



Oracle ホワイトペーパー
2011 年 2 月

ゾーンクラスター-仮想クラスターの配備方法と 配備する理由

はじめに	1
クラスタアプリケーションの統合	2
サーバー仮想化テクノロジー	3
ハードウェアパーティション	4
仮想マシン	4
オペレーティングシステムの仮想化	5
リソース管理	6
サーバー仮想化アプローチの選択	6
ゾーンクラスタの概要	7
クラスタ用語	7
グローバルクラスタとゾーンクラスタ	8
セキュリティー分離	9
アプリケーション障害分離	10
リソース管理	10
専用クラスタモデル	10
ゾーンクラスタのユースケース	11
複数組織の統合	11
機能統合	12
多層統合	14
コスト削減	15
管理ワークロードの軽減	16
ゾーンクラスタの設計	16
仮想ノード	16
クラスタメンバーシップ	17
セキュリティー	18
ファイルシステム	19
ストレージデバイス	21
ネットワーク	22
管理の概要	24
ゾーンクラスタ管理	24
アプリケーション管理	25
ゾーンクラスタ構成の例	26

予備的な構成	26
ゾーンクラスタ構成	26
ゾーンクラスタ管理	30
ノードとクラスタ全体スコープ	30
システム識別	31
ノードサポート	31
ファイルシステムサポート	32
ストレージデバイスサポート	34
ネットワークサポート	35
起動操作と停止操作	37
削除操作	37
ゾーンクラスタ情報の表示	37
複製操作	38
その他のゾーンサブコマンド	38
Oracle Solaris OS のコマンド対話処理	38
ゾーンクラスタ管理 GUI	39
要約	39
著者について	39
謝辞	40
関連資料	41

はじめに

多くの組織は、より良いコンピュータシステムの利用法を探し求めています。仮想化テクノロジーを使用すると、1 つのシステム上の複数のアプリケーションを安全な方法で統合できます。このホワイトペーパーでは、ゾーンクラスタ (Oracle Solaris Containers クラスタとも呼ばれる) を紹介します。これは、Oracle Solaris ゾーンが仮想ノードとして構成される仮想クラスタです。ゾーンクラスタは、1 つのクラスタ上の複数のクラスタアプリケーションの統合をサポートします。

このホワイトペーパーは、次のトピックで構成されています。

- 2 ページの「クラスタアプリケーションの統合」では、統合化の原動力について説明しています。
- 3 ページの「サーバー仮想化テクノロジー」では、オラクルの仮想化テクノロジーの概要を示しています。ここでは、Oracle Solaris ゾーンに重点が置かれています。
- 7 ページの「ゾーンクラスタの概要」では、ゾーンクラスタを紹介するとともに、そのユーティリティを示すさまざまなユースケースを示しています。
- 16 ページの「ゾーンクラスタの設計」では、ゾーンクラスタの全体的な設計について説明しています。
- 23 ページの「管理の概要」では、ゾーンクラスタの管理の概要を示しています。
- 25 ページの「ゾーンクラスタ構成の例」では、ゾーンクラスタ構成の例を使用して、段階的な手順を示しています。
- 30 ページの「ゾーンクラスタ管理」では、一般的なゾーンクラスタの管理タスクについて説明しています。

このホワイトペーパーでは、Oracle Solaris クラスタおよび Oracle Solaris ゾーンに関する概念について十分な知識があることを前提にしています。

クラスタアプリケーションの統合

つい最近になって、単一のクラスタを 1 つのアプリケーションまたは密接に関連した一連のアプリケーション専用割り当てることが一般的となっています。専用クラスタを使用することで、リソース管理が簡略化され、アプリケーション障害分離が実現しました。また、比較的低コストのコンピュータハードウェアによってこのアプローチが利用可能となっています。図 1 は、異なるデータベースをサポートする複数のクラスタで複数のアプリケーションに対応するこの一般的なアプローチを示しています。

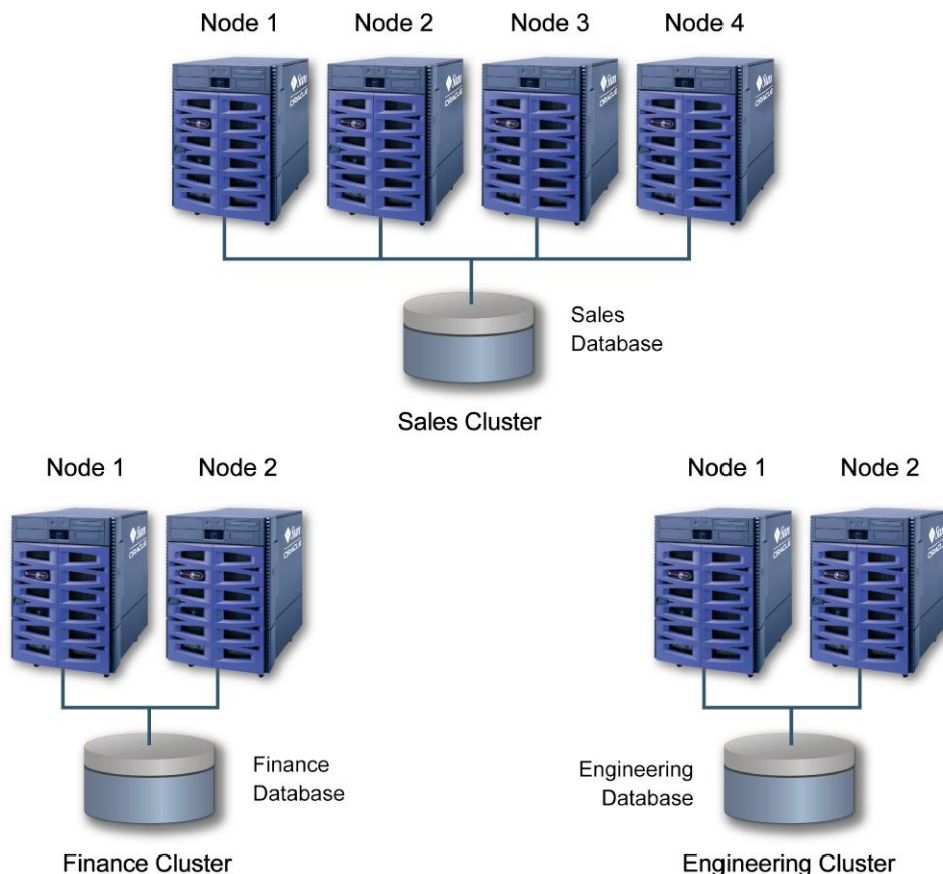


図 1. 構成の例: 異なるデータベースをサポートする複数のクラスタ

ムーアの法則は今なお、コンピュータに適用され、業界はさらに強力なコンピュータを製造しています。プロセッサがより強力になる傾向に伴い、ほかのリソースでも記憶容量の増大やネットワーク帯域幅の拡大などの進歩が見られます。パワーが強大になるにつれて、価格性能比も向上しています。アプリケーションの処理要求は大きくなっていますが、多くの場合、これらの要求が増加率は、システムの処理能力に比べてかなりゆるやかです。その結果、多くのクラスタは、プロセッサ、ストレージ、ネットワークなどのあらゆる領域で処理能力が著しく向上していることから、ほとんどの場合でアイドル状態となっています。

そのように大量の処理能力がアイドル状態になっていることは、より良いシステム利用をはかるための願ってもない機会につながります。組織は、この未使用の能力を再利用する方法を求め、単一のクラスタ上で複数のアプリケーションをホストするアプローチに移行しています。ただし、とくにセキュリティーやリソース管理の領域では、アプリケーション間の相互作用に関する懸念があるため、慎重な姿勢が見受けられます。

仮想化テクノロジーは、このようなセキュリティーの懸念に対処し、単一のハードウェア構成に基づく異なるクラスタで複数のアプリケーションをホストする安全な方法を提供します。

サーバー仮想化テクノロジー

オラクルは、ネットワーク、ストレージ、デスクトップ、サーバー、オペレーティングシステムの仮想化に対応する広範な仮想化テクノロジーを提供しています。このセクションでは、オラクルによるサーバーおよびオペレーティングシステムの仮想化オプションに焦点を当てます。これら仮想化の選択は、単一システム上に複数のアプリケーションをホスティングすることを促進し、さらに以下の点も促進します。

- ハードウェアパーティション
- 仮想マシン (VM)
- オペレーティングシステムの仮想化
- リソース管理

図 2 は、オラクルによって提供されている仮想化テクノロジーの比較結果の要約を示しています。これらのテクノロジーに基づいて、異なるレベルの分離、リソース粒度、柔軟性を実現できます。次のセクションでは、それぞれのテクノロジーの概要を示します。

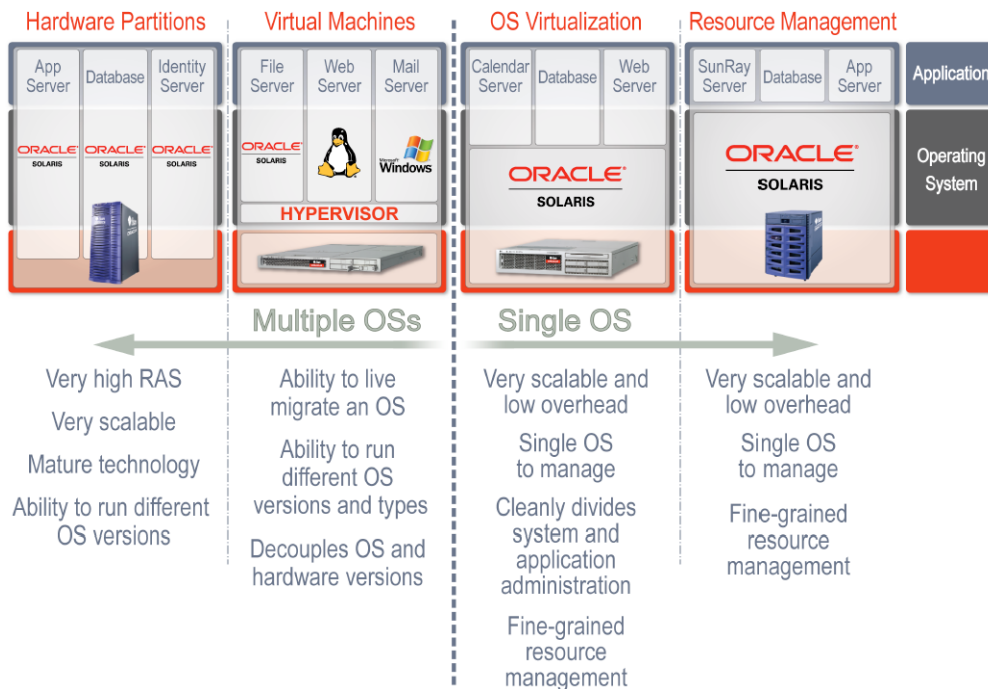


図 2. 仮想化テクノロジーの比較

ハードウェアパーティション

オラクルの select SPARC® Server などのハイエンドシステムによっては、物理的にハードウェアを分離します。その場合は、複数のハードウェアパーティションに構成できます。それぞれのハードウェアパーティションは、ほかのパーティションから電氣的に分離され、独立したオペレーティングシステムのコピーを実行します。そのため、1 つのハードウェアパーティションで発生したハードウェア障害もソフトウェア障害もほかのパーティションに影響することはありません。

ソフトウェアは、それぞれのハードウェアパーティションを、オペレーティングシステムを実行できる独立したコンピュータと見なします。また、それぞれのハードウェアパーティションでは、異なる種類や異なるバージョンのオペレーティングシステムを実行できます。

それぞれのハードウェアパーティションは独立しているため、ハードウェアパーティション間の干渉は起こりません。したがって、アプリケーションに応じたスケラビリティが実現されます。

動的ドメインでは、アプリケーションをシャットダウンすることなく、ハードウェアをハードウェアパーティションに追加したり、ハードウェアパーティションから削除したりできます。システム管理者は、ハードウェアパーティション間でリソースを移動できますが、単一のオペレーティングシステムの下で管理しているときのように簡単にリソースを共有することはできません。

この仮想化アプローチの主な短所は、コストがかかり、結果的にハイエンドシステムに限定されることです。

仮想マシン

仮想マシン (VM) テクノロジーは、異なるベンダーによる複数のオペレーティングシステムをサポートします。物理マシンごとの多くの OS 環境をサポートできます。VM アプローチでは、従来のオペレーティングシステムと物理マシン間に、ハイパーバイザと呼ばれるファームウェア層を配置します。ハイパーバイザは、オペレーティングシステムに対する仮想マシンインタフェースを提供し、仮想マシン内でオペレーティングシステムからの要求を仲介します。したがって、ハイパーバイザは、それぞれが異なるオペレーティングシステムイメージを実行できる複数マシンのように見せる事ができます。

チップマルチスレッディング (CMT) テクノロジー対応の UltraSPARC® T1/T2-T2 Plus ベースサーバーのオンボードハイパーバイザを使用する Oracle VM Server for SPARC (以前の Sun Logical Domains) や Oracle VM Server for x86 などのさまざまな仮想化テクノロジーが存在します。これらのソリューションでは、仮想マシン内のオペレーティングシステムを変更して効率良くハイパーバイザを扱えるようにする準仮想化と呼ばれるテクニックを使用します。他社は、同様に オラクルの Sun x86 サーバーで実行される Hyper-V や VMware などの仮想化テクニックを提供しています。

それぞれの仮想マシンでは、異なる種類とバージョンのオペレーティングシステムを実行できるので、異なるオペレーティングシステムを単一の物理サーバー上で実行可能です。仮想マシンでは、ライブオペレーティングシステム全体を 1 つのマシンから別のマシンに移行することもできます。仮想マシンアプローチでは、ハードウェア障害分離は行えません。ただし、仮想マシンインタフェースによって、オペレーティングシステムに対する汎用マシンインタフェースが提供され、オペレーティングシステムから基本的なハードウェアの詳細を切り離します。これにより、プラットフォームごとにオペレーティングシ

システムを変更する必要性が軽減されます。

仮想マシンアプローチの 1 つの短所は、仲介に必要な作業によって生じるハイパーバイザ層のオーバーヘッドです。このようなオーバーヘッドの一部は、リソースを仮想マシン専用に割り当てることで除去されますが、そのリソースは共有できなくなります。

オペレーティングシステムの仮想化

オペレーティングシステムの仮想化アプローチでは、単一のオペレーティングシステムイメージに基づいて、アプリケーションまたは一連のアプリケーション用の分離環境を作成します。この分野で、オラクルが提供している Oracle Solaris ゾーン¹ は、Oracle Solaris 10 OS に欠かせない要素です。Oracle Solaris ゾーンは、ソフトウェアによって定義された柔軟な境界を使用して、ソフトウェアアプリケーションとサービスを分離し、Oracle Solaris 10 OS の単一インスタンス内で、多くのプライベート実行環境を作成できるようにします。

基礎となる Oracle Solaris OS には、単一の**大域ゾーン**があります。これは、システム用のデフォルトゾーンと、システム全体の管理制御に使用されるゾーンの両方を表します (図 3 を参照)。大域ゾーンのシステム管理者は、非大域ゾーンを 1 つ以上作成でき、これらの非大域ゾーンで利用可能になるすべてのリソースを特定します。非大域ゾーン内のアプリケーションまたはユーザーは、包含するゾーン外で何かを表示したり影響を与えたりできないので、強力なセキュリティが確立されます。

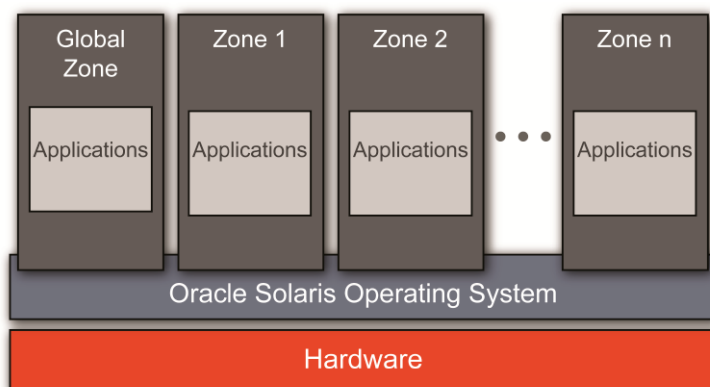


図 3. Oracle Solaris ゾーン

ゾーンはリソース管理の粒度の単位です。ゾーン内のアプリケーションでは、ゾーンに割り当てられたリソースを共有する必要があります。システム管理者は、リソースをゾーン専用に割り当てたり、ゾーンによって使用されるリソースの特定の共有を付与したりできます。たとえば、システム管理者は 50 の Fair Share Scheduler の配分をゾーン A に割り当て、30 をゾーン B、20 をゾーン C に付与できます。その結果、ゾーン A は CPU リソースの 50% を取得し、ゾーン B は 30%、ゾーン C は 20% を取得します。

ゾーンは障害分離の単位を提供します。たとえば、1 つのゾーン内で再起動を試行すると、マシンの再起動ではなくゾーン再起動が実行されます。この分離によって、アプリケーションの動作または障害の影響がほかのゾーン内のアプリケーションに及ばないように簡単に操作できます。

¹ Oracle Solaris Containers では、Oracle Solaris ゾーンによって提供される境界分離と、システムリソース制御を組み合わせます。ただし、ゾーンはリソース管理から独立して使用できます。多くのユーザーは、ゾーンおよびコンテナという用語を同じ意味で使用しています。

複数のオペレーティングシステムを使用するテクノロジーと比較すると、Oracle Solaris ゾーンモデルを使用した場合は、管理対象のオペレーティングシステムが 1 つのみなので、管理オーバーヘッドが小さくなります。また、Oracle Solaris ゾーンは、ハイパーバイザによるオーバーヘッドがなく、特殊なハードウェアを要求しません。また、Oracle Solaris 10 などのあらゆるプラットフォームで利用できます。Oracle Solaris ゾーンは、アプリケーションを分離しますが、オペレーティングシステム障害やハードウェア障害は分離しません。

リソース管理

リソース管理は、アプリケーションやプロジェクトと呼ばれるアプリケーションの集合によって使用されるリソースを管理する Oracle Solaris OS の機能です。Oracle Solaris OS の単一コピー内で、リソースをゾーン、アプリケーションまたはプロジェクト専用に使えます。したがって、CPU、スワップ、メモリなどのリソースを特定のプログラムに割り当てることができます。ただし、すべてのアプリケーションが相互に認識し、影響を受ける場合があります。つまり、リソース管理では、セキュリティーまたはアプリケーション障害分離を行えません。

サーバー仮想化アプローチの選択

高可用性という観点で仮想化アプローチの選択を促す重要な要因として、次のものがあります。

- 電氣的分離
- オペレーティングシステムの障害分離
- 異なるオペレーティングシステムバージョンのサポート
- アプリケーション障害分離
- セキュリティー分離
- リソース管理

ユーザーが仮想化を採用する気持ちになる最も一般的なシナリオの 1 つとして、複数の組織をサポートするアプリケーションの統合要求があります。これは、このレポートの重要な焦点でもあります。

このシナリオで、仮想化アプローチがどのように選択されるかを検討してみましょう。

- 最新のハードウェアは高レベルの信頼性を確立しています。また、クラスタの高可用性機能により、システムは一般的な多くのハードウェア障害を克服できるようになっています。したがって、電氣的分離は必須ではありません。
- Oracle Solaris オペレーティングシステムは、信頼性が高く、広範なアプリケーションをサポートできます。つまり、単一のオペレーティングシステムイメージによって、このシナリオでのほとんどの顧客のニーズが満たされます。
- マシンを再起動することによってエラーから回復する重要なアプリケーションが存在します。したがって、1 つの組織のアプリケーションがほかの組織のアプリケーションに影響しないようにするには、アプリケーション障害分離が必須です。
- 異なるそれぞれの組織は、そのデータの非公開性を維持する必要があります。したがって、セキュリティー分離が必須となります。

- 通常の場合、それぞれの組織はコンピュータサービスに応じて課金されます。したがって、それぞれの組織は、支払った分の保証が得られるようにする必要があります。この保証は、リソース管理によって得られます。

ゾーンは、十分に統合された機能セットでオーバーヘッドを低く抑えながらこのシナリオの要求を満たします。このホワイトペーパーでは、オペレーティングシステムの仮想化テクノロジーアプローチに焦点を絞っていきます。

ゾーンクラスタの概要

これまでに説明した仮想化テクノロジー製品は、いずれも単一マシン製品です。ゾーンクラスタは、Oracle Solaris ゾーンの原理を拡張し、クラスタ全体で動作するようにして、アプリケーションをサポートできるようにします。

クラスタ用語

Oracle Solaris 10 OS が変更されたため、Oracle Solaris ホスト、クラスタ、クラスタノードなどの基本的なクラスタの概念を見直す必要があります。

- Oracle Solaris ホスト-Oracle Solaris ホストは、1 つの Oracle Solaris イメージと 1 つの Oracle Solaris Cluster イメージのみをサポートする構成です。次のエンティティは、1 つの Oracle Solaris ホストと見なすことができます。
 - 仮想マシンまたはハードウェアドメインとともに構成されていない「ベアメタル」物理マシン
 - Oracle VM Server for SPARC ゲストドメイン
 - Oracle VM Server for SPARC I/O ドメイン
 - ハードウェアドメイン
- クラスタノード-クラスタノードには 2 つのプロパティがあります。つまり、クラスタノードはクラスタアプリケーションをホストするとともに、クラスタメンバーシップに投票するクラスタのメンバーになることができます。
- クラスタ-クラスタは、1 つ以上のクラスタノードのコレクションです。それらのノードは排他的にそのコレクションに属します。

Oracle Solaris 10 には、ゾーンの概念が導入されています。Oracle Solaris 10 は、大域ゾーンまたはそれ以外のゾーンである 1 つのゾーン内ですべてのアプリケーションを実行します。クラスタアプリケーションは常にゾーンで実行されるため、クラスタノードは常にゾーンに一致します。

1 つの物理マシンで複数の Oracle Solaris ホストをホストできるようになったので、単一マシン上に複数の Oracle Solaris 10 OS イメージを存在させることができます。単一の Oracle Solaris 10 OS イメージには、1 つの大域ゾーンのみがあり、異なるゾーンクラスタに属す任意の数のゾーンが存在できます。したがって、単一の Oracle Solaris ホストで、複数のクラスタノードをサポートできます。ただし、単一の Oracle Solaris ホスト上の各クラスタノードは、独立したクラスタに属します。

グローバルクラスタとゾーンクラスタ

Oracle Solaris 10 では、グローバルクラスタ、ゾーンクラスタという 2 種類のクラスタを構成できます。

- グローバルクラスタ-グローバルクラスタは、Oracle Solaris ホストのコレクションですべての大域ゾーンを包含します。グローバルクラスタには、オプションでメンバーシップ投票を行わない非大域ゾーンを含めることができます。
- ゾーンクラスタ-ゾーンクラスタは、すべてクラスタブランドのゾーンである 1 つ以上の非大域ゾーンで構成されます。ゾーンクラスタの各クラスタノードは、異なる Oracle Solaris ホスト上に存在します。

ゾーンクラスタノードでは、その同じ Oracle Solaris ホスト上の大域ゾーンをクラスタモードで起動して、ゾーンクラスタノードを動作可能にする必要があります。すべてのゾーンクラスタノードは、同じグローバルクラスタに属す Oracle Solaris ホスト上に存在する必要があります。ゾーンクラスタノードは、その同じグローバルクラスタに対応する Oracle Solaris ホストのサブセットとして存在できます。ゾーンクラスタはグローバルクラスタに依存しますが、グローバルクラスタはどのゾーンクラスタにも依存しません。

図 4 は、Oracle Solaris クラスタによってサポートされる 4 台のマシンハードウェア構成を示しています。

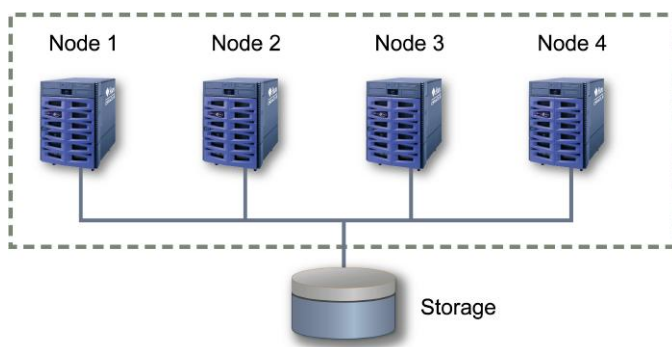


図 4. クラスタ用の 4 台のマシンハードウェア構成

図 5 は、その同じ 4 台のマシンハードウェア構成で動作する複数のクラスタを示しています。重要なポイントは、クラスタノードは、必ずしもハードウェアとの 1 対 1 の関係があるとは限らないクラスタソフトウェア構造だということです。[Sun Cluster が Solaris 9 で動作している場合、ホストとクラスタノードは同じものでした。また、ハードウェアドメインを除いて、クラスタノードは物理マシンでもありました。この従来のクラスタはわかりやすい可能性がありますが、柔軟性に欠ける短所もありました。]

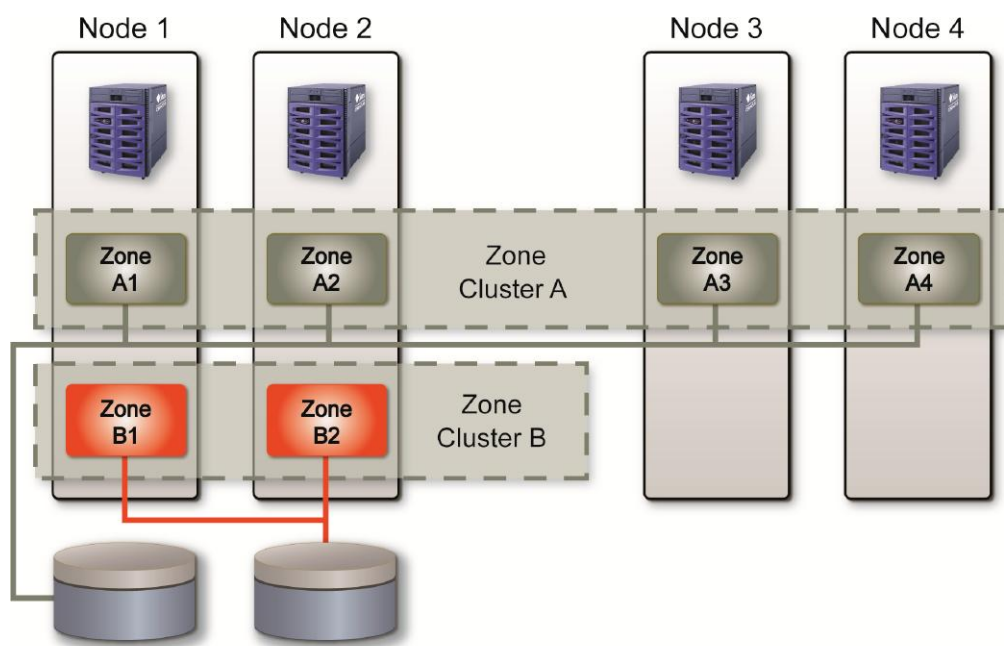


図 5.4 4 台のマシン構成に配備された複数のクラスタ

ゾーンクラスタは、仮想クラスタと見なすことができます。ゾーンクラスタは、ゾーンクラスタが従来のクラスタであるという錯覚をクラスタアプリケーションに提示します。

次のセクションでは、ゾーンクラスタの機能について詳細に説明し、このテクノロジーで可能となるさまざまなユースケースを提供します。

セキュリティー分離

クラスタアプリケーションには、フェイルオーバー、スケーラブルという 2 つの基本タイプがあります。フェイルオーバーアプリケーションは、1 回に 1 つのノードのみで実行できる単一インスタンスのアプリケーションです。システムは、ノード障害が発生した場合または管理要求が提出された場合に、フェイルオーバーアプリケーションを別のノード上で再起動します。スケーラブルアプリケーションは、異なるノード上で実行されている複数のアプリケーションインスタンスで構成されます。

Oracle Solaris ゾーンは、1 台のマシン上のみで作成され存在します。したがって、現在の Oracle Solaris ゾーンを、フェイルオーバーまたはスケーラブルアプリケーションのセキュリティーコンテナとして使用することはできません。このホワイトペーパーでは、とくにフェイルオーバーアプリケーションおよびスケーラブルアプリケーション用のセキュリティーコンテナとして使用するクラスタ全体ゾーンを紹介し、ゾーンクラスタは、クラスタ全体ゾーンの実装に対応します。

クラスタ全体ゾーンは、フェイルオーバーアプリケーションとスケーラブルアプリケーションの両方に使用できるセキュリティーコンテナです。包含するクラスタ全体ゾーンの外でフェイルオーバーまたはスケーラブルアプリケーションのインスタンスを実行することはできません。つまり、すべてのスケーラブルアプリケーションのインスタンスは常に単一のクラスタ全体ゾーン内で実行され、フェイルオーバーアプリケーションをクラスタ全体ノードの外でスイッチオーバーしたりフェイルオーバーしたりすることはできません。

ゾーンクラスタはセキュリティーのゾーンモデルに従います。ゾーンクラスタ内から表示したり影響を与えたりできる唯一のリソースは、システム管理者が大域ゾーンからゾーンクラスタに構成したリソースです。ゾーンクラスタ内からリソースをゾーンクラスタに追加することはできません。同じように、ゾーンクラスタ内からゾーンクラスタ構成パラメータを変更することはできません。たとえば、ゾーンクラスタに割り当てられた CPU の共有をゾーンクラスタ内から変更することはできません。

アプリケーション障害分離

オペレーティングシステムは、一定レベルのアプリケーション障害分離に対応します。たとえば、1 つのアプリケーションインスタンスで問題が発生しても、ほかのすべてのアプリケーションで障害が発生することはありません。ただし、ほかのアプリケーションの障害の原因になるアクションがアプリケーションによって実行される場合があります。たとえば、アプリケーションは再起動の指示をノードに送信できますが、これは明らかにそのノード上のすべてのアプリケーションに影響します。

Oracle Solaris ゾーン機能セットは、1 つのアプリケーションの誤動作がほかのアプリケーションに影響する可能性を軽減できるように設計されています。上記の同じ例の場合、システムはゾーン内で発行された再起動コマンドを「ゾーン再起動」コマンドと見なし、再起動コマンドがそのゾーン外のアプリケーションに影響しないようにします。

Oracle Solaris ゾーンでは、そのゾーン外のほかのアプリケーションに悪影響を及ぼす可能性がある多くの操作が無効になります。ゾーンクラスタは、この原則のサポートを維持しています。

リソース管理

Oracle Solaris オペレーティングシステムには、長期に渡ってリソース管理サブシステムが採用されています。管理者は、Oracle Solaris リソース管理ソフトウェアを使用して、特定の目的専用リソースを割り当てたり、リソースタイプの特定の共有をプロジェクトまたはアプリケーションに割り当てたりできます。

Oracle Solaris リソースマネージャーソフトウェアには、ゾーンレベルでリソースを管理するための一定レベルの粒度が追加されています。大域ゾーンのシステム管理者は、ゾーンのリソース利用を管理できますが、オペレーティングシステムでは、これらの制御をゾーン内から変更できないようにします。これは、とくに Oracle Solaris Cluster アプリケーションを単一システム上で統合するときに重要になります。ゾーンクラスタはこのゾーン機能を保持します。

専用クラスタモデル

ゾーンは、ゾーンがそのゾーン内のアプリケーション専用のマシンであるという錯覚を単一マシンアプリケーションに与えます。ゾーンクラスタは、ゾーンクラスタがそのゾーンクラスタ内のクラスタアプリケーション専用のクラスタであるという錯覚をクラスタアプリケーションに与えます。同じように、ユーザーはゾーンクラスタにログインすると、ゾーンクラスタを従来のクラスタとして認識します。

ゾーンクラスタは簡略化されたクラスタです。必要最低限のアプローチが採用され、次のようなクラスタアプリケーションの直接サポートに必要なコンポーネントのみが存在します。

- ファイルシステム
- ストレージデバイス
- ネットワーク
- クラスタメンバーシップ

クォーラムデバイスやハートビートなどの、クラスタアプリケーションで必要とされないクラスタコンポーネントは存在しません。ゾーンクラスタは、物理システムのすべての面を模倣するわけではありません。たとえば、ゾーンクラスタは、ゾーンクラスタのゾーン内でネストされているゾーンをサポートしません。

ゾーンクラスタのユースケース

このセクションでは、次のようなさまざまなユースケースを検討し、ゾーンクラスタのユーティリティを紹介합니다。

- 複数組織統合
- 機能統合
- 多層統合
- コスト削減
- 管理ワークロードの軽減

複数組織の統合

複数の組織のクラスタアプリケーションを統合するソリューションは、次の重要な要件を満たしている必要があります。

- **セキュリティ分離**-ソリューションでは、異なる組織のアプリケーションとユーザーが、相互に表示したり影響を与えたりできないようにする必要があります。それぞれの組織は、自己の情報の非公開性を維持することを強く求めます。
- **アプリケーション障害分離**-ソリューションでは、1つのアプリケーションの障害の影響がほかの領域内のアプリケーションに及ばないようにする必要があります。それぞれの組織は、そのスケジュールにほかの組織の問題の影響が及ばないことを要求します。
- **リソース管理**-ソリューションでは、それぞれの組織のアプリケーションによるリソースの利用を制御する必要があります。コンピュータリソースは無償では得られません。コストを異なる組織に配分する必要があります。また、それぞれの組織は支払った分のリソースを確実に受け取ることを要求します。

ゾーンクラスタはこれらのコア要求を満たします。図 6 は、1つの 4 ノード物理クラスタ上の 2 つの異なる組織からデータベースを統合する例を示しています。この例では、Sales ゾーンクラスタが 4 台のマシンにまたがり、Finance ゾーンクラスタが 2 台のマシンにまたがっています。

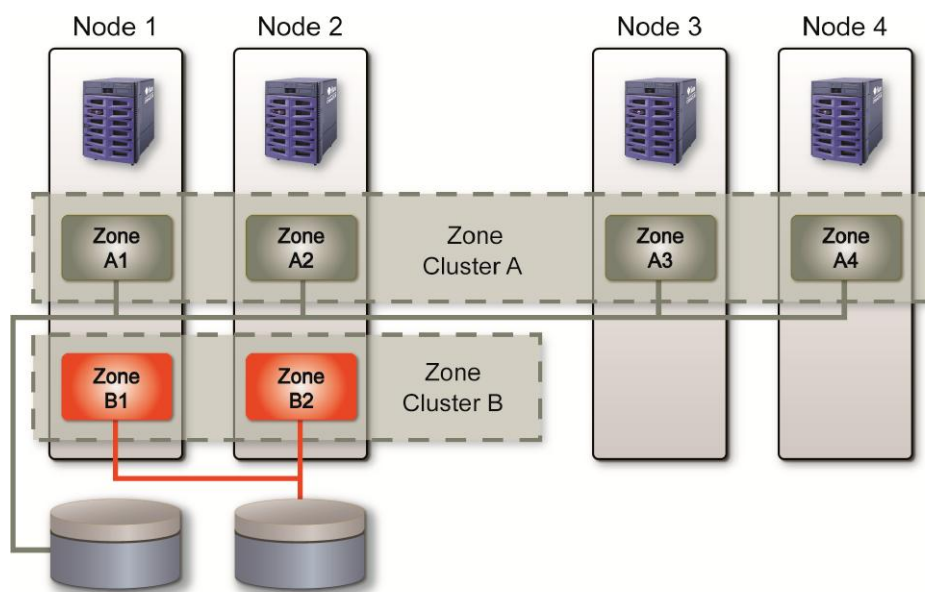


図 6.2 2つの組織のアプリケーションの統合例

機能統合

多くの組織は、異なる目的に合わせて異なる専用クラスタを割り当てています。最も一般的な業務分担の一例を次に示します。

- 製造
- テスト
- 開発

より慎重な組織は、その製造システムに異質のアクティビティが加わることのリスクを嫌う場合がありますが、多くの組織は、単一の物理クラスタ上でテストアクティビティと開発アクティビティを統合することに積極姿勢を示す可能性があります。図 7 は、ゾーンクラスタを使用して、単一の物理クラスタ上でテストおよび開発アクティビティが統合されている例を示しています。

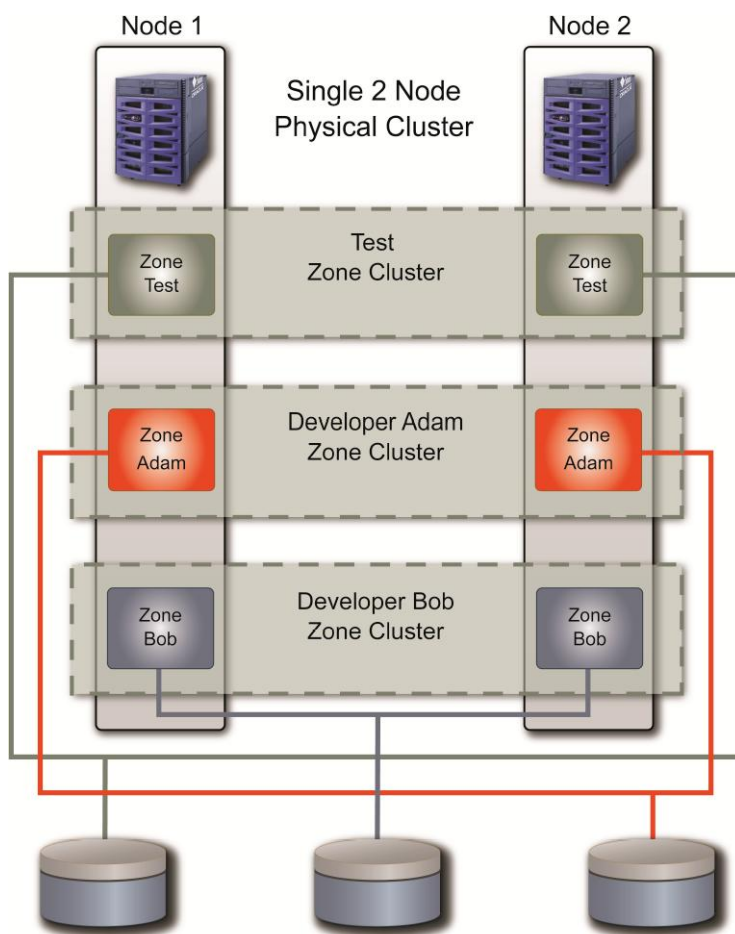


図 7. 機能統合の例

機能統合での重要な要素は、アプリケーション障害分離とリソース管理です。

開発者にとっては、物理クラスタ上で専用時間をスケジュールすることが一般的となっています。これは、ほかの開発者のアクティビティが作業に影響する可能性があるためです。ゾーンクラスタを使用すると、異なるゾーンクラスタ内のアプリケーションを、ほかのゾーンクラスタ内のアプリケーションの動作から独立して操作できます。ほかのゾーンクラスタ内でアプリケーション障害が発生しても、包含するゾーンクラスタを超えてアプリケーションに影響が及ぶことはありません。したがって、エンジニアは、ほかのゾーンクラスタに影響を与えずに、ゾーンを再起動したり、1 つのゾーンクラスタ内でアプリケーションをスイッチオーバーしたりできます。

システム管理者は、ほかのゾーンクラスタに影響を与えずに、ゾーンクラスタの作成または廃棄を動的に実行できます。ゾーンクラスタ数の制限は、8,192 ゾーンクラスタという非常に大きな値なので、ゾーンクラスタの数は実質的に無制限と言えます。これらのファクタの組み合わせにより、組織は各開発者用にゾーンクラスタを構成できます。また、それぞれの開発者にとっては、並行作業が可能になります。各開発者用に 1 つ以上のゾーンクラスタを構成することで、開発者は専用クラスタ時間 (多くの場合は不便な閑散時などに) をスケジュールする必要がなくなります。したがって、開発時間の短縮が促進される可能性があります。

クラスタアプリケーションによっては、クラスタアプリケーションを専用クラスタで実行することが想定されます。ゾーンクラスタでは、このクラスの複数のクラスタアプリケー

ションを実行できます。この機能は、とくにこのクラスのアプリケーションで変更またはリリースレベルをテストするときに重要となります。

多層統合

一般的な 3 層データセンターモデルでは、次の層が識別されます。

- フロントエンド
- アプリケーション
- データベース

ゾーンクラスタは、3 つのすべての層のアプリケーションの統合をサポートします。図 8 は、フロントエンド層にスケーラブルな Apache Web サーバー、アプリケーション層に Java™ 2 Platform, Enterprise Edition (J2EE) アプリケーションサーバー、データベース層に RAC データベースを使用した例を示しています。すべての層で、独立したゾーンクラスタを使用します。

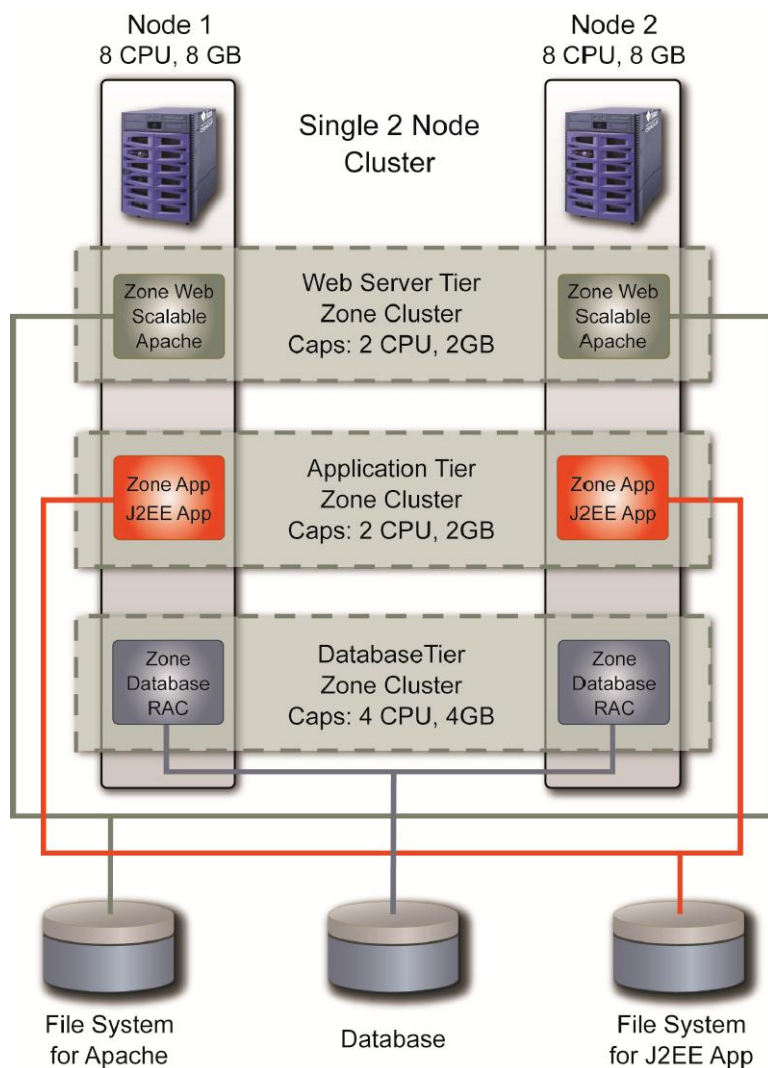


図 8. 多層統合の例

コスト削減

それぞれのゾーンクラスタは、リソース管理の粒度を提供します。組織は、ゾーンクラスタのこの基本機能を使用して、コストを制御できます。多くのソフトウェアベンダーは、CPU 数によって表されるシステムの処理能力に基づいた価格設定モデルを使用しています。管理者は、特定の CPU または特定の数の CPU を使用するようにゾーンを構成できます。アプリケーションがゾーンで実行される場合、そのアプリケーションでは、アプリケーションを包含しているゾーンで利用可能な CPU のみを使用できます。CPU に関するこのいずれかの制限を使用してアプリケーションライセンス料金を判定できるので、著しい節減につながる可能性があります。

図 8 の 2 ノードクラスタと同じようにそれぞれのマシンが構成される 4 ノードクラスタの例を検討してみましょう。それぞれのマシンには 8 個の CPU が搭載されていることを想定します。物理クラスタは、3 つのソフトウェアアプリケーションをホストします。特定製品の料金指定に関する法的問題を避けるため、この例では、フロントエンド層が AppFrontEnd アプリケーション、アプリケーション層が AppApp、データベース層が AppDB をホストすることを想定します。

ソフトウェアライセンスに伴う次の架空料金を想定します。

- AppFrontEnd Per CPU 料金 (FEfee): CPU 当たり 1 万ドル
- AppAPP Per CPU 料金 (APfee): CPU 当たり 1 万 5000 ドル
- AppDB Per CPU 料金 (DBfee): CPU 当たり 2 万ドル

価格設定の式は次のようになります。

$$(FEfee + APfee + DBfee) \times \text{ノード当たりの CPU 数} \times \text{ノード数}$$

この構成の場合の総ライセンス費用は、次のようになります。

$$(1 \text{ 万ドル} + 1 \text{ 万 } 5000 \text{ ドル} + 2 \text{ 万ドル}) \times 8 \times 4 = 144 \text{ 万ドル}$$

ここで、同じアプリケーションを使用する同じ物理クラスタについて検討してみましょう。ただし、ここではそれぞれのアプリケーションに独立したゾーンクラスタを使用します。この構成では、AppFrontEnd をホストするゾーンクラスタに CPU を 2 個、AppApp をホストするゾーンクラスタに CPU を 2 個、AppDB をホストするゾーンクラスタに CPU を 4 個割り当てます。

この構成の場合の価格設定の式は次のようになります。

$$((FEfee \times \text{CPU 数}) + (APfee \times \text{CPU 数}) + (DBfee \times \text{CPU 数})) \times \text{ノード数}$$

この構成の場合の総ライセンス費用は、次のようになります。

$$((1 \text{ 万ドル} \times 2) + (1 \text{ 万 } 5000 \text{ ドル} \times 2) + (2 \text{ 万ドル} \times 4)) \times 4 = 52 \text{ 万ドル}$$

この例では、2/3 近くコストが節減されています。上記の単純なコストの式を使用して、ゾーンクラスタを使用するときのコスト節減を計算できます。同じクラスタ上で複数のアプリケーションを実行するときは、それぞれのアプリケーションが一部の CPU 上で実行されます。この状況では、単一のアプリケーションにすべての CPU が使用されるわけではありません。したがって、そのアプリケーションに使用されない CPU に対してライセンス料金を支払う必要はなくなります。このように、組織がすでに単一の物理クラスタ上

でクラスタアプリケーションを統合している場合でも、ゾーンクラスタを使用すると、著しいコスト削減が実現する可能性があります。

コスト削減機能は、Oracle Solaris 10 の初期リリース以来、Oracle Solaris リソース管理ソフトウェアを使用することによって広く認められるようになってきました。最も初期のアプローチでは、管理者がプール用のプロセッサセットを定義し、そのプールにゾーンを割り当てていました。その結果、そのゾーン内のすべてのアプリケーションがそれらの特定の CPU に制限されます。現在は、さらに簡単にこのような操作を行えます。管理者は、ゾーンを構成するときに CPU の数を指定するか、特定の CPU を選択できるようになったのです。

注:-このコスト削減ポリシーの詳細については、41 ページの「関連資料」を参照してください。

管理ワークロードの軽減

ゾーンクラスタは、グローバルクラスタよりもかなり単純です。たとえば、クォーラムデバイスは不要なため、ゾーンクラスタに定足数デバイスは存在しません。グローバルクラスタとそのグローバルクラスタ上のすべてのゾーンクラスタによって、同じオペレーティングシステムが共有されます。

オペレーティングシステムの更新およびパッチを適用する必要があるのは、グローバルクラスタの場合もそのすべてのゾーンクラスタの場合も、1 回のみです。これはつまり、ゾーンクラスタをグローバルクラスタの代わりに使用できる場合に、管理タスクが軽減されることを表します。

この原理の一例として、多くのデータベースが存在する大規模な金融機関を検討してみましょう。この金融機関は、クラスタ当たり 6~8 のデータベースを統合することを計画していて、クラスタのサイズは最大で 16 ノードになります。ノードの総数はほぼ 200 に到達します。金融機関は、データベースごとにゾーンクラスタを使用する予定です。このアプローチを選択した 1 つの動機は、一連のグローバルクラスタに対し一連のゾーンクラスタでは管理作業が著しく軽減されることです。

ゾーンクラスタの設計

前のセクションで、ゾーンクラスタの概念について概要を示し、ゾーンクラスタの使用法について説明しました。このセクションでは、高レベルでのゾーンクラスタ設計について説明し、何がどのようにサポートされるかを示します。

仮想ノード

ゾーンクラスタは一連のゾーンで構成され、それぞれのゾーンが仮想ノードを表します。ゾーンクラスタの各ゾーンは、独立したマシン上で構成されます。そのため、ゾーンクラスタ内の仮想ノード数の上限は、グローバルクラスタ内のマシン数に制限されます。

Oracle Solaris は、*BrandZ* フレームワークを使用して、ゾーンの変更と拡張をサポートします。ゾーンクラスタ設計には、クラスタブランドと呼ばれる新しいゾーンのブランドが導入されています。クラスタブランドは、元のネイティブなブランドタイプに基づいていて、クラスタの機能を強化します。*BrandZ* フレームワークは、ほかのソフトウェアがゾーンのブランドタイプに応じてアクションを実行できるさまざまなフックを提供します。たと

例えば、ゾーンの起動時に呼び出すソフトウェアのフックがあり、ゾーンクラスタではこのフックを利用して、仮想ノードが起動されたことをクラスタソフトウェアに通知します。

注:-ゾーンクラスタでは BrandZ フレームワークを使用するので、Oracle Solaris 10 5/08 が最低限必要です。

ユーザーは、ゾーンのブランドタイプについて意識する必要はありません。顧客およびアプリケーションの視点から見ると、クラスタブランドゾーンは、クラスタサポートが有効になっているネイティブゾーンと同じような感覚で動作します。

クラスタメンバーシップ

それぞれのゾーンクラスタには固有のメンバーシップの概念があります。ゾーンクラスタのメンバーシップの形式は、ゾーンなしで Solaris 9 OS 上で動作するクラスタの場合と同じです。ゾーンクラスタで実行されるアプリケーションは、大域ゾーンで実行される場合と同じ種類の情報を受け取ります。つまり、メンバーシップに関して、アプリケーションはゾーンクラスタおよび大域ゾーンで同様に実行されます。

必然的に、ゾーンクラスタのゾーンは、ホストマシン上の大域ゾーンが操作可能になるまでは操作できません。ゾーンクラスタのゾーンを起動するには、大域ゾーンがクラスタモードで起動されている必要があります。また、ゾーンクラスタのゾーンは、マシンの起動後に自動的に起動されるように構成するか、または管理者がゾーンの起動タイミングを手動で制御できます。ゾーンクラスタのゾーンは、エラーが発生する場合があります。また、管理者が手動でゾーンを停止したり再起動したりできます。これらのイベントが発生すると、いずれの場合もゾーンクラスタでそのメンバーシップが自動的に更新されます。

メンバーシップの監視

システムはゾーンクラスタのメンバーシップ情報を保持します。それぞれのマシンは、ゾーンクラスタメンバーシップモニター (ZCMM) と呼ばれるコンポーネントをホストします。このコンポーネントによって、そのマシン上のすべてのクラスタブランドゾーンの状態が監視されます。ZCMM は、どのゾーンがどのゾーンクラスタに属するかを認識します。

最初に、ノード参加またはノード離脱によりグローバルクラスタノードのメンバーシップが変更されるケースを検討しましょう。大域ゾーン内のノード再構成プロセスによって、新しいクラスタメンバーシップが判定されます。マシンの再構成プロセスが完了すると、システムによって、新しい ZCMM リーダーが選択されます。以前の ZCMM リーダーが存続している場合は、変更はありません。それ以外の場合は、システムによって任意に 1 つの ZCMM がリーダーとして選択されます。ZCMM リーダーは、それぞれのノード上の ZCMM からゾーンクラスタ仮想ノード情報を収集し、その情報をコンパイルした後、新しいメンバーシップをすべての ZCMM に配信します。

ゾーンクラスタ仮想ノードの状態が変更された場合は、そのマシン上の ZCMM によってその情報が ZCMM リーダーに転送されます。これにより、ゾーンクラスタの再構成がトリガーされます。ZCMM リーダーは、そのゾーンクラスタの新しいメンバーシップ情報を、各マシン上の ZCMM に配信します。

この設計では、ゾーンクラスタのメンバーシップ情報が迅速に更新されて配布されます。クラスタ経験があるユーザーの方は、グローバルクラスタの再構成後はグローバルクラスタが回復するまでに時間がかかることをご存知でしょう。グローバルクラスタノードの再

構成後に、ゾーンクラスタメンバーシッププロセスが開始され、グローバルクラスタの回復プロセスが完了するよりもかなり前に完了します。これにより、ゾーンクラスタメンバーシップを更新する際の著しい遅延が回避されます。

セキュリティ

ゾーンクラスタのセキュリティ設計は、Oracle Solaris ゾーン機能のセキュリティ設計に従っています。ゾーンはセキュリティコンテナです。オペレーティングシステムは、ファイルシステム、デバイス、ネットワークなどのリソースへのすべてのアクセス要求をチェックして、そのようなアクセス許可がそのゾーンに付与されているかどうかを判定します。アクセス権が付与されていないと、オペレーティングシステムはアクセスを拒否します。アプリケーションでは、システム呼び出しやドアなどの限られた数の通信チャンネルを通じて、ソフトウェアに要求を送信できます。オペレーティングシステムは、要求の送信元のゾーンを識別する情報を使用して、それぞれの通信にタグを付けます。カーネルまたは大域ゾーン内のソフトウェアは、Trusted と見なされ、送信元ゾーンに基づいてアクセス権をチェックして、未許可のアクセスを拒否する役割を担います。非大域ゾーン内のアプリケーションは、カーネルまたは大域ゾーン内のソフトウェアに干渉できません。総合的な結果として、システムはアプリケーションによるアクセスを、許可された項目のみに制限します。

図 9 は、ゾーンクラスタの全体的なセキュリティアーキテクチャーを示しています。

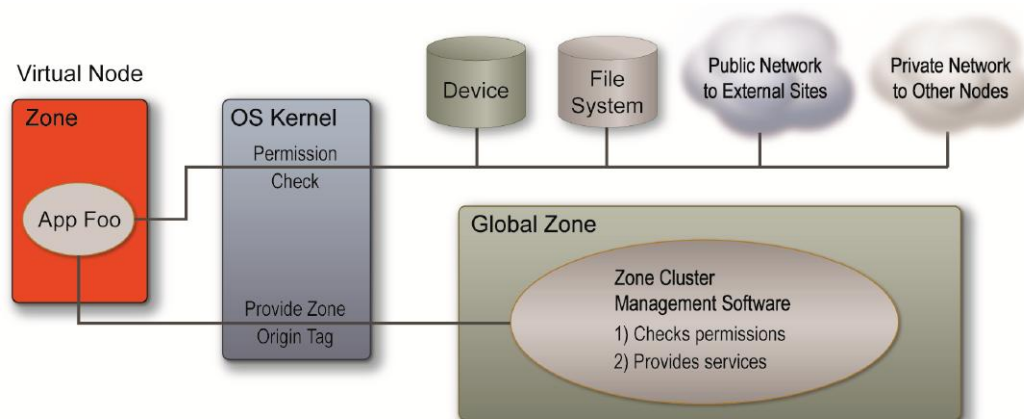


図 9. ゾーンクラスタのセキュリティアーキテクチャー

ゾーンクラスタのゾーンコンポーネントは、セキュリティの観点から見ると等価です。この設計により、ゾーンクラスタのすべてのゾーンコンポーネントで、権限に関連したパラメータが同じになります。たとえば、ゾーン名とゾーンルートパスは、どのゾーンコンポーネントでも同じです。

Solaris Trusted Extensions のサポート

軍や諜報機関などの政府機関および大規模な金融機関は、非常に厳密なセキュリティポリシーを持っています。さまざまなセキュリティラベルを使用してデータの分類（または機密扱い）を行い、ユーザーはそのようなデータに対し粒度のあるアクセス権を与えられます。従来、このような要件は「Multi-Level Security MLS」と呼ばれています。Oracle Solaris Trusted Extensions は、Multi-Level Secure (MLS) オペレーティングシステムです。

Trusted Extensions は、単一マシンセキュリティコンテナの概念に基づいていて、Solaris Containers をセキュリティコンテナとして使用します。ただし、単一のマシンがシングルポイント障害になります。複数のシステム上で実行されるゾーンのグループは、単一セキュリティコンテナとして同時に動作でき、しかも可用性の高いプラットフォームを提供します。そのような単一のクラスタ全体セキュリティコンテナに属しているプロセス、ファイルシステム、共有デバイス、ネットワークは、障害時または管理者要求に従い必要に応じて 1 つのシステムから別のシステムに移動できますが、常にその「クラスタ全体セキュリティコンテナ」の境界内に留まります。したがって、高可用性とマルチレベルセキュリティの両方が実現します。Oracle Solaris Cluster と Trusted Extensions の機能を組み合わせることで、このようなクラスタ全体セキュリティコンテナを提供できます。

ファイルシステム

ゾーンクラスタは、ローカル、高可用性、クラスタ、NFS ファイルシステムなどのさまざまな種類のファイルシステムへのアクセスをサポートします。以降のサブセクションでは、各種のファイルシステムがどのようにゾーンクラスタでサポートされるかについて説明します。

ローカルファイルシステム

ゾーンクラスタの仮想ノードには、その特定の仮想ノードのみで使用できるファイルシステムへのアクセス権を与えることができます。これはローカルファイルシステムと呼ばれます。ローカルファイルシステムのゾーンクラスタサポートのルールは、ネイティブゾーンの場合と同じです。この機能の場合、ゾーンクラスタは基本的なゾーンファイルシステムサポートに依存します。ゾーンクラスタのローカルファイルシステムサポートには、Oracle Solaris Zettabyte File System (ZFS)、UFS、Veritas VxFS、Sun QFS が含まれます。

高可用性ファイルシステム

高可用性ファイルシステムは、複数の仮想ノードから利用できますが、1 回に 1 つの仮想ノードのみにマウントされます。クラスタでは、仮想ノード障害の発生時または管理者が実行する手動スイッチオーバーコマンドに応じて、高可用性ファイルシステムを別の仮想ノードにマウントします。高可用性ファイルシステムは、いったんマウントされると、ローカルファイルシステムとまったく同様に動作します。

ゾーンクラスタでは、高可用性ファイルシステムを読み取りおよび書き込み権限を使用してマウントするときに、仮想ノードに直接マウントします。ある時点で特定の高可用性ファイルシステムにアクセスできるのは 1 つのゾーンクラスタのみです。ゾーンクラスタでは、読み取り権限のみが設定されたループバックマウントを使用して、仮想ノード用に高可用性ファイルシステムをマウントできます。読み取り専用権限が設定された高可用性ファイルシステムは、複数のゾーンクラスタで共有できます。

ゾーンクラスタは、HASStoragePlus サブシステムを使用して、高可用性ファイルシステムをサポートします。ゾーンクラスタは、高可用性ファイルシステムに関する情報を、Cluster Configuration Repository に記録します。また、ゾーンクラスタは、管理者が高可用性ファイルシステムへのアプリケーションの依存性を指定したとき、および実際にファイルシステムをマウントするときにアクセス権を検証します。

ゾーンクラスタの高可用性ファイルシステムサポートには、Oracle Solaris ZFS、UFSm、Veritas VxFS、Sun QFS が含まれます。

クラスタファイルシステム

ゾーンクラスタでは、2つの方法を使用して、クラスタファイルシステム（グローバルファイルシステムと呼ばれる場合もある）を、仮想ノードで実行されるアプリケーションで利用できるようにします。

1つ目の方法では、ループバックマウントとは対照的に、クラスタファイルシステムをすべての仮想ノードに直接マウントします。このアプローチは、共有 QFS ファイルシステムを使用するときに採用されます。共有 QFS ファイルシステムは、大域ゾーンで実行されるスケラブルマウントポイント (SUNW.ScalMountPoint)_リソースの制御下にあります。このリソースは、仮想ノードのゾーンパスを基準にした場所にファイルシステムをマウントします。この方法で共有 QFS ファイルシステムをマウントする場合は、ある時点で1つのゾーンクラスタのみにマウントできます。

2つ目の方法では、ループバックマウントを使用して、すべての仮想ノードにクラスタファイルシステムをマウントします。このアプローチは、プロキシファイルシステムを使用するときに採用されます。プロキシファイルシステムテクノロジーでは、UFS および Veritas VxFS ファイルシステムを、グローバルファイルシステムとして使用できます。UFS または VxFS クラスタファイルシステムは、ゾーンクラスタで実行される HAStoragePlus (SUNW.HAStoragePlus) リソースの管理下にあります。このリソースによって使用される方法は、大域ゾーンで実行され、その結果ファイルシステムをゾーンクラスタの仮想ノードにループバックマウントできます。共有 QFS を使用するクラスタファイルシステムとは異なり、UFS または VxFS を使用するクラスタファイルシステムは、複数のゾーンクラスタの仮想ノード上に同時マウントできます。また、ファイルシステムは、仮想ノードごとという原則に従って、読み取り/書き込みまたは読み取り専用でマウントできます。

ただし、上記の2つの方法では、多くの一般機能を共有しません。ゾーンクラスタでは、すべての仮想ノードに対し、クラスタファイルシステムを同じマウントポイントにマウントします。マウントしたクラスタファイルシステムは、ゾーンクラスタのすべての仮想ノードにマウントされます。1つの仮想ノードで障害が発生しても、ほかの仮想ノードからのクラスタファイルシステムへのアクセスは停止されません。ゾーンクラスタでは、参加する仮想ノードに、ゾーンクラスタのほかの仮想ノードでの現在の状況とまったく同じマウントポイントにマウントされた同じクラスタファイルシステムが自動的に割り当てられます。ゾーンクラスタは、クラスタファイルシステムに関する情報を、Cluster Configuration Repository に記録します。また、ゾーンクラスタは、管理者がクラスタファイルシステムへのアプリケーションの依存性を指定したとき、および実際にファイルシステムをマウントするときにアクセス権を検証します。

NFS ファイルシステム

NFS ファイルシステムは、2つの方法のいずれかを使用して、ゾーンクラスタ内にマウントできます。1番目の方法は、必要に応じて、NFS ファイルシステムを個々のノード（ここでは仮想ノード）にマウントする基本的アプローチです。2番目の方法では、スケラブルマウントポイント (SUNW.ScalMountPoint) リソースの管理下にあるゾーンクラスタ内のすべての仮想ノードに NFS ファイルシステムをマウントします。また、2番目の方法では、ゾーンクラスタに参加する仮想ノードへの NFS 自動マウントに関するクラスタファイルシステムのセクションで説明した機能を共有します。この方法は、大域ゾーン内でも使用できます。

NFS ファイルシステムは、大域ゾーンなどの1つのゾーンにインポートしてから、ループバックマウントを使用して、別のゾーンにエクスポートすることはできません。

ストレージデバイス

ゾーンクラスタは、ディスク、RAID ユニット、ネットワーク接続ストレージ (NAS) デバイスなどのストレージデバイスへの直接アクセスをサポートします。

ディスクと RAID デバイス

ゾーンクラスタは、通常のディスクと RAID ユニットの両方への直接アクセスをサポートします。また、ゾーンクラスタはボリュームマネージャーデバイスへのアクセスを、ディスクと同様にサポートします。ゾーンクラスタでは、1 つのゾーンクラスタのみから特定のディスクまたは RAID ユニットに直接アクセスできます。

ファイルシステムを介してディスクまたは RAID ユニット上のデータがアクセスされる場合は、管理者がディスクまたは RAID ユニットへの直接アクセス権を付与しないようにすることをお勧めします。アプリケーションにディスクまたは RAID ユニットへの直接アクセス権があると、そのアプリケーションから IOCTL 呼び出しが行われる可能性があります。悪意のあるアプリケーションで不正な IOCTL 呼び出しが行われると、ドライバが誤動作し、オペレーティングシステム障害が発生する場合があります。この場合は、信頼性に関する懸念が生じますが、セキュリティリスクは生じません。ファイルシステムを使用すると、このような信頼性の懸念は解消されます。ただし、ユーザーによっては、直接アクセスが必要でリスクについて理解しているので、ゾーンクラスタはこの機能をサポートしています。

ゾーンクラスタはディスクと RAID デバイスのフェンシングをサポートします。フェンシング機能を使用すると、クラスタを離脱したノードが共有ストレージのデータを変更できなくなります。これは、これまでの長期に渡って Oracle Solaris クラスタによってサポートされてきた非常に重要なデータ整合性機能です。ゾーンクラスタ機能は、ゾーンクラスタのノードのフェンシングをサポートするので、データ整合性が維持されます。

大域ゾーンが動作し続けているマシン上で仮想ノード障害が発生しても、オペレーティングシステムは、すべての入力が終了するまで、ゾーンにダウンマークを付けません。また、ゾーンクラスタは、オペレーティングシステムによってゾーンにダウンマークが付けられるまで、仮想ノードにダウンマークを付けません。したがって、仮想ノードがダウンした後は、離脱した仮想ノードからそれ以後入出力が行われないことを Oracle Solaris Cluster で保証できます。Oracle Solaris Cluster は、どのような場合も、離脱したノードから入出力が行われないようにします。

Oracle Solaris Cluster は、次のいずれの場合もディスクと RAID ユニットのフェンシングをサポートします。

- デバイスへの直接アクセスが行われる場合
- ボリュームマネージャーを介してアクセスが行われる場合
- ファイルシステムを介してアクセスが行われる場合

ゾーンクラスタはボリュームマネージャーデバイスへのアクセスもサポートします。ゾーンクラスタは、ボリュームマネージャーデバイスの基本的なゾーンサポートに依存します。Oracle Solaris Volume Manager (SVM) 管理も Veritas Volume Manager 管理も、大域ゾーンから行う必要があります。ゾーンクラスタは、大域ゾーンからのゾーンクラスタメンバーシップ変更後の自動ボリュームマネージャー再構成をサポートします。

注:-このホワイトペーパーが書かれて時点では、Sun Cluster デバイス用の Oracle Solaris Volume Manager は、ゾーンクラスタで動作しますが、ほかの種類の Oracle Solaris Volume Manager デバイスと Veritas ボリュームマネージャードバイスは動作しません。ただし、大域ゾーン内のボリューム上にマウントされたファイルシステムは、ボリュームで無効な場合もゾーンクラスタで動作するように構成できます。

NAS デバイス

一部の NAS デバイスは、iSCSI プロトコルを介してストレージをエクスポートでき、NAS デバイスはディスクに類似したストレージデバイスのように動作します。その場合に、クラスタは、ストレージデバイスのみを認識し、フェンシングサポートを含めて、ディスクとまったく同様に iSCSI LUN を管理します。

NAS ユニットには NFS を介してアクセスできます。その場合に、管理者はゾーンクラスタノード内で NFS クライアントマウントを実行し、データアクセスは NFS プロトコルに従います。

ネットワーク

ゾーンクラスタは、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両方を使用して、ネットワーク通信をサポートします。パブリックネットワークはクラスタ外の通信を表し、プライベートネットワークはクラスタノード間の通信を表します。

プライベートインターコネク

アプリケーションの観点から見て、ゾーンクラスタ内のプライベートインターコネクサポートは、大域ゾーン内のプライベートインターコネクサポートと同じです。システムは、`clprivnet` ドライバを使用して、使用可能なすべてのパス間でトラフィックをストライプ化し、少なくとも 1 つのパスが機能し続けている限り重複なしでデータ配信を保証します。プライベートインターコネクは最大 6 つまで存在できます。クラスタは、残されているものがなくなるまで、任意の数のプライベートインターコネク障害から透過的に回復します。

ゾーンクラスタは、プライベートインターコネク設定を自動化します。ゾーンクラスタは、グローバルクラスタが構成されたときに確立されたプライベートネットワークネットマスクと IP アドレスのプールからゾーンクラスタの仮想ノード用のネットマスクと IP アドレスを自動的に選択します。ゾーンクラスタは、グローバルクラスタのプライベートインターコネク構成に基づいて、適切な NIC を自動的に検出します。

ゾーンクラスタでは、グローバルクラスタによって使用されるものと同じ命名規則を使用します。それぞれのゾーンクラスタには、クラスタプライベートネットワーク名用の独立した名前空間があります。ルックアップが行われると、要求者がゾーンクラスタに基づいているときは要求者のゾーンクラスタ用の名前空間が使用され、それ以外の場合は、大域ゾーン用の名前空間が使用されます。ゾーンクラスタでは、グローバルクラスタのものと同じ物理プライベートインターコネクを使用しますが、ゾーンクラスタのトラフィックを分離するために一意のネットマスクと IP アドレスを使用するので、セキュリティ分離が有効になります。

ゾーンクラスタでは、クラスタでプライベートインターコネク用の Infiniband (IB) ネットワークを使用するときに、Reliable Datagram Sockets バージョン (RDS v1) も利用します。

パブリックネットワーク

ゾーンクラスタは、個々のゾーンと同じように、NIC を介して IP アドレスを使用することでパブリックネットワークと通信します。大域ゾーン内のシステム管理者は、`clzonecluster` コマンドを使用して、IP アドレスと NIC のその組み合わせに応じた特権をそのゾーンクラスタに付与します。システム管理者は、IP ネットワークマルチパス (IPMP) グループの NIC を 1 つのみ指定し、システムはその IPMP グループ内のいずれかの NIC を使用して、その IP アドレスを利用する権限を付与します。これらのネットワーク権限は、セキュリティの理由により、ゾーンクラスタ内からは変更できません。

LogicalHost リソースは、1 回に 1 つの仮想ノードのみからアクティブになる IP アドレスです。管理者は、1 つの仮想ノードから別の仮想ノードに LogicalHost リソースをスイッチオーバーできます。システムは、アプリケーションによって LogicalHost リソースが使用されるノード上で自動的にその LogicalHost リソースを構成します。仮想ノードで障害が発生すると、システムは LogicalHost リソースを、依存するアプリケーションとともに生存中のノードに自動的に移動します。ゾーンクラスタは、LogicalHost リソースの作成時、およびノード上の LogicalHost リソースのアクティブ化時に、LogicalHost のアクセス権を検証します。LogicalHost リソースは、常に 1 つのゾーンクラスタ内に留まります。

SharedAddress リソースは、クラスタ全体でアクティブになるように見える IP アドレスですが、実際の IP アドレスは、1 つのマシン上でホストされ、選択された負荷分散ポリシーに基づいて、着信トラフィックがクラスタ全体に分散されます。システムは、ノード障害が発生すると、SharedAddress に対応する IP アドレスをホストするノードを自動的に変更します。ゾーンクラスタは、SharedAddress リソースの作成時、およびノード上の SharedAddress リソースのアクティブ化時に、SharedAddress のアクセス権を検証します。SharedAddress リソースは、厳密に 1 つのゾーンクラスタ内のみで動作します。

一部のクラスタアプリケーションでは、`ifconfig` コマンドを実行して、そのクラスタアプリケーションによって使用される IP アドレスを管理します。ゾーンクラスタは次の `ifconfig` コマンドをサポートします。

- `ifconfig -a`

ゾーンクラスタに属する論理インタフェース、ゾーンクラスタ用に許可された IP アドレスをサポートするように明示的に構成された物理 NIC、許可された IP アドレスをサポートするように 1 つ以上の NIC が明示的に構成されている IPMP グループに属する NIC、`clprivnet` インタフェースをリストします。

- `ifconfig addif`

IP アドレスがゾーンクラスタ用に許可されて、この IP アドレスをサポートするように NIC が明示的に構成されるか、この IP アドレス をサポートするように NIC が明示的に構成されている IPMP グループに NIC が属する論理インタフェースを追加します。

- `ifconfig [up | down | plumb | removeif]`

論理インタフェース上で指定されたアクションを実行します。この論理インタフェースは、すでにゾーンクラスタに属している必要があります。

`ifconfig` コマンドには、非常に多くのサブコマンドとオプションがあります。ゾーンクラスタではほかのサブコマンドを使用できません。また、ほとんどのオプションを使用できません。目的は、クラスタアプリケーションに必要なコマンドを有効にするとともに、

ゾーン環境内のユーザーがほかのゾーンに影響を与えられるようにするコマンドを無効にすることです。

一般に、それぞれのゾーンクラスタでは、パブリックネットワーク通信用の IPMP グループを形成するために 2 つの NIC へのアクセスが必要で、プライベートインターコネクト用には 2 つの NIC へのアクセスが必要になります。それぞれのゾーンクラスタで、これらの接続専用の物理 NIC が必要になった場合は、必要な物理 NIC の数が、配備される新しいゾーンクラスタに応じたペースで増大します。ただし、ゾーンクラスタではネットワーク構成 `ip-type=shared` を使用するので、物理 NIC を、ゾーンクラスタ間で共有できます。これは、異なるゾーンクラスタで相互のトラフィックが認識されない安全な方法で行われます。

ローカルゾーンからネットワークアクセスを行うためのもう 1 つのオプションは、`ip-type=exclusive` です。このアプローチでは、接続ごとに専用の NIC が必要になります。ゾーンクラスタは `ip-type=exclusive` をサポートしません。

IP アドレスが不足している場合は、パブリック IP アドレスを使用しないようにゾーンクラスタノードを構成できます。そのように構成すると、ゾーンクラスタノードの使用に一定の制限が加わります。第一に、パブリック IP アドレスがない場合は、ゾーンが構成されている大域ゾーンから `zlogin` コマンドを使用することが、ゾーンにアクセスする唯一の方法となります。第二に、この方法で構成されたゾーンクラスタは、スケラブルサービスをホストできません。これは、そのようなサービスには、発信パケットを送信するためのパブリックネットワークが必要になるためです。

管理の概要

ゾーンクラスタに関連する管理役割には、システム管理者、アプリケーション管理者という 2 つの明確な種類があります。システム管理者は、ゾーンクラスタプラットフォームを管理します。アプリケーション管理者は、ゾーンクラスタ内のアプリケーションとそのリソースを管理します。データベース管理者は、一般的なアプリケーション管理者役割と言えます。次のサブセクションでは、これらの管理アクティビティについて説明します。

ゾーンクラスタ管理

ゾーンクラスタプラットフォームは、大域ゾーンのみから管理できます。ゾーンクラスタはゾーンクラスタ内で作成できません。これは、ゾーンクラスタがまだ存在しないためです。また、ゾーンクラスタ管理のほとんどが、そのゾーンクラスタ内の特定のリソースの使用を許可する作業で構成されることも、大域ゾーンのみからゾーンクラスタプラットフォームの管理を行うもう 1 つの理由となっています。セキュリティー分離は、最大の関心事の 1 つです。したがって、セキュリティーに関連する変更は、ゾーンクラスタ内からは行えません。

`clzonecluster` コマンドは、単一のコマンドでクラスタ全体に影響を与えるために、どのクラスタノードからでも実行できます。つまり、`clzonecluster` コマンドは、シングルポイント管理をサポートするので、大きいクラスタに対してコマンドを繰り返し実行する必要がなくなります。`clzonecluster` コマンドは、Oracle Solaris `zonecfg` コマンドと `zoneadm` コマンドの機能の組み合わせですが、それぞれのコマンドの形式に厳密に従います。`clzonecluster` コマンドは、ファイルシステム、デバイス、ネットワークのグローバルコンテキストなどのゾーンクラスタ固有の機能のサポートを追加します。

clzonecluster コマンドは、特定の仮想ノードに対してローカルなリソースのノード範囲も導入します。clzonecluster コマンドは、プライベートインターコネクットの構成を自動化し、グローバルクラスタの知識を使用して、ほとんどの Oracle Solaris sysidcfg 設定に適したデフォルトを自動的に割り当てるので、管理作業が軽減されます。ゾーンクラスタソフトウェアには、zonecfg または zoneadm コマンドによって間違った構成が行なわれないように、Oracle Solaris とのインタフェースが用意されています。

コマンドラインインタフェースに加えて、Oracle Solaris Cluster Manager for Oracle Solaris ソフトウェアは、ゾーンクラスタを管理するためのグラフィカルユーザーインタフェースを提供します。Oracle Solaris Cluster Manager の場合は Web ベースなので、ブラウザベースのユーザーインタフェースと呼ばれることもあります。

アプリケーション管理

ゾーンクラスタ内で実行されるアプリケーションは、その特定のゾーンクラスタ内または大域ゾーンから管理できます。

Oracle Solaris Cluster には、Resource Group Manager (RGM) サブシステムの制御下にあるアプリケーションを管理するための構成のしっかりしたフレームワークがあります。アプリケーション、ファイルシステム、ストレージデバイス、ネットワークリソースは、すべて RGM リソースとして識別されます。管理者は機能的な関係を依存性として識別します。たとえば、管理者は、アプリケーション FOO がファイルシステム /BAR に依存するように指定できます。つまり、この場合アプリケーション FOO を起動するには、ファイルシステム /BAR を事前に正しくマウントする必要があります。管理者は場所の関係を、アフィニティーとして指定できます。依存性関係とアフィニティー関係の両方に複数の形式があります。管理者は、データサービスのアプリケーションとその関連のリソースを、リソースグループに配置します。依存性は、同じリソースグループまたは異なるリソースグループ内のリソース間に確立できます。また、アフィニティーはリソースグループ間に確立できます。管理者は、可用性の問題に関連するポリシーをさらに選択できます。大域ゾーン内で使用できるすべての Oracle Solaris アプリケーション管理機能を、ゾーンクラスタ内で使用できます。

ゾーンクラスタはセキュリティーのゾーンモデルに従います。つまり、ゾーン内のユーザーは、そのゾーン内のみで何かを表示したり影響を与えたりできます。それぞれのゾーンクラスタには、アプリケーション管理に使用する固有の名前空間があるので、アプリケーション管理用の分離環境が実現されます。したがって、異なるゾーンクラスタ内の管理者間で名前が重複することはありません。多くの場合、管理者は新しい構成を作成するときに、作業用の構成をテンプレートとして使用します。異なる名前空間により、作業用の構成を比較的簡単にコピーできます。

複数のデータサービスを同じゾーンクラスタ内に配置できます。したがって、任意の数のアプリケーションを、単一のゾーンクラスタ内に配置できます。ただし、それぞれの異なるデータサービスをその専用のゾーンクラスタに配置することをお勧めします。その 1 つの理由は、ゾーンクラスタのゾーンレベルリソース制御を使用して、ライセンス料金を制御できることです。

Oracle Solaris Cluster に用意されている一連のアプリケーション管理用コマンドは、大域ゾーンの場合もゾーンクラスタの場合も同じです。コマンドを 1 つのゾーンクラスタ内で実行すると、そのコマンドは、そのクラスタ内のものだけを管理できます。同じコマンドが大域ゾーン内で実行される場合、管理者は、コマンドをゾーンクラスタ内で動作させるか、または大域ゾーン内で動作させるかを選択できます。

Oracle Solaris Cluster Manager ソフトウェアは、ゾーンクラスタ内のアプリケーションを管理するための GUI インタフェースを提供します。Oracle Solaris Cluster Manager は、大域ゾーンからのみ実行します。

Oracle Solaris Cluster 製品には、クラスタアプリケーションをインストールする際に必要な管理作業を著しく軽減する、データサービス構成ウィザードが用意されています。初期リリースには、さまざまな Oracle® Real Application Clusters (RAC) データベース構成をサポートするデータサービス構成ウィザードが含まれています。管理者は、Oracle Solaris Cluster Manager を使用して、ウィザードにアクセスします。アプリケーションは、構成ウィザードを使用しないで構成することもできます。

ゾーンクラスタ構成の例

このセクションでは、一般的なゾーンクラスタを作成する代表例を紹介します。30 ページの「ゾーンクラスタ管理」では、一連のゾーンクラスタ管理タスクのすべてについて詳しく説明しています。

予備的な構成

ゾーンクラスタを作成する前に、システム管理者は次のような多くの前提条件を満たす必要があります。

- Oracle Solaris オペレーティングシステムと Oracle Solaris Cluster ソフトウェアをクラスタ内のすべてのマシンにインストールする。
- グローバルクラスタを構成して、グローバルクラスタをクラスタモードで起動する。
- ゾーンクラスタによって使用されるストレージボリュームを作成する。
- ゾーンクラスタによって使用されるファイルシステムを作成する。
- ゾーンクラスタによって使用される NIC 向けの IPMP グループを作成する。
- 新しく作成したゾーンクラスタで使用されるルートパスワードの暗号化値を特定する (詳細については、`sysidcfg(4)` マニュアルページを参照)。

これらの操作の実行手順については、Oracle Solaris and Oracle Solaris Cluster 3.3 のマニュアルを参照してください。

ゾーンクラスタ構成

この例で使用する `clzonecluster` コマンドは、任意のノードから実行でき、クラスタ全体に対して動作します。通常の場合、Oracle Solaris Cluster コマンドには、長形式と省略形式があります。`clzonecluster` コマンドの省略形式は `clzc` です。どちらのコマンドも同じパラメータをとり、同じタスクを実行します。

このセクションでは、図 2 のような 2 つの仮想ノードで構成された代表的なゾーンクラスタの構成例について説明を加えながら検討します。このゾーンクラスタの目的は、Oracle Solaris Volume Manager for Sun Cluster 上で、Sun QFS 共有ファイルシステムを使用する Oracle RAC データベースをサポートすることと、フェイルオーバーファイルシステム上で実行されるアプリケーションをサポートすることです。

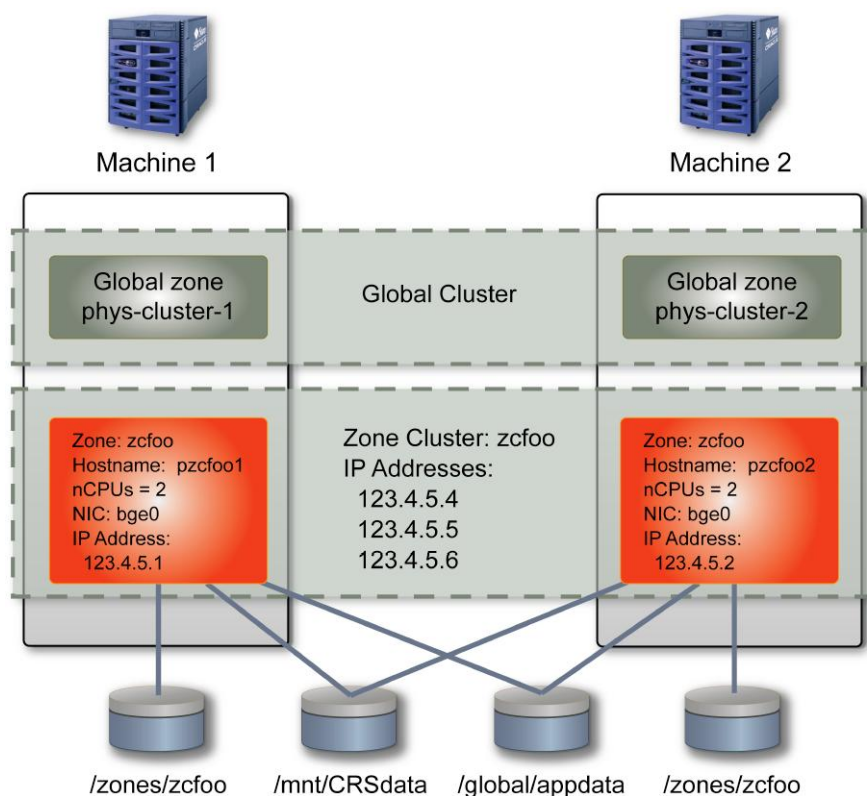


図 10. ゾーンクラスタ構成の例

1. 最初の手順は、マシンを起動すると必ず起動される `zcfoo` という名前のゾーンクラスタを作成することです。

```
# clzonecluster configure zcfoo
clzc:zcfoo> create
clzc:zcfoo> set zonepath=/zones/zcfoo
clzc:zcfoo> set autoboot=true
```

`create` コマンドは、疎ルートゾーンで構成されるゾーンクラスタを作成します。完全ルートゾーンクラスタを構成するには、`-b` オプションを `create` コマンドに渡します。既存のゾーンクラスタをテンプレートとして使用して、ゾーンクラスタを構成することもできます。たとえば、`zcfoo` をテンプレートとして使用してゾーンクラスタ `zcbar` を構成するには、次のコマンドを使用できます。

```
# clzonecluster configure zcbar
clzc:zcbar> create -t zcfoo
```

このコマンドを実行すると、プロパティが `zcfoo` と共通する `zcbar` ゾーンクラスタが構成されます。プロパティはあとから必要に応じて、`zcbar` 用にカスタマイズできます。

2. 専用 CPU を追加します。固定された数の CPU (使用可能な総数よりも少ない) を使用すると、ライセンス費用を効果的に軽減できます。この例では、各マシン上でこのゾーンクラスタ専用 2 つの CPU のみを使用するように選択します。

```
clzc:zcfoo> add dedicated-cpu
clzc:zcfoo:dedicated-cpu> set ncpus=2
clzc:zcfoo:dedicated-cpu> end
```

3. 次に、ゾーンクラスタのノードを追加します。ノード用のパブリック IP アドレスおよびネットワークインタフェースとともに、物理ホストマシンと仮想ノード名を指定する必要があります。次のコマンドは、physcluster-1 マシン上で動作する pzcfoo1 という名前の仮想ノードを最初に作成します。次に、phys-cluster-2 マシン上に、2 番目の仮想ノード pzcfoo2 を作成します。

```
clzc:zcfoo> add node
clzc:zcfoo:node> set physical-host=phys-cluster-1
clzc:zcfoo:node> set hostname=pzcfoo1
clzc:zcfoo:node> add net
clzc:zcfoo:node:net> set address=123.4.5.1
clzc:zcfoo:node:net> set physical=bge0
clzc:zcfoo:node:net> end
clzc:zcfoo:node> end

clzc:zcfoo> add node
clzc:zcfoo:node> set physical-host=phys-cluster-2
clzc:zcfoo:node> set hostname=pzcfoo2
clzc:zcfoo:node> add net
clzc:zcfoo:node:net> set address=123.4.5.2
clzc:zcfoo:node:net> set physical=bge0
clzc:zcfoo:node:net> end
clzc:zcfoo:node> end
```

4. IP アドレスを指定します。この例で作成されるゾーンクラスタには、Oracle RAC で Virtual IP (VIP) アドレスとして使用される 2 つの IP アドレスと、フェイルオーバーアプリケーションで使用する 1 つの IP アドレスが必要です。これらの IP アドレスは、ゾーンクラスタ内の任意のノードで使用できるため、グローバルコンテキストで指定されます。

```
clzc:zcfoo> add net
clzc:zcfoo:net> set address=123.4.5.4
clzc:zcfoo:net> end

clzc:zcfoo> add net
clzc:zcfoo:net> set address=123.4.5.5
clzc:zcfoo:net> end

clzc:zcfoo> add net
clzc:zcfoo:net> set address=123.4.5.6
clzc:zcfoo:net> end
```

システムは、指定されたサブネットを使用して、IP アドレスに適合する NIC を自動的に判定します。システムは、すでにこのゾーンクラスタに構成されたこのサブネット上の IP アドレスとともに NIC を検索します。システムは、この NIC をこのゾーンクラスタに使用できることを認識し、この NIC を使用して、指定された IP アドレスをホストします。

システムは、clprivnet インタフェースを使用して、クラスタプライベートインタコネクトを有効にするための構成を自動的に実行します。これには、プライベートネットワーク上のサブネットと一連の IP アドレスの割り当てが含まれます。そのサポートのためにほかの操作を行う必要はありません。

5. Sun QFS 共有ファイルシステムを指定します。この例で作成されるゾーンクラスタには、Oracle RAC をサポートするための Sun QFS 共有ファイルシステムが必要です。管理者は、クラスタファイルシステムをグローバルコンテキストで指定します。これは、これらのファイルシステムが複数のノード上で動作するためです。special パラメータを使用して、マスター設定ファイル (MCF) に表示されるとおりに Sun QFS 共有ファイルシステムを指定します。Sun QFS 共有ファイルシステムには、raw パラメータを使用しないでください。

```
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/mnt/CRSdata
clzc:zcfoo:fs> set special=CrsData
clzc:zcfoo:fs> set type=samfs
clzc:zcfoo:fs> end
```

6. UFS 高可用性ローカルファイルシステムを指定します。この例で作成されるゾーンクラスタには、フェイルオーバーアプリケーション用の UFS 高可用性ローカルファイルシステムが必要です。管理者は、高可用性ローカルファイルシステムをグローバルコンテキストで指定します。これは、1 回に 1 つのノード上のみでアクティブになる条件下でも、これらのファイルシステムが複数のノード上で動作するためです。

```
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/global/appdata
clzc:zcfoo:fs> set special=/dev/md/app/dsk/d20
clzc:zcfoo:fs> set raw=/dev/md/app/rdisk/d20
clzc:zcfoo:fs> set type=ufs
clzc:zcfoo:fs> add options [logging,nodevices]
clzc:zcfoo:fs> end
```

7. sysid 情報を追加します。Oracle Solaris ゾーンには、sysidcfg の情報が必要です。唯一の必須フィールドは root_password です。これは、/etc/shadow ファイルに格納される形式の暗号化値です。これらのパラメータは、ゾーンクラスタのどのゾーンコンポーネントでも同じです。

```
clzc:zcfoo> add sysid
clzc:zcfoo:sysid> set root_password=<encrypted passwd from /etc/shadow>
clzc:zcfoo:sysid> end
```

8. ここで、ゾーンクラスタ構成を確認してコミットします。

```
clzc:zcfoo> verify
clzc:zcfoo> commit
clzc:zcfoo> exit
```

9. ゾーンをインストールします。ゾーンをインストールするための個別の手順があります。

```
# clzonecluster install zcfoo
```

インストールには時間がかかる場合があります。インストールが完了するとゾーンクラスタがインストールされますが、起動はされません。

10. ゾーンクラスタを起動します。

```
# clzonecluster boot zcfoo
```

boot コマンドを実行すると、インポートされた SMF マニフェストなどのコンポーネントゾーンが初期化されます。システム識別が完了すると、それぞれのゾーンが再起動されます。この時点で、ゾーンクラスタの使用準備が整います。

ゾーンクラスタ管理

clzonecluster コマンドは、作成から変更、制御、最終的な破棄に至るまでのあらゆるゾーンクラスタ管理アクティビティをサポートします。clzonecluster コマンドは、シングルポイント管理をサポートします。つまり、コマンドはどのノードからでも実行でき、クラスタ全体に対して動作します。clzonecluster コマンドは Oracle Solaris zonecfg コマンドおよび zoneadm コマンドの機能を組み合わせたもので、クラスタ機能のサポートを追加します。

ゾーンクラスタは広範な機能をサポートします。このセクションでは、これらの各機能の構成プロセスについて説明します。このセクションは、構成される主要機能に従って編成されています。ゾーンクラスタはネイティブゾーンによって提供される機能セットに基づきます。このセクションでは、ゾーンクラスタ用に追加された機能またはネイティブゾーンのブランドタイプサポートとはやや異なる機能について説明します。

注: ネイティブゾーンサポートについては、Oracle Solaris OS のマニュアルを参照してください。

ノードとクラスタ全体スコープ

ネイティブゾーンには、構成時に使用される最上位スコープ、リソーススコープという 2 レベルのスコープがあります。最上位スコープを使用するときは、zonecfg コマンドでゾーンのプロパティを指定します。一方、zonecfg コマンドには、デバイス、ファイルシステム、ネットワークリソースなどの項目のリソーススコープもあります。リソーススコープは、特定のリソースに関する情報を指定します。

clzonecluster コマンドは、スコープの可能性を拡張します。最上位スコープはクラスタ全体に適用する情報を指定します。たとえば、zonepath は、ゾーンクラスタを構成するすべてのゾーンで同じにする必要があります。clzonecluster コマンドは、ノードスコープを追加して、ゾーンクラスタ内の特定のノードのみに適用する情報を指定できるようにします。ノードスコープは、zonecfg コマンドの最上位スコープに相当します。

次の例は、既存のノードのノードスコープ内に入りたりノードスコープ外に出たりする方法を示しています。add node サブコマンドは、作成されるノードのノードスコープに自動的に入ります。end サブコマンドでは、ノードスコープから出て、最上位スコープに戻ります。この後の例では、ノードスコープ内で実行できるタスクを示します。

```
clzc:zcfoo> add node
clzc:zcfoo:node> set physical-host=phys-cluster-3
clzc:zcfoo:node> end
clzc:zcfoo>
```

clzonecluster コマンドは、クラスタ全体をベースにしたリソースと、特定のノードに対してローカルなリソースの両方の管理をサポートします。管理者が最上位またはクラスタスコープからリソーススコープに入ると、リソースがクラスタ全体に適用されます。管理者がノードスコープからリソーススコープに入ると、リソースがその特定のノードに適用されます。たとえば、管理者は、最上位またはクラスタスコープからクラスタファイルシステムリソースを指定します。または、ノードスコープからローカルファイルシステムを指定します。

システム識別

Oracle Solaris ゾーンは、管理者が Oracle Solaris `sysidcfg` (4) 機能を使用して、一連のシステム識別パラメータを指定するまでは動作できません。この情報は、ゾーンクラスタを最初に作成するときに指定します。これらのプロパティは、ゾーンクラスタのどのゾーン上でも同じです。

`clzonecluster` コマンドには、ゾーンクラスタのすべてのゾーンに対してこの情報を指定するための `sysid` リソーススコープがあります。多くの場合は、適切なデフォルト値がシステムによって提供されるので、自動的に判定できる基本情報を管理者が入力する必要はありません。たとえば、ゾーンクラスタのデフォルトタイムゾーンは、物理クラスタの場合と同じです。

この時点で管理者が入力する必要があるのは、暗号化されたルートパスワードのみです。必要に応じて、システムロケールやタイムゾーンなどのほかの情報も入力できます。

次に示すのは、ゾーンクラスタのすべてのゾーンに対して、多くのフィールドの `sysidcfg` (4) 情報を指定する例です。この例では、管理者がすでにゾーンクラスタの構成プロセスを開始していることを前提にします。

```
clzc:zcfoo> add sysid
clzc:zcfoo:sysid> set root_password=<encrypted passwd from /etc/shadow>
clzc:zcfoo:sysid> set name_service="NIS {domain_name=scdev.example.com
  name_server=timber(1.2.3.4)}"
clzc:zcfoo:sysid> set nfs4_domain=dynamic
clzc:zcfoo:sysid> set security_policy=NONE
clzc:zcfoo:sysid> set system_locale=C
clzc:zcfoo:sysid> set terminal=xterms
clzc:zcfoo:sysid> set timezone=US/Pacific
clzc:zcfoo:sysid> end
clzc:zcfoo>
```

これらのパラメータについては、`sysidcfg` (4) マニュアルページを参照してください。

ノードサポート

ゾーンクラスタでは、ゾーンクラスタからノードを追加したり削除したりできます。

ノードの追加

ゾーンクラスタを最初に作成するときに、ある一定のノードセットを指定する必要があります。管理者は初期構成後にノードをゾーンクラスタに追加することもできます。管理者が既存のゾーンクラスタにノードを追加すると、追加したノードに、ゾーンクラスタのすべてのグローバルプロパティが自動的に適用されます。これには、クラスタファイルシステムなどのグローバルリソースに関する情報が含まれます。

管理者は、ゾーンクラスタノードと同じマシン上に存在する (1) グローバルクラスタノードのホスト名と、(2) ゾーンクラスタノードのホスト名を指定する必要があります。ホスト名には、`telnet` を使用する場合などに、ネットワークを介してゾーンクラスタノードに到達しようとするユーザーが指定できる特定の IP アドレスが含まれます。ほとんどの状況で、管理者はネットワークからのゾーンへのアクセスをサポートするネットワーク情報も指定する必要があります。これは、ネットワークからゾーンへのログインを有効にするために必要です。エントリを RGM リソースグループノードリストに追加して、RGM リソースグループに有効な場所を指定するときに、ゾーンホスト名が使用されます。

次の例では、既存のゾーンクラスター `zcfoo` にゾーンを追加します。ゾーンはグローバルクラスターノード `phys-cluster-3` に追加され、仮想ノードにホスト名 `zc-node-3` が割り当てられます。

```
# clzonecluster configure zcfoo
clzc:zcfoo> add node
clzc:zcfoo:node> set physical-host=phys-cluster-3
clzc:zcfoo:node> set hostname=zc-node-3
clzc:zcfoo:node> add net
clzc:zcfoo:node:net> set physical=hme0
clzc:zcfoo:node:net> set address=123.4.5.5
clzc:zcfoo:node:net> end
clzc:zcfoo:node> end
clzc:zcfoo> exit
```

ノードの削除

管理者は、ゾーンクラスターの構成プロセスの進行中にノードを削除できます。次のコマンドは、指定された物理ホスト上のノードを削除します。

```
clzc:zcfoo> remove node physical-host=phys-cluster-2
```

ゾーンクラスターが構成済みで動作可能な場合、管理者は最初に `clzonecluster (1CL)` コマンドを使用してそのゾーンクラスターノードを停止してから、ノードをアンインストールする必要があります。そのあとで、`remove` サブコマンドを使用して、仮想ノードを削除できます。

次のコマンドは、すでに構成されて動作可能になっている仮想ノードを削除する例を示しています。

```
# clzonecluster halt -n phys-cluster-2 zcfoo
# clzonecluster uninstall -n phys-cluster-2 zcfoo
# clzonecluster configure zcfoo
clzc:zcfoo> remove node physical-host=phys-cluster-2
clzc:zcfoo> exit
#
```

ファイルシステムサポート

ゾーンクラスターは、ローカルファイルシステム、共有 Sun QFS ファイルシステム、高可用性ファイルシステムという 3 種類のファイルシステムをサポートします。次のセクションでは、種類ごとのサポートについて説明します。

ローカルファイルシステム

ローカルファイルシステムは、1 つのノードのみにマウントできます。ローカルファイルシステムは、ネイティブゾーンによってサポートされるファイルシステムの一つです。`clzonecluster` コマンドは、現在ローカルファイルシステムを構成する機能をサポートしていません。その代わりに、管理者は `zonecfg` コマンドを使用して、ローカルファイルシステムを構成できます。

Sun QFS 共有ファイルシステム

Sun QFS 共有ファイルシステムは、ゾーンクラスタのすべてのノード上から同時アクセスできます。管理者は、最上位スコープで Sun QFS 共有ファイルシステムを指定します。

次に示すのは、Sun QFS 共有ファイルシステムの構成例です。

```
# clzonecluster configure zcfoo
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/qfs/ora_home
clzc:zcfoo:fs> set special=oracle_home
clzc:zcfoo:fs> set type=samfs
clzc:zcfoo:fs> end
clzc:zcfoo> exit
#
```

dir エントリは、zonepath を基準にしたマウントポイントです。special エントリは、Sun QFS マスター設定ファイル (MCF) に表示されるとおりの Sun QFS ファイルシステムの名前です。raw エントリは、Sun QFS ファイルシステムの構成時は使用しません。options エントリは Sun QFS ファイルシステムの構成時は、clzonecluster コマンドと一緒に使用しません。その代わりに、オプションを MCF ファイルと vfstab ファイルで指定します。

UFS または Veritas VxFS クラスタファイルシステム

VxFS クラスタファイルシステムは、ゾーンクラスタのすべてのノード上から同時アクセスできます。管理者は、最上位スコープでクラスタファイルシステムを指定します。

次に示すのは、クラスタファイルシステムの構成例です。

```
# clzonecluster configure zcfoo
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/oradata/flash_recovery_area
clzc:zcfoo:fs> set special=/global/zcfoo/orafsl
clzc:zcfoo:fs> set type=lofs
clzc:zcfoo:fs> end
clzc:zcfoo> exit
#
```

dir エントリは、zonepath を基準にしたマウントポイントです。special エントリは、大域ゾーン内のクラスタファイルシステムのマウントポイントです。クラスタファイルシステムは、マウント操作を実行するためにそのメソッドが大域ゾーンで実行される、ゾーンクラスタ内の SUNW.HAStoragePlus リソースの制御下にあります。options エントリはクラスタファイルシステムの構成時は、clzonecluster コマンドと一緒に使用しません。その代わりに、オプションを vfstab ファイルで指定します。

高可用性ファイルシステム

フェイルオーバーファイルシステムとも呼ばれる高可用性ファイルシステムは、1 回に 1 つのノードのみにマウントされます。システムでは、ノード障害または管理者コマンドに応じて高可用性ファイルシステムをノード間で移動できます。管理者は高可用性ファイルシステムを最上位スコープで指定します。

次に示すのは、高可用性ファイルシステムとしての UFS ファイルシステムの構成例です。

```
# clzonecluster configure zcfoo
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/mnt/foo-app
clzc:zcfoo:fs> set special=/dev/md/foo-ds/dsk/d20
clzc:zcfoo:fs> set raw=/dev/md/foo-ds/rdisk/d20
clzc:zcfoo:fs> set type=ufs
clzc:zcfoo:fs> end
clzc:zcfoo> exit
#
```

ゾーンクラスタは高可用性ファイルシステムとしての Oracle Solaris ZFS もサポートします。ゾーンクラスタは、Oracle Solaris ZFS ストレージプールの粒度で Oracle Solaris ZFS をサポートします。Oracle Solaris Cluster では、個々のファイルシステムではなく、Oracle Solaris ZFS ストレージプール全体をノード間で移動します。

次の例では、Oracle Solaris ZFS ストレージプール zpoo11 を高可用性として構成しています。

```
# clzonecluster configure zcfoo
clzc:zcfoo> add dataset
clzc:zcfoo:dataset> set name=zpoo11
clzc:zcfoo:dataset> end
clzc:zcfoo> exit
#
```

ストレージデバイスサポート

ゾーンクラスタでは、ローカルデバイスやクラスタ全体デバイスなどのストレージデバイスを直接使用できます。

ローカルデバイス

ローカルストレージデバイスは、1 台のマシンのみに接続できるデバイスです。clzonecluster コマンドは、現在ローカルデバイスをサポートしていません。管理者は、代わりに zonecfg コマンドを使用して、ローカルデバイスを構成できます。

クラスタ全体デバイス

クラスタ全体ストレージデバイスは、2 つの方法のいずれかで複数のノードから使用できます。Oracle Solaris Volume Manager for Sun Cluster などのデバイスは、複数のノードから同時に使用できます。その他のデバイスも複数のノードから使用できますが、ある時点でデバイスにアクセスできるのは 1 つのノードのみです。複数のマシンに接続されている通常の SVM デバイスはその一例です。

管理者はクラスタ全体デバイスを最上位コンテキストで構成します。一連の Oracle Solaris Volume Manager for Sun Cluster デバイスを構成する次の例のように、デバイスを特定するときにワイルドカードを使用できます。

```
# clzonecluster configure zcfoo

clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/md/oraset/dsk/*
clzc:zcfoo:device> end

clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/md/oraset/rdsk/*
clzc:zcfoo:device> end
```

```

clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/md/1/dsk/*
clzc:zcfoo:device> end

clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/md/1/rdsk/*
clzc:zcfoo:device> end
clzc:zcfoo:> exit

```

Oracle Solaris Volume Manager for Sun Cluster メタセットまたはメタデバイス (またはその両方) をゾーンクラスタにエクスポートするときは、論理デバイスパスと物理デバイスパスの両方を指定する必要があることに注意してください。上の例の場合、`oraset` のセット番号は 1 です。メタセットのセット番号を探すには、次の例のように `ls -l` コマンドを実行して、セット名を指定します。

このコマンドの出力には、セット番号が含まれているシンボリックリンクが表示されます。

```
# ls -l /dev/md/oraset
```

DID デバイスも構成できます。次の例では、DID デバイス `d10` を構成しています。

```

# clzonecluster configure zcfoo
clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/did/*dsk/d10s*
clzc:zcfoo:device> end
clzc:zcfoo:> exit
#

```

ネットワークサポート

ゾーンクラスタは、次のセクションで説明するようにパブリックネットワーキングとプライベートネットワーキングの両方をサポートします。

プライベートインターコネクト

プライベートインターコネクトは、クラスタのノード間のネットワーク接続を表します。システムは、プライベートインターコネクトでの通信をサポートするように、自動的にゾーンクラスタを構成できます。物理クラスタのインストール時に指定されたプライベートネットワークサブネットのプールからサブネットを自動的に選択します。システムはその後に、ゾーンクラスタの各仮想ノードの IP アドレスを割り当てます。システムソフトウェアによって、異なるゾーンクラスタのプライベートネットワークが、独立した名前空間に分離されます。その結果、それぞれのゾーンクラスタに、専用の Oracle Solaris Cluster プライベートネットワークドライバ (`clprivnet`) サポートが効果的に設定されて、同じ物理ネットワークが共有されます。

ゾーンクラスタにプライベートインターコネクトサポートが必要ない場合は、管理者がこの機能を無効にするかオフにできます。1 つのフェイルオーバーアプリケーションのみをサポートするゾーンクラスタは、プライベートインターコネクトを必要としない構成の一例です。

次の例は、ゾーンクラスタを作成するときにこの機能をオフにする方法を示しています。プロパティは最上位コンテキストで設定する必要があります。

```
clzc:zcfoo> set enable_priv_net=false
```

注:-動作中のゾーンクラスタ上でこのプライベートインターコネクト機能を変更することはできません。

パブリックネットワーク

パブリックネットワークは、クラスタ外のネットワーク通信を表します。ゾーンクラスタには、ローカルネットワークとクラスタ全体ネットワークの両方のサポートが含まれます。

- ローカルネットワークサポート

ローカルネットワークリソースは、1つのノードによって排他的に使用されます。clzonecluster コマンドは、現在ローカルネットワークリソースをサポートしていません。その代わりに、管理者は zonecfg コマンドを使用して、ローカルネットワークリソースを構成できます。

- クラスタ全体ネットワークサポート

ネットワークリソースは、クラスタの複数のノード上で使用するよう構成できます。IP アドレスは、1回に1つのノード上でのみホストできます。ただし、この種のネットワークリソースは、仮想ノード間で随時に移動できます。

次のエンティティには、この種のネットワークリソースが必要です。

- 論理ホスト
- 共有アドレス
- Oracle RAC Virtual IP (VIP) アドレス
- クラスタアプリケーションによって直接管理される IP アドレス (plumb、unplumb、up、down、addif などのコマンドを使用)

次の例では、クラスタ全体で使用できるネットワークリソースを構成しています。

```
# clzonecluster configure zcfoo
clzc:zcfoo> add net
clzc:zcfoo:net> set address=123.4.5.5
clzc:zcfoo:net> end
clzc:zcfoo> exit
#
```

ネットワークインタフェースは、クラスタ全体ネットワークリソース用にユーザーが指定できないことに注意してください。

システムは指定されたネットワークリソースのサブネットを特定します。システムでは、(1) このゾーンクラスタで使用するをすでに許可されている NIC、または (2) このゾーンクラスタで使用するをすでに許可されている IP ネットワークマルチパス (IPMP) グループ内の NIC 上で、指定された IP アドレスを使用できます。通常の場合は、ログインなどの目的でそれぞれのゾーンで使用するよう構成されたネットワークリソースが存在します。ここでは、IP アドレスと NIC の両方をチェックする厳密なゾーンセキュリティポリシーに従います。

起動操作と停止操作

管理者は、物理クラスタとまったく同じように、ゾーンクラスタ全体を随時に手動で起動したり停止したりできます。この例のゾーンクラスタは、ノードが起動すると自動的に起動され、ノードが停止されると自動的に停止します。次のコマンドは、構成されたすべてのノード上で、ゾーンクラスタ全体を起動および停止します。

```
# clzonecluster boot zcfoo
# clzonecluster halt zcfoo
```

管理者は、ゾーンクラスタの個々のノードを起動したり停止したりできます。通常の場合、管理者は、ソフトウェアアップグレードなどの管理タスクを実行するために個々のノードを停止して再起動します。次の例では、指定されたノードを起動して停止します。

```
# clzonecluster boot -n <base-cluster-node> zcfoo
# clzonecluster halt -n <base-cluster-node> zcfoo
```

注:-ゾーンクラスタのゾーンコンポーネントは、ゾーンコンポーネントをホストしているマシンがクラスタモードで起動されたときのみ、クラスタモードで起動できます。

`cluster shutdown` コマンドを使用して、ゾーンクラスタを停止することもできます。大域ゾーン内で `cluster shutdown` を実行すると、すべてのゾーンクラスタおよび物理クラスタが停止します。ゾーンクラスタ内で `cluster shutdown` を実行すると、その特定のゾーンクラスタが停止します。これは、`clzonecluster` コマンドでゾーンクラスタ全体を停止することと同じです。

削除操作

ゾーンクラスタを削除するには、事前にすべてのリソースグループとその関連のリソースを削除する必要があります。次に、ゾーンクラスタを停止してアンインストールしてから削除を行います。管理者は、次のコマンドを実行してゾーンクラスタを破棄します。

```
# clzonecluster halt zcfoo
# clzonecluster uninstall zcfoo
# clzonecluster delete zcfoo
```

ゾーンクラスタ情報の表示

2 つのサブコマンド `status` および `list` を使用して、ゾーンクラスタに関する情報を取得します。`list` サブコマンドは、システム上で構成されているゾーンクラスタを一覧表示します。

`status` サブコマンドは、ノードごとのホスト名や状態などのゾーンクラスタに関する情報を表示します。次の例では、特定のゾーンクラスタの情報を表示します。

```
# clzonecluster status -v zcfoo
=== Zone Clusters ===

--- Zone Cluster Status ---

Name      Node Name      Zone HostName  Status      Zone Status
-----
zcfoo     phys-cluster-1  giggles-1     Online     Running
          phys-cluster-2  giggles-2     Online     Running
          phys-cluster-3  giggles-3     Online     Running
```


複製操作

clone サブコマンドは、Oracle Solaris zoneadm 複製コマンドと同じように、ゾーンクラスタを複製します。clone サブコマンドを実行する前に、管理者が最初にゾーンクラスタを構成する必要があります。clone サブコマンドでは、参照ゾーンクラスタを使用して、この特定のゾーンクラスタのインストール方法を判定します。通常の場合、システムでは clone サブコマンドを使用して、より迅速にゾーンクラスタをインストールできます。

その他のゾーンサブコマンド

Oracle Solaris zonecfg および zoneadm は、その他のサブコマンドをサポートします。clzonecluster コマンドは、これらの追加サブコマンドのほとんどをサポートします。実際、clzonecluster コマンドは次のサブコマンドをサポートしません。

- attach
- detach

clzonecluster コマンドは、Oracle Solaris zonecfg および zoneadm コマンドによってサポートされている他のサブコマンドをサポートします。ちがいは、clzonecluster コマンドがサブコマンドをゾーンクラスタのすべてのゾーンに適用することです。詳細については、zonecfg と zoneadm のマニュアルページを参照してください。

注:-ゾーンクラスタの初期リリースは、Oracle Solaris 10 5/08 OS に基づいています。追加サブコマンドのサポートについては、Oracle Solaris Cluster のマニュアルとリリーススケジュールを参照してください。

Oracle Solaris OS のコマンド対話処理

Oracle Solaris OS には、ゾーンを管理するためのコマンドが含まれています。必然的に、Oracle Solaris zonecfg および zoneadm コマンドは、グローバルファイルシステムなどの単一マシンゾーンに存在しない項目を管理できません。Oracle Solaris OS コマンドは、セキュリティ関連プロパティなどの、複数のノード上で同じになる必要があるリソースを管理しません。Oracle Solaris OS コマンドは、ゾーンクラスタのゾーンコンポーネントの一部のローカル機能を管理するために使用します。

たとえば、zonecfg コマンドを使用して、次の項目を構成できます。

- ローカルファイルシステム
- ローカル Oracle Solaris ZFS プール
- ローカルデバイス
- ローカルネットワークリソース
- 1 つのノード上のリソース制御プロパティ

ただし、zonecfg コマンドは、次の項目を構成できません。

- ゾーン名
- ゾーンパス
- limitpriv プロパティ
- Solaris リソースマネージャープール
- inherit-pkg-dir プロパティ
- クラスタファイルシステムなどのクラスタ全体リソース

zoneadm コマンドは、ローカルノード上のゾーンを起動したり停止したりできます。また、ローカルノード上のゾーンの状態をリストできます。

ゾーンクラスタ管理 GUI

Oracle Solaris Cluster は、テキストベースの対話型 clsetup コマンドインタフェースと、管理処理用 Oracle Solaris Cluster Manager for Oracle Solaris ブラウザベースグラフィカルユーザーインターフェース (GUI) の両方を提供します。Oracle Solaris Cluster Manager ブラウザベース GUI は、大域ゾーンの管理者がゾーンクラスタ内のリソースグループとリソースを表示して管理できるように機能強化されています。Oracle Solaris Cluster Manager は、ゾーンクラスタでは実行されません。

要約

ゾーンクラスタは、アプリケーションの制御と管理を行うためのセキュリティー保護された環境を提供します。クラスタアプリケーションは、この環境を専用プライベートクラスタとして認識します。

このレポートでは、この機能について幅広く説明していますが、より詳細な情報については、Oracle Solaris Cluster のマニュアルを参照してください。

著者について

Tim Read は、Oracle Solaris Cluster Group のソフトウェア開発者です。その主な役割は、Oracle Solaris Geographic Edition 製品の開発です。2010 年に出版された『Oracle Solaris Cluster Essentials』などをはじめとして、高可用性および障害回復についての多くのホワイトペーパーと著作があります。

Tim Read は、英国のバーミンガム大学で物理学および天体物理学分野の理学士号を取得しています。

このホワイトペーパーは、Dr. Ellard Roush によるオリジナルのソースマテリアルに基づいて更新されたものです。

謝辞

ゾーンクラスタ機能の開発は、プロジェクトチームによって達成されました。さまざまな面で開発に貢献した次のすべてのチームメンバーに感謝します。Zoram Thanga、Pramod Rao、Tirthankar Das、Sambit Nayak、Himanshu Ashiya、Varun Balegar、Prasanna Kunisetty、Gia-Khanh Nguyen、Robert Bart、Suraj Verma、Harish Mallya、Ritu Agrawal、Madhan Balasubramanian、Ganesh Ram Nagarajan、Bharathi Subramanian、Thorsten Frueauf、Charles Debardeleben、Venkateswarlu Tella、Hai-Yi Cheng、Lina Muryanto、Jagrithi Buddharaja、Nils Pedersen、Burt Clouse。

関連資料

表 20. 詳細情報の参照用資料

説明	URL
Oracle Solaris Cluster	http://www.oracle.com/us/products/servers-storage/solaris/cluster-067314.html
Sun Cluster Wiki	http://wikis.sun.com/display/SunCluster/Home
「ゾーンクラスタの構成」、Sun Cluster ソフトウェア Installation Guide for Solaris OS	http://download.oracle.com/docs/cd/E19680-01/821-1255/ggzen/index.html
Oracle Solaris Cluster 3.3 Documentation Center	http://download.oracle.com/docs/cd/E19680-01/821-1261/index.html
Oracle Solaris Cluster Concepts Guide for Solaris OS	http://download.oracle.com/docs/cd/E19680-01/821-1254/index.html
Oracle Solaris Cluster System Administration Guide for Solaris OS	http://download.oracle.com/docs/cd/E19680-01/821-1257/index.html
System Administration Guide: Solaris Containers— リソース管理と Solaris ゾーン	http://download.oracle.com/docs/cd/E19253-01/817-1592/index.html
clzonecluster(1CL) マニュアルページ	http://download.oracle.com/docs/cd/E19680-01/821-1263/6nm8r5igu/index.html
sysidcfg(4) マニュアルページ	http://download.oracle.com/docs/cd/E19253-01/816-5174/6mbb98uiq/index.html
zoneadm(1M) マニュアルページ	http://download.oracle.com/docs/cd/E19253-01/816-5166/6mbb1kgoa/index.html
zonectg(1M) マニュアルページ	http://download.oracle.com/docs/cd/E19253-01/816-5166/6mbb1kqoc/index.html
Oracle パーティション分割と価格決定ポリシー	http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf



Zone Clusters—How to Deploy
Virtual Clusters and Why
2011 年 2 月
著者: Tim Read
Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

海外からのお問い合わせ窓口:

Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com



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

Copyright © 2011, 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を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 と Java は、Oracle およびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

AMD、Opteron、AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。AMD、Opteron、AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。Intel、Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスに基づいて使用される SPARC International, Inc の商標または登録商標です。UNIX は X/Open Company, Ltd. からライセンス提供された登録商標です。0410

SOFTWARE. HARDWARE. COMPLETE.