



Oracleホワイト・ペーパー
2014年3月

Oracle VM Server for SPARCでのOracle Real Application Clustersの実行

はじめに	1
Oracle VM Server for SPARCでのOracle RACの実行.....	2
Oracle VM Server for SPARCの概要	3
Oracle RACノード	5
デプロイメント・オプション	5
ソフトウェア要件	6
Oracle VM Server for SPARCの要件	6
Oracle Database	6
ハードウェア構成	6
サーバー構成	9
構成のガイドライン	9
構成の詳細	9
ソフトウェアのインストールと構成	11
Oracle Solarisのインストール	11
Oracle VM Server for SPARCのインストール.....	12
制御ドメインの構成	12
ゲスト・ドメイン	14
ゲスト・ドメインの構成	15
ネットワーク構成	17
ネットワーク・レイアウト	17
パブリック・ネットワーク.....	17
プライベート・ネットワーク	18
ネットワーク・インタフェースのサマリー	19
ネットワーク・インタフェースとホスト名.....	20
プライベート・ネットワーク構成	22
ストレージ構成	23
共有ディスクと仮想ディスク	23
ゲスト・ドメインへの共有ディスクの追加.....	24
Oracleのインストール	25
クラスタ・インターコネクトに対するIPMPの構成.....	25
追加情報	26
NTP	26

CPUの動的な再構成	27
メモリの動的な再構成	27
パフォーマンスに関する考慮事項	28
ジャンボ・フレーム	29
まとめ	30
付録	31
Logical Domains Managerの構成例	31
1番目の制御ドメイン (rac01) の構成	31
1番目のゲスト・ドメイン (ldom1) の構成	34
2番目の制御ドメイン (rac02) の構成	37
2番目のゲスト・ドメイン (ldom2) の構成	40
参考資料	43

はじめに

このホワイト・ペーパーでは、Oracle VM Server for SPARC® (旧称Sun Logical Domains) で構成したOracleサーバー上でのOracle® Real Application Clusters (Oracle RAC) の実行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利用すると、1つの物理システム上に複数の仮想システムを作成できるため、Oracle RACワークロードに対してCPUとメモリのリソースをきめ細かく割り当てることができます。このソリューションを、システム当たり最大3,072CPUスレッドを処理するOracle SPARCサーバー上にデプロイすると、開発環境と本番環境の両方に対応した強力なプラットフォームが提供されます。開発環境では、1つの物理サーバー上に複数のOracle RACノードをデプロイしてハードウェア・コストを削減し、本番環境では物理サーバーを共有するか、または各Oracleノードを別々の物理サーバーにデプロイして可用性を向上できます。

このホワイト・ペーパーでは、次のトピックについて説明します。

- “Oracle VM Server for SPARCでのOracle Real Application Clustersの実行”ではOracle RACとOracle VM Server for SPARCの概要とともに、このソリューションで使用できる各種のデプロイメント・オプションについて説明します。
- “ソフトウェア要件”ではソフトウェアとファームウェアの要件を示します。“ハードウェア構成”では構成例で使用されたハードウェアについて説明します。
- “ソフトウェアのインストールと構成”、“ネットワーク構成”、“ストレージ構成”では、構成例のインストールと構成に必要な手順について説明します。
- “追加情報”では、パフォーマンスに関する考慮事項と既知の問題を含む補足情報を提供します。

Oracle RAC のインストールの詳細な例については、次のホワイト・ペーパーを参照してください。
[『Installing Oracle RAC 11gR2 on the Oracle Solaris 11 OS by Using Oracle VM Server for SPARC.』](#)

Oracle VM Server for SPARCでのOracle RACの実行

Oracle Real Application Clusters (Oracle RAC) は評価の高いOracle Database Enterprise Editionの関連製品です。Oracle RACは、従来のシェアード・ナッシング構成や共有ディスク構成の限界を克服した共有キャッシュ・アーキテクチャを基盤とするクラスタ・データベースであり、すべてのビジネス・アプリケーションに対応する高い拡張性と可用性を備えたデータベース・ソリューションを実現します。Oracle RACは、Oracleエンタープライズ・グリッド・アーキテクチャの主要コンポーネントです。

Oracle RACはOracle Clusterwareを活用して、クラスタ化されたデータベース環境に必要なノード間通信を実行します。Oracle Clusterwareは、サーバー・ファームをクラスタに変換するテクノロジーです。一般的なクラスタとは、単一のシステムとして共同で機能する個別サーバーのグループです。Oracle Clusterwareは、このシステムに必要な連携を実現するためのインテリジェント機能であり、Oracleエンタープライズ・グリッド・アーキテクチャの主要コンポーネントでもあります。

一般的なOracle RACのインストールでは、Oracle Automatic Storage Management (Oracle ASM) が基本的なクラスタ・ボリューム・マネージャとしての役割を果たします。Oracle ASMは、すべてのサーバーとストレージ・プラットフォームに共通する単純なストレージ管理インタフェースをデータベース管理者に提供します。また、Oracleデータベース・ファイル専用の垂直に統合されたファイル・システムとボリューム・マネージャとして、Raw I/Oのパフォーマンスとファイル・システムの簡単な管理を実現します。さらに、Oracleエンタープライズ・グリッド・アーキテクチャにおける共有ストレージ・プールの基盤となります。

Oracle VM Server for SPARCは、SPARC TシリーズやSPARC Mシリーズなどの幅広い[SPARCサーバー](#)でサポートされる、仮想化およびパーティション化のソリューションです。

SPARC Mシリーズ・サーバーは任意で複数の物理ドメイン (“PDom”) に分割でき、それぞれで独立したOracle VM Server for SPARC環境をホストできます。Oracle VM Server for SPARCを使用すると、1つの物理システムまたは物理ドメイン上に複数の仮想システムを作成できます。各仮想システムは論理ドメインと呼ばれ、それぞれ独自のOracle Solarisオペレーティング・システム・コピーを実行します。最大3,072個の仮想CPUと32TBの物理メモリで構成されたOracle SPARCサーバーは強力なシステムであり、このサーバーに論理ドメインを構成することで、簡単に複数の物理サーバーを1つのプラットフォームに統合および仮想化できます。

Oracle VM Server for SPARC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。

Oracle DatabaseとOracle Real Application Clusters (Oracle RAC) は業界有数のデータベース・アプリケーションであり、各種のサーバーで頻繁に使用されています。仮想化と統合を利用することで、Oracle VM Server for SPARCなどの特定の認定仮想環境で仮想デバイスを使用して、Oracle DatabaseとOracle RACを実行できるようになりました。また、複数のOracle DatabaseサーバーやOracle RACノードを同じ物理プラットフォームにデプロイできるため、データベースのデプロイメントに新たな方法が加えられました。

Oracle VM Server for SPARCの概要

Oracle VM Server for SPARCでは、論理ドメインにその特性を定義したロールが設定されています。ドメイン・ロールはドメインがどのように構成されているか、どのリソースを所有しているか、どのようなサービスを提供しているかによって決まります。1つのドメインには、1つまたは複数のロールを設定できます。Oracle環境に必要であり、かつサポートされている適切なデバイスとリソースをドメインが提供する限り、Oracle RACノードはどのようなロールを持つドメインにもデプロイできます。

論理ドメインには次のうちのいずれか1つ、または複数のロールを割り当てることができます。

- **I/Oドメイン** – I/Oドメインは、一部の物理I/Oリソース（物理ディスクや物理ネットワーク・インタフェースなど）に直接アクセスできます。I/OドメインにはダイレクトI/O（DIO）機能、またはシングル・ルートI/O仮想化（SR-IOV）機能を使用して、PCIeバスまたはPCIeスロット、もしくはオンボードのPCIeデバイスを構成できます。I/Oドメインで実行され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）オペレーティング・システム・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。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。同様に、I/OドメインでのOracle RACの実行および構成は、その他のOracle SPARCサーバーでのOracle RACの実行および構成と何ら変わりありません。I/Oドメインは物理I/Oデバイスに直接アクセスできるため、最適なI/Oパフォーマンスが提供されます。ただし、単一プラットフォーム上に作成できるI/Oドメインの数は、そのプラットフォームで利用できるI/Oリソースによって制限されます。この制限は、サーバー上のPCIeバスおよびPCIeデバイスの数とSR-IOV仮想機能によって決まります。結果的に、作成できるI/Oドメインの数は限られます。次に説明するように、通常I/Oドメインは、サービス・ドメインでもあり、これによりその物理I/Oデバイスが仮想デバイスとして他のドメインで利用できるようになります。
- **ルート・ドメイン** – ルート・ドメインは、PCIeバス（別名PCIeルート・コンプレックス）を割り当てられたドメインです。また、ルート・ドメインは物理I/Oデバイスを保有しており、直接アクセスできるため、I/Oドメインでもあります。1つのプラットフォーム上に作成できるルート・ドメインの数は、プラットフォーム上で利用できるPCIeバスの数によって制限されます。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。ルート・ドメインはI/Oドメインでもあるため、I/Oドメインでの実行と同じようにルート・ドメインでOracle RACを実行できます。

ドメインには次のうちのいずれか1つ、または複数のロールを割り当てることができます。

- I/Oドメイン
- ルート・ドメイン
- 制御ドメイン
- サービス・ドメイン
- ゲスト・ドメイン

I/OドメインでOracle RACを実行するとパフォーマンスが向上しますが、1台のサーバーに対して作成できるI/Oドメイン数には制限があります。また、制御ドメインでOracle RACを実行することは、セキュリティ上の懸念から一般に推奨されていません。サービス・ドメインでOracle RACを実行することも、パフォーマンス上の理由から、仮想デバイス・サービスへ悪影響を与える可能性を回避するため、一般には推奨されていません。Oracle RACはゲスト・ドメインで実行でき、仮想デバイスを使用できます。

- **制御ドメイン** - 制御ドメインは、サーバーの電源を入れると最初に起動されるドメインであり、Logical Domains Managerを実行するドメインです。Logical Domains Managerは、プラットフォーム上のすべてのドメインを構成および管理するために使用されます。制御ドメインには「プライマリ」という名前が付けられており、物理I/Oの起動を必要とすることから、ルート・コンプレックスのI/Oドメインでもあります。そのため、I/Oドメインでの実行と同じように、制御ドメインでOracle RACを実行できます。次に説明するように、通常、制御ドメインは、サービス・ドメインでもあり、他のドメインで利用可能にすることで、その物理I/Oリソースを活用できます。ただし、管理上およびセキュリティ上の懸念から、通常、制御ドメインでOracle RACを実行することは推奨されていません。

制御ドメインはハイパーバイザに対して特権的接続をすることで、その他のドメインを制御します。制御ドメインのセキュリティは入念に保護する必要があります。制御ドメインのセキュリティが侵害され、悪意あるユーザーが制御ドメインの特権アクセスを獲得すると、プラットフォーム上のその他すべてのドメインを支配できる可能性があります。セキュリティ・リスクを防止するため、制御ドメイン内で実行されるすべてのアプリケーションを正しく構成する必要があります。

- **サービス・ドメイン** - サービス・ドメインは仮想ディスク・サービスや仮想ネットワーク・サービスを他のドメインに提供するドメインです。また、通常サービス・ドメインは、I/Oドメインでもあることから、提供する仮想I/Oデバイスの基盤となる物理デバイスへのアクセスを得られます。そのため、I/Oドメインでの実行と同じように、サービス・ドメインでOracle RACを実行できます。ただし、パフォーマンス上の理由と、仮想デバイス・サービスへの悪影響を回避するため、サービス・ドメインでOracle RACを実行することは、一般には推奨されていません。

サービス・ドメインはCPU、メモリ、I/Oといったリソースをいくらか消費して、その他のドメインに仮想I/Oサービスを提供します。したがって、その他のドメインに提供されるサービスによるワークロードと、このドメイン自体で実行されるアプリケーション（Oracle RACなど）によって生成されるすべてのワークロードを十分に処理できるリソースをサービス・ドメインに構成する必要があります。

また、Oracle RACによってその実行システムがリブートされる場合があります。サービス・ドメインがリブートされても、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、リセットされたりすることはありません。代わりに、サービス・ドメインが停止している間、これらのゲスト・ドメインからのI/Oリクエストは一時停止され、サービス・ドメインが再び稼働すると自動的に再開されます。代替の冗長サービス・ドメインが構成されている場合は、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を続行できます。Oracle RACをサービス・ドメインで実行する際は、Oracle RACによってサービス・ドメインがリブートされる可能性があり、この場合はこのサービス・ドメインを使用しているゲスト・ドメインのI/Oリクエストが一時的に停止されることに留意してください。そのような理由から、Oracle RACをサービス・ドメインで実行することは、そのクライアントのゲスト・ドメインが別のサービス・ドメインの冗長パスからI/Oにアクセスできない限り、推奨されません。

- **ゲスト・ドメイン** - ゲスト・ドメインはI/Oドメインではなく、1つまたは複数のサービス・ドメインによって提供される仮想デバイス・サービスのコンシューマです。ゲスト・ドメインには物理的なI/Oデバイスはなく、仮想ディスクや仮想ネットワーク・インタフェースといった仮想I/Oデバイスのみが備わっています。これがドメインで実行されているアプリケーションにI/Oを提供する一般的な方法です。Oracle RACはゲスト・ドメインで実行でき、また仮想デバイスを使用できます。この構成が本書の中心になります。

Oracle RACノード

Oracle RACは、ドメインのロールに関係なくあらゆるドメイン上で実行できますが、上述の理由から、制御ドメインやサービス・ドメインでのOracle RACの実行は推奨されていません。ゲスト・ドメイン上で実行するか、またはI/Oドメイン上で実行するかがOracle RACのおもな選択肢になります。I/Oドメインは物理I/Oデバイスに直接アクセスできるため、最適なI/Oパフォーマンスを提供しますが、1つのプラットフォーム上に作成できるI/Oドメイン数は限られています。その一方で、ゲスト・ドメインのリソースやデバイスはすべて仮想化されているため、ゲスト・ドメインは柔軟性がより高く、より動的な処理がサポートされており、他のサーバーへのライブ・マイグレーションも実行可能です。

1つの構成において、I/Oドメインで実行されているOracle RACノードとゲスト・ドメインで実行されているOracle RACノードを混在させないでください。このような構成は技術的には可能ですが、設定が複雑になるため、管理が難しくなり、エラーが発生しやすくなります。Oracle RACクラスタに含まれるすべてのOracle RACノードをI/Oドメインで実行するか、またはすべてのOracle RACノードをゲスト・ドメインで実行するかのいずれかにしてください。

デプロイメント・オプション

サーバーを仮想化すると、複数の仮想マシンとオペレーティング・システムを1つの物理システム上で実行できます。これにより、同じクラスタに含まれる複数のノードを1つの物理システムでホストできます。図1に示すとおり、論理ドメインへのOracle RACのデプロイメントにはおもに次の2タイプがあります。

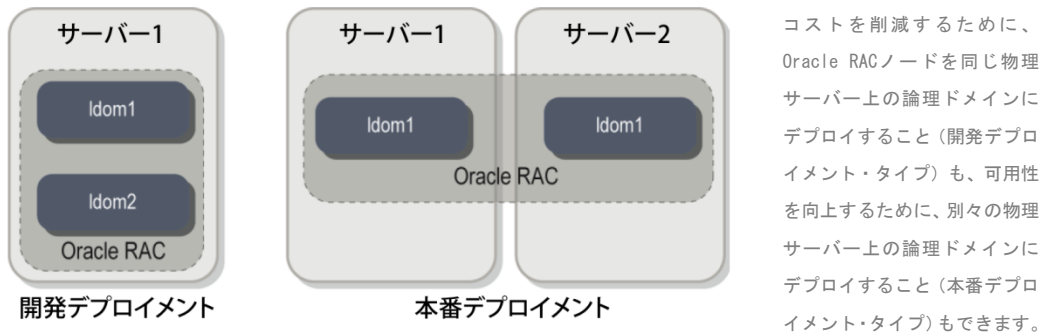


図1: デプロイメント・オプション

- 開発** - すべてのOracle RACノードが同じ物理サーバー上のドメインにデプロイされています。このデプロイメントは必要なハードウェア数が少ない（物理サーバー1台のみ）ため、開発や評価に適しています。ただし、この構成では物理サーバーがシングル・ポイント障害になるため、本番環境向けではありません。サーバーが完全に停止するとすべてのノードも停止し、Oracle RACクラスタ全体が使用できなくなります。
- 本番** - Oracle RACノードが別々の物理サーバーにデプロイされています。この構成ではノードが異なる物理サーバーにデプロイされており、シングル・ポイント障害が存在しないため、このデプロイメントは本番環境に推奨されています。その他のアプリケーションは、干渉を発生させることなく同じサーバー上の他のドメインにデプロイできます。

言うまでもなく、両方のデプロイメント・オプションを組み合わせることができます。1つの方法として、異なる物理ドメイン間にはハードウェア冗長性があるため、それぞれのOracle RACノードを同じM5-32やM6-32サーバー上の異なる物理ドメイン（“PDom”）にデプロイする方法があります。同じ物理サーバー上にデプロイされたノードはシングル・ポイント障害になり得るという点に留意してください。

ソフトウェア要件

この項では、Oracle VM Server for SPARCへOracle RACをデプロイする際のソフトウェア要件について説明します。対象ソフトウェアには、Logical Domains Manager、必要なオペレーティング・システム・バージョンとパッチ、ファームウェア、Oracle RACが含まれます。

Oracle VM Server for SPARCの要件

Oracle VM Server for SPARCでOracle RACを実行するには、バージョン2.2以降のLogical Domains Managerを使用する必要があります。このホワイト・ペーパーが執筆された時点での最新バージョンである3.1は、パフォーマンスや可用性、管理性を向上する機能を提供しています。すべてのドメインで、オペレーティング・システムはOracle Solaris 10 10/11 (Update 10)以降、またはOracle Solaris 11を使用する必要があります。

サーバーには、使用するOracle VM Server for SPARCのバージョンに対応したファームウェアをインストールする必要があります。たとえば、Oracle VM Server for SPARC 3.1リリース・ノートでの“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。このドキュメントは次のWebサイトから参照できます。
<http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html>

Oracle Database

Oracle VM Server for SPARCでOracle RACを実行するには、Oracle Database 10g Release 2 (Oracle 10.2.0.4)以降を使用し、すべてのノードに同じパッチ・セットを適用する必要があります。特定のバージョンがサポートされているかどうか、また追加要件があるかどうかを確認するには、Oracle ドキュメントを参照してください。次のOracle Support Webサイトでは、さまざまな情報が提供されています。

- <https://support.oracle.com>
- <http://www.oracle.com/technetwork/database/clustering/tech-generic-unix-new-166583.html>

ハードウェア構成

Oracle VM Server for SPARCをサポートしているすべてのサーバーで、Oracle RACを使用できます。Oracle VM Server for SPARCをサポートしているシステムの一覧については、最新の『*Oracle VM Server for SPARC*リリース・ノート』ドキュメントを参照してください。選択する構成およびデプロイメント・オプションによって、外部ストレージ・アレイやネットワーク・スイッチなどの追加ハードウェアが必要になる場合があります。

このドキュメントでは、例として2ノードのOracle RACクラスタを取り上げます。このクラスタでは、各Oracle RACノードが異なる物理サーバー上のOracle VM Server for SPARCゲスト・ドメインにデプロイされています（本番デプロイメント・タイプ）。図2に、ハードウェア構成とケーブル配線を示します。

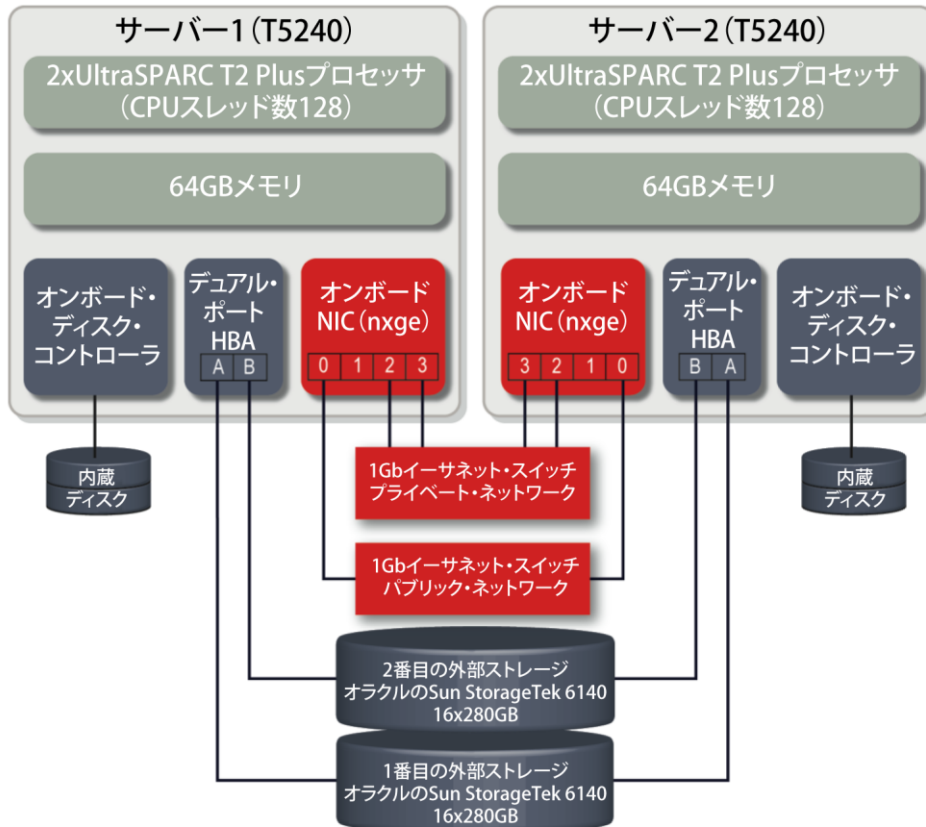


図2: ハードウェア構成

この例では、次のハードウェアが使用されています。その他のハードウェア・プラットフォームでは詳細構成が異なります。

- 128個のCPUスレッドと64GBのメモリを備えたOracle Sun SPARC Enterprise T5240サーバー2台。これはかなり旧式のサーバーで、スケーラビリティとパフォーマンスは最新のSPARCサーバーと比べ大幅に劣りますが、Oracle SolarisとOracle VM Server for SPARCの最新バージョンに対する互換性があり、引き続き、パフォーマンスが重要ではないテスト・システムの良い例となります。Sun SPARC Enterprise T5240サーバーには、2基のUltraSPARC T2 Plusプロセッサが含まれています。各UltraSPARC T2 Plusプロセッサは8個のCPUコアで構成されており、それぞれのCPUコアに8つのCPUスレッドが含まれるため、システムごとに提供されるCPUスレッド数は合計で $2 \times 8 \times 8 = 128$ になります。Oracle VM Server for SPARCソフトウェアはそれぞれのCPUスレッドを仮想CPUとして仮想化するため、128個の仮想CPUを使用してドメインを作成できます。

- サーバーごとに1つの内部ディスクが、制御ドメインのシステム・ディスクとして使用されます。Sun SPARC Enterprise T5240サーバーには、最大16の内部ディスクを搭載できます。
- 280GBのファイバ・チャネル (FC) ディスク・ドライブを16個搭載したOracle Sun StorageTek 6140ストレージ・アレイ2台。各ストレージ・アレイは両方のサーバーに接続されており、各サーバーは同じアレイ・コントローラ (AまたはB) に接続されています。1番目のストレージ・アレイはゲスト・ドメインごとに1つのローカル・ディスクを提供します。2番目のストレージ・アレイはOracle RACノードに5つの共有ディスク (投票ディスク、OCRディスク、ASMディスク) を提供します。各ボリュームを所有するのは1つのコントローラのみであるため、このボリュームを使用する各サーバーを同じ所有コントローラに接続する必要があります (たとえば、サーバーAはコントローラAのポート1に、サーバーBはコントローラAのポート2に接続)。
- 4Gbデュアル・ポートFC-ALホスト・バス・アダプタ (HBA) 2つ (サーバー1台につき1つ)。これらのHBAは2つのストレージ・アレイを両方のサーバーに接続するために使用されます。各HBAに2つのポートが搭載されています。それぞれのサーバーで、1番目のポートが1番目のストレージ・アレイに接続され、2番目のポートが2番目のストレージ・アレイに接続されます。
- 1Gbのイーサネット・スイッチ2個。1個のスイッチを使用して2台のサーバーが相互接続され、Oracle RACが使用するプライベート・ネットワークが作成されます。もう1個のスイッチも2台のサーバーに接続されますが、これはパブリック・ネットワークを提供し、必要に応じてLANやWANに接続されます。
- 各サーバー上のオンボード・ネットワーク・インタフェースはネットワーク通信に使用されます。Sun SPARC Enterprise T5240サーバーには、4つのオンボード1Gbイーサネット・ネットワーク・インタフェースが搭載されています。1番目のインタフェース (ポート0) はパブリック・ネットワークに使用され、対応するスイッチに接続されます。残りのうち2つのインタフェース (ポート2とポート3) はプライベート・ネットワークへの冗長アクセスの提供に使用され、もう一方のスイッチに接続されます。もう1つのインタフェース (ポート1) は使用されません。

この構成では、最小限のハードウェア冗長性を提供していることに注意してください。Oracle RACの最小要件を満たすため、プライベート・ネットワークへの二重接続を提供しています。ハードウェアの冗長性を向上するには、次の方法があります。

- 追加のHBAを使用して、ストレージ・アレイへの冗長アクセス・パスを提供します。
- ハードウェアまたはソフトウェアのRAIDソリューションを使用して、ディスク上のデータを複製します。
- 空いているオンボード・ネットワーク・インタフェース (ポート1) を使用して、パブリック・ネットワークへの冗長アクセス・パスを提供します。
- 追加のネットワーク・スイッチを使用して冗長プライベート・ネットワークを作成し、パブリック・ネットワークへの冗長アクセスを提供します。
- 追加のネットワーク・インタフェース・アダプタを使用して、プライベート・ネットワークとパブリック・ネットワークへアクセスする際のPCIバス・レベルの冗長性を提供します。

ネットワークとストレージの冗長性を確保する方法について、詳しくは参考資料の項を参照してください。

サーバー構成

この項では、論理ドメイン上にOracle RACを構成する際に推奨される一般的な構成ガイドラインと、本書の構成例で使用された構成の詳細情報について、説明します。

構成のガイドライン

次の構成ガイドラインは、論理ドメイン上にOracle RACを構成する際の推奨事項です。

- ・ システムが単一ドメイン（制御ドメイン）で構成されている場合、またはセキュリティおよび可用性に関する考慮事項を完全に理解している場合を除いて、Oracle RACを制御ドメインで実行することは推奨されていません。
- ・ クライアント・ゲスト・ドメインが追加のサービス・ドメイン経由でI/Oデバイスに対して冗長アクセスを保持している場合、または可用性に対する要件が緩やかな場合を除き、サービス・ドメインでのOracle RACの実行は推奨されていません。
- ・ Oracle RAC構成では、すべてのOracle RACノードをI/Oドメインで実行するか、またはすべてのノードをゲスト・ドメインで実行するかのいずれかにします。1つのOracle RAC構成において、I/Oドメインで実行されているOracle RACノードとゲスト・ドメインで実行されているOracle RACノードを混在させないでください。
- ・ Oracle RACを実行している各ドメインには、最小でも8個の仮想CPU（1CPUコア）と4GBメモリを構成する必要があります。これは説明のために、少ないリソースを割り当てています。
- ・ CPUコアはパフォーマンス上の理由からコア境界に割り当てられます。これは、CPUスレッドを8ずつ割り当てるか、CPUリソースを割り当てるためにコア全体の割当てを使用することで実施できます。
- ・ Oracle RACゲスト・ドメインに仮想デバイス（仮想ディスクや仮想ネットワーク）を提供する各サービス・ドメインには、最小でも16個のコア境界に割り当てられた仮想CPU（2CPUコア）と4GBメモリを構成する必要があります。
- ・ ゲスト・ドメインでOracle RACを実行する場合、これらのゲスト・ドメインでIPマルチパス（IPMP）を使用してIPプライベート・ネットワークの冗長性を構成する必要があります。
- ・ Oracle RACゲスト・ドメインによって共有されている仮想ディスクは、SCSIコマンドおよびマルチホスト・ディスク・コントロール機能を提供するために、物理的なSCSIディスクまたはSCSI LUNで全体をバックアップする必要があります。共有仮想ディスクをファイルやボリュームでバックアップしないでください。また、1枚のディスクを使用した共有仮想ディスクは推奨されていません。

構成の詳細

このドキュメントで使用した例では、各サーバーに2つのドメイン（制御ドメインと1つのゲスト・ドメイン）が構成されています（図3）。制御ドメインはプライマリ・ドメインでもあり、I/Oドメインでもあります。また、制御ドメインはサービス・ドメインとしても使用され、ゲスト・ドメインに仮想デバイス・サービス（仮想ディスクや仮想ネットワーク）を提供します。分かりやすくするため、“制御ドメイン”という名前を、制御、ルート、I/O、サービスというすべてのドメイン・ロールを示すために使用します。

それぞれのドメインには、表1に示したリソースが割り当てられました。

表1: 各ドメインへの割当てリソース

ドメイン	CPU	メモリ	デバイス
制御ドメイン	32x仮想CPU (4つのCPUコア)	16GB	すべてのPCIバス (2つのストレージ・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	48x仮想CPU (6つのCPUコア)	24GB	制御ドメインによって提供された仮想ディスクと仮想ネットワーク・インタフェース

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。残りの48の仮想CPU (6CPUコア) と24GBのメモリが引き続き使用可能です。リソースの追加が必要な場合 (例: より重要なワークロードを処理するため)、これらのリソースを使用して追加のゲスト・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の貴重な機能により、リソースは必要に応じて動的に割り当てられます。

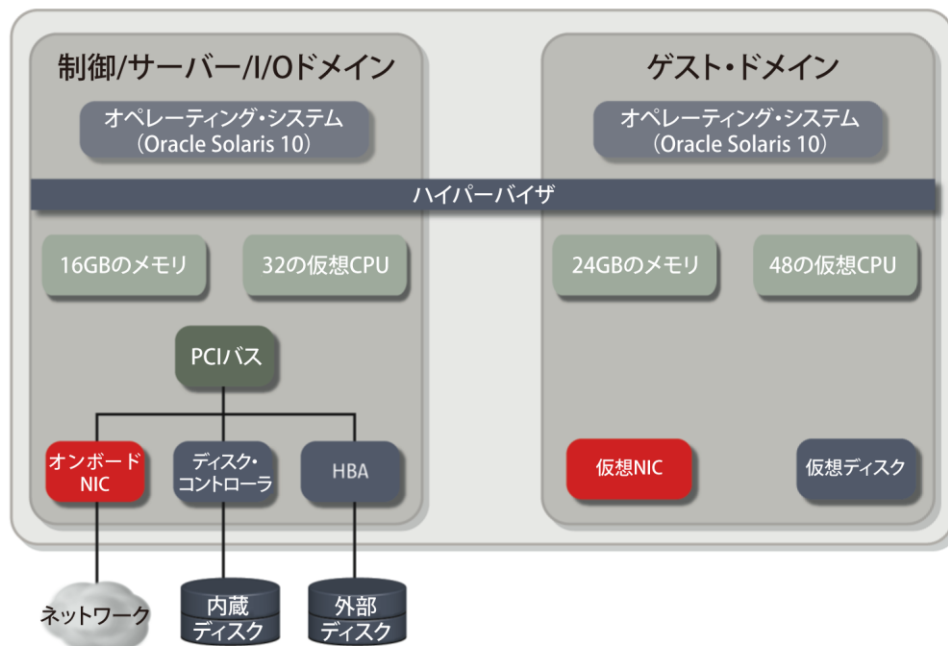


図3: 物理サーバー内のリソース割当て

サービス・ドメインとして機能するため、制御ドメインは次の仮想デバイス・サービスを持ちます。

- 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される、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される、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。

- プライマリ・ネットワーク・インタフェース (nxge0) に関連付けられた、1つの仮想スイッチ・サービス (primary-vsw0)。この仮想スイッチはOracle RACに必要とされるパブリック・ネットワークの一部になります。このインタフェースは仮想スイッチに直接接続されているため、制御ドメインからゲスト・ドメインと外部ネットワークへのネットワーク接続 (primary-vsw0に関連付けられたnxge0経由) が可能になります。
- 物理ネットワーク・インタフェース (nxge2とnxge3) に関連付けられた、2つの仮想スイッチ・サービス (primary-vsw1とprimary-vsw2)。これらの仮想スイッチはOracle RACに必要とされるプライベート・ネットワークの一部になります。

パブリック・ネットワークとプライベート・ネットワークの全体的な構成については、“ネットワーク構成”の項で説明しています。

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 (下記を参照)
- Oracle VM Server for SPARCのインストール (下記を参照)
- 制御ドメインの構成
- 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、その他の構成作業については本書の後半で説明します。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されます。

Oracle Solarisのインストール

Oracle SPARCシステムにはOracle Solaris OSがプリインストールされており、Oracle Solaris 11にはOracle VM Server for SPARCがプリインストールされています。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。

それぞれのインストール・ポリシーを遵守し、要件を満たすように、システム全体を再インストールすることもできます。その場合は、最初の手順としてOracle Solaris OSをインストールし、適切なパッチを適用します。SPARCプロセッサを搭載したサーバーへのOracle Solaris OSのインストール方法について、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。この例では、サーバーの最初の内部ディスクにオペレーティング・システムがインストールされます。

Oracle Solaris OSをインストールしたら、システムを構成し、論理ドメインの使用を有効化します。

Oracle VM Server for SPARCのインストール

論理ドメインのインストールの全手順について、詳しくは『*Oracle VM Server for SPARC管理ガイド*』を参照してください。基本的には、それぞれの物理サーバー上で次のタスクを実行します。ここでは、Oracle Solaris OSと必要なパッチがそれぞれのサーバーにすでにインストールされているとします。Oracle VM Server for SPARCソフトウェアは、デフォルトでOracle Solaris 11 OSに含まれているため、次のインストール手順は省略できることに注意してください。プリインストールされているOracle VM Server for SPARCが最新バージョンでない場合は、http://docs.oracle.com/cd/E49214_01/html/E40610/upgradingtoldoms3.1.html#LDSAGupdates11task に記載されている手順を使用します。

Oracle Solaris 10の場合

1. インストールする予定のOracle VM Server for SPARCのリリースに合ったシステム・ファームウェアがインストールされていることを確認します。『*Oracle VM Server for SPARCリリース・ノート*』を参照して適切なファームウェア・バージョンを確認し、『*Oracle VM Server for SPARC管理ガイド*』を参照してシステム・ファームウェアのアップグレード手順を確認します。
2. [オラクルのWebサイト](#)から、Oracle VM Server for SPARCソフトウェアをダウンロードします。
3. アーカイブを解凍します。
4. SUNWldm.vパッケージをインストールします。

```
# pkgadd -d OVM_Server_SPARC-2_2/Product SUNWldm.v
```

5. Logical Domains Manager (ldmd) サービスとVirtual Network Terminal Server (vntsd) サービスが有効化されていることを確認します。

```
# svcadm enable ldmd
# svcadm enable vntsd
```

制御ドメインの構成

Oracle VM Server for SPARCソフトウェアのインストールが終了したら、現在のシステムを制御ドメイン(別名プライマリ・ドメイン)として再構成します。それぞれの物理サーバー上で次のタスクを実行します。

各物理サーバー上の制御ドメインを構成します。

1. 仮想ディスク・サーバー (vds) と仮想コンソール・コンセントレータ (vcc) を追加します。

```
# ldm add-vds primary-vds0 primary
# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

2. システムのプライマリ・ネットワーク・インタフェースを決定します。通常は、最初に構成されたインタフェース (nxge0またはigb0) がこれに当たります (ただし、必ずしも最初に構成されたインタフェースというわけではありません)。制御ドメインでSolaris 11を実行している場合、ネットワーク・アダプタ・ドライバには”バニティ名” (通常はnet0) を使用します。サーバーの実際のデバイス名に応じて、次の手順で使用されているコマンドを調整します。サーバーに適切なネットワーク・デバイスとネットワーク・ドライバを選択するためには、ハードウェアとSolarisのドキュメントを確認してください。
3. プライマリ・ネットワーク・インタフェースに関連付けられた仮想スイッチを追加します。この仮想スイッチはパブリック・ネットワーク用に使用されます。

```
# ldm add-vsw net-dev=nxge0 primary-vsw0 primary
```

Oracle RACプライベート・ネットワーク用に選択された物理ネットワーク・インタフェースに関連する、2番目と3番目の仮想スイッチを追加します。プライベート・ネットワーク・インタフェースにリンク・ベースのIPMP冗長性を実装するため、両方のスイッチにlinkprop=phys-stateというオプションを設定する必要があります。

```
# ldm add-vsw net-dev=nxge2 linkprop=phys-state primary-vsw1 primary
# ldm add-vsw net-dev=nxge3 linkprop=phys-state primary-vsw2 primary
```

制御ドメインでSolaris 10を実行している場合、プライマリ・インタフェースが最初の仮想スイッチ・インタフェースになるように変更します。次のコマンドを実行すると、制御ドメインでnxge0ではなくvsw0がインタフェースとして使用されます。

```
# mv /etc/hostname.nxge0 /etc/hostname.vsw0
```

4. 制御ドメイン (プライマリ) に32個の仮想CPU (4CPUコア) と16GBのメモリを構成します。下位互換性を維持するため、set-coreの代わりにldm set-vcpu 32 primary構文を使用できますが、コア全体の割当てが推奨されることに留意して下さい。T4以前のマシンでは、ハードウェア暗号化アクセラレータ (コアごとに1つの暗号化アクセラレータ) をドメインに割り当てるため、ldm set-cryptoコマンドを使用します。このコマンドはT4以降のサーバーでは不要です。

```
# ldm set-core 4 primary
# ldm set-crypto 4 primary # 1 per core, only for servers prior to T4
# ldm start-reconf primary
# ldm set-mem 16g primary
```

5. 構成を保存してシステムをリブートします。

```
# ldm add-spconfig initial
# init 6
```

システムをリブートすると、Oracle VM Server for SPARCが有効化され、システムが制御ドメイン (プライマリ・ドメイン) として構成されます。その後で、制御ドメインから追加のドメインを作成および構成できます。

ゲスト・ドメイン

制御ドメインの構成が終了すると、Oracle RACノードとして使用されるゲスト・ドメインを作成できます。各物理サーバーに1つのゲスト・ドメインを作成します。最初のゲスト・ドメインは`ldom1`という名前で1番目のサーバー上に作成し、2番目のゲスト・ドメインは`ldom2`という名前で2番目のサーバー上に作成します。

各ゲスト・ドメイン (`ldom1`と`ldom2`) の作成時には、次のリソースが割り当てられています。

- 48個の仮想CPU (6コア)
- 6個の暗号化アクセラレータ (T4以前のシステム用) Oracle RACはSPARCハードウェアの暗号化アクセラレーションを活用できます。
- 16GBのメモリ
- 仮想スイッチ`primary-vsw0`に接続された、1つの仮想ネットワーク・インタフェース (`vnet0`)。この仮想ネットワーク・インタフェースはパブリック・ネットワークへのアクセスを提供します。
- 仮想スイッチ`primary-vsw1`と`primary-vsw2`に接続された、2つの仮想ネットワーク・インタフェース (`vnet1`と`vnet2`)。この仮想ネットワーク・インタフェースはプライベート・ネットワークへのアクセスを提供します。
- 1つの仮想ディスク (ゲスト・ドメインでは`c0d0`として表示される)、実際は1番目のストレージ・アレイのLUN。ドメイン`ldom1`はストレージ・アレイ (`c3t0d1`) のLUN1を使用し、`ldom2`は同じストレージ・アレイ (`c3t0d2`) のLUN2を使用します。この仮想ディスクはゲスト・ドメインのシステム・ディスクとして使用され、オペレーティング・システムをホストします。

構成の説明を分かりやすくするため、ゲスト・ドメインは最初1つのディスク (`c0d0`) のみで作成され、このディスクがオペレーティング・システム向けのシステム・ディスクとして使用されます。Oracle RAC構成に必要なその他のディスクは、後で追加します。ゲスト・ドメインの初期構成を図4に示します。

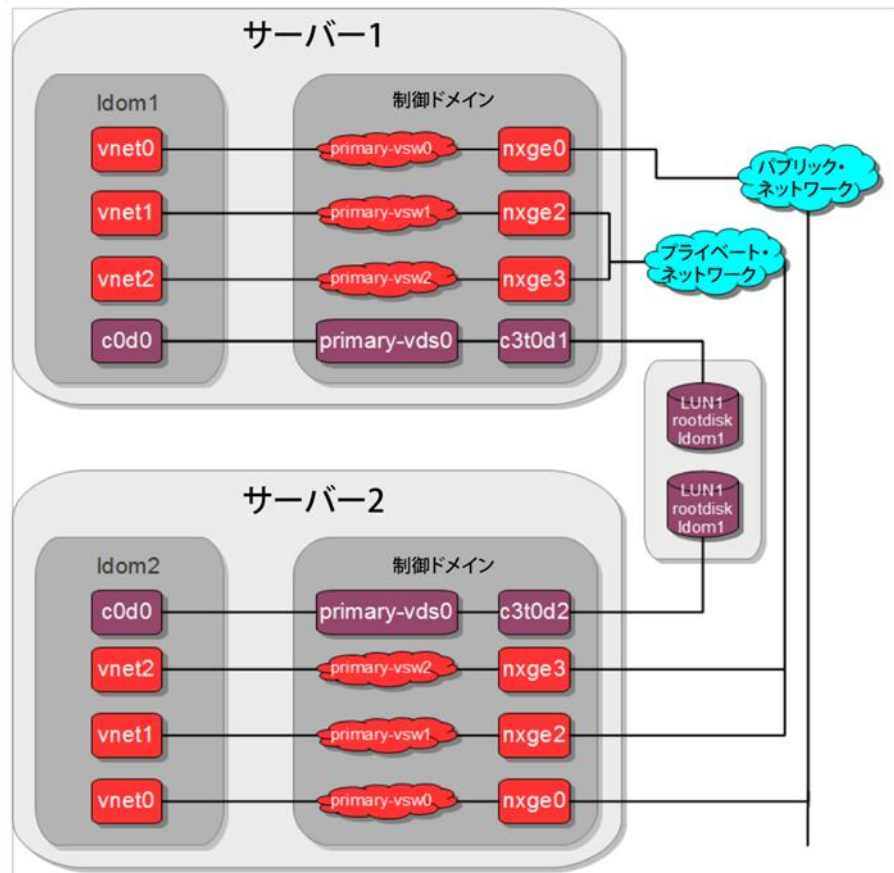


図4: ゲスト・ドメインの初期構成

ゲスト・ドメインの構成

各ゲスト・ドメインを作成するには、次のコマンドを使用します。

1. サーバー1上の制御ドメインからゲスト・ドメインldom1を作成します。この例では、コア全体の割当て(48個の仮想CPUスレッドの代わりに6個のコアを割り当てる)について、再度説明します。もう一度述べますが、`set-crypto`コマンドは、サーバーがT4の場合は不要であるため、このコマンドを省略します。

```
# ldm create ldom1
# ldm set-core 6 ldom1
# ldm set-memory 24G ldom1
# ldm set-crypto 6 ldom1
# ldm add-vnet vnet0 primary-vsw0 ldom1
# ldm add-vnet linkprop=phys-state vnet1 primary-vsw1 ldom1
# ldm add-vnet linkprop=phys-state vnet2 primary-vsw2 ldom1
# ldm add-vdsdev /dev/rdisk/c3t0d1s2 ldom1@primary-vds0
# ldm add-vdisk ldom1 ldom1@primary-vds0 ldom1
```

2. サーバー2上の制御ドメインからゲスト・ドメインldom2を作成します。

```
# ldm create ldom2
# ldm set-core 6 ldom2
# ldm set-memory 24G ldom2
# ldm set-crypto 6 ldom2
# ldm add-vnet vnet0 primary-vsw0 ldom2
# ldm add-vnet linkprop=phys-state vnet1 primary-vsw1 ldom2
# ldm add-vnet linkprop=phys-state vnet2 primary-vsw2 ldom2
# ldm add-vdsdev /dev/rdisk/c3t0d2s2 ldom2@primary-vds0
# ldm add-vdisk ldom2 ldom2@primary-vds0 ldom2
```

3. ドメインの作成が終了したら、次のコマンドを使用してドメインをバインドし、起動します。ldom1をバインドして開始するには、サーバー1上の制御ドメインから次のコマンドを実行します。

```
# ldm bind ldom1
# ldm start ldom1
```

4. ldom2をバインドして開始するには、サーバー2上の制御ドメインから次のコマンドを実行します。

```
# ldm bind ldom2
# ldm start ldom2
```

ドメインがバインドされると、ドメイン・コンソールにアクセスできるようになります。実行するには、コマンド“ldm ls”を使用して、ドメインに関連付けられたコンソール・ポートを取得します。次に、適切なコンソール・ポートでtelnetコマンドを使用して、このコンソールにアクセスします。たとえば、ldm lsコマンドの出力によって、ドメインに関連付けられたコンソール・ポートが5000であると分かった場合、コマンド“telnet localhost 5000”を使用して制御ドメインからドメイン・コンソールにアクセスします。

すべてのゲスト・ドメインが開始されたら、これらのドメインに適切なOracle Solaris OSとパッチをインストールします。インストールは、ネットワーク経由またはDVDから、もしくはDVD ISOイメージを使用して実行します。詳しくは、『Oracle VM Server for SPARC管理ガイド』を参照してください。

すべてのゲスト・ドメインへのインストールが完了したら、Oracle RACで使用できるようにシステムを設定します。その他の設定作業には、パブリック・ネットワークとプライベート・ネットワークの構成（次項“ネットワーク構成”を参照）や、共有ストレージの追加（“ストレージ構成”の項を

参照)が含まれます。

ネットワーク構成

この項では、本書の例で使用されたネットワーク構成について、ネットワーク・インターフェース・レイアウト、IPアドレス割当て、ホスト名を含めて説明します。また、IPMP構成についても説明します。

ネットワーク・レイアウト

この構成には次の2つのネットワークが必要です。

- ・ パブリック・ネットワークは一般的な用途に使用でき、各ドメインに接続されています。このネットワークは、LANやWANをはじめとするその他すべてのネットワークに相互接続できます。
- ・ プライベート・ネットワークはOracle RACによって使用されます（ハートビートやキャッシュ・フュージョンなどのため）。このネットワークは2つのOracleノードに相互接続されています。その他のネットワークには接続しないでください。特に、プライベート・ネットワークをパブリック・ネットワークに接続することは避けてください。Oracle RAC以外のアプリケーションからこのプライベート・ネットワークを使用しないでください。

パブリック・ネットワーク

図5に示すとおり、パブリック・ネットワークは制御ドメイン (`vsw0`) とゲスト・ドメイン (`vnet0`) のプライマリ・ネットワーク・インターフェースに接続されています。また、各インターフェースは、制御ドメインの1番目の仮想スイッチ (`primary-vsw0`) に接続されています。この仮想スイッチは `nxge0` インターフェースを使用して、外部のパブリック・ネットワークにも接続されています。必要に応じて、このパブリック・ネットワークはLANやWANに接続できます。

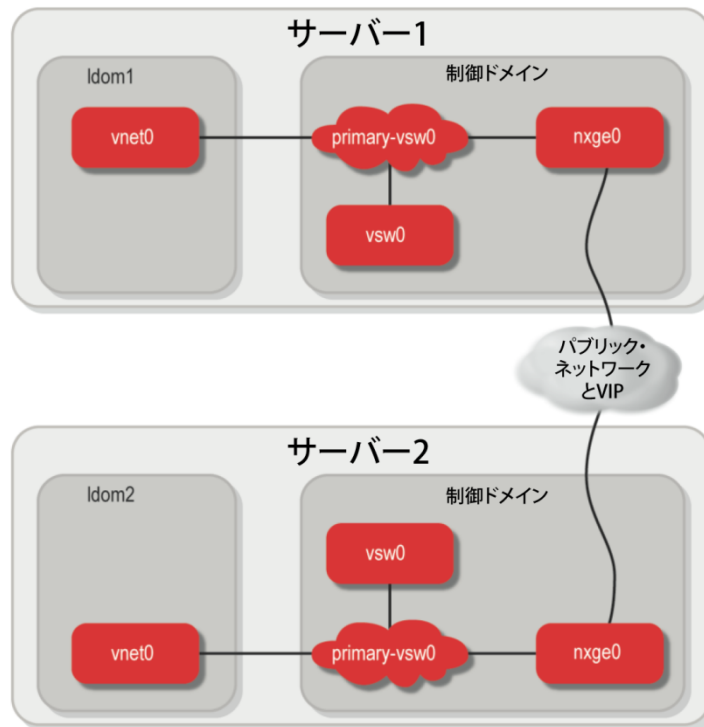


図5: パブリック・ネットワーク構成

プライベート・ネットワーク

図6に示すとおり、プライベート・ネットワークはゲスト・ドメイン (vnet1とvnet2) のセカンダリ・ネットワーク・インタフェースに接続されています。また、各インタフェース (vnet1とvnet2) は仮想スイッチ (primary-vsw1とprimary-vsw2) に接続されています。2台のサーバーは各システム上のnxge2とnxge3を使用して、物理ネットワーク・スイッチに接続されています。リンク・ベースのIPMPがvnet1とvnet2に対して使用されており、冗長性が提供されています。プライベート・ネットワークに接続するために、制御ドメインに仮想ネットワーク・インタフェースを作成する必要はありません。

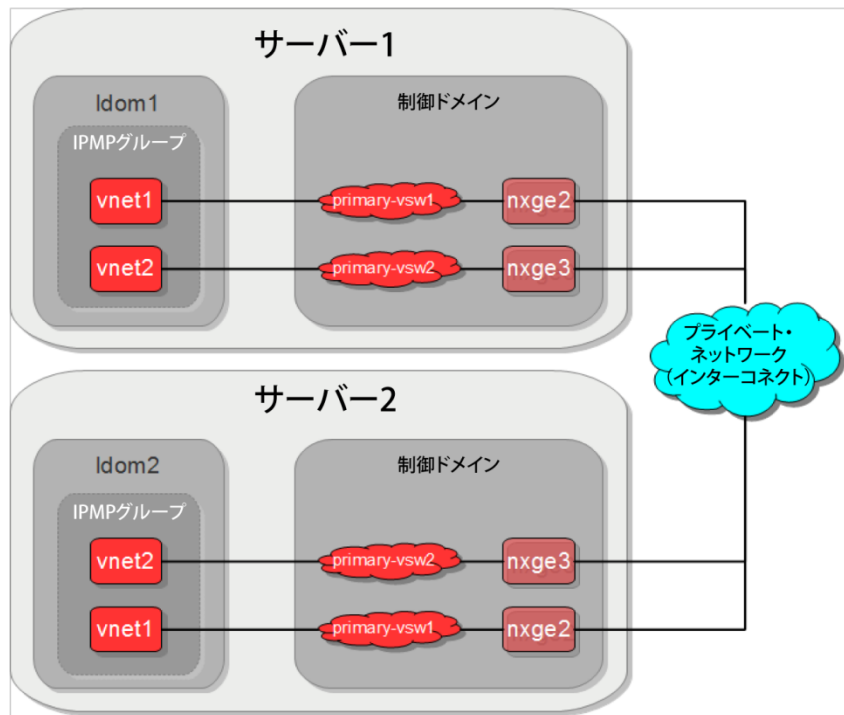


図6: プライベート・ネットワーク構成

ネットワーク・インタフェースのサマリー

図7に、1台のサーバー上にある制御ドメインとゲスト・ドメインのネットワーク・インタフェースの相互接続状況をまとめます。

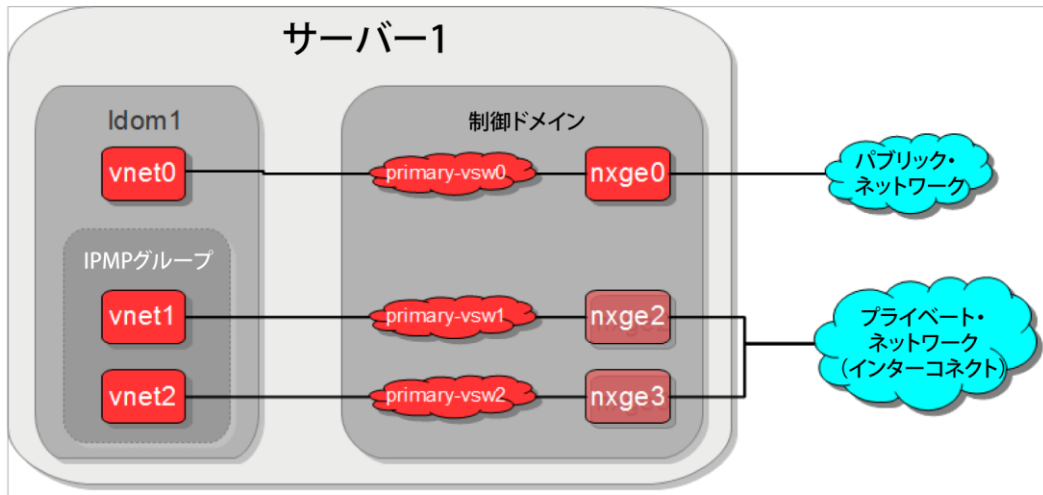


図7：物理サーバー内のネットワーク

ネットワーク・インタフェースとホスト名

この構成では、複数のホスト名とIPアドレスが必要です。表2に各種ホスト名とともに、ネットワーク・インタフェースやシステムとの関連を示します。

表2：ホスト名とIPアドレスの割当て

サーバー	ドメイン	インタフェース	ホスト名	説明
制御ドメイン - サーバー#1				
1	制御	vsw0	rac01	パブリック・ネットワーク、vswインタフェース（プライマリ）
ゲスト・ドメイン - サーバー#1				
1	ldom1	vnet0	rac-node1	ノード1、パブリック・ネットワーク（プライマリ）
1	ldom1	vnet1	node1-priv	ノード1、プライベート・ネットワーク（IPMPグループのプライマリ）
1	ldom1	vnet2	割当てなし	ノード1、プライベート・ネットワーク（IPMPグループのスタンバイ）
1	ldom1	なし	rac-nodevp1	ノード1、パブリック・ネットワーク、仮想IP
制御ドメイン - サーバー#2				
2	制御	vsw0	rac02	パブリック・ネットワーク、vswインタフェース（プライマリ）
ゲスト・ドメイン - サーバー#2				
2	ldom2	vnet0	rac-node2	ノード2、パブリック・ネットワーク（プライマリ）
2	ldom2	vnet1	node2-priv	ノード2、プライベート・ネットワーク（IPMPグループのプライマリ）
2	ldom2	vnet2	割当てなし	ノード2、プライベート・ネットワーク（IPMPグループのスタンバイ）
2	ldom2	なし	rac-nodevp2	ノード2、パブリック・ネットワーク、仮想IP

上記のホスト名と関連IPアドレスは、ドメインで使用されているネーミング・サービスに定義する必要があります。この際、DNSやNIS、または/etc/hostsなどを使用します。この定義は、すべてのドメインを通じて一貫している必要があります。また、DNSでVIP名とSCAN名（11g Release 2の場合）を使用する必要があります。

この構成例では、すべてのドメインで次のような/etc/hostsファイルが定義されています。

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1    localhost    localhost

# Public Network

10.1.9.101   rac01        # vsw0 - control domain server-1
10.1.9.102   rac02        # vsw0 - control domain server-2
10.1.9.111   rac-node1    # vnet0 - guest domain, server-1
10.1.9.112   rac-node2    # vnet0 - guest domain, server-2
10.1.9.121   rac-nodevp1  # vip of rac-node1
10.1.9.122   rac-nodevp2  # vip of rac-node2

# Private Network

192.168.10.111 node1-priv   # vnet1,2 - guest domain, server-1
192.168.10.112 node2-priv   # vnet1,2 - guest domain, server-2
```

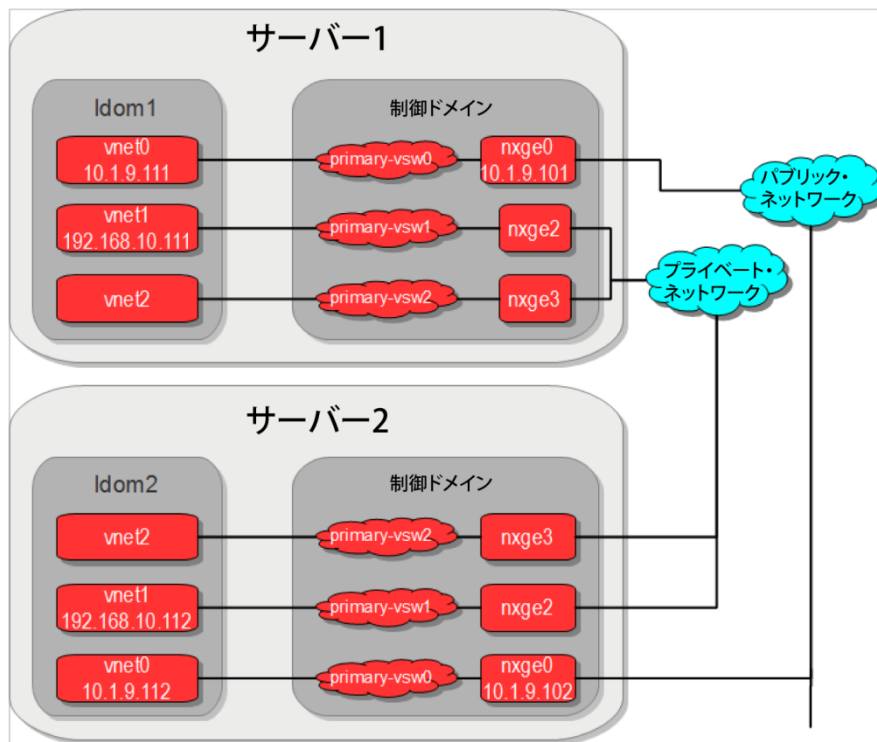



図8：IPアドレスの割当て

図8に、各ネットワーク・インタフェースに対するIPアドレスの割当てを示します。

- nxge0インタフェースは1番目の仮想スイッチ (primary-vsw0) に関連付けられており、システムはこの仮想スイッチのインタフェース (vsw0) を使用するため、このインタフェースにはIPアドレスは割り当てられていません (この区別は、Solaris 11の制御ドメインでは不要です)。
- 2番目と3番目の仮想スイッチ (primary-vsw1とprimary-vsw2) は物理ネットワーク・インタフェースnxge2およびnxge3に関連付けられています。
- vnet1インタフェースとvnet2インタフェースは、ネットワーク192.168.10.0/24に属しており、Oracle RACプライベート・ネットワークで使用されます。リンク・ベースのIPMPはこのインタフェース上に構成されています。このネットワークをユーザーが使用することはできません。
- vnet0インタフェースとvsw0インタフェースは、ネットワーク10.1.9.0/24に属しており、パブリック・ネットワークと仮想IP (VIP) で使用されます。このネットワークはユーザーに提供されています。

プライベート・ネットワーク構成

プライベート・ネットワークの構成には、ネットワーク・インタフェース名の指定とアクティブ/スタンバイ・モードでのリンク・ベースのIPMPの構成が含まれます。

vnet1インタフェースとvnet2インタフェースに対してアクティブ/スタンバイ・モードでIPMPを構成するためには、Solaris 10で次の設定を行います。

1. ゲスト・ドメインldom1での設定：

```
# cat /etc/hostname.vnet1
node1-priv group ipmp1
# cat /etc/hostname.vnet2
group ipmp1 standby
```

2. ゲスト・ドメインldom2での設定：

```
# cat /etc/hostname.vnet1
node2-priv group ipmp2
# cat /etc/hostname.vnet2
group ipmp2 standby
```

IPMPはOracle Solaris 11では、構成ファイルの編集に依存しない、別の方法で構成されます。代わりに、各ドメインでは次のようなコマンドを使用します。

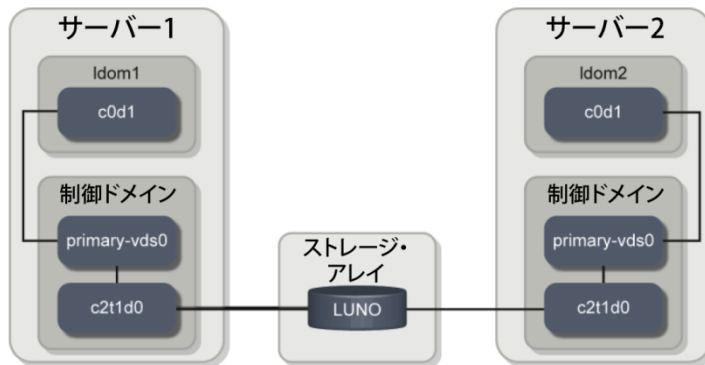
```
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i vnet1 -i vnet2 ipmp0
```

ストレージ構成

ゲスト・ドメインは最初、オペレーティング・システム向けのシステム・ディスクとして使用される1つの仮想ディスク (c0d0) のみで作成されます。Oracle RACを実行するには、両方のゲスト・ドメインに追加ディスクを割り当てる必要があります。追加ディスクは両方のゲスト・ドメインから参照できる必要があります。

共有ディスクと仮想ディスク

図9に、ゲスト・ドメインに対して共有LUN (外部ストレージからエクスポートしたもの) を仮想ディスクとして仮想化する様子を示します。ストレージ・アレイは制御ドメインに接続されており、この制御ドメインによって、LUNが仮想ディスクとしてゲスト・ドメインにエクスポートされます。ゲスト・ドメインに仮想ディスク・ドライブを追加する場合、両方のノードで仮想ディスクのデバイス名が一致するように、同じ順序で両方のシステムにLUNをエクスポートすることを推奨します。



LUNはマルチホストされているため、両方の制御ドメイン上で同じ名前 (c2t1d0) を使用してアクセスできます。

図9：共有ストレージには、同じデバイス名を使用して複数のドメインからアクセスできます。

ゲスト・ドメインへの共有ディスクの追加

この構成では、2番目のストレージ・アレイの5つのLUNが共有ディスクとして使用されています。両方のサーバーのハードウェア構成が同じであることから、2番目のストレージ・アレイのLUNは両方の制御ドメインから同じデバイス名 (c2t1d0~c2t1d4) で表示されます。

注：LUNはサーバーごとに異なる名前が表示される可能性があります (実際に、頻繁にそうなります)。制御ドメインではLUN名が一致する必要はありませんが、ゲスト・ドメインのLUN名は一致しなければなりません。これは、すべてのドメインを通じて同じ順序でLUNをインポートすることで、簡単に実現できます。

仮想ディスクとしてLUNをエクスポートするには、次のコマンドを実行します。次の例では、

c2t1d0s2がOCR LUNのデバイス名になります。

1. ldom1にLUNを追加するには、サーバー1上の制御ドメインから次のコマンドを実行します。

```
# ldm add-vdsdev /dev/rdisk/c2t1d0s2 ocr1@primary-vds0
# ldm add-vdisk ocr1 ocr1@primary-vds0 ldom1
```

2. ldom2にLUNを追加するには、サーバー2上の制御ドメインから次のコマンドを実行します。

```
# ldm add-vdsdev /dev/rdisk/c2t1d0s2 ocr2@primary-vds0
# ldm add-vdisk ocr2 ocr2@primary-vds0 ldom2
```

新しいディスクはすぐにゲスト・ドメインに追加されます。ディスクが表示されたら、任意のゲスト・ドメインからformat (1m) コマンドを使用して、新しいディスクを分割してOCRに必要なパーティションを作成します。正しいパーティションが参照できることをすべてのゲスト・ドメインから確認します。

それぞれの共有ディスク (投票ディスクとASMディスク) に対して、同じ手順を繰り返します。すべての共有ディスクがゲスト・ドメインに追加された後の最終的な構成を図10に示します。

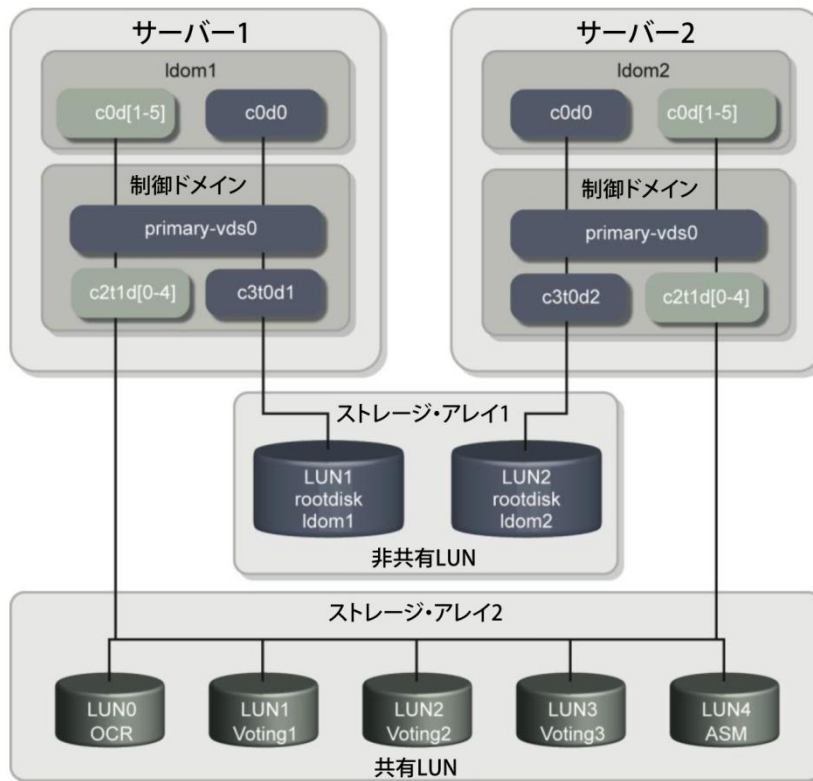


図10: ストレージのケーブル配線

Oracleのインストール

Oracle RACとOracleデータベース・ソフトウェアのインストールは、標準のOracleインストールと類似しています。パブリック・ネットワークとVIPネットワークにはネットワーク10.1.9.0/24を使用し、プライベート・ネットワークには192.168.10.0/24を使用します。はじめにOracle Clusterwareをインストールし、次に最新のパッチ・セットを適用します。続いてOracleデータベースをインストールし、同じパッチ・セットを今回はOracleデータベースに適用します。

クラスタ・インターコネクトに対するIPMPの構成

クラスタ・インターコネクトに対するIPMPの構成については、<http://support.oracle.com>のDoc ID 1069584.1を参照してください。Oracle 11g Release 2の場合、Oracleをインストールし、次に示すインストール後の手順を実行している間は、プライベート・インタフェースとしてvnet1を選択すれば十分です。

```
# $GRID_HOME/bin/oifcfg getif
vnet0 10.6.241.0 global public
vnet1 192.168.1.0 global cluster_interconnect
# $GRID_HOME/bin/oifcfg setif -global vnet2/192.168.1.0:cluster_interconnect
# $GRID_HOME/bin/oifcfg getif
vnet0 10.6.241.0 global public
vnet1 192.168.1.0 global cluster_interconnect
vnet2 192.168.1.0 global cluster_interconnect
```

追加情報

この項では、Oracle VM Server for SPARCへのOracle RACのデプロイメントに関する追加情報を提供します。次の内容が含まれます。

- NTP
- CPUの動的な再構成
- メモリの動的な再構成
- パフォーマンスに関する考慮事項
- ジャンボ・フレーム

NTP

異なるドメイン間で、システム・クロックを同期する必要があります。このため、すべてのドメインでネットワーク時刻の同期プロトコル（NTP）を使用します。

この例では、1番目のサーバー（rac01）の制御ドメインがタイム・ソースとして使用されるとともに、NTPサーバーとして構成されています。

```
# grep -v ^# /etc/inet/ntp.conf
server 127.127.1.0 prefer
broadcast 224.0.1.1 ttl 4
enable auth monitor
driftfile /var/ntp/ntp.drift
statsdir /var/ntp/ntpstats/
filegen peerstats file peerstats type day enable
filegen loopstats file loopstats type day enable
filegen clockstats file clockstats type day enable
keys /etc/inet/ntp.keys
trustedkey 0
requestkey 0
controlkey 0
# touch /var/ntp/ntp.drift
# svcadm enable ntp
```

その他のドメイン（rac02、rac-node1、rac-node2）はNTPクライアントとして構成されています。

```
# grep -v ^# /etc/inet/ntp.conf
server rac01 prefer
slewalways yes
disable pll
# svcadm enable ntp
```

CPUの動的な再構成

Oracle VM Server for SPARCでは、CPUの動的な再構成がサポートされています。制御ドメインで`ldm add-vcpu`または`ldm rm-vcpu`コマンドを使用すると、任意のアクティブ・ドメイン（制御ドメインを含む）からCPUを動的に追加または削除できます。`ldm add-core`または`ldm rm-core`コマンドを使用することで、CPUのコア全体を単位として、割り当てることができます。これらのコマンドはOracle RACを実行しているドメインで使用できます。

ドメインのCPUを減らす場合、ドメインのワークロードを効率的に処理するために十分なCPUが残されていることを確認してください（Oracle RACゲスト・ドメインには常に少なくとも8つの仮想CPUが割り当てられている必要があります）。制御ドメインからCPUの利用率を表示するためには、`ldm list`コマンドを使用できます。`vmstat`、`mpstat`、および`prstat`コマンドは、CPUリソースの応答待ちによるCPUの待機時間および利用率を示すために、仮想環境で実行されていない場合のSolarisと同じように使用できます。CPU数の変更は、再起動なしですぐに認識されます。

ドメインにハード・パーティショニングが構成されている場合、このドメインではCPUコア全体に対してのみ追加や削除を実行できます。また、ドメインに割り当てられるCPUコアの数は、`ldm set max-cores`コマンドを使用して、ドメインごとにCPUコアの数を指定することで制限できます。

同じ物理サーバーに複数のドメインがデプロイされている場合、CPUリソース・マネージャ・ポリシーを設定することでワークロードの状態の変化に応じて、CPUの動的な再構成を自動化できます。

つまり、CPUを必要とするドメインにCPUを割り当て、CPUを必要としないドメインから余分なCPUを取り除くために、ポリシーを設定できます。次の例では、負荷および時刻に応じて最小24から最大64までの範囲で`ldom1`に対しCPUが追加または削除されます。ドメインでCPUの利用率が25%以下に低下している場合は、24個のCPUの下限值に達するか、CPU利用率が25%に達するまでCPUが削除されます。CPUの利用率が75%を超えた場合は、64個のCPUの上限值に達するか、CPU利用率が75%以下に低下するまでCPUが追加されます。

```
# ldm add-policy tod-begin=09:00 tod-end=18:00 util-lower=25 util-upper=75 vcpu-min=24
vcpu-max=64 attack=8 decay=8 priority=1 name=high-usage ldom1
```

メモリの動的な再構成

Oracle VM Server for SPARCでは、メモリの動的な再構成がサポートされています。`ldm add-mem`、`ldm rm-mem`、または`ldm set-mem`コマンドを使用すると、任意のアクティブ・ドメインからメモリを動的に追加または削除できます。メモリを追加または削除する際、Solarisを再起動する必要はありません。ゲスト・ドメインに割り当てるメモリ・サイズを減らす場合は、このドメインで実行されている各Oracleデータベース・インスタンスに割り当てられているメモリの量について考慮してください。この量を下回るとスワップが発生し、パフォーマンスの低下が起るため、そのような設定は避けてください。

- メモリ・サイズを小さくするために、はじめに、ドメインで実行されているOracleデータベース・インスタンスへ割り当てられたメモリ・サイズを減らさなければならない場合があります。また、使用しているメモリ管理の種類によっては、これらのインスタンスを再起動する必要があります。
- ドメインにメモリを追加した後に、追加されたメモリを有効利用するため、このドメインで

実行されているデータベース・インスタンスのメモリ設定を調整しなければならない場合があります。また、使用しているメモリ管理の種類によっては、これらのインスタンスを再起動する必要があります。

パフォーマンスに関する考慮事項

この例で使用した構成では、制御ドメインがI/Oとネットワーク・フローの処理を行います。テストでは、制御ドメイン上のCPUのシステム・ロードは、300MB/秒のストレージI/Oロード（各制御ドメイン）と約25MB/秒のネットワーク・ロードに対して、およそ15%でした。

ゲスト・ドメインでスワップ・アクティビティが発生すると、追加のI/Oリクエストが生成され、制御ドメイン、およびサービス・ドメインでの処理が必要になるため、ゲスト・ドメインでスワップ・アクティビティが発生しないようにしてください。スワップの回避は、非仮想化環境におけるベスト・プラクティスですが、仮想マシンでは、さらなる重要性を持ちます。

複数ドメイン間でCPUコアを共有しないでください。特に、Oracleデータベースを実行している2つのドメイン間での共有は避けてください。同じCPUコアのCPUスレッドが別々のドメインに割り当てられると、各コアのレベル1キャッシュの競合のため、CPUスレッドの効率が低下する場合があります。前述のように、これはコア全体の制約（whole-core制約）を使用して達成できませんが、CPUスレッドを8ずつ（コアあたりのスレッド数）均等に割り当てることでも達成できます。

Non Uniform Memory Access (NUMA) の影響により、ドメイン内のメモリ待機時間が増加するため、より大きなM5-32やT5-8といったマルチソケット・サーバーにおいて、ドメインが複数のソケットを対象とすることを避けることも有効です。これは、『Oracle VM Server for SPARC 3.1管理ガイド』の「ドメインへの物理リソースの割当て」で説明されているように、名前付きのCPUリソースによっても達成できますが、ドメインがバインドされている場合には、NUMA待機時間を最適化するOracle VM Server for SPARC Logical Domains Managerによっても自動的に最適化されます。

ジャンボ・フレーム

Oracle VM Server for SPARC 2.2以降では、仮想スイッチと仮想ネットワーク・デバイスでジャンボ・フレームを使用するよう構成できます。また、Maximum Transmission Unit (MTU) を最大16,000バイトに設定できます。仮想スイッチと仮想ネットワーク・デバイスにジャンボ・フレームを構成する方法について、詳しくは『*Oracle VM Server for SPARC管理ガイド*』を参照してください。

ジャンボ・フレームを構成するには、すべてのNICとネットワーク・スイッチでジャンボ・フレームがサポートされている必要があります。次に、nxge2インタフェースとnxge3インタフェースでジャンボ・フレームをサポートするための構成例を示します。ゲスト・ドメインを作成する前の、プライマリ・ドメインの構成中にジャンボ・フレームを構成することを推奨します。こうすることでシステムの再起動回数が減るため、時間が節約できます。

1. ジャンボ・フレームを構成するネットワーク・インタフェースのデバイス・パス (nxge2およびnxge3) を決定します。

```
# cat /etc/path_to_inst | grep nxge
"/pci@500/pci@0/pci@c/network@0" 0 "nxge"
"/pci@500/pci@0/pci@c/network@0,1" 1 "nxge"
"/pci@500/pci@0/pci@c/network@0,2" 2 "nxge"
"/pci@500/pci@0/pci@c/network@0,3" 3 "nxge"
```

2. ドライバの構成ファイル (nxge.conf) を編集して、指定したデバイスでジャンボ・フレームを有効化します。

```
# cd /platform/sun4v/kernel/drv
# vi nxge.conf
name = "pciex108e,abcd" parent = "/pci@500/pci@0/pci@c" unit-address = "0,2"
accept_jumbo = 1;
name = "pciex108e,abcd" parent = "/pci@500/pci@0/pci@c" unit-address = "0,3"
accept_jumbo = 1;
# init 6
```

3. MTUサイズ=9000をサポートするように、仮想スイッチを変更します。

```
# dladm show-link
vsw0          type: non-vlan mtu: 1500      device: vsw0
vsw1          type: non-vlan mtu: 1500      device: vsw1
vsw2          type: non-vlan mtu: 1500      device: vsw2
nxge0         type: non-vlan mtu: 1500      device: nxge0
nxge1         type: non-vlan mtu: 1500      device: nxge1
nxge2         type: non-vlan mtu: 9194      device: nxge2
nxge3         type: non-vlan mtu: 9194      device: nxge3
```



```
# ldm set-vsw mtu=9000 primary-vsw2
# ldm set-vsw mtu=9000 primary-vsw3
# init 6
```

4. ゲスト・ドメインのvnet2とvnet3は、自動的に9000というMTUサイズを受け取ります。

まとめ

Oracle RACは、Oracle VM Server for SPARCが構成されたサーバーにインストールできます。仮想化テクノロジーであるOracle VM Server for SPARCを利用すると、1つの物理システム上に複数の仮想システムを作成できます。開発用に適した低コストのオプションとして、同じ物理サーバー上の論理ドメインに複数のOracle RACノードを構成できます。また、別々の物理サーバー上の論理ドメインにOracle RACノードをデプロイして、本番デプロイメントの可用性を向上させることができます。

このホワイト・ペーパーでは、2つの異なる物理サーバー上の論理ドメインにOracle RACノードをデプロイした、本番デプロイメント・タイプの構成例を示しました。論理ドメインの作成と構成、ネットワーク設定、ストレージ構成を含む構成プロセス全体を、段階的な手順で説明しました。また、Oracle VM Server for SPARCへのOracle RACのデプロイメントを計画する管理者を支援するため、構成ガイドラインとソフトウェア要件も提供されています。

付録

Logical Domains Managerの構成例

この項では、本書の例として使用されたOracle VM Server for SPARCの構成を示します。これらは例にすぎず、実際の構成とは異なる場合があります。

1番目の制御ドメイン (rac01) の構成

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	32	16G	2.9%	2d 16h 18m
SOFTSTATE							
Solaris running							
UUID							
701ddc0f-a616-e982-bba8-a9bbfbec2c04							
MAC							
00:14:4f:aa:c4:b8							
HOSTID							
0x84aac4b8							
CONTROL							
failure-policy=ignore							
extended-mapin-space=off							
cpu-arch=native							
DEPENDENCY							
master=							
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)						
5	(40, 41, 42, 43, 44, 45, 46, 47)						
VCPU							
VID	PID	CID	UTIL	STRAND			
0	0	0	3.9%	100%			
1	1	0	2.9%	100%			
2	2	0	2.7%	100%			
3	3	0	0.6%	100%			

4	4	0	1.3%	100%
5	5	0	4.7%	100%
6	6	0	1.2%	100%
7	7	0	2.2%	100%
8	8	1	1.4%	100%
9	9	1	2.0%	100%
10	10	1	5.7%	100%
11	11	1	0.2%	100%
12	12	1	0.1%	100%
13	13	1	4.7%	100%
14	14	1	9.9%	100%
15	15	1	0.2%	100%
16	16	2	0.4%	100%
17	17	2	1.8%	100%
18	18	2	2.8%	100%
19	19	2	4.2%	100%
20	20	2	0.7%	100%
21	21	2	0.2%	100%
22	22	2	0.9%	100%
23	23	2	4.7%	100%
24	40	5	7.8%	100%
25	41	5	0.1%	100%
26	42	5	0.2%	100%
27	43	5	1.8%	100%
28	44	5	7.7%	100%
29	45	5	0.2%	100%
30	46	5	0.2%	100%
31	47	5	18%	100%

MAU

ID	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)
5	(40, 41, 42, 43, 44, 45, 46, 47)

MEMORY

RA	PA	SIZE
0x10000000	0x4a0000000	256M
0x402000000	0x12000000	2G
0x890000000	0x130000000	14080M

CONSTRAINT

cpu=whole-core

max-cores=unlimited

```

threading=max-throughput

VARIABLES
  keyboard-layout=US-English

IO
  DEVICE          PSEUDONYM      OPTIONS
  pci@400         pci_0
  pci@500         pci_1
  pci@400/pci@0/pci@d MB/PCIE0
  pci@400/pci@0/pci@c MB/PCIE1
  pci@400/pci@0/pci@1 MB/HBA
  pci@500/pci@0/pci@d MB/PCIE4
  pci@500/pci@0/pci@9 MB/PCIE5
  pci@500/pci@0/pci@c MB/NET0

VCC
  NAME          PORT-RANGE
  primary-vcc0  5000-5100

VSW
  NAME          MAC          NET-DEV  ID  DEVICE      LINKPROP  DEFAULT-VLAN-ID  PVID
  VID          MTU    MODE
  primary-vsw0  00:14:4f:fb:d0:c0 nxge0    0   switch@0    1         1
  1500
  primary-vsw1  00:14:4f:fa:99:8b nxge2    1   switch@1    phys-state 1         1
  9000
  primary-vsw2  00:14:4f:f9:87:e8 nxge3    2   switch@2    phys-state 1         1
  9000

VDS
  NAME          VOLUME      OPTIONS      MPGROUP      DEVICE
  primary-vds0  ldom1
  ocr1
  voting11
  voting21
  voting31
  ASM1
  /dev/rdisk/c3t0d1s2
  /dev/rdisk/c2t1d0s2
  /dev/rdisk/c2t1d1s2
  /dev/rdisk/c2t1d2s2
  /dev/rdisk/c2t1d3s2
  /dev/rdisk/c2t1d4s2

VCONS
  NAME          SERVICE      PORT
  SP

```

1番目のゲスト・ドメイン (ldom1) の構成

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldom1	active	-n----	5000	48	24G	15%	3h 58m
SOFTSTATE							
Solaris running							
UUID							
f3e1006b-16d4-e266-8447-8eccc1518345							
MAC							
00:14:4f:fa:f7:c8							
HOSTID							
0x84faf7c8							
CONTROL							
failure-policy=ignore							
extended-mapin-space=off							
cpu-arch=native							
DEPENDENCY							
master=							
CORE							
CID	CPUSET						
3	(24, 25, 26, 27, 28, 29, 30, 31)						
4	(32, 33, 34, 35, 36, 37, 38, 39)						
6	(48, 49, 50, 51, 52, 53, 54, 55)						
9	(72, 73, 74, 75, 76, 77, 78, 79)						
10	(80, 81, 82, 83, 84, 85, 86, 87)						
11	(88, 89, 90, 91, 92, 93, 94, 95)						
VCPU							
VID	PID	CID	UTIL	STRAND			
0	24	3	51%	100%			
1	25	3	32%	100%			
2	26	3	22%	100%			
3	27	3	28%	100%			
4	28	3	33%	100%			
5	29	3	37%	100%			
6	30	3	19%	100%			
7	31	3	29%	100%			
8	32	4	22%	100%			
9	33	4	28%	100%			
10	34	4	29%	100%			

11	35	4	24%	100%
12	36	4	16%	100%
13	37	4	16%	100%
14	38	4	15%	100%
15	39	4	15%	100%
16	48	6	15%	100%
17	49	6	33%	100%
18	50	6	31%	100%
19	51	6	23%	100%
20	52	6	21%	100%
21	53	6	23%	100%
22	54	6	29%	100%
23	55	6	27%	100%
24	72	9	4.3%	100%
25	73	9	1.7%	100%
26	74	9	5.4%	100%
27	75	9	3.7%	100%
28	76	9	2.3%	100%
29	77	9	3.8%	100%
30	78	9	2.9%	100%
31	79	9	2.8%	100%
32	80	10	4.5%	100%
33	81	10	1.9%	100%
34	82	10	1.8%	100%
35	83	10	3.7%	100%
36	84	10	3.4%	100%
37	85	10	3.5%	100%
38	86	10	3.7%	100%
39	87	10	3.0%	100%
40	88	11	2.0%	100%
41	89	11	5.3%	100%
42	90	11	5.4%	100%
43	91	11	6.8%	100%
44	92	11	3.2%	100%
45	93	11	2.8%	100%
46	94	11	1.6%	100%
47	95	11	4.9%	100%

MAU

ID	CPUSET
3	(24, 25, 26, 27, 28, 29, 30, 31)
4	(32, 33, 34, 35, 36, 37, 38, 39)
6	(48, 49, 50, 51, 52, 53, 54, 55)
9	(72, 73, 74, 75, 76, 77, 78, 79)
10	(80, 81, 82, 83, 84, 85, 86, 87)
11	(88, 89, 90, 91, 92, 93, 94, 95)

MEMORY

RA	PA	SIZE
----	----	------

```

0x2000000      0x92000000      2528M
0x400000000    0x4b0000000    13856M
0x800000000    0x820000000    8G

CONSTRAINT

cpu=whole-core

max-cores=unlimited

threading=max-throughput

VARIABLES

boot-device=/virtual-devices@100/channel-devices@200/disk@0:a vnet0
keyboard-layout=US-English

NETWORK

NAME          SERVICE          ID  DEVICE      MAC              MODE  PVID  VID
MTU  LINKPROP
vnet0        primary-vsw0@primary  0   network@0   00:14:4f:f8:51:5f  1
1500
vnet1        primary-vsw1@primary  1   network@1   00:14:4f:f8:b4:fb  1
9000 phys-state
vnet2        primary-vsw2@primary  2   network@2   00:14:4f:fa:13:f0  1
9000 phys-state

DISK

NAME          VOLUME          TOUT  DEVICE      SERVER
primary-vds0  ldom1@primary-vds0  disk@0  primary
ocr1@primary-vds0  disk@1  primary
voting11@primary-vds0  disk@2  primary
voting21@primary-vds0  disk@3  primary
voting31@primary-vds0  disk@4  primary
ASM1@primary-vds0  disk@5  primary

VCONS

NAME          SERVICE          PORT
ldom1        primary-vcc0@primary  5000

```

2番目の制御ドメイン (rac02) の構成

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	32	16G	2.8%	5d 106h 22m
SOFTSTATE							
Solaris running							
UUID							
701ddc0f-a616-e982-bba8-a9ccfbec2c04							
MAC							
00:14:4f:bb:c4:b8							
HOSTID							
0x84bbc4b8							
CONTROL							
failure-policy=ignore							
extended-mapin-space=off							
cpu-arch=native							
DEPENDENCY							
master=							
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)						
5	(40, 41, 42, 43, 44, 45, 46, 47)						
VCPU							
VID	PID	CID	UTIL	STRAND			
0	0	0	3.9%	100%			
1	1	0	2.9%	100%			
2	2	0	2.7%	100%			
3	3	0	0.6%	100%			
4	4	0	1.3%	100%			
5	5	0	4.7%	100%			
6	6	0	1.2%	100%			
7	7	0	2.2%	100%			
8	8	1	1.4%	100%			
9	9	1	2.0%	100%			
10	10	1	5.7%	100%			
11	11	1	0.2%	100%			
12	12	1	0.1%	100%			
13	13	1	4.7%	100%			


```
14 14 1 9.9% 100%
15 15 1 0.2% 100%
16 16 2 0.4% 100%
17 17 2 1.8% 100%
18 18 2 2.8% 100%
19 19 2 4.2% 100%
20 20 2 0.7% 100%
21 21 2 0.2% 100%
22 22 2 0.9% 100%
23 23 2 4.7% 100%
24 40 5 7.8% 100%
25 41 5 0.1% 100%
26 42 5 0.2% 100%
27 43 5 1.8% 100%
28 44 5 7.7% 100%
29 45 5 0.2% 100%
30 46 5 0.2% 100%
31 47 5 18% 100%

MAU
ID CPUSSET
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)
5 (40, 41, 42, 43, 44, 45, 46, 47)

MEMORY
RA PA SIZE
0x10000000 0x4a0000000 256M
0x402000000 0x12000000 2G
0x890000000 0x130000000 14080M

CONSTRAINT

cpu=whole-core

max-cores=unlimited

threading=max-throughput

VARIABLES

keyboard-layout=US-English

IO
DEVICE PSEUDONYM OPTIONS
pci@400 pci_0
pci@500 pci_1
```

```

pci@400/pci@0/pci@d MB/PCIE0
pci@400/pci@0/pci@c MB/PCIE1
pci@400/pci@0/pci@1 MB/HBA
pci@500/pci@0/pci@d MB/PCIE4
pci@500/pci@0/pci@9 MB/PCIE5
pci@500/pci@0/pci@c MB/NET0

VCC
NAME          PORT-RANGE
primary-vcc0  5000-5100

VSW
NAME          MAC          NET-DEV  ID  DEVICE      LINKPROP  DEFAULT-VLAN-ID  PVID
VID          MTU   MODE
primary-vsw0  00:14:4f:fa:d0:c0 nxge0    0  switch@0    1          1
1500
primary-vsw1  00:14:4f:fb:88:8b nxge2    1  switch@1    phys-state 1          1
9000
primary-vsw2  00:14:4f:fb:86:e8 nxge3    2  switch@2    phys-state 1          1
9000

VDS
NAME          VOLUME      OPTIONS      MPGROUP      DEVICE
primary-vds0  ldom2       ocr2         /dev/rdsk/c3t0d2s2
              voting12    /dev/rdsk/c2t1d0s2
              voting22    /dev/rdsk/c2t1d1s2
              voting32    /dev/rdsk/c2t1d2s2
              ASM2       /dev/rdsk/c2t1d3s2
              /dev/rdsk/c2t1d4s2

VCONS
NAME          SERVICE      PORT
SP

```

2番目のゲスト・ドメイン (ldom2) の構成

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldom2	active	-n----	5000	48	24G	19%	12h 38m
SOFTSTATE							
Solaris running							
UUID							
f3e1006b-16d4-e266-8447-8ebbb2619456							
MAC							
00:14:4f:fb:43:cc							
HOSTID							
0x84fb43cc							
CONTROL							
failure-policy=ignore							
extended-mapin-space=off							
cpu-arch=native							
DEPENDENCY							
master=							
CORE							
CID	CPUSET						
3	(24, 25, 26, 27, 28, 29, 30, 31)						
4	(32, 33, 34, 35, 36, 37, 38, 39)						
6	(48, 49, 50, 51, 52, 53, 54, 55)						
9	(72, 73, 74, 75, 76, 77, 78, 79)						
10	(80, 81, 82, 83, 84, 85, 86, 87)						
11	(88, 89, 90, 91, 92, 93, 94, 95)						
VCPU							
VID	PID	CID	UTIL	STRAND			
0	24	3	51%	100%			
1	25	3	32%	100%			
2	26	3	22%	100%			
3	27	3	28%	100%			
4	28	3	33%	100%			
5	29	3	37%	100%			
6	30	3	19%	100%			
7	31	3	29%	100%			
8	32	4	22%	100%			
9	33	4	28%	100%			
10	34	4	29%	100%			
11	35	4	24%	100%			

12	36	4	16%	100%
13	37	4	16%	100%
14	38	4	15%	100%
15	39	4	15%	100%
16	48	6	15%	100%
17	49	6	33%	100%
18	50	6	31%	100%
19	51	6	23%	100%
20	52	6	21%	100%
21	53	6	23%	100%
22	54	6	29%	100%
23	55	6	27%	100%
24	72	9	4.3%	100%
25	73	9	1.7%	100%
26	74	9	5.4%	100%
27	75	9	3.7%	100%
28	76	9	2.3%	100%
29	77	9	3.8%	100%
30	78	9	2.9%	100%
31	79	9	2.8%	100%
32	80	10	4.5%	100%
33	81	10	1.9%	100%
34	82	10	1.8%	100%
35	83	10	3.7%	100%
36	84	10	3.4%	100%
37	85	10	3.5%	100%
38	86	10	3.7%	100%
39	87	10	3.0%	100%
40	88	11	2.0%	100%
41	89	11	5.3%	100%
42	90	11	5.4%	100%
43	91	11	6.8%	100%
44	92	11	3.2%	100%
45	93	11	2.8%	100%
46	94	11	1.6%	100%
47	95	11	4.9%	100%

MAU

ID	CPUSET
3	(24, 25, 26, 27, 28, 29, 30, 31)
4	(32, 33, 34, 35, 36, 37, 38, 39)
6	(48, 49, 50, 51, 52, 53, 54, 55)
9	(72, 73, 74, 75, 76, 77, 78, 79)
10	(80, 81, 82, 83, 84, 85, 86, 87)
11	(88, 89, 90, 91, 92, 93, 94, 95)

MEMORY

RA	PA	SIZE
0x2000000	0x92000000	2528M

```

0x400000000    0x4b0000000    13856M
0x800000000    0x820000000    8G

```

CONSTRAINT

```

cpu=whole-core

max-cores=unlimited

threading=max-throughput

```

VARIABLES

```

boot-device=/virtual-devices@100/channel-devices@200/disk@0:a vnet0
keyboard-layout=US-English

```

NETWORK

NAME	SERVICE	ID	DEVICE	MAC	MODE	PVID	VID
MTU	LINKPROP						
vnet0	primary-vsw0@primary	0	network@0	00:14:4f:f9:02:a7		1	
1500							
vnet1	primary-vsw1@primary	1	network@1	00:14:4f:f8:c4:fb		1	
9000	phys-state						
vnet2	primary-vsw2@primary	2	network@2	00:14:4f:fa:23:f0		1	
9000	phys-state						

DISK

NAME	VOLUME	TOUT	DEVICE	SERVER
primary-vds0	ldom2@primary-vds0		disk@0	primary
	ocr2@primary-vds0		disk@1	primary
	voting12@primary-vds0		disk@2	primary
	voting22@primary-vds0		disk@3	primary
	voting32@primary-vds0		disk@4	primary
	ASM2@primary-vds0		disk@5	primary

VCONS

NAME	SERVICE	PORT
ldom2	primary-vcc0@primary	5000

参考資料

次の表に、本書で示したオラクルの関連資料のリストを示します。

表3：参考資料

参考資料	URL
オラクルの仮想化	http://www.oracle.com/jp/technologies/virtualization/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/overview/index.html
Oracle Real Application Clusters (Oracle RAC)	http://www.oracle.com/technetwork/jp/products/clustering/overview/index.html
Oracle Clusterware	http://www.oracle.com/technetwork/jp/database/database-technologies/clusterware/overview/index.html
Oracle Automatic Storage Management	http://www.oracle.com/technetwork/jp/products/cloud-storage/index.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/vmsvrsparc-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/vmsvrsparc-availability-163930-ja.pdf



Oracle VM Server for SPARCでの
Oracle Real Application Clustersの実行
2014年3月、バージョン1.2
著者 : Alexandre Chartre, Roman
Ivanov, John Mchugh, Jeff Savit

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

お問い合わせ窓口
Oracle Direct

TEL 0120-155-096
URL oracle.com/jp/direct



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

Copyright © 2014, 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。

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。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。UNIXは、The Open Groupの登録商標です。0114

Hardware and Software, Engineered to Work Together