワークロードを統合する場合に幅広く使用されています。Oracle VM Server for SPARCの最新バージョンでは、ネイティブに近いパフォーマンスを発揮し、極めて柔軟性に富み、簡単にリソースを共有できる仮想I/Oを実現できます。

最大限のパフォーマンスと他のドメインからの独立性が必要な場合、特にライブ・マイグレーションまたは動的な構成変更が不要なアプリケーションの場合は、ルート・ドメインに構成するのがもっとも良い選択肢でしょう。SR-IOVとDIOはネイティブのパフォーマンスで動作し、リソースの粒度がルート・コンプレックス・ドメインよりも細かいですが、固有のデバイスが必要であり、物理デバイスのバスを所有するドメインにI/Oドメインが依存することになります。

ディスク・デバイスのベスト・プラクティス

ルート・ドメインに設定した物理ディスクI/O、または正規のSR-IOVファイバ・チャネルの仮想機能を使用して設定した物理ディスクI/Oのパフォーマンス特性は、仮想化されていない環境のそれと同等になるはずです。次の各項を実践することで、仮想ディスクのパフォーマンスと柔軟性を実現できます。

》パフォーマンスを最大にするには、ディスク全体（SAN LUNまたはディスク丸ごと）をバックエンドとして使用します。これが、もっとも単純でもっともパフォーマンスの高い仮想ディスクI/Oの選択肢です。サーバー間でドメインをマイグレーションする必要がある場合は、ディスクが複数のホストに表示されます。

》仮想化されていない環境の場合と同様に、複数のディスクに負荷を分散してレイテンシを低減し、キューイングを削減します。

》Solarisおよび論理ドメインのソフトウェア更新にはパフォーマンス改善が含まれるため、ソフトウェアおよびファームウェアのバージョンを最新レベルで維持して、最新のパフォーマンス強化が適用されるようにします。たとえば、Solaris 11.1 SRU 19.6とSolaris 10パッチ150400-13では、Stefan Hinkerのブログの『Improved vDisk Performance for LDoms』
https://blogs.oracle.com/cmt/entry/improved_vdisk_performance_for_ldoms で解説されているとおり、仮想ディスクのパフォーマンスが大幅に改善しています。

》ファイル・システム内のファイルを使用する仮想ディスクは設定が簡単で便利ですが、パフォーマ

ンスは劣ります。テストや開発用、またはディスクI/O要件が厳しくないアプリケーションの場合は、簡易な方法として使用可能です。しかし、このディスク・バックエンドではSCSIリザベーションがサポートされないため、ドメイン間またはホスト間での共有は行わないでください。

- Oracle VM Server for SPARC 3.3以降では、ゲスト・ドメインで仮想SCSIホストバスアダプタ（HBA）を使用することができます。この機能については、https://docs.oracle.com/cd/E64668_01/html/E64643/usingvhbas.htmlで説明されていますが、ネイティブのSolarisデバイス・ドライバでのSANのLUNへのアクセスが使用できます。
- ZFSをディスク・バックエンドに使用することができます。これには便利な機能（クローン、スナップショット、圧縮、高度なデータ整合性）がありますが、RAWデバイスと比較してオーバーヘッドが発生し、いくつかの制限があります。
    - ローカル・ディスク・バックエンドまたはSAN ZFSディスク・バックエンド（サービス・ドメインにマウントされたZFSプール内のオブジェクト）は、ライブ・マイグレーションでは使用できません。zpoolは一度に1つのホストにしかマウントできないためです。これは、Oracle ZFS Storage ZS3などのZFSをローカル・ファイル・システムとして使用するシステムからNFSまたはiSCSIを使用したネットワーク経由でサービス・ドメインに表示されるZFS内のディスク・バックエンドには該当しません。
    - 常時、ZFS圧縮を使用します。デフォルトのアルゴリズム（`zfs set compression=on`）ではCPUがほとんど使用されません。また、送信されるブロックが少なくなるため、実際にI/Oが高速化し領域の消費量は減少します。この設定は、ZFSデータセット・レベルで適用され、ZFSリソースをホストするシステムで実行されます。
    - ZFSボリューム"zvol"（‘`zfs create -V`’）のパフォーマンスはZFSデータセット内のフラット・ファイルをしのぐ可能性があります。
    - アプリケーション用途に応じてZFSの`recordsize`（ファイルの場合）または`volblocksize`（ZFSボリュームの場合）を設定します。これは、非仮想環境での推奨事項と同じです。また、I/Oの回数とオーバーヘッドを増加させるread-modify-writeサイクルを回避します。特に、`recordsize`のデフォルト（128K）は少量のランダムI/Oには適しません。この設定は、ZFSリソースをホストするシステムに適用されます。
    - ZFSは他のバックエンドより多くのメモリを必要とします。サービス・ドメインに十分なメモリがあることを確認し、メモリ不足のまま実行されることのないように、負荷状況下で監視してください。最小構成のサービス・ドメイン（4GB RAM）は、十分ではない場合があります。
- Oracle VM Server for SPARC 3.1.1以降では、ファイバ・チャネル・デバイスにSR-IOVを使用できます。認定に関する詳細は、Oracle VM Server for SPARCのリリース・ノートを参照してください。SR-IOVを使用すると、認定ホスト・バス・アダプタ上のI/Oパフォーマンスを向上することができます。
- NFSおよびiSCSI上の仮想ディスクは、高速（10GbE）ネットワーク上で優れた性能を発揮します。
    - iSCSIの方がNFSよりパフォーマンスが優れている可能性がありますが、実際の結果はパフォーマンス・チューニングに依存します。
    - 非仮想アプリケーションに対して実行するのと同じネットワーク・チューニングを実行します。NFSの場合は、マウント・オプションを指定して`atime`を無効にし、ハード・マウントを使用し（以下に記載した、`mpgroup`を使用して認識された冗長ディスクの使用時は除く）、読み込みサイズと書き込みサイズを大きくします。サービス・ドメインの`/etc/system`に“`set nfs:nfs3_bsize=1048576`”または“`set`

nfs:nfs4_bsize=1048576”（NFSバージョン3クライアントまたはバージョン4クライアントのI/Oサイズの拡張）および“set rpcmod:clnt_max_conns=8”（同時実行数の増加）を含めると、NFSのパフォーマンスが向上する可能性があります。

- NFSバックエンドまたはiSCSIバックエンドをZFSで提供する場合は、標準のSolaris ZFSを使用するかZFS Storage Applianceを使用するかに関係なく、サーバーに十分なキャッシュ用メモリがあることを確認してください。また、同期書き込みを高速化するために、書き込み用に最適化されたソリッド・ステート・ディスク（SSD）をZFS Intent Log（ZIL）用として搭載してください。

- **どのデバイスでも徹底的にネイティブ・パフォーマンスを追求する場合は、PCIeルート・コンプレックス・ドメインを使用して物理I/Oを使用します。こうすることで、何の妥協も手間もなく確実にパフォーマンスを確保することができます。**

ネットワーク・デバイスのベスト・プラクティス

SR-IOV仮想機能、ダイレクトI/Oデバイス、またはルート・コンプレックス・ドメインのネイティブI/Oで提供する物理ネットワークは、仮想化されていない環境の場合と同じであるため、ネットワーク・デバイスのベスト・プラクティスとしては、仮想ネットワーク・デバイスについてのみ説明します。仮想ネットワークのパフォーマンスを目的とした構成方法については、MOS Note『**Best Practices for Oracle Solaris Network Performance with Oracle VM Server for SPARC**』（**Doc ID 1908136.1**）を参照してください。

- サービス・ドメインには、10GbEの帯域幅ごとに8GBのメモリと2つのコアを割り当てます。ゲスト・ドメインには、10GbEの帯域幅ごとに4GBのメモリとコア全体を割り当てる必要があります。これは、仮想ネットワーク・デバイスにCPUをスムーズに分配できるだけの十分なリソースを提供するためです。

- 現行バージョンのOracle VM Server for SPARCとSolarisでは、Large Segment Offload（LSO）とRxDringのサポートを使用して、ネットワーク・レイテンシとCPU使用率を大幅に削減します。この変更と、上記のリソース割当てにより、基本的にワイヤースピードで10GbEを動作させることができます。

  - ドメインでは、Solaris 11 SRU 9.6、またはパッチ150031-07を適用したSolaris 10以降を実行する必要があります。

  - 各ドメインに対して‘ldm set-domain extended-mapin-space=on mydomain’を発行します。事前にこの設定を行っていない場合は、変更を適用するためにドメイン（制御ドメインを含む）をリブートする必要があります。現在はこれが新しいドメインのデフォルトであるため、変更が必要なのは、このサポートが導入される前に作成したドメインのみです。変更が必要なドメインがあるかどうかを確認するには、制御ドメインにログインして‘ldm list -l|grep extended’を実行し、″extended-mapin-space=off″になっているドメインがないか確認します。

  - このオプションを使用するには、ゲストごとに4MBのメモリが必要です。

  - 各ドメインで‘kstat -p|grep dring_mode’を実行し、dring_modeが正しいかどうか確認します。値が4であれば、モードは正しく設定されています。

  - LSOを有効にすると、ジャンボ・フレームを使用した場合にリンク・レイヤーで得られるのと同様のパフォーマンス上のメリットを享受できるため、ジャンボ・フレームを使用する必要はなく、推奨もされません。LSOが有効になっているかどうかを確認するには、コマンド‘kstat -p|grep lso’を実行します。tcp:0:tcpstat:tcp_lso_disabledの値が0、lso_enabledの値が1に設定されている必要があります。

》異なるサービス・ドメインの仮想ネットワーク・デバイスを使用して、ゲスト・ドメインにIPネットワーク・マルチパス（IPMP）グループを構成します。こうすると、デバイスまたはサービス・ドメインの損失に対する耐障害性が実現され、アクセスできなくなることがなくなります。非仮想システムの場合と同様に、サイトの優先順位に基づき、リンク・ベースまたはプローブ・ベースのいずれのIPMPも使用できます。リンク・ベースのIPMPの場合は、仮想ネットワーク・デバイスを定義するときに“linkprop=phys-state”を指定し、物理リンクの状態を仮想ネットワーク・リンクに適用させます。

》サービス・ドメインにリンク・アグリゲーションを構成し、帯域幅、ロードバランシング、およびネットワーク・リンク・フェイルオーバーにIEEE 802.3adの機能を使用できるようにします。リンク・アグリゲーションは、物理ネットワーク・リンクではなく仮想スイッチ・バックエンド・デバイスのアグリゲーションを使用して有効にします。リンク・アグリゲーションはゲスト・ドメインでのIPMPの使用を補完する手法で、この2つの手法を組み合わせることができます。

》複数のネットワーク・リンクを使用し、ジャンボ・フレーム（LSOを使用しない場合）を使用し、VLANを作成し、TCPウィンドウおよび他のシステム設定を調整します。この方法は非仮想環境の場合と同じで、理由も同じです。

》論理ドメイン・チャネルが過度に使用されている場合は、仮想ネットワーク・スイッチにプロパティ“inter-vnet-link=off”を設定します。ゲスト間のすべての通信トラフィックは、ゲスト・ドメイン間で直接通信されるのではなく仮想スイッチを経由するため、この設定にすると、同じ仮想スイッチ上にあるドメイン間のパケット通信にかかる時間がわずかに増加します。

》同じ仮想スイッチに複数のドメインを組み合わせている場合は、ネットワーク帯域幅制御を使用し、共有ネットワーク・リソースを過剰に消費している仮想マシンについて、‘ldm set-vnet maxbw=200M net0 mydomain’などを考慮します。

》Solarisのゾーンを使用するなど、Solaris 11のネットワーク仮想化をゲスト・ドメインで使用する場合は、複数のMACアドレスを持つ仮想ネットワーク・デバイスを構成する必要があります。そのためには、‘ldm add-vnet alt-mac-addrs=auto,auto,auto,auto vnet1 primary-vsw0 mydomain1’のように、alt-mac-addrパラメータを使用して仮想ネットワーク・デバイスを作成します。詳しい説明と例については、https://blogs.oracle.com/jsavit/entry/vnics_on_vnets_now_available を参照してください。

ライブ・マイグレーションのベスト・プラクティス

ライブ・マイグレーションは、ドメインの実行を継続しながら、あるサーバー（ソース・サーバー）から別のサーバー（ターゲット・サーバー）へゲスト・ドメインを移動する場合に使用することができます。また、メンテナンスのためにサーバーを退避させる、サーバー・プール全体に仮想マシンのロードバランシングを行う、リソースを解放して他のドメインに追加のCPUやメモリを割り当てられるようにする、といったことを可能にすることで、ドメイン環境内での運用を改善する目的に使用することもできます。

ライブ・マイグレーションには、次のようないくつかの要件があります。

》ライブ・マイグレーションを実行するには、互換性のあるバージョンのOracle VM Server for SPARCおよびファームウェアを実行しているサーバーが必要です。推奨としては、サーバー間ではOracle VM Server for SPARCを同じバージョンに統一し、同じサーバータイプ上では、同じファームウェア・レベルを使用する必要があります(たとえば、T7のすべてのシステムは同じファームウェア・レベルを実行する必要があります)。無停止で新しいソフトウェアおよびファームウェア・レベルへアップグレードする際にライブ・マイグレーションを行う場合は例外です。バージョンによる互換性については、リリース・ノートを参照して下さい。

》CPU間ライブ・マイグレーション時以外では、ソース・サーバーとターゲット・サーバーは、クロック周波数が同じCPUチップを搭載している必要があります。『Oracle VM Server for SPARC管理ガイド』の「CPUのドメイン・マイグレーション要件」に説明がありますが、ドメインの`cpu-arch`プロパティを設定すると、異なる種類のプロセッサを搭載したシステム間での移行が可能になります。デフォルト設定は`native`です。他の値に設定すると、設定によっては暗号化処理をオフロードするといった高度なCPU機能の一部が無効になる場合があります。このため、サーバーがすべて同じ種類である場合は、`native`設定にしてください。`cpu-arch`を変更できるのは、ドメインが実行されていない場合だけです。そのため、異なるCPU間のマイグレーションは事前に計画する必要があります。

》ターゲット・マシンには、マイグレーションするドメインで使用されているリソースに対応できるだけの十分なCPUとメモリが必要です。を必要とします。

》ゲスト・ドメイン（仮想デバイスを排他的に使用するドメイン）のみをライブ・マイグレーションできます。物理I/O（専用のPCIeバス、ダイレクトI/Oカード、SR-IOV）を持つドメインのライブ・マイグレーションはできません。ドメインから（たとえばIPMPペアの一部である）SR-IOVデバイスの設定を解除し、ドメインをライブ・マイグレーションして、ターゲット・システムでSR-IOVデバイスを再度設定し割り当てることは可能です。

》ゲスト・ドメインではネットワーク接続されたストレージを使用する必要があり、移行するドメイン内の各仮想ディスク・バックエンドがターゲット・マシンに定義されている必要があります。この共有ストレージには、SANディスクか、NFSまたはiSCSIで提供されるストレージを使用できます。仮想ディスク・バックエンドには、ソース・マシンのものと同じボリューム名およびサービス名を付ける必要があります。ソース・マシンとターゲット・マシンとでバックエンドへのパスが異なっても構いませんが、同じバックエンドを参照する必要があります。

》マイグレーションするドメイン内の各仮想ネットワーク・デバイスに対し、対応する仮想ネットワーク・スイッチがターゲット・マシンに存在する必要があります。各仮想ネットワーク・スイッチには、ソース・マシンでデバイスが接続されている仮想ネットワーク・スイッチと同じ名前を付ける必要があります。

》ソース・サーバーとターゲット・サーバー間ではネットワークへのアクセスが必要です。

これらの要件を満たしているドメインは、シャットダウンせずに移行できます。次のような複数のベスト・プラクティスが適用されます。

》ライブ・マイグレーションでは、ソース（移行元）ホストの制御ドメインのCPUに大きな負荷がかかります。制御ドメインにCPUコアを追加すると、ドメインの移行時間と一時停止時間を大幅に短縮することができます。CPUコアは、ライブ・マイグレーション前に追加してライブ・マイグレーション後に取り外すことができます。

》SPARC T3やそれ以前のモデルなど、旧SPARCサーバーの場合は、制御ドメインの各CPUコアに暗号化アクセラレータを追加します（‘`ldm set-crypto 2 primary`’）。これにより、SPARCプロセッサのハードウェア・アクセラレーションが使用され、移行処理が高速化されます。SPARC T4以降のプロセッサの場合は、管理者が何もしなくても、ハードウェア暗号化アクセラレーションは常時利用可能です。

》ライブ・マイグレーションを実行すると、マイグレーションするドメインのCPUにも負荷が集中するため、アクティビティが少ない時間帯にドメインを移行してください。SPARC M7およびT7サーバーでSolaris 11.3以降を使用することにより、CPU負荷が大幅に低減します。

》メモリの変更量が多いゲスト・ドメインの場合は、メモリの内容を場合によっては何度も送信し直す必要があるため、マイグレーションにかかる時間が長くなります。変更されたページの追跡にかかるオーバーヘッドも、ゲスト・ドメインのCPU使用率を増加させますが、SPARC M7およびT7サーバーで

Solaris 11.3以降を使用している場合には、CPU負荷が大幅に低減します。一般的に、ライブ・マイグレーションは大規模なゲスト・ドメインほど時間が長くなり、メモリを大量に使用するアプリケーションに対しては必ずしも適切な手段ではありません。このようなケースについては、アプリケーション・レベルのクラスタリングの方が効率的で便利であることがよくあります。

- ライブ・マイグレーションには広い帯域幅のネットワークを使用する必要があります。
- クロック動作の整合性が確保されるように、マイグレーションするゲスト・ドメインはNetwork Time Protocol（NTP）を使用する必要があります。
- ライブ・マイグレーションが可能なのはゲスト・ドメインのみです。I/Oドメインはマイグレーションの対象外です。

Oracle VM Server for SPARCでは、ライブ・マイグレーションは他の仮想化テクノロジーほど運用上大きな位置付けを占める機能ではありません。一部の仮想化プラットフォームには、システムを停止せずにシステム・ソフトウェアを更新する機能がないため、ライブ・マイグレーションを使用してサーバーを"退避"させる必要があります。Oracle VM Server for SPARCの場合は、代替サービス・ドメインを使用することで、サーバーを退避させる必要もなく"ローリング・アップグレード"の実行が可能になります。たとえば、制御ドメインとサービス・ドメインのSolarisを通常営業時間中に同時にアップデートし、その後、片方ずつリブートして新しいレベルにSolarisをアップグレードすることができます。1つのサービス・ドメインがリブートしている間、仮想I/Oは代替サービス・ドメインで続行されるため、アプリケーションの可用性を損なうことなく、すべてのサービス・ドメインのSolarisをアップグレードできます。

Oracle VM Server for SPARCを使用する場合は、ドメインのCPUリソース、メモリ・リソース、I/Oリソースの増減もできます。これらのリソースが必要なときは、使用率の低いサーバーにドメインを移行してリソースを増やすよりも、同じサーバー上の別のドメインのCPUやメモリの割当て量を減らす方が簡単かつ高速で、ドメインの移行はまったく必要ありません。その結果、Oracle VM Server for SPARCを使用すると、ライブ・マイグレーションを使用しなくても運用上対応可能なケースも数多くあります。

ライブ・マイグレーションはどのアプリケーションにも適しているわけではありません。マイグレーションの最後のフェーズではレイテンシや一時停止期間が変化しやすいため、特にリアルタイムのネットワーク・ハートビートを使用してアプリケーションの状態を監視するアプリケーションは、移行時に正しく動作しない可能性があります。また、移行中にネットワーク・パケットがドロップする可能性もあります。ライブ・マイグレーションはアプリケーション要件を考慮して検討する必要があります。

Oracle Real Applicationのようなアプリケーション・フレームワークでは複数ノードのアプリケーションが提供され、これらはほとんどの場合、移行する必要がまったくありません。クラスタ内のノードを停止する方が簡単かつ高速であり、異なるホストで別のノードを起動することも任意でできます。障害が発生したシステムからドメインをライブ・マイグレーションすることはできないため、ライブ・マイグレーションが高可用性設計の代わりになるとは考えないでください。

可用性のベスト・プラクティス

可用性というトピックは範囲が広いため、この項ではOracle VM Server for SPARCに固有の側面のみを取り上げ、ドメインで実行されるアプリケーションとゲストの信頼性、可用性、保守性（RAS）の向上について説明します。

まず、可用性を向上させるためのガイドラインのうち、Oracle VM Server for SPARCに該当するものをいくつか紹介します。

- **単一障害点（Single Point Of Failure:SPOF）の回避**：システムは、コンポーネント障害が発生してもアプリケーション・サービスが中断しないように構成する必要があります。SPOFを回避する一般的な方法としては、コンポーネント障害が発生しても中断なしにサービスを続行できるように、冗長化する方法があります。Oracle VM Server for SPARCを使用すると、SPOFを回避するシステムを構成することができます。

- **ビジネス要件に即したリソースと作業量のレベルで可用性を構成**：作業量とリソースはビジネス要件に即している必要があります。本番環境での可用性の要件はテスト/開発環境と異なるため、余裕のあるリソースを割り当てて可用性を高くする必要があります。同じ本番環境でも、重大性のレベル、停止に対する許容度、リカバリ要件、修復時間要件は異なる場合があります。"あらゆることを実行"しようとして複雑になった設計より、単純な設計の方が理解しやすく効果的なこともあります。

- **適切なレイヤまたはレベルのプラットフォーム・スタックで可用性を設計**：可用性はアプリケーション、データベース、またはそれらが依存する仮想化レイヤ、ハードウェア・レイヤおよびネットワーク・レイヤに提供でき、これらすべてを組み合わせたものにも提供できます。ネットワーク・ロードバランサによって可用性が提供されるステートレスWebアプリケーションや、Oracle Real Application Clusters（Oracle RAC）やWebLogicなどのように独自の耐障害性や可用性に対する機能を備えたエンタープライズ・アプリケーションの場合は、耐障害性を備えた仮想化を設計する必要がないこともあります。

- **仮想かどうかに関わらず、ほとんどの場合は同じアーキテクチャを使用**：たとえば、デバイス・パスの喪失やディスク・メディアの障害に対する耐障害性については、どのドメインでも必要となるため同じ設計が適用されます。

- **ドメインを使用するかどうかに関わらず、ほとんどの場合は同じテクニックを使用**：構成手順の多くはドメインを使用するかどうかに関わらず同じです。たとえば、IPMPの構成や冗長ZFSプールの作成は、ゲスト・ドメイン内かどうかに関係なく、ほとんど同じ構成手順です。仮想ネットワークと仮想ディスク・デバイスを持つゲスト・ドメインをプロビジョニングするための構成手順と選択肢がありますが、これについては後述します。

- **ドメインを使用する場合は異なる点もある**：Oracle VM Server for SPARCには従来の仮想マシン環境と異なる固有のドメインがあり、代替サービス・ドメインという機能を提供します。これは、ドメイン障害が発生した場合に耐障害性を提供するものですが、"ローリング・アップグレード"による保守性の向上も可能にします。これは、SPOFといえる単一の基盤上ですべての仮想I/Oを構築する従来の仮想マシン環境とOracle VM Server for SPARCとの重要な差別化要因です。代替サービス・ドメインは、耐障害性を備えた本番の論理ドメイン環境において幅広く使用されます。

- **論理ドメインのコマンドで実行されるタスクもあるが、ゲストで実行されるタスクもある**：たとえば、Oracle VM Server for SPARCでは、ゲスト・ドメインに対して複数のネットワーク接続を提供し、ネットワークの耐障害性はIPマルチパス（IPMP）を使用してゲスト・ドメインで構成します。これは、非仮想システムの場合と基本的に同じです。一方、仮想ディスクの可用性は仮想化レイヤおよびサービス・ドメインで構成するため、ゲスト・ドメインで詳細な設定を行う必要がなく、あらかじめ耐障害性を備えたディスクがゲスト・ドメインに表示されます。

- **ライブ・マイグレーションは"高可用性"にあらず**："継続利用が可能"という意味では、ライブ・マイグレーションは高可用性ではありません。サーバーがダウンした場合、そのサーバーからはライブ・マイグレーションせず、他の場所でクラスタまたは仮想マシンを再起動します。ただし、ライブ・マイグレーションを使用することで保守性が向上する（サービスの計画停止や計画保守の前に、稼働中のドメインをサーバーから移動させることができる）ため、これをRASソリューションに取り入れることができます。

- **ドメインの構成を保存**：電源のオン/オフを行ってもドメインの構成を引き続き有効にするためには、ドメインの構成を変更した後に‘`ldm add-config` *`configname`*’コマンドを発行して、変更内容をサー

ビス・プロセッサに保存します。サービス・プロセッサ（SP）の構成と制約を含むbootsetsは、構成をSPに保存したときに制御ドメインに保存され、電源オン/オフ後に制約データベースを再ロードするときに使用されます。制御ドメインの/var/opt/SUNwldm/autosave-<configname>にコピーが保存されるため、サービス・プロセッサに保存されている構成よりディスク上のコピーの方が新しい場合は自動リカバリが行われるように、ポリシーを設定することができます。これについては、http://docs.oracle.com/cd/E64668_01/html/E64643/managingldomsconfigurations.html を参照してください。

- **ドメイン構成をバックアップ**：ドメイン構成は、XMLファイルにエクスポートして保存します。XMLファイルは別のホストに保存できます。‘ldm list-constraints -x mydom > mydom.xml’を使用すると1つのドメインの情報が保存され、‘ldm list-constraints -x > everything.xml’を使用するとシステム全体の情報が保存されます。別のサーバーにドメイン構成をリストアする必要がある場合は、‘ldm add-dom -I mydom.xml; ldm bind mydom; ldm start mydom’のようなシーケンスを使用すると個々のドメインをリストアでき、‘ldm init-system -i everything.xml’を使用すると工場出荷時のデフォルト・モードからドメイン構成全体をリストアできます。

- **ドメインの依存関係を構成**：ドメインが相互に依存している場合は、それがサービス・ドメインの要件によるものかアプリケーションの要件によるものかに関係なく、ドメイン間の関係を明示し、ドメイン同士の関係を定義して障害ポリシーを設定することができます。詳しくは、http://docs.oracle.com/cd/E64668_01/html/E64643/configuredomaindependencies.html を参照してください。

ネットワークの可用性

ゲスト・ドメインのネットワーク接続には、冗長仮想ネットワーク・デバイスを使用して耐障害性を持たせます。これは、仮想化されていない環境で冗長物理ネットワーク・デバイスを使用するのと同じです。ゲスト・ドメインには複数の仮想ネットワーク・デバイスを定義します。各デバイスは、異なる物理バックエンドに関連付けられている異なる仮想スイッチ（"vswitch"）に接続します。バックエンドは物理リンクまたはアグリゲーションのいずれでも構いませんが、アグリゲーションの方がデバイス障害に対する耐障害性を向上でき、パフォーマンスも向上する可能性があります。

Solaris IPマルチパス（IPMP）はSolarisの物理インスタンスの場合と同様にゲスト・ドメイン内で構成します。IPMPグループ内のリンクで障害が発生した場合は、残っているリンクで継続的にネットワーク接続を提供します。ゲスト・ドメイン内での目に見える違いはデバイス名です。これは、nxge0やixgbe0のような物理リンク名の代わりに、仮想ネットワーク・リンクの"合成"デバイス名（vnet0、vnet1など）が使用されるためです。

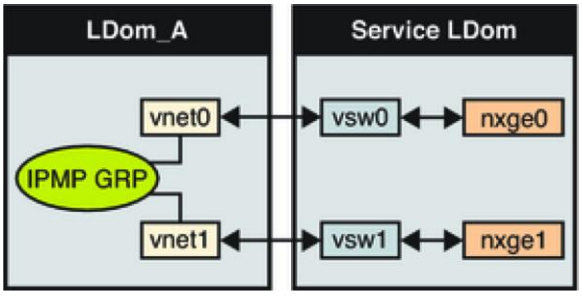

図1：異なる仮想スイッチ・インスタンスに接続された2つの仮想ネットワーク

こうすることで、スイッチ、ケーブル、NICなどの障害から保護します。制御ドメインにprimary-vsw0とprimary-vsw1という名前の仮想スイッチがあるとします。それぞれ、異なる物理ネットワーク・デバイスを使用しています。つまり、primary-vsw0は制御ドメインのnet0に接続し、primary-vsw1はnet1に接続しているということです。これは、次のように簡単に設定できます。

```
# ldm add-vnet linkprop=phys-state vnet0 primary-vsw0 ldg1

# ldm add-vnet linkprop=phys-state vnet1 primary-vsw1 ldg1
```

オプションのパラメータ（linkprop=phys-state）は、リンク・ベースのIPマルチパスをゲスト・ドメインで使用するよう指定しているため、リンク・ステータスの変更をSolarisで検出できます。テストIPアドレスを持つプローブ・ベースのIPMPをSolarisで使用する場合は、このパラメータを省略できます。詳しい説明は、**『仮想ネットワークの使用とLogical Domains環境でのIPMPの構成』**（http://docs.oracle.com/cd/E64668_01/html/E64643/configuringipmpinldomsenv.html）を参照してください。残っている作業はゲスト・ドメインでのIPMPの構成ですが、これは非ドメイン環境の場合と同じです。

異なるサービス・ドメインから仮想ネットワーク・デバイスをプロビジョニングすると、耐障害性が強化されます。サービス・ドメインのいずれが障害を起こしたりシャットダウンしたりしても、残っているサービス・ドメインが提供する仮想ネットワーク・デバイスを経由してネットワーク・アクセスが継続されます。使用するサービス・ドメインの切替えは、ゲスト・ドメインのSolaris OSには透過的に実行されます。構成上の違いは、上記のコマンド・シーケンスで異なるサービス・ドメインの仮想スイッチを使用するところだけです。

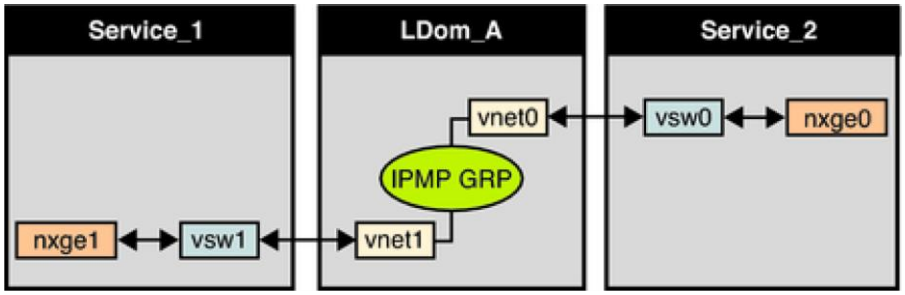

図2：それぞれが異なるサービス・ドメインに接続されている仮想ネットワーク・デバイス

ディスクの可用性

冗長性を構成する目的は基本的に、データ損失や停止を招かずにメディア障害やパス障害に耐えられるようにすることにあります。Oracle VM Server for SPARCでは、物理システムと同様の手法を使用します。つまり、耐障害性を備えたバックエンドとともに仮想ディスクを設定するか、耐障害性を備えていないディスクをミラー化するか、またはその両方を行います。

どの仮想ディスクもバックエンドを所有しますが、バックエンドには物理ディスクまたはLUN、iSCSIターゲット、ZFSボリューム（"zvol"）またはSolaris Volume Manager（SVM）などのボリューム・マネージャをベースとするディスク・ボリューム、またはファイル・システム（NFSを含む）上のファイルを使用できます。次の図は、ゲスト・ドメインにある仮想ディスク・クライアント（"vdc"）のデバイス・ドライバと、これと連携するサービス・ドメインにある仮想ディスク・サーバー（"vds"）のドライバの関係を示しています。これらは仮想ディスクと物理バックエンドを関連付ける論理ドメイン・チャネル（LDC）で接続されています。

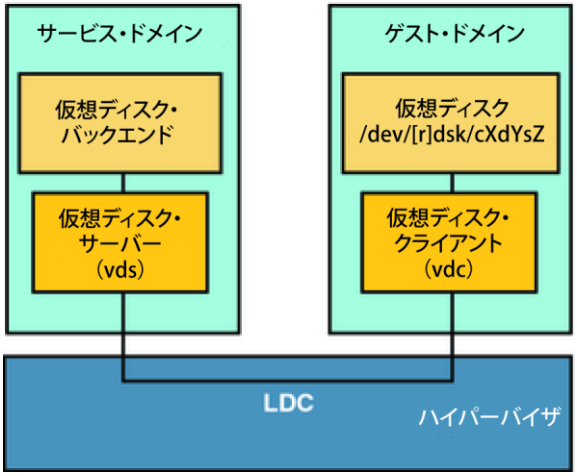

図3：Oracle VM Server for SPARCを使用した仮想ディスク

ドメイン構成におけるもっとも重要な選択の1つは、ディスク・バックエンドをLUNやローカル・ディスクといった物理デバイスにするか、サービス・ドメインにマウントしたファイル・システム内のファイルにするかを決めることです。この選択は、パフォーマンスや可用性に影響を与え、ライブ・マイグレーションの実行やSCSIリザベーションのサポートといった機能にも影響します。物理デバイス（LUNなど）はもっともパフォーマンスに優れていますが、ファイル・ベースのディスク・バックエンドはストレージ・アレイから物理デバイスを割り当てる必要がなく、必要に応じて作成できるため、利便性に優れています。次に、物理デバイス・バックエンド、ファイル・ベース・バックエンド、ZFSボリュームの使用例を示します。

LUNをベースとする仮想ディスク

```
# ldm add-vdsdev /dev/rdsk/c0t5000C5003330AC6Bd0s2 myguestdisk0@primary-vds0
# ldm add-vdisk my0 myguestdisk0@primary-vds0 mydom
```

ファイル（この例ではNFS上のファイル）をベースとする仮想ディスク

```
# mkfile -n 20g /ldomsnfs/ldom0/disk0.img
# ldm add-vdsdev /ldomsnfs/ldom0/disk0.img ldom0disk0@primary-vds0
# ldm add-vdisk vdisk00 ldom0disk0@primary-vds0 ldg0
```

ZFSボリュームをベースとする仮想ディスク

```
# zfs create -V 20g rpool/export/home/ldoms/ldom1-disk0
# ldm add-vdsdev /dev/zvol/rdsk/rpool/export/home/ldoms/ldom1-disk0 ¥
            ldg1-disk0@primary-vds0
# ldm add-vdisk vdisk10 ldg1-disk0@primary-vds0 ldg1
```

1つ目の例では、サービス・ドメイン（この例では制御ドメイン）から認識できるデバイスを選んでゲストに提示しています。サービス・ドメインでは、デバイスをフォーマットしたり、デバイスに

ファイル・システムをマウントしたりしません。これはもっとも単純なバックエンドで、サービス・ドメインはすべてのI/Oアクティビティを"パススルー"します。パフォーマンス重視の場合はこの設定がベスト・プラクティスとして推奨されます。

他の2つの例では、サービス・ドメインから認識できるファイル・システム内にあるファイルまたはZFSボリュームから仮想ディスクを作成しています。間接レイヤがありファイルのバッファリングが行われるためレイテンシが増加しますが、ミラー化やRAIDZといったZFSの機能、サービス・ドメインのファイル・システムで提供される圧縮やクローン作成などの機能をゲスト・ドメインで利用できます。

機能的な違いもあります。たとえば、（NFSではない）ローカルのZFSディスク・バックエンドは、ライブ・マイグレーションに使用できません。ZFSプールは一度に1つのSolarisインスタンス（この場合は1つの制御ドメイン）にしかインポートできないためです。

ディスクにメディア障害への耐性を持たせることができますが、その場合は、RAID環境またはZFS環境にある冗長ディスク・バックエンドを使用する、ゲスト・ドメイン内の仮想ディスクを冗長ZFSプールに組み込む、といった非仮想環境の場合と同じ手法を使用します。

また、デバイスへの冗長パスの可用性の確保も同様に考慮する必要があります。ファイバ・チャネル・ストレージ・エリア・ネットワーク（FC SAN）でもNFSデバイスでも複数のアクセス・パスがサポートされます。FC SANコントローラには、複数のサーバーに冗長アクセスを提供するためのホスト・インタフェースを複数構成できます。このとき、なるべく複数のFCスイッチを使用します。各サーバーには、FCファブリックに接続するためのホスト・バス・アダプタ（HBA）カードを複数構成できます。これらを併用することで、FCコントローラ上の個々のHBA、ケーブル、FCスイッチ、またはホスト・インタフェースの損失に対して耐障害性を実現します。これは、メインフレームのDASDで長い間実行されてきた、複数のチャネルと制御ユニットを構成する手法に似ています。

同じ考え方は、NFSまたはiSCSIをベースとする仮想ディスクでも使用できます。こうしたケースでは、物理トランスポートにイーサネットを使用し、耐障害性は複数のネットワーク・デバイスを使用することにより実現できます。ネットワーク・デバイスとしては、アグリゲートまたはIPMPグループも使用できます。

冗長ディスク・パスは、マルチパス・グループ（"mpgroup"）を使用して定義した仮想ディスクを使用して論理ドメインに認識されます。これについては、『Oracle VM Server for SPARC管理ガイド』の「仮想ディスク・マルチパスの構成」の項（http://docs.oracle.com/cd/E64668_01/html/E64643/configuringvdiskmultipathing.html）を参照してください。これにより、サービス・ドメインのいずれかが停止した場合でも継続して動作するアクティブ/パッシブ・ペアが作成されます。mpgroupをSCSIリザベーションが必要なアプリケーションに使用することはできません。SCSIリザベーションが必要な場合は、代わりに仮想HBA（vHBA）を使用する方法があります。

論理ドメイン管理者は、同じディスク・バックエンド（ここは重要であるため強調しますが、複数のパスからアクセスされる同じデバイスのことです）に複数のパスを定義し、いずれかのパスを使用してゲスト・ドメインにディスクを追加します。この複数のサービス・ドメインで利用できるNFSバックエンドを使用した例を下記で説明しています。

```
# ldm add-vdsdev mpgroup=mpgroup1 /ldoms/myguest/disk0.img disk@primary-vds0

# ldm add-vdsdev mpgroup=mpgroup1 /ldoms/myguest/disk0.img disk@alternate-vds0

# ldm add-vdisk mydisk disk@primary-vds0 myguest
```

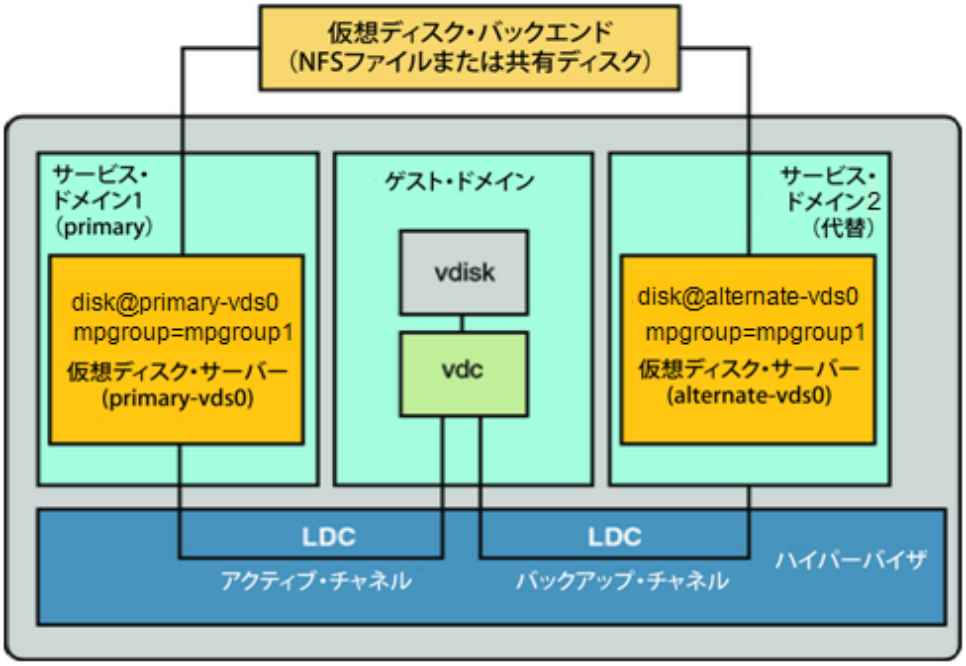

図4：仮想ディスク・マルチパスの構成

複数のサービス・ドメインを使用したサービス・ドメインの可用性

サービス・ドメインが1つの場合、そこが単一障害点(SPOF)になりかねません。サービス・ドメインで障害が発生したり保守作業のためにサービス・ドメインが停止したりしている場合は、ドメインから提供されているデバイスが使用できなくなり、I/Oにアクセスするゲスト・ドメインは、サービス・ドメインがリストアされるまで一時停止します。この種の中断をなくすために、複数のサービス・ドメインを使用するというベスト・プラクティスが幅広く使用されています。この方法は、サービス・ドメインを１つずつ停止して保守作業を行う"ローリング・アップグレード"にも使用でき、ライブ・マイグレーションを使用してサーバーを"退避"させずにサービスの損失を防ぐことができます。

この項では、代替サービス・ドメインの構成について詳しく説明します。詳細については、『Oracle VM Server for SPARC管理ガイド』の「PCIeバスの割り当てによってルート・ドメインを作成する方法」の項（http://docs.oracle.com/cd/E64668_01/html/E64643/rootdomainwithpcibuses.html）を参照してください。

- 制御ドメインおよびI/Oドメインが独自の物理I/Oを使用して動作できるように、使用しているシステムの複数のPCIeバス上にネットワーク・デバイスとディスク・デバイスが物理的に構成されていることを確認します。サービス・ドメインがそのディスク・デバイスやネットワーク・デバイスについて他のサービス・ドメインに依存しないようにする必要があります。依存していると、そこが単一障害点(SPOF)となり、サービス・ドメインが複数存在する価値がなくなります。
- 制御ドメインに必要なPCIeバスがどれかを識別します。少なくとも、制御ドメインのブート・ディスクおよびログインに使用されるネットワーク・リンクが接続されているバスが対象になります。制御ドメインがサービス・ドメインでもある場合、一般に、ゲスト・ドメインにサービスを提供する仮想デバイスが使用するバックエンド・デバイスが接続されているバスも対象になります。
- ルート・ドメインと代替サービス・ドメインに必要なPCIeバスがどれかを識別します。制御ドメインの場合と同様、ドメインのブートに必要なデバイスと、仮想ネットワークとディスク・バックエンドに使用されるすべてのデバイスが接続されているバスが対象となります。

》代替サービス・ドメインとして使用するルート・ドメインを定義します。

》選択したバスを制御ドメインからサービス・ドメインに移動します。これには、制御ドメインの遅延再構成とリブートが必要になります。

》制御ドメインから、制御ドメインとサービス・ドメイン両方を使用する仮想ディスク・サービスとネットワーク・サービスを定義します。

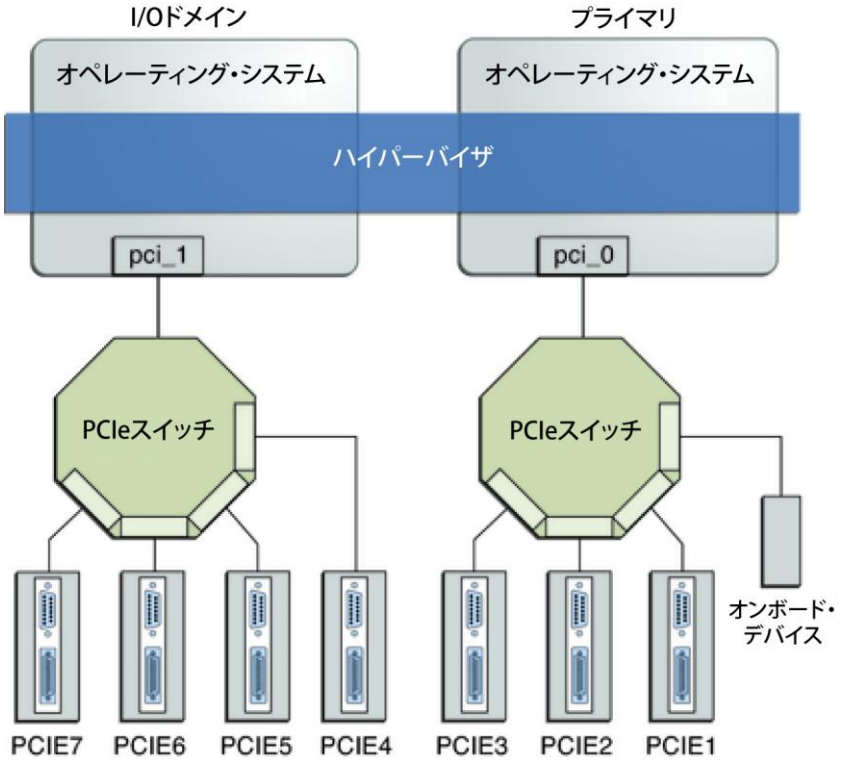

図5：I/OドメインへのPCIeバスの割り当て

複数のサービス・ドメインを使用した構成例

この例では、サーバーに制御ドメインとルート・コンプレックス・ドメインを構成します。これらのドメインは両方ともルート・ドメインで、ゲスト・ドメインに仮想ディスクおよび仮想ネットワーク・サービスを提供するサービス・ドメインとして使用します。

それぞれのドメインには、表1に示す一連のリソースを割り当てます。

ドメイン	CPUおよびメモリ	デバイス
制御ドメイン	2CPUコア（16仮想CPU） 16 GB	PCIバス（ストレージ・アレイに接続されたHBA、内部ディスク、オンボードNIC）
セカンダリ・サービス・ドメイン	2CPUコア（16仮想CPU） 16 GB	PCIバス（ストレージ・アレイに接続されたHBA、内部ディスク、オンボードNIC）
ゲスト・ドメイン（可変数）	アプリケーション要件に応じた必要な量	制御ドメインおよびセカンダリ・サービス・ドメインが提供する仮想ディスクおよびネットワーク・デバイス

表1：

この構成では、すべてのシステム・リソースを使用しません。より重要なワークロードを処理するなどでリソースの追加が必要になった場合は、未割当てのリソースを使用して追加のドメインを作成したり、既存のドメインを再構成したりできます。使用可能なリソースを表示するためには、‘`ldm list-devices`’コマンドを使用できます。Oracle VM Server for SPARCには、必要に応じてリソースを動的に割り当てられるという便利な機能があります。

サービス・ドメインとして動作させるために、制御ドメインには次の仮想デバイス・サービスを構成します。

- 物理ディスクを仮想ディスクとしてゲスト・ドメインにエクスポートするために使用する1つの仮想ディスク・サービス（`primary-vds0`）。
- ゲスト・ドメインの仮想コンソールへのアクセスを提供するために使用する1つの仮想コンソール・コンセントレータ・サービス（`primary-vcc0`）。このサービスは5000～5100のポート範囲を使用して、仮想コンソール・アクセスを提供します。
- 1つ以上の仮想スイッチ・サービス（この例では、ネットワーク`net0`に関連付けられている`primary-vsw0`）。

セカンダリ・サービス・ドメインも同様のサービスを定義します。ただし、ドメイン名を示すように変更したサービス名が使用されます。

ソフトウェアのインストールと構成

サーバー・ソフトウェアのインストールには、次のタスクが含まれます。

- Oracle Solarisのインストール
- Oracle VM Server for SPARCのインストール
- 制御ドメインの構成
- セカンダリ・サービス・ドメインの構成
- ゲスト・ドメインの構成

ネットワークおよびストレージ関連のタスクを含む、その他の構成タスクについては後述します。Oracleのインストールは、ネットワーク・デバイスとストレージ・デバイスの構成が終了した後に独立した手順として実施します。

Oracle Solarisのインストール

Oracle SPARCシステムにはOracle Solarisがプリインストールされており、Oracle Solaris 11にはOracle VM Server for SPARCがあらかじめ含まれています。適切なリリースのOracle Solaris OSとOracle VM Server for SPARCがサーバーにインストールされており、必要なパッチが適用されていることを確認してください。

独自のインストール・ポリシーと要件に適合したシステムになるように、システム全体を再インストールすることもできます。SPARCサーバーにOracle Solaris OSをインストールする方法について詳しくは、Oracle Solaris OSのインストール・ドキュメントを参照してください。

Oracle Solaris OSをインストールしたら、論理ドメインを使用するようにシステムを構成できます。

Oracle VM Server for SPARCのインストール

この例は、Oracle Solaris 11 OSと必要なパッチがすでにインストールされていることを前提としています。Oracle VM Server for SPARCソフトウェアは、Oracle Solaris 11 OSにデフォルトで含まれているため、インストールする必要はありません。論理ドメインのインストールに関する全手順は、『Oracle VM Server for SPARCインストールガイド』（http://docs.oracle.com/cd/E64668_01/html/E64657/index.html）を参照してください。

制御ドメインの構成

この例では、制御ドメインおよびサービス・ドメインとして使用するルート・コンプレックス・ドメイン、およびセカンダリ・サービス・ドメインとして使用するセカンド・ルート・コンプレックス・ドメインの作成方法を、SPARC T5-8を使用して説明します。最初に再構成した場合は、インストール済みのSolarisインスタンスが制御ドメインになります。再構成の内容は、デフォルト・サービスの定義と、制御ドメインの再サイジングです。サイジングでは、後で定義する他のドメインに使用できるようにCPUリソースとメモリ・リソースを解放します。また、制御ドメイン用に保持するバス、およびルート・コンプレックス・ドメインに割り当てるバスの識別も行います。

次に示すように、システムはCPUリソース、I/Oリソースおよびメモリ・リソースをすべて所有する1つの制御ドメインが構成された状態で納品されます。簡潔にするために、一部は省略されています。

```
primary# ldm list -l
NAME             STATE    FLAGS   CONS    VCPU  MEMORY   UTIL   NORM   UPTIME
primary          active   -n-c--  UART    1024  1047296M 0.0%   0.0%   2d 5h 11m
----省略----
CORE
    CID    CPUSET
    0      (0, 1, 2, 3, 4, 5, 6, 7)
    1      (8, 9, 10, 11, 12, 13, 14, 15)
    2      (16, 17, 18, 19, 20, 21, 22, 23)
    3      (24, 25, 26, 27, 28, 29, 30, 31)
----省略----
    124    (992, 993, 994, 995, 996, 997, 998, 999)
```

```
        125     (1000, 1001, 1002,   1003, 1004,   1005, 1006,   1007)
        126     (1008, 1009, 1010,   1011, 1012,   1013, 1014,   1015)
        127     (1016, 1017, 1018,   1019, 1020,   1021, 1022,   1023)
VCPU
        VID     PID     CID     UTIL    NORM STRAND
        0       0       0       4.7%    0.2%   100%
        1       1       0       1.3%    0.1%   100%
        2       2       0       0.2%    0.0%   100%
        3       3       0       0.1%    0.0%   100%
----省略----
        1020    1020    127     0.0%    0.0%   100%
        1021    1021    127     0.0%    0.0%   100%
        1022    1022    127     0.0%    0.0%   100%
        1023    1023    127     0.0%    0.0%   100%
----省略----
IO
        DEVICE                  PSEUDONYM               OPTIONS
        pci@300                 pci_0
        pci@340                 pci_1
        pci@380                 pci_2
        pci@3c0                 pci_3
        pci@400                 pci_4
        pci@440                 pci_5
        pci@480                 pci_6
        pci@4c0                 pci_7
        pci@500                 pci_8
        pci@540                 pci_9
        pci@580                 pci_10
        pci@5c0                 pci_11
        pci@600                 pci_12
        pci@640                 pci_13
        pci@680                 pci_14
        pci@6c0                 pci_15
----省略----
```

基本的なドメイン構成

次のコマンドが、仮想ディスク・サービス、コンソール・サービスおよびネットワーク・サービスの定義と制御ドメインのサイズ変更を行うための基本的な構成手順です。全体像が分かるように手順は省略せずに表示していますが、可用性に関する構成については一部省略しています。この例では、仮想スイッチにnet0を使用するものとします。これはSolaris 11の制御ドメインであるため、デバイス・ドライバではなくネットワーク・インタフェースのバニティ名（net0）を使用します。

```
primary# ldm add-vds primary-vds0 primary

primary# ldm add-vcc port-range=5000-5100 primary-vcc0 primary

primary# ldm add-vswitch net-dev=net0 primary-vsw0 primary

primary# ldm set-core 2 primary

primary# svcadm enable vntsd

primary# ldm start-reconf primary

primary# ldm set-mem 16g primary

primary# shutdown -y -g0 -i6
```

これが標準的な制御ドメイン構成です。リブートすると制御ドメインのサイズが変更されます。サービス・プロセッサに構成を保存すると、電源のオン/オフを行っても変更が持続します。

```
primary# ldm list

NAME          STATE  FLAGS  CONS    VCPU   MEMORY UTIL   NORM   UPTIME

primary       active -n-cv- UART    16     16G    3.3%   2.5%   4m

primary# ldm add-spconfig initial
```

電源のオン/オフを行っても変更が持続するように、構成は必ずサービス・プロセッサに保存してください。また、他のシステムにエクスポートできるように、XMLとして保存してください。詳しくは、http://docs.oracle.com/cd/E64668_01/html/E64643/configurationmanagement.html を参照してください。

ルート・コンプレックス・ドメインに割り当てるバスの決定

次は、制御ドメインで保持する必要があるバスと他のルート・ドメインに割り当て可能なバスを特定します。この手順については、『Oracle VM Server for SPARC 管理ガイド』の「PCIeバスの割り当てによってルート・ドメインを作成する方法」を参照してください。
http://docs.oracle.com/cd/E64668_01/html/E64643/configurepciexpressbusesacrossmultipleldoms.html

まず、ルート・プール・ディスク（本番環境ではこれはミラー化される場合があります）に使用されているPCIeバスを識別します。そのために、デバイス名を取得し、次にmpathadmコマンドを使用してイニシエータ・ポート名を取得し、そのデバイス・パスを確認します。

```
primary # zpool status rpool
```

```
 pool: rpool
state: ONLINE
 scan: none requested
config:
      NAME                        STATE    READ     WRITE    CKSUM
      rpool                       ONLINE   0        0        0
      c0t5000CCA01605A11Cd0s0     ONLINE   0        0        0
errors: No known data errors
primary # mpathadm show lu /dev/rdsk/c0t5000CCA01605A11Cd0s0
Logical Unit: /dev/rdsk/c0t5000CCA01605A11Cd0s2
----省略----
       Paths:
              Initiator Port Name: w508002000145d1b1
----省略----
primary # mpathadm show initiator-port w508002000145d1b1
Initiator Port:     w508002000145d1b1
       Transport Type:     unknown
       OS Device File:
/devices/pci@300/pci@1/pci@0/pci@4/pci@0/pci@c/scsi@0/iport@1
```

その結果、ブート・ディスクはバスpci@300上にあることが分かります。これは、疑似デバイスpci_0に対応しています。

次に、ネットワークに使用するバスを特定します。（ixgbe0をベースとする）インタフェースnet0はプライマリ・インタフェースであり、仮想スイッチはこれでホストされるため、このインタフェースが取り付けられているバスを保持する必要があります。

```
primary# ls -l /dev/ix*
lrwxrwxrwx 1 root root                  31 Jun 21 12:04 /dev/ixgbe ->
../devices/pseudo/clone@0:ixgbe
lrwxrwxrwx 1 root root                  65 Jun 21 12:04 /dev/ixgbe0 ->
../devices/pci@300/pci@1/pci@0/pci@4/pci@0/pci@8/network@0:ixgbe0
lrwxrwxrwx 1 root root                  67 Jun 21 12:04 /dev/ixgbe1 ->
../devices/pci@300/pci@1/pci@0/pci@4/pci@0/pci@8/network@0,1:ixgbe1
lrwxrwxrwx 1 root root                  65 Jun 21 12:04 /dev/ixgbe2 ->
../devices/pci@6c0/pci@1/pci@0/pci@c/pci@0/pci@4/network@0:ixgbe2
lrwxrwxrwx 1 root root                  67 Jun 21 12:04 /dev/ixgbe3 ->
../devices/pci@6c0/pci@1/pci@0/pci@c/pci@0/pci@4/network@0,1:ixgbe3
```

これを、障害管理アーキテクチャ（Fault Management Architecture：FMA）の障害プロキシに使用するデバイス/dev/usbecmに対しても繰り返す必要があります。通常、同じバス上にあるため、バスの割当てに違いはありませんが、SPARC T5-2 および SPARC M5-32 または M6-32 サーバー・モデルでは制御ドメイン用に追加のバスを保持する必要があります。詳細については、SPARC T5-2システムのユーザーはDoc ID 1948940.1のMOS Noteを、SPARC M5-32またはSPARC M6-32のユーザーはDoc ID 1942045.1のMOS Noteを参照してください。

この場合、ディスクとネットワークはバスpci@300（pci_0）上にあり、代替サービス・ドメインに提供できるネットワーク・デバイスはpci@6c0（pci_15）上にあります。

このサービス・ドメインにディスク・アクセスを提供するために必要なバスも特定します。先ほど、制御ドメインのルート・プールはpci@300上のc0t5000CCA01605A11Cd0s0にあることを確認しました。現在のところ、制御ドメインからすべてのバスおよびデバイスにアクセスできるため、他にどのディスクが使用できるかを、formatコマンドを使用して確認できます。2番目のディスクが、バスpci@6c0上にあります。

```
primary# format

Searching for disks...done

AVAILABLE DISK SELECTIONS:

        0. c0t5000CCA01605A11Cd0 <HITACHI-H109060SESUN600G-A244 cyl 64986
        alt 2 hd 27

sec 66>

        /scsi_vhci/disk@g5000cca01605a11c

        /dev/chassis/SPARC_T5-8.1239BDC0F9//SYS/SASBP0/HDD0/disk

        1. c0t5000CCA016066100d0 <HITACHI-H109060SESUN600G-A244 cyl 64986
        alt 2 hd 27

sec 668>

        /scsi_vhci/disk@g5000cca016066100

        /dev/chassis/SPARC_T5-8.1239BDC0F9//SYS/SASBP1/HDD4/disk

Specify disk (enter its number):^C


primary# mpathadm show lu /dev/dsk/c0t5000CCA016066100d0s0

Logical Unit: /dev/rdsk/c0t5000CCA016066100d0s2

----省略----

Paths:

       Initiator Port Name:       w508002000145d1b0

----省略----

primary# mpathadm show initiator-port w508002000145d1b0

Initiator Port:     w508002000145d1b0

       Transport Type:     unknown
```

```
       OS Device File:
/devices/pci@6c0/pci@1/pci@0/pci@c/pci@0/pci@c/scsi@0/iport@1
```

この時点で、バスの再割当てに必要な情報をすべて入手できました。

制御ドメインのブート・ディスクとプライマリ・ネットワーク・デバイスはpci@300 (pci_0) 上にあります。

未使用のディスクとネットワーク・デバイスはpci@6c0 (pci_15) 上にあり、pci@6c0は他のドメインに使用できます。

これで、セカンダリ・サービス・ドメインを定義できる状態になったので、上記のバスを制御ドメインから削除し、代替サービス・ドメインとして使用するルート・コンプレックス・ドメインに割り当てます。バスを削除すると、もう一度リブートする必要がありますが、制御ドメインのメモリ・サイズを変更するためにリブートする前にバス情報を取得しておくことで回避することもできます。

```
primary# ldm add-dom secondary
primary# ldm set-core 2 secondary
primary# ldm set-mem 16g secondary
primary# ldm start-reconf primary
primary# ldm rm-io pci_15 primary
primary# init 6
```

制御ドメインをリブートした後、未割当てのバスpci_15をセカンダリ・ドメインに割り当てます。この時点で、ネットワーク・インストール・サーバーを使用してこのドメインにSolarisをインストールすることができます。または、制御ドメインの仮想ディスク・サービスでエクスポートした.isoファイル内にあるブート・メディアからSolarisをインストールすることもできます。サービス・ドメインで仮想I/Oデバイスを使用すると他のサービス・ドメインへの依存性が発生するため、通常はそのようなことをしませんが、今回はSolarisをインストールした後で仮想デバイスを削除するため、容認します。

```
primary# ldm add-io pci_15 secondary
primary# ldm bind secondary
primary# ldm start secondary
```

pci@6c0上のネットワーク・デバイスは、サービス・ドメインではixgbe0としてリストされます。

```
secondary# ls -l /dev/ixgb*
lrwxrwxrwx    1 root       root          31 Jun 21 10:34 /dev/ixgbe ->
../devices/pseudo/clone@0:ixgbe
lrwxrwxrwx    1 root       root          65 Jun 21 10:34 /dev/ixgbe0 ->
../devices/pci@6c0/pci@1/pci@0/pci@c/pci@0/pci@4/network@0:ixgbe0
lrwxrwxrwx    1 root       root          67 Jun 21 10:34 /dev/ixgbe1 ->
../devices/pci@6c0/pci@1/pci@0/pci@c/pci@0/pci@4/network@0,1:ixgbe1
```

冗長サービスの定義

これで、専用のPCIeバス上で個別にブートして実行できるルートI/Oドメインができました。残っている作業は、冗長ディスク・サービスと冗長ネットワーク・サービスを定義し、先ほど制御ドメインで定義したサービスと対応させることだけです。

```
primary# ldm add-vds secondary-vds0 secondary

primary# ldm add-vsw net-dev=net0 secondary-vsw0 secondary

primary# ldm add-spconfig mysecondary
```

ゲスト・ドメイン

制御ドメインおよびセカンダリ・サービス・ドメインが提供するサービスを使用するゲスト・ドメインを作成できます。これを説明するために、IPマルチパス（IPMP）の使用を可能にする2つのネットワーク・デバイスを持ち、それぞれに制御ドメインと代替ドメインの両方からのパスが指定された2つの仮想ディスクを持つゲスト・ドメインを定義します。

```
primary# ldm add-dom ldg1

primary# ldm set-core 16 ldg1

primary# ldm set-mem 64g ldg1

primary# ldm add-vnet linkprop=phys-state ldg1net0 primary-vsw0 ldg1

primary# ldm add-vnet linkprop=phys-state ldg1net1 secondary-vsw0 ldg1

primary# ldm add-vdsdev mpgroup=ldg1group /dev/dsk/xxxxx ldg1disk0@primary-vds0

primary# ldm add-vdsdev mpgroup=ldg1group /dev/dsk/xxxxx ldg1disk0@secondary-vds0

primary# ldm add-vdisk ldg1disk0 ldg1disk0@primary-vds0 ldg
```

上の`/dev/dsk/xxxxx`はLUNへのデバイス・パスに置き換えます。これらは同じLUNを参照する、各ドメインからのパスである必要があります。

仮想ネットワーク・デバイスの定義では、`linkprop=phys-state`を指定しています。これは、仮想ネットワーク・デバイスでサービス・ドメインのリンク障害を検出してフェイルオーバーを実行できるように、物理リンクの状態変化を仮想デバイスに渡す必要があることを意味します。また、それぞれの仮想ディスクに`mpgroup`を指定しています。これは、制御ドメインとセカンダリ・ドメインのいずれからでも各ディスクにアクセスできるように、各ディスクへのパスを2つ作成しています。

この時点では、特別な手順を踏まなくてもゲスト・ドメインにSolarisをインストールできます。

耐障害性の構成およびテスト

マルチパス・ディスクI/Oはゲスト・ドメインに対して透過的です。これをテストするために、制御ドメインまたはセカンダリ・サービス・ドメインを順番にリブートし、大きな影響もなくディスクI/O操作が継続されることを観察します。

ネットワークを冗長化するには、ゲスト・ドメインでIPマルチパス（IPMP）を構成する必要があります。ゲスト・ドメインには2つのネットワーク・デバイス（制御ドメインが提供するnet0とセカンダリ・ドメインが提供するnet1）があります。次のコマンドをゲスト・ドメインで実行して、ネットワーク接続を冗長化します。

```
ldg1# ipadm create-ipmp ipmp0

ldg1# ipadm add-ipmp -i net0 -i net1 ipmp0

ldg1# ipadm create-addr -T static -a 10.134.116.224/24 ipmp0/v4addr1

ldg1# ipadm create-addr -T static -a 10.134.116.225/24 ipmp0/v4addr2

ldg1# ipadm show-if

IFNAME        CLASS         STATE         ACTIVE        OVER

lo0           loopback      ok            yes           --

net0          ip            ok            yes           --

net1          ip            ok            yes           --

ipmp0         ipmp          ok            yes           net0 net1

ldg1# ipadm show-addr

ADDROBJ             TYPE          STATE         ADDR

lo0/v4              static        ok            127.0.0.1/8

ipmp0/v4addr1       static        ok            10.134.116.224/24

ipmp0/v4addr2       static        ok            10.134.116.225/24

lo0/v6              static        ok            ::1/128
```

耐障害性のテストとしては、セカンダリ・サービス・ドメインと制御ドメインを一度に1つずつリブートし、ゲスト・ドメインへのネットワーク・セッションに何も影響がないことを確認します。1つのリンクで障害が発生したときと、それがリストアされたときに、ゲスト・ドメインのコンソールに次のようなメッセージが表示されました。

```
Jul  9   10:35:51  ldg1   in.mpathd[107]:  The link has gone down on net1

Jul  9   10:35:51  ldg1   in.mpathd[107]:  IP interface failure detected on
net1 of group ipmp0

Jul  9   10:37:37  ldg1   in.mpathd[107]:  The link has come up on net1
```

サービス・ドメインの1つが停止しているときにゲスト・ドメインでdladmとipadmを実行したところ、リンク・ステータスは次のように表示されました。

```
ldg1# ipadm show-if
IFNAME        CLASS         STATE         ACTIVE        OVER
lo0           loopback      ok            yes           --
net0          ip            ok            yes           --
net1          ip            failed        no            --
ipmp0         ipmp          ok            yes           net0 net1
ldg1#  dladm show-phys
LINK              MEDIA           STATE     SPEED      DUPLEX        DEVICE
net0              Ethernet        up        0          unknown       vnet0
net1              Ethernet        down      0          unknown       vnet1
ldg1#  dladm show-link
LINK          CLASS         MTU    STATE         OVER
net0          phys          1500   up            --
net1          phys          1500   down          --
```

サービス・ドメインのリブートが終了すると、"down"だったステータスが"up"に戻りました。システムが停止することはまったくなく、リストアされたパスを有効化し直すための手動作業も不要でした。

結論

Oracle VM Server for SPARCは、1つの物理システム上に複数の仮想システムを作成することを可能にする仮想テクノロジーです。同じ物理サーバー上に複数の論理ドメインを構成することで、コストを削減できるうえ、運用時の俊敏性と効率性が向上します。

本書では、高いパフォーマンスと可用性の実現を目的としたOracle VM Server for SPARCの構成方法のベスト・プラクティスを紹介しました。アプリケーション要件やシステム要件に応じて、複数のオプションを利用できます。導入・構築の計画に役立つように、構成ガイドラインとソフトウェア要件についても説明しました。

追加情報

タイトル	URL
オラクルの仮想化	http://www.oracle.com/jp/technologies/virtualization/overview/index.html
Oracle Technology NetworkのOracle VM関連情報	http://www.oracle.com/technetwork/jp/server-storage/vm/overview/index.html
Oracle VM Server for SPARCのテクニカル・ホワイト・ペーパー	http://www.oracle.com/technetwork/jp/server-storage/vm/documentation/logical-domains-articles-1968330-ja.html
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	http://www.oracle.com/technetwork/jp/articles/systems-hardware-architecture/vmsrvrsparc-reliability-163931-ja.pdf
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	*http://www.oracle.com/technetwork/jp/articles/systems-hardware-architecture/vmsrvrsparc-availability-163930-ja.pdf*
How to Get the Best Performance from Oracle VM Server for SPARC	*http://www.oracle.com/technetwork/articles/servers-storage-admin/solaris-network-vm-sparc-2275380.html*
SPARCサーバー・システムのドキュメント	*http://www.oracle.com/technetwork/jp/server-storage/sun-sparc-enterprise/documentation/index.html*
Oracle SPARC Serverのホワイト・ペーパー	*http://www.oracle.com/technetwork/jp/server-storage/sun-sparc-enterprise/documentation/index.html*

お問い合わせ窓口

Oracle Digital

TEL 0120-155-096
URL oracle.com/jp/contact-us

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Hardware and Software, Engineered to Work Together